LilyPond

Le système de gravure musicale

Manuel de notation

L'équipe de développement de LilyPond

Ce document constitue le manuel de notation de GNU LilyPond 2.14.2. Sa lecture requiert une familiarité avec le contenu présenté dans le Section "Manuel d'initiation" dans *Manuel d'initiation*.

Pour connaître la place qu'occupe ce manuel dans la documentation, consultez la page Section "Manuels" dans *Informations générales*.

Si vous ne disposez pas de certains manuels, la documentation complète se trouve sur `http://www.lilypond.org/`.

Copyright © 1999–2011 par les auteurs.

The translation of the following copyright notice is provided for courtesy to non-English speakers, but only the notice in English legally counts.

La traduction de la notice de droits d'auteur ci-dessous vise à faciliter sa compréhension par le lecteur non anglophone, mais seule la notice en anglais a valeur légale.

> Vous avez le droit de copier, distribuer et/ou modifier ce document selon les termes de la Licence GNU de documentation libre, version 1.1 ou tout autre version ultérieure publiée par la Free Software Foundation, "sans aucune section invariante". Une copie de la licence est fournie à la section "Licence GNU de documentation libre".
>
> Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.1 or any later version published by the Free Software Foundation; with no Invariant Sections. A copy of the license is included in the section entitled "GNU Free Documentation License".

Pour LilyPond version 2.14.2

Table des matières

1 Notation musicale générale 1

1.1 Hauteurs 1

1.1.1 Écriture des hauteurs de note 1

Hauteurs avec octave absolue 1

Octaves relatives 2

Altérations 5

Nom des notes dans d'autres langues 7

1.1.2 Modification de plusieurs hauteurs 8

Vérifications d'octave 9

Transposition 10

1.1.3 Gravure des hauteurs 13

Clefs 13

Armure 16

Marques d'octaviation 18

Instruments transpositeurs 19

Altérations accidentelles automatiques 20

Ambitus 27

1.1.4 Têtes de note 30

Têtes de note spécifiques 30

Têtes de note avec nom de note 31

Têtes de note à forme variable 33

Improvisation 36

1.2 Rythme 36

1.2.1 Écriture du rythme 37

Durées 37

Nolets 39

Changement d'échelle des durées 43

Liaisons de prolongation 44

1.2.2 Écriture des silences 47

Silences 47

Silences invisibles 49

Silences valant une mesure 50

1.2.3 Gravure du rythme 54

Métrique 54

Indication métronomique 59

Levées 62

Musique sans métrique 63

Notation polymétrique 64

Découpage automatique des notes 67

Gravure de lignes rythmiques 68

1.2.4 Barres de ligature 70

Barres de ligature automatiques 70

Définition des règles de ligature automatique 73

Barres de ligature manuelles 80

Liens de croches en soufflet 83

1.2.5 Mesures 84

Barres de mesure 84

Numéros de mesure 88

Vérification des limites et numéros de mesure . . . 92
Indications de repère . . . 93
1.2.6 Fonctionnalités rythmiques particulières . . . 95
Notes d'ornement . . . 95
Alignement et cadences . . . 100
Gestion du temps . . . 100
1.3 Signes d'interprétation . . . 101
1.3.1 Signes d'interprétation attachés à des notes . . . 102
Articulations et ornements . . . 102
Nuances . . . 104
Personnalisation des indications de nuance . . . 109
1.3.2 Signes d'interprétation sous forme de courbe . . . 111
Liaisons d'articulation . . . 111
Liaisons de phrasé . . . 114
Signes de respiration . . . 116
Chutes et sauts . . . 117
1.3.3 Signes d'interprétation sous forme de ligne . . . 117
Glissando . . . 118
Arpèges . . . 119
Trilles . . . 122
1.4 Répétitions et reprises . . . 124
1.4.1 Répétition d'un long passage . . . 124
Répétitions courantes . . . 124
Indications de reprise manuelles . . . 129
Répétitions explicites . . . 131
1.4.2 Autres types de répétition . . . 133
Répétitions de mesure . . . 133
Répétitions en trémolo . . . 135
1.5 Notes simultanées . . . 137
1.5.1 Monophonie . . . 137
Notes en accords . . . 137
Répétition d'accords . . . 138
Expressions simultanées . . . 139
Clusters . . . 140
1.5.2 Plusieurs voix . . . 141
Polyphonie sur une portée . . . 141
Styles de voix . . . 143
Résolution des collisions . . . 144
Regroupement automatique de parties . . . 148
Saisie de musique en parallèle . . . 151
1.6 Notation sur la portée . . . 153
1.6.1 Gravure des portées . . . 154
Initialisation de nouvelles portées . . . 154
Regroupement de portées . . . 155
Imbrication de regroupements de portées . . . 159
Séparation des systèmes . . . 160
1.6.2 Modification de portées individuelles . . . 161
Symbole de la portée . . . 162
Portées d'ossia . . . 164
Masquage de portées . . . 168
1.6.3 Écriture de parties séparées . . . 171
Noms d'instrument . . . 171
Citation d'autres voix . . . 174
Mise en forme d'une citation . . . 178

1.7 Annotations éditoriales ... 182
1.7.1 Dans la portée ... 182
Indication de la taille de fonte musicale ... 182
Doigtés ... 183
Dictée à trous ... 186
Coloration d'objets ... 186
Parenthèses ... 188
Hampes ... 188
1.7.2 Hors de la portée ... 189
Info-bulle ... 189
Quadrillage temporel ... 190
Crochets d'analyse ... 192
1.8 Texte ... 193
1.8.1 Ajout de texte ... 194
Commentaires textuels ... 194
Indication textuelle avec extension ... 195
Indications textuelles ... 197
Texte indépendant ... 201
1.8.2 Mise en forme du texte ... 202
Introduction au formatage de texte ... 202
Sélection de la fonte et de la taille ... 204
Alignement du texte ... 206
Éléments graphiques dans du texte formaté ... 210
Notation musicale dans du texte formaté ... 212
Texte avec sauts de page ... 215
1.8.3 Fontes ... 216
Tout savoir sur les fontes ... 216
Attribution d'une fonte en particulier ... 217
Choix des fontes par défaut ... 218

2 Notation spécialisée ... 219

2.1 Musique vocale ... 219
2.1.1 Vue d'ensemble de la musique vocale ... 219
Références en matière de musique vocale ... 219
Saisie des paroles ... 220
Alignement des paroles sur la mélodie ... 221
Durée automatique des syllabes ... 223
Durée explicite des syllabes ... 225
Plusieurs syllabes sur une note ... 226
Plusieurs notes pour une même syllabe ... 227
Traits d'union et de prolongation ... 230
2.1.2 Situations particulières en matière de paroles ... 231
Travail avec des paroles et variables ... 231
Positionnement vertical des paroles ... 232
Positionnement horizontal des syllabes ... 237
Paroles et reprises ... 239
Paroles alternatives ... 245
2.1.3 Couplets ... 247
Numérotation des couplets ... 247
Indication de nuance dans les couplets ... 247
Indication du personnage et couplets ... 248
Rythme différent selon le couplet ... 248
Paroles en fin de partition ... 250
Paroles sur plusieurs colonnes en fin de partition ... 251

2.1.4 Chansons 253
Références en matière de chanson 253
Feuille de chant 254
2.1.5 Chorale 254
Références en matière de chorale 254
Mise en forme d'une partition chorale 255
Division de voix 256
2.1.6 Opéras et musiques de scène 257
Références en matière d'opéra et musique de scène 258
Indication du rôle 258
Citation-repère 260
Musique parlée 264
Dialogue et musique 264
2.1.7 Chants liturgiques 266
Références en matière de chant liturgique 266
Cantiques et hymnes 266
Psalmodie 273
Mesure incomplète et musique liturgique 276
2.1.8 Musique vocale ancienne 278
2.2 Instruments utilisant des portées multiples 278
2.2.1 Vue d'ensemble des claviers 279
Généralités sur les instruments à clavier 279
Changement de portée manuel 280
Changement de portée automatique 281
Lignes de changement de portée 283
Hampes et changements de portée 283
2.2.2 Piano 285
Pédales de piano 285
2.2.3 Accordéon 286
Symboles de jeux 286
2.2.4 Harpe 290
Généralités sur la harpe 290
Pédales de harpe 290
2.3 Cordes non frettées 291
2.3.1 Vue d'ensemble de la notation pour cordes non frettées 291
Références en matière de cordes non frettées 292
Indications d'archet 292
Harmoniques 293
Snap (Bartók) pizzicato 294
2.4 Instruments à cordes frettées 294
2.4.1 Vue d'ensemble des cordes frettées 294
Références en matière de cordes frettées 294
Indications du numéro de corde 295
Tablatures par défaut 296
Tablatures personnalisées 302
Tablatures sous forme d'étiquette 304
Tablatures prédéfinies 312
Tablatures automatiques 320
Doigtés pour la main droite 323
2.4.2 Guitare 325
Indication de la position et du barré 325
Indication des harmoniques et notes étouffées 325
2.4.3 Banjo 326
Tablatures pour banjo 326

2.5 Percussions 326
2.5.1 Vue d'ensemble des percussions 326
Références en matière de notation pour percussions 326
Notation de base pour percussions 326
Portées de percussion 327
Notes fantômes 329
2.6 Instruments à vent 330
2.6.1 Vue d'ensemble des instruments à vent 330
Références en matière d'instruments à vent 330
Doigtés pour vents 331
2.6.2 Cornemuse 333
Définitions pour la cornemuse 334
Exemple pour la cornemuse 334
2.6.3 Bois 335
2.6.3.1 Diagrammes pour bois 335
2.7 Notation des accords 343
2.7.1 Mode accords 343
Généralités sur le mode accords 343
Accords courants 344
Extensions et altération d'accords 346
2.7.2 Gravure des accords 346
Impression des noms d'accord 346
Personnalisation des noms d'accord 350
2.7.3 Basse chiffrée 350
Introduction à la basse chiffrée 350
Saisie de la basse chiffrée 352
Gravure de la basse chiffrée 352
2.8 Contemporary music 352
2.8.1 Pitch and harmony in contemporary music 353
References for pitch and harmony in contemporary music 353
Microtonal notation 353
Contemporary key signatures and harmony 353
2.8.2 Contemporary approaches to rhythm 353
References for contemporary approaches to rhythm 353
Tuplets in contemporary music 353
Contemporary time signatures 353
Extended polymetric notation 353
Beams in contemporary music 353
Bar lines in contemporary music 353
2.8.3 Graphical notation 353
2.8.4 Contemporary scoring techniques 353
2.8.5 New instrumental techniques 353
2.8.6 Further reading and scores of interest 353
Books and articles on contemporary musical notation 353
Scores and musical examples 354
2.9 Notations anciennes 354
2.9.1 Introduction aux notations anciennes 354
Formes de notation ancienne prises en charge 354
2.9.2 Signes de note alternatifs 354
Têtes de note anciennes 355
Altérations anciennes 355
Silences anciens 356
Clefs anciennes 357
Crochets anciens 359

Métriques anciennes ... 359
2.9.3 Signes de note supplémentaires ... 360
Articulations anciennes ... 360
Guidons ... 361
Divisions ... 362
Ligatures ... 362
Ligatures mensurales ... 363
Neumes ligaturés grégoriens ... 364
2.9.4 Contextes prédéfinis ... 370
Contextes du chant grégorien ... 370
Les contextes de la musique mensurale ... 370
2.9.5 Transcription de musique mensurale ... 371
Éditions ancienne et moderne à partir d'une même source ... 371
Des incipits ... 371
Mise en forme de la musique mensurale ... 371
Transcription de chant grégorien ... 371
2.9.6 Notation éditoriale ... 371
Altérations accidentelles suggérées ... 371
Notation du rythme dans la musique baroque ... 372
2.10 Musiques du monde ... 372
2.10.1 Noms de note et altérations non-occidentaux ... 372
2.10.2 Musique arabe ... 373
Références pour la musique arabe ... 373
Noms des notes en arabe ... 374
Armures arabes ... 375
Métriques arabes ... 376
Exemple de musique arabe ... 377
Lectures complémentaires pour la musique arabe ... 378
2.10.3 Musique classique turque ... 378
Références pour la musique classique turque ... 378
Noms des notes en turc ... 379

3 Généralités en matière d'entrée et sortie ... 380

3.1 Agencement du code ... 380
3.1.1 Structure d'une partition ... 380
3.1.2 Plusieurs partitions dans un même ouvrage ... 381
3.1.3 Plusieurs éditions pour une même source ... 382
3.1.4 Nom des fichiers de sortie ... 383
3.1.5 Structure de fichier ... 384
3.2 Titres et entêtes ... 386
3.2.1 Création de titres ... 386
3.2.2 Titrages personnalisés ... 389
3.2.3 Référencement des numéros de page ... 390
3.2.4 Table des matières ... 391
3.3 Travail sur des fichiers texte ... 393
3.3.1 Insertion de fichiers LilyPond ... 393
3.3.2 Différentes éditions à partir d'une même source ... 394
Utilisation de variables ... 394
Utilisation de balises ... 396
Globalisation des réglages ... 399
3.3.3 Codage du texte ... 399
3.3.4 Affichage de notation au format LilyPond ... 400
3.4 Contrôle des sorties ... 401
3.4.1 Extraction de fragments musicaux ... 401

3.4.2 Ignorer des passages de la partition ... 401
3.4.3 Formats de sortie alternatifs ... 402
3.4.4 Changement des fontes musicales ... 402
3.5 Sortie MIDI ... 402
3.5.1 Création de fichiers MIDI ... 403
Noms d'instrument ... 403
3.5.2 Le bloc MIDI ... 405
3.5.3 Contenu de la sortie MIDI ... 406
Éléments pris en compte dans le MIDI ... 406
Éléments non pris en compte dans le MIDI ... 406
3.5.4 Répétitions et MIDI ... 406
3.5.5 Gestion des nuances en MIDI ... 407
Indications de nuance ... 407
Amplitude du volume en MIDI ... 408
Égalisation de plusieurs instruments (i) ... 409
Égalisation de plusieurs instruments (ii) ... 410
3.5.6 MIDI et percussions ... 411

4 Gestion de l'espace ... 412
4.1 Mise en forme de la page ... 412
4.1.1 Le bloc `\paper` ... 412
4.1.2 Format du papier et adaptation automatique ... 413
Format du papier ... 413
Adaptation automatique au format ... 414
4.1.3 Variables d'espacement vertical fixe ... 414
4.1.4 Variables d'espacement vertical fluctuant ... 415
Structure des variables d'espacement vertical fluctuant ... 415
Liste des variables d'espacement fluctuant ... 416
4.1.5 Variables d'espacement horizontal ... 417
Variables de marge et de largeur ... 417
Variables spécifiques pour l'impression recto-verso ... 418
Variables d'indentation et de décalage ... 419
4.1.6 Autres variables du bloc `\paper` ... 419
Variables de gestion des sauts de ligne ... 419
Variables de gestion des sauts de page ... 420
Variables de gestion des numéros de page ... 421
Variables supplémentaires ... 421
4.2 Mise en forme de la partition ... 422
4.2.1 Le bloc `\layout` ... 422
4.2.2 Définition de la taille de portée ... 423
4.3 Sauts ... 424
4.3.1 Sauts de ligne ... 424
4.3.2 Sauts de page ... 426
4.3.3 Optimisation des sauts de page ... 427
4.3.4 Optimisation des tournes ... 427
4.3.5 Minimisation des sauts de page ... 428
4.3.6 Sauts explicites ... 429
4.3.7 Recours à une voix supplémentaire pour gérer les sauts ... 430
4.4 Espacement vertical ... 432
4.4.1 Espacement vertical au sein d'un système ... 432
Propriétés d'espacement au sein d'un système ... 433
Espacement de portées isolées ... 436
Espacement de portées regroupées ... 437
Espacement des lignes rattachées à des portées ... 438

4.4.2 Positionnement explicite des portées et systèmes ... 439
4.4.3 Résolution des collisions verticales ... 446
4.5 Espacement horizontal ... 447
4.5.1 Généralités sur l'espacement horizontal ... 447
4.5.2 Changement d'espacement en cours de partition ... 449
4.5.3 Modification de l'espacement horizontal ... 449
4.5.4 Longueur de ligne ... 451
4.5.5 Notation proportionnelle ... 452
4.6 Réduction du nombre de pages de la partition ... 459
4.6.1 Mise en évidence de l'espacement ... 460
4.6.2 Modification de l'espacement ... 461

5 Modification des réglages prédéfinis ... 463
5.1 Contextes d'interprétation ... 463
5.1.1 Tout savoir sur les contextes ... 463
Score – le père de tous les contextes ... 463
Contextes de haut niveau – les systèmes ... 463
Contextes de niveau intermédiaire – les portées ... 463
Contextes de bas niveau – les voix ... 464
5.1.2 Création d'un contexte ... 464
5.1.3 Conservation d'un contexte ... 464
5.1.4 Modification des greffons de contexte ... 464
5.1.5 Modification des réglages par défaut d'un contexte ... 466
5.1.6 Définition de nouveaux contextes ... 467
5.1.7 Alignement des contextes ... 468
5.2 En quoi consiste la référence des propriétés internes ... 469
5.2.1 Navigation dans les références du programme ... 469
5.2.2 Interfaces de rendu ... 470
5.2.3 Détermination de la propriété d'un objet graphique (grob) ... 471
5.2.4 Conventions de nommage ... 472
5.3 Modification de propriétés ... 472
5.3.1 Vue d'ensemble de la modification des propriétés ... 472
5.3.2 La commande `\set` ... 472
5.3.3 La commande `\override` ... 474
5.3.4 La commande `\tweak` ... 474
5.3.5 `\set` ou `\override` ... 474
5.3.6 Modification de listes associatives ... 475
5.4 Propriétés et contextes utiles ... 475
5.4.1 Modes de saisie ... 475
5.4.2 Direction et positionnement ... 475
5.4.3 Ordre des contextes de mise en forme ... 475
5.4.4 Distances et unités de mesure ... 475
5.4.5 Propriétés des symboles de la portée ... 475
5.4.6 Extenseurs et prolongateurs ... 475
Utilisation de `spanner-interface` ... 475
Utilisation de `line-spanner-interface` ... 475
5.4.7 Visibilité des objets ... 475
Suppression des stencils ... 475
Transparence des objets ... 476
Blanchiment des objets ... 476
Utilisation de break-visibility ... 476
Considérations spécifiques ... 476
5.4.8 Styles de ligne ... 476
5.4.9 Rotation des objets ... 476

Rotation des objets de mise en forme ... 476
Rotation des étiquettes ... 476
5.5 Retouches avancées ... 476
5.5.1 Alignement des objets ... 476
Détermination directe de `X-offset` et `Y-offset` ... 476
Utilisation de `side-position-interface` ... 476
Utilisation de `self-alignment-interface` ... 476
Utilisation des procédures `centered-on-parent` ... 476
Utilisation de `break-aligned-interface` ... 476
5.5.2 Regroupement vertical d'objets graphiques ... 476
5.5.3 Modification des stencils ... 476
5.5.4 Modification de l'allure des éléments ... 476
Modification des liaisons ... 476
5.6 Utilisation de fonctions musicales ... 476
5.6.1 Syntaxe d'une fonction de substitution ... 476
5.6.2 Exemples de fonction de substitution ... 476

Annexe A Tables du manuel de notation ... 477

A.1 Table des noms d'accord ... 477
A.2 Modificateurs d'accord usuels ... 478
A.3 Diagrammes d'accord prédéfinis ... 481
A.4 Instruments MIDI ... 489
A.5 Liste des couleurs ... 489
A.6 La fonte Feta ... 491
Glyphes de clef ... 491
Glyphes de métrique ... 491
Glyphes de chiffre ... 492
Glyphes d'altération ... 492
Glyphes de tête de note par défaut ... 493
Glyphes de tête de note spéciale ... 493
Glyphes de tête de note à forme variable ... 494
Glyphes de silence ... 498
Glyphes de crochet de croche ... 499
Glyphes de point ... 499
Glyphes de nuance ... 499
Glyphes de script ... 500
Glyphes de flèche ... 502
Glyphes d'extrémité d'accolade ... 502
Glyphes de pédale ... 502
Glyphes d'accordéon ... 503
Glyphes de style vaticana ... 503
Glyphes de style medicaea ... 504
Glyphes de style Hufnagel ... 505
Glyphes de style mensural ... 505
Glyphes de style néomensural ... 508
Glyphes de style Petrucci ... 509
Glyphes de style Solesmes ... 509
A.7 Styles de tête de note ... 510
A.8 Text markup commands ... 510
A.8.1 Font ... 511
A.8.2 Align ... 519
A.8.3 Graphic ... 534
A.8.4 Music ... 540
A.8.5 Instrument Specific Markup ... 543

A.8.6 Other ... 547
A.9 Text markup list commands ... 552
A.10 Liste des signes d'articulation ... 553
Scripts d'articulation ... 553
Scripts d'ornement ... 554
Scripts de point d'orgue et point d'arrêt ... 554
Scripts spécifiques à certains instruments ... 554
Scripts de reprise et de répétition ... 554
Scripts pour musique ancienne ... 555
A.11 Notes utilisées en percussion ... 555
A.12 Glossaire technique ... 556
callback ... 556
glyph ... 556
grob ... 556
interface ... 556
lexer ... 556
output-def ... 556
parser ... 556
parser variable ... 556
prob ... 556
simple-closure ... 556
smob ... 556
stencil ... 556
A.13 Liste des propriétés de contexte ... 557
A.14 Propriétés de mise en forme ... 567
A.15 Fonctions mucicales ... 583
A.16 Types de prédicats prédéfinis ... 589
R5RS primary predicates ... 589
R5RS secondary predicates ... 589
Guile predicates ... 589
LilyPond scheme predicates ... 589
LilyPond exported predicates ... 590
A.17 Fonctions Scheme ... 591

Annexe B Aide-mémoire ... 612

Annexe C Grammaire de LilyPond ... 616

Annexe D GNU Free Documentation License ... 636
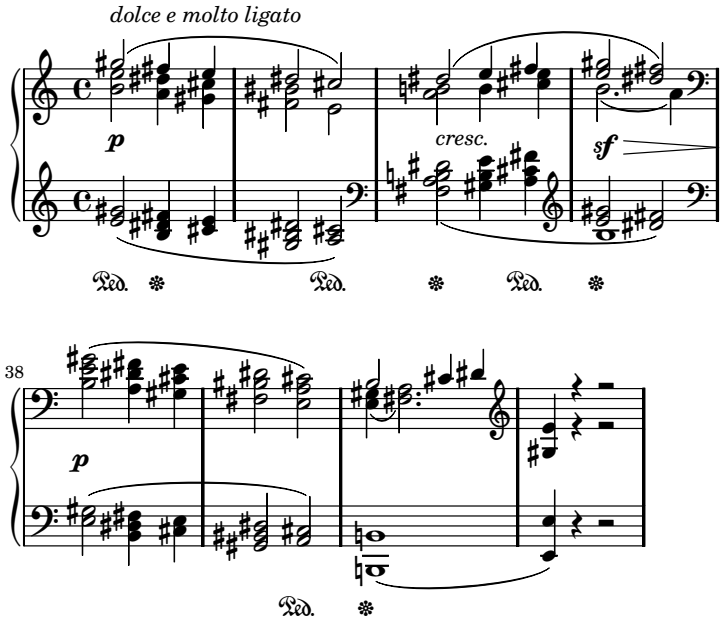
Annexe E Index des commandes LilyPond ... 643

Annexe F Index de LilyPond ... 652

1 Notation musicale générale

Ce chapitre explique comment créer la notation musicale standard.

1.1 Hauteurs



Cette section détaille la façon d'indiquer la hauteur des notes, sous trois aspects : la saisie des hauteurs, la modification des hauteurs et les options de gravure.
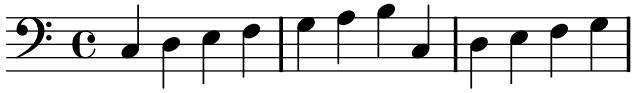
1.1.1 Écriture des hauteurs de note

Cette section explique la manière d'indiquer les hauteurs de note. Deux modes permettent d'indiquer l'octave des notes : le mode absolu, et le mode relatif. Ce dernier est le plus pratique lors de la saisie d'un fichier source au clavier de l'ordinateur.

Hauteurs avec octave absolue

La hauteur s'écrit – à moins de préciser une autre langue – avec la notation batave, en utilisant les lettres de `a` à `g`. Les notes `c` (do) et `b` (si) sont écrites une octave sous le do central.
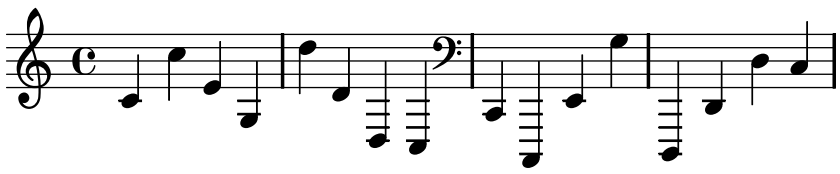
```
{
  \clef bass
  c4 d e f
  g4 a b c
  d4 e f g
}
```



L'octave peut être précisée sous forme d'une série d'apostrophes `'` ou d'une série de virgules `,`. Chaque `'` hausse la note d'une octave ; chaque `,` baisse la note d'une octave.

```
{
  \clef treble
  c'4 c'' e' g
```

```
  d''4 d' d c
  \clef bass
  c,4 c,, e, g
  d,,4 d, d c
}
```



Voir aussi

Glossaire musicologique : Section “Noms des notes” dans *Glossaire*.

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Octaves relatives

On spécifie les octaves en ajoutant `'` ou `,` aux noms de hauteur. En recopiant de la musique, on a vite fait de mettre une note à la mauvaise octave ; ce genre d’erreur est difficile à retrouver. Le mode d’écriture en octave relative prévient ces erreurs dans la mesure où elles deviennent beaucoup plus évidentes : une seule erreur décale le reste de la pièce à une mauvaise octave.
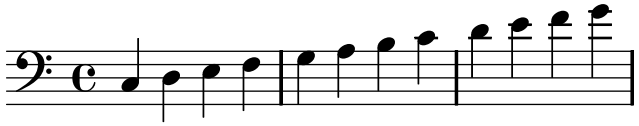
`\relative` *`hauteur_de_référence expression_musicale`*

En mode relatif, chaque note est considérée comme étant le plus proche possible de celle qui la précède. L’octave des notes mentionnées dans *expression_musicale* va être calculée de la manière suivante :

- Si aucun signe de changement d’octave n’est utilisé, l’intervalle de base entre la note actuelle et la précédente sera toujours au plus d’une quarte. Cet intervalle est déterminé sans tenir compte des altérations.
- Un signe de changement d’octave `'` ou `,` peut être ajouté pour hausser ou baisser la note d’une octave par rapport à la hauteur calculée sans spécification.
- Ces signes de changement d’octave peuvent être multipliés. Par exemple, `''` ou `,,` ajouteront une octave supplémentaire.
- Lorsque l’on entre en mode `\relative`, une hauteur absolue de départ peut être spécifiée, et agira dès lors comme si elle précédait la première note de *expression_musicale*. Nous vous recommandons de faire référence à un do (`c`).
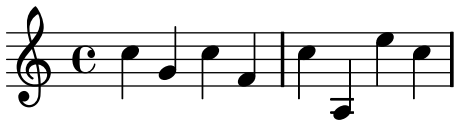
Voici le mode `\relative` en action.

```
\relative c {
  \clef bass
  c d e f
  g a b c
  d e f g
}
```
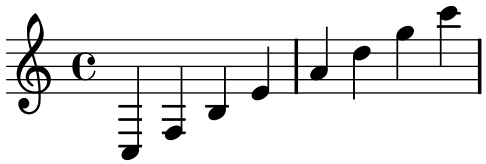


On utilise les signes de changement d’octave pour les intervalles dépassant la quarte.

```
\relative c'' {
  c g c f,
  c' a, e'' c
}
```
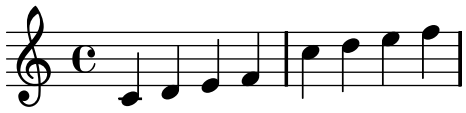


Bien que ne comportant aucun signe de changement d'octave, une séquence de notes peut tout à fait couvrir un intervalle important.

```
\relative c {
  c f b e
  a d g c
}
```



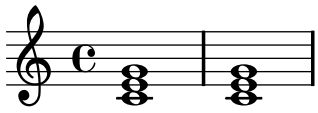
Lorsque plusieurs blocs `\relative` sont imbriqués, c'est la dernière clause `\relative` qui est prise en compte.

```
\relative c' {
  c d e f
  \relative c'' {
    c d e f
  }
}
```



`\relative` est sans effet sur un bloc `\chordmode`.
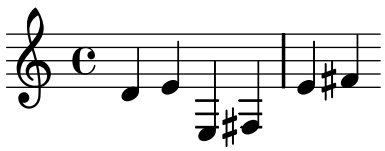
```
\new Staff {
  \relative c''' {
    \chordmode { c1 }
  }
  \chordmode { c1 }
}
```



`\relative` n'est pas permis au sein d'un bloc `\chordmode`.

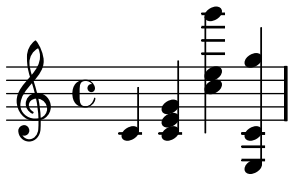
Pour utiliser le mode d'octave relative dans de la musique transposée, une clause `\relative` additionnelle doit être placée au sein du bloc `\transpose`.

```
\relative c' {
  d e
  \transpose f g {
    d e
    \relative c' {
      d e
    }
  }
}
```
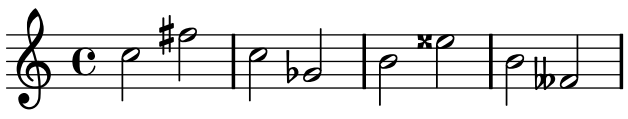


Si l'expression précédente est un accord, c'est la première note de l'accord qui détermine l'emplacement de la première note du prochain accord. À l'intérieur de l'accord, les notes sont placées relativement à celle qui précède. Examinez avec attention l'exemple suivant, et tout particulièrement le positionnement des do.

```
\relative c' {
  c
  <c e g>
  <c' e g'>
  <c, e, g''>
}
```



Comme nous l'avons vu, l'octaviation est déterminée sans tenir compte des altérations. Ainsi un mi double-dièse qui suit un si naturel sera placé au-dessus de celui-ci, alors qu'un fa double-bémol se retrouvera en dessous. En d'autres termes, une quarte doublement augmentée demeure considérée comme un intervalle plus petit qu'une quinte diminuée, bien que la quarte doublement augmentée soit de sept demi-tons et la quinte diminuée de seulement six demi-tons.

```
\relative c'' {
  c2 fis
  c2 ges
  b2 eisis
  b2 feses
}
```



Voir aussi

Glossaire musicologique : Section “quinte” dans *Glossaire*, Section “intervalle” dans *Glossaire*, Section “nom des notes” dans *Glossaire*.

Manuel de notation : [Vérifications d’octave], page 9.

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.
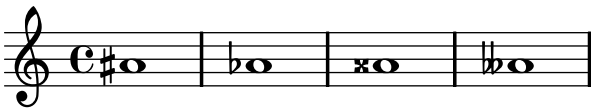
Référence des propriétés internes : Section “RelativeOctaveMusic” dans *Référence des propriétés internes*.

Altérations

> **Note :** Les nouveaux utilisateurs sont parfois déroutés par la gestion des altérations et de l’armure. Pour LilyPond, une hauteur n’est que du matériau brut ; l’armure et la clef ne feront que déterminer comment ce matériau sera retranscrit. Un simple `c` signifie tout bonnement « do naturel » quelles que soient l’armure et la clef en question. Pour plus d’information, reportez-vous au chapitre Section “Altérations et armure” dans *Manuel d’initiation*.
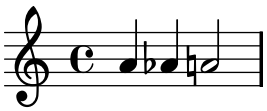
Dans la notation par défaut, un *dièse* est formé en ajoutant `is` après le nom de note, un *bémol* en ajoutant `es`. Les *double-dièses* et *double-bémols* sont obtenus en ajoutant respectivement `isis` ou `eses` au nom de note. Ce sont les noms de note hollandais. Pour les autres langues, consultez [Nom des notes dans d’autres langues], page 7.

```
ais1 aes aisis aeses
```



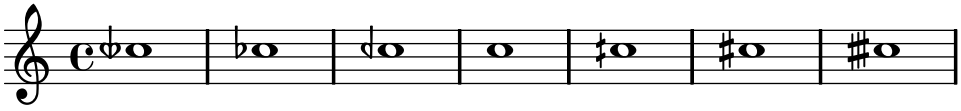
Un bécarre annule l’effet d’une altération, qu’elle soit accidentelle ou à l’armure. Cependant, dans la syntaxe des noms de note, les bécarres ne s’écrivent pas avec un suffixe ; un simple nom de note donnera une note bécarre.

```
a4 aes a2
```



Les demi-bémols et demi-dièses s’écrivent en ajoutant respectivement `eh` et `ih`. Voici une série de dos altérés en hauteurs croissantes :
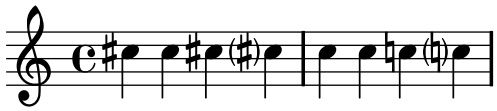
```
ceseh1 ces ceh c cih cis cisih
```



Les micro-intervalles sont aussi exportés dans le fichier MIDI.

Normalement, les altérations sont imprimées automatiquement, mais il se peut que vous vouliez les imprimer manuellement. On peut forcer l’impression d’une altération, dite « de précaution », en ajoutant un point d’exclamation `!` après la hauteur de note. Une altération entre parenthèses peut être obtenue en ajoutant un point d’interrogation `?` après la hauteur de note. Ces signes peuvent aussi être utilisés pour imprimer des bécarres.
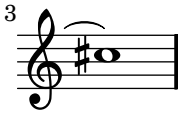
```
cis cis cis! cis? c c c! c?
```



Lorsqu'une note est prolongée par une liaison de tenue, l'altération ne sera réimprimée que s'il y a un saut de ligne.

```
cis1 ~ cis ~
\break
cis
```
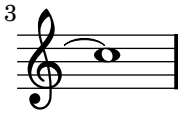




Morceaux choisis

*Non répétition de l'altération après saut de ligne*sur liaison de prolongation

Cet exemple illustre comment, lorsqu'une note affublée d'une altération accidentelle est prolongée, ne pas répéter cette altération après un saut de ligne.

```
\relative c'' {
  \override Accidental #'hide-tied-accidental-after-break = ##t
  cis1~ cis~
  \break
  cis
}
```
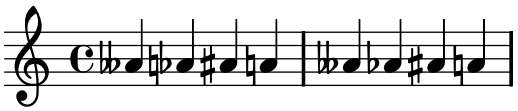




Suppression des bécarres superflus

En accord avec les règles standards de l'écriture musicale, on grave un bécarre avant un dièse ou un bémol si on a besoin d'annuler une altération précédente. Pour modifier ce comportement, assignez la propriété `extraNatural` du contexte `Staff` à la valeur `##f` (faux).

```
\relative c'' {
  aeses4 aes ais a
  \set Staff.extraNatural = ##f
  aeses4 aes ais a
}
```



Voir aussi

Glossaire musicologique : Section “dièse” dans *Glossaire*, Section “bémol” dans *Glossaire*, Section “double dièse” dans *Glossaire*, Section “double bémol” dans *Glossaire*, Section “Nom des notes” dans *Glossaire*, Section “quart de ton” dans *Glossaire*.

Manuel d’initiation : Section “Altérations et armure” dans *Manuel d’initiation*.

Manuel de notation : [Altérations accidentelles automatiques], page 20, [Altérations accidentelles suggérées], page 371, [Nom des notes dans d’autres langues], page 7.

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Références des propriétés internes : Section “Accidental_engraver” dans *Référence des propriétés internes*, Section “Accidental” dans *Référence des propriétés internes*, Section “AccidentalCautionary” dans *Référence des propriétés internes*, Section “accidental-interface” dans *Référence des propriétés internes*.

Problèmes connus et avertissements
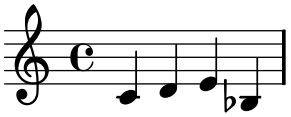
Il n’y a pas de standard universellement accepté pour noter le bémol et demi (qui abaisse la hauteur trois quarts de ton), le symbole de LilyPond n’est donc conforme à aucun standard.

Nom des notes dans d’autres langues

Vous disposez de jeux prédéfinis de noms de note et altérations pour plusieurs autres langues. Pour les utiliser, il suffit de déclarer, en début de fichier, la langue que vous utilisez. Voici comment, par exemple, utiliser l’italien pour votre saisie :

```
\language "italiano"

\relative do' {
  do re mi sib
}
```



Les langues disponibles ainsi que les noms de note utilisés sont les suivants :
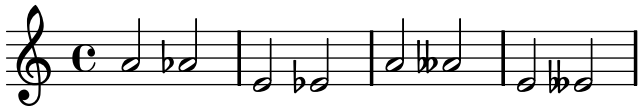
Langue	**Nom des notes**
`nederlands`	c d e f g a bes b
`catalan`	do re mi fa sol la sib si
`deutsch`	c d e f g a b h
`english`	c d e f g a bf b
`espanol`	do re mi fa sol la sib si
`italiano`	do re mi fa sol la sib si
`norsk`	c d e f g a b h
`portugues`	do re mi fa sol la sib si
`suomi`	c d e f g a b h
`svenska`	c d e f g a b h
`vlaams`	do re mi fa sol la sib si

et les suffixes d’altération correspondants :

Langue	dièse	bémol	double dièse	double bémol
nederlands	-is	-es	-isis	-eses
catalan	-d/-s	-b	-dd/-ss	-bb
deutsch	-is	-es	-isis	-eses
english	-s/-sharp	-f/-flat	-ss/-x/-sharpsharp	-ff/-flatflat
espanol	-s	-b	-ss/-x	-bb
italiano	-d	-b	-dd	-bb
norsk	-iss/-is	-ess/-es	-ississ/-isis	-essess/-eses
portugues	-s	-b	-ss	-bb
suomi	-is	-es	-isis	-eses
svenska	-iss	-ess	-ississ	-essess
vlaams	-k	-b	-kk	-bb

Notez qu'en hollandais, en allemand, en norvégien et en suédois, un *la* altéré de *bémol* se note `aes` et `aeses`. Ces formes sont contractées en `as` et `ases` (ou plus communément `asas`). Dans certains fichiers linguistiques, seules ces formes abrégées ont été définies.

```
a2 as e es a ases e eses
```



Certaines musiques utilisent des microtonalités, pour lesquelles les altérations sont des fractions de dièse ou bémol « normaux ». Le tableau suivant répertorie les noms de note en quart de ton, tels que définis dans plusieurs fichiers linguistiques. Les préfixes *semi-* et *sesqui-* correspondent au *demi-* et *trois demis*. À noter qu'aucune définition n'existe à ce jour pour le norvégien, le suédois, le catalan et l'espagnol.

Langue	semi-dièse	semi-bémol	sesqui-dièse	sesqui-bémol
nederlands	-ih	-eh	-isih	-eseh
deutsch	-ih	-eh	-isih	-eseh
english	-qs	-qf	-tqs	-tqf
espanol	-cs	-cb	-tcs	-tcb
italiano	-sd	-sb	-dsd	-bsb
portugues	-sqt	-bqt	-stqt	-btqt

La plupart des langues dont nous venons de parler correspondent à la musique classique occidentale au tempérament égal – le concept de *Common Practice Period* en anglais. LilyPond prend néanmoins en charge d'autres systèmes de notation, comme indiqué au chapitre Section 2.10.1 [Noms de note et altérations non-occidentaux], page 372.

Voir aussi

Glossaire musicologique : Section "Nom des notes" dans *Glossaire*, Section "Common Practice Period" dans *Glossaire*.

Manuel de notation : Section 2.10.1 [Noms de note et altérations non-occidentaux], page 372.

Fichiers d'initialisation : '`scm/define-note-names.scm`'.

Morceaux choisis : Section "Hauteurs" dans *Morceaux choisis*.

1.1.2 Modification de plusieurs hauteurs
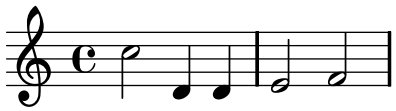
Cette partie traite de la manière de modifier les hauteurs de note.

Vérifications d'octave

Les tests d'octave rendent la correction d'erreurs d'octave plus facile dans le mode d'octave `relative` – un `,` ou un `'` oublié, ça n'arrive pas qu'aux autres !
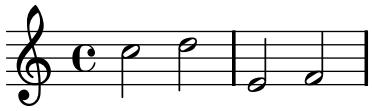
Une note peut être suivie de =*apostrophes/virgules* pour indiquer à quelle **octave absolue** elle devrait être. Dans l'exemple suivant, le premier `d` générera un avertissement, puisqu'on attend un `d''` – intervalle inférieur à la quarte – mais qu'on obtient un `d'`. Sur la partition, l'octave sera corrigée pour donner un `d'` et la prochaine note sera calculée en fonction de ce `d'` et non de `d''`.

```
\relative c'' {
  c2 d='4 d
  e2 f
}
```



Il existe aussi une vérification d'octave qui ne produit pas de musique imprimée, ayant pour syntaxe `\octaveCheck` *hauteur_référence* – ***`hauteur_référence`*** étant spécifiée en mode absolu. Cette commande vérifie que l'intervalle entre la note qui précède et ***`hauteur_référence`*** est inférieur à la quinte comme il se doît en mode relatif. Dans le cas contraire, un message sera émis et la note précédente ne sera pas modifiée. Les notes suivantes seront, par contre, positionnées relativement à ***`hauteur_référence`***.
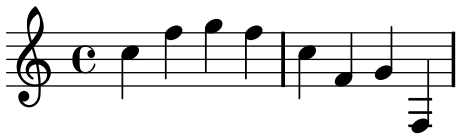
```
\relative c'' {
  c2 d
  \octaveCheck c'
  e2 f
}
```



Dans les deux mesures qui suivent, les premier et troisième `\octaveCheck` échouent, mais le deuxième est concluant.

```
\relative c'' {
  c4 f g f

  c4
  \octaveCheck c'
  f
  \octaveCheck c'
  g
  \octaveCheck c'
  f
}
```



Voir aussi

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “RelativeOctaveCheck” dans *Référence des propriétés internes*.

Transposition

Une expression musicale peut être transposée avec `\transpose`. En voici la syntaxe :
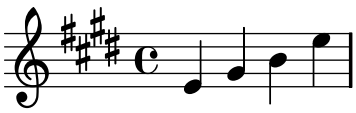
```
\transpose note_de_départ note_d_arrivée expression_musicale
```

Cela signifie que *`expression_musicale`* est transposé de l’intervalle séparant *`note_de_départ`* et *`note_d_arrivée`* : toute note dont la hauteur était `note_de_départ` est changée en `note_d_arrivée` ; les autres notes seront changées selon le même intervalle. Les deux hauteurs s’expriment en octave absolue.

> **Note :** La musique contenue dans un bloc `\transpose` est en octaves absolus, sauf à inclure dans ce même bloc une clause `\relative`.
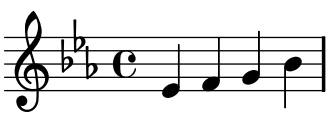
Prenons comme exemple une pièce écrite en ré majeur. Si cette pièce est un peu trop basse pour l’interprète, elle peut être transposée en mi majeur. Vous noterez que l’armure est automatiquement modifiée.

```
\transpose d e {
  \relative c' {
    \key d \major
    d4 fis a d
  }
}
```



Regardons maintenant une partie écrite pour violon – un instrument en ut. Si cette partie doit être jouée par une clarinette en la (écrite à la tierce mineure supérieure, un do écrit donnant un la réel), la transposition suivante créera la partie appropriée.
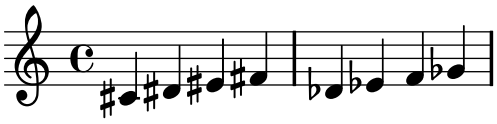
```
\transpose a c' {
  \relative c' {
    \key c \major
    c4 d e g
  }
}
```



La présence de `\key c \major` s’explique par le fait que, bien que les notes soient effectivement transposées, l’armure ne sera imprimée que dans la mesure où elle est explicitement mentionnée.

`\transpose` fait la distinction entre les notes enharmoniques : `\transpose c cis` et `\transpose c des` transposeront la pièce un demi-ton plus haut, au détail près que la première version écrira des dièses et la deuxième des bémols.

```
music = \relative c' { c d e f }
\new Staff {
  \transpose c cis { \music }
  \transpose c des { \music }
}
```



On peut aussi utiliser `\transpose` pour entrer des notes écrites pour un instrument transpositeur. Normalement, les hauteurs dans LilyPond sont écrites en ut, c'est à dire en sons réels, mais elles peuvent être écrites dans un autre ton. Prenons l'exemple d'un morceau pour trompette en si bémol commençant sur un ré à l'oreille ; on pourrait écrire

```
musiqueEnSiBemol = { e4 ... }
\transpose c bes, \musiqueEnSiBemol
```

Pour imprimer cette musique en fa – et de ce fait produire une partie de cor au lieu d'un conducteur en notes réelles – on utilisera un deuxième `\transpose` :

```
musiqueEnSiBemol = { e4 ... }
\transpose f c' { \transpose c bes, \musiqueEnSiBemol }
```

Pour plus d'information à ce sujet, consultez [Instruments transpositeurs], page 19.

Morceaux choisis

Transposition et réduction du nombrer d'altérations accidentelles

Cet exemple, grâce à un peu de code Scheme, donne la priorité aux enharmoniques afin de limiter le nombre d'altérations supplémentaires. La règle appliquable est :

- Les altérations doubles sont supprimées
- Si dièse -> Do
- Mi dièse -> Fa
- Do bémol -> Si
- Fa bémol -> Mi

Cette façon de procéder aboutit à plus d'enharmoniques naturelles.

```
#(define (naturalize-pitch p)
   (let ((o (ly:pitch-octave p))
         (a (* 4 (ly:pitch-alteration p)))
         ;; alteration, a, in quarter tone steps,
         ;; for historical reasons
         (n (ly:pitch-notename p)))
     (cond
      ((and (> a 1) (or (eq? n 6) (eq? n 2)))
       (set! a (- a 2))
       (set! n (+ n 1)))
      ((and (< a -1) (or (eq? n 0) (eq? n 3)))
       (set! a (+ a 2))
       (set! n (- n 1))))
     (cond
      ((> a 2) (set! a (- a 4)) (set! n (+ n 1)))
      ((< a -2) (set! a (+ a 4)) (set! n (- n 1))))
     (if (< n 0) (begin (set! o (- o 1)) (set! n (+ n 7))))
```
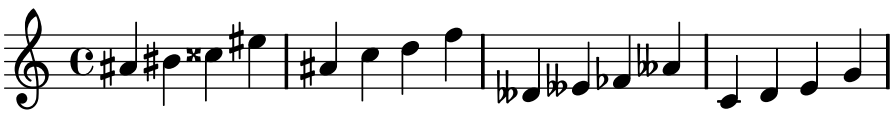
```
      (if (> n 6) (begin (set! o (+ o 1)) (set! n (- n 7))))
      (ly:make-pitch o n (/ a 4))))

#(define (naturalize music)
   (let ((es (ly:music-property music 'elements))
         (e (ly:music-property music 'element))
         (p (ly:music-property music 'pitch)))
     (if (pair? es)
         (ly:music-set-property!
          music 'elements
          (map (lambda (x) (naturalize x)) es)))
     (if (ly:music? e)
         (ly:music-set-property!
          music 'element
          (naturalize e)))
     (if (ly:pitch? p)
         (begin
           (set! p (naturalize-pitch p))
           (ly:music-set-property! music 'pitch p)))
     music))

naturalizeMusic =
#(define-music-function (parser location m)
   (ly:music?)
   (naturalize m))

music = \relative c' { c4 d e g }

\score {
  \new Staff {
    \transpose c ais { \music }
    \naturalizeMusic \transpose c ais { \music }
    \transpose c deses { \music }
    \naturalizeMusic \transpose c deses { \music }
  }
  \layout { }
}
```



Voir aussi

Manuel de notation : [Octaves relatives], page 2, [Instruments transpositeurs], page 19.

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriété internes : Section “TransposedMusic” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Si vous voulez utiliser en même temps `\transpose` et `\relative`, vous devez mettre `\transpose` en dehors de `\relative`, puisque `\relative` n'aura aucun effet sur la musique apparaissant dans un `\transpose`.
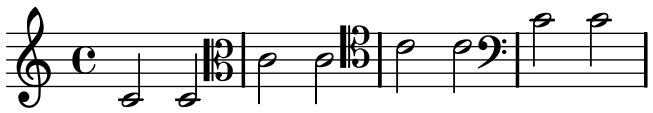
1.1.3 Gravure des hauteurs

Nous allons voir dans cette partie comment influer sur la gravure des hauteurs.

Clefs

La clef indique quelles lignes de la portée correspondent à quelles hauteurs. Elle peut changer au fil du morceau, comme dans l'exemple suivant qui indique le do médium dans différentes clefs :

```
\clef treble
c2 c
\clef alto
c2 c
\clef tenor
c2 c
\clef bass
c2 c
```



LilyPond met à votre disposition d'autres clefs :
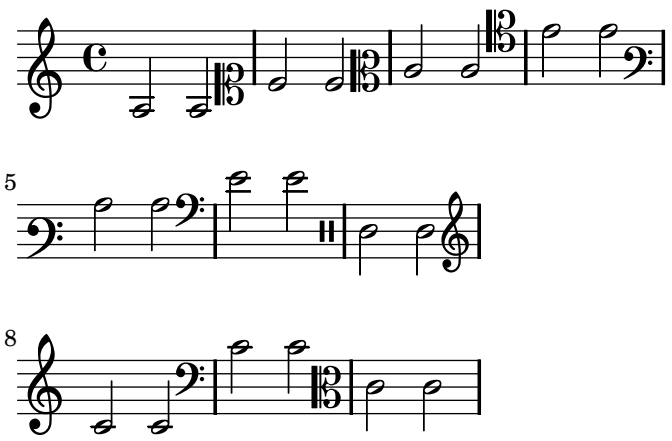
```
\clef french
c2 c
\clef soprano
c2 c
\clef mezzosoprano
c2 c
\clef baritone
c2 c

\break

\clef varbaritone
c2 c
\clef subbass
c2 c
\clef percussion
c2 c

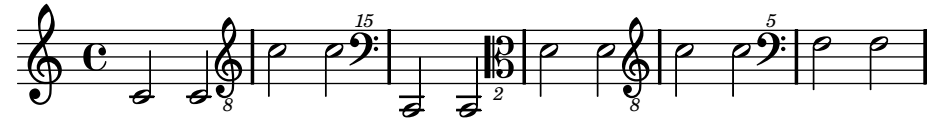
\break

\clef G   % synonym for treble
c2 c
\clef F   % synonym for bass
c2 c
\clef C   % synonym for alto
c2 c
```



En ajoutant `_8` ou `^8` au nom de la clé, celle-ci est transposée à l'octave respectivement inférieure ou supérieure, et `_15` ou `^15` la transpose de deux octaves. D'autres nombres entiers peuvent être utilisés selon les besoins. L'argument *clefname* doit être mis entre guillemets lorsqu'il contient des caractères supplémentaires. Par exemple,

```
\clef treble
c2 c
\clef "treble_8"
c2 c
\clef "bass^15"
c2 c
\clef "alto_2"
c2 c
\clef "G_8"
c2 c
\clef "F^5"
c2 c
```



Il existe d'autres clefs, destinées à un usage particulier. Voir à ce sujet Section 2.9 [Notations anciennes], page 354 [Tablatures par défaut], page 296 et [Tablatures personnalisées], page 302.

Morceaux choisis

Affinage des propriétés d'une clef

La commande `\clef "treble_8"` équivaut à définir `clefGlyph`, `clefPosition` – qui contrôle la position verticale de la clé – `middleCPosition` et `clefOctavation`. Une clé est imprimée lorsque l'une de ces propriétés, hormis `middleCPosition`, est modifiée. Les exemples suivant font apparaître des possibilités de réglage manuel de ces propriétés.

Modifier le glyphe, la position de la clef ou son octaviation ne changera pas la position des notes ; il faut pour y parvenir modifier aussi la position du do médium. Le positionnement est relatif à la ligne médiane, un nombre positif faisant monter, chaque ligne ou interligne comptant pour 1. La valeur de `clefOctavation` devrait être de 7, -7, 15 ou -15, bien que rien ne vous empêche de lui affecter une autre valeur.

Lorsqu'un changement de clef intervient en même temps qu'un saut de ligne, la nouvelle clef est imprimer à la fois en fin de ligne et au début de la suivante. Vous pouvez toujours supprimer cette « clef de précaution » en affectant la valeur `end-of-line-invisible` à la propriété `explicitClefVisibility` du contexte `Staff`. Le comportement par défaut sera réactivé par `\unset Staff.explicitClefVisibility`.

Les exemples qui suivent illustrent les différentes possibilités de définir ces propriétés manuellement. Sur la première ligne, la position relative des notes par rapport aux clefs sont préservées, ce qui n'est pas le cas pour la deuxième ligne.

```
\layout { ragged-right = ##t }

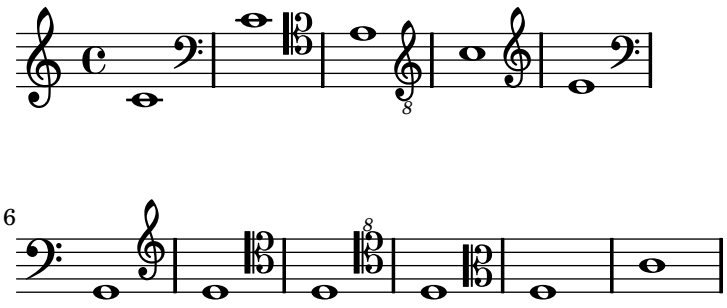
{
  % The default treble clef
  c'1
  % The standard bass clef
  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  \set Staff.middleCPosition = #6
  c'1
  % The baritone clef
  \set Staff.clefGlyph = #"clefs.C"
  \set Staff.clefPosition = #4
  \set Staff.middleCPosition = #4
  c'1
  % The standard choral tenor clef
  \set Staff.clefGlyph = #"clefs.G"
  \set Staff.clefPosition = #-2
  \set Staff.clefOctavation = #-7
  \set Staff.middleCPosition = #1
  c'1
  % A non-standard clef
  \set Staff.clefPosition = #0
  \set Staff.clefOctavation = #0
  \set Staff.middleCPosition = #-4
  c'1 \break

  % The following clef changes do not preserve
  % the normal relationship between notes and clefs:

  \set Staff.clefGlyph = #"clefs.F"
  \set Staff.clefPosition = #2
  c'1
  \set Staff.clefGlyph = #"clefs.G"
  c'1
  \set Staff.clefGlyph = #"clefs.C"
  c'1
  \set Staff.clefOctavation = #7
  c'1
  \set Staff.clefOctavation = #0
  \set Staff.clefPosition = #0
  c'1

  % Return to the normal clef:

  \set Staff.middleCPosition = #0
  c'1
}
```



Voir aussi

Manuel de notation : Section 2.9 [Notations anciennes], page 354, [Tablatures par défaut], page 296, [Tablatures personnalisées], page 302.

Morceaux choisis: Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Clef_engraver” dans *Référence des propriétés internes*, Section “Clef” dans *Référence des propriétés internes*, Section “OctavateEight” dans *Référence des propriétés internes*, Section “clef-interface” dans *Référence des propriétés internes*.

Armure

> **Note :** Les nouveaux utilisateurs sont parfois déroutés par la gestion des altérations et de l’armure. Pour LilyPond, une hauteur n’est que du matériau brut ; l’armure et la clef ne feront que déterminer comment ce matériau sera retranscrit. Un simple `c` signifie tout bonnement « do naturel » quelles que soient l’armure et la clef en question. Pour plus d’information, reportez-vous au chapitre Section “Altérations et armure” dans *Manuel d’initiation*.
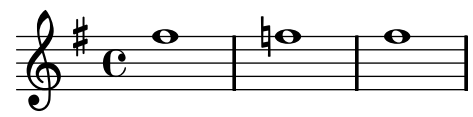
L’armure indique la tonalité dans laquelle la pièce doit être jouée. Elle comprend un ensemble d’altérations (dièses ou bémols) à la clef, c’est-à-dire au début de la portée. Elle peut varier en cours de morceau.

On définit ou modifie l’armure avec la commande `\key`

```
\key hauteur mode
```

Ici, *mode* doit être `\major` ou `\minor` afin d’avoir respectivement *hauteur*-majeur ou *hauteur*-mineur. Vous pouvez aussi avoir recours aux modes anciens que sont `\ionian`, `\locrian`, `\aeolian`, `\mixolydian`, `\lydian`, `\phrygian` et `\dorian`.

```
\key g \major
fis1
f
fis
```
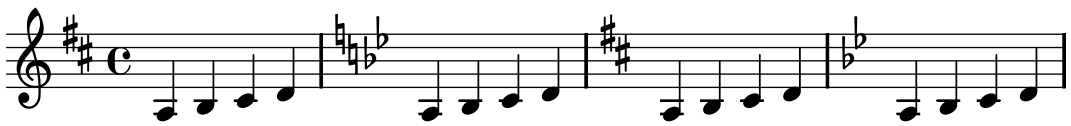


Morceaux choisis

*Suppression des bécarres superflus après un changement de*tonalité

Après un changement de tonalité, un bécarre est imprimé pour annuler toute altération précédente. Ceci peut être supprimé en réglant à `"false"` la propriété `printKeyCancellation` du contexte `Staff`.

```
\relative c' {
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
  \set Staff.printKeyCancellation = ##f
  \key d \major
  a4 b cis d
  \key g \minor
  a4 bes c d
}
```



Armures inhabituelles

La commande `\key` détermine la propriété `keySignature` d'un contexte `Staff`.

Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété. Il s'agit en l'occurence de définir une liste :
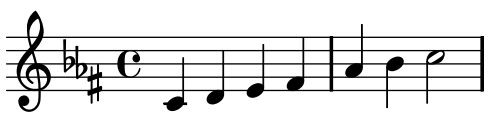
`\set Staff.keySignature = #`(((octave . pas) . altération) ((octave . pas) . altération) ...)`

dans laquelle, et pour chaque élément, `octave` spécifie l'octave (0 pour celle allant du do médium au si supérieur), `pas` la note dans cette octave (0 pour do et 6 pour si), et `altération` sera `,SHARP ,FLAT ,DOUBLE-SHARP` etc. (attention à la virgule en préfixe).

Une formulation abrégée – `(pas . altération)` – signifie que l'altération de l'élément en question sera valide quel que soit l'octave.

Voici, par exemple, comment générer une gamme par ton :

```
\relative c' {
  \set Staff.keySignature = #`(((0 . 6) . ,FLAT)
                              ((0 . 5) . ,FLAT)
                              ((0 . 3) . ,SHARP))
  c4 d e fis
  aes4 bes c2
}
```



Voir aussi

Glossaire musicologique : Section “mode d’église” dans *Glossaire*, Section “scordatura” dans *Glossaire*.

Manuel d’initiation : Section “Altérations et armure” dans *Manuel d’initiation*.

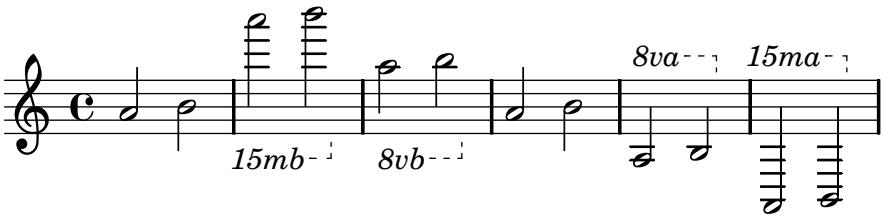
Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “KeyChangeEvent” dans *Référence des propriétés internes*, Section “Key_engraver” dans *Référence des propriétés internes*, Section “Key_performer” dans *Référence des propriétés internes*, Section “KeyCancellation” dans *Référence des propriétés internes*, Section “KeySignature” dans *Référence des propriétés internes*, Section “key-cancellation-interface” dans *Référence des propriétés internes*, Section “key-signature-interface” dans *Référence des propriétés internes*.

Marques d’octaviation

Les marques d’octaviation, *Ottava*, permettent d’introduire une transposition spécifique d’une octave pour la portée en cours. C’est la fonction `ottava` qui s’en charge.

```
a2 b
\ottava #-2
a2 b
\ottava #-1
a2 b
\ottava #0
a2 b
\ottava #1
a2 b
\ottava #2
a2 b
```

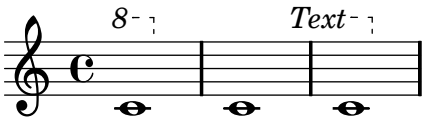


Morceaux choisis

Texte des marques d’octaviation

En interne, la fonction `\ottava` détermine les propriétés `ottavation` (p.ex. en `"8va"` ou `"8vb"`) et `centralCPosition`. Vous pouvez modifier le texte d’une marque d’octaviation en définissant `ottavation` après avoir fait appel à `ottava` :

```
{
  \ottava #1
  \set Staff.ottavation = #"8"
  c''1
  \ottava #0
  c'1
  \ottava #1
  \set Staff.ottavation = #"Text"
  c''1
}
```



Voir aussi

Glossaire musicologique : Section “octaviation” dans *Glossaire*.

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Ottava_spanner_engraver” dans *Référence des propriétés internes*, Section “OttavaBracket” dans *Référence des propriétés internes*, Section “ottava-bracket-interface” dans *Référence des propriétés internes*.

Instruments transpositeurs

Lorsque l’on saisit une partition d’ensemble incluant des instruments transpositeurs, certaines parties peuvent être dans une autre tonalité que la *tonalité de concert*. Il faudra en pareil cas indiquer la tonalité spécifique de ces *instruments transpositeurs*, sous peine de fichier MIDI erroné et de citations incorrectes. Pour plus de détails sur les citations, consultez le chapitre [Citation d’autres voix], page 174.

`\transposition` *`hauteur`*

La hauteur donnée en argument à `\transposition` doit correspondre à la note entendue lorsqu’un `do` écrit sur la portée est joué par l’instrument transpositeur. Cette hauteur doit être mentionnée en *mode absolu*. Par exemple, lorsque vous saisissez une partition en notes réelles, toutes les voix devraient être en ut ; si un instrument joue un ton au dessus, il faudra lui ajouter un `\transposition d'`. La commande `\transposition` s’utilise **si et seulement si** les notes à saisir **ne sont pas** dans la tonalité de concert.
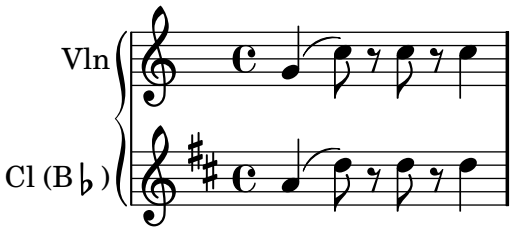
Voici un fragment pour violon et clarinette en si bémol (*B-flat*) pour lequel les parties respectives ont été recopiées à partir du conducteur. Les deux instruments sont à l’unisson.

```
\new GrandStaff <<
  \new Staff = "violin" {
    \relative c'' {
      \set Staff.instrumentName = #"Vln"
      \set Staff.midiInstrument = #"violin"
      % not strictly necessary, but a good reminder
      \transposition c'

      \key c \major
      g4( c8) r c r c4
    }
  }
  \new Staff = "clarinet" {
    \relative c'' {
      \set Staff.instrumentName = \markup { Cl (B\flat) }
      \set Staff.midiInstrument = #"clarinet"
      \transposition bes

      \key d \major
      a4( d8) r d r d4
    }
```

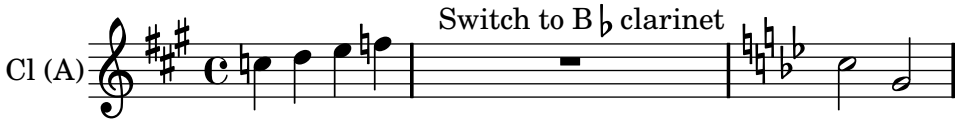
```
  }
>>
```



La `\transposition` peut évoluer au cours d'un morceau. Un clarinettiste peut être amené à jongler entre une clarinette en la et une autre en si bémol.

```
\set Staff.instrumentName = #"Cl (A)"
\key a \major
\transposition a
c d e f
\textLengthOn
s1*0^\markup { Switch to B\flat clarinet }
R1

\key bes \major
\transposition bes
c2 g
```



Voir aussi

Glossaire musicologique : Section "tonalité de concert" dans *Glossaire*, Section "instrument transpositeur" dans *Glossaire*.

Manuel de notation : [Citation d'autres voix], page 174, [Transposition], page 10.

Morceaux choisis : Section "Hauteurs" dans *Morceaux choisis*.

Altérations accidentelles automatiques

LilyPond dispose d'une fonction chargée de regrouper les règles suivant lesquelles s'impriment les altérations. Elle s'invoque de la manière suivante :

```
\new Staff <<
  #(set-accidental-style 'voice)
  { ... }
>>
```

La règle de gestion des altérations s'applique par défaut au contexte `Staff` en cours, exception faite des styles `piano` et `piano-cautionary` comme nous allons le voir. Cette fonction accepte un éventuel argument supplémentaire chargé de spécifier le champ d'action de la règle à suivre. À titre d'exemple, il faudra utiliser, pour que toutes les portées d'un même système – contexte `StaffGroup` – soient soumises à la même règle :

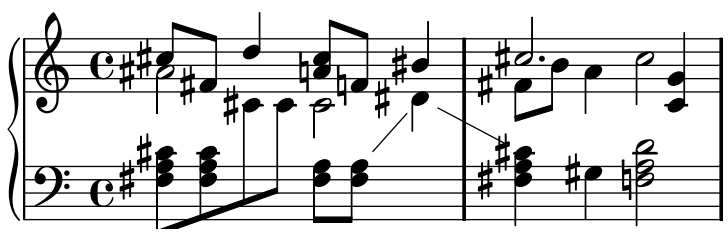
```
#(set-accidental-style 'voice 'StaffGroup)
```

Nous vous présentons ci-après les différentes régles d'altération prises en charge. Pour les besoins de la démonstration, nous partirons de l'exemple suivant :

```
musicA = {
  <<
    \relative c' {
      cis'8 fis, d'4 <a cis>8 f bis4 |
      cis2. <c, g'>4 |
    }
    \\
    \relative c' {
      ais'2 cis, |
      fis8 b a4 cis2 |
    }
  >>
}

musicB = {
  \clef bass
  \new Voice {
    \voiceTwo \relative c' {
      <fis, a cis>8 <fis a cis>
      \change Staff = up
      cis' cis
      \change Staff = down
      <fis, a> <fis a>
      \showStaffSwitch
      \change Staff = up
      dis'4 |
      \change Staff = down
      <fis, a cis>4 gis <f a d>2 |
    }
  }
}

\new PianoStaff {
  <<
    \context Staff = "up" {
      #(set-accidental-style 'default)
      \musicA
    }
    \context Staff = "down" {
      #(set-accidental-style 'default)
      \musicB
    }
  >>
}
```

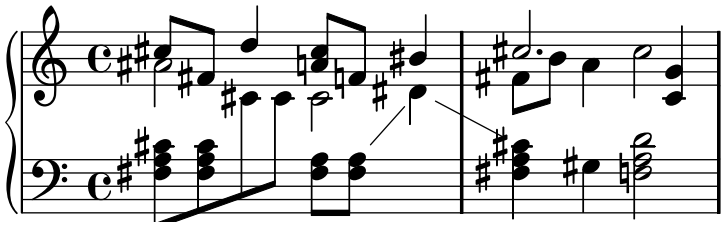


Notez bien que pour appliquer le même style aux deux portées, seules les dernières lignes de cet exemple nous intéressent.

```
\new PianoStaff {
  <<
    \context Staff = "haut" {
      %%% voici la ligne à modifier en conséquence :
      #(set-accidental-style 'default 'Score)
      \musicA
    }
    \context Staff = "bas" {
      \musicB
    }
  >>
}
```

`default`

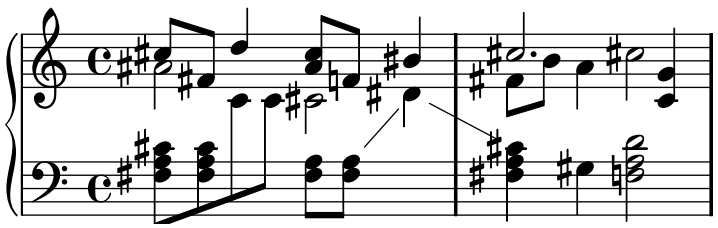
C'est la règle d'impression par défaut, qui se rapporte à l'usage en vigueur au XVIIIe siècle : les altérations accidentelles sont valables tout une mesure, et uniquement à leur propre octave. C'est la raison pour laquelle il n'y a pas de bécarre avant le *si* de la deuxième mesure, ni avant le dernier *do*.



`voice`

En principe, LilyPond se souvient de toutes les altérations présentes sur la portée (contexte `Staff`). Avec cette règle, cependant, les altérations sont indépendantes pour chacune des voix tout en obéissant à la règle `default`.

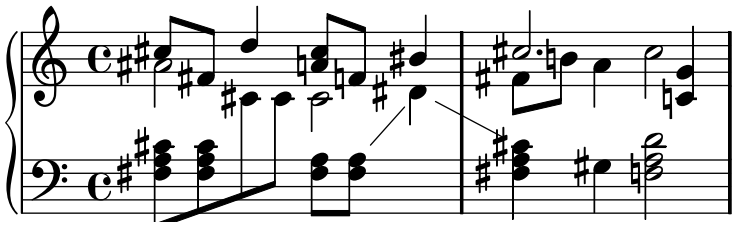
Les altérations d'une voix sont de fait ignorées dans les autres voix, ce qui peut donner lieu à un résultat malencontreux. Dans l'exemple suivant, il est difficile de dire si le deuxième *la* est dièse ou naturel. La règle `voice` n'est donc à envisager que dans le cas de voix devant être lues par des musiciens différents. S'il s'agit d'un « conducteur », ou d'une portée destinée à un seul musicien, il vaut mieux utiliser `modern` ou `modern-cautionary`.



`modern`

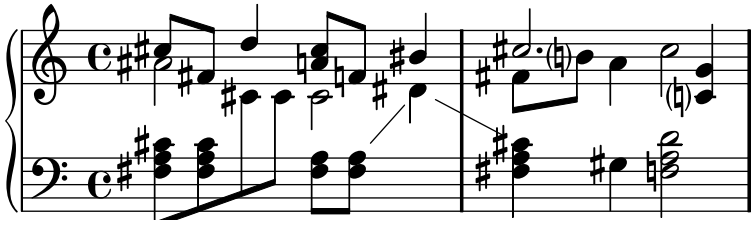
Cette règle est la plus courante au XXe siècle. Les altérations accidentelles sont imprimées comme avec le style `default`, mais lorsqu'une note non-altérée apparaît à une octave différente, ou bien dans la mesure suivante, des bécarres de précaution

sont ajoutés. Dans l'exemple suivant, notez ainsi les deux bécarres dans la deuxième mesure de la main droite.



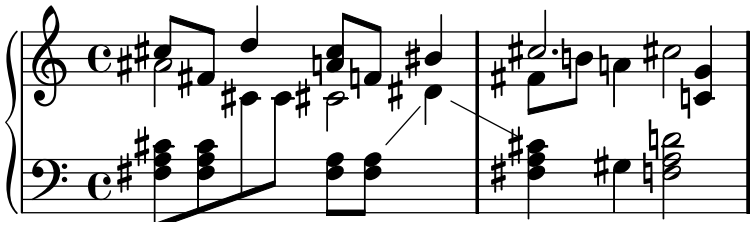
`modern-cautionary`

Cette règle est équivalente à `modern`, mais les bécarres de précaution (absents dans la règle `default`) sont imprimés de façon particulière : soit plus petit, soit (par défaut) entre parenthèses – il est possible de le définir au moyen de la propriété `cautionary-style` de l'objet `AccidentalSuggestion`.



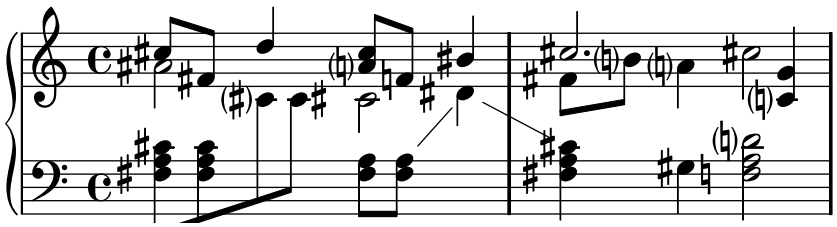
`modern-voice`

Cette règle sert aux altérations dans de la musique polyphonique destinée autant à des musiciens différents qu'à quelqu'un qui lirait l'ensemble des voix. Les altérations sont imprimées voix par voix, mais les autres voix d'un même contexte `Staff` en *tiennent compte* cette fois. C'est pourquoi le *la* de la dernière mesure est affublé d'un bécarre bien qu'il y en ait déjà eu un dans la mesure précédente, et que le *ré* de la main gauche en ait un alors qu'il avait auparavant concerné la main droite.



`modern-voice-cautionary`

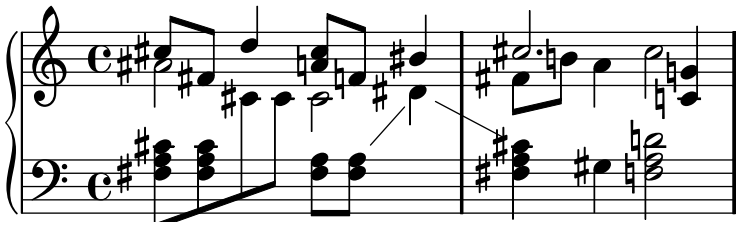
Cette régle est similaire à la précédente, mais les altérations de précaution (celles que n'aurait pas ajoutées `voice`), sont imprimées de façon particulière. On retrouve donc toutes les altérations qu'imprimerait `default`, mais certaines sont considérées comme étant « de précaution ».



`piano`

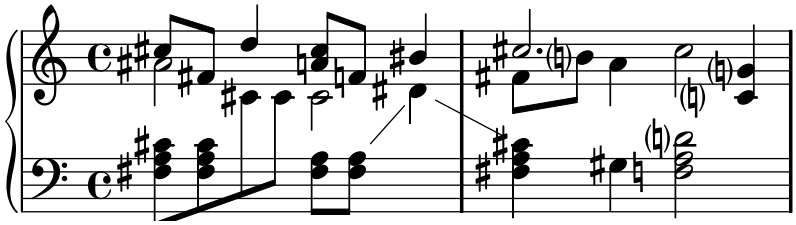
Cette règle est communément employée pour les partitions de piano au XXe siècle. Très similaire à `modern` de par son comportement, elle s'en distingue en ce que les altérations tiennent compte des autre portées du contexte `GrandStaff` ou `PianoStaff`.

Cette règle s'applique par défaut dans un `GrandStaff` et dans un `PianoStaff`.



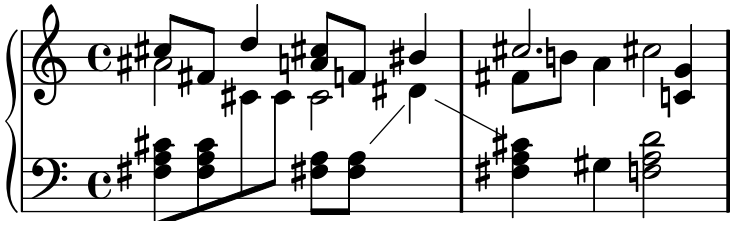
`piano-cautionary`

Identique au style `piano`, mais les altérations de précaution sont imprimées différemment.



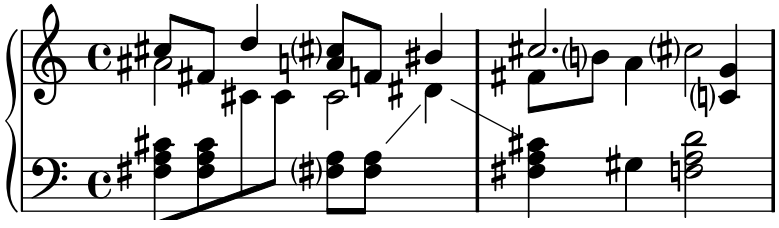
`neo-modern`

Cette règle suit les pratiques de la musique contemporaine : les altérations accidentelles apparaîssent comme dans le style `modern`, à ceci près qu'elles sont répétées dans la même mesure – sauf si elles concernent deux notes consécutives.



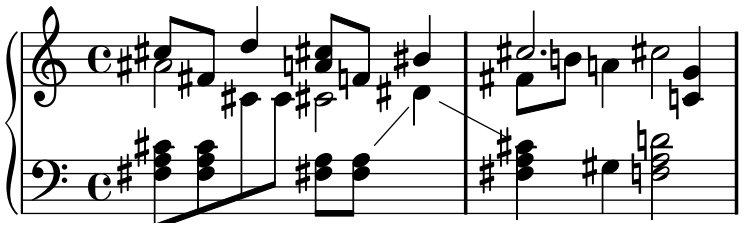
`neo-modern-cautionary`

Identique au style `neo-modern`, mais les altérations de précaution sont imprimées différemment.



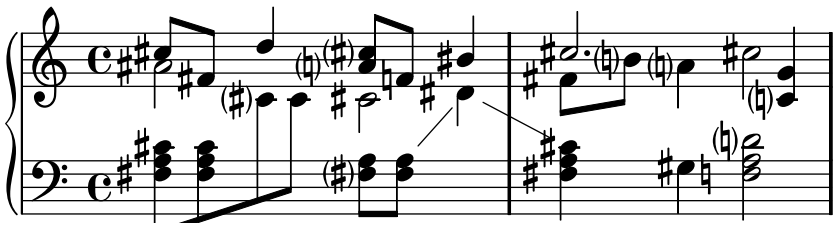
`neo-modern-voice`

Cette règle sert aux altérations dans de la musique polyphonique destinée autant à des musiciens différents qu'à quelqu'un qui lirait l'ensemble des voix. Les altérations sont imprimées voix par voix comme avec le style `neo-modern` mais les autres voix dans le même contexte `Staff` en tiennent aussi compte.



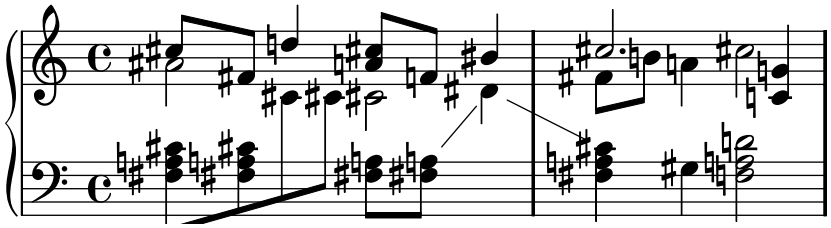
`neo-modern-voice-cautionary`

Cette règle est indentique à `neo-modern-voice`, mais les altérations de précaution sont imprimées soit entre parenthèses (par défaut), soit en plus petit.



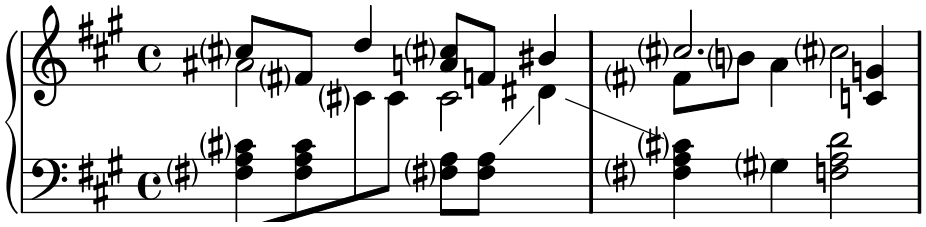
`dodecaphonic`

Cette règle reproduit ce que certains compositeurs du début du XXe siècle ont introduit dans leur désir d'abolir la distinction entre les notes naturelles ou non. Ainsi, **chaque** note est affublée d'une altération, même si elle est naturelle.



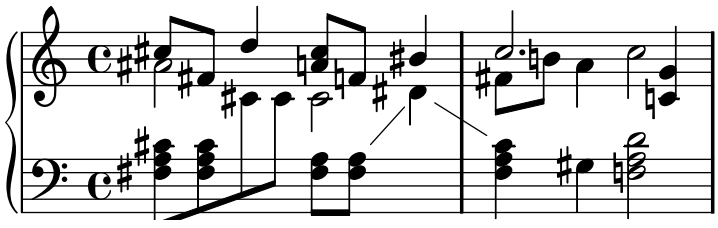
`teaching`

Cette règle est à usage pédagogique : l'impression d'une simple gamme fera apparaître une altération de précaution pour toute note altérée. Les altérations accidentelles sont imprimées selon le style `modern`, et une altération de précaution est ajoutée pour chaque dièse ou bémol à la clef – sauf dans le cas de notes consécutives.



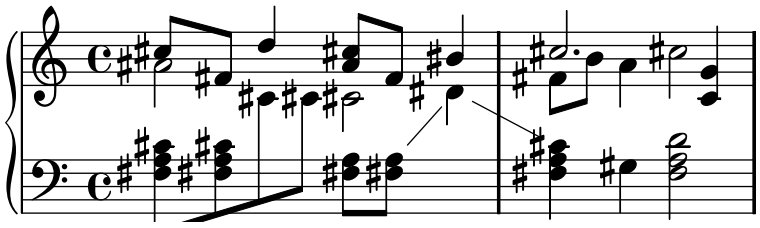
`no-reset`

C'est la même règle que `default`, mais l'effet des altérations accidentelles ne cesse jamais, même dans les mesures suivantes.



`forget`

Tout le contraire de `no-reset` : l'effet des altérations cesse aussitôt ; toutes les altérations, quelque soit leur place dans la mesure, sont de ce fait imprimées en fonction de l'éventuelle armure. À l'inverse du style `dodecaphonic`, aucun bécarre ne sera ici imprimé.



Voir aussi

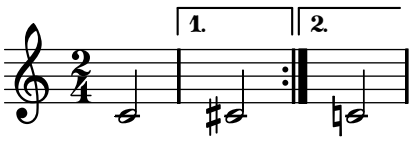
Morceaux choisis : Section "Hauteurs" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Accidental" dans *Référence des propriétés internes*, Section "Accidental_engraver" dans *Référence des propriétés internes*, Section "GrandStaff" dans *Référence des propriétés internes* et Section "PianoStaff" dans *Référence des propriétés internes*, Section "Staff" dans *Référence des propriétés internes*, Section "AccidentalSuggestion" dans *Référence des propriétés internes*, Section "AccidentalPlacement" dans *Référence des propriétés internes*. Section "accidental-suggestion-interface" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les notes simultanées sont considérées comme des évènements séquentiels. Ceci implique que, dans un accord, les altérations accidentelles seront imprimées comme si les notes de l'accord apparaissaient l'une après l'autre, en fonction de l'ordre dans lequel elles ont été saisies – ce qui peut poser problème lorsqu'au sein d'un accord certaines altérations dépendent les unes des autres. Ce problème est à résoudre manuellement, en insérant des `!` et des `?` après les notes concernées.
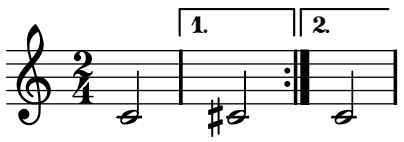
L'absence d'altération de précaution est déterminée par l'examen de la mesure précédente. Néanmoins, lorsqu'un bloc `\alternative` suit une section `\repeat volta N`, la logique voudrait que l'on regarde la dernière mesure *jouée* plutôt que la dernière *imprimée*. Dans l'exemple qui suit, vous conviendrez que le do de la seconde alternative ne nécessite pas son bécarre.



L'astuce suivante, qui définit temporairement le recours au style `forget`, permet d'obtenir quelque chose de présentable.

```
forget = #(define-music-function (parser location music) (ly:music?) #{
  #(set-accidental-style 'forget)
  $music
  #(set-accidental-style 'modern)
#})
{
  #(set-accidental-style 'modern)
  \time 2/4
```

```
  \repeat volta 2 {
    c'2
  }
  \alternative {
     cis'
     \forget c'
  }
}
```
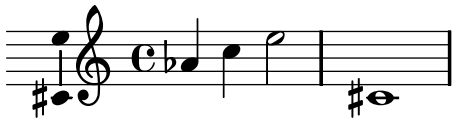


Ambitus

L'*ambitus* est l'amplitude des hauteurs d'une voix donnée dans une partition. Ce terme peut aussi désigner la tessiture qu'un instrument est capable d'atteindre. Souvent, cet *ambitus* est imprimé au début des partitions vocales, afin que les exécutants puissent voir au premier coup d'œil s'ils sont en mesure de tenir la partie en question.

Pour exprimer l'*ambitus* d'une pièce, on indique avant la clef deux notes représentant la hauteur la plus basse et la plus haute. Les éventuelles altérations accidentelles seront automatiquement ajoutées.

```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}

\relative c'' {
  aes c e2
  cis,1
}
```
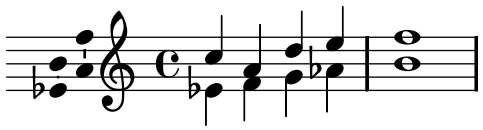


Morceaux choisis

Un ambitus par voix

L'`ambitus` peut être individualisé par voix. Il faut en pareil cas éviter qu'ils se chevauchent.

```
\new Staff <<
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c'' {
    \override Ambitus #'X-offset = #2.0
    \voiceOne
    c4 a d e
    f1
  }
```
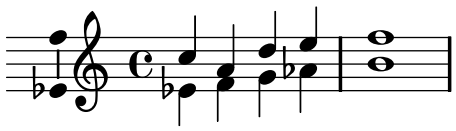
```
  \new Voice \with {
    \consists "Ambitus_engraver"
  } \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Ambitus sur plusieurs voix

Si plusieurs voix se trouvent sur une même portée, on peut attribuer le graveur `Ambitus_engraver` au contexte `Staff` afin d'obtenir l'ambitus de toutes les voix cumulées, non d'une seule des voix actives.

```
\new Staff \with {
  \consists "Ambitus_engraver"
  }
<<
  \new Voice \relative c'' {
    \voiceOne
    c4 a d e
    f1
  }
  \new Voice \relative c' {
    \voiceTwo
    es4 f g as
    b1
  }
>>
```



Réglage de l'affichage d'un ambitus

L'affichage d'un *ambitus* peut s'affiner pour répondre à vos préférences en matière d'esthétique.

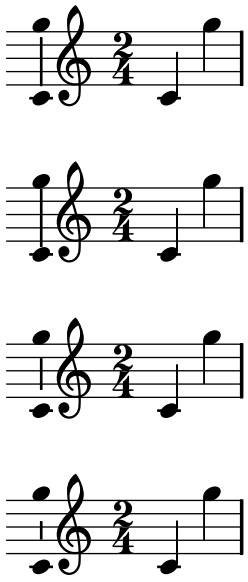
```
\layout {
  \context {
    \Voice
    \consists "Ambitus_engraver"
  }
}
```

```
\new Staff {
  \time 2/4
  % Default setting
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine #'gap = #0
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine #'gap = #1
  c'4 g''
}

\new Staff {
  \time 2/4
  \override AmbitusLine #'gap = #1.5
  c'4 g''
}
```



Voir aussi

Glossaire musicologique : Section “ambitus” dans *Glossaire*.

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Ambitus_engraver” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*, Section “Staff” dans *Référence des propriétés internes*, Section “Ambitus” dans *Référence des propriétés internes*, Section “AmbitusAccidental” dans *Référence des propriétés internes*, Section “AmbitusLine” dans *Référence des propriétés internes*, Section “AmbitusNoteHead” dans *Référence des propriétés internes*, Section “ambitus-interface” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

LilyPond ne gère pas les collisions entre plusieurs *ambitus* présents sur une même portée.
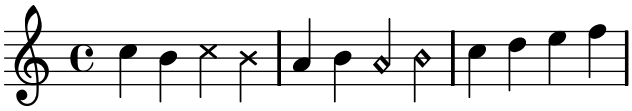
1.1.4 Têtes de note

Nous allons voir dans ce chapitre comment modifier l'aspect des têtes de note.

Têtes de note spécifiques

L'apparence des têtes de note peut évoluer au cours de la partition :

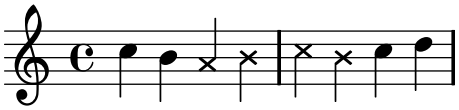
```
c4 b
\override NoteHead #'style = #'cross
c4 b
\revert NoteHead #'style
a b
\override NoteHead #'style = #'harmonic
a b
\revert NoteHead #'style
c4 d e f
```



Pour une liste exhaustive des styles de tête de note, consultez Section A.7 [Styles de tête de note], page 510.

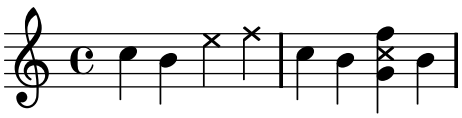
Certains instruments utilisent des têtes de note différentes à des fins spécifiques – des croix (style `cross`) pour le *parlato* des chanteurs ou les notes étouffées des guitares :

```
c4 b
\xNotesOn
 a b c4 b
\xNotesOff
c4 d
```



Cette commande opère aussi bien sur des notes isolées qu'au sein d'un accord, dans une portée traditionnelle ou dans un contexte de tablature :
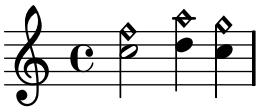
```
c4 b
\xNote { e f }
c b < g \xNote c f > b
```



Vous pouvez utiliser, en lieu et place de `\xNote`, `\xNotesOn` et `\xNotesOff`, les commandes `\deadNote`, `\deadNotesOn` et `\deadNotesOff`.

Il existe un raccourci pour les notes en losange qui indiquent les harmoniques des cordes :

```
<c f\harmonic>2 <d a'\harmonic>4 <c g'\harmonic>
```



Commandes prédéfinies

`\harmonic`, `\xNotesOn`, `\xNotesOff`, `\xNote`.

Voir aussi

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

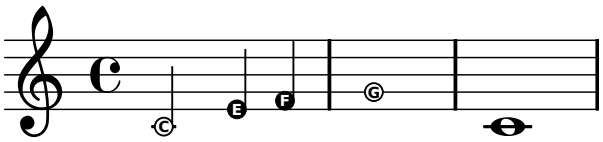
Manuel de notation : Section A.7 [Styles de tête de note], page 510, [Notes en accords], page 137, [Indication des harmoniques et notes étouffées], page 325.

Référence des propriétés internes : Section “note-event” dans *Référence des propriétés internes*, Section “Note_heads_engraver” dans *Référence des propriétés internes*, Section “Ledger_line_engraver” dans *Référence des propriétés internes*, Section “NoteHead” dans *Référence des propriétés internes*. Section “LedgerLineSpanner” dans *Référence des propriétés internes*, Section “note-head-interface” dans *Référence des propriétés internes*, Section “ledger-line-spanner-interface” dans *Référence des propriétés internes*.

Têtes de note avec nom de note

Les notes « easy play » comportent le nom de la note à l’intérieur de la tête. On l’utilise dans des partitions pour débutants. L’impression doit être de plus grande taille, afin que les lettres soient lisibles. Voir à ce propos Section 4.2.2 [Définition de la taille de portée], page 423.

```
#(set-global-staff-size 26)
\relative c' {
  \easyHeadsOn
  c2 e4 f
  g1
  \easyHeadsOff
  c,1
}
```



Commandes prédéfinies

`\easyHeadsOn`, `\easyHeadsOff`.

Morceaux choisis

Numbers as easy note heads

Easy notation note heads use the `note-names` property of the `NoteHead` object to determine what appears inside the note head. By overriding this property, it is possible to print numbers representing the scale-degree.

A simple engraver can be created to do this for every note head object it sees.

```
#(define Ez_numbers_engraver
   (list
    (cons 'acknowledgers
          (list
           (cons 'note-head-interface
                 (lambda (engraver grob source-engraver)
                   (let* ((context (ly:translator-context engraver))
                          (tonic-pitch (ly:context-property context 'tonic))
                          (tonic-name (ly:pitch-notename tonic-pitch))
                          (grob-pitch
                           (ly:event-property (event-cause grob) 'pitch))
                          (grob-name (ly:pitch-notename grob-pitch))
                          (delta (modulo (- grob-name tonic-name) 7))
                          (note-names
                           (make-vector 7 (number->string (1+ delta)))))
                     (ly:grob-set-property! grob 'note-names note-names))))))))

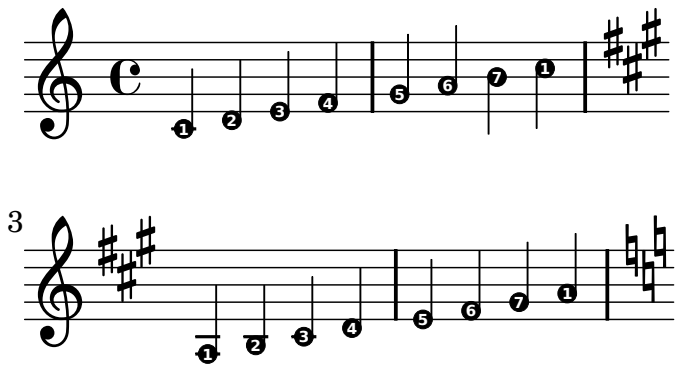
#(set-global-staff-size 26)

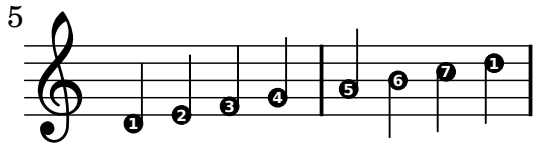
\layout {
  ragged-right = ##t
  \context {
    \Voice
    \consists \Ez_numbers_engraver
  }
}

\relative c' {
  \easyHeadsOn
  c4 d e f
  g4 a b c \break

  \key a \major
  a,4 b cis d
  e4 fis gis a \break

  \key d \dorian
  d,4 e f g
  a4 b c d
}
```





Voir aussi

Manuel de notation : Section 4.2.2 [Définition de la taille de portée], page 423.
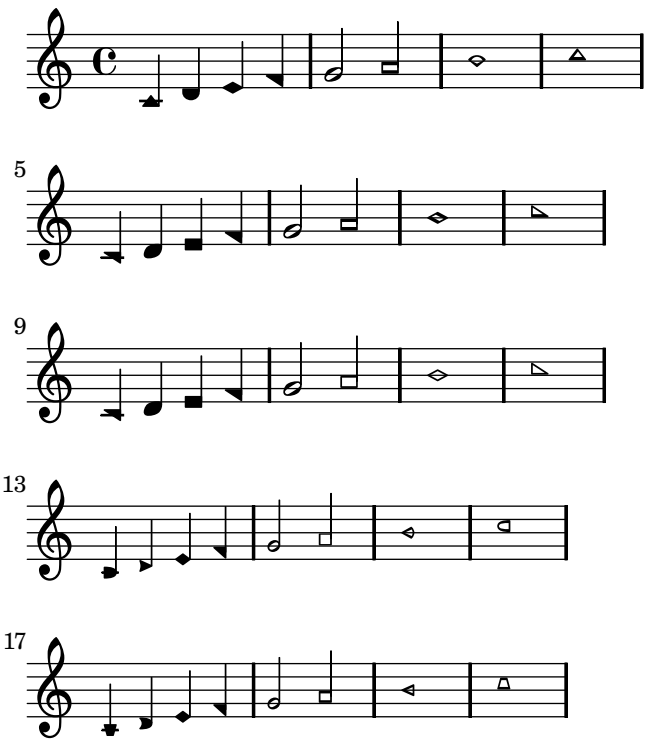
Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “note-event” dans *Référence des propriétés internes*, Section “Note_heads_engraver” dans *Référence des propriétés internes*, Section “NoteHead” dans *Référence des propriétés internes*, Section “note-head-interface” dans *Référence des propriétés internes*.

Têtes de note à forme variable
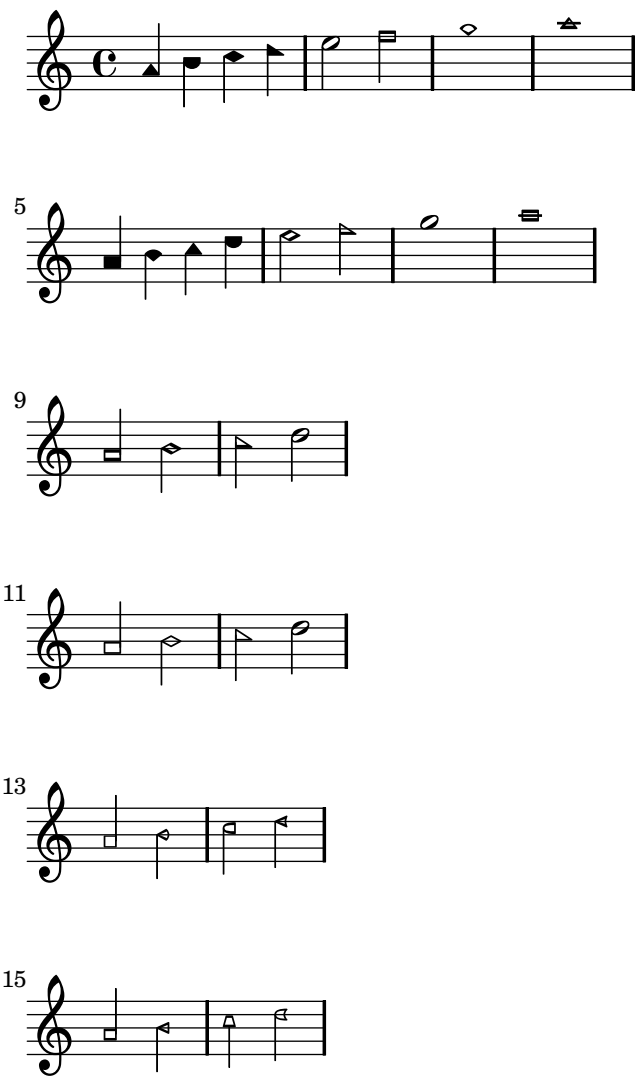
En notation profilée, le profil d’une tête de note correspond à la fonction harmonique de cette note dans la gamme. Ce style de notation était très en vogue dans les recueils de chansons américains du XIXe siècle. Voici comment procéder :

```
\aikenHeads
c, d e f g2 a b1 c \break
\sacredHarpHeads
c,4 d e f g2 a b1 c \break
\southernHarmonyHeads
c,4 d e f g2 a b1 c \break
\funkHeads
c,4 d e f g2 a b1 c \break
\walkerHeads
c,4 d e f g2 a b1 c \break
```



Les profils sont déterminés par la hauteur dans la gamme, le premier degré étant défini par la commande `\key`. Pour une tonalité mineure, les degrés sont déterminés par rapport au relatif majeur :

```
\key a \minor
\aikenHeads
a b c d e2 f g1 a \break
\aikenHeadsMinor
a,4 b c d e2 f g1 a \break
\sacredHarpHeadsMinor
a,2 b c d \break
\southernHarmonyHeadsMinor
a2 b c d \break
\funkHeadsMinor
a2 b c d \break
\walkerHeadsMinor
a2 b c d \break
```



Commandes prédéfinies

\aikenHeads, \aikenHeadsMinor, \funkHeads, \funkHeadsMinor, \sacredHarpHeads. \sacredHarpHeadsMinor, \southernHarmonyHeads, \southernHarmonyHeadsMinor, \walkerHeads, \walkerHeadsMinor.

Morceaux choisis

Profilage des notes selon leur degré dans la gamme

La propriété `shapeNoteStyles` permet d'affecter un profil particulier à chaque degré de la gamme – à partir de l'armure ou de la propriété `tonic`. Ses valeurs sont constituées d'une liste de symboles, qu'il s'agisse de formes géométriques (`triangle`, `cross`, ou `xcircle`) ou basés sur la tradition des graveurs américains (avec quelques noms de note latins).

LilyPond dispose de deux raccourcis, `\aikenHeads` et `\sacredHarpHeads`, permettant de reproduire déanciens recueils de chansons américaines.

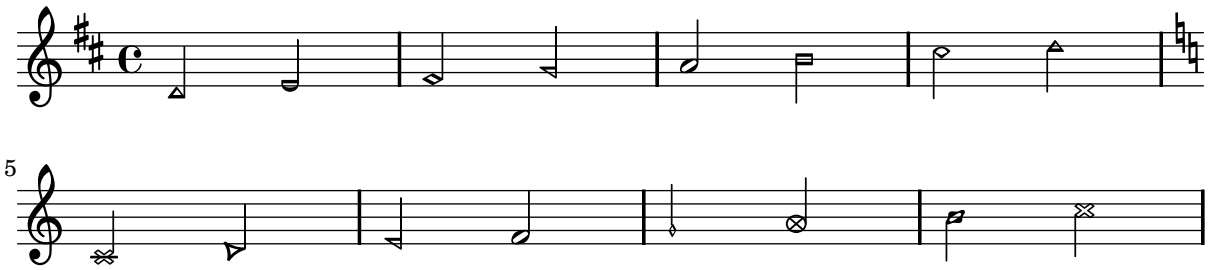
L'exemple suivant montre plusieurs manières de profiler les têtes de note, ainsi que la capacité de trnsposer tout en respectant la fonction harmonique de chaque note dans la gamme.

```
fragment = {
  \key c \major
  c2 d
  e2 f
  g2 a
  b2 c
}

\new Staff {
  \transpose c d
  \relative c' {
    \set shapeNoteStyles = #'#(do re mi fa
                               #f la ti)
    \fragment
  }

  \break

  \relative c' {
    \set shapeNoteStyles = #'#(cross triangle fa #f
                               mensural xcircle diamond)
    \fragment
  }
}
```



Pour une liste exhaustive des styles de tête de note, consultez Section A.7 [Styles de tête de note], page 510.

Voir aussi

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

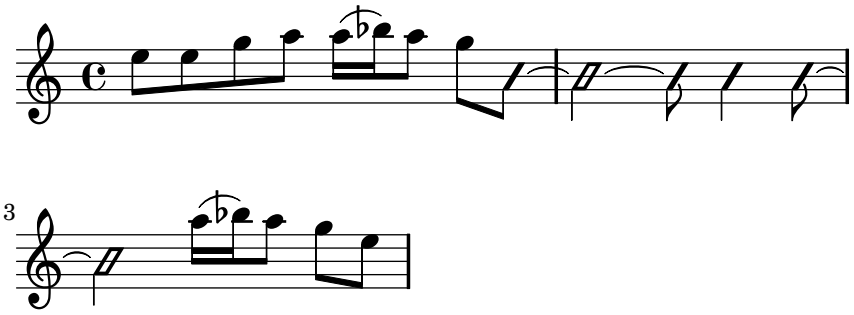
Manuel de notation : Section A.7 [Styles de tête de note], page 510.

Référence des propriétés internes : Section “note-event” dans *Référence des propriétés internes*, Section “Note_heads_engraver” dans *Référence des propriétés internes*, Section “NoteHead” dans *Référence des propriétés internes*, Section “note-head-interface” dans *Référence des propriétés internes*.

Improvisation

L’improvisation peut quelquefois s’indiquer à l’aide de notes de forme allongée (*slash*). L’interprète jouera alors les notes qu’il veut, en respectant toutefois le rythme affiché. Ces têtes de notes sont créées ainsi :

```
\new Voice \with {
  \consists "Pitch_squash_engraver"
} {
  e8 e g a a16( bes) a8 g
  \improvisationOn
  e8 ~
  e2 ~ e8 f4 f8 ~
  f2
  \improvisationOff
  a16( bes) a8 g e
}
```



Commandes prédéfinies

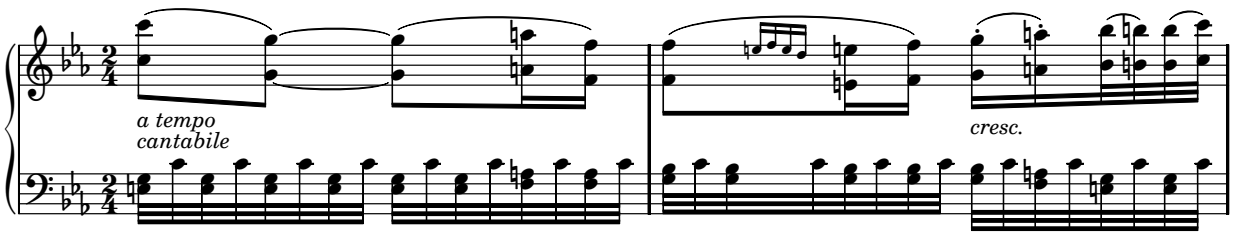
`\improvisationOn`, `\improvisationOff`.

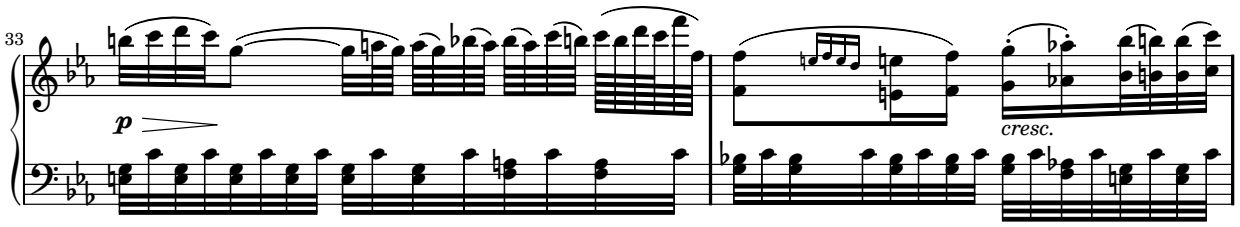
Voir aussi

Morceaux choisis : Section “Hauteurs” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Pitch_squash_engraver” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*, Section “RhythmicStaff” dans *Référence des propriétés internes*.

1.2 Rythme
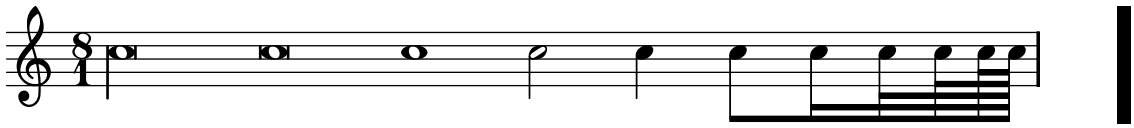




Cette section traite du rythme : durées, silences, barres de ligature et de mesure.

1.2.1 Écriture du rythme

Durées
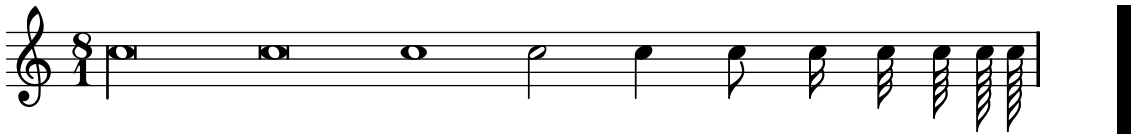
Dans les modes de notes, d'accords et de paroles, les durées sont écrites avec des chiffres et des points : les durées sont indiquées par leur valeur fractionnaire par rapport à la durée d'une ronde. Une noire, par exemple, qui équivaut à un 1/4 de ronde – *quarter note* en anglais – s'écrit `4`, alors qu'une blanche – *half-note*, 1/2 ronde – s'écrit `2`. Pour des notes plus longues qu'une ronde, vous devrez utiliser les commandes `\longa` pour une longue, et `\breve` pour une brève, aussi appelée carrée. Des durées plus courtes que la quintuple croche – 1/128 de ronde – sont possibles, à condition de les ligaturer.

```
\time 8/1
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c128 c128
```



Voici ces mêmes durées sans la fonction de ligature automatique.

```
\time 8/1
\autoBeamOff
c\longa c\breve c1 c2
c4 c8 c16 c32 c64 c128 c128
```



Une note dont la durée est de quatre brèves s'obtient par la commande `\maxima`. Celle-ci n'est toutefois disponible que dans le cadre de la notation ancienne. Pour plus de détails, voir Section 2.9 [Notations anciennes], page 354.
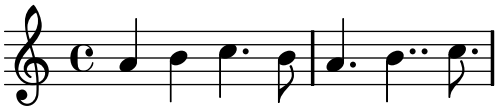
Si la durée d'une note n'est pas précisée, elle est alors assimilée à la durée de la note précédente. La valeur par défaut pour la première note est la noire (`4`).

```
a a a2 a a4 a a1 a
```



Pour obtenir des notes pointées, ajoutez simplement un point (`.`) au chiffre. Les notes doublement pointées sont créées de la même façon.

```
a4 b c4. b8 a4. b4.. c8.
```



Certaines durées ne peuvent s'obtenir à partir de chiffres et de points, mais uniquement en « liant » deux ou plusieurs notes entre elles. Voir [Liaisons de prolongation], page 44 à ce sujet.

Quant à la manière de spécifier la durée des syllabes ou bien d'aligner des paroles par rapport aux notes, reportez vous au chapitre Section 2.1 [Musique vocale], page 219.

Espacer les notes selon leur durée relative est tout à fait possible. Pour plus de détails à ce sujet et sur les autres réglages propres à cette forme de notation, reportez vous à Section 4.5.5 [Notation proportionnelle], page 452.

Les points sont normalement haussés pour éviter les lignes de portée, sauf dans certaines polyphonies. Des commandes prédéfinies permettent de manuellement forcer une orientation particulière, comme indiqué au chapitre Section 5.4.2 [Direction et positionnement], page 475.

Commandes prédéfinies
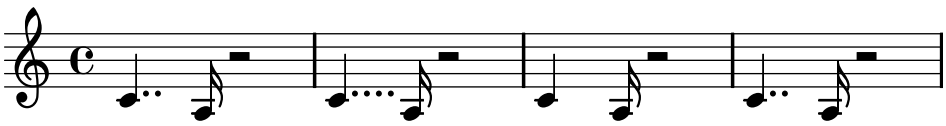
`\autoBeamOn`, `\autoBeamOff`, `\dotsUp`, `\dotsDown`, `\dotsNeutral`.

Morceaux choisis

Spécification du nombre de points d'augmentation d'une note

Voici comment modifier le nombre de points d'augmentation affectés à une note en particulier.

```
\relative c' {
  c4.. a16 r2 |
  \override Dots #'dot-count = #4
  c4.. a16 r2 |
  \override Dots #'dot-count = #0
  c4.. a16 r2 |
  \revert Dots #'dot-count
  c4.. a16 r2 |
}
```



Voir aussi

Glossaire musicologique : Section "breve" dans *Glossaire*, Section "longa" dans *Glossaire*, Section "maxima" dans *Glossaire*, Section "valeur des notes" dans *Glossaire*, Section "Noms de durée (notes et silences)" dans *Glossaire*.

Manuel de notation : [Barres de ligature automatiques], page 70, [Liaisons de prolongation], page 44, [Hampes], page 188, Section 1.2.1 [Écriture du rythme], page 37, Section 1.2.2 [Écriture des silences], page 47, Section 2.1 [Musique vocale], page 219, Section 2.9 [Notations anciennes], page 354, Section 4.5.5 [Notation proportionnelle], page 452.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Dots" dans *Référence des propriétés internes*, Section "DotColumn" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Il n'existe pas à proprement parler de limite (inférieure ou supérieure) en terme de durée d'un silence. Cependant, le nombre de glyphes disponibles ne couvre que les silences allant du trente-deuxième de soupir à la maxime (valant huit pauses).

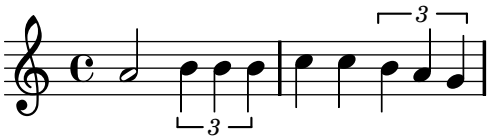
Nolets

Les nolets – triolets, quintolets, etc. – sont obtenus en multipliant toutes les durées d'une expression musicale par une fraction.

```
\times fraction { expression_musicale }
```

La durée de l'*expression_musicale* sera multipliée par la fraction. Le dénominateur de cette fraction sera imprimé au-dessus ou au-dessous des notes, parfois avec un crochet. Le nolet le plus courant est le triolet, dans lequel 3 notes ont la durée de 2, et où les notes durent donc 2/3 de leur valeur écrite.
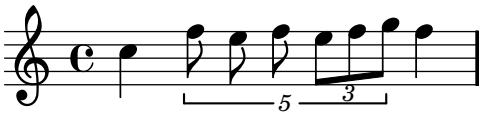
```
a2 \times 2/3 { b4 b b }
c4 c \times 2/3 { b4 a g }
```



Des commandes prédéfinies permettent de déroger au positionnement automatique du crochet en surplomb ou au-dessous des notes – voir le chapitre Section 5.4.2 [Direction et positionnement], page 475.

Les nolets peuvent être imbriqués ; par exemple,

```
\autoBeamOff
c4 \times 4/5 { f8 e f \times 2/3 { e[ f g] } } f4 |
```



Lorsque, dans une imbrication, les nolets débutent au même instant, il vous faut recourir à la commande `\tweak`.

Vous pouvez interférer sur la durée des notes sans imprimer de crochet, comme indiqué au chapitre [Changement d'échelle des durées], page 43.

Commandes prédéfinies

`\tupletUp`, `\tupletDown`, `\tupletNeutral`.

Morceaux choisis
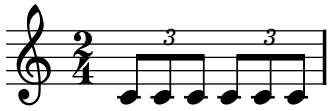
Plusieurs triolets avec une seule commande \times

La propriété `tupletSpannerDuration` spécifie la longueur voulue de chaque crochet. Avec elle, vous pouvez faire plusieurs nolets en ne tapant `\times` qu'une fois, ce qui évite une longue saisie.

Dans l'exemple suivant, deux triolets sont imprimés avec une seule fonction `\times`.

Pour plus d'information sur `make-moment`, voir la section appropriée du manuel de notation.

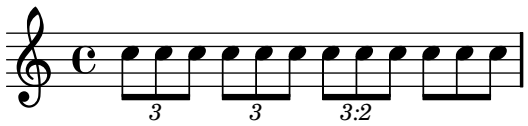
```
\relative c' {
  \time 2/4
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
  \times 2/3 { c8 c c c c c }
}
```



Modifier l'apparence du chiffre de nolet

L'apparence du chiffre est déterminée par la propriété `text` dans `TupletNumber`. La valeur par défaut imprime seulement le dénominateur, mais si elle est définie par la fonction `tuplet-number::calc-fraction-text`, la fraction entière *num*:*den* sera imprimée à la place.

```
\relative c'' {
  \times 2/3 { c8 c c }
  \times 2/3 { c8 c c }
  \override TupletNumber #'text = #tuplet-number::calc-fraction-text
  \times 2/3 { c8 c c }
  \override TupletNumber #'stencil = ##f
  \times 2/3 { c8 c c }
}
```



Nolets au chiffrage inhabituel

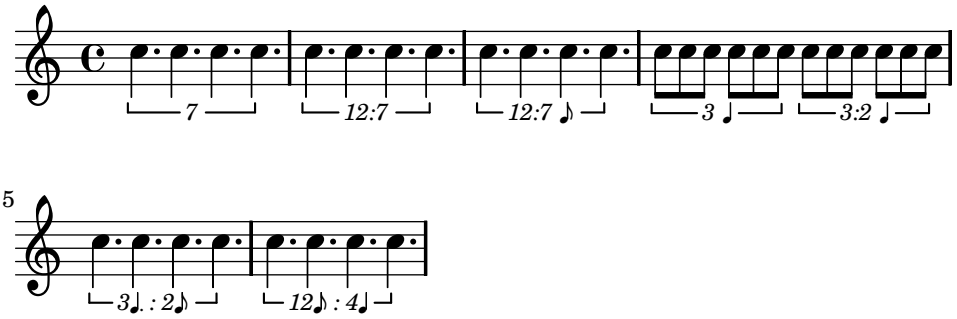
LilyPond sait aussi gérer des nolets dont le chiffrage imprimé ne correspond pas exactement à la fraction de mesure à laquelle ils se réfèrent, tout comme ceux auxquels une valeur de note vient en complément au chiffre.

```
\relative c'' {
  \once \override TupletNumber #'text =
    #(tuplet-number::non-default-tuplet-denominator-text 7)
  \times 2/3  { c4. c4. c4. c4. }
  \once \override TupletNumber #'text =
    #(tuplet-number::non-default-tuplet-fraction-text 12 7)
  \times 2/3  { c4. c4. c4. c4. }
  \once \override TupletNumber #'text =
    #(tuplet-number::append-note-wrapper
      (tuplet-number::non-default-tuplet-fraction-text 12 7) "8")
  \times 2/3  { c4. c4. c4. c4. }

  \once \override TupletNumber #'text =
    #(tuplet-number::append-note-wrapper
      tuplet-number::calc-denominator-text "4")
  \times 2/3  { c8 c8 c8 c8 c8 c8 }
  \once \override TupletNumber #'text =
    #(tuplet-number::append-note-wrapper
```

```
      tuplet-number::calc-fraction-text "4")
  \times 2/3  { c8 c8 c8 c8 c8 c8 }

  \once \override TupletNumber #'text =
    #(tuplet-number::fraction-with-notes "4." "8")
  \times 2/3  { c4. c4. c4. c4. }
  \once \override TupletNumber #'text =
    #(tuplet-number::non-default-fraction-with-notes 12 "8" 4 "4")
  \times 2/3  { c4. c4. c4. c4. }
}
```

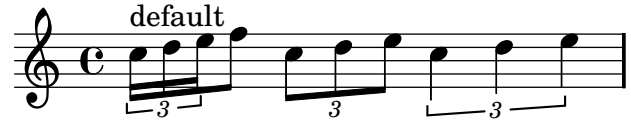


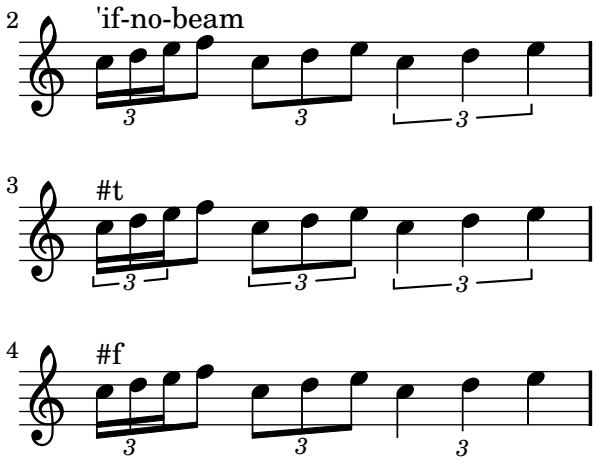
Contrôle de l'impression des crochets de nolet

Selon la tradition, les crochets indicateurs de nolet sont toujours imprimés sauf dans le cas où ils seraient de la même longuer qu'une ligature. LilyPond permet, au travers de la propriété `'bracket-visibility`, de contôler précisément leur affichage : déterminée à `#t`, ils seront toujours imprimés ; `#f` permet de ne jamais les imprimer, et `#'if-no-beam` les imprimera en l'absence de ligature.

```
music = \relative c'' {
  \times 2/3 { c16[ d e } f8]
  \times 2/3 { c8 d e }
  \times 2/3 { c4 d e }
}

\new Voice {
  \relative c' {
    << \music s4^"default" >>
    \override TupletBracket #'bracket-visibility = #'if-no-beam
    << \music s4^"'if-no-beam" >>
    \override TupletBracket #'bracket-visibility = ##t
    << \music s4^"#t" >>
    \override TupletBracket #'bracket-visibility = ##f
    << \music s4^"#f" >>
  }
}
```

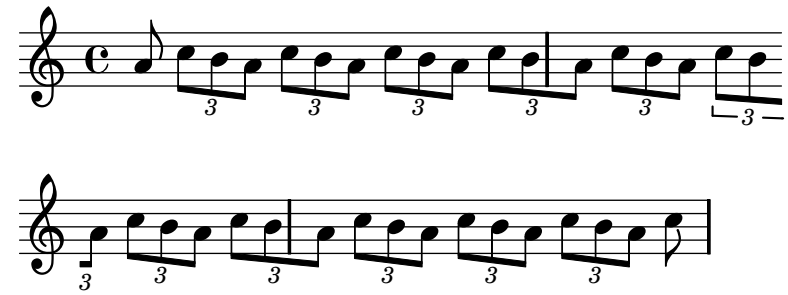




Saut de ligne au milieu d'un nolet avec ligature

Cet exemple peu académique démontre comment il est possible d'insérer un saut de ligne dans un nolet portant une ligature. Ces ligatures doivent toutefois être explicites.

```
\layout {
  \context {
    \Voice
    % Permit line breaks within tuplets
    \remove "Forbid_line_break_engraver"
    % Allow beams to be broken at line breaks
    \override Beam #'breakable = ##t
  }
}
\relative c'' {
  a8
  \repeat unfold 5 { \times 2/3 { c[ b a] } }
  % Insert a manual line break within a tuplet
  \times 2/3 { c[ b \bar "" \break a] }
  \repeat unfold 5 { \times 2/3 { c[ b a] } }
  c8
}
```



Voir aussi

Glossaire musicologique : Section "triolet" dans *Glossaire*, Section "nolet" dans *Glossaire*, Section "polymétrie" dans *Glossaire*.

Manuel d'initiation : Section "Méthodes de retouche" dans *Manuel d'initiation*.

Manuel de notation : [Gestion du temps], page 100, [Changement d'échelle des durées], page 43, Section 5.3.4 [La commande d'affinage (tweak)], page 474, [Notation polymétrique], page 64.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Références des propriétés internes : Section “TupletBracket” dans *Référence des propriétés internes*, Section “TupletNumber” dans *Référence des propriétés internes*, Section “TimeScaledMusic” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Lorsqu'un nolet en début de portée est précédé d'une note d'ornement, celle-ci doit se placer avant la commande `\times` pour éviter toute erreur. Ailleurs dans la partition, les notes d'ornement peuvent se placer au sein de l'expression constituant le nolet.

Lorsqu'un nolet en début de mouvement est combiné avec une indication de tempo – grâce à la commande `\tempo` –, l'expression contenant le nolet doit impérativement faire partie d'un bloc `\new Voice` comme indiqué au chapitre Section “Les voix contiennent la musique” dans *Manuel d'initiation*.

Changement d'échelle des durées

Vous pouvez altérer la durée des notes, silences ou accords en leur joignant une fraction *N/D*, donnant « **N/D* » – ou « **N* » si *D=1*. Ceci ne modifiera en rien l'apparence des notes ou silences produits, mais affectera le positionnement de l'objet dans la mesure, ainsi que le rendu MIDI. Cette fraction peut elle-même être multipliée du style `*M*N/D`.

Dans l'exemple suivant, les trois premières notes prennent exactement deux temps, mais aucun triolet n'est imprimé.

```
\time 2/4
% Alter durations to triplets
a4*2/3 gis4*2/3 a4*2/3
% Normal durations
a4 a4
% Double the duration of chord
<a d>4*2
% Duration of quarter, appears like sixteenth
b16*4 c4
```
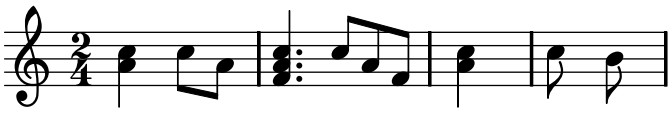


La durée d'un silence invisible ou saut de notes (*skip*) peut elle aussi être affectée d'un multiplicateur. Cette technique permet tout simplement de sauter plusieurs mesures, comme par exemple un `s1*23`.

Il est tout à fait possible d'échelonner des fragments musicaux plus ou moins longs à l'aide d'une simple fraction, comme si chaque note, accord ou silence était affecté de ce même quotient. L'apparence de cette musique ne sera en rien modifiée ; seule la durée des notes est multipliée en interne par la fraction *numérateur/dénominateur*. N'oubliez pas de préserver les espaces de part et d'autre du point. Voici un exemple illustrant la manière de comprimer ou étirer de la musique :

```
\time 2/4
% Normal durations
<c a>4 c8 a
% Scale music by *2/3
\scaleDurations #'(2 . 3) {
  <c a f>4. c8 a f
}
% Scale music by *2
```

```
\scaleDurations #'(2 . 1) {
  <c' a>4 c8 b
}
```



Cette technique est tout à fait appropriée à la notation polymétrique – voir [Notation polymétrique], page 64.

Voir aussi

Manuel de notation : [Nolets], page 39, [Silences invisibles], page 49, [Notation polymétrique], page 64.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Liaisons de prolongation

Une liaison de tenue (ou de prolongation) relie deux notes adjacentes de même hauteur. Dans les faits, elle prolonge la durée d’une note.

> **Note :** Une liaison de tenue ne doit pas être confondue avec une liaison d’**articulation** ou de **phrasé**. Une liaison de tenue est un moyen parmi d’autres pour prolonger la durée d’une note, tout comme les points.
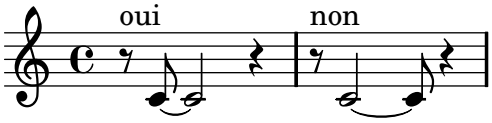
Une liaison de tenue s’indique au moyen d’un tilde `~`.

```
a2 ~ a
```



Les liaisons de tenue sont utilisées soit lorsque la note dépasse de la mesure, soit quand les points ne suffisent pas à donner la bonne durée. Lorsque l’on utilise ces liaisons, les valeurs rythmiques les plus longues doivent s’aligner sur les subdivisions de la mesure, comme ici :

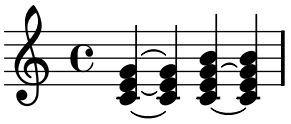
```
\relative {
  r8^"oui" c8 ~ c2 r4 |
  r8^"non" c2 ~ c8 r4
}
```



Lorsque l’on doit lier de nombreuses notes sur plusieurs mesures, il devient plus facile d’avoir recours à la division automatique des notes – voir [Découpage automatique des notes], page 67. Ce procédé divise automatiquement les notes trop longues, et les lie par-delà les barres de mesure.
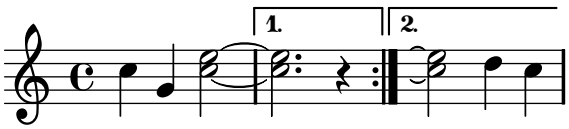
Quand une liaison de tenue se trouve entre deux accords, toutes les notes de même hauteur entre ces deux accords sont reliées. S’il n’y en a aucune, aucune liaison n’est créée. Il est également possible de lier partiellement deux accords, en mettant les liaisons à l’intérieur des accords.
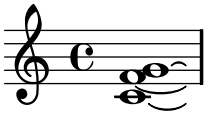
```
<c e g> ~ <c e g>
<c~ e g~ b> <c e g b>
```



Lorsqu'une mesure « de seconde fois » après une reprise commence sur une note liée, la liaison doit être répétée, comme ici :

```
\repeat volta 2 { c g <c e>2 ~ }
\alternative {
  % First alternative: following note is tied normally
  { <c e>2. r4 }
  % Second alternative: following note has a repeated tie
  { <c e>2\repeatTie d4 c } }
```



Les liaisons « Laissez vibrer » (*L.v.*) sont utilisées pour le piano, la harpe, et certains instruments de percussion. Elles indiquent à l'instrumentiste de laisser sonner la note ou l'accord au lieu de l'étouffer. Elles s'indiquent de la manière suivante :
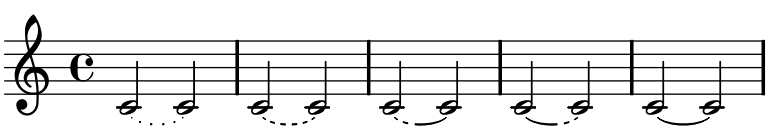
```
<c f g>1\laissezVibrer
```



Le positionnement vertical des liaisons de prolongation peut être affiné – voir à ce sujet les « commandes prédéfinies » et, pour de plus amples détails, Section 5.4.2 [Direction et positionnement], page 475.
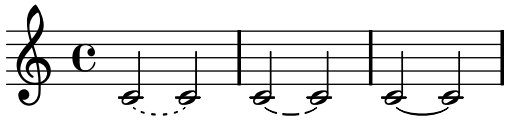
Les liaisons de prolongation peuvent s'imprimer sous la forme de ligne continue, discontinue ou pointillée.

```
\tieDotted
c2 ~ c
\tieDashed
c2 ~ c
\tieHalfDashed
c2 ~ c
\tieHalfSolid
c2 ~ c
\tieSolid
c2 ~ c
```



Il est même possible d'en personnaliser l'allure :

```
\tieDashPattern #0.3 #0.75
c2 ~ c
\tieDashPattern #0.7 #1.5
c2 ~ c
\tieSolid
c2 ~ c
```



Que ce soit pour une tenue ou un phrasé, le motif d'une ligne discontinue formant une liaison se définit de la même manière. Pour de plus amples détails à ce sujet, reportez vous au chapitre [Liaisons d'articulation], page 111.

Commandes prédéfinies

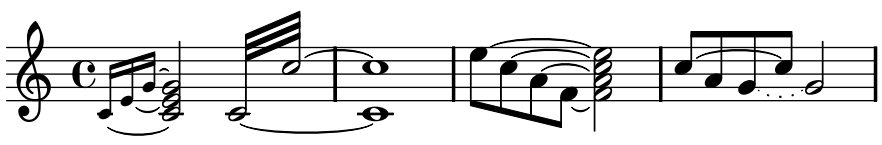
`\tieUp`, `\tieDown`, `\tieNeutral`, `\tieDotted`, `\tieDashed`, `\tieDashPattern`, `\tieHalfDashed`, `\tieHalfSolid`, `\tieSolid`.

Morceaux choisis

Liaison de tenue et arpège

Les liaisons de tenue servent parfois à rendre un accord arpégé. Dans ce cas, les notes liées ne sont pas toutes consécutives. Il faut alors assigner à la propriété `tieWaitForNote` la valeur `#t` (*true* pour « vrai »). Cette même méthode peut servir, par exemple, à lier un trémolo à un accord.

```
\relative c' {
  \set tieWaitForNote = ##t
  \grace { c16[ ~ e ~ g] ~ } <c, e g>2
  \repeat tremolo 8 { c32 ~ c' ~ } <c c,>1
  e8 ~ c ~ a ~ f ~ <e' c a f>2
  \tieUp
  c8 ~ a
  \tieDown
  \tieDotted
  g8 ~ c g2
}
```



Dessin à main levée de liaisons de tenue

Il est possible de graver manuellement les liaisons de tenue, en modifiant la propriété `tie-configuration`. Pour chaque paire, le premier nombre indique la distance à la portée, en espaces de portée, et le second la direction (1 pour haut, −1 pour bas).

```
\relative c' {
  <c e g>2 ~ <c e g>
  \override TieColumn #'tie-configuration =
    #'((0.0 . 1) (-2.0 . 1) (-4.0 . 1))
```

```
  <c e g>2 ~ <c e g>
}
```



Voir aussi

Glossaire musicologique : Section “liaison de tenue” dans *Glossaire*, Section “laissez vibrer” dans *Glossaire*.

Manuel de notation : [Liaisons d’articulation], page 111, [Découpage automatique des notes], page 67.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “LaissezVibrerTie” dans *Référence des propriétés internes*, Section “LaissezVibrerTieColumn” dans *Référence des propriétés internes*, Section “TieColumn” dans *Référence des propriétés internes*, Section “Tie” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Un changement de portée, lorsqu’une liaison de tenue est active, ne peut produire une liaison oblique.

Un changement de clé ou d’octave pendant une liaison de tenue produit un résultat indéfini. Dans ces cas là, il est préférable d’utiliser un *legato*.

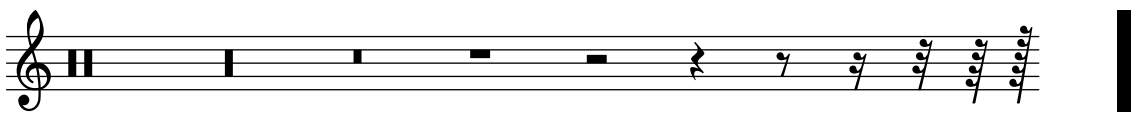
1.2.2 Écriture des silences

On saisit les silences dans une expression musicale tout comme les notes.

Silences

Les silences sont écrits comme des notes avec le nom de note `r` – premier caractère du mot *rest*. Les durées supérieures à la pause s’indiquent à l’aide de commandes prédéfinies :

```
\new Staff {
  % These two lines are just to prettify this example
  \time 16/1
  \override Staff.TimeSignature #'stencil = ##f
  % Print a maxima rest, equal to four breves
  r\maxima
  % Print a longa rest, equal to two breves
  r\longa
  % Print a breve rest
  r\breve
  r1 r2 r4 r8 r16 r32 r64 r128
}
```



Les pauses d’une mesure complète, qui sont placées au centre de la mesure, doivent être entrées comme des mesures de silence. Elles peuvent être utilisées pour une seule mesure comme

pour plusieurs, et leur utilisation est expliquée dans la section [Silences valant une mesure], page 50.

Pour spécifier explicitement la position verticale d'un silence, écrivez une note suivie de `\rest`. Un silence de même durée sera placé à la position où serait imprimée la note. Cela rend plus facile la mise en place de musique polyphonique, puisque le formateur automatique de collision des silences laissera ces silences tranquilles.

```
a4\rest d4\rest
```



Morceaux choisis

Styles de silences

Les silences peuvent être gravés selon différents styles.

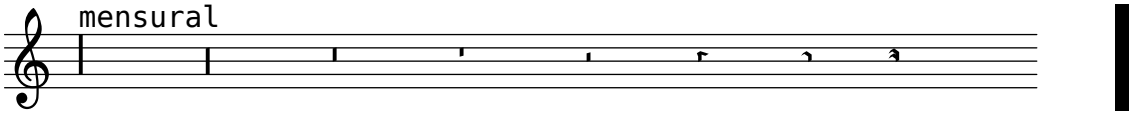
```
\layout {
  indent = 0
  \context {
    \Staff
    \remove "Time_signature_engraver"
  }
}

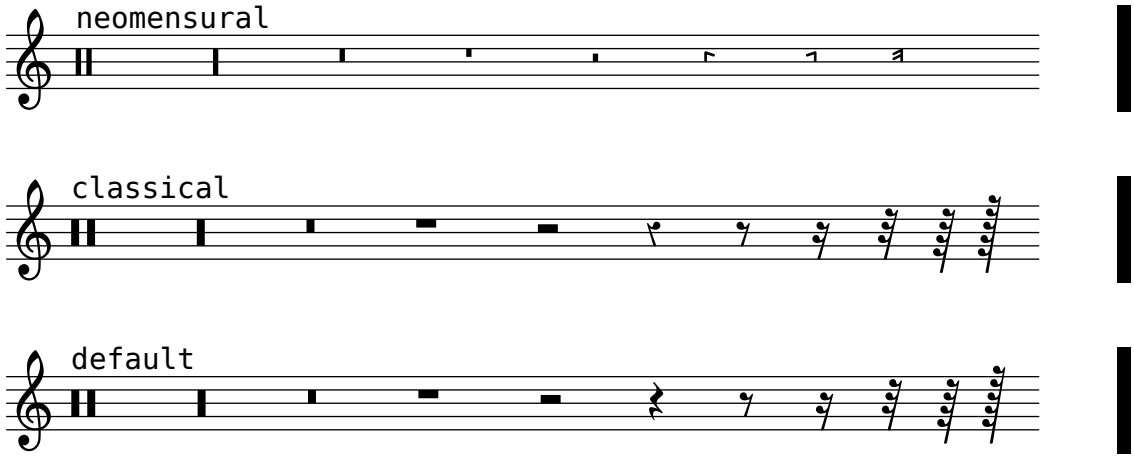
\new Staff \relative c {
  \cadenzaOn
  \override Staff.Rest #'style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest #'style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest #'style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""

  \override Staff.Rest  #'style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}
```





Voir aussi

Glossaire musicologique : Section “breve” dans *Glossaire*, Section “longa” dans *Glossaire*, Section “maxima” dans *Glossaire*.

Manuel de notation : [Silences valant une mesure], page 50.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Rest” dans *Référence des propriétés internes*.
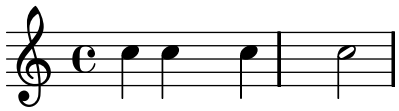
Problèmes connus et avertissements

Il n’existe pas à proprement parler de limite (inférieure ou supérieure) en terme de durée d’un silence. Cependant, le nombre de glyphes disponibles ne couvre que les silences allant du trente-deuxième de soupir à la maxime (équivalant à huit pauses).

Silences invisibles
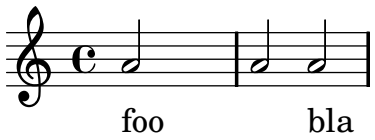
Un silence invisible – que l’on pourrait appeler un « saut » – peut être entré comme une note avec le nom de note `s` ou avec `\skip` *`durée`* :

```
c4 c s c
s2 c
```



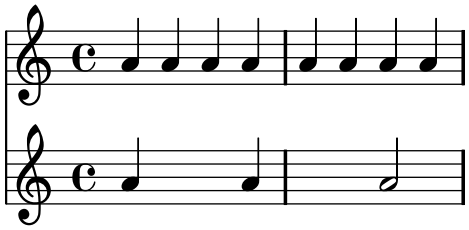
La syntaxe `s` est seulement disponible pour les modes d’entrée de notes et d’accords. Dans les autres situations, pour l’entrée de paroles par exemple, vous devrez utiliser la commande `\skip`, qui requiert une durée explicite ; cette durée ne sera pas prise en considération dès lors que les paroles suivent le rythme des notes de la mélodie à laquelle vous les aurez associées à l’aide des commandes `\addlyrics` ou `\lyricsto`.

```
<<
  {
    a2 \skip2 a2 a2
  }
  \new Lyrics {
    \lyricmode {
      foo2 \skip 1 bla2
    }
  }
>>
```



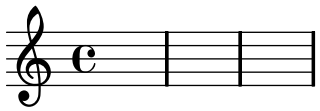
Gardez à l'esprit que `\skip` est une commande, et de ce fait n'affectera en rien la durée des notes qui suivent, contrairement à un `s`.

```
<<
  {
    \repeat unfold 8 {a4}
  }
  {
    a4 \skip 2 a |
    s2 a
  }
>>
```



La commande de saut génère simplement une case musicale vide. Le code de saut `s` crée tout de même les contextes `Staff` et `Voice` lorsque nécessaire, à l'instar des notes ou des silences :

```
s1 s s
```



Un `\skip` ne fait que sauter du temps musical ; il ne produit rien du tout, pas même un symbole transparent.

```
% This is valid input, but does nothing
\skip 1 \skip1 \skip 1
```

Voir aussi

Manuel d'initiation : Section "Visibilité et couleur des objets" dans *Manuel d'initiation*.

Manuel de notation : [Dictée à trous], page 186, Section 5.4.7 [Visibilité des objets], page 475.

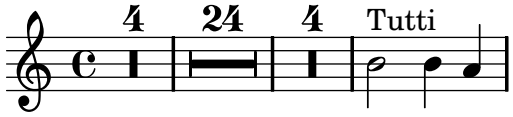
Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Référence des propriétés internes : Section "SkipMusic" dans *Référence des propriétés internes*.

Silences valant une mesure

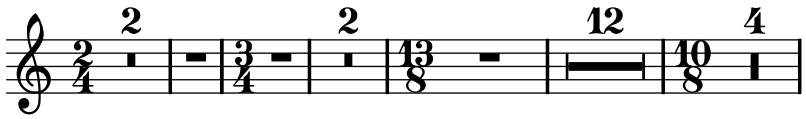
Un silence valant une ou plusieurs mesures entières s'entre avec un `R` majuscule.

```
% Rest measures contracted to single measure
\compressFullBarRests
R1*4
R1*24
R1*4
b2^"Tutti" b4 a4
```
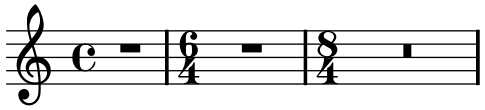


Ceci ne peut être utile que pour une mesure complètement vide. Sa durée doit donc correspondre à la longueur de la mesure telle que définie par la métrique. C'est la raison pour laquelle on utilisera aussi des points d'augmentation ou des fractions :

```
\compressFullBarRests
\time 2/4
R1 | R2 |
\time 3/4
R2. | R2.*2 |
\time 13/8
R1*13/8 | R1*13/8*12 |
\time 10/8
R4*5*4 |
```



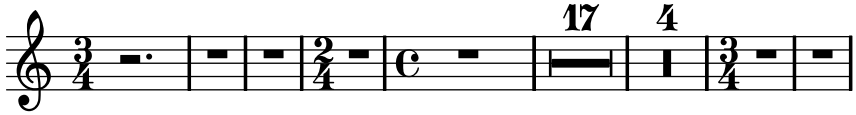
Un `R` qui s'étend sur une seule mesure s'imprime tantôt comme une pause, tantôt comme une brève – ou « bâton de pause » – et sera centré sur la mesure quelle qu'en soit la métrique :

```
\time 4/4
R1 |
\time 6/4
R1*3/2 |
\time 8/4
R1*2 |
```



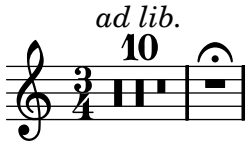
Par défaut, un silence multimesures sera répété sur autant de mesures que nécessaire. Il peut aussi n'être imprimé qu'une seule fois, surplombé du nombre de mesures vides – ou « à compter » :

```
% Default behavior
\time 3/4 r2. | R2.*2 |
\time 2/4 R2 |
\time 4/4
% Rest measures contracted to single measure
\compressFullBarRests
r1 | R1*17 | R1*4 |
% Rest measures expanded
\expandFullBarRests
\time 3/4
R2.*2 |
```
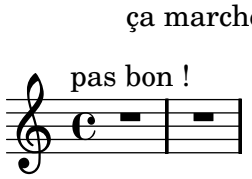


Vous pouvez aussi ajouter du texte à un silence multimesures en utilisant la syntaxe *note*-`markup` (cf. Section 1.8.2 [Mise en forme du texte], page 202). La variable `\fermataMarkup` quant à elle permet d'ajouter un point d'orgue :

```
\compressFullBarRests
\time 3/4
R2.*10^\markup { \italic "ad lib." }
R2.^\fermataMarkup
```



> **Note :** C'est `MultiMeasureRestText` qui créera le texte, et non `TextScript`. Les commandes de dérogation ou de redéfinition doivent s'adresser à l'objet concerné, comme vous pouvez le constater dans l'exemple suivant.

```
% Ceci échouera : il y a erreur quant à l'objet spécifié
\override TextScript #'padding = #5
R1^"pas bon !"
% Formulation correcte, qui fonctionnera
\override MultiMeasureRestText #'padding = #5
R1^"ça marche !"
```



Un silence multimesures placé directement après une commande `\partial` risque fort de perturber le vérificateur de limites et numéros de mesure.

Commandes prédéfinies

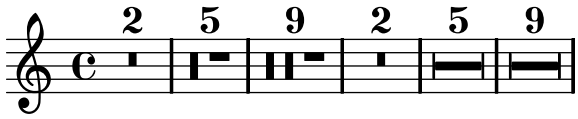
`\textLengthOn`, `\textLengthOff`, `\fermataMarkup`, `\compressFullBarRests`, `\expandFullBarRests`.

Morceaux choisis

Modifier l'apparence d'un silence multi-mesures

Dans le cas où ce silence dure moins de dix mesures, LilyPond imprime sur la portée des « silences d'église » – *Kirchenpause* en allemand – et qui sont une simple suite de rectangles. La propriété `expand-limit` permet d'obtenir un silence unique :

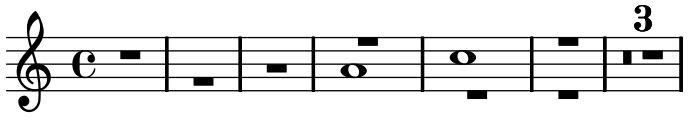
```
\relative c'' {
  \compressFullBarRests
  R1*2 | R1*5 | R1*9
  \override MultiMeasureRest #'expand-limit = #3
  R1*2 | R1*5 | R1*9
}
```



Positionnement des silences multi-mesures

Si l'on peut positionner verticalement un silence simple en le rattachant à une note, il n'en va pas de même pour un silence multi-mesures. Néanmoins, et uniquement dans le cadre de musique polyphonique, les silences multi-mesures sont positionnés différemment selon qu'ils appartiennent à une voix au numéro pair ou impair. Le positionnement des silences multi-mesures peut se contrôler ainsi :

```
\relative c'' {
  % MMR - Multi-Measure Rest
  % MMRs by default are set under the fourth line
  R1
  % They can be moved with an override
  \override MultiMeasureRest #'staff-position = #-2
  R1
  % A value of 0 is the default position;
  % the following trick moves the rest to the center line
  \override MultiMeasureRest #'staff-position = #-0.01
  R1
  % MMRs in odd-numbered voices are under the top line
  << { R1 } \\ { a1 } >>
  % MMRs in even-numbered voices are under the bottom line
  << { c1 } \\ { R1 } >>
  % They remain separated even in empty measures
  << { R1 } \\ { R1 } >>
  % This brings them together even though there are two voices
  \compressFullBarRests
  <<
    \revert MultiMeasureRest #'staff-position
    { R1*3 }
    \\
    \revert MultiMeasureRest #'staff-position
    { R1*3 }
  >>
}
```



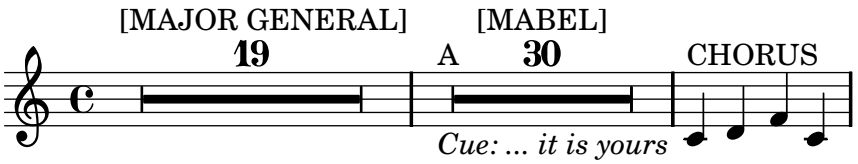
Ajout de texte à un silence multi-mesures

Lorsque du texte est attaché à un silence multi-mesures, il sera centré dans la mesure, au-dessus ou en dessous de la portée. Afin d'étirer la mesure dans le cas ou ce texte est relativement long, il suffit d'insérer un silence invisible auquel on attache le texte en question, avant le silence multi-mesures.

Rappelez-vous qu'un silence invisible génère une barre de mesure. Le texte attaché à ce silence invisible sera alors aligné sur la gauche de là où serait positionnée la note. Cependant, si la taille de la mesure est déterminée par la longueur du texte, il apparaîtra comme centré.

```
\relative c' {
  \compressFullBarRests
  \textLengthOn
```

```
  s1*0^\markup { [MAJOR GENERAL] }
  R1*19
  s1*0_\markup { \italic { Cue: ... it is yours } }
  s1*0^\markup { A }
  R1*30^\markup { [MABEL] }
  \textLengthOff
  c4^\markup { CHORUS } d f c
}
```



Voir aussi

Glossaire musicologique : Section “silence multimesures” dans *Glossaire*.

Manuel de notation : [Durées], page 37, Section 1.8 [Texte], page 193, Section 1.8.2 [Mise en forme du texte], page 202, [Commentaires textuels], page 194.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “MultiMeasureRest” dans *Référence des propriétés internes*. Section “MultiMeasureRestNumber” dans *Référence des propriétés internes*, Section “MultiMeasureRestText” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Vous ne pouvez pas utiliser de doigtés (p. ex. `R1*10-4`) pour positionner des nombres au dessus d’un silence multimesure, le numéro de doigt (4) risquant de chevaucher le nombre de mesures à compter (10).

Condenser plusieurs silences en un unique silence multimesures ne peut être automatisé.
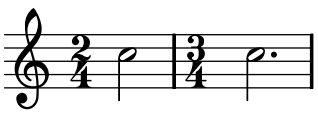
Les silences multimesures peuvent générer des collisions avec d’autres silences.

1.2.3 Gravure du rythme

Métrique

Le chiffre de mesure indique le mètre d’une pièce : une alternance régulière de temps forts et de temps faibles. Il est indiqué par une fraction au début de la portée :

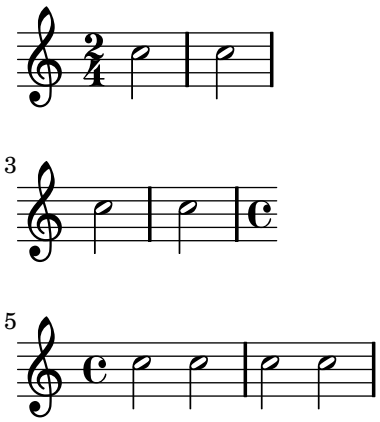
```
\time 2/4 c2
\time 3/4 c2.
```



La métrique est imprimée en début de morceau, et à chaque fois qu’elle est modifiée. Si cette modification intervient au niveau d’un saut de ligne, une métrique « de précaution » sera imprimée avant de passer à la ligne suivante. Ce comportement par défaut peut être modifié, comme indiqué au chapitre Section 5.4.7 [Visibilité des objets], page 475.
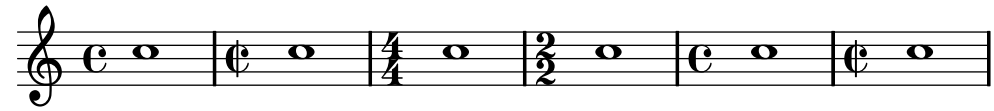
```
\time 2/4
c2 c
\break
```

```
c c
\break
\time 4/4
c c c c
```



Le symbole de métrique utilisé pour les mesures à 2/2 et 4/4 peut être changé pour un style numérique :

```
% Default style
\time 4/4 c1
\time 2/2 c1
% Change to numeric style
\numericTimeSignature
\time 4/4 c1
\time 2/2 c1
% Revert to default style
\defaultTimeSignature
\time 4/4 c1
\time 2/2 c1
```
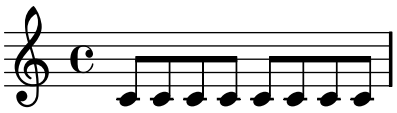


Les métriques anciennes font l'objet d'un Section "chapitre particulier" dans *Manuel de notation*.

En plus de déterminer la métrique qui sera imprimée, la commande `\time` réglera aussi les valeurs par défaut des propriétés `baseMoment`, `beatStructure` et `beamExceptions`. Les valeurs prédéterminées par défaut de ces différentes propriétés sont inscrites dans le fichier '`scm/time-signature-settings.scm`'. Vous pouvez les modifier de la manière suivante :

```
\score {
  \relative c' {
    \overrideTimeSignatureSettings
      #'(4 . 4)  % timeSignatureFraction
      #'(1 . 4)  % baseMomentFraction
      #'(3 1)    % beatStructure
      #'()       % beamExceptions
    \time 4/4
    \repeat unfold 8 { c8 } |
  }
```
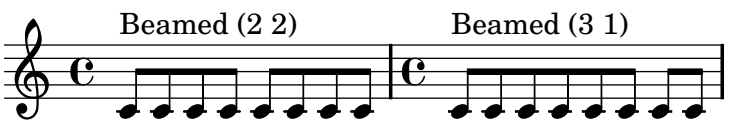
```
}
```



`\overrideTimeSignatureSettings` prend quatre arguments :

1. *`timeSignatureFraction`*, un doublet Scheme indiquant la métrique,
2. *`baseMomentFraction`*, un doublet Scheme comprenant les numérateur et dénominateur de la fraction indiquant la base de la pulsation,
3. *`beatStructure`*, une liste Scheme indiquant la structure de cette pulsation, en unité de base,
4. *`beamExceptions`*, une liste associative des règles de ligature pour cette métrique, en dehors de celles basées sur le temps comme indiqué à la rubrique [Définition des règles de ligature automatique], page 73.

Le contexte auquel doivent s'appliquer les dérogations introduites par un `\overrideTimeSignatureSettings` doit exister avant l'appel de ces réglages. Autrement dit, vous devrez explicitement mentionner le contexte avant que ne survienne l'appel à `\overrideTimeSignatureSettings` s'il n'y a pas déjà d'élément musical.
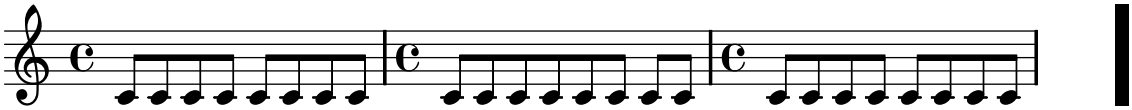
```
\score {
  \relative c' {
    % This call will fail because the context isn't yet instantiated
    \overrideTimeSignatureSettings
      #'(4 . 4)  % timeSignatureFraction
      #'(1 . 4)  % baseMomentFraction
      #'(3 1)    % beatStructure
      #'()       % beamExceptions
    \time 4/4
    c8^\markup {"Beamed (2 2)"}
    \repeat unfold 7 { c8 } |
    % This call will succeed
    \overrideTimeSignatureSettings
      #'(4 . 4)  % timeSignatureFraction
      #'(1 . 4)  % baseMomentFraction
      #'(3 1)    % beatStructure
      #'()       % beamExceptions
    \time 4/4
    c8^\markup {"Beamed (3 1)"}
    \repeat unfold 7 { c8 } |
  }
}
```



Vous pouvez revenir à tout moment aux réglages prédéterminés d'une métrique :

```
\score{
  \relative c' {
    \repeat unfold 8 { c8 } |
```

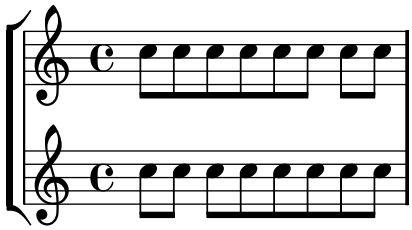
```
    \overrideTimeSignatureSettings
      #'(4 . 4)  % timeSignatureFraction
      #'(1 . 4)  % baseMomentFraction
      #'(3 1)    % beatStructure
      #'()       % beamExceptions
    \time 4/4
    \repeat unfold 8 { c8 } |
    \revertTimeSignatureSettings #'(4 . 4)
    \time 4/4
    \repeat unfold 8 { c8 } |
  }
}
```



Le fait de déplacer du contexte `Score` au contexte `Staff` à la fois le `Timing_translator` et le `Default_bar_line_engraver` permet d'obtenir des réglages particuliers pour les différentes portées d'un regroupement :

```
\score {
  \new StaffGroup <<
     \new Staff {
        \overrideTimeSignatureSettings
          #'(4 . 4)  % timeSignatureFraction
          #'(1 . 4)  % baseMomentFraction
          #'(3 1)    % beatStructure
          #'()       % beamExceptions
        \time 4/4
        \repeat unfold 8 {c''8}
     }
     \new Staff {
        \overrideTimeSignatureSettings
          #'(4 . 4)  % timeSignatureFraction
          #'(1 . 4)  % baseMomentFraction
          #'(1 3)    % beatStructure
          #'()       % beamExceptions
        \time 4/4
        \repeat unfold 8 {c''8}
     }
  >>
  \layout {
    \context {
      \Score
      \remove "Timing_translator"
      \remove "Default_bar_line_engraver"
    }
    \context {
      \Staff
      \consists "Timing_translator"
      \consists "Default_bar_line_engraver"
    }
```

```
  }
}
```



Commandes prédéfinies

`\numericTimeSignature`, `\defaultTimeSignature`.

Morceaux choisis

Changement de métrique sans affecter les règles de ligature

La commande `\time` gère les propriétés `timeSignatureFraction`, `beatLength`, `beatGrouping` et `measureLength` dans le contexte `Timing`, normalement rattaché à `Score`. Le fait de modifier la valeur de `timeSignatureFraction` aura pour effet de changer l'apparence du symbole affiché sans pour autant affecter les autres propriétés de la métrique :

```
\markup {
  This snippet is deprecated as of 2.13.5 and will be removed in 2.14
}
```
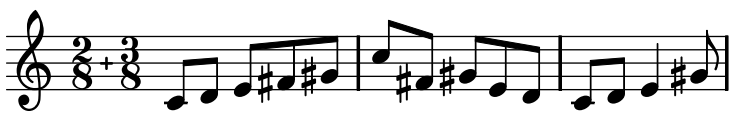
This snippet is deprecated as of 2.13.5 and will be removed in 2.14

Métrique décomposée

Des métriques telles que « 5/8 » peuvent s'interpréter sous une forme décomposée — « 3/8 + 2/8 » par exemple — qui combine plusieurs métriques. LilyPond est capable de rendre ce type de notation, plus facile à lire et à interpréter, en imprimant cette métrique composite et en adaptant les règles de ligature automatique en conséquence.

```
#(define ((compound-time one two num) grob)
   (grob-interpret-markup grob
                          (markup #:override '(baseline-skip . 0) #:number
                                  (#:line ((#:column (one num))
                                           #:vcenter "+"
                                           (#:column (two num)))))))
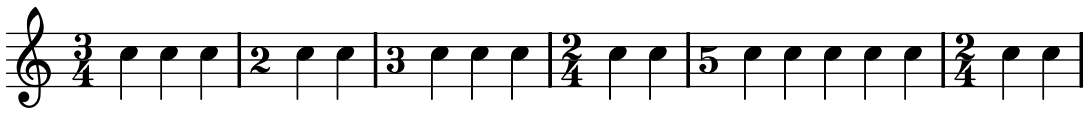
\relative c' {
  \override Staff.TimeSignature #'stencil = #(compound-time "2" "3" "8")
  \time 5/8
  \set Staff.beatStructure = #'(2 3)
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```



*Affichage seulement du numérateur d'une métrique (au*lieu d'une fraction)

La métrique est parfois indiquée non pas par une fraction (p.ex. 7/4) mais simplement par son numérateur (7 dans ce cas). L'instruction `\override Staff.TimeSignature #'style = #'single-digit` permet de déroger au style par défaut de manière permanente – un `\revert Staff.TimeSignature #'style` d'annuler ces modifications. Lorsque cette métrique sous le forme d'une seul chiffre ne se présente qu'une seule fois, il suffit de faire précéder l'instruction `\override` d'un simple `\once`.

```
\relative c'' {
  \time 3/4
  c4 c c
  % Change the style permanently
  \override Staff.TimeSignature #'style = #'single-digit
  \time 2/4
  c4 c
  \time 3/4
  c4 c c
  % Revert to default style:
  \revert Staff.TimeSignature #'style
  \time 2/4
  c4 c
  % single-digit style only for the next time signature
  \once \override Staff.TimeSignature #'style = #'single-digit
  \time 5/4
  c4 c c c c
  \time 2/4
  c4 c
}
```



Voir aussi

Glossaire musicologique : Section "métrique" dans *Glossaire*.

Manuel de notation : [Métriques anciennes], page 359, [Gestion du temps], page 100.
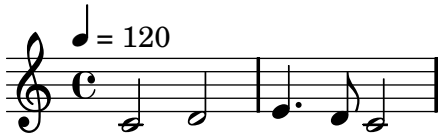
Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Références des propriétés internes : Section "TimeSignature" dans *Référence des propriétés internes*, Section "Timing_translator" dans *Référence des propriétés internes*.

Indication métronomique
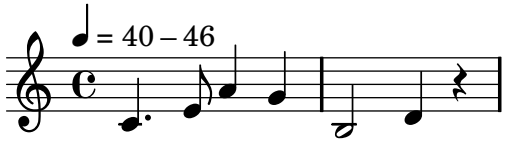
Une indication métronomique s'insère tout simplement comme ceci :

```
\tempo 4 = 120
c2 d
e4. d8 c2
```
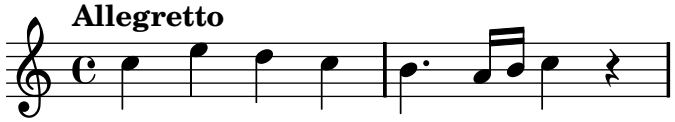


Lorsque le réglage précis du métronome est laissé à l'appréciation de l'exécutant, vous pouvez cependant lui fournir une plage :

```
\tempo 4 = 40 ~ 46
c4. e8 a4 g
b,2 d4 r
```
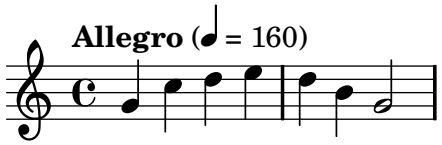


Vous pouvez préférer une indication textuelle :

```
\tempo "Allegretto"
c4 e d c
b4. a16 b c4 r4
```
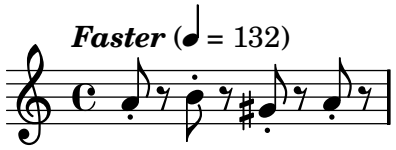


Lorsque vous combinez des indications métronomiques sous forme textuelle et numérique, l'indication numérique sera placée entre parenthèses :

```
\tempo "Allegro" 4 = 160
g4 c d e
d4 b g2
```
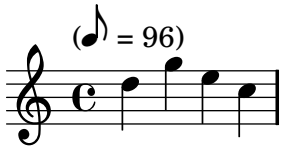


En matière d'indication textuelle, vous pouvez utiliser n'importe quel objet de type *markup*, comme ici :

```
\tempo \markup { \italic Faster } 4 = 132
a8-. r8 b-. r gis-. r a-. r
```



Mentionner une indication textuelle vide vous permet de mettre entre parenthèses l'indication numérique :

```
\tempo "" 8 = 96
d4 g e c
```



Morceaux choisis

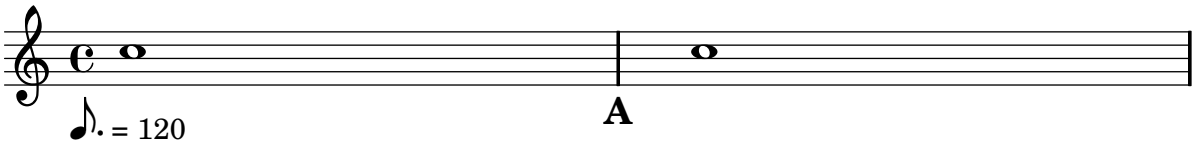
Impression du métronome et des repères sous la portée

Les indications de tempo et les marques de repère s'impriment par défaut au-dessus de la portée. Le fait de régler en conséquence la propriété `direction` des objets `MetronomeMark` ou `RehearsalMark` les placera au-dessous de la portée.

```
\layout { ragged-right = ##f }

{
  % Metronome marks below the staff
  \override Score.MetronomeMark #'direction = #DOWN
  \tempo 8. = 120
  c''1
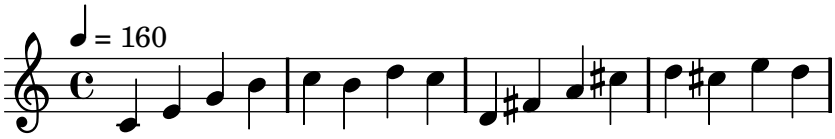
  % Rehearsal marks below the staff
  \override Score.RehearsalMark #'direction = #DOWN
  \mark \default
  c''1
}
```



Changement de tempo sans indication sur la partition

Vous pouvez indiquer un changement de tempo pour le fichier MIDI sans pour autant l'imprimer. Il suffit alors de le rendre invisible aux interprètes.
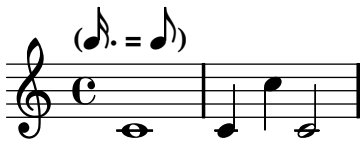
```
\score {
  \new Staff \relative c' {
    \tempo 4 = 160
    c4 e g b
    c4 b d c
    \set Score.tempoHideNote = ##t
    \tempo 4 = 96
    d,4 fis a cis
    d4 cis e d
  }
  \layout { }
  \midi { }
}
```



Création d'une indication métronomique sous forme d'étiquette

Vous pouvez créer des indications de tempo sous la forme d'étiquettes textuelles – des objets `markup` – notamment des équivalences. Cependant, elles n'apparaîtront pas dans le fichier MIDI.

```
\relative c' {
  \tempo \markup {
    \concat {
      (
      \smaller \general-align #Y #DOWN \note #"16." #1
      " = "
      \smaller \general-align #Y #DOWN \note #"8" #1
      )
    }
  }
  c1
  c4 c' c,2
}
```



Pour de plus amples détails, veuillez consulter Section 1.8.2 [Mise en forme du texte], page 202.

Voir aussi

Glossaire musicologique : Section “métronome” dans *Glossaire*, Section “indication métronomique” dans *Glossaire*, Section “indication du tempo” dans *Glossaire*, Section “marque de métronome” dans *Glossaire*.

Manuel de notation : Section 1.8.2 [Mise en forme du texte], page 202, Section 3.5 [Sortie MIDI], page 402.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “MetronomeMark” dans *Référence des propriétés internes*.
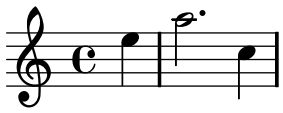
Levées

Les mesures incomplètes, telles que les anacrouses ou levées, doivent être entrées avec la commande `\partial`. La syntaxe de cette commande est

```
\partial durée
```
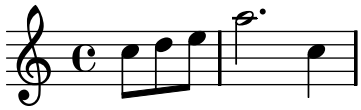
où *`durée`* correspond à la valeur rythmique devant être ajoutée avant la première mesure complète :

```
\partial 4 e4 |
a2. c,4 |
```



Une mesure incomplète peut être de n’importe quelle longueur inférieure à la mesure complète :

```
\partial 8*3 c8 d e |
a2. c,4 |
```



Le programme traduit cette commande `\partial` *`durée`* en

```
\set Timing.measurePosition = -durée
```

Par exemple, `\partial 8*3` est traduit en :

```
\set Timing.measurePosition = #(ly:make-moment -3 8)
```

La propriété `measurePosition` contient un nombre rationnel qui indique, à ce point précis, où l'on en est de la mesure. Notez qu'il s'agit du résultat d'une soustraction ; `\partial 4` signifie, pour le programme : « Dans cette mesure, il reste juste une noire ».

Voir aussi

Glossaire musicologique : Section "anacrouse" dans *Glossaire*.

Manuel de notation : [Notes d'ornement], page 95.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Références des propriétés internes : Section "Timing_translator" dans *Référence des propriétés internes*.
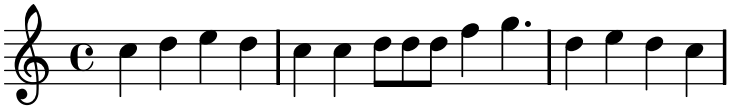
Problèmes connus et avertissements

`\partial` n'est destiné à être utilisé qu'en début de pièce. Si on l'utilise ailleurs qu'au début, des messages d'erreurs peuvent s'afficher. Utilisez plutôt `\set Timing.measurePosition` en pareil cas.

Musique sans métrique
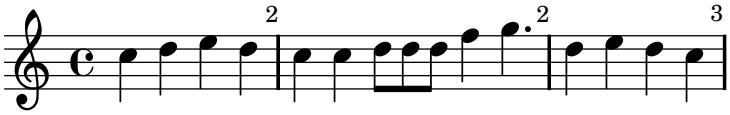
Les barres de mesure et les numéros de mesure sont calculés automatiquement, ce qui n'est pas souhaitable dans le cas d'une musique non mesurée – les cadences par exemple. Les commandes `\cadenzaOn` et `\cadenzaOff` permettent de désactiver et de rétablir ce comportement automatique.

```
c4 d e d
\cadenzaOn
c4 c d8[ d d] f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```
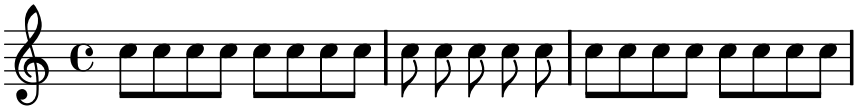


La numérotation reprend dès la fin d'une cadence, comme si celle-ci n'avait pas existé :

```
% Show all bar numbers
\override Score.BarNumber #'break-visibility = #all-visible
c4 d e d
\cadenzaOn
c4 c d8[ d d] f4 g4.
\cadenzaOff
\bar "|"
d4 e d c
```



L'instruction `\cadenzaOn` désative les ligatures automatiques ; elles seront réactivées après un `\cadenzaOff`. Toutes les ligatures devront donc être indiquées de manière explicite tout au long de la cadence – voir [Barres de ligature manuelles], page 80).

```
\repeat unfold 8 { c8 }
\cadenzaOn
\repeat unfold 5 { c8 }
\bar"|"
\cadenzaOff
\repeat unfold 8 { c8 }
```



Notez bien que ces commandes prédéfinies affecteront toutes les portées de la partition, même si vous ne les placez que dans un simple contexte `Voice`. Pour éviter ce désagrément, transférez le `Timing_translator` du contexte `Score` au contexte `Staff`, comme indiqué au chapitre [Notation polymétrique], page 64.

Commandes prédéfinies

`\cadenzaOn`, `\cadenzaOff`.

Voir aussi

Glossaire musicologique : Section "cadence" dans *Glossaire*.

Manuel de notation : Section 5.4.7 [Visibilité des objets], page 475, [Notation polymétrique], page 64.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Problèmes connus et avertissements

LilyPond ne change de ligne ou de page qu'au niveau des barres de mesure. Si votre musique non mesurée s'étend sur plus d'une ligne, il vous faudra insérer des barres de mesure invisibles,

```
\bar ""
```

pour indiquer où des sauts de ligne peuvent prendre place.

Au risque d'obtenir des messages d'erreur inhabituels, il est fortement recommandé de créer explicitement un contexte `Voice` lorsqu'un morceau débute par un `\cadenzaOn`.

```
\new Voice {
  \relative c' {
    \cadenzaOn
    c16[^"Solo Free Time" d e f] g2.
    \bar "||"
    \cadenzaOff
  }
}
```

Notation polymétrique

LilyPond gère les métriques composites, aussi bien de manière explicite que de manière détournée – modification de l'indicateur de métrique et échelonnement de la durée des notes.

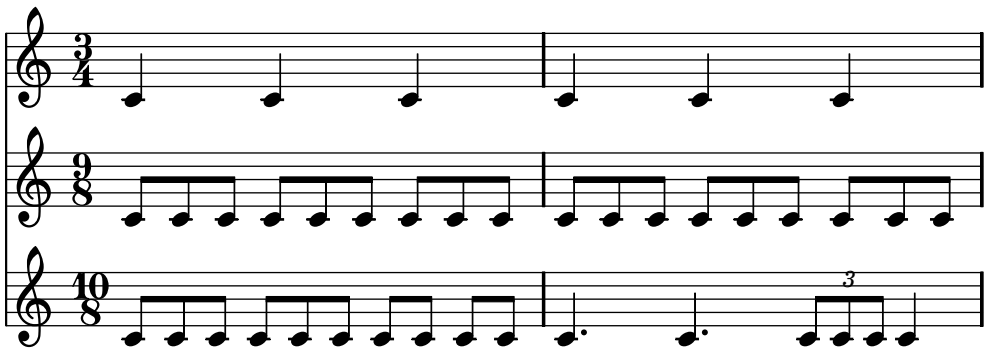
Portées aux métriques différentes, mesures d'égale longueur

Il suffit, pour obtenir cette forme de notation, de tout d'abord affecter une même métrique aux différentes portées. L'indicateur de métrique sera ensuite remplacé dans chacune des portées

par un quotient donné en argument à la propriété `timeSignatureFraction`. La durée des notes sera enfin proratisée selon la métrique commune (voir [Métrique], page 54) grâce à la fonction `\scaleDurations`, qui s'utilise tout comme la commande `\times` – sans toutefois créer de crochet de nolet (voir [Changement d'échelle des durées], page 43).

L'exemple suivant utilise parallèlement des mesures à 3/4, 9/8 et 10/8. Pour la deuxième portée les durées sont multipliées par 2/3 de telle sorte que 2/3 * 9/8 = 3/4 ; pour la troisième elles sont multipliées par 3/5, de telle sorte que 3/5 * 10/8 = 3/4. Les ligatures devront être explicites, la fonction d'échelonnement venant perturber les règles de ligature automatique.

```
\relative c' <<
  \new Staff {
    \time 3/4
    c4 c c |
    c c c |
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = #'(9 . 8)
    \scaleDurations #'(2 . 3)
    \repeat unfold 6 { c8[ c c] }
  }
  \new Staff {
    \time 3/4
    \set Staff.timeSignatureFraction = #'(10 . 8)
    \scaleDurations #'(3 . 5) {
      \repeat unfold 2 { c8[ c c] }
      \repeat unfold 2 { c8[ c] } |
      c4. c4. \times 2/3 { c8[ c c] } c4
    }
  }
>>
```



Portées aux métriques différentes, mesures de longueur inégale

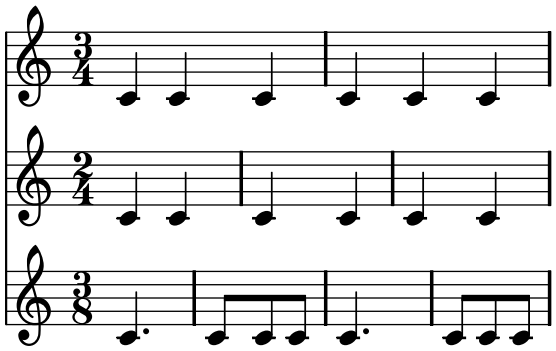
Il arrive aussi que chaque portée ait sa propre métrique. Vous y parviendrez en déplaçant le `Timing_translator` et le `Default_bar_line_engraver` dans le contexte `Staff`.

```
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
  }
```

```
  \context {
    \Staff
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  }
}

% Now each staff has its own time signature.

\relative c' <<
  \new Staff {
    \time 3/4
    c4 c c |
    c c c |
  }
  \new Staff {
    \time 2/4
    c4 c |
    c c |
    c c |
  }
  \new Staff {
    \time 3/8
    c4. |
    c8 c c |
    c4. |
    c8 c c |
  }
>>
```



Morceaux choisis
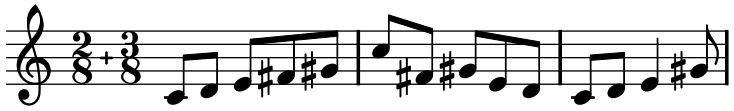
Métrique décomposée

Des métriques telles que « 5/8 » peuvent s'interpréter sous une forme décomposée — « 3/8 + 2/8 » par exemple — qui combine plusieurs métriques. LilyPond est capable de rendre ce type de notation, plus facile à lire et à interpréter, en imprimant cette métrique composite et en adaptant les règles de ligature automatique en conséquence.

```
#(define ((compound-time one two num) grob)
   (grob-interpret-markup grob
                          (markup #:override '(baseline-skip . 0) #:number
                                  (#:line ((#:column (one num))
```

```
                                                #:vcenter "+"
                                                (#:column (two num)))))))

\relative c' {
  \override Staff.TimeSignature #'stencil = #(compound-time "2" "3" "8")
  \time 5/8
  \set Staff.beatStructure = #'(2 3)
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```



Voir aussi

Glossaire musicologique : Section “polymétrie” dans *Glossaire*, Section “métrique composite” dans *Glossaire*, Section “métrique” dans *Glossaire*.

Manuel de notation : [Métrique], page 54, [Changement d’échelle des durées], page 43.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Références des propriétés internes : Section “TimeSignature” dans *Référence des propriétés internes*, Section “Timing_translator” dans *Référence des propriétés internes*, Section “Default_bar_line_engraver” dans *Référence des propriétés internes*, Section “Staff” dans *Référence des propriétés internes*.

Problèmes connus et avertissements
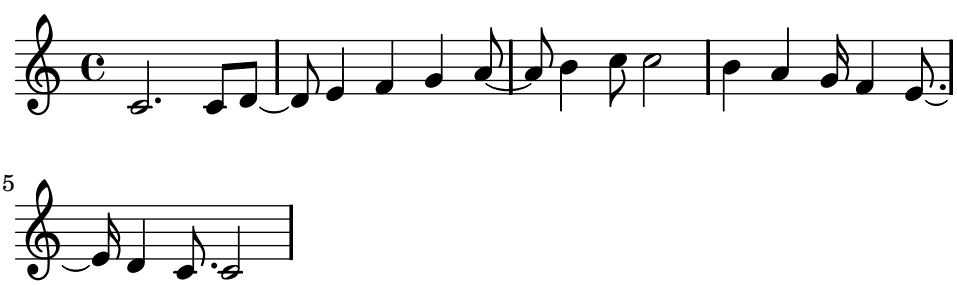
L’utilisation de métriques différentes en parallèle entraine un alignement vertical. De ce fait, les barres de mesure ont tendance à fausser l’espacement régulier entre les différentes portées.

Découpage automatique des notes

On peut convertir automatiquement les notes longues en notes liées. Il suffit pour cela de remplacer le graveur `Note_heads_engraver` par le graveur `Completion_heads_engraver`. Dans l’exemple suivant, les notes qui dépassent de la mesure sont divisées et liées.

```
\new Voice \with {
  \remove "Note_heads_engraver"
  \consists "Completion_heads_engraver"
}

{ c2. c8 d4 e f g a b c8 c2 b4 a g16 f4 e d c8. c2 }
```



Ce graveur divise toutes les notes qui sortent de la mesure, et insère des liaisons de prolongation. Dans la pratique, cette fonctionnalité permet de déboguer des partitions complexes : si les mesures ne sont pas entièrement remplies, alors les liaisons de prolongation montrent exactement la durée des décalages de mesure.

Voir aussi

Glossaire musicologique : Section “liaison de tenue” dans *Glossaire*.

Manuel d’initiation : Section “Tout savoir sur les graveurs” dans *Manuel d’initiation*, Section “Ajout et suppression de graveurs” dans *Manuel d’initiation*.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Références des propriétés internes : Section “Note_heads_engraver” dans *Référence des propriétés internes*, Section “Completion_heads_engraver” dans *Référence des propriétés internes*, Section “Forbid_line_break_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

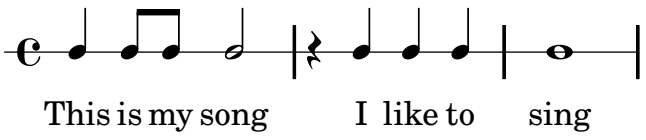
Bien que toutes les durées – particulièrement celles contenant des nolets – ne puissent être exactement représentées avec des notes normales et des points, le graveur `Completion_heads_engraver` n’insérera pas de nolet.

Le `Completion_heads_engraver` affecte seulement les notes, il ne divise pas les silences.

Gravure de lignes rythmiques

Au moyen d’une portée rythmique – *rhythmic staff* en anglais – on peut montrer seulement le rythme d’une mélodie : toutes les notes sont ramenées à la même hauteur, sur une portée d’une seule ligne.

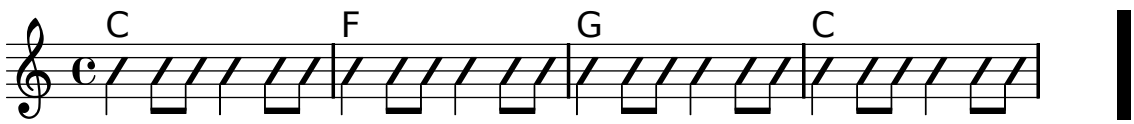
```
<<
  \new RhythmicStaff {
    \new Voice = "myRhythm" {
      \time 4/4
      c4 e8 f g2
      r4 g g f
      g1
    }
  }
  \new Lyrics {
    \lyricsto "myRhythm" {
      This is my song
      I like to sing
    }
  }
>>
```



L’utilisation combinée du `Pitch_squash_engraver` et de `\improvisationOn` permet d’afficher la structure rythmique d’une grille d’accords :

```
<<
  \new ChordNames {
    \chordmode {
```

```
      c1 f g c
    }
  }
  \new Voice \with {
    \consists Pitch_squash_engraver
  } \relative c'' {
    \improvisationOn
    c4 c8 c c4 c8 c
    f4 f8 f f4 f8 f
    g4 g8 g g4 g8 g
    c4 c8 c c4 c8 c
  }
>>
```



Commandes prédéfinies

`\improvisationOn`, `\improvisationOff`.

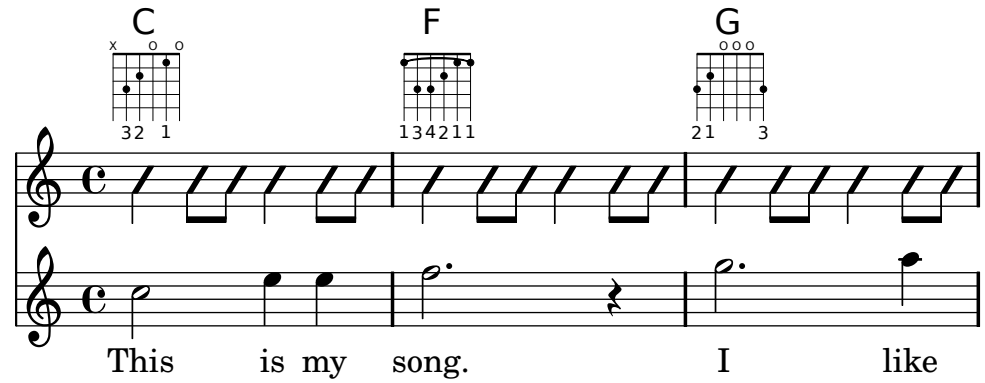
Morceaux choisis

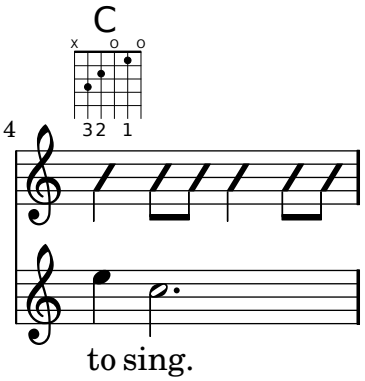
Rythmique et guitare

En matière de notation pour guitare, il arrive que soient indiqués les « coups de gratte » en plus de la mélodie, grilles d'accords et diagrammes de tablature.

```
\include "predefined-guitar-fretboards.ly"
<<
  \new ChordNames {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new FretBoards {
    \chordmode {
      c1 | f | g | c
    }
  }
  \new Voice \with {
    \consists "Pitch_squash_engraver"
  } {
    \relative c'' {
      \improvisationOn
      c4 c8 c c4 c8 c
      f4 f8 f f4 f8 f
      g4 g8 g g4 g8 g
      c4 c8 c c4 c8 c
    }
  }
  \new Voice = "melody" {
    \relative c'' {
```

```
      c2 e4 e4
      f2. r4
      g2. a4
      e4 c2.
    }
  }
  \new Lyrics {
    \lyricsto "melody" {
      This is my song.
      I like to sing.
    }
  }
>>
```





Voir aussi

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.
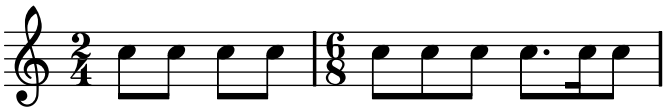
Référence des propriétés internes : Section “RhythmicStaff” dans *Référence des propriétés internes*. Section “Pitch_squash_engraver” dans *Référence des propriétés internes*.

1.2.4 Barres de ligature

Barres de ligature automatiques

LilyPond décide automatiquement de la manière de grouper les notes et d’imprimer les ligatures.
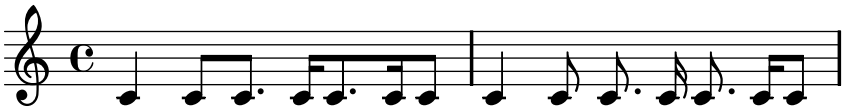
```
\time 2/4 c8 c c c
\time 6/8 c8 c c c8. c16 c8
```



Lorsque ce comportement automatisé n'est pas satisfaisant, on peut définir des groupements manuellement – voir [Barres de ligature manuelles], page 80. Dans le cas où le groupe de notes en question contient un silence, il est **impératif** d'indiquer explicitement les début et fin de la ligature.

Lorsque les ligatures automatiques ne sont pas nécessaires, il suffit de désactiver la fonctionnalité par un `\autoBeamOff` – réactivation par `\autoBeamOn` :

```
c4 c8 c8. c16 c8. c16 c8
\autoBeamOff
c4 c8 c8. c16 c8.
\autoBeamOn
c16 c8
```



Note : Si des ligatures sont utilisées dans les paroles d'une chanson (pour indiquer des mélismes), les ligatures automatiques doivent être désactivées, avec `\autoBeamOff`, et indiquées manuellement.

Note : L'utilisation conjointe de `\partcombine` et de `\autoBeamOff` peut produire des résultats quelque peu surprenants ; ceci fait l'objet d'un exemple particulier à la rubrique morceaux choisis.

Des règles de dérogation au comportement automatique par défaut sont possibles ; voir [Définition des règles de ligature automatique], page 73.

Commandes prédéfinies
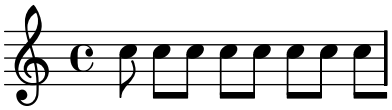
`\autoBeamOff`, `\autoBeamOn`.
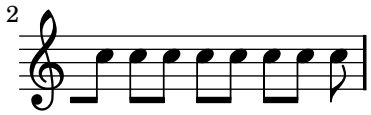
Morceaux choisis

Ligature au moment d'un saut de ligne

Il est normalement impensable qu'un saut de ligne tombe au milieu d'une ligature. LilyPond permet néanmoins de l'obtenir.

```
\relative c'' {
  \override Beam #'breakable = ##t
  c8 c[ c] c[ c] c[ c] c[ \break
  c8] c[ c] c[ c] c[ c] c
}
```
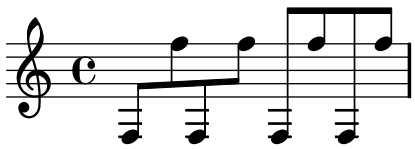




Ligature et directions de hampe inversées

LilyPond insère automatiquement des ligatures coudées — certaines hampes vers le haut, d'autres vers le bas — lorsqu'il détecte un intervalle important entre des têtes de notes. Ce comportement peut être changé par l'intermédiaire de l'objet `auto-knee-gap` — défini par défaut à '`5,5`' espace, plus la largeur et la pente de la ligature en question.

```
{
  f8 f''8 f8 f''8
  \override Beam #'auto-knee-gap = #6
  f8 f''8 f8 f''8
}
```



Partcombine et autoBeamOff

Le fonction `\autoBeamOff` dans le cadre d'un `\partcombine` agit de façon bien particulière ; c'est pourquoi il vaut mieux tout d'abord recourir à

```
\set Staff.autobeaming = ##f
```
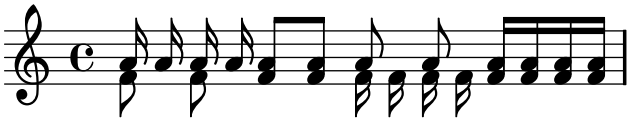
pour désactiver les ligatures automatiques pour l'ensemble de la portée concernée.

L'instruction `\partcombine` fonctionne apparament sur la base de trois voix : solo hampes montantes, solo hampes descendantes et ensemble hampes montantes.

Lorsque `\autoBeamOff` apparaît dans le premier argument de la combinaison, il s'applique à la voix active à ce moment précis, qu'il s'agisse du solo hampes montantes ou du combiné hampes montantes. Lorsqu'elle est introduite dans le second argument, la commande `\autoBeamOff` s'appliquera au solo hampes descendantes.

Vous devrez donc, afin que `\autoBeamOff` soit pleinement opérationnel dans le cadre d'un `\partcombine`, l'introduire aux **trois** niveaux.

```
{
  %\set Staff.autoBeaming = ##f % turns off all autobeaming
  \partcombine
  {
    \autoBeamOff % applies to split up stems
    \repeat unfold 4 a'16
    %\autoBeamOff % applies to combined up stems
    \repeat unfold 4 a'8
    \repeat unfold 4 a'16
  }
  {
    \autoBeamOff % applies to down stems
    \repeat unfold 4 f'8
    \repeat unfold 8 f'16 |
  }
}
```



Voir aussi

Manuel de notation : [Barres de ligature manuelles], page 80, [Définition des règles de ligature automatique], page 73.

Fichiers d'initialisation : '`scm/auto-beam.scm`'.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Références des propriétés internes : Section "Auto_beam_engraver" dans *Référence des propriétés internes*, Section "Beam_engraver" dans *Référence des propriétés internes*, Section "Beam" dans *Référence des propriétés internes*. Section "BeamEvent" dans *Référence des propriétés internes*, Section "BeamForbidEvent" dans *Référence des propriétés internes*, Section "beam-interface" dans *Référence des propriétés internes*, Section "unbreakable-spanner-interface" dans *Référence des propriétés internes*.
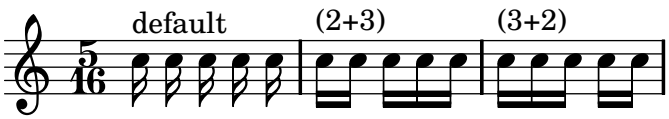
Problèmes connus et avertissements

Les ligatures peuvent générer des collisions avec des têtes de note ou altérations appartenant à d'autres voix.

Définition des règles de ligature automatique

Dans la plupart des cas, une ligature automatique se termine à la fin d'une pulsation. Ce sont les propriétés de contexte `baseMoment` et `beatStructure` qui détermineront là où prend fin la pulsation. La propriété `beatStructure` est constituée d'une liste d'éléments Scheme qui définit la longueur de chaque pulsation, prenant `baseMoment` comme unité. L'unité de base (le `baseMoment`) correspond par défaut au dénominateur de la métrique. D'autre part, chaque unité de `baseMoment` constitue une seule pulsation.

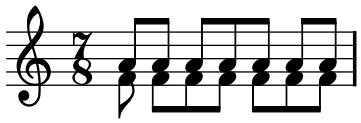
```
\time 5/16
c16^"default" c c c c |
\set Timing.beatStructure = #'(2 3)
c16^"(2+3)" c c c c |
\set Timing.beatStructure = #'(3 2)
c16^"(3+2)" c c c c |
```



Les effets de règles de ligature peuvent être restreints à un contexte particulier. En l'absence de règle particulière déterminée dans un contexte de niveau inférieur, les règles définies au niveau directement supérieur s'appliqueront.

```
\new Staff {
  \time 7/8
  \set Staff.beatStructure = #'(2 3 2)
  <<
    \new Voice = one {
      \relative c'' {
        a8 a a a a a a
      }
    }
    \new Voice = two {
      \relative c' {
```
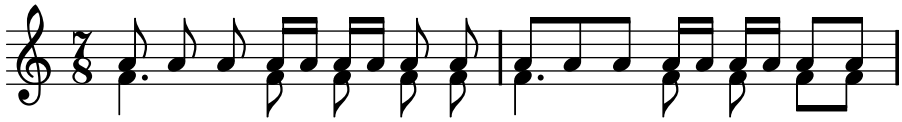
```
        \voiceTwo
        \set Voice.beatStructure = #'(1 3 3)
        f8 f f f f f f
      }
    }
  >>
}
```



Lorsque plusieurs voix cohabitent sur une même portée et que les règles de ligature doivent s'appliquer sans distinction, il faut spécifier que ces règles affectent le contexte `Staff` :

```
\time 7/8
% rhythm 3-1-1-2
% Context Voice specified --  does not work correctly
% Because of autogenerated voices, all beating will
% be at baseMoment (1 . 8)
\set beatStructure = #'(3 1 1 2)
<< {a8 a a a16 a a a a8 a} \\ {f4. f8 f f f} >>

% Works correctly with context Staff specified
\set Staff.beatStructure = #'(3 1 1 2)
<< {a8 a a a16 a a a a8 a} \\ {f4. f8 f f f} >>
```



Vous pouvez ajuster la valeur de `baseMoment` afin d'obtenir des ligatures selon vos besoin. Notez cependant que la valeur de `beatStructure` devra être en corrélation avec cette nouvelle valeur de `baseMoment`.

```
\time 5/8
\set Timing.baseMoment = #(ly:make-moment 1 16)
\set Timing.beatStructure = #'(7 3)
\repeat unfold 10 { a16 }
```
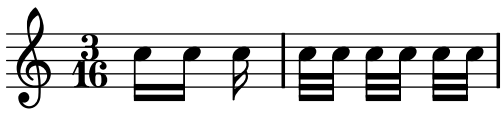


`baseMoment` constitue un **moment**, autrement dit une unité de durée musicale. La fonction Scheme `ly:make-moment` est tout particulièrement chargée de créer cette quantité de type *moment* – voir [Gestion du temps], page 100 pour plus de précisions.

La pulsation – *baseMoment* en anglais – découle directement de la métrique telle que définie par la commande `\time`. Elle est par défaut égale à un sur le dénominateur de la métrique. Les exceptions à cette règle par défaut sont répertoriées dans le fichier '`scm/time-signature-settings.scm`'. Pour savoir comment jouer avec la valeur de `baseMoment` selon la métrique, reportez vous au chapitre [Métrique], page 54.

Les règles de ligature et de subdivision spécifiques sont enregistrées dans la propriété `beamExceptions`. Ses valeurs par défaut, rangées par métrique et type de règle, sont répertoriées dans le fichier '`scm/time-signature-settings.scm`'.

Les règles spécifiques autres que celles concernant la terminaison des ligatures sont gérées par la propriété `beamExceptions`.

```
\time 3/16
\set Timing.beatStructure = #'(2 1)
\set Timing.beamExceptions =
  #'(                          ;ouvre de la liste associative
     (end .                    ;ceci concerne la terminaison des ligatures
      (                        ;ouvre la liste des points de terminaison
       ((1 . 32) . (2 2 2))    ;règle pour les triples -- groupées à la double
      )))                      %referme la liste
c16 c c |
\repeat unfold 6 { c32 } |
```



`beamExceptions` contient la liste des règles de ligature selon leur type.

Le seul type de règle pris en charge à ce jour est `#'end`, qui concerne les terminaisons.

Chaque règle est constituée d'une liste de doublets associatifs en langage Scheme (un *alist* pour les puristes), qui indique la durée de base et sa règle de regroupement.

```
#'((durée-type1 . groupement-1)
   (durée-type2 . groupement-2)
   (durée-type3 . groupement-3))
```

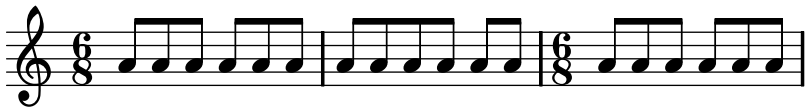
durée-type est constitué d'une paire indiquant la durée de base – par exemple `(1 . 16)` pour une double croche.

groupement est constitué d'une liste Scheme qui indique le regroupement à effectuer, en unité de type de ligature.

> **Note :** La propriété `beamExceptions` doit contenir absolument **toutes** les exceptions. Il n'est en effet pas possible d'en ajouter, modifier ou suprimer *a posteriori*. Cela peut paraître fastidieux, mais toutes les règles de ligature devraient être appréciées avant de les spécifier.

Lorsqu'intervient un changement de métrique, les valeurs par défaut de `Timing.baseMoment`, `Timing.beatStructure` et `Timing.beamExceptions` sont réinitialisées. Il suffit donc, pour revenir aux règles de ligature par défaut d'un contexte `Timing`, de spécifier à nouveau la métrique.
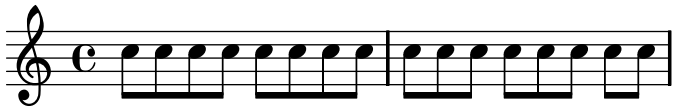
```
\time 6/8
\repeat unfold 6 { a8 }
% group (4 + 2)
\set Timing.beatStructure = #'(4 2)
\repeat unfold 6 { a8 }
% go back to default behavior
\time 6/8
\repeat unfold 6 { a8 }
```



Les règles de ligature automatique par défaut sont répertoriées, par métrique, dans le fichier ‘scm/time-signature-settings.scm’. Les manières de déroger à ce comportement sont abordées au chapitre [Métrique], page 54.

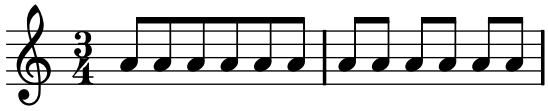
De nombreuses règles de ligature automatique comportent une clé `beamExceptions`. Par exemple, s'il n'y a que des croches dans une mesure à 4/4, celles-ci seront réparties en deux groupes. Le fait de ne pas réinitialiser `beamExceptions` lors d'un aménagement de la pulsation – l'élément `beatStructure` – empêchera l'application de cette dérogation.

```
\time 4/4
\set Timing.baseMoment = #(ly:make-moment 1 8)
\set Timing.beatStructure = #'(3 3 2)
% This won't beam (3 3 2) because of beamExceptions
\repeat unfold 8 {c8} |
% This will beam (3 3 2) because we clear beamExceptions
\set Timing.beamExceptions = #'()
\repeat unfold 8 {c8}
```



De la même manière, les croches d'une mesure à 3/4 sont par défaut ligaturées sur la mesure entière. Réinitialiser `beamExceptions` vous permettra alors de ligaturer les croches sur la base du temps.

```
\time 3/4
% by default we beam in (3) due to beamExceptions
\repeat unfold 6 {a8} |
% This will beam (1 1 1) due to beatLength
\set Timing.beamExceptions = #'()
\repeat unfold 6 {a8}
```



Principes de la ligature automatique

Lorsqu'elle est activée, la gestion automatisée des ligatures est directement liée aux propriétés `baseMoment`, `beatStructure` et `beamExceptions`.

Les règles déterminant le positionnement des ligatures automatiques s'appliquent dans l'ordre suivant de priorité :

- Une ligature explicite – indiquée par `[...]` – sera toujours respectée ; sinon
- si une règle explicite de terminaison a été définie grâce à la propriété `beamExceptions` pour un type de ligature spécifique dans la métrique en cours, c'est elle qui s'appliquera ; sinon
- si une règle explicite de terminaison a été définie grâce à la propriété `beamExceptions` pour un type de ligature plus large, c'est elle qui s'appliquera ; sinon
- utilisation des valeurs de `baseMoment` et `beatStructure` pour regrouper les notes par des ligatures.

Le type de ligature correspond à la durée la plus courte dans le groupe.

Pour information, les règles de ligature par défaut sont répertoriées dans le fichier ‘scm/time-signature-settings.scm’.

Morceaux choisis
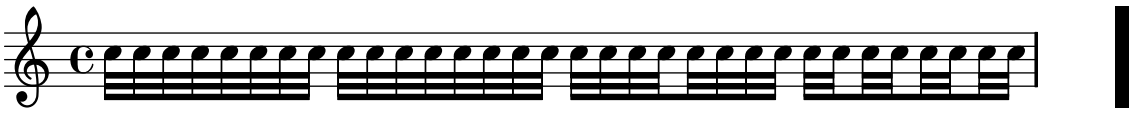
Subdivision des ligatures

Les ligatures d'une succession de notes de durée inférieure à la croche ne sont pas subdivisées par défaut. Autrement dit, tous les traits de ligature (deux ou plus) seront continus. Ce comportement peut être modifié afin de diviser la ligature en sous-groupes grâce à la propriété `subdivideBeams`. Lorsqu'elle est activée, les ligatures seront subdivisées selon un intervalle défini par `baseMoment` ; il n'y aura alors plus qu'un seul trait de ligature entre chaque sous-groupe. Par défaut, `baseMoment` fixe la valeur de référence par rapport à la métrique en vigueur. Il faudra donc lui fournir, à l'aide de la fonction `ly:make-moment`, une fraction correspondant à la durée du sous-groupe désiré comme dans l'exemple ci-dessous. Gardez à l'esprit que, si vous venez à modifier `baseMoment`, vous devrez probablement adapter `beatStrusture` afin qu'il reste en adéquation avec les nouvelles valeurs de `baseMoment`.

```
\relative c'' {
  c32[ c c c c c c c]
  \set subdivideBeams = ##t
  c32[ c c c c c c c]

  % Set beam sub-group length to an eighth note
  \set baseMoment = #(ly:make-moment 1 8)
  \set beatStructure = #'(2 2 2 2)
  c32[ c c c c c c c]

  % Set beam sub-group length to a sixteenth note
  \set baseMoment = #(ly:make-moment 1 16)
  \set beatStructure = #'(4 4 4 4)
  c32[ c c c c c c c]
}
```

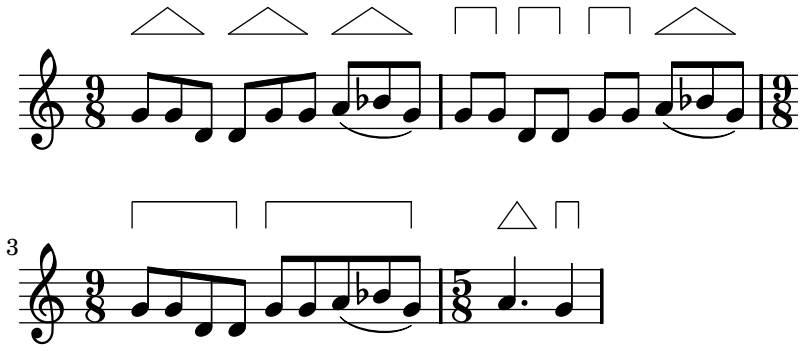


Signes de direction signes de sous-groupe

Les règles de ligature par mesure sont gérées par la propriété `beatStructure`. Ses valeurs par défaut sont répertoriées, par métrique, dans le fichier `scm/time-signature-settings.scm`. Elles sont modifiables grâce à la commande `\set`. La fonction Scheme `set-time-signature` permet quant à elle de définir à la fois la métrique et la pulsation. Celle-ci prend trois arguments : le nombre de pulsations, la durée de la pulsation et le regroupement des pulsations dans la mesure. `\time` et `set-time-signature` s'appliquent tous deux au contexte `Timing` ; ils ne redéfiniront donc pas les valeurs de `beatStructure` ou `baseMoment` lorsque qu'elles sont modifiées dans un contexte de niveau inférieur comme `Voice` par exemple. Si l'on fait appel au `Measure_grouping_engraver`, la fonction `set-time-signature` créera aussi des symboles `MeasureGrouping`. Ces symboles aident à la lecture des œuvres modernes à la rythmique complexe. Dans l'exemple qui suit, la mesure à 9/8 est divisée en 2, 2, 2 et 3, alors que la mesure à 5/8 répond aux règles par défaut contenues dans le fichier `scm/beam-settings.scm`.

```
\score {
  \new Voice \relative c'' {
    \time 9/8
    g8 g d d g g a( bes g) |
    \set Timing.beatStructure = #'(2 2 2 3)
```

```
    g8 g d d g g a( bes g) |
    #(set-time-signature 9 8 '(4 5))
    g8 g d d g g a( bes g) |
    \time 5/8
    a4. g4 |
  }
  \layout {
    \context {
      \Staff
      \consists "Measure_grouping_engraver"
    }
  }
}
```
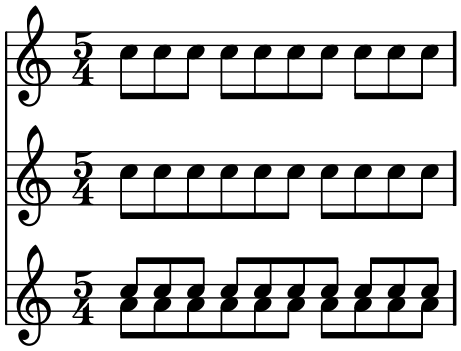


Définition de règles de ligature pour la partition

Les règles de ligatures définies au niveau du contexte `Score` s'appliqueront à toutes les portées. Il est toutefois possible de moduler au niveau `Staff` ou `Voice` :

```
\relative c'' {
  \time 5/4
  % Set default beaming for all staves
  \set Score.baseMoment = #(ly:make-moment 1 8)
  \set Score.beatStructure = #'(3 4 3)
  <<
    \new Staff {
      c8 c c c c c c c c c
    }
    \new Staff {
      % Modify beaming for just this staff
      \set Staff.beatStructure = #'(6 4)
      c8 c c c c c c c c c
    }
    \new Staff {
      % Inherit beaming from Score context
      <<
        {
          \voiceOne
          c8 c c c c c c c c c
        }
        % Modify beaming for this voice only
        \new Voice {
          \voiceTwo
```

```
          \set Voice.beatStructure = #'(6 4)
          a8 a a a a a a a a a
        }
      >>
    }
  >>
}
```



Voir aussi

Fichiers d'initialisation : '`scm/beam-settings.scm`'.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

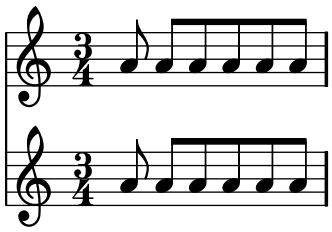
Référence des propriétés internes : Section "Auto_beam_engraver" dans *Référence des propriétés internes*, Section "Beam" dans *Référence des propriétés internes*, Section "BeamForbidEvent" dans *Référence des propriétés internes*, Section "beam-interface" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Si une partition se termine alors qu'une ligature automatique est restée inachevée, cette dernière ligature ne sera pas imprimée du tout. C'est également valable dans le cas d'une musique polyphonique saisie avec la syntaxe `<< ... \\ ... >>`, où une voix se terminerait sans que la dernière ligature ne soit achevée. Le plus simple, en pareil cas, est de spécifier manuellement les dernières ligatures.

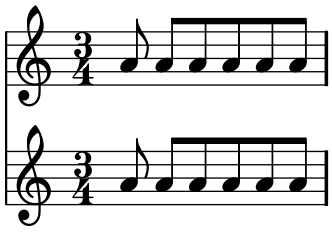
Le traducteur `Timing` est par défaut affecté au contexte `Score`. Définir la métrique dans une portée aura donc des effets sur les ligatures de toutes les autres. Par voie de conséquence, la définition de la métrique apparaissant dans une autre portée annulera les aménagements précédemment apportés aux règles de ligature. Il est donc préférable, pour éviter tout désagrément, de ne spécifier la métrique que dans une seule portée.

```
<<
  \new Staff {
    \time 3/4
    \set Timing.baseMoment = #(ly:make-moment 1 8)
    \set Timing.beatStructure = #'(1 5)
    \set Timing.beamExceptions = #'()
    \repeat unfold 6 { a8 }
  }
  \new Staff {
    \repeat unfold 6 { a8 }
  }
>>
```



Vous pouvez adapter les règles de ligature par défaut pour une métrique particulière de telle sorte que ces règles que vous aurez définies soient toujours prises en compte. La modification des règles de ligature automatiques sont abordées au chapitre [Métrique], page 54.
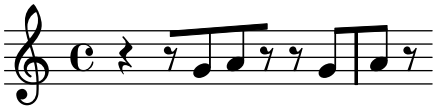
```
<<
  \new Staff {
    \overrideTimeSignatureSettings
      #'(3 . 4)         % timeSignatureFraction
      #'(1 . 8)         % baseMomentFraction
      #'(1 5)           % beatStructure
      #'()              % beamExceptions
    \time 3/4
    \repeat unfold 6 { a8 }
  }
  \new Staff {
    \time 3/4
    \repeat unfold 6 { a8 }
  }
>>
```



Barres de ligature manuelles
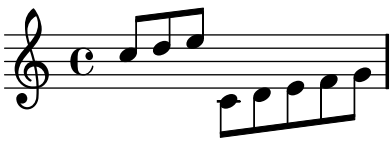
Dans certaines situations, il peut s'avérer nécessaire de supplanter l'algorithme de groupement automatique des notes, par exemple pour prolonger une ligature par-dessus un silence ou une barre de mesure, ou bien pour suivre le rythme des paroles plutôt que celui des notes. Le début et la fin de la ligature sont alors indiqués respectivement par `[` et `]`.

```
r4 r8[ g' a r] r8 g[ | a] r
```
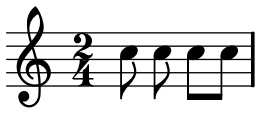


Le positionnement des ligature manuelles se détermine comme pour toute indication attachée à une note :

```
c8^[ d e] c,_[ d e f g]
```
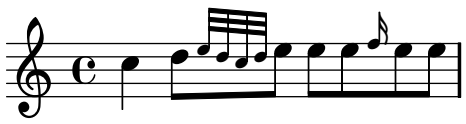


Le fait d'affubler une note particulière d'un `\noBeam` aura pour effet de l'empêcher d'être ligaturée :

```
\time 2/4
c8 c\noBeam c c
```
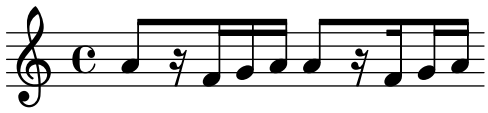


Notes d'ornement et normales font l'objet d'un traitement distinct. Il est dont possible de ligaturer ou non des notes d'ornement sans géner ce qui est en place au niveau de la notation normale.

```
c4 d8[
\grace { e32[ d c d] }
e8] e[ e
\grace { f16 }
e8 e]
```



LilyPond peut déterminer automatiquement les sous-groupes à l'intérieur d'un groupement de notes, bien que le résultat ne soit pas toujours optimal. Les propriétés `stemLeftBeamCount` et `stemRightBeamCount` permettent alors d'ajuster ce comportement. Lorsque l'une ou l'autre de ces propriétés est définie, elle ne s'applique qu'une seule fois, après quoi sa définition est effacée. Dans l'exemple qui suit, le dernier `fa` n'a de ligature supplémentaire que sur sa gauche ; autrement dit, c'est la ligature à la croche qui est importante.

```
a8[ r16 f g a]
a8[ r16
\set stemLeftBeamCount = #2
\set stemRightBeamCount = #1
f16
\set stemLeftBeamCount = #1
g16 a]
```



Commandes prédéfinies

`\noBeam`.

Morceaux choisis

Crochet rectiligne et débordement de ligature

En combinant `stemLeftBeamCount`, `stemRightBeamCount` et des paires de `[]`, vous pourrez obtenir des crochets rectilignes et des ligatures qui débordent à leurs extrémités.

Pour des crochets rectilignes à droite sur des notes isolées, il suffit d'ajouter une paire d'indicateurs de ligature `[]` et de déterminer `stemLeftBeamCount` à zéro, comme dans l'exemple 1.

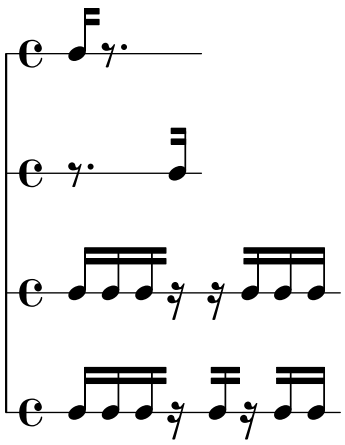
Pour des crochets rectiligne à gauche, c'est `stemRightBeamCount` qu'il faudra déterminer (exemple 2).

Pour que les barres de ligature débordent sur la droite, `stemRightBeamCount` doit avoir une valeur positive ; pour un débrodement à gauche, c'est sur `stemLeftBeamCount` qu'il faut jouer. Tout ceci est illustré par l'exemple 3.

Il est parfois judicieux, lorsqu'une note est encadrée de silences, de l'affubler de crochets rectilignes de part et d'autre. L'exemple 4 montre qu'il suffit d'adjoindre à cette note un `[]`.

(Notez bien que `\set stemLeftBeamCount` sera toujours synonyme de `\once \set`. Autrement dit, la détermination des ligatures n'est pas « permanente » ; c'est la raison pour laquelle les crochets du `c'16[]` isolé du dernier exemple n'ont rien à voir avec le `\set` indiqué deux notes auparavant.)

```
\score {
  <<
    % Example 1
    \new RhythmicStaff {
      \set stemLeftBeamCount = #0
      c16[]
      r8.
    }
    % Example 2
    \new RhythmicStaff {
      r8.
      \set stemRightBeamCount = #0
      c16[]
    }
    % Example 3
    \new RhythmicStaff {
      c16 c
      \set stemRightBeamCount = #2
      c16 r r
      \set stemLeftBeamCount = #2
      c16 c c
    }
    % Example 4
    \new RhythmicStaff {
      c16 c
      \set stemRightBeamCount = #2
      c16 r
      c16[]
      r16
      \set stemLeftBeamCount = #2
      c16 c
    }
  >>
}
```



Voir aussi

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 475, [Notes d'ornement], page 95.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Beam" dans *Référence des propriétés internes*, Section "BeamEvent" dans *Référence des propriétés internes*, Section "Beam_engraver" dans *Référence des propriétés internes*, Section "beam-interface" dans *Référence des propriétés internes*, Section "Stem_engraver" dans *Référence des propriétés internes*.

Liens de croches en soufflet
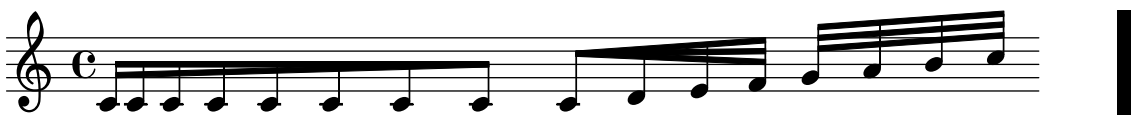
Les ligatures en soufflet permettent d'indiquer qu'un petit groupe de notes se joue en accélérant ou en ralentissant sans pour autant modifier le tempo du morceau. L'étendue du soufflet s'indique par `[` et `]`, et son orientation est déterminée par la propriété `grow-direction` de l'objet `Beam`.

Lorsque la sortie MIDI doit refléter les *ritardando* ou *accelerando* indiqués par une ligature en soufflet, les notes qui la composent doivent être regroupées dans une expression musicale délimitée par des accolades, précédée de la commande `featherDurations`. Cette commande détermine le ratio entre les durées des premières et dernières notes du groupe en question.

Les crochets indiquent l'étendue de la ligature et les accolades les notes concernées par une modification de leur durée. Il s'agit en général du même groupe de notes, mais les deux commandes sont indépendantes l'une de l'autre.

Dans l'exemple ci-après, les huit doubles croches occupent exactement le même espace qu'une blanche, mais la première est moitié moins longue que la dernière et celles qui les séparent s'allongent peu à peu. Les quatre triples croches qui suivent vont s'accélérant, alors que les quatre dernières gardent un tempo régulier.

```
\override Beam #'grow-direction = #LEFT
\featherDurations #(ly:make-moment 2 1)
{ c16[ c c c c c c c] }
\override Beam #'grow-direction = #RIGHT
\featherDurations #(ly:make-moment 2 3)
{ c32[ d e f] }
% revert to non-feathered beams
\override Beam #'grow-direction = #'()
{ g32[ a b c] }
```



Si le résultat imprimable ne reflète les durées que de manière approximative, la sortie MIDI sera quant à elle parfaitement « ponctuelle ».

Commandes prédéfinies

`\featherDurations`.

Voir aussi

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Problèmes connus et avertissements

La commande `\featherDurations` ne permet de traiter que de très courts extraits, avec une faible amplitude.
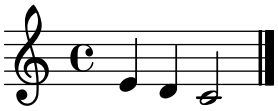
1.2.5 Mesures

Barres de mesure

Les barres de mesures délimitent les mesures, mais peuvent aussi indiquer une reprise. En principe, elles sont insérées automatiquement en respectant la métrique en vigueur.

Il est possible de forcer l'impression d'une barre de mesure spéciale, avec la commande `\bar` – c'est d'ailleurs l'habitude en fin de morceau, où l'on voit une double barre :

```
e4 d c2 \bar "|."
```



Rien ne s'oppose à ce que la dernière note d'une mesure ne s'arrête avant la barre de mesure ; on considère simplement qu'elle se prolonge sur la mesure suivante. Des débordements à répétition finissent par générer une musique comprimée ou qui sort de la page, pour la simple et bonne raison que les sauts de ligne automatiques ne peuvent intervenir qu'à la fin d'une mesure complète, autrement dit lorsque toutes les notes sont terminées avant la fin de la mesure.

> **Note :** Une durée erronée peut empêcher les sauts de ligne, ce qui conduit à une musique compressée, voire à un débordement de la page.

Il est possible d'autoriser un saut de ligne même s'il n'y a pas de barre de mesure visible, en utilisant :

```
\bar ""
```

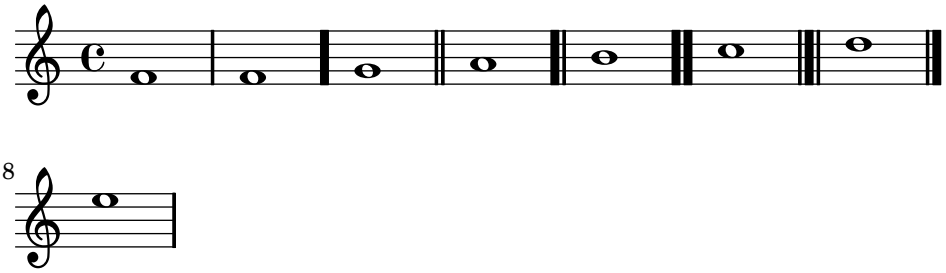
Ceci insérera une barre de mesure invisible, et permettra – sans pour autant le forcer – de sauter de ligne à cet endroit, sans incrémenter le numéro de mesure. Pour forcer le saut de ligne, référez vous à Section 4.3.1 [Sauts de ligne], page 424.

Cette barre invisible, ainsi que d'autres barres spéciales, peuvent être insérées manuellement n'importe où. Lorsqu'elles coïncident avec la fin d'une mesure, elles remplacent la simple barre que LilyPond aurait insérée automatiquement. Dans le cas contraire, la barre spécifiée s'insérera là où vous l'aurez positionnée.

Ces insertions n'affectent en rien le calcul du positionnement automatique des barres de mesure à suivre ni les propriétés y afférentes – numérotation, altérations accidentelles, sauts de ligne. . .
Lorsqu'une barre manuelle est insérée à l'endroit où viendrait se placer une barre normale, seul l'effet visuel en sera modifié.

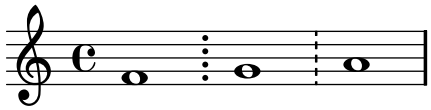
Vous disposez de deux types de barres simples et de cinq différentes doubles barres :

```
f1 \bar "|"
f1 \bar "."
g1 \bar "||"
a1 \bar ".|"
b1 \bar ".|."
c1 \bar "|.|"
d1 \bar "|."
e1
```



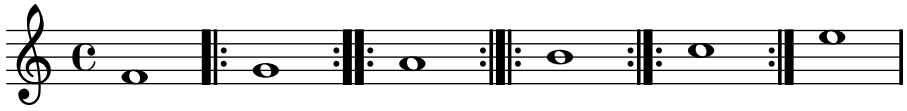
ainsi que d'une barre en pointillé et d'une discontinue :

```
f1 \bar ":"
g1 \bar "dashed"
a1
```



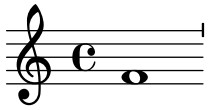
et de cinq types de barre de reprise :

```
f1 \bar "|:"
g1 \bar ":|:"
a1 \bar ":|.|:"
b1 \bar ":|.:"
c1 \bar ":|"
e1
```



De plus, une barre de mesure peut s'imprimer sous la forme d'une coche :

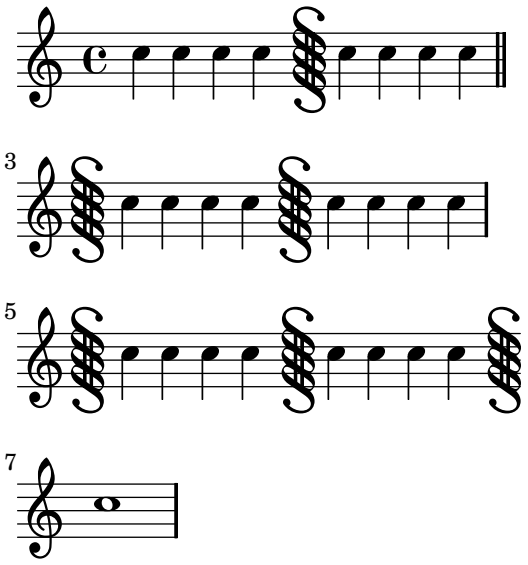
```
f1 \bar "'"
```



On rencontre habituellement ces signes dans le chant grégorien ; nous vous invitons dans ce cadre particulier à plutôt utiliser `\divisioMinima`, comme indiqué au paragraphe [Divisions], page 362 du chapitre consacré au grégorien.

L'insertion d'un *segno* directement sur la portée s'obtient à l'aide de trois types de barre de mesure, comme indiqué ci-dessous :

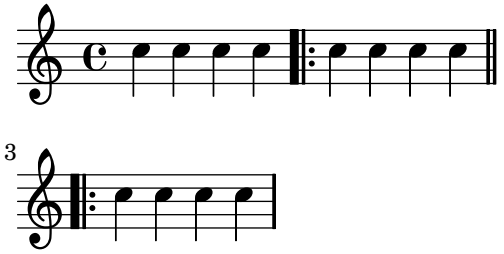
```
c4 c c c
\bar "S"
c4 c c c \break
\bar "S"
c4 c c c
\bar "|S"
c4 c c c \break
\bar "|S"
c4 c c c
\bar "S|"
c4 c c c \break
\bar "S|"
c1
```



Bien que l'on puisse insérer des barres de reprise manuellement, LilyPond n'en déduira pas pour autant qu'il s'agit d'un passage à répéter. Il est préférable d'indiquer les passages répétés à l'aide des différentes commandes de reprise (voir Section 1.4 [Répétitions et reprises], page 124) qui se chargeront d'imprimer le type de barre approprié.

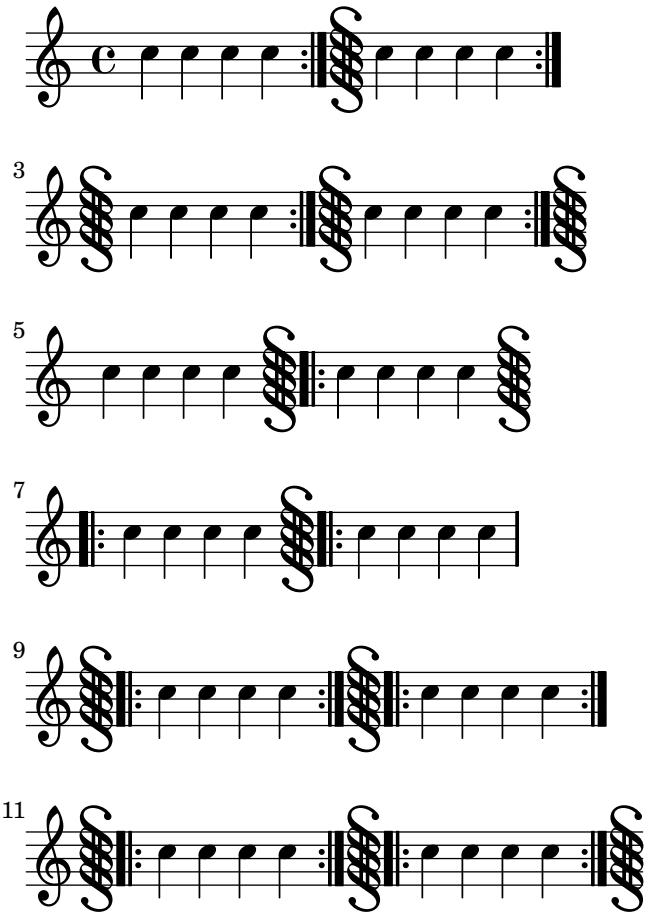
Dans les faits, un `"||:"` équivaut à un `"|:"` sauf s'il intervient à un saut de ligne : une double barre sera alors imprimée en fin de portée, et la barre de reprise au début de la nouvelle.
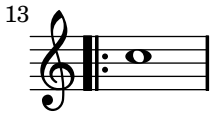
```
c4 c c c
\bar "||:"
c4 c c c \break
\bar "||:"
c4 c c c
```



LilyPond dispose de six différents moyens de combiner une barre de reprise avec un signe *segno* :
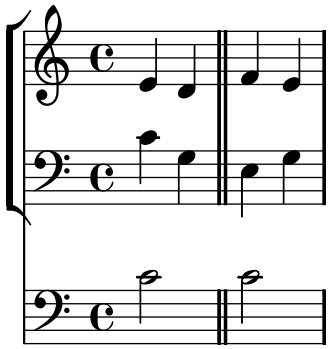
```
c4 c c c
\bar ":|S"
c4 c c c \break
\bar ":|S"
c4 c c c
\bar ":|S."
c4 c c c \break
\bar ":|S."
c4 c c c
\bar "S|:"
c4 c c c \break
\bar "S|:"
c4 c c c
\bar ".S|:"
c4 c c c \break
\bar ".S|:"
c4 c c c
\bar ":|S|:"
c4 c c c \break
\bar ":|S|:"
c4 c c c
\bar ":|S.|:"
c4 c c c \break
\bar ":|S.|:"
c1
```





Dans une partition comprenant plusieurs portées, la commande `\bar` placée sur une portée s'applique automatiquement à toutes les portées. Les barres de mesure que l'on obtient alors sont d'un seul tenant sur les portées d'un `StaffGroup`, d'un `PianoStaff` ou d'un `GrandStaff`.

```
<<
  \new StaffGroup <<
    \new Staff {
      e4 d
      \bar "||"
      f4 e
    }
    \new Staff { \clef bass c4 g e g }
  >>
  \new Staff { \clef bass c2 c2 }
>>
```



Morceaux choisis

La commande '`\bar` *`type_de_barre`*' sert de raccourci pour '`\set Timing.whichBar =` *`type_de_barre`*'. Dès que l'on définit `whichBar`, une barre de mesure est créée selon le style défini.

Le type de barre de mesure par défaut utilisé pour l'insertion automatique est `"|"`. Vous pouvez en changer à tout moment grâce à '`\set Timing.defaultBarType =` *`type_de_barre`*'.

Voir aussi

Manuel de notation : Section 4.3.1 [Sauts de ligne], page 424, Section 1.4 [Répétitions et reprises], page 124, [Regroupement de portées], page 155.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

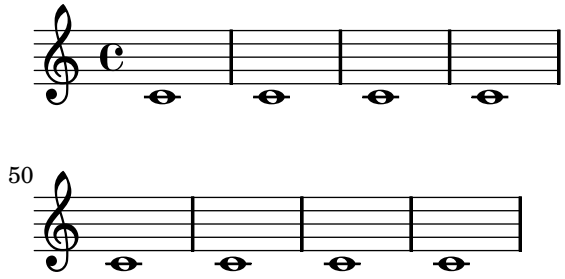
Référence des propriétés internes : Section "BarLine" dans *Référence des propriétés internes* (faisant partie du contexte `Staff`), Section "SpanBar" dans *Référence des propriétés internes* (sur plusieurs portées), Section "Timing_translator" dans *Référence des propriétés internes* (pour les propriétés liées au temps).

Numéros de mesure

Les numéros de mesure sont imprimés par défaut à chaque début de ligne, sauf la première. Ce nombre est stocké par la propriété `currentBarNumber` qui sera mise à jour à chaque mesure. Vous pouvez aussi le définir de manière arbitraire :

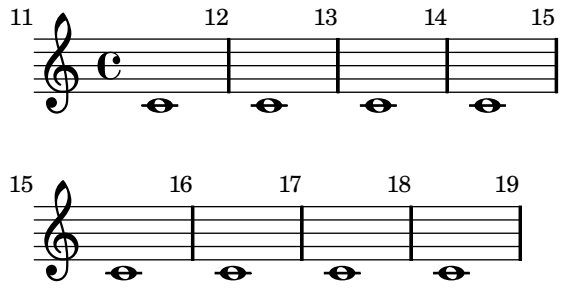
```
c1 c c c
\break
```

```
\set Score.currentBarNumber = #50
c1 c c c
```



Vous pouvez imprimer un numéro de mesure à intervalle régulier plutôt qu'en tête de chaque ligne. Pour y arriver, il faudra dans un premier temps annuler le comportement par défaut afin que les numéros puissent être imprimés ailleurs qu'en début de ligne. Tout ceci est contrôlé par la propriété `break-visibility` du `BarNumber`. Elle se compose de trois commutateurs – définis à « vrai » (`#t`) ou « faux » (`#f`) – pour spécifier si le numéro de mesure est visible ou non. Les valeurs sont rangées dans l'ordre suivant : `visible en fin de ligne`, `visible en cours de ligne`, `visible en début de ligne`. Voici comment imprimer partout les numéros de mesure :

```
\override Score.BarNumber #'break-visibility = #'#(#t #t #t)
\set Score.currentBarNumber = #11
% Permit first bar number to be printed
\bar ""
c1 | c | c | c
\break
c1 | c | c | c
```

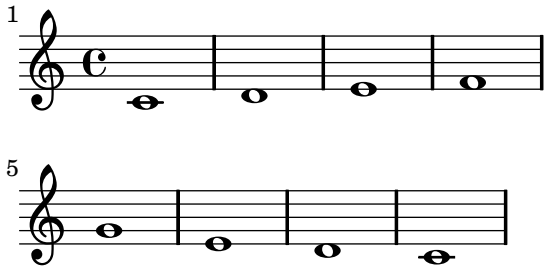


Morceaux choisis

Afficher le numéro de la première mesure

Par défaut, LilyPond n'affiche pas le premier numéro de mesure s'il est inférieur à 2. Le fait de définir `barNumberVisibility` à `all-bar-numbers-visible` vous permettra d'imprimer n'importe quel numéro pour la première mesure. À noter que l'impression d'un numéro de mesure ne peut intervenir que s'il y a une barre. Aussi, pour pouvoir le faire au début d'un morceau, devrez-vous ajouter une barre vide avant la première note.

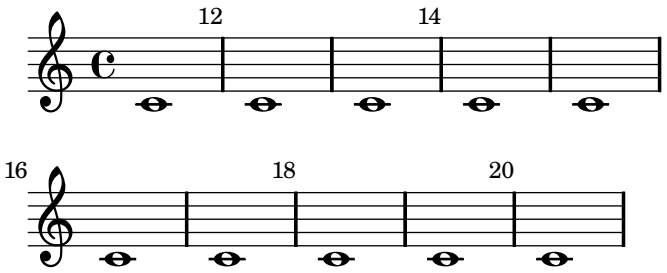
```
\relative c' {
  \set Score.barNumberVisibility = #all-bar-numbers-visible
  \bar ""
  c1 | d | e | f \break
  g1 | e | d | c
}
```



Imprimer les numéros de mesure à intervalle régulier

Vous pouvez imprimer un numéro de mesure à intervalle régulier plutôt qu'en tête de chaque ligne seulement, en recourrant à la propriété `barNumberVisibility`. Voici comment afficher le numéro toutes les deux mesures sauf en fin de ligne.

```
\relative c' {
  \override Score.BarNumber #'break-visibility = #end-of-line-invisible
  \set Score.currentBarNumber = #11
  % Permit first bar number to be printed
  \bar ""
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c | c | c | c
  \break
  c1 | c | c | c | c
}
```



Inscrire le numéro de mesure dans un cadre ou un cercle

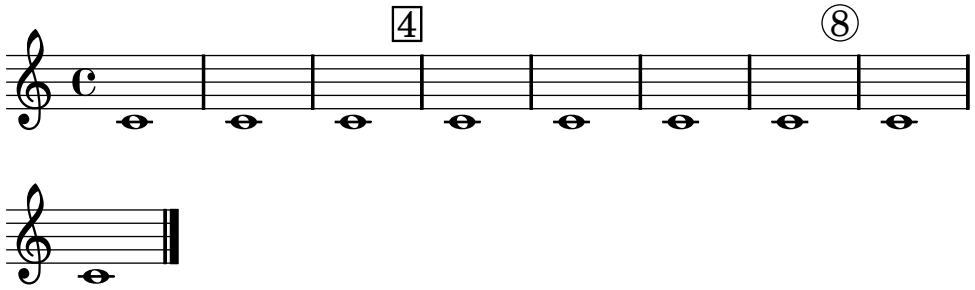
Les numéros de mesures peuvent être encadrés ou entourés d'un cercle.

```
\relative c' {
  % Prevent bar numbers at the end of a line and permit them elsewhere
  \override Score.BarNumber #'break-visibility = #end-of-line-invisible
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 4)

  % Increase the size of the bar number by 2
  \override Score.BarNumber #'font-size = #2

  % Draw a box round the following bar number(s)
  \override Score.BarNumber #'stencil
    = #(make-stencil-boxer 0.1 0.25 ly:text-interface::print)
  \repeat unfold 5 { c1 }

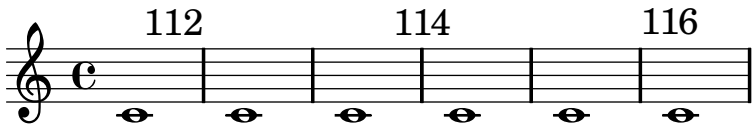
  % Draw a circle round the following bar number(s)
  \override Score.BarNumber #'stencil
    = #(make-stencil-circler 0.1 0.25 ly:text-interface::print)
  \repeat unfold 4 { c1 } \bar "|."
}
```



Alignement des numéros de mesure

Les numéros de mesure s'alignent en principe sur la droite de l'objet dont ils dépendent. C'est normalement le coin gauche de la portée ou, en cours de ligne, à gauche de la barre. Vous pouvez toutefois les centrer par rapport à la barre ou les afficher à droite de la barre.

```
\relative c' {
  \set Score.currentBarNumber = #111
  \override Score.BarNumber #'break-visibility = #all-visible
  % Increase the size of the bar number by 2
  \override Score.BarNumber #'font-size = #2
  % Print a bar number every second measure
  \set Score.barNumberVisibility = #(every-nth-bar-number-visible 2)
  c1 | c1
  % Center-align bar numbers
  \override Score.BarNumber #'self-alignment-X = #CENTER
  c1 | c1
  % Left-align bar numbers
  \override Score.BarNumber #'self-alignment-X = #LEFT
  c1 | c1
}
```
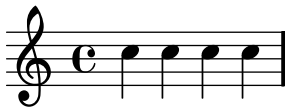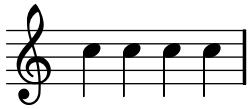


Supprimer les numéros de mesure d'une partition

Désactiver le graveur concerné — `Bar_number_engraver` — donnera une partition — contexte `Score` — sans numéros de mesure.

```
\layout {
  \context {
    \Score
    \remove "Bar_number_engraver"
  }
}

\relative c'' {
  c4 c c c \break
  c4 c c c
}
```





Voir aussi

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.

Référence des propriétés internes : Section “BarNumber” dans *Référence des propriétés internes*, Section “Bar_number_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les numéros de mesure peuvent entrer en collision avec les crochets d’un Section “StaffGroup” dans *Référence des propriétés internes*. La propriété `padding` – décalage – de l’objet Section “BarNumber” dans *Référence des propriétés internes* permet alors d’ajuster leur positionnement.

Vérification des limites et numéros de mesure

Les tests de limite de mesure (ou tests de mesure) aident à détecter les erreurs dans les durées. Un test de mesure s’écrit avec une barre verticale, `|`. Lors du traitement, elle doit correspondre à une barre de mesure. Sinon, un avertissement est émis qui indique le numéro de ligne où est détectée l’erreur. Dans l’exemple suivant, le deuxième test de mesure signale une erreur.

```
\time 3/4 c2 e4 | g2 |
```

Le test de mesure peut être aussi utilisé dans les paroles, par exemple :
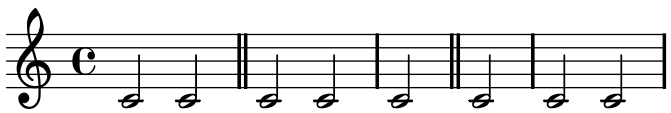
```
\lyricmode {
 \time 2/4
 Twin -- kle | Twin -- kle
}
```

Des durées incorrectes font échouer les tests de mesure et peuvent souvent mettre la partition sens dessus dessous, particulièrement s’il s’agit de musique polyphonique. Vérifier les tests de mesure qui ont échoué et les durées incorrectes est un bon moyen de commencer à corriger sa partition.

Lorsque plusieurs tests successifs présentent un même décalage, seul le message d’avertissement concernant la première occurrence est affiché. L’origine du problème est de fait plus évidente.

Il est aussi possible d’attribuer une autre valeur au symbole `|`, en assignant une expression musicale à `pipeSymbol`. Dans l’exemple suivant, le `|` servira à insérer une double barre là où il apparaît, au lieu de simplement vérifier que la fin de la mesure est atteinte.

```
pipeSymbol = \bar "||"
{
  c'2 c' |
  c'2 c'
  c'2 | c'
  c'2 c'
}
```



Lorsque l'on recopie de longues pièces, il peut être utile de vérifier que les numéros de mesure de LilyPond correspondent à l'original que l'on recopie. Cela se fait avec `\barNumberCheck`. Par exemple,

```
\barNumberCheck #123
```

affiche un avertissement lors du traitement si le numéro de mesure à ce point (variable `currentBarNumber`) n'est pas 123.

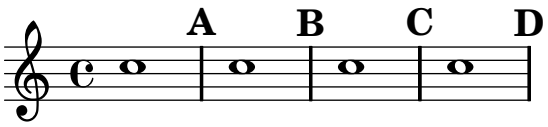
Voir aussi

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Indications de repère

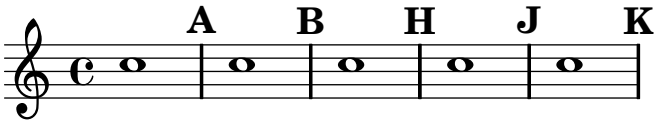
Indiquer un repère s'obtient grâce à la commande `\mark`.

```
c1 \mark \default
c1 \mark \default
c1 \mark \default
c1 \mark \default
```



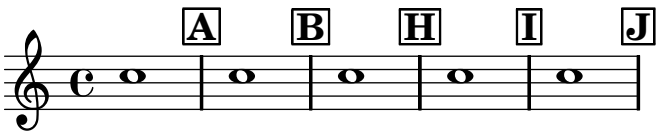
Lorsque vous utilisez `\mark \default`, le repère s'incrémente automatiquement ; toutefois donner un nombre en argument permet de spécifier manuellement le repère en question. La valeur à utiliser est enregistrée dans la propriété `rehearsalMark`.

```
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
```



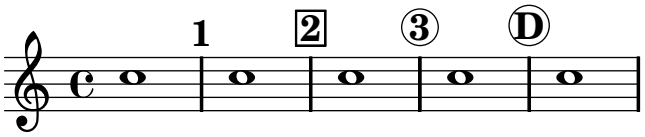
La lettre **I** n'est pas utilisée, conformément aux usages de la gravure. Cependant, vous pourrez intégrer la lettre **I** en utilisant l'une des commandes suivantes selon que ce repère doive être simple, inclus dans un rectangle ou dans un cercle :

```
\set Score.markFormatter = #format-mark-alphabet
\set Score.markFormatter = #format-mark-box-alphabet
\set Score.markFormatter = #format-mark-circle-alphabet
\set Score.markFormatter = #format-mark-box-alphabet
c1 \mark \default
c1 \mark \default
c1 \mark #8
c1 \mark \default
c1 \mark \default
```



Le style de repère est déterminé par la propriété `markFormatter`. Il s'agit d'une fonction qui prend en arguments le repère en cours (un entier) ainsi que le contexte en cours, et retournera un objet de type *markup*. Dans l'exemple qui suit, `markFormatter` est réglé pour une procédure type. Quelques mesures plus loin, son comportement est modifié pour imprimer un repère encadré.

```
\set Score.markFormatter = #format-mark-numbers
c1 \mark \default
c1 \mark \default
\set Score.markFormatter = #format-mark-box-numbers
c1 \mark \default
\set Score.markFormatter = #format-mark-circle-numbers
c1 \mark \default
\set Score.markFormatter = #format-mark-circle-letters
c1
```



Le fichier '`scm/translation-functions.scm`' comporte les définitions de `format-mark-numbers` (comportement par défaut), `format-mark-box-numbers`, `format-mark-letters` et `format-mark-box-letters`. Vous pouvez vous en inspirer pour d'autres fonctions de formatage.

`format-mark-barnumbers`, `format-mark-box-barnumbers` et `format-mark-circle-barnumbers` permettent d'imprimer le numéro de mesure au lieu des compteurs alphabétique ou numérique.

On peut aussi spécifier manuellement une marque de repère :

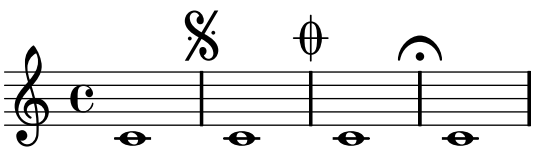
```
\mark "A1"
```

`Score.markFormatter` sera sans effet sur des repères ainsi définis. Un `\markup` peut néanmoins s'utiliser en argument.

```
\mark \markup{ \box A1 }
```

Un `\mark` peut contenir un glype musical tel que le signe *segno*.

```
c1 \mark \markup { \musicglyph #"scripts.segno" }
c1 \mark \markup { \musicglyph #"scripts.coda" }
c1 \mark \markup { \musicglyph #"scripts.ufermata" }
c1
```



Pour connaître les différents symboles accessibles par `\musicglyph`, consultez Section A.6 [La fonte Feta], page 491.

Pour affiner le positionnement des repères, veuillez vous référer à Section 1.8.2 [Mise en forme du texte], page 202, et tout particulièrement ce qui concerne la `break-alignable-interface` au chapitre Section 5.5.1 [Alignement des objets], page 476.

Voir aussi

Manuel de notation : Section A.6 [La fonte Feta], page 491, Section 1.8.2 [Mise en forme du texte], page 202, Section 5.5.1 [Alignement des objets], page 476.

Fichiers d'initialisation : '`scm/translation-functions.scm`' contient les définitions de `format-mark-numbers` et `format-mark-letters`. Elles seront source d'inspiration pour d'autres fonctions de formatage.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.
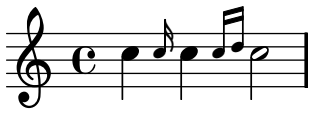
Référence des propriétés internes : Section "MarkEvent" dans *Référence des propriétés internes*, Section "Mark_engraver" dans *Référence des propriétés internes*, Section "RehearsalMark" dans *Référence des propriétés internes*.

1.2.6 Fonctionnalités rythmiques particulières

Notes d'ornement
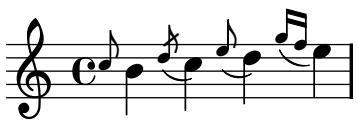
Les petites notes sont des ornements entièrement écrits. Leur taille est un peu plus petite que celle des notes normales et elles n'occupent pas de temps dans la mesure.

```
c4 \grace c16 c4
\grace { c16[ d16] } c2
```
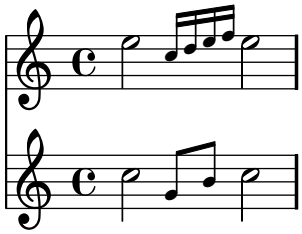


Les plus courantes sont les acciaccatures, qui doivent se jouer très vite, et qui s'écrivent sous forme d'une petite note barrée (sur la hampe) et liée. L'appoggiature est une petite note non barrée, qui vole une fraction à la durée de la note réelle qui la suit.

```
\grace c8 b4
\acciaccatura d8 c4
\appoggiatura e8 d4
\acciaccatura { g16[ f] } e4
```
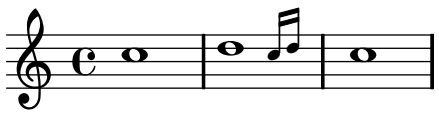


Les petites notes se placent de façon synchrone entre les différentes portées. Dans l'exemple suivant, il y a deux petites double-croches pour chaque petite croche.

```
<<
  \new Staff { e2 \grace { c16[ d e f] } e2 }
  \new Staff { c2 \grace { g8[ b] } c2 }
>>
```



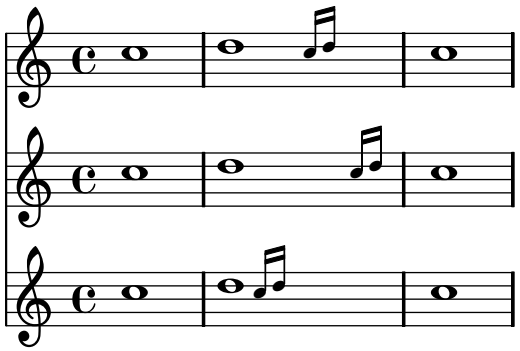
La commande `\afterGrace` sert à placer une petite note après une note réelle – et non *avant* comme d'ordinaire. Cette commande requiert deux arguments : la note réelle, et la ou les petites notes qui suivent.

```
c1 \afterGrace d1 { c16[ d] } c1
```
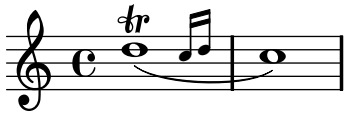


Les petites notes se placent alors aux 3/4 de la durée de la note réelle. Cette fraction peut être changée en définissant `afterGraceFraction`. Dans l'exemple suivant, vous pouvez observer la différence entre le comportement par défaut, à 15/16 et enfin à la moitié de la durée de base.

```
<<
  \new Staff {
    c1 \afterGrace d1 { c16[ d] } c1
  }
  \new Staff {
    #(define afterGraceFraction (cons 15 16))
    c1 \afterGrace d1 { c16[ d] } c1
  }
  \new Staff {
    #(define afterGraceFraction (cons 1 2))
    c1 \afterGrace d1 { c16[ d] } c1
  }
>>
```



Des silences invisibles permettent d'ajuster l'espace entre les petites notes et la note réelle. Nous pourrions positionner ces petites notes à sept huitièmes de la durée de la note de base :

```
\new Voice {
  <<
    { d1^\trill_( }
    { s2 s4. \grace { c16[ d] } }
  >>
  c1)
}
```



Les expressions `\grace` obéissent à des règles typographiques particulières, notamment pour régler l'orientation et la taille des objets. De ce fait, toute subtilité de mise en forme devra être indiquée **à l'intérieur** de l'expression introduite par `\grace` ; ces réglages additionnels doivent également être désactivés dans cette même expression.

```
\new Voice {
  \acciaccatura {
    \stemDown
    f16->
    \stemNeutral
  }
  g4 e c2
}
```
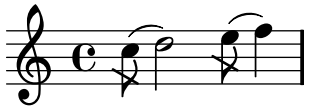


Morceaux choisis

Utilisation de hampe barrée pour une note normale

Le trait que l'on trouve sur les hampes des acciaccatures peut être appliqué dans d'autres situations.
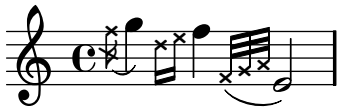
```
\relative c'' {
  \override Stem #'stroke-style = #"grace"
  c8( d2) e8( f4)
}
```



Mise en forme des notes d'ornement

Il est possible de changer globalement la mise en forme des petites notes dans un morceau, au moyen de la fonction `add-grace-property`. Ici, par exemple, on ôte la définition de l'orientation des objets `Stem` pour toutes les petites notes, afin que les hampes ne soient pas toujours orientées vers le haut, et on leur préfère des têtes en forme de croix.

```
\relative c'' {
  \new Staff {
    #(remove-grace-property 'Voice 'Stem 'direction)
    #(add-grace-property 'Voice 'NoteHead 'style 'cross)
    \new Voice {
       \acciaccatura { f16 } g4
       \grace { d16[ e] } f4
       \appoggiatura { f,32[ g a] } e2
    }
  }
}
```
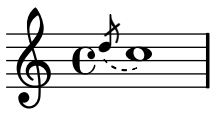


Redéfinition des réglages de mise en forme par défaut des notes d'ornement

Vous pouvez modifier les valeurs des variables `startGraceMusic`, `stopGraceMusic`, `startAcciaccaturaMusic`, `stopAcciaccaturaMusic`, `startAppoggiaturaMusic`, et `stopAppoggiaturaMusic` afin d'en personnaliser les effets. Pour plus de détails, voir le fichier '`ly/grace-init.ly`'.

```
startAcciaccaturaMusic = {
  s1*0(
  \override Stem #'stroke-style = #"grace"
  \slurDashed
}

stopAcciaccaturaMusic = {
  \revert Stem #'stroke-style
  \slurSolid
  s1*0)
}

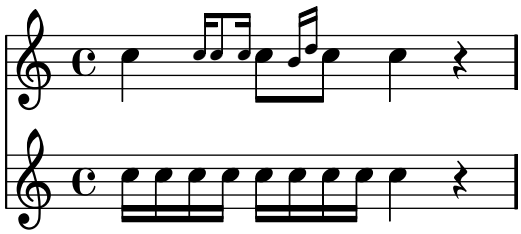
\relative c'' {
  \acciaccatura d8 c1
}
```



Positionnement des notes d'ornement avec espace flottant

Lorsque vous activez la propriété `'strict-grace-spacing`, l'espacement des notes d'ornement se fera de manière « élastique ». Autrement dit, elles seront décollées de leur note de rattachement : LilyPond commence par espacer les notes normales, puis les ornements sont placés à la gauche de leur note de rattachement.

```
\relative c'' {
  <<
    \override Score.SpacingSpanner #'strict-grace-spacing = ##t
    \new Staff \new Voice {
      \afterGrace c4 { c16[ c8 c16] }
      c8[ \grace { b16[ d] } c8]
      c4 r
    }
    \new Staff {
      c16 c c c c c c c c4 r
    }
  >>
}
```



Voir aussi

Glossaire musicologique : Section “ornements” dans *Glossaire*, Section “acciacature” dans *Glossaire*, Section “appogiature” dans *Glossaire*.

Fichiers d’initialisation : ‘`ly/grace-init.ly`’.

Morceaux choisis : Section “Rythme” dans *Morceaux choisis*.
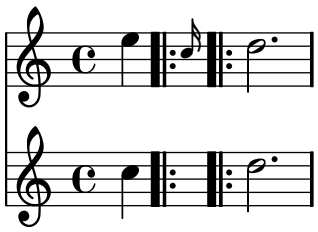
Référence des propriétés internes : Section “GraceMusic” dans *Référence des propriétés internes*, Section “Grace_beam_engraver” dans *Référence des propriétés internes*, Section “Grace_engraver” dans *Référence des propriétés internes*, Section “Grace_spacing_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Un groupe de notes ligaturées consituant une *acciaccatura* apparaîtra comme une *appoggiatura*, c’est-à-dire sans trait.
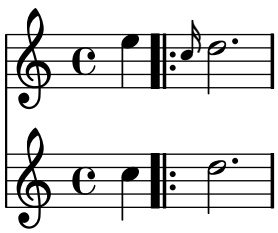
La synchronisation des petites notes se fait de façon parfois surprenante, car les autres objets de la portée – barre de mesure, armure, etc. – sont eux aussi synchrones. Pensez-y lorsque vous mêlez des portées comprenant des petites notes et d’autres sans :

```
<<
  \new Staff { e4 \bar "|:" \grace c16 d2. }
  \new Staff { c4 \bar "|:" d2. }
>>
```



Il est possible de remédier à cela en insérant, sur les autres portées, des silences invisibles dans une expression précédée de `\grace` et correspondant à la durée des petites notes :

```
<<
  \new Staff { e4 \bar "|:" \grace c16 d2. }
  \new Staff { c4 \bar "|:" \grace s16 d2. }
>>
```



Seules des expressions musicales séquentielles peuvent être utilisées pour des petites notes ; il n’est pas possible d’imbriquer ni de juxtaposer des sections de petites notes, faute de quoi le traitement du code peut échouer ou produire des erreurs.

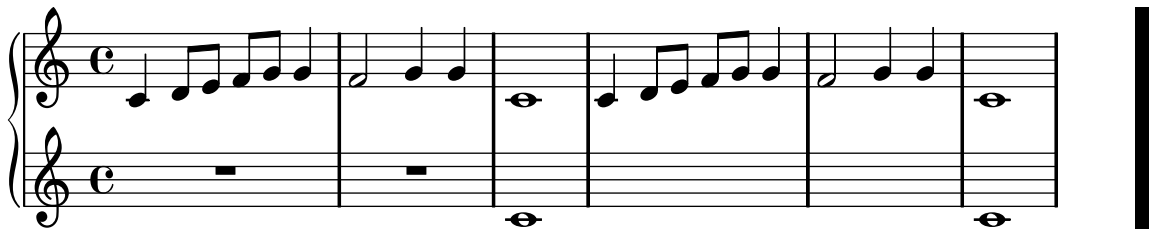
Alignement et cadences

Dans un contexte orchestral, une cadence constitue un problème spécifique. Lors du montage d'une partition contenant une cadence, tous les autres instruments doivent sauter autant de notes que ce qu'en comporte la cadence, faute de quoi ils démarreraient trop tôt ou trop tard.

Les fonctions `mmrest-of-length` ou `skip-of-length` permettent de résoudre ce problème. Ces fonctions Scheme prennent en argument un fragment de musique, et génèrent un `\skip` ou un silence multimesures d'une durée correspondant à ce fragment.

```
MyCadenza = \relative c' {
  c4 d8 e f g g4
  f2 g4 g
}

\new GrandStaff <<
  \new Staff {
    \MyCadenza c'1
    \MyCadenza c'1
  }
  \new Staff {
    #(ly:export (mmrest-of-length MyCadenza))
    c'1
    #(ly:export (skip-of-length MyCadenza))
    c'1
  }
>>
```



Voir aussi

Glossaire musicologique : Section "cadenza" dans *Glossaire*.

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Gestion du temps

Le temps est administré par le `Timing_translator`, qui réside en principe dans le contexte `Score`. Un alias, `Timing`, sera ajouté au contexte auquel le `Timing_translator` est rattaché. Déclarer explicitement un contexte `Voice` ou `Staff` assure l'existence de cet alias.

`Timing` dispose des propriétés suivantes afin de garder trace du minutage de la partition.

`currentBarNumber`
: Le numéro de la mesure en cours. Un exemple d'utilisation se trouve au chapitre [Numéros de mesure], page 88.
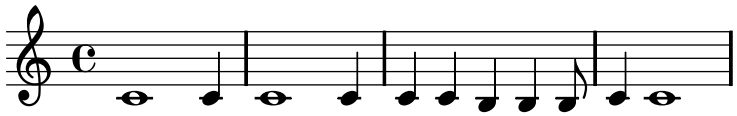
`measureLength`
: La longueur de la mesure, dans la métrique en cours. Pour une mesure à 4/4, elle est de 1, et de 3/4 pour une mesure à 6/8. Sa valeur détermine où peut s'insérer une barre et comment seront générées les ligatures automatiques.

`measurePosition`
: Le moment où l'on en est dans la mesure en cours. Cette quantité est remise à 0 dès lors qu'on dépasse `measureLength` ; la variable `currentBarNumber` est alors incrémentée.

`timing`
: Lorsqu'on lui assigne la valeur *vrai*, les valeurs ci-dessus mentionnées sont mises à jour à chaque pas. Fixée à *faux*, le graveur restera indéfiniment dans la mesure en cours.

Le calage peut être modifié en réglant explicitement l'une de ces variables. Dans l'exemple qui suit, nous réglons la métrique à 4/4, tout en fixant `measureLength` à 5/4. Arrivé à 4/8 dans la troisième mesure, nous avançons de 1/8, en assignant 5/8 à `measurePosition`, raccourcissant donc cette mesure d'une croche. La barre de mesure suivante tombera donc à 9/8 et non à 5/8.

```
\new Voice \relative c' {
  \set Timing.measureLength = #(ly:make-moment 5 4)
  c1 c4 |
  c1 c4 |
  c4 c
  \set Timing.measurePosition = #(ly:make-moment 5 8)
  b4 b b8 |
  c4 c1 |
}
```



Comme le montre cet exemple, `ly:make-moment n d` construit une durée de n/d fois une ronde. Par conséquent, `ly:make-moment 1 8` correspond à une croche, et `ly:make-moment 7 16` à la durée de sept doubles croches.
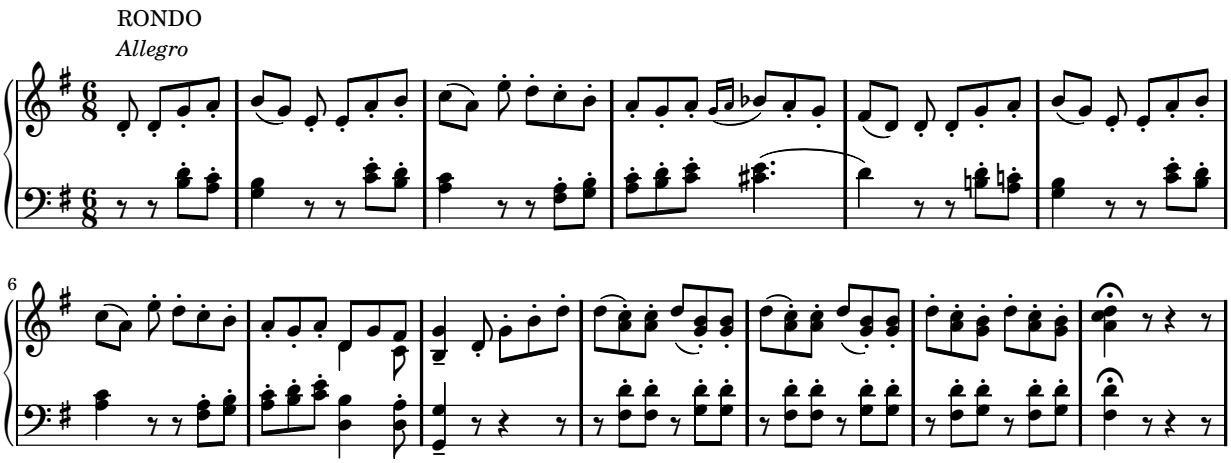
Voir aussi

Manuel de notation : [Numéros de mesure], page 88, [Musique sans métrique], page 63

Morceaux choisis : Section "Rythme" dans *Morceaux choisis*.

Références des propriétés internes : Section "Timing_translator" dans *Référence des propriétés internes*, Section "Score" dans *Référence des propriétés internes*

1.3 Signes d'interprétation



Ce chapitre traite des différentes indications d'interprétation que l'on peut trouver sur les partitions.

1.3.1 Signes d'interprétation attachés à des notes

Nous allons voir au cours de ces lignes comment ajouter aux notes des indications d'interprétation – articulation, ornementation, nuance – et aborderons la manière de générer vos propres signes.
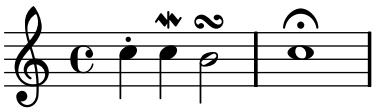
Articulations et ornements

Les différents symboles qui indiquent des ponctuations ou des modes de jeu différents s'ajoutent aux notes de la manière suivante :

```
note\nom
```
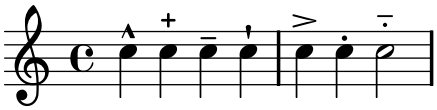
Les valeurs de *nom* sont répertoriées dans l'annexe Section A.10 [Liste des signes d'articulation], page 553. En voici un exemple :

```
c4\staccato c\mordent b2\turn
c1\fermata
```



Certains signes d'articulation disposent d'un raccourci. On les ajoute à chaque note au moyen d'un tiret suivi du caractère correspondant à l'articulation désirée. C'est entre autres le cas pour *marcato*, *stopped*, *tenuto*, *staccatissimo*, *accent*, *staccato*, et *portato*, comme l'illustre l'exemple ci-dessous.

```
c4-^  c-+  c--  c-|
c4->  c-.  c2-_
```
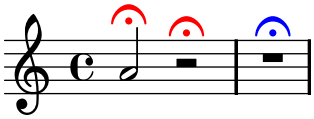


Même si LilyPond place automatiquement ces symboles, selon les règles contenues dans le fichier '`scm/script.scm`', il est possible de l'obliger à les positionner au-dessus ou en-dessous de la note, comme indiqué au chapitre Section 5.4.2 [Direction et positionnement], page 475.

Les articulations sont des objets de type `script` ; les propriétés de ces objets sont abordées plus en détail au chapitre Section "Script" dans *Référence des propriétés internes*.

Une articulation peut se rattacher aussi bien à un silence qu'à une note, mais **pas à un silence multi-mesures**. Il existe cependant un cas particulier : le point d'orgue – ou point d'arrêt – attaché à un silence valant l'intégralité de la mesure. LilyPond dispose à cet effet de la commande `\fermataMarkup`, qui créera un objet `MultiMeasureRestText` rattaché à ce « silence multi-mesures ».

```
\override Script #'color = #red
\override MultiMeasureRestText #'color = #blue
a2\fermata r\fermata
R1\fermataMarkup
```



En dehors des articulations habituelles, vous pouvez adjoindre du texte – avec ou sans mise en forme – à n'importe quelle note. Voir à ce propos [Commentaires textuels], page 194.

Pour plus d'information sur la manière d'ordonner `Scripts` et `TextScripts`, consultez le chapitre Section "Positionnement des objets" dans *Manuel d'initiation*.

Morceaux choisis
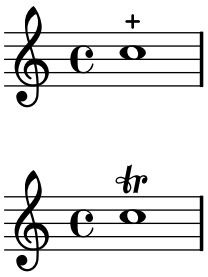
Modification de la signification des raccourcis pour les signes d'articulation

Les raccourcis sont répertoriés dans le fichier '`ly/script-init.ly`', dans lequel on retrouve les variables `dashHat`, `dashPlus`, `dashDash`, `dashBar`, `dashLarger`, `dashDot`, et `dashUnderscore` ainsi que leur valeur par défaut. Ces valeurs peuvent être modifiées selon vos besoins. Il suffit par exemple, pour affecter au raccourci `-+` (`dashPlus`) le symbole du trille en lieu et place du `+` (caractère plus), d'assigner la valeur `trill` à la variable `dashPlus` :

```
\relative c'' { c1-+ }

dashPlus = "trill"

\relative c'' { c1-+ }
```
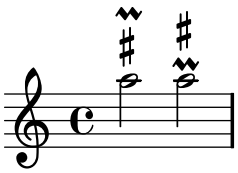


Contrôle de l'ordre vertical des articulations et ornements

Les symboles s'ordonnent verticalement suivant la propriété `script-priority`. Plus sa valeur numérique est faible, plus le symbole sera proche de la note. Dans l'exemple suivant, l'objet `TextScript` – le dièse – a d'abord la propriété la plus basse, et il est donc placé plus près de la note ; ensuite c'est l'objet `Script` – le mordant – qui a la propriété la plus basse, et il se place alors sous le dièse. Lorsque deux objets ont la même priorité, c'est l'ordre dans lequel ils sont indiqués qui détermine lequel sera placé en premier.

```
\relative c''' {
  \once \override TextScript #'script-priority = #-100
  a2^\prall^\markup { \sharp }
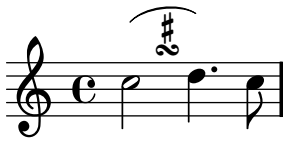
  \once \override Script #'script-priority = #-100
  a2^\prall^\markup { \sharp }
}
```



Création d'un grupetto retardé

Obtenir un *grupetto* retardé et dans lequel la note la plus basse est altérée requiert quelques surcharges. La propriété `outside-staff-priority` doît être désactivée (`#f`) pour éviter qu'elle prenne le pas sur la propriété `avoid-slur`. La valeur assignée à `halign` permet de gérer horizontalement le « retard ».

```
\relative c'' {
  \once \override TextScript #'avoid-slur = #'inside
  \once \override TextScript #'outside-staff-priority = ##f
  c2(^\markup \tiny \override #'(baseline-skip . 1) {
    \halign #-4
    \center-column {
      \sharp
      \musicglyph #"scripts.turn"
    }
  }
  d4.) c8
}
```



Voir aussi

Glossaire musicologique : Section "tenuto" dans *Glossaire*, Section "accent" dans *Glossaire*, Section "staccato" dans *Glossaire*, Section "portato" dans *Glossaire*.

Manuel d'initiation : Section "Positionnement des objets" dans *Manuel d'initiation*.

Manuel de notation : [Commentaires textuels], page 194, Section 5.4.2 [Direction et positionnement], page 475, Section A.10 [Liste des signes d'articulation], page 553, [Trilles], page 122.

Fichiers installés : '`scm/script.scm`'.

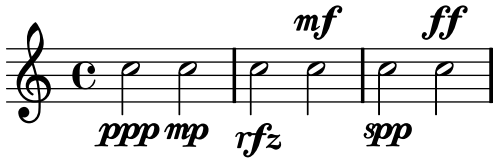
Morceaux choisis : Section "Signes d'interprétation" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Script" dans *Référence des propriétés internes*, Section "TextScript" dans *Référence des propriétés internes*.

Nuances

À chaque nuance absolue correspond une commande qui peut être indiquée après une note : `c4\ff` par exemple. Les commandes de nuance disponibles sont `\ppppp`, `\pppp`, `\ppp`, `\pp`, `\p`, `\mp`, `\mf`, `\f`, `\ff`, `\fff`, `\ffff`, `\fffff`, `\fp`, `\sf`, `\sff`, `\sp`, `\spp`, `\sfz`, et `\rfz`. Les nuances se placent aussi bien en dessous qu'au-dessus de la portée ; pour plus d'information, consultez Section 5.4.2 [Direction et positionnement], page 475.

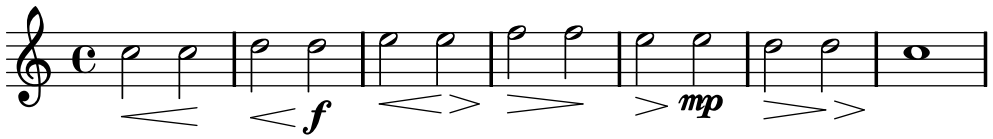
```
c2\ppp c\mp
c2\rfz c^\mf
c2_\spp c^\ff
```



Un crescendo est délimité par `\<` et `\!`, ou peut se terminer par une commande de nuance explicite, ou bien un decrescendo ou un nouveau crescendo. Il en va de même pour un diminuendo.

Au lieu de `\<` et `\>`, vous pouvez utiliser `\cr` et `\decr`, auquel cas LilyPond n'imprimera pas de soufflet (*hairpin* en anglais).

```
c2\< c\!
d2\< d\f
e2\< e\>
f2\> f\!
e2\> e\mp
d2\> d\>
c1\!
```
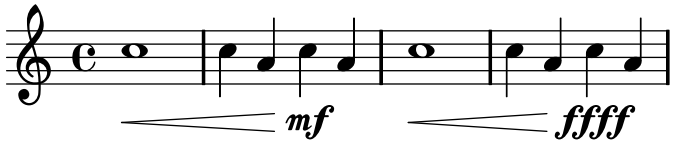


Un soufflet terminé par un simple `\!` prendra fin sur la droite de la tête de note à laquelle il est attaché. Dans le cas où il se termine par l'intervention d'un autre soufflet (contraire ou non), il prendra fin au milieu de la tête de note affublée d'un `\<` ou d'un `\>`, et le nouveau soufflet débutera à l'extrémité droite de cette même tête de note.

```
c1\< | c4 a c\< a | c4 a c\! a\< | c4 a c a\!
```



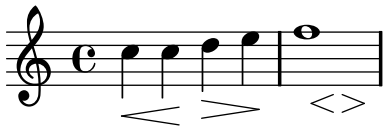
Il en va de même lorsqu'un soufflet est interrompu par une nuance explicite. Notez bien que la largeur occupée par cette nuance explicite influe sur la terminaison du soufflet :

```
c1\< | c4 a c\mf a | c1\< | c4 a c\ffff a
```
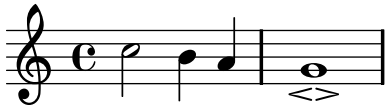


Les indications de nuance sont attachées aux notes ; aussi, lorsque l'on veut faire se succéder plusieurs nuances pendant une note tenue, il faudra avoir recours à des silences invisibles :

```
c4\< c\! d\> e\!
<< f1 { s4 s4\< s4\> s4\! } >>
```
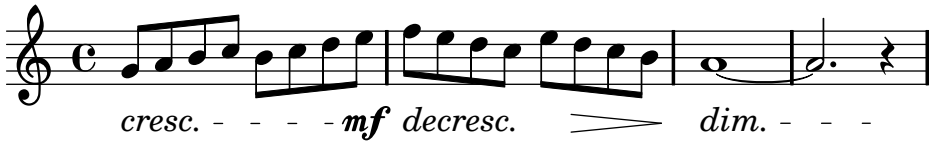


On peut avoir recours à l'indication `\espressivo` pour indiquer un crescendo suivi d'un decrescendo sur une même note. Gardez à l'esprit qu'il s'agit d'une articulation, et en aucun cas d'une nuance.

```
c2 b4 a
g1\espressivo
```
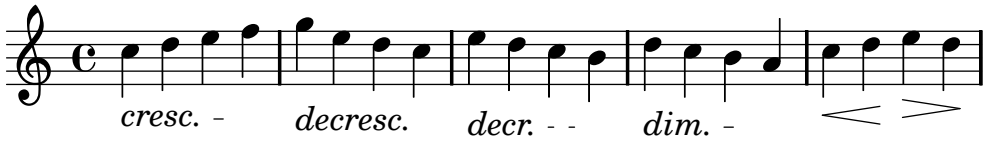


La commande `\cresc` permet d'indiquer textuellement le début d'un crescendo. `\decresc` ou `\dim` marquent le début d'un decrescendo. Les lignes d'extension sont gérées automatiquement.

```
g8\cresc a b c b c d e\mf |
f8\decresc e d c e\> d c b |
a1\dim ~ |
a2. r4\! |
```



Une indication textuelle peut indiquer, au lieu d'un soufflet, un changement de nuance :

```
\crescTextCresc
c4\< d e f\! |
\dimTextDecresc
g4\> e d c\! |
\dimTextDecr
e4\> d c b\! |
\dimTextDim
d4\> c b a\! |
\crescHairpin
\dimHairpin
c4\< d\! e\> d\! |
```



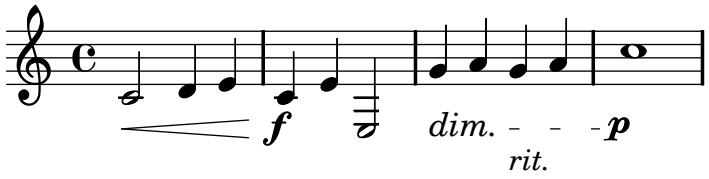
Pour créer des indications de nuance qui restent alignées avec les nuances habituelles, reportez-vous au chapitre [Personnalisation des indications de nuance], page 109.

Le positionnement vertical des nuances est géré par le Section "DynamicLineSpanner" dans *Référence des propriétés internes*.

L'utilisation d'un contexte `Dynamics` permet de graver les nuances sur leur propre ligne – Il suffit de placer des silences invisibles pour gérer le temps. Bien que le contexte `Dynamics` accepte des notes pour indiquer les durées, celles-ci ne seront pas imprimées. Le contexte `Dynamics` peut aussi contenir des indications textuelles avec ou sans extenseur, ainsi que les indications de pédale.

```
<<
  \new Staff \relative c' {
    c2 d4 e |
    c4 e e,2 |
    g'4 a g a |
    c1 |
  }
  \new Dynamics {
    s1\< |
    s1\f |
    s2\dim s2-"rit." |
    s1\p |
```

```
  }
>>
```



Commandes prédéfinies

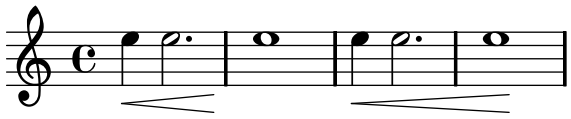
`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`, `\crescTextCresc`, `\dimTextDim`, `\dimTextDecr`, `\dimTextDecresc`, `\crescHairpin`, `\dimHairpin`.

Morceaux choisis

Soufflets et barres de mesure

En principe, un soufflet – (de)crescendo imprimé sous forme graphique – commence au bord gauche de la note de départ, et se termine au bord droit de la note d'arrivée. Cependant, si la note d'arrivée est sur un premier temps, le soufflet s'arrêtera au niveau de la barre de mesure qui la précède. Ce comportement peut être annulé en assignant *faux* (`#f`) à la propriété `'to-barline` :
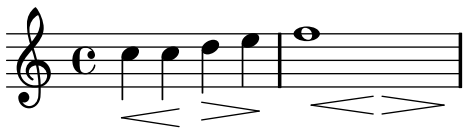
```
\relative c'' {
  e4\< e2.
  e1\!
  \override Hairpin #'to-barline = ##f
  e4\< e2.
  e1\!
}
```



Ajustement de la longueur d'un soufflet

Si un soufflet est trop court, il suffit d'ajuster la propriété `minimum-length` de l'objet `Hairpin` pour l'allonger.

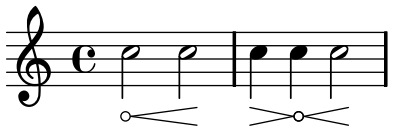
```
\relative c'' {
  c4\< c\! d\> e\!
  \override Hairpin #'minimum-length = #5
  << f1 { s4 s\< s\> s\! } >>
}
```



Impression de soufflets « al niente »

Des crescendos ou decrescendos *al niente* peuvent être indiqués de manière graphique, en assignant *vrai* (`#t`) à la propriété `circled-tip`, ce qui affiche un cercle à leur extrémité.

```
\relative c'' {
  \override Hairpin #'circled-tip = ##t
  c2\< c\!
  c4\> c\< c2\!
}
```



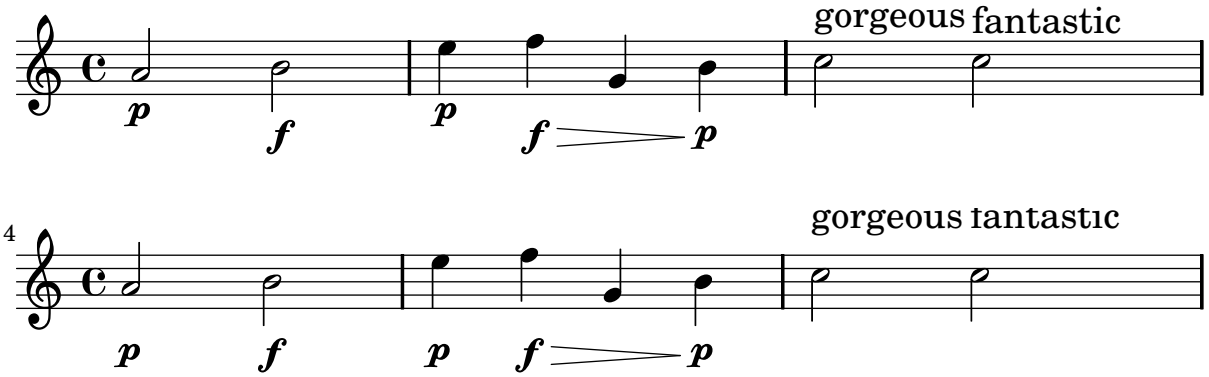
Alignement vertical des nuances indications textuelles

Vous pourrez, en jouant sur la propriété `'Y-extent`, aligner les différents objets `DynamicLineSpanner` (souflets ou textuels) quelle que soit leur étendue, par rapport à un même point de référence. Tous les éléments seront alors centrés sur une même ligne, ce qui sera visuellement plus agréable.

C'est le même principe qui sert à aligner les indications textuelles sur une ligne de référence.

```
music = \relative c' {
  a'2\p b\f
  e4\p f\f\> g, b\p
  c2^\markup { \huge gorgeous } c^\markup { \huge fantastic }
}

{
  \music
  \break
  \override DynamicLineSpanner #'staff-padding = #2.0
  \override DynamicLineSpanner #'Y-extent = #'(-1.5 . 1.5)
  \override TextScript #'Y-extent = #'(-1.5 . 1.5)
  \music
}
```

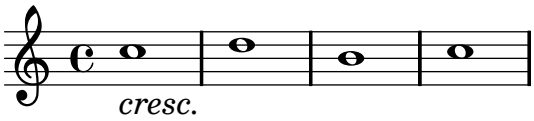


Masquage de l'extension des nuances textuelles

Les crescendos et decrescendos indiqués textuellement – tels que *cresc.* ou *dim.* – sont suivis de pointillés qui montrent leur étendue. On peut empêcher l'impression de ces pointillés avec :

```
\relative c'' {
  \override DynamicTextSpanner #'style = #'none
```
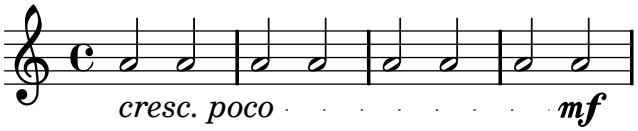
```
  \crescTextCresc
  c1\< | d | b | c\!
}
```



Modifidation du texte et de l'extension de nuances textuelles

Le texte par défaut des crescendos et decrescendos se change en modifiant les propriétés de contexte `crescendoText` et `decrescendoText`. L'aspect de la ligne d'extension est fonction de la propriété `'style` du `DynamicTextSpanner`. Sa valeur par défaut est `'hairpin`, mais d'autres valeurs sont disponibles, comme `'line`, `'dashed-line` et `'dotted-line`.

```
\relative c'' {
  \set crescendoText = \markup { \italic { cresc. poco } }
  \set crescendoSpanner = #'text
  \override DynamicTextSpanner #'style = #'dotted-line
  a2\< a
  a2 a
  a2 a
  a2 a\mf
}
```



Voir aussi

Glossaire musicologique : Section "al niente" dans *Glossaire*, Section "crescendo" dans *Glossaire*, Section "decrescendo" dans *Glossaire*, Section "soufflet" dans *Glossaire*.

Manuel d'initiation : Section "Articulations et nuances" dans *Manuel d'initiation*.

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 475, [Personnalisation des indications de nuance], page 109, Section 3.5.3 [Contenu de la sortie MIDI], page 406, Section 3.5.5 [Gestion des nuances en MIDI], page 407.

Morceaux choisis : Section "Signes d'interprétation" dans *Morceaux choisis*.

Référence des propriétés internes : Section "DynamicText" dans *Référence des propriétés internes*, Section "Hairpin" dans *Référence des propriétés internes*, Section "DynamicLineSpanner" dans *Référence des propriétés internes*, Section "Dynamics" dans *Référence des propriétés internes*.
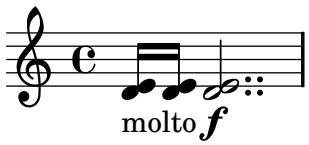
Personnalisation des indications de nuance

La manière la plus simple de personnaliser une indication de nuance consiste à utiliser un objet `\markup`.

```
moltoF = \markup { molto \dynamic f }

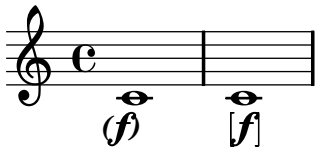
\relative c' {
  <d e>16_\moltoF <d e>
  <d e>2..
```
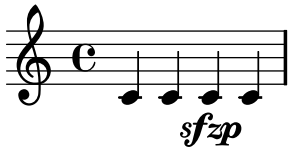
```
}
```



Vous pouvez créer des indications de nuance éditoriales (entre parenthèses ou crochets) grâce aux étiquettes (*mode markup*) ; la syntaxe en est abordée au chapitre Section 1.8.2 [Mise en forme du texte], page 202.

```
roundF = \markup {
    \center-align \concat { \bold { \italic ( }
          \dynamic f \bold { \italic ) } } }
boxF = \markup { \bracket { \dynamic f } }
\relative c' {
  c1_\roundF
  c1_\boxF
}
```



Grâce à la fonction `make-dynamic-script`, vous pouvez créer de nouvelles marques textuelles que vous combinerez avec les signes de nuance.

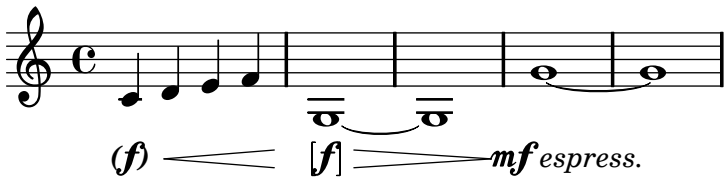
```
sfzp = #(make-dynamic-script "sfzp")
\relative c' {
  c4 c c\sfzp c
}
```



`make-dynamic-script` accepte en argument tout objet de type *markup*. Notez bien que la police des nuances ne contient que les caractères `f,m,p,r,s` et `z`, et que les marques de nuance possèdent des propriétés particulières et prédéfinies quant à leur police. Aussi, lorsque vous créez du texte en pareille situation, nous vous recommandons d'utiliser `\normal-text` pour annuler ces propriétés. L'intérêt majeur de recourrir à la fonction `make-dynamic-script` plutôt qu'à un simple *markup* réside dans l'assurance que ces objets personnalisés et les soufflets seront alignés lorsqu'attachés à une même note.

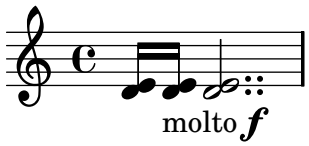
```
roundF = \markup { \center-align \concat {
           \normal-text { \bold { \italic ( } }
           \dynamic f
           \normal-text { \bold { \italic ) } } } }
boxF = \markup { \bracket { \dynamic f } }
mfEspress = \markup { \center-align \line {
              \hspace #3.7 mf \normal-text \italic espress. } }
roundFdynamic = #(make-dynamic-script roundF)
```

```
boxFdynamic = #(make-dynamic-script boxF)
mfEspressDynamic = #(make-dynamic-script mfEspress)
\relative c' {
  c4_\roundFdynamic\< d e f
  g,1~_\boxFdynamic\>
  g
  g'~\mfEspressDynamic
  g
}
```



La constrution d'une indication de nuance personnalisée peut aussi se faire en langage Scheme ; voir Section "Construction Scheme d'un marqueur" dans *Extension de LilyPond* pour en connaître les modalités.

```
moltoF = #(make-dynamic-script
            (markup #:normal-text "molto"
                    #:dynamic "f"))
\relative c' {
  <d e>16 <d e>
  <d e>2..\moltoF
}
```



L'utilisation des fontes en mode *markup* est abordée au chapitre [Sélection de la fonte et de la taille], page 204.

Voir aussi

Manuel de notation : Section 1.8.2 [Mise en forme du texte], page 202, [Sélection de la fonte et de la taille], page 204, Section 3.5.3 [Contenu de la sortie MIDI], page 406, Section 3.5.5 [Gestion des nuances en MIDI], page 407.

Morceaux choisis : Section "Signes d'interprétation" dans *Morceaux choisis*.

Manuel d'extension : Section "Construction Scheme d'un marqueur" dans *Extension de LilyPond*.

1.3.2 Signes d'interprétation sous forme de courbe

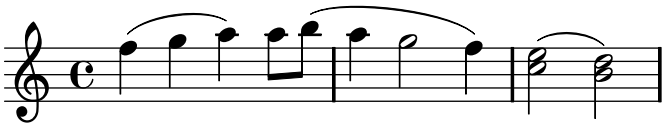
Ce chapitre traite des signes d'interprétation imprimés sous forme de courbe : liaisons d'articulation ou de phrasé, respirations, chutes et sauts.

Liaisons d'articulation

Une liaison d'articulation indique que les notes doivent être jouées liées, ou *legato*. Ces liaisons s'indiquent au moyen de parenthèses.

> **Note :** Lorsque la musique est polyphonique, la liaison doît se terminer dans la voix où elle a été entamée.

```
f4( g a) a8 b(
a4 g2 f4)
<c e>2( <b d>2)
```
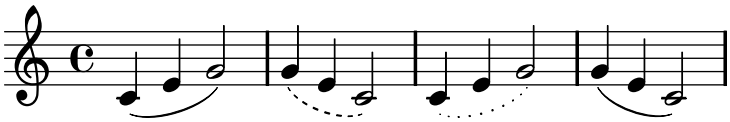


Vous pouvez décider de l'orientation des liaisons par rapport à la portée, comme indiqué au chapitre Section 5.4.2 [Direction et positionnement], page 475.

Une seule liaison d'articulation peut être imprimée à la fois. S'il est nécessaire d'imprimer une liaison plus longue, englobant des liaisons plus courtes, utilisez des [Liaisons de phrasé], page 114.
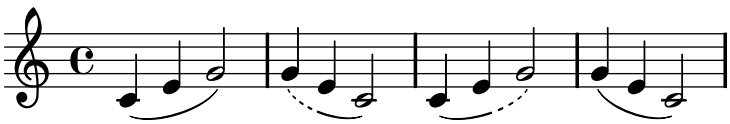
Une liaison est par défaut dessinée d'un trait plein. Il est aussi possible de l'imprimer sous la forme de tirets ou en pointillé :

```
c4( e g2)
\slurDashed
g4( e c2)
\slurDotted
c4( e g2)
\slurSolid
g4( e c2)
```



En utilisant `\slurHalfDashed`, la première moitié de la liaison aura un trait discontinu et continu pour la seconde. L'inverse s'obtient avec `\slurHalfSolid`.
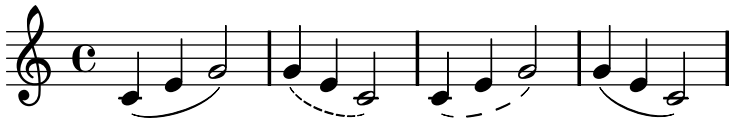
```
c4( e g2)
\slurHalfDashed
g4( e c2)
\slurHalfSolid
c4( e g2)
\slurSolid
g4( e c2)
```



Vous pouvez même personnaliser la densité des tirets d'une liaison :

```
c4( e g2)
\slurDashPattern #0.7 #0.75
g4( e c2)
\slurDashPattern #0.5 #2.0
c4( e g2)
\slurSolid
```

```
g4( e c2)
```



Commandes prédéfinies

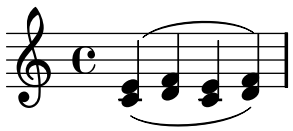
`\slurUp`, `\slurDown`, `\slurNeutral`, `\slurDashed`, `\slurDotted`, `\slurHalfDashed`, `\slurHalfSolid`, `\slurDashPattern`, `\slurSolid`.

Morceaux choisis

Accords et double liaison d'articulation

Certains auteurs utilisent deux liaisons lorsqu'ils veulent lier des accords. Dans LilyPond, il faut pour cela activer la propriété `doubleSlurs` :
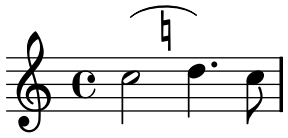
```
\relative c' {
  \set doubleSlurs = ##t
  <c e>4( <d f> <c e> <d f>)
}
```



Positionnement d'une annotation à l'intérieur d'une liaison

Lorsqu'il vous faut inscrire une annotation à l'intérieur d'une liaison, la propriété `outside-staff-priority` doît être désactivée.

```
\relative c'' {
  \override TextScript #'avoid-slur = #'inside
  \override TextScript #'outside-staff-priority = ##f
  c2(^\markup { \halign #-10 \natural } d4.) c8
}
```
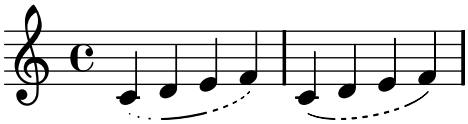


Dessin d'une liaison d'articulation au trait discontinu

Grâce à la propriété `dash-definition`, une liaison d'articulation peut être formée de traits discontinus variables. `dash-definition` se compose d'une liste de `segments-discontinus` (*dash-elements*). Chaque `segment-discontinu` contient une liste de paramètres qui déterminent le comportement du trait pour une section de la liaison.

Cette liaison se définit selon le paramètre de Bézier `t` qui est compris entre 0 (l'extrémité gauche de la liaison) et 1 (l'extrémité droite de la liaison). Chaque `segment-discontinu` se composera selon la liste `(t-début t-fin segment-style segment-taille)`. La portion de liaison allant de `t-début` à `t-fin` aura un trait `segment-style` de longueur `segment-taille`. `segment-taille` est exprimé en espace de portée ; un `segment-style` à 1 donnera un trait plein.

```
\relative c' {
  \once \override
    Slur #'dash-definition = #'((0 0.3 0.1 0.75)
                                (0.3 0.6 1 1)
                                (0.65 1.0 0.4 0.75))
  c4( d e f)
  \once \override
    Slur #'dash-definition = #'((0 0.25 1 1)
                                (0.3 0.7 0.4 0.75)
                                (0.75 1.0 1 1))
  c4( d e f)
}
```



Voir aussi

Glossaire musicologique : Section “liaison” dans *Glossaire*.

Manuel d’initiation : Section “Non-imbrication des crochets et liaisons” dans *Manuel d’initiation*.

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 475, [Liaisons de phrasé], page 114.
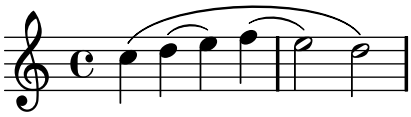
Morceaux choisis : Section “Signes d’interprétation” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Slur” dans *Référence des propriétés internes*.

Liaisons de phrasé

Une liaison de phrasé relie plusieurs notes en délimitant une phrase musicale. On indique les points de départ et d’arrivée avec `\(` et `\)` respectivement.

```
c4\( d( e) f(
e2) d\)
```
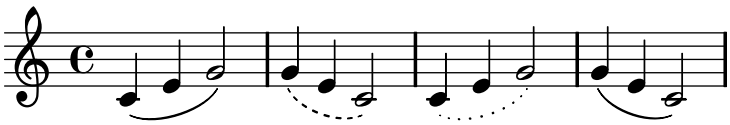


D’un point de vue typographique, rien ne distingue une liaison de phrasé d’une liaison d’articulation. Cependant, LilyPond les considère comme des objets différents. Une commande `\slurUp` n’affectera donc pas une liaison de phrasé. Vous pouvez décider de l’orientation des liaisons de phrasé par rapport à la portée, comme indiqué au chapitre Section 5.4.2 [Direction et positionnement], page 475.

Il n’est pas possible d’avoir plusieurs liaisons de phrasé en même temps.

Une liaison est par défaut dessinée d’un trait plein. Il est aussi possible de l’imprimer sous la forme de tirets ou en pointillé :
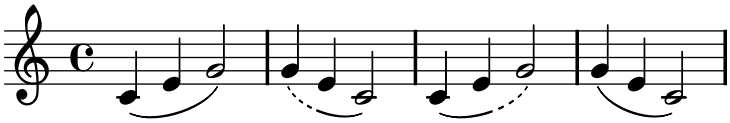
```
c4\( e g2\)
\phrasingSlurDashed
g4\( e c2\)
\phrasingSlurDotted
c4\( e g2\)
```

```
\phrasingSlurSolid
g4\( e c2\)
```
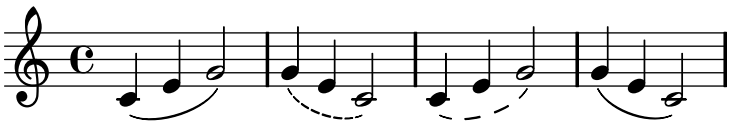


En utilisant `\phrasingslurHalfDashed`, la première moitié de la liaison aura un trait discontinu et continu pour la seconde. L'inverse s'obtient avec `\phrasingslurHalfSolid`.

```
c4\( e g2\)
\phrasingSlurHalfDashed
g4\( e c2\)
\phrasingSlurHalfSolid
c4\( e g2\)
\phrasingSlurSolid
g4\( e c2\)
```



Vous pouvez même personnaliser la densité des tirets d'une liaison :

```
c4\( e g2\)
\phrasingSlurDashPattern #0.7 #0.75
g4\( e c2\)
\phrasingSlurDashPattern #0.5 #2.0
c4\( e g2\)
\phrasingSlurSolid
g4\( e c2\)
```



La personnalisation des lignes discontinues est identique pour les liaisons de phrasé et les liaisons d'articulation. Pour plus de détails, référez-vous aux morceaux choisis de la section [Liaisons d'articulation], page 111.

Commandes prédéfinies

`\phrasingSlurUp`, `\phrasingSlurDown`, `\phrasingSlurNeutral`, `\phrasingSlurDashed`, `\phrasingSlurDotted`, `\phrasingSlurHalfDashed`, `\phrasingSlurHalfSolid`, `\phrasingSlurDashPattern`, `\phrasingSlurSolid`.

Voir aussi

Manuel d'initiation : Section "Non-imbrication des crochets et liaisons" dans *Manuel d'initiation*.

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 475, [Liaisons d'articulation], page 111.
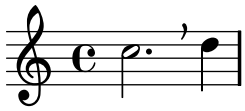
Morceaux choisis : Section "Signes d'interprétation" dans *Morceaux choisis*.

Référence des propriétés internes : Section "PhrasingSlur" dans *Référence des propriétés internes*.

Signes de respiration
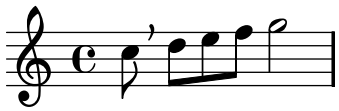
Les indications de respiration sont indiquées par la commande `\breathe`.

```
c2. \breathe d4
```



Un signe de respiration interrompt obligatoirement les ligatures, même automatiques. Pour passer outre ce fonctionnement, voir [Barres de ligature manuelles], page 80.

```
c8 \breathe d e f g2
```



LilyPond gère les *divisiones*, signes utilisés en notation ancienne pour indiquer les respirations. Pour de plus amples détails, voir [Divisions], page 362.

Morceaux choisis

Modification de l'indicateur de respiration

On peut choisir le glyphe imprimé par cette commande, en modifiant la propriété `text` de l'objet `BreathingSign`, pour lui affecter n'importe quelle indication textuelle.

```
\relative c'' {
  c2
  \override BreathingSign #'text = \markup { \musicglyph #"scripts.rvarcomma" }
  \breathe
  d2
}
```
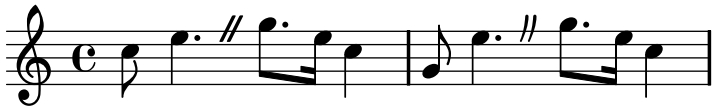


Insertion d'une césure

Une surcharge de la propriété `'text` de l'objet `BreathingSign` permet de créer une marque de césure. LilyPond dispose également d'une variante courbée.

```
\relative c'' {
  \override BreathingSign #'text = \markup {
    \musicglyph #"scripts.caesura.straight"
  }
  c8 e4. \breathe g8. e16 c4

  \override BreathingSign #'text = \markup {
    \musicglyph #"scripts.caesura.curved"
  }
  g8 e'4. \breathe g8. e16 c4
}
```



Voir aussi

Glossaire musicologique : Section “césure” dans *Glossaire*.

Manuel de notation : [Divisions], page 362.

Morceaux choisis : Section “Signes d’interprétation” dans *Morceaux choisis*.

Référence des propriétés internes : Section “BreathingEvent” dans *Référence des propriétés internes*, Section “BreathingSign” dans *Référence des propriétés internes*, Section “Breathing_sign_engraver” dans *Référence des propriétés internes*.

Chutes et sauts

Des indications de désinence peuvent être obtenues au moyen de la commande `\bendAfter`. Leur direction s’indique au moyen des signes plus (vers le haut) ou moins (vers le bas). Le chiffre indique l’intervalle avec la note de départ.

```
c2-\bendAfter #+4
c2-\bendAfter #-4
c2-\bendAfter #+6.5
c2-\bendAfter #-6.5
c2-\bendAfter #+8
c2-\bendAfter #-8
```



Lorsque vous saisissez des chutes ou des sauts, **n’oubliez pas** le tiret (`-`) qui précède la commande `\bendAfter`.

Morceaux choisis

Ajustement du galbe des chutes ou sauts

La propriété `shortest-duration-space` peut devoir être retouchée pour ajuster l’apparence des chutes ou sauts.

```
\relative c'' {
  \override Score.SpacingSpanner #'shortest-duration-space = #4.0
  c2-\bendAfter #5
  c2-\bendAfter #-4.75
  c2-\bendAfter #8.5
  c2-\bendAfter #-6
}
```



Voir aussi

Glossaire musical : Section “chute” dans *Glossaire*, Section “saut” dans *Glossaire*.

Morceaux choisis : Section “Signes d’interprétation” dans *Morceaux choisis*.
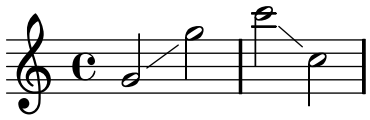
1.3.3 Signes d’interprétation sous forme de ligne

Cette partie traite de la manière de générer des signes d’interprétation d’aspect linéaire, tels les glissandos, arpèges et trilles.

Glissando

Un glissando relie une hauteur à une autre en passant par chaque hauteur intermédiaire. On l'obtient en accolant la commande `\glissando` à la première note.

```
g2\glissando g'
c2\glissando c,
```
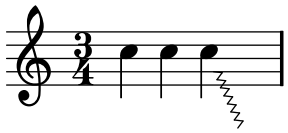


Un glissando est indiqué graphiquement, par une ligne ou des vaguelettes – voir Section 5.4.8 [Styles de ligne], page 476.

Morceaux choisis

Glissando contemporain

De nos jours, il peut arriver que la note d'arrivée d'un glissando soit absente de la partition. Pour ce faire, il vous faudra utiliser une cadence et « masquer » la note d'arrivée.

```
\relative c'' {
  \time 3/4
  \override Glissando #'style = #'zigzag
  c4 c
  \cadenzaOn
  c4\glissando
  \hideNotes
  c,,4
  \unHideNotes
  \cadenzaOff
  \bar "|"
}
```



Voir aussi

Glossaire musicologique : Section "glissando" dans *Glossaire*.

Manuel de notation : Section 5.4.8 [Styles de ligne], page 476.

Morceaux choisis : Section "Signes d'interprétation" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Glissando" dans *Référence des propriétés internes*.
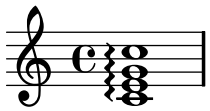
Problèmes connus et avertissements

Il n'est pas possible d'imprimer un texte (tel que *gliss.*) le long de la ligne de glissando.

Arpèges

On peut indiquer qu'un accord doit être arpégé en lui accolant la commande `\arpeggio` :

```
<c e g c>1\arpeggio
```
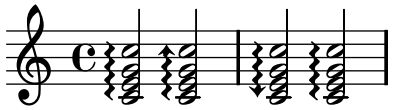


LilyPond dispose de différents graphismes pour indiquer un arpège ; `\arpeggioNormal` reviendra au style par défaut.

```
<c e g c>2\arpeggio

\arpeggioArrowUp
<c e g c>2\arpeggio

\arpeggioArrowDown
<c e g c>2\arpeggio

\arpeggioNormal
<c e g c>2\arpeggio
```



Des crochets indiquent que l'accord devra être plaqué et non arpégé :

```
<c e g c>2

\arpeggioBracket
<c e g c>2\arpeggio

\arpeggioParenthesis
<c e g c>2\arpeggio

\arpeggioParenthesisDashed
<c e g c>2\arpeggio

\arpeggioNormal
<c e g c>2\arpeggio
```



Les indications d'arpeggio peuvent se présenter sous la forme de ligne discontinue à l'aide de la propriété `'dash-details`. Pour plus de détails à ce propos, consultez [Liaisons d'articulation], page 111.

Un arpège peut parfois s'écrire de manière explicite, à l'aide de liaisons de tenue. Pour plus d'information, voir [Liaisons de prolongation], page 44.

Commandes prédéfinies

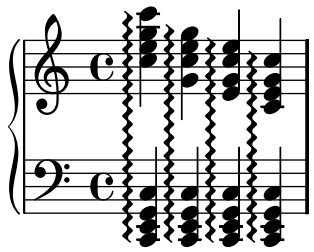
`\arpeggio`, `\arpeggioArrowUp`, `\arpeggioArrowDown`, `\arpeggioNormal`, `\arpeggioBracket`, `\arpeggioParenthesis` `\arpeggioParenthesisDashed`.

Morceaux choisis

Arpège distribué sur une partition pour piano

Dans une double portée pour piano (`PianoStaff`), un arpège peut s'étendre sur les deux portées grâce à la propriété `PianoStaff.connectArpeggios`.

```
\new PianoStaff \relative c'' <<
  \set PianoStaff.connectArpeggios = ##t
  \new Staff {
    <c e g c>4\arpeggio
    <g c e g>4\arpeggio
    <e g c e>4\arpeggio
    <c e g c>4\arpeggio
  }
  \new Staff {
    \clef bass
    \repeat unfold 4 {
      <c,, e g c>4\arpeggio
    }
  }
>>
```

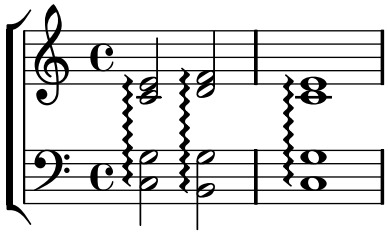


Arpège distribué pour un autre contexte que le piano

Il est possible de distribuer un arpège sur plusieurs portées d'un système autre que le `PianoStaff` dès lors que vous incluez le `Span_arpeggio_engraver` au contexte `Score`.

```
\score {
  \new ChoirStaff {
    \set Score.connectArpeggios = ##t
    <<
      \new Voice \relative c' {
        <c e>2\arpeggio
        <d f>2\arpeggio
        <c e>1\arpeggio
      }
      \new Voice \relative c {
        \clef bass
        <c g'>2\arpeggio
        <b g'>2\arpeggio
        <c g'>1\arpeggio
      }
```
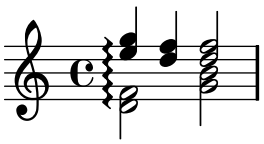
```
    >>
  }
  \layout {
    \context {
      \Score
      \consists "Span_arpeggio_engraver"
    }
  }
}
```



Arpège distribué sur plusieurs voix

Affecter le graveur `Span_arpeggio_engraver` au contexte de la portée (`Staff`) permet de distribuer un arpège sur plusieurs voix :

```
\new Staff \with {
  \consists "Span_arpeggio_engraver"
}
\relative c' {
  \set Staff.connectArpeggios = ##t
  <<
    { <e' g>4\arpeggio <d f> <d f>2 }
    \\
    { <d, f>2\arpeggio <g b>2 }
  >>
}
```



Voir aussi

Glossaire musicologique : Section “arpeggio” dans *Glossaire*.

Manuel de notation : [Liaisons d’articulation], page 111, [Liaisons de prolongation], page 44.

Morceaux choisis : Section “Signes d’interprétation” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Arpeggio” dans *Référence des propriétés internes*, Section “Slur” dans *Référence des propriétés internes*, Section “PianoStaff” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Il est impossible de mêler au même instant, dans un contexte `PianoStaff`, des lignes d’arpèges connectées et d’autres non connectées.

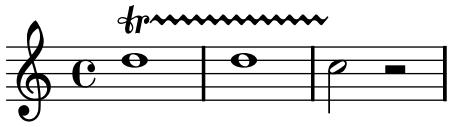
Il n’est pas possible d’imprimer des lignes d’arpège connectées sous forme de parenthèse.

Trilles

Les trilles brefs s'indiquent comme n'importe quelle ponctuation, avec un simple `\trill` ; voir [Articulations et ornements], page 102.

Les trilles plus longs sont délimités par `\startTrillSpan` et `\stopTrillSpan` :

```
d1\startTrillSpan
d1
c2\stopTrillSpan r2
```



Lorsqu'un saut de ligne intervient alors qu'une prolongation de trille est présente, l'indication de trille et sa prolongation sont rappellées sur la première note de la nouvelle ligne :

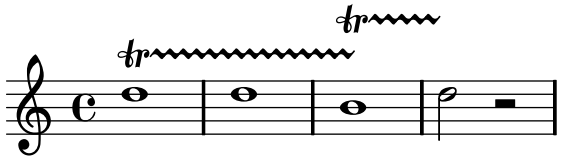
```
d1\startTrillSpan
\break
d1
c2\stopTrillSpan r2
```
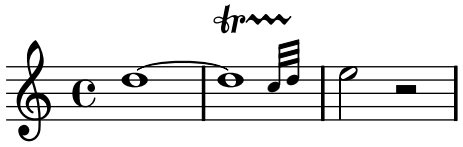




Lorsque des trilles interviennent sur une succession de hauteurs différentes, point n'est besoin d'expliciter la commande `\stopTrillSpan` puisque l'apparition d'un nouveau trille interrompt de fait celui qui le précédait :

```
d1\startTrillSpan
d1
b1\startTrillSpan
d2\stopTrillSpan r2
```



Dans l'exemple suivant, un trille se combine avec des notes d'ornement. La syntaxe d'une telle construction ainsi que le moyen de positionner les notes d'ornement avec précision est expliquée au chapitre [Notes d'ornement], page 95.

```
d1~\afterGrace
d1\startTrillSpan { c32[ d]\stopTrillSpan }
e2 r2
```
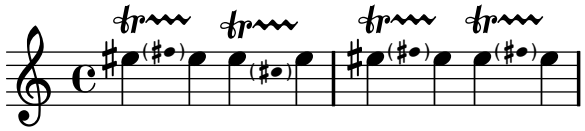


Les trilles qui font intervenir une hauteur précise peuvent être indiqués par la commande `pitchedTrill`. Le premier argument est la note réelle ; le second est une hauteur qui sera imprimée comme une tête de note noire entre parenthèses.

```
\pitchedTrill
e2\startTrillSpan fis
d2 c2\stopTrillSpan
```



Dans l'exemple suivant, le second trille de la deuxième mesure est ambigu – le fa qui est forcé n'est pas diésé. Pour lever toute ambiguïté, il suffit de forcer l'impression de l'altération en ajoutant un `!` à la note considérée.

```
\pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan cis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis
eis4\stopTrillSpan
\pitchedTrill
eis4\startTrillSpan fis!
eis4\stopTrillSpan
```



Commandes prédéfinies

`\startTrillSpan`, `\stopTrillSpan`.

Voir aussi
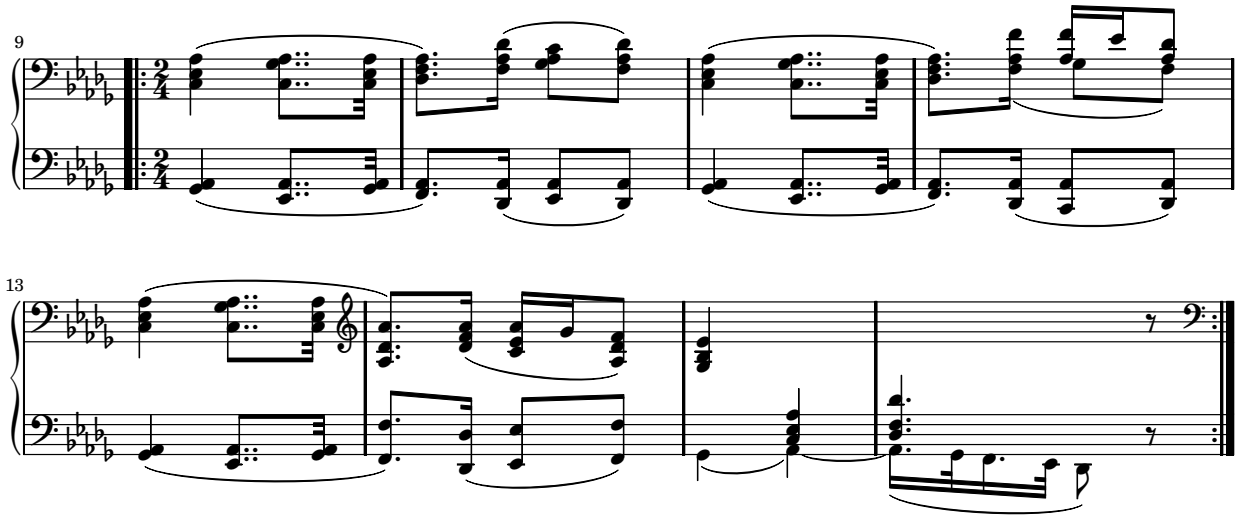
Glossaire musicologique : Section "trille" dans *Glossaire*.

Manuel de notation : [Articulations et ornements], page 102, [Notes d'ornement], page 95.

Morceaux choisis : Section "Signes d'interprétation" dans *Morceaux choisis*.

Référence des propriétés internes : Section "TrillSpanner" dans *Référence des propriétés internes*.

1.4 Répétitions et reprises



La répétition est une notion essentielle en musique, et il existe de nombreuses façons de mettre en œuvre et noter ce concept. LilyPond prend en charge les types de répétition suivants :

`volta` Le passage répété n'est pas développé, mais il est encadré par des barres de reprise et peut se terminer par plusieurs fins alternatives – ou *volte* – imprimées de gauche à droite sous des crochets. Lorsque la répétition commence au début de la pièce, aucune barre de reprise n'est gravée au début de la partition. Il s'agit de la notation courante des reprises avec fins alternatives.

`unfold` La musique répétée est développée dans la partition autant de fois qu'indiqué. Ceci est particulièrement utile pour de la musique répétitive.

`percent` Des barres obliques ou signes de pourcentage indiquent la répétition de temps ou de mesures.

`tremolo` Ce type permet de réaliser des trémolos sous forme de liens de croches.

1.4.1 Répétition d'un long passage

Cette section présente la syntaxe des répétitions longues – c'est-à-dire plusieurs mesures. Ces répétitions peuvent prendre deux formes : encadrées par des barres de reprises, ou bien développées dans la partition. Les barres et autres signes de reprise peuvent être contrôlés manuellement.

Répétitions courantes
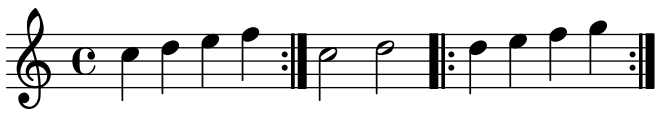
On peut indiquer une répétition de la façon suivante :

```
\repeat volta nombre_de_fois expression_musicale
```

où *`expression_musicale`* représente ce qui doît être répété.

Les reprises courantes, sans alternative, s'indiquent comme ceci :

```
\repeat volta 2 { c4 d e f }
c2 d
\repeat volta 2 { d4 e f g }
```
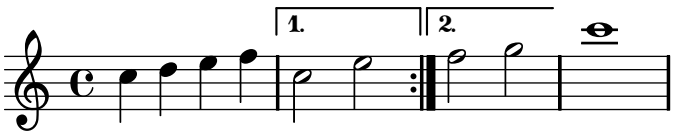


On peut ajouter une fin alternative à l'aide de la commande `\alternative`. Chaque *`alternative`* est une expression musicale en elle-même ; il faudra donc les regrouper par des accolades.

```
\repeat volta nombre_de_fois expression_musicale
\alternative {
  { expression_musicale }
}
```

Si l'on donne trop peu d'alternatives en regard du nombre de fois où le passage doit être rejoué, la première alternative sera jouée plusieurs fois.
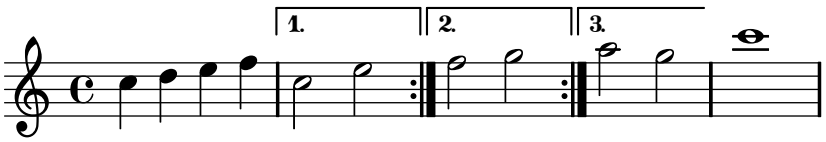
Voici une simple reprise avec une fin alternative :

```
\repeat volta 2 { c4 d e f | }
\alternative {
  { c2 e | }
  { f2 g | }
}
c1
```



Et une répétition avec plusieurs alternatives :

```
\repeat volta 3 { c4 d e f | }
\alternative {
  { c2 e | }
  { f2 g | }
  { a2 g | }
}
c1
```
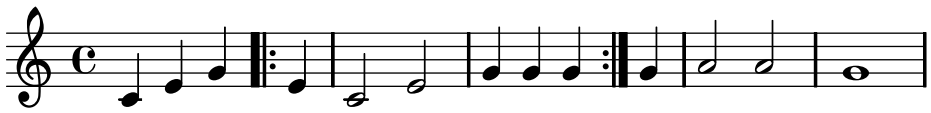


Note : Lorsqu'il y a plus d'une alternative, prenez garde à ce qu'aucun caractère n'apparaisse entre l'accolade fermant une alternative et l'accolade ouvrant la suivante, au risque de ne pas obtenir le nombre voulu d'alternatives.

Note : Une clause `\relative` ne doît jamais se trouver à l'intérieur d'une section `\repeat` : vous aurez immanquablement des portées parasites. Voir Section "Apparition d'une portée supplémentaire" dans *Utilisation des programmes*.
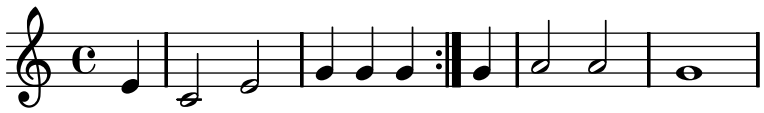
Lorsqu'une reprise sans fins alternatives débute au milieu d'une mesure, elle devrait se terminer aussi au milieu d'une mesure, de telle sorte que les mesures soient complètes. En pareil cas, les indications de reprise ne constituent pas des barres de mesure à proprement parler ; il n'est donc pas nécessaire de faire appel à la commande `\partial` ou à des contrôles d'intégrité de mesure.

```
% no \partial here
c4 e g  % no bar check here
% no \partial here
\repeat volta 4 {
  e4 |
  c2 e |
  % no \partial here
  g4 g g  % no bar check here
}
% no \partial here
g4 |
a2 a |
g1 |
```



Il est possible de créer des reprises en début de morceau avec une levée. Le cas est similaire à ce que nous venons de voir. Toutefois, l'utilisation d'un `\partial` est ici nécessaire pour respecter cette entâme.

```
\partial 4  % required
\repeat volta 4 {
  e4 |
  c2 e |
  % no \partial here
  g4 g g  % no bar check here
}
% no \partial here
g4 |
a2 a |
g1 |
```
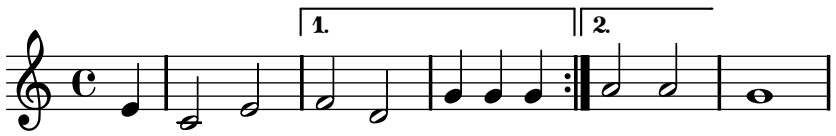


Lorsqu'une reprise débute par une mesure incomplète et a des fins alternatives, il est indispensable d'ajuster manuellement la propriété `Timing.measureLength` selon les préceptes suivants :

- au début de chacune des mesures incomplètes du bloc `\alternative`, ce qui est le cas en principe pour chaque fin d'alternative sauf – la plupart du temps – pour la dernière.
- au début de chaque alternative à l'exception de la première.

```
\partial 4
\repeat volta 2 { e4 | c2 e | }
\alternative {
  {
    f2 d |
    \set Timing.measureLength = #(ly:make-moment 3 4)
    g4 g g  % optional bar check is allowed here
  }
```

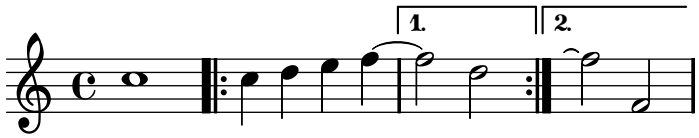
```
  {
    \set Timing.measureLength = #(ly:make-moment 4 4)
    a2 a |
  }
}
g1 |
```



La propriété `measureLength` est abordée plus en détail au chapitre [Gestion du temps], page 100.

Des liaisons de tenue peuvent être ajoutées à toute alternative :

```
c1
\repeat volta 2 { c4 d e f ~ }
\alternative {
  { f2 d }
  { f2\repeatTie f, }
}
```
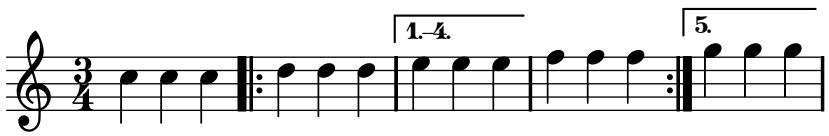


Morceaux choisis

Diminution de la taille du crochet d'alternative

Les crochets indiquant les fins alternatives s'étalent tout au long ce celle-ci. On peut les raccourcir en définissant la propriété `voltaSpannerDuration`. Dans l'exemple suivant, le crochet ne se prolonge que sur une mesure à 3/4.

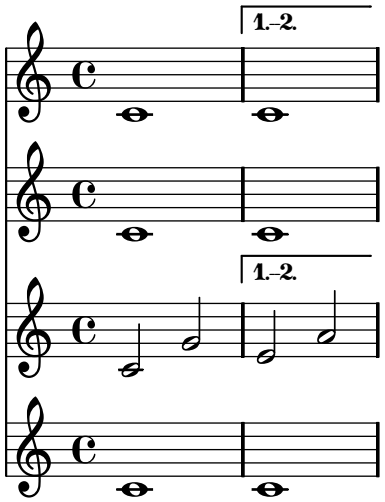
```
\relative c'' {
  \time 3/4
  c4 c c
  \set Score.voltaSpannerDuration = #(ly:make-moment 3 4)
  \repeat volta 5 { d4 d d }
  \alternative {
    {
      e4 e e
      f4 f f
    }
    { g4 g g }
  }
}
```



Ajout du crochet de reprise à d'autres portées

D'ordinaire, le graveur `Volta_engraver` réside dans le contexte `Score` ; les crochets précédant une reprise s'impriment donc seulement au-dessus de la portée du haut. On peut ajuster cela en déplaçant ce graveur vers les contextes de portée (`Staff`) qui doivent comporter ces crochets.

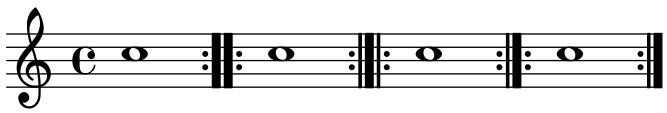
```
<<
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
  \new Staff \with { \consists "Volta_engraver" } { c'2 g' e' a' }
  \new Staff { \repeat volta 2 { c'1 } \alternative { c' } }
>>
```



Succession de reprise et style de barre par défaut

LilyPond dispose de trois styles de barre différents pour indiquer une succession de reprises. Vous devez opter pour un style par défaut, à l'aide de la propriété `doubleRepeatType`.

```
\relative c'' {
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|.|:"
  \repeat volta 1 { c1 }
  \set Score.doubleRepeatType = #":|.:"
  \repeat volta 1 { c1 }
}
```



Voir aussi

Glossaire musicologique : Section "répétition" dans *Glossaire*, Section "volta" dans *Glossaire*.

Manuel de notation : [Barres de mesure], page 84, Section 5.1.4 [Modification des greffons de contexte], page 464, [Gestion du temps], page 100.

Morceaux choisis : Section "Répétitions" dans *Morceaux choisis*.

Référence des propriétés internes : Section “VoltaBracket” dans *Référence des propriétés internes*, Section “RepeatedMusic” dans *Référence des propriétés internes*, Section “VoltaRepeatedMusic” dans *Référence des propriétés internes*, Section “UnfoldedRepeatedMusic” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

L’extension d’une liaison à partir d’un bloc `\repeat` sur un bloc `\alternative` n’est possible que pour la première alternative. Selon le même principe, une liaison ne saurait partir de la fin d’une alternative pour se terminer au début de la reprise.

Le développement, à l’aide de la commande `\unfoldRepeats`, d’une répétition qui commence sur une mesure incomplète et contient un bloc `alternative` avec modification de la propriété `measureLength` entraînera des messages d’erreur concernant le placement des barres de mesure.

Des reprises imbriquées telles que

```
\repeat ...
\repeat ...
\alternative
```

présentent une ambiguïté, dans la mesure où l’on ne sait à quelle section `\repeat` attribuer la section `\alternative`. Pour résoudre cette ambiguïté, il convient de toujours insérer la commande `\alternative` à l’intérieur de la section `\repeat`. Il est préférable, dans une telle situation, d’utiliser des accolades pour plus de clarté.

Indications de reprise manuelles

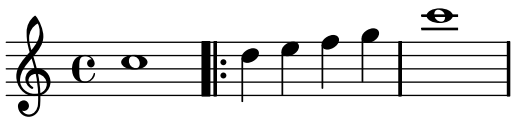
> **Note :** Les méthodes présentées dans les lignes qui suivent ne devraient servir à indiquer que des constructions de répétition inhabituelles. En règle générale, il vaut mieux recourrir à la fonction `\repeat` pour créer une reprise ou bien insérer la barre de mesure adéquate. Pour plus d’information, voir le chapitre [Barres de mesure], page 84.

La propriété `repeatCommands` sert à contrôler la mise en forme des reprises. On la définit par une suite de commandes de reprise Scheme.

`start-repeat`

Pour imprimer une barre de reprise `|:`

```
c1
\set Score.repeatCommands = #'(start-repeat)
d4 e f g
c1
```
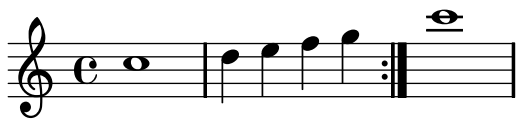


Traditionnellement, on n’imprime pas de signe de reprise en début de morceau.

`end-repeat`

Pour imprimer une barre de reprise `:|`

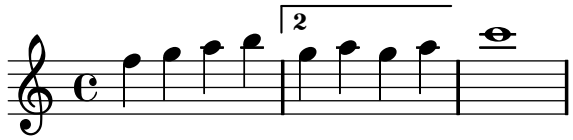
```
c1
d4 e f g
\set Score.repeatCommands = #'(end-repeat)
c1
```



`(volta` *nombre*`) ... (volta #f)`

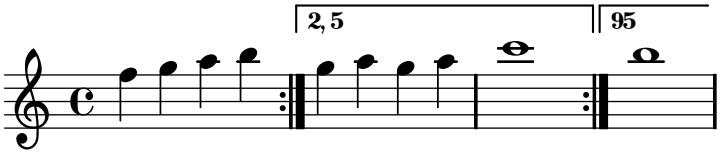
Pour obtenir un crochet indiquant le numéro de l'alternative. Pour que le crochet s'imprime effectivement, il faut spécifier explicitement l'endroit où il doît se terminer.

```
f4 g a b
\set Score.repeatCommands = #'((volta "2"))
g4 a g a
\set Score.repeatCommands = #'((volta #f))
c1
```
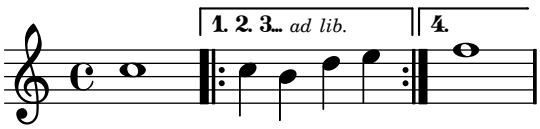


Plusieurs commande de reprise peuvent intervenir au même moment :

```
f4 g a b
\set Score.repeatCommands = #'((volta "2, 5") end-repeat)
g4 a g a
c1
\set Score.repeatCommands = #'((volta #f) (volta "95") end-repeat)
b1
\set Score.repeatCommands = #'((volta #f))
```



Le crochet indiquant une alternative peut contenir aussi du texte. Il peut s'agir d'un ou plusieurs nombres ou bien d'une indication textuelle (*markup*) – voir Section 1.8.2 [Mise en forme du texte], page 202. Le plus simple, dans le cas d'une indication textuelle, est de tout d'abord définir ce `markup`, puis de l'inclure dans une liste Scheme.

```
voltaAdLib = \markup { 1. 2. 3... \text \italic { ad lib. } }
\relative c'' {
  c1
  \set Score.repeatCommands = #(list(list 'volta voltaAdLib) 'start-repeat)
  c4 b d e
  \set Score.repeatCommands = #'((volta #f) (volta "4.") end-repeat)
  f1
  \set Score.repeatCommands = #'((volta #f))
}
```

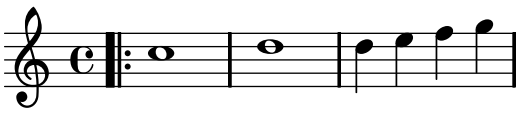


Morceaux choisis

Impression d'une barre de reprise en début de morceau

Bien qu'allant à l'encontre des usages en matière de gravure, vous imprimerez une barre de reprise (`|:`) en début de partition si vous surchargez la propriété adéquate :

```
\relative c'' {
  \once \override Score.BreakAlignment #'break-align-orders =
    #(make-vector 3 '(instrument-name
                      left-edge
                      ambitus
                      breathing-sign
                      clef
                      key-signature
                      time-signature
                      staff-bar
                      custos))
  \once \override Staff.TimeSignature #'space-alist =
    #'((first-note . (fixed-space . 2.0))
       (right-edge . (extra-space . 0.5))
       ;; free up some space between time signature
       ;; and repeat bar line
       (staff-bar . (extra-space . 1)))
  \bar "|:"
  c1
  d1
  d4 e f g
}
```



Voir aussi

Manuel de notation : [Barres de mesure], page 84, Section 1.8.2 [Mise en forme du texte], page 202.

Morceaux choisis : Section "Répétitions" dans *Morceaux choisis*.

Référence des propriétés internes : Section "VoltaBracket" dans *Référence des propriétés internes*, Section "RepeatedMusic" dans *Référence des propriétés internes*, Section "VoltaRepeatedMusic" dans *Référence des propriétés internes*.
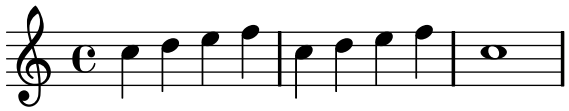
Répétitions explicites

Adjoindre à la commande `\repeat` l'opion `unfold` permet de s'affranchir de ressaisir de la musique répétitive. En voici la syntaxe :

```
\repeat unfold nombre_de_fois expression_musicale
```
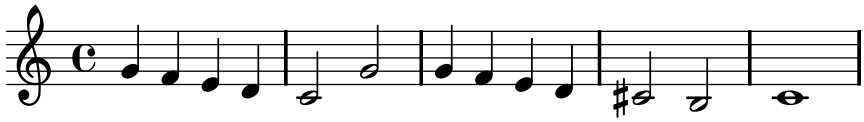
Le contenu de *expression_musicale* sera donc expansé autant de fois que stipulé par *nombre_de_fois*.

```
\repeat unfold 2 { c4 d e f }
c1
```



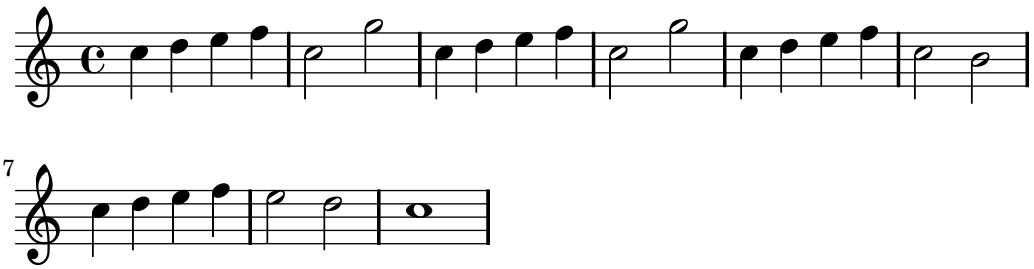
Une répétition expansée peut aussi avoir une fin alternative :

```
\repeat unfold 2 { g4 f e d }
\alternative {
  { c2 g' }
  { cis,2 b }
}
c1
```
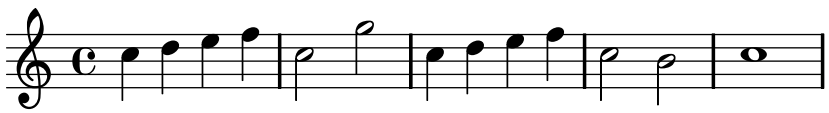


Si l'on donne trop peu d'alternatives en regard du nombre de fois où le passage doit être rejoué, la première alternative sera jouée plusieurs fois.

```
\repeat unfold 4 { c4 d e f }
\alternative {
  { c2 g' }
  { c,2 b }
  { e2 d }
 }
c1
```



S'il y a par contre plus d'alternatives que de répétitions, les alternatives superflues seront tout simplement ignorées et ne seront pas imprimées.

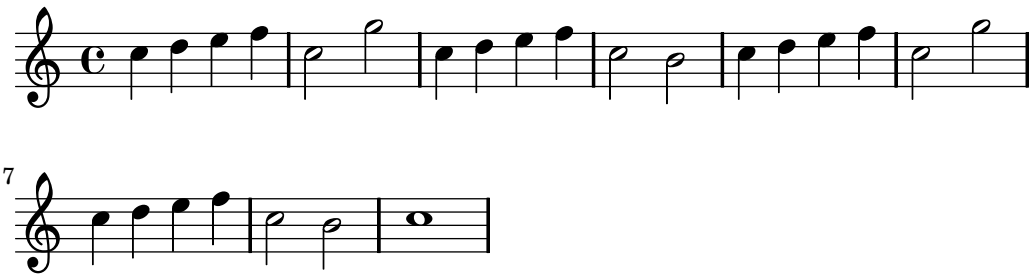
```
\repeat unfold 2 { c4 d e f }
\alternative {
  { c2 g' }
  { c,2 b }
  { e2 d }
}
c1
```



Vous pouvez imbriquer plusieurs fonctions `unfold`, qu'elles comportent ou non des fins alternatives :

```
\repeat unfold 2 {
  \repeat unfold 2 { c4 d e f }
  \alternative {
```

```
    { c2 g' }
    { c,2 b }
  }
}
c1
```



Note : L'insertion d'un `\relative` dans une section `\repeat` sans déclaration explicite du contexte `Voice` génèrera une portée supplémentaire – voir Section "Apparition d'une portée supplémentaire" dans *Utilisation des programmes*.

Voir aussi

Morceaux choisis : Section "Répétitions" dans *Morceaux choisis*.

Référence des propriétés internes : Section "RepeatedMusic" dans *Référence des propriétés internes*, Section "UnfoldedRepeatedMusic" dans *Référence des propriétés internes*.

1.4.2 Autres types de répétition

Nous abordons ici les reprises de courte durée. Il en existe deux formes, à savoir la répétition d'une même note sur quelques mesures – représentée par une barre oblique ou le signe pourcent – et les trémolos.

Répétitions de mesure
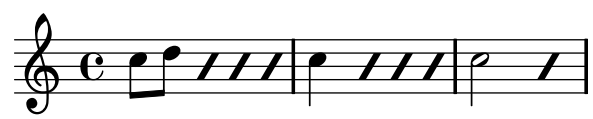
Le style de « reprise en pourcent » sert à répéter une séquence de notes. Elle sera imprimée une fois, puis remplacée par un symbole spécial.

En voici la syntaxe :

```
\repeat percent nombre expression_musicale
```
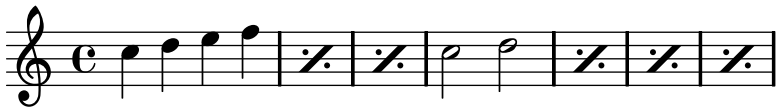
Les séquences inférieures à une mesure sont remplacées par une barre oblique.

```
\repeat percent 4 { c8 d }
\repeat percent 4 { c4 }
\repeat percent 2 { c2 }
```
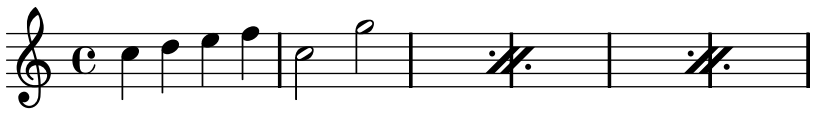


Les séquences d'une ou deux mesures sont remplacées par un symbole qui ressemble au symbole de pourcentage.

```
\repeat percent 3 { c4 d e f }
\repeat percent 4 { c2 d }
```



```
\repeat percent 3 { c4 d e f | c2 g' }
```

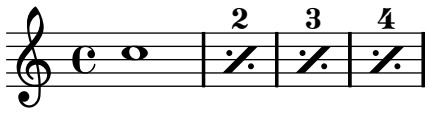


Morceaux choisis

Compteur de répétition en pourcent

Les répétitions de plus de 2 mesures sont surmontées d'un compteur, si l'on active la propriété `countPercentRepeats` comme le montre l'exemple suivant :

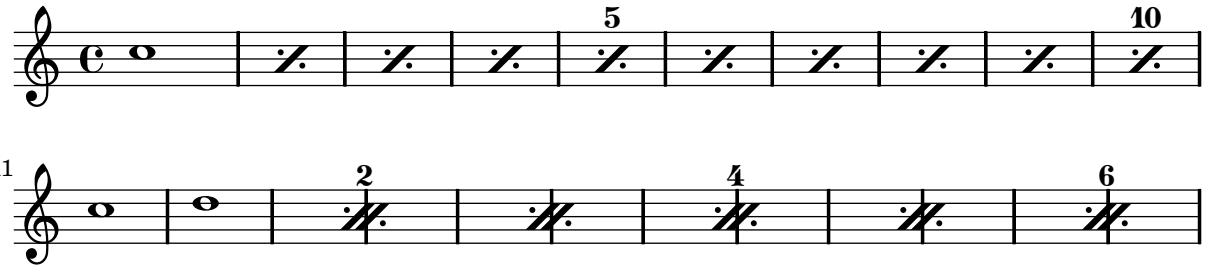
```
\relative c'' {
  \set countPercentRepeats = ##t
  \repeat percent 4 { c1 }
}
```



Affichage du numéro de répétition en pourcent

Le numéro de mesure répétée sera imprimé à intervalle régulier si vous déterminez la propriété de contexte `repeatCountVisibility`.

```
\relative c'' {
  \set countPercentRepeats = ##t
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 5)
  \repeat percent 10 { c1 } \break
  \set repeatCountVisibility = #(every-nth-repeat-count-visible 2)
  \repeat percent 6 { c1 d1 }
}
```
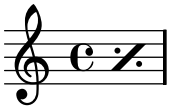


Répétition en pourcent isolée

Des symboles de pourcentage isolés peuvent aussi être obtenus, au moyen d'un silence multi-mesures dont on modifie l'aspect :

```
makePercent =
#(define-music-function (parser location note) (ly:music?)
   "Make a percent repeat the same length as NOTE."
   (make-music 'PercentEvent
               'length (ly:music-length note)))
```

```
\relative c'' {
  \makePercent s1
}
```



Voir aussi

Glossaire musicologique : Section “percent repeat” dans *Glossaire*, Section “simile” dans *Glossaire*.

Morceaux choisis : Section “Répétitions” dans *Morceaux choisis*.

Référence des propriétés internes : Section “RepeatSlash” dans *Référence des propriétés internes*, Section “PercentRepeat” dans *Référence des propriétés internes*, Section “DoublePercentRepeat” dans *Référence des propriétés internes*, Section “DoublePercentRepeatCounter” dans *Référence des propriétés internes*, Section “PercentRepeatCounter” dans *Référence des propriétés internes*, Section “PercentRepeatedMusic” dans *Référence des propriétés internes*.

Problèmes connus et avertissements
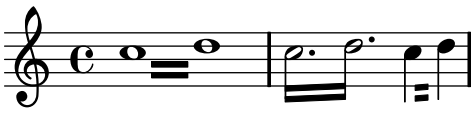
LilyPond ne dispose que de trois signes différents pour indiquer une répétition « en pourcent » : une simple barre oblique qui représente un seul temps quelle que soit la durée des notes répétées ; une barre oblique simple entourée de points qui représente une mesure entière ; une double barre oblique entourée de points et placée sur la barre de mesure pour indiquer une répétition de deux mesures.

Répétitions en trémolo

Il y a deux formes de trémolo : la répétition alernative de deux notes ou accords, et la répétition rapide d’une seule note ou d’un accord. Lorsqu’il est constitué d’une alternance répétitive, le trémolo s’indique en ajoutant des barres de ligature entre les deux notes ou accords concernés. Lorsqu’il s’agit d’une répétition rapide, des barres de ligature penchées sont ajoutées à la note en question.

On peut placer une notation de trémolo entre deux notes, avec la commande `\repeat` suivie du style trémolo :

```
\repeat tremolo 8 { c16 d }
\repeat tremolo 6 { c16 d }
\repeat tremolo 2 { c16 d }
```



La syntaxe de `\repeat tremolo` requiert espressément deux notes encadrées par des accolades, et le nombre de répétitions exprimé en durée d’une note (pointée ou non). Ainsi, dans l’exemple ci-dessus, `\repeat tremolo 7` est valide car correspond à une note doublement pointée, à l’inverse de `\repeat tremolo 9`.

La durée d’un trémolo est égale à la durée de l’expression entre accolades multipliée par le nombre de fois à répéter : `\repeat tremolo 8 { c16 d16 }` correspond donc à la valeur d’une ronde, et sera représenté par deux rondes séparées par des barres de trémolo.
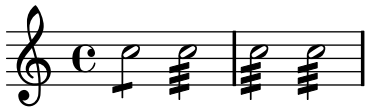
On peut indiquer de la même manière un trémolo sur une seule note, qu’il faudra alors laisser sans accolades :

```
\repeat tremolo 4 c'16
```



Le même résultat s'obtient en faisant suivre la note considérée de deux points et d'un nombre (`note:`*`nombre`*). Le nombre en question correspond à la valeur de la subdivision ; il doit être au moins de 8, auquel cas la hampe sera barrée par un seul trait de ligature. Si ce nombre est omis, la dernière valeur – telle que mémorisée dans `tremoloFlags` – sera utilisée.

```
c2:8 c:32
c: c:
```

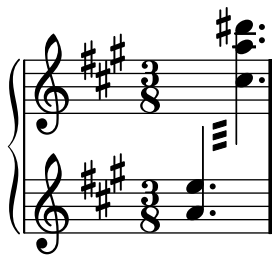


Morceaux choisis

Trémolo et changement de portée

Dans la mesure où `\repeat tremolo` requiert deux arguments musicaux pour un trémolo d'accords, la note ou l'accord de la portée opposée doît être encadré par des accolades et se voir adjoindre la commande `\change Staff`.
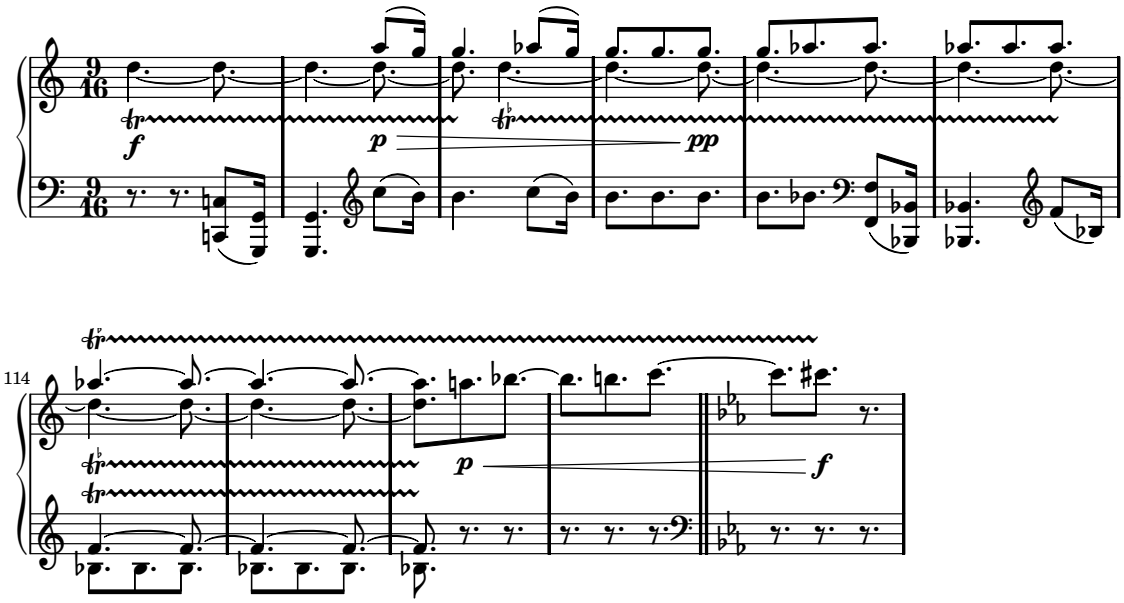
```
\new PianoStaff <<
  \new Staff = "up" \relative c'' {
    \key a \major
    \time 3/8
    s4.
  }
  \new Staff = "down" \relative c'' {
    \key a \major
    \time 3/8
    \voiceOne
    \repeat tremolo 6 {
      <a e'>32
      {
        \change Staff = "up"
        \voiceTwo
        <cis a' dis>32
      }
    }
  }
>>
```



Voir aussi

Morceaux choisis : Section “Répétitions” dans *Morceaux choisis*.

1.5 Notes simultanées



La notion musicale de polyphonie fait référence au fait d'avoir plus d'une voix simultanément dans une pièce. Dans LilyPond, la notion de polyphonie fait référence au fait d'avoir plus d'une voix sur la même portée.
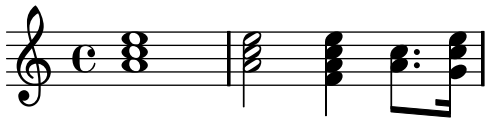
1.5.1 Monophonie

Nous allons voir ici comment gérer plusieurs notes simultanées dans une même voix.

Notes en accords

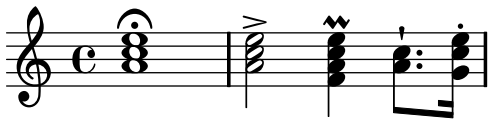
Un accord est formé en mettant une série de hauteurs entre `<` et `>`. Un accord peut être suivi d'une durée comme une simple note.

```
<a c e>1 <a c e>2 <f a c e>4 <a c>8. <g c e>16
```



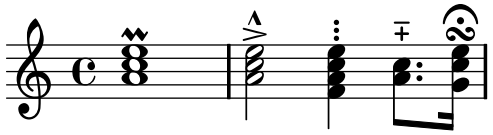
Un accord peut être suivi d'une indication d'articulation comme une simple note.

```
<a c e>1\fermata <a c e>2-> <f a c e>4\prall <a c>8.^| <g c e>16-.
```



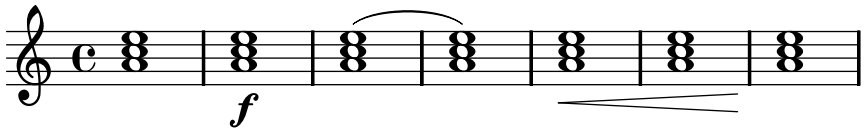
Certaines notes, au sein même d'un accord, peuvent être affectées d'une articulation ou d'un ornement :

```
<a c\prall e>1 <a-> c-^ e>2 <f-. a c-. e-.>4 <a-+ c-->8. <g\fermata c e\turn>16
```



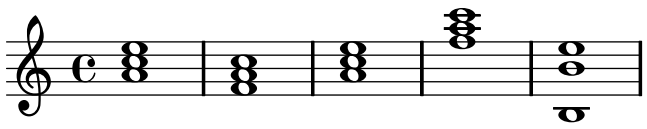
Certains éléments de notation, tels que nuances, soufflets et liaisons, ne peuvent se rattacher qu'à l'accord et non aux notes qui le composent, sous peine de ne les voir s'imprimer.

```
<a\f c( e>1 <a c) e>\f <a\< c e>( <a\! c e>) <a c e>\< <a c e> <a c e>\!
```



Les accords peuvent être saisis en mode relatif. Dans un accord, l'octave de chaque hauteur saisie est fonction de la précédente, à l'exception de la première qui, elle, sera positionnée en fonction de la première hauteur mentionnée dans l'accord précédent ou de la note individuelle précédente. Les autres notes au sein de l'accord se positionnent relativement à la précédente.

```
<a c e>1 <f a c> <a c e> <f' a c> <b, e b,>
```



Pour plus d'information à propos des accords, voir Section 2.7 [Notation des accords], page 343.

Voir aussi

Glossaire musical : Section "accord" dans *Glossaire*.

Manuel d'initiation : Section "Combinaison de notes en accords" dans *Manuel d'initiation*.

Manuel de notation : Section 2.7 [Notation des accords], page 343. [Articulations et ornements], page 102, [Octaves relatives], page 2, Section 1.5.2 [Plusieurs voix], page 141.

Morceaux choisis : Section "Notes simultanées" dans *Morceaux choisis*.

Problèmes connus et avertissements

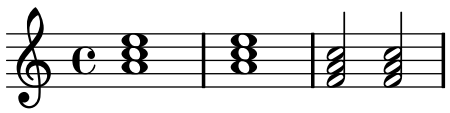
Un accord comportant plus de deux notes dans le même « espace de portée » – tel que '`<e f! fis!>`' – conduit immanquablement à des chevauchements. En fonction de la situation, un meilleur rendu peut nécessiter de recourir à

- l'utilisation temporaire de Section 1.5.2 [Plusieurs voix], page 141, '`<< f! \\ <e fis!> >>`',
- une transcription enharmonique d'une ou plusieurs hauteurs, '`<e f ges>`', ou
- des [Clusters], page 140.

Répétition d'accords

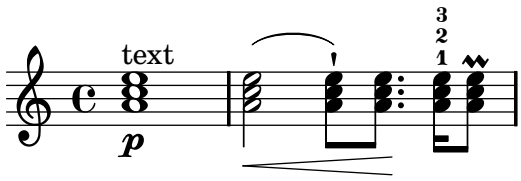
Dans le but de vous épargner de la saisie, LilyPond dispose d'un raccourci – symbolisé par la lettre `q` – qui a pour effet de répéter le dernier accord saisi :

```
<a c e>1 q <f a c>2 q
```
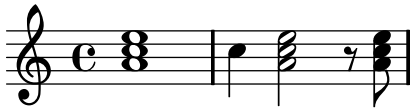


À l'instar de n'importe quel accord, le symbole de répétition peut être affublé d'une durée, de signes d'articulation, *markups*, liaisons, ligatures... En fait, c'est la structure du dernier accord qui est dupliquée.

```
<a c e>1\p^"text" q2\<( q8)[-| q8.]\! q16-1-2-3 q8\prall
```



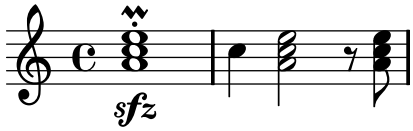
Dans la mesure où le symbole de répétition d'accord enregistre la structure du dernier accord construit, il est tout à fait possible de l'utiliser même après une succession de notes individuelles et de silences :

```
<a c e>1 c4 q2 r8 q8
```



Le symbole de répétition d'accord ne prend en charge que les hauteurs, en aucun cas les nuances, articulations ou ornementations, qu'elles aient été attachées aux notes le composant ou à l'ensemble.

```
<a-. c\prall e>1\sfz c4 q2 r8 q8
```



Voir aussi

Manuel de notation : Section 2.7 [Notation des accords], page 343, [Articulations et ornements], page 102.
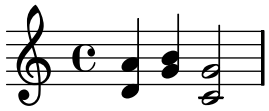
Fichiers d'initialisation : '`ly/chord-repetition-init.ly`'.

Expressions simultanées
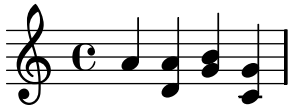
Lorsqu'une ou plusieurs expressions musicales sont encadrées par des doubles chevrons, elles sont considérées comme étant simultanées. Si la première expression débute par une note unique ou si l'intégralité de l'expression simultanée est explicitement rattachée à une voix en particulier, elle sera placée sur une seule portée. Dans le cas contraire, les éléments d'une expression simultanée seront placés sur des portées distinctes.

Voici deux exemples d'expression simultanée sur une même portée :

```
\new Voice {  % explicit single voice
  << { a4 b g2 } { d4 g c,2 } >>
}
```



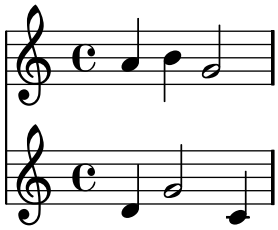
```
% single first note
a << { a4 b g } { d4 g c, } >>
```



Cette manière de procéder est utile lorsque les éléments de l'expression ont des rythmes identiques. Dès que vous tenterez d'attacher sur une même hampe des notes de durée différente, vous générerez des erreurs.

Dans l'exemple suivant, l'expression simultanée génère implicitement plusieurs portées :

```
% no single first note
<< { a4 b g2 } { d4 g2 c,4 } >>
```

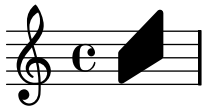


En pareil cas, des rythmes différents ne sont source d'aucun problème.

Clusters

Un cluster indique un agrégat de sons. On peut le représenter par une plage limitée par un *ambitus* (notes extrêmes). On obtient une telle notation en appliquant la fonction `makeClusters` à une séquence d'accords, comme

```
\makeClusters { <g b>2 <c g'> }
```



Des notes ordinaires et des clusters peuvent cohabiter sur une même portée, y compris simultanément – en pareil cas, rien ne sera fait pour tenter d'empêcher les chevauchements entre notes et clusters.

Voir aussi

Glossaire musicologique : Section "cluster" dans *Glossaire*.

Morceaux choisis : Section "Notation simultanée" dans *Morceaux choisis*.

Référence des propriétés internes : Section "ClusterSpanner" dans *Référence des propriétés internes*, Section "ClusterSpannerBeacon" dans *Référence des propriétés internes*, Section "Cluster_spanner_engraver" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

L'apparence d'un cluster sera extrèmement resserrée s'il ne comporte au moins deux accords.

Dans la mesure où un cluster ne possède pas de hampe, il n'y a aucun moyen d'en connaître la durée ; cependant la longueur du signe imprimé dépend directement de la durée affectée aux accords qui le définissent.

Seul un silence peut séparer deux clusters.

Les clusters ne sont pas reproduits en MIDI.

1.5.2 Plusieurs voix

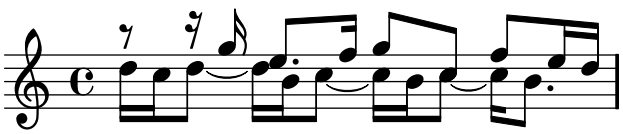
Nous allons nous intéresser, dans les paragraphes qui suivent, à la gestion de notes simultanées réparties sur plusieurs voix ou plusieurs portées.

Polyphonie sur une portée

Instanciation explicite des voix

La manière la plus facile d'entrer des fragments avec plus d'une voix sur une portée est est la suivante :

```
\new Staff <<
  \new Voice = "first"
    { \voiceOne r8 r16 g e8. f16 g8[ c,] f e16 d }
  \new Voice= "second"
    { \voiceTwo d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```



Vous constaterez que les voix sont créées explicitement et qu'elles sont nommées. Les commandes `\voiceOne` ... `\voiceFour` déterminent les voix de telle sorte que les première et troisième auront des hampes vers le haut, et les deuxième et quatrième vers le bas. D'autre part, les notes des troisième et quatrième voix seront quelque peu décalées, tout comme leurs silences, afin d'éviter les collisions. La commande `\oneVoice` permet de retrouver les réglages par défaut.

Polyphonie temporaire
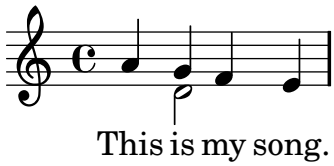
Un fragment temporairement polyphonique se construit de la manière suivante :

```
<< { \voiceOne ... }
  \new Voice { \voiceTwo ... }
>> \oneVoice
```

En fait, la première expression d'une polyphonie temporaire reste dans le même contexte `Voice` que celui existant auparavant et qui perdurera après ce fragment. Les autres expressions entre doubles chevrons seront assignées à des voix temporaires distinctes. C'est la raison pour laquelle les paroles qui suivaient la voix avant la polyphonie continueront à le faire durant ce passage polyphonique et après lui.

```
<<
  \new Voice = "melody" {
    a4
    <<
      {
        \voiceOne
        g f
```

```
      }
      \new Voice {
        \voiceTwo
        d2
      }
    >>
    \oneVoice
    e4
  }
  \new Lyrics \lyricsto "melody" {
  This is my song.
  }
>>
```
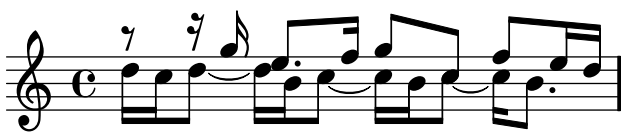


Vous remarquerez que les commandes `\voiceOne` et `\voiceTwo` permettent d'obtenir des réglages différents pour chacune des voix.

La construction avec un double antislash

Une construction de la forme `<< {...} \\ {...} >>`, dans laquelle plusieurs expressions sont séparées par des doubles obliques inversées, se comporte différemment de celle sans séparateurs : **tous** les membres de cette contruction seront assignés à de nouveaux contextes de voix. Ces contextes de voix, créés implicitement, portent les noms `"1"`, `"2"`, etc. Dans chacun de ces contextes, le positionnement des liaisons, la direction des hampes, etc. sont réglés de manière appropriée. En voici un exemple :

```
<<
  { r8 r16 g e8. f16 g8[ c,] f e16 d }
  \\
  { d16 c d8~ d16 b c8~ c16 b c8~ c16 b8. }
>>
```
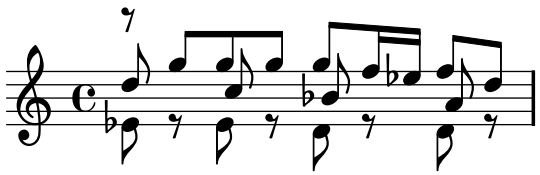


Cette syntaxe peut être utilisée dans la mesure où la création puis la disparition de voix temporaires sont sans conséquence. Les réglages de ces voix créées implicitement sont les mêmes que si elles avaient été créées à l'aide des commandes `\voiceOne` à `\voiceFour`, dans leur ordre d'apparition.

Dans l'exemple qui suit, la voix intermédiaire à des hampes vers le haut. Nous la plaçons donc en troisième position, de telle sorte qu'elle adopte les réglages de `\voiceThree` qui correspondent à ce que nous voulons. Grâce à des espaces invisibles, nous évitons de surcharger la portée avec des demis soupirs.

```
<<
  { r8 g g  g g f16 ees f8 d }
  \\
  { ees,8 r ees r d r d r }
```

```
  \\
  { d'8 s c s bes s a s }
>>
```
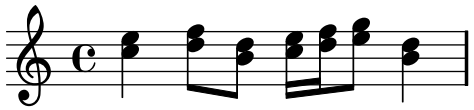


En dehors des cas les plus simples, nous vous invitons à toujours créer les contextes de voix de manière explicite. Voir à ce sujet Section “Contextes et graveurs” dans *Manuel d’initiation* et Section “Instanciation explicite des voix” dans *Manuel d’initiation*.

Identité rythmique

Lorsque l’on doît saisir des fragments de musique parallèle qui ont le même rythme, on peut les combiner dans un contexte de voix unique et par voie de conséquence former des accords. Il suffit pour cela de les regrouper dans une construction de musique simultanée simple au sein d’une voix explicite :

```
\new Voice <<
  { e4 f8 d e16 f g8 d4 }
  { c4 d8 b c16 d e8 b4 }
>>
```



Prenez garde que les différents éléments doivent impérativement avoir la même structure rythmique, sous peine de ligature aléatoire et de messages d’avertissement.

Commandes prédéfinies

`\voiceOne`, `\voiceTwo`, `\voiceThree`, `\voiceFour`, `\oneVoice`.

Voir aussi

Manuel d’initiation : Section “Les voix contiennent la musique” dans *Manuel d’initiation*, Section “Instanciation explicite des voix” dans *Manuel d’initiation*.

Manuel de notation : [Portées de percussion], page 327, [Silences invisibles], page 49, [Hampes], page 188.

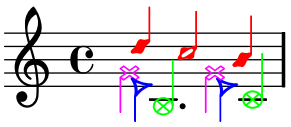
Morceaux choisis : Section “Notation simultanée” dans *Morceaux choisis*.

Styles de voix

Opter pour des couleurs et des têtes de notes spécifiques selon la voix permet de les identifier plus facilement :

```
<<
  { \voiceOneStyle d4 c2 b4 }
  \\
  { \voiceTwoStyle e,2 e }
  \\
  { \voiceThreeStyle b2. c4 }
  \\
  { \voiceFourStyle g'2 g }
```

```
>>
```



La commande `\voiceNeutralStyle` permet de revenir à la présentation normale.

Commandes prédéfinies

`\voiceOneStyle`, `\voiceTwoStyle`, `\voiceThreeStyle`, `\voiceFourStyle`, `\voiceNeutralStyle`.

Voir aussi
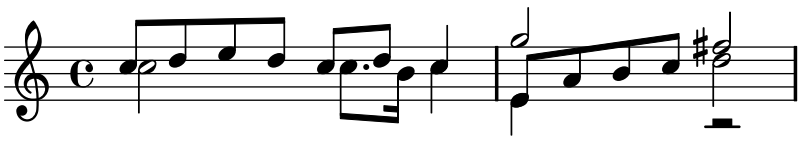
Manuel d'initiation : Section "J'entends des Voix" dans *Manuel d'initiation*, Section "Autres sources de documentation" dans *Manuel d'initiation*.

Morceaux choisis : Section "Notation simultanée" dans *Morceaux choisis*.

Résolution des collisions

Les notes de hauteur identique appartenant à des voix différentes, même si leur hampe sont opposées, verront leur tête automatiquement fusionnér. Les notes dont la tête diffère ou bien qui ont la hampe dans la même direction ne seront pas automatiquement fusionnées. Les silences, lorsqu'ils sont dans une autre voix et à l'opposé des hampes seront décalés verticalement. Vous constaterez, dans l'exemple suivant, que la fusion échoue aux premier et troisième temps de la première mesure, ainsi qu'au premier temps de la deuxième mesure.
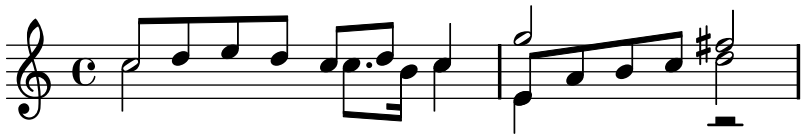
```
<<
  {
    c8 d e d c d c4
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



Cependant, vous pouvez fusionner une tête de blanche avec une tête de croche – jamais avec une noire. Les têtes du premier temps de la première mesure ont bien fusionné :
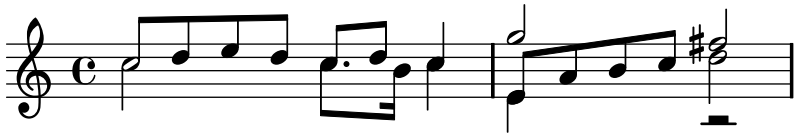
```
<<
  {
    \mergeDifferentlyHeadedOn
    c8 d e d c d c4
    g'2 fis
  } \\ {
```

```
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



De même, vous pouvez fusionner les têtes de notes pointées et non-pointées comme au troisième temps de la première mesure :
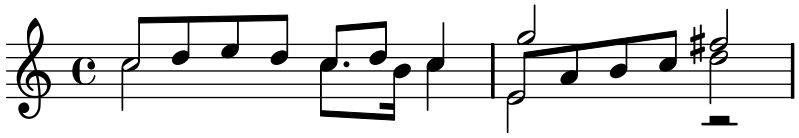
```
<<
  {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c8 d e d c d c4
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



Lorsque trois notes ou plus s'agglutinent dans un même empilement, `\mergeDifferentlyHeadedOn` ne peut mener à bien la fusion des deux notes qui devraient l'être. Pour obtenir une fusion optimale, appliquez un décalage (`\shift`) à la note qui ne devrait pas fusionner. Ici, on applique un `\shiftOn` pour décaler le *sol* de l'empilement ; le rendement de `\mergeDifferentlyHeadedOn` est alors comme il faut.

```
<<
  {
    \mergeDifferentlyHeadedOn
    \mergeDifferentlyDottedOn
    c8 d e d c d c4
    \shiftOn
    g'2 fis
  } \\ {
    c2 c8. b16 c4
    e,2 r
```

```
  } \\ {
    \oneVoice
    s1
    e8 a b c d2
  }
>>
```



Les commandes `\shiftOn`, `\shiftOnn`, et `\shiftOnnn` déterminent le niveau de décalage des accords de la voix en cours. Les voix externes – habituellement, les voix une et deux – sont affectées de `\shiftOff`, alors que les voix internes – trois et quatre – sont affectées de `\shiftOn`. `\shiftOnn` et `\shiftOnnn` sont des niveaux supplémentaires de décalage.

Les têtes de notes ne fusionneront que dans la mesure où leur hampe sont opposées – implicitement parce qu'appartenant aux voix une ou deux, ou bien explicitement.

Commandes prédéfinies

`\mergeDifferentlyDottedOn`, `\mergeDifferentlyDottedOff`, `\mergeDifferentlyHeadedOn`, `\mergeDifferentlyHeadedOff`.

`\shiftOn`, `\shiftOnn`, `\shiftOnnn`, `\shiftOff`.

Morceaux choisis
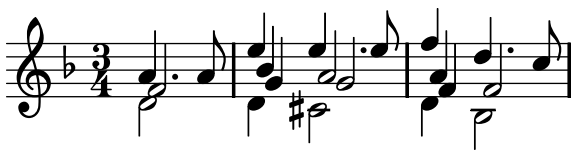
Ajout de voix pour éviter les collisions

Dans certains cas de musique polyphonie complexe, une voix supplémentaire peut permettre d'éviter les risques de collision. Lorsque quatre voix parallèles ne suffisent pas, la fonction Scheme `context-spec-music` permet d'ajouter des d'autres voix.

```
voiceFive = #(context-spec-music (make-voice-props-set 4) 'Voice)

\relative c'' {
  \time 3/4
  \key d \minor
  \partial 2
  <<
    {
      \voiceOne
      a4. a8
      e'4 e4. e8
      f4 d4. c8
    }
    \\
    {
      \voiceThree
      f,2
      bes4 a2
      a4 s2
    }
    \\
    {
```

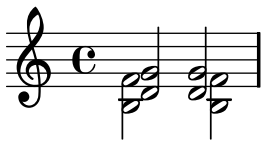
```
      \voiceFive
      s2
      g4 g2
      f4 f2
    }
    \\
    \bar "||"{
      \voiceTwo
      d2
      d4 cis2
      d4 bes2
    }
  >>
}
```



Décalage horizontal forcé

Quand LilyPond est dépassé, la propriété `force-hshift` de l'objet `NoteColumn`, et des silences à hauteur déterminée, peuvent s'avérer utiles pour dicter au programme les choix de placement. On travaille ici en espace de portée.

```
\relative c' <<
  {
    <d g>2 <d g>
  }
  \\
  {
    <b f'>2
    \once \override NoteColumn #'force-hshift = #1.7
    <b f'>2
  }
>>
```



Voir aussi

Glossaire musicologique : Section “polyphonie” dans *Glossaire*.

Manuel d'initiation : Section “Notes simultanées” dans *Manuel d'initiation*, Section “Les voix contiennent la musique” dans *Manuel d'initiation*, Section “Exemple concret” dans *Manuel d'initiation*.

Morceaux choisis : Section “Notation simultanée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “NoteColumn” dans *Référence des propriétés internes*, Section “NoteCollision” dans *Référence des propriétés internes*, Section “RestCollision” dans *Référence des propriétés internes*.

Regroupement automatique de parties

Le regroupement automatique de parties vous permet de fusionner deux pupitres sur une seule portée, ceci dans le but de créer des partitions d’orchestre. Lorsque les deux parties sont identiques sur une certaine durée, une seule s’affiche. Lorsqu’elles diffèrent, deux voix séparées apparaissent, avec des hampes dont la direction est gérée automatiquement. Vous pouvez aussi identifier et faire ressortir les solos et parties *a due*.

Voici la syntaxe qui permet de combiner des parties :
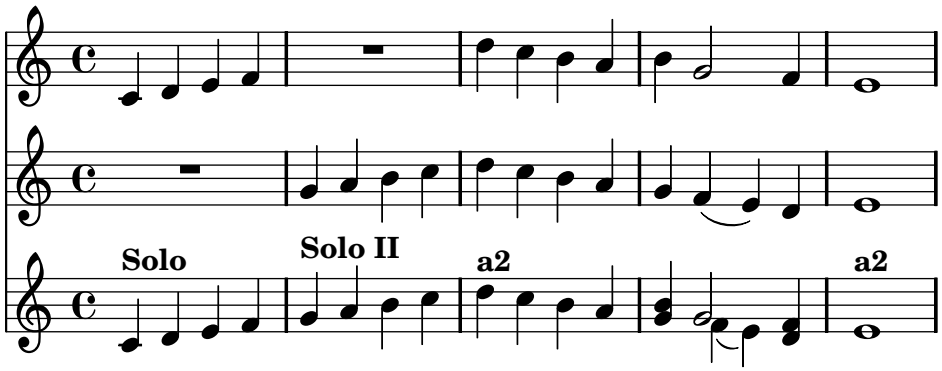
```
\partcombine expression_musicale_1 expression_musicale_2
```

L’exemple suivant illustre les fonctionnalités élémentaires du combinateur de parties : positionner les parties sur une portée, gérer la direction des hampes et de la polyphonie. Les identificateurs sont les mêmes pour la combinaison et les parties séparées.

```
instrumentOne = \relative c' {
  c4 d e f
  R1
  d'4 c b a
  b4 g2 f4
  e1
}

instrumentTwo = \relative g' {
  R1
  g4 a b c
  d c b a
  g f( e) d
  e1
}

<<
  \new Staff \instrumentOne
  \new Staff \instrumentTwo
  \new Staff \partcombine \instrumentOne \instrumentTwo
>>
```



Les notes de la troisième mesure n’apparaissent qu’une seule fois, alors qu’elles ont été spécifiés deux fois (une fois dans chacune des parties). La direction des hampes et des liaisons de tenue ou de phrasé est gérée automatiquement, selon qu’il s’agisse d’un solo ou d’un unisson. La première

partie, dont le contexte s'appellera `one`, aura toujours ses hampes dirigées vers le haut et sera notée « Solo », alors que la deuxième, appelée `two`, aura des hampes vers le bas et sera notée « Solo II ». Les parties à l'unisson seront estampillées d'un « a2 » par défaut.

LilyPond interprète dans un contexte `Voice` chacun des arguments fournis à `\partcombine`. Si vous travaillez avec des octaves relatives, spécifiez `\relative` dans chacune des expressions musicales, comme ceci :

```
\partcombine
  \relative ... expression_musicale_1
  \relative ... expression_musicale_2
```

Une section `\relative` à l'extérieur du `\partcombine` restera sans effet sur les hauteurs de *expression_musicale_1* ou de *expression_musicale_2*.

Morceaux choisis

Combinaison de deux parties sur une même portée

L'outil de combinaison de parties (la commande `\partcombine`) permet d'avoir deux parties différentes sur une même portée. LilyPond ajoute automatiquement des indications textuelles, telles que « solo » ou « a2 ». Si votre intention n'est que de fusionner les parties, sans ajouter de texte, assignez faux à la propriété `printPartCombineTexts`. Dans le cas de partitions vocales, et plus particulièrement d'hymnes, ces « solo/a2 » ne sont d'aucune utilité, aussi vaut-il mieux les désactiver. Dans le cas où il y aurait alternance entre *solo* et *tutti*, il vaut mieux faire appel à de la musique polyphonique standard.
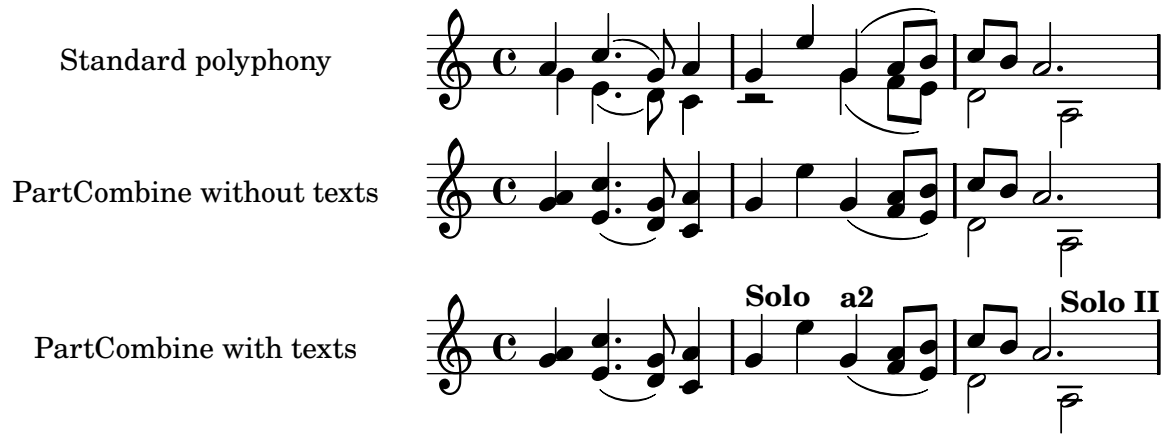
Voici trois moyens d'imprimer deux parties sur un même portée : en polyphonie normale, avec `\partcombine` sans indication supplémentaire, et avec `\partcombine` commentée.

```
musicUp = \relative c'' {
  \time 4/4
  a4 c4.( g8) a4 |
  g4 e' g,( a8 b) |
  c b a2.
}

musicDown = \relative c'' {
  g4 e4.( d8) c4 |
  r2 g'4( f8 e) |
  d2 \stemDown a
}

\score {
  <<
    <<
    \new Staff {
      \set Staff.instrumentName = #"Standard polyphony"
      << \musicUp \\ \musicDown >>
    }
    \new Staff \with { printPartCombineTexts = ##f } {
      \set Staff.instrumentName = #"PartCombine without texts"
      \partcombine \musicUp \musicDown
    }
    \new Staff {
      \set Staff.instrumentName = #"PartCombine with texts"
      \partcombine \musicUp \musicDown
```
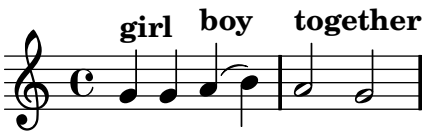
```
    }
    >>
  >>
  \layout {
    indent = 6.0\cm
    \context {
      \Score
      \override SystemStartBar #'collapse-height = #30
    }
  }
}
```



Modification des indications de parties combinées

Lorsque vous regroupez automatiquement des parties, vous pouvez modifier le texte qui sera affiché pour les solos et pour les parties à l'unisson :

```
\new Staff <<
  \set Staff.soloText = #"girl"
  \set Staff.soloIIText = #"boy"
  \set Staff.aDueText = #"together"
  \partcombine
    \relative c'' {
      g4 g r r
      a2 g
    }
    \relative c'' {
      r4 r a( b)
      a2 g
    }
>>
```



Voir aussi

Glossaire musicologique : Section “a due” dans *Glossaire*, Section “partie” dans *Glossaire*.

Manuel de notation : Section 1.6.3 [Écriture de parties séparées], page 171.

Morceaux choisis : Section “Notation simultanée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “PartCombineMusic” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

`\partcombine` ne prend en charge que deux voix.

Lorsque `printPartCombineTexts` est actif et que les deux voix jouent souvent les mêmes notes, le combinateur peut afficher `a2` plus d’une fois par mesure.

`\partcombine` ne peut s’inscrire dans un bloc `\times`.

`\partcombine` ne peut s’inscrire dans un bloc `\relative`.

En interne, `\partcombine` interprète les deux arguments en tant que `Voice`s, dénommées `one` et `two`, puis décide de quand les parties seront fusionnées. Par conséquent, si les arguments changent pour d’autres noms de contexte Section “Voice” dans *Référence des propriétés internes*, les événements qu’ils contiendraient seront ignorés. De la même manière, le combinateur n’est pas conçu pour travailler avec des paroles ; il s’arrête dès qu’il est explicitement fait appel à l’une des voix pour y attacher des paroles.

`\partcombine` n’examine que l’attaque des notes. Il n’est donc pas en mesure de déterminer si une note attaquée précédemment est encore jouée ou non, ce qui peut engendrer quelques problèmes.

Saisie de musique en parallèle
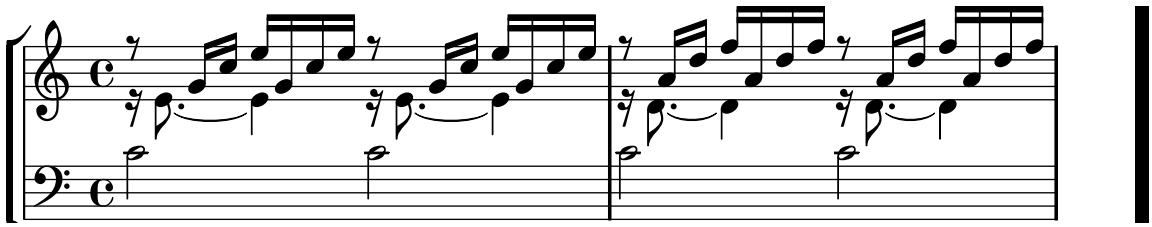
On peut écrire plusieurs voix de façon entremêlée. La fonction `\parallelMusic` prend en charge une liste des variables à créer, ainsi qu’une expression musicale. Le contenu des différentes mesures de l’expression musicale deviennent les valeurs des variables respectives que vous pourrez ensuite utiliser pour imprimer la partition.

> **Note :** Les contrôles de barre de mesure `|` sont obligatoires et le mesures doivent être de longueur identique.

```
\parallelMusic #'(voiceA voiceB voiceC) {
  % Bar 1
  r8 g'16 c'' e'' g' c'' e'' r8 g'16 c'' e'' g' c'' e'' |
  r16 e'8.~   e'4           r16 e'8.~   e'4           |
  c'2                       c'2                       |

  % Bar 2
  r8 a'16 d'' f'' a' d'' f'' r8 a'16 d'' f'' a' d'' f'' |
  r16 d'8.~   d'4           r16 d'8.~   d'4           |
  c'2                       c'2                       |

}
\new StaffGroup <<
  \new Staff << \voiceA \\ \voiceB >>
  \new Staff { \clef bass \voiceC }
>>
```
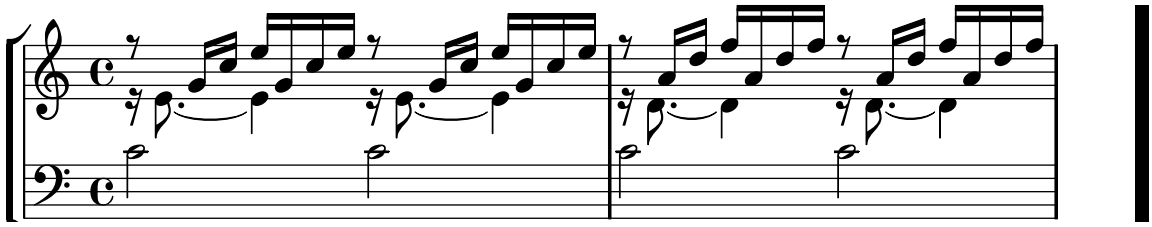


Vous pouvez travailler en mode relatif. Notez cependant que la commande `\relative` n'apparaît pas au sein du bloc `\parallelMusic`. Le calcul des hauteurs relatives s'effectue voix par voix, et non au fil des lignes saisies ; en d'autres termes, les notes de la `voiceA` ignorent tout de celles de la `voiceB`.

```
\parallelMusic #'(voiceA voiceB voiceC) {
  % Bar 1
  r8 g16 c e g, c e r8 g,16 c e g, c e  |
  r16 e8.~ e4       r16 e8.~  e4        |
  c2                c                   |

  % Bar 2
  r8 a,16 d f a, d f r8 a,16 d f a, d f |
  r16 d8.~  d4       r16 d8.~  d4       |
  c2                 c                  |

 }
\new StaffGroup <<
  \new Staff << \relative c'' \voiceA \\ \relative c' \voiceB >>
  \new Staff \relative c' { \clef bass \voiceC }
>>
```



Ceci fonctionne bien avec la musique pour piano. L'exemple suivant affecte quatre mesures à quatre variables :
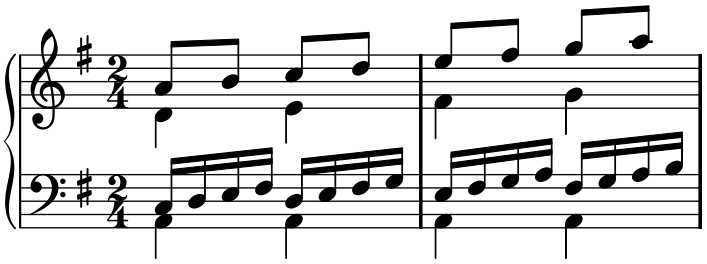
```
global = {
  \key g \major
  \time 2/4
}

\parallelMusic #'(voiceA voiceB voiceC voiceD) {
  % Bar 1
  a8    b     c   d     |
  d4          e         |
  c16 d e fis d e fis g  |
  a4          a         |

  % Bar 2
  e8      fis  g     a   |
  fis4         g         |
  e16 fis g  a fis g a b |
  a4           a         |
```

```
  % Bar 3 ...
}

\score {
  \new PianoStaff <<
     \new Staff {
       \global
       <<
         \relative c'' \voiceA
         \\
         \relative c'  \voiceB
       >>
     }
     \new Staff {
       \global \clef bass
       <<
         \relative c \voiceC
         \\
         \relative c \voiceD
       >>
     }
  >>
}
```
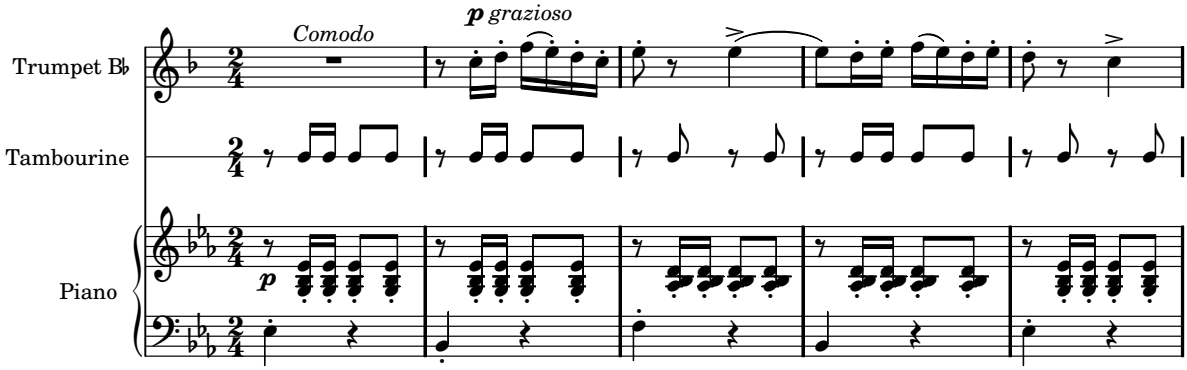


Voir aussi

Manuel d'initiation : Section "Organisation du code source avec des variables" dans *Manuel d'initiation*.

Morceaux choisis : Section "Notation simultanée" dans *Morceaux choisis*.

1.6 Notation sur la portée



Cette section aborde les détails de gravure de la portée, la réalisation de partitions comprenant plusieurs portées et l'ajout d'indications globales d'exécution, présentes sur certaines portées seulement.

1.6.1 Gravure des portées
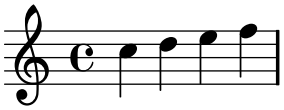
Nous allons voir ici comment créer des portées et comment les regrouper.

Initialisation de nouvelles portées

Les *portées* – en anglais *staff* (*staves* au pluriel) – sont créées à l'aide des commandes `\new` ou `\context`. Pour de plus amples détails, consultez Section 5.1.2 [Création d'un contexte], page 464.

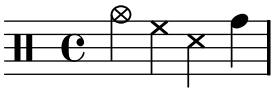
Le contexte de portée standard s'appelle `Staff` :

```
\new Staff { c4 d e f }
```



Le contexte `DrumStaff` crée une portée à cinq lignes correspondant à une batterie traditionnelle et chacun des instruments est représenté par un symbole spécifique. Les éléments sont saisis en mode batterie, initialisé par la commande `\drummode`, chaque composante étant spécifiée par son nom. Pour de plus amples détails, consultez [Portées de percussion], page 327.

```
\new DrumStaff {
  \drummode { cymc hh ss tomh }
}
```



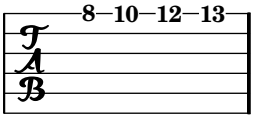
Un `RhythmicStaff` est composé d'une portée à ligne unique chargée de présenter les valeurs rythmiques saisies. Seules sont imprimées les durées. Pour de plus amples détails, consultez [Gravure de lignes rythmiques], page 68.

```
\new RhythmicStaff { c4 d e f }
```



Un `TabStaff` crée une portée de tablature correpondant aux six cordes d'une guitare standard. Pour de plus amples détails, consultez [Tablatures par défaut], page 296.
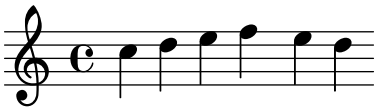
```
\new TabStaff { c4 d e f }
```



Lilypond dispose aussi de deux contextes dédiés à la musique ancienne : `MensuralStaff` et `VaticanaStaff`. Ils sont abordés plus en détails au chapitre Section 2.9.4 [Contextes prédéfinis], page 370.

Le contexte `GregorianTranscriptionStaff` permet d'obtenir des éditions modernes du grégorien. Bien entendu, il est dépourvu de barres de mesure.

```
\new GregorianTranscriptionStaff { c4 d e f e d }
```



Vous pourrez toujours définir d'autres contextes de portée selon vos besoins, en suivant les indications fournies au chapitre Section 5.1.6 [Définition de nouveaux contextes], page 467.

Voir aussi

Glossaire musicologique : Section "staff" dans *Glossaire*, Section "portées" dans *Glossaire*.

Manuel de notation : Section 5.1.2 [Création d'un contexte], page 464, [Portées de percussion], page 327, [Gravure de lignes rythmiques], page 68, [Tablatures par défaut], page 296, Section 2.9.4 [Contextes prédéfinis], page 370, [Symbole de la portée], page 162, [Contextes du chant grégorien], page 370, [Les contextes de la musique mensurale], page 370, Section 5.1.6 [Définition de nouveaux contextes], page 467.

Morceaux choisis : Section "Notation sur la portée" dans *Morceaux choisis*.

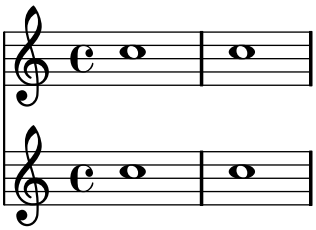
Référence des propriétés internes : Section "Staff" dans *Référence des propriétés internes*, Section "DrumStaff" dans *Référence des propriétés internes*, Section "GregorianTranscriptionStaff" dans *Référence des propriétés internes*, Section "RhythmicStaff" dans *Référence des propriétés internes*, Section "TabStaff" dans *Référence des propriétés internes*, Section "MensuralStaff" dans *Référence des propriétés internes*, Section "VaticanaStaff" dans *Référence des propriétés internes*, Section "StaffSymbol" dans *Référence des propriétés internes*.

Regroupement de portées

LilyPond dispose de différents contextes permettant de regrouper des portées individuelles et d'obtenir ainsi des « systèmes ». Chacun de ces contextes définira le style de regoupement, avec son signe particulier en début de portée et ses règles de gestion des barres de mesure.

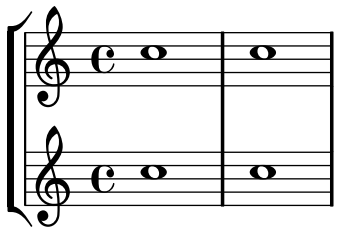
Lorsqu'aucun contexte particulier n'est spécifié, les propriétés suivantes s'appliqueront par défaut : les portées du groupe ne sont pas reliées, hormis par une simple ligne verticale en début de ligne, et les barres de mesure sont indépendantes.

```
<<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



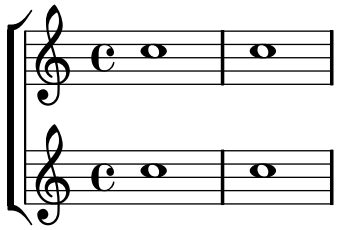
Dans un `StaffGroup`, le groupe de portées est signifié par un crochet, et les barres de mesure sont d'un seul tenant.

```
\new StaffGroup <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



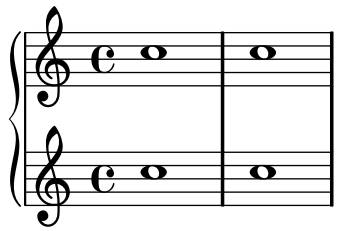
Dans un `ChoirStaff`, le groupe de portées est signifié par un crochet sur la gauche, et les barres de mesure sont individuelles.

```
\new ChoirStaff <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



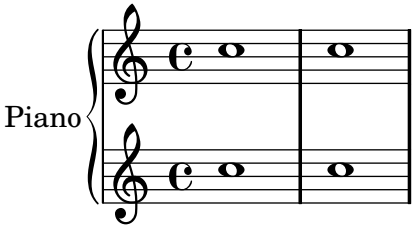
Dans un `GrandStaff`, le groupe de portées est signifié par une accolade sur la gauche, et les barres de mesure sont d'un seul tenant.

```
\new GrandStaff <<
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



Le contexte `PianoStaff` est indentique au `GrandStaff`, sauf qu'il gère automatiquement l'affichage du nom d'instrument – voir [Noms d'instrument], page 171 pour plus de détails.

```
\new PianoStaff <<
  \set PianoStaff.instrumentName = #"Piano"
  \new Staff { c1 c }
  \new Staff { c1 c }
>>
```



Les barres de mesure au début de chaque système adoptent l'un des styles `SystemStartBar`, `SystemStartBrace` ou `SystemStartBracket`. Dans chaque contexte, seul l'un de ces styles est

utilisé, et c'est la propriété `systemStartDelimiter` qui détermine lequel. Un quatrième style, `SystemStartSquare`, doit quant à lui être spécifié de manière explicite.

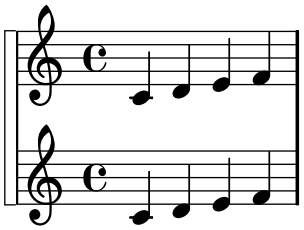
Vous pouvez aussi créer vos propres contextes de regroupement, comme l'explique Section 5.1.6 [Définition de nouveaux contextes], page 467.

Morceaux choisis

Indication de regroupement de portées par un rectangle

Un regroupement de portées sera indiqué par un simple rectangle – `SystemStartSquare` – en début de ligne dès lors que vous le mentionnerez explicitement au sein d'un contexte `StaffGroup` ou `ChoirStaffGroup`.

```
\score {
  \new StaffGroup { <<
  \set StaffGroup.systemStartDelimiter = #'SystemStartSquare
    \new Staff { c'4 d' e' f' }
    \new Staff { c'4 d' e' f' }
  >> }
}
```



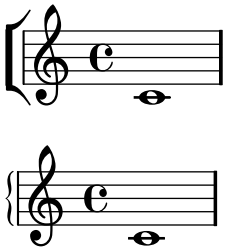
Indicateur de regroupement et portée unique

Lorsque, dans des regoupements de type `ChoirStaff` ou `StaffGroup`, une seule portée est active, aucune indication n'est donnée en début de ligne. Surcharger la propriété adéquate permet de modifier ce comportement par défaut.

Notez bien que dans le cas des `PianoStaff` et `GrandStaff`, pour lesquels le délimiteur de système est une accolade et non un crochet, il ne s'agit pas de la même propriété – voir le deuxième *système* de l'exemple.

```
\markup \left-column {
  \score {
    \new StaffGroup <<
      % Must be lower than the actual number of staff lines
      \override StaffGroup.SystemStartBracket #'collapse-height = #1
      \override Score.SystemStartBar #'collapse-height = #1
      \new Staff {
        c'1
      }
    >>
    \layout { }
  }
  \null
  \score {
    \new PianoStaff <<
      \override PianoStaff.SystemStartBrace #'collapse-height = #1
      \override Score.SystemStartBar #'collapse-height = #1
```

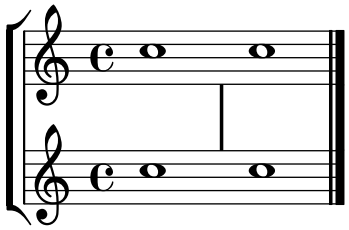
```
      \new Staff {
        c'1
      }
    >>
    \layout { }
  }
}
```



Présentation à l'ancienne (barres de mesure entre les portées)

En musique mensurale, les barres de mesure ne traversent pas les portées. Pour obtenir ce résultat avec un `StaffGroup` plutôt qu'en utilisant un `ChoirStaff`, il faudra rendre « transparentes » les portions de barre qui recouvrent les portées.

```
global = {
  \override Staff.BarLine #'transparent = ##t
  s1 s
  % the final bar line is not interrupted
  \revert Staff.BarLine #'transparent
  \bar "|."
}
\new StaffGroup \relative c'' {
  <<
    \new Staff { << \global { c1 c } >> }
    \new Staff { << \global { c c } >> }
  >>
}
```



Voir aussi

Glossaire musicologique : Section "accolade" dans *Glossaire*, Section "crochet" dans *Glossaire*, Section "système" dans *Glossaire*.

Manuel de notation : [Noms d'instrument], page 171, Section 5.1.6 [Définition de nouveaux contextes], page 467.

Morceaux choisis : Section "Notation sur la portée" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Staff" dans *Référence des propriétés internes*, Section "StaffGroup" dans *Référence des propriétés internes*, Section "ChoirStaff" dans

Référence des propriétés internes, Section “GrandStaff” dans *Référence des propriétés internes*, Section “PianoStaff” dans *Référence des propriétés internes*, Section “SystemStartBar” dans *Référence des propriétés internes*, Section “SystemStartBrace” dans *Référence des propriétés internes*, Section “SystemStartBracket” dans *Référence des propriétés internes*, Section “SystemStartSquare” dans *Référence des propriétés internes*.
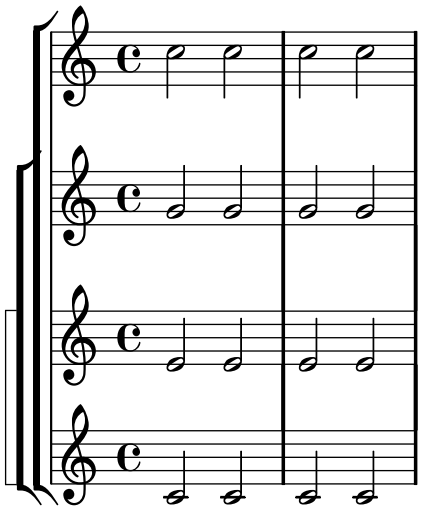
Problèmes connus et avertissements

Un `PianoStaff` ne peut contenir, par défaut, de `ChordNames`.

Imbrication de regroupements de portées

Les accolades et crochets qui délimitent les systèmes peuvent être imbriqués en profondeur. Chaque niveau inférieur aura son propre délimiteur, en plus de celui du niveau supérieur.

```
\new StaffGroup <<
  \new Staff { c2 c | c2 c }
  \new StaffGroup <<
    \new Staff { g2 g | g2 g }
    \new StaffGroup \with {
      systemStartDelimiter = #'SystemStartSquare
    }
    <<
      \new Staff { e2 e | e2 e }
      \new Staff { c2 c | c2 c }
    >>
  >>
>>
```



Vous pouvez aussi créer vos propres contextes d’imbrication, comme l’explique Section 5.1.6 [Définition de nouveaux contextes], page 467.

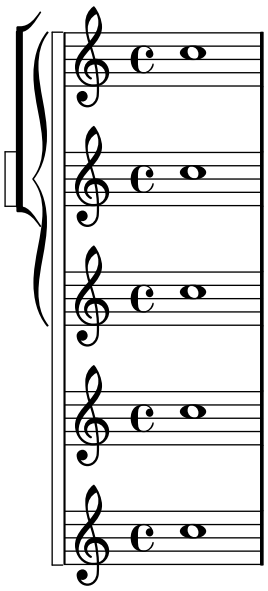
Morceaux choisis

Imbrications de regroupements de portées

La propriété `systemStartDelimiterHierarchy` permet de créer des regroupements imbriqués complexes. La commande `\set StaffGroup.systemStartDelimiterHierarchy` prend en argument la liste alphabétique des sous-groupes à hiérarchiser. Chaque sous-groupe peut être affublé d’un délimiteur particulier. Chacun des regroupements intermédiaires doit être borné par

des parenthèses. Bien que des éléments de la liste puissent être omis, le premier délimiteur embrassera toujours l'intégralité des portées. Vous disposez des quatre délimiteurs `SystemStartBar`, `SystemStartBracket`, `SystemStartBrace` et `SystemStartSquare`.

```
\new StaffGroup
\relative c'' <<
  \set StaffGroup.systemStartDelimiterHierarchy
    = #'(SystemStartSquare (SystemStartBrace (SystemStartBracket a
                              (SystemStartSquare b)  ) c ) d)
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
  \new Staff { c1 }
>>
```



Voir aussi

Manuel de notation : [Regroupement de portées], page 155, [Noms d'instrument], page 171, Section 5.1.6 [Définition de nouveaux contextes], page 467.

Morceaux choisis : Section "Notation sur la portée" dans *Morceaux choisis*.

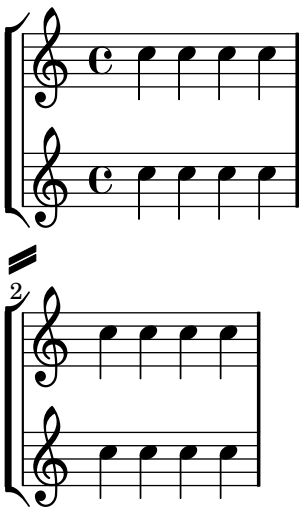
Référence des propriétés internes : Section "StaffGroup" dans *Référence des propriétés internes*, Section "ChoirStaff" dans *Référence des propriétés internes*, Section "SystemStartBar" dans *Référence des propriétés internes*, Section "SystemStartBrace" dans *Référence des propriétés internes*, Section "SystemStartBracket" dans *Référence des propriétés internes*, Section "SystemStartSquare" dans *Référence des propriétés internes*.

Séparation des systèmes

Le nombre de systèmes peut varier d'une page à l'autre ; vous povez, en pareil cas, rendre plus évidente la séparation entre les systèmes en l'indiquant visuellement. Ce « séparateur » est absent par défaut, mais vous pouvez l'activer par une option au sein du bloc `\paper`.

```
\book {
  \score {
```

```
    \new StaffGroup <<
      \new Staff {
        \relative c'' {
          c4 c c c
          \break
          c4 c c c
        }
      }
      \new Staff {
        \relative c'' {
          c4 c c c
          \break
          c4 c c c
        }
      }
    >>
  }
  \paper {
    system-separator-markup = \slashSeparator
    % following commands are needed only to format this documentation
    paper-width = 100\mm
    paper-height = 100\mm
    tagline = ##f
  }
}
```



Voir aussi

Manuel de notation : Section 4.1 [Mise en forme de la page], page 412.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

1.6.2 Modification de portées individuelles
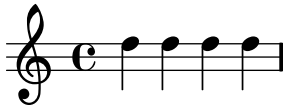
Cette section explique le réglage de la gravure de chaque portée, comme la taille de portée ou le nombre de lignes ; sont aussi décrits la suspension et la reprise de portées, ainsi que les portées d’*ossia*.

Symbole de la portée

Les notes, nuances, etc. sont regroupées dans un assemblage de lignes horizontales, que l'on nomme la portée (en anglais *staff*, et *staves* au pluriel). Dans LilyPond, ces lignes sont dessinées au moyen d'un objet graphique (*grob*) à part entière, nommé `StaffSymbol` – symbole de portée. Modifier les propriétés d'un `StaffSymbol` changera l'apparence de la portée, dès lors qu'elles auront été définies avant de créer la portée en question.
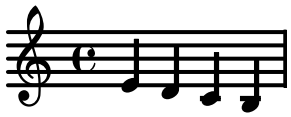
Vous pouvez modifier le nombre de lignes d'une portée. Néanmoins, la position de la clef et celle du do médium demanderont parfois un ajustement afin d'être en phase avec cette nouvelle portée. Pour plus d'explications, reportez-vous aux exemples du chapitre [Clefs], page 13.

```
\new Staff \with {
  \override StaffSymbol #'line-count = #3
}
{ d4 d d d }
```
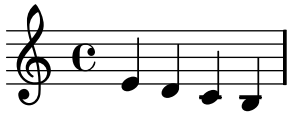


Lorsque vous modifierez l'épaisseur des lignes, gardez à l'esprit que les lignes supplémentaires et les hampes seront aussi modifiées. En effet, elles dépendent directement de l'épaisseur des lignes de la portée.

```
\new Staff \with {
  \override StaffSymbol #'thickness = #3
}
{ e4 d c b }
```



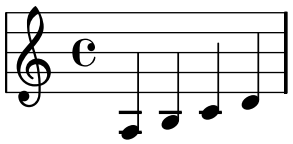
L'épaisseur des lignes supplémentaires (*ledger lines*) peut être déterminée indépendamment des lignes de la portée. Dans l'exemple suivant, les deux nombres constituent des coefficients multiplicateurs de l'épaisseur de la ligne. Ils sont fournis en argument pour déterminer l'épaisseur des lignes supplémentaires.

```
\new Staff \with {
  \override StaffSymbol #'ledger-line-thickness = #'(1 . 0.2)
}
{ e4 d c b }
```



Modifier l'équidistance des lignes de la portée affectera aussi les lignes supplémentaires.

```
\new Staff \with {
  \override StaffSymbol #'staff-space = #1.5
}
{ a4 b c d }
```



Vous trouverez de plus amples détails sur les propriétés du `StaffSymbol` au chapitre Section "staff-symbol-interface" dans *Référence des propriétés internes*.

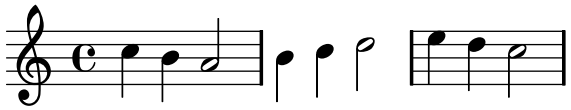
Les propriétés d'une portée peuvent être modifiées en cours de partition à l'aide des commandes `\stopStaff` et `\startStaff` :

```
c2 c
\stopStaff
\override Staff.StaffSymbol #'line-count = #2
\startStaff
b2 b
\stopStaff
\revert Staff.StaffSymbol #'line-count
\startStaff
a2 a
```



En règle générale, `\startStaff` et `\stopStaff` permettent d'entamer puis clôturer une portée n'importe où dans une partition.

```
c4 b a2
\stopStaff
b4 c d2
\startStaff
e4 d c2
```



Commandes prédéfinies
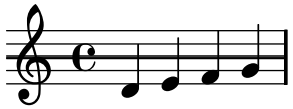
`\startStaff`, `\stopStaff`.

Morceaux choisis

Empâtement de certaines lignes d'une portée

Vous pourriez avoir envie, dans un but pédagogique, de rendre certaines lignes d'une portée plus épaisses que les autres, comme la ligne médiane ou bien pour mettre en exergue la ligne portant la clé de sol. Il suffit pour cela d'ajouter une ligne qui sera accolée à celle qui doît être mise en évidence, grâce à la propriété `line-positions` de l'objet `StaffSymbol`.

```
{
  \override Staff.StaffSymbol #'line-positions = #'(-4 -2 -0.2 0 0.2 2 4)
  d'4 e' f' g'
}
```



Voir aussi

Glossaire musicologique : Section “ligne” dans *Glossaire*, Section “ligne suplémentaire” dans *Glossaire*, Section “portée” dans *Glossaire*.

Manuel de notation : [Clefs], page 13.
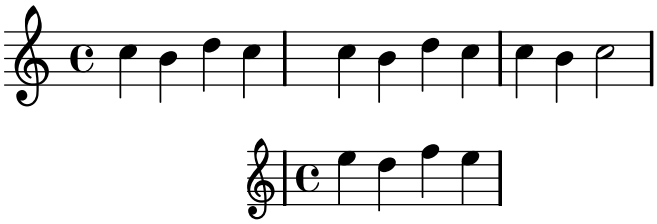
Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “StaffSymbol” dans *Référence des propriétés internes*, Section “staff-symbol-interface” dans *Référence des propriétés internes*.

Portées d’ossia

Une portée d’*ossia* – ou de variante – s’obtient en créant, à l’endroit approprié, une nouvelle portée simultanée :

```
\new Staff \relative c'' {
  c4 b d c
  <<
    { c4 b d c }
    \new Staff { e4 d f e }
  >>
  c4 b c2
}
```



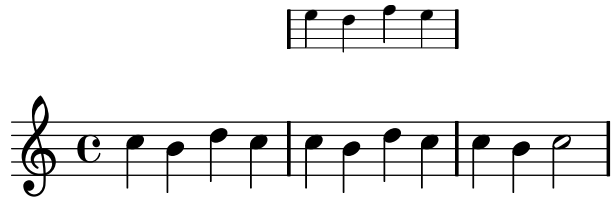
L’exemple ci-dessus n’est probablement pas ce qui vous conviendra le plus. Afin que cette ossia se place au-dessus de la portée à laquelle elle se réfère, étant par ailleurs dépourvue de métrique et de clef, et d’une taille légèrement inférieure, vous devrez avoir recours à quelques retouches. Le manuel d’initiation aborde une technique particulière pour obternir ce résultat au chapitre Section “Expressions musicales imbriquées” dans *Manuel d’initiation*.

L’exemple qui suit utilise, pour aligner la portée d’ossia, la propriété `alignAboveContext`. Cette méthode est tout à fait appropriée lorsqu’il y a un nombre restreint d’ossias.

```
\new Staff = main \relative c'' {
  c4 b d c
  <<
    { c4 b d c }

    \new Staff \with {
      \remove "Time_signature_engraver"
      alignAboveContext = #"main"
      fontSize = #-3
      \override StaffSymbol #'staff-space = #(magstep -3)
      \override StaffSymbol #'thickness = #(magstep -3)
      firstClef = ##f
    }
    { e4 d f e }
  >>
  c4 b c2
```
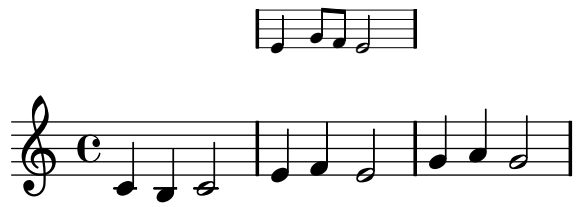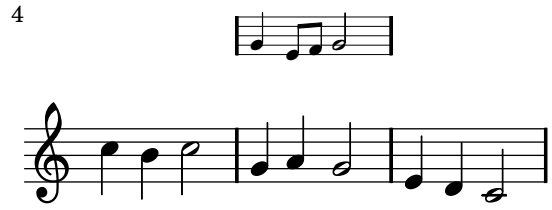
```
}
```



Dans le cas où de nombreux et courts fragments d'ossia affecteraient une même portée, il est judicieux de créer un contexte `Staff` vide auquel sera attribué un *identificateur*. Il suffira alors, pour entamer un fragment d'ossia, de *faire appel* à ce contexte grâce aux commandes `\startStaff` et `\stopStaff`. Vous vous rendrez compte à l'utilisation des avantages que procure cette façon de procéder, bien plus que dans l'exemple suivant.

```
<<
  \new Staff = ossia \with {
    \remove "Time_signature_engraver"
    \override Clef #'transparent = ##t
    fontSize = #-3
    \override StaffSymbol #'staff-space = #(magstep -3)
    \override StaffSymbol #'thickness = #(magstep -3)
  }
  { \stopStaff s1*6 }

  \new Staff \relative c' {
    c4 b c2
    <<
      { e4 f e2 }
      \context Staff = ossia {
        \startStaff e4 g8 f e2 \stopStaff
      }
    >>
    g4 a g2 \break
    c4 b c2
    <<
      { g4 a g2 }
      \context Staff = ossia {
        \startStaff g4 e8 f g2 \stopStaff
      }
    >>
    e4 d c2
  }
>>
```
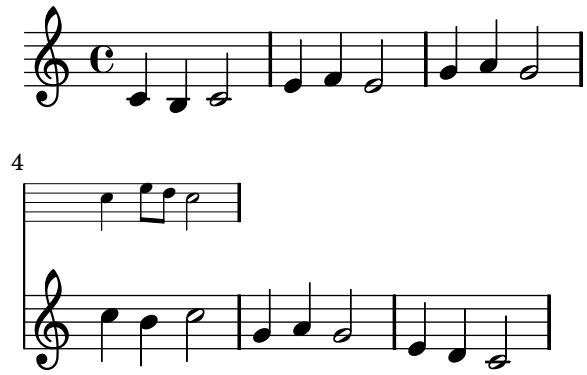




Vous pourriez aussi recourir à la commande `\Staff \RemoveEmptyStaves` pour créer votre portée d'ossia. Cependant, cette méthode reste limitée à l'apparition de ces ossias en début de ligne. Pour plus d'information au sujet de la commande `\Staff \RemoveEmptyStaves`, reportez-vous au chapitre [Masquage de portées], page 168.

```
<<
  \new Staff = ossia \with {
    \remove "Time_signature_engraver"
    \override Clef #'transparent = ##t
    fontSize = #-3
    \override StaffSymbol #'staff-space = #(magstep -3)
    \override StaffSymbol #'thickness = #(magstep -3)
  } \relative c'' {
    R1*3
    c4 e8 d c2
  }
  \new Staff \relative c' {
    c4 b c2
    e4 f e2
    g4 a g2 \break
    c4 b c2
    g4 a g2
    e4 d c2
  }
>>

\layout {
  \context {
    \Staff \RemoveEmptyStaves
    \override VerticalAxisGroup #'remove-first = ##t
  }
}
```
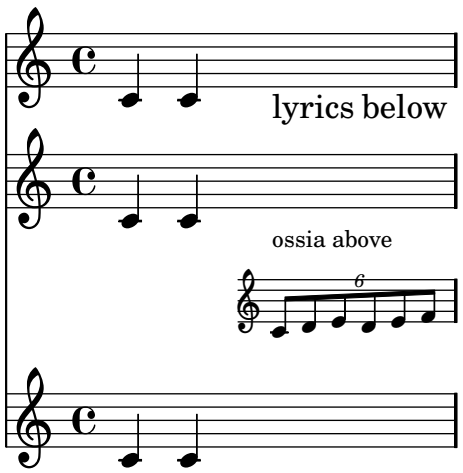


Morceaux choisis

Positionnement d'une ossia et des paroles

Cet exemple illustre la manière de positionner une portée d'ossia et des paroles à l'aide des propriétés de contexte `alignBelowContext` et `alignAboveContext`.

```
\paper {
  ragged-right = ##t
}

\relative c' <<
  \new Staff = "1" { c4 c s2 }
  \new Staff = "2" { c4 c s2 }
  \new Staff = "3" { c4 c s2 }
  { \skip 2
    <<
      \lyrics {
        \set alignBelowContext = #"1"
        lyrics4 below
      }
      \new Staff \with {
        alignAboveContext = #"3"
        fontSize = #-2
        \override StaffSymbol #'staff-space = #(magstep -2)
        \remove "Time_signature_engraver"
      } {
        \times 4/6 {
          \override TextScript #'padding = #3
          c8[^"ossia above" d e d e f]
        }
      }
    >>
  }
>>
```



Voir aussi

Glossaire musicologique : Section “ossia” dans *Glossaire*, Section “portée” dans *Glossaire*, Section “Frenched staff” dans *Glossaire*.

Manuel d’initiation : Section “Expressions musicales imbriquées” dans *Manuel d’initiation*, Section “Taille des objets” dans *Manuel d’initiation*, Section “Longueur et épaisseur des objets” dans *Manuel d’initiation*.

Manuel de notation : [Masquage de portées], page 168.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “StaffSymbol” dans *Référence des propriétés internes*.

Masquage de portées

Désactiver le graveur `Staff_symbol_engraver` dans un contexte `Staff` permet de masquer des lignes. La commande `\stopStaff` aura le même effet.

```
\new Staff \with {
  \remove "Staff_symbol_engraver"
}
\relative c''' { a8 f e16 d c b a2 }
```

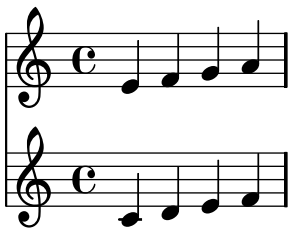


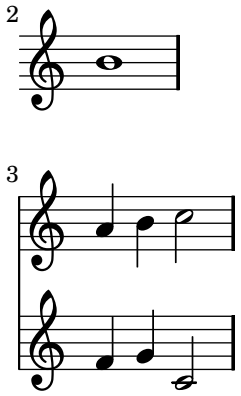
L’instruction `\Staff \RemoveEmptyStaves` placée dans un bloc `\layout` aura pour effet de masquer toute portée qui ne contient rien. Dans les partitions d’orchestre, les portées qui n’ont que des silences sont habituellement masquées afin de gagner de la place. Ce style d’édition s’appelle en anglais « French Score ». Cette fonctionnalité masque et supprime toutes les portées vides d’une partition, hormis celles du premier système.

> **Note :** Une portée est considérée comme vide dès lors qu’elle ne contient que des silences multimesures, des silences visibles ou invisibles (ou d’espacement – les *\skip*) ou bien une combinaison de ces éléments.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
  }
}

\relative c' <<
  \new Staff {
    e4 f g a \break
    b1 \break
    a4 b c2
  }
  \new Staff {
    c,4 d e f \break
    R1 \break
    f4 g c,2
  }
>>
```





`\Staff \RemoveEmptyStaves` permet aussi de gérer des fragments d'ossia attachés à une portée. Pour plus de détails, voir [Portées d'ossia], page 164.

La commande `\VaticanaStaff \RemoveEmptyStaves` permet de masquer des portées vides dans les contextes de musique ancienne. Pour des contextes `RhythmicStaff`, il faudra utiliser `\RhythmicStaff \RemoveEmptyStaves`.

Commandes prédéfinies

`\Staff \RemoveEmptyStaves`, `\VaticanaStaff \RemoveEmptyStaves`, `\RhythmicStaff \RemoveEmptyStaves`.

Morceaux choisis

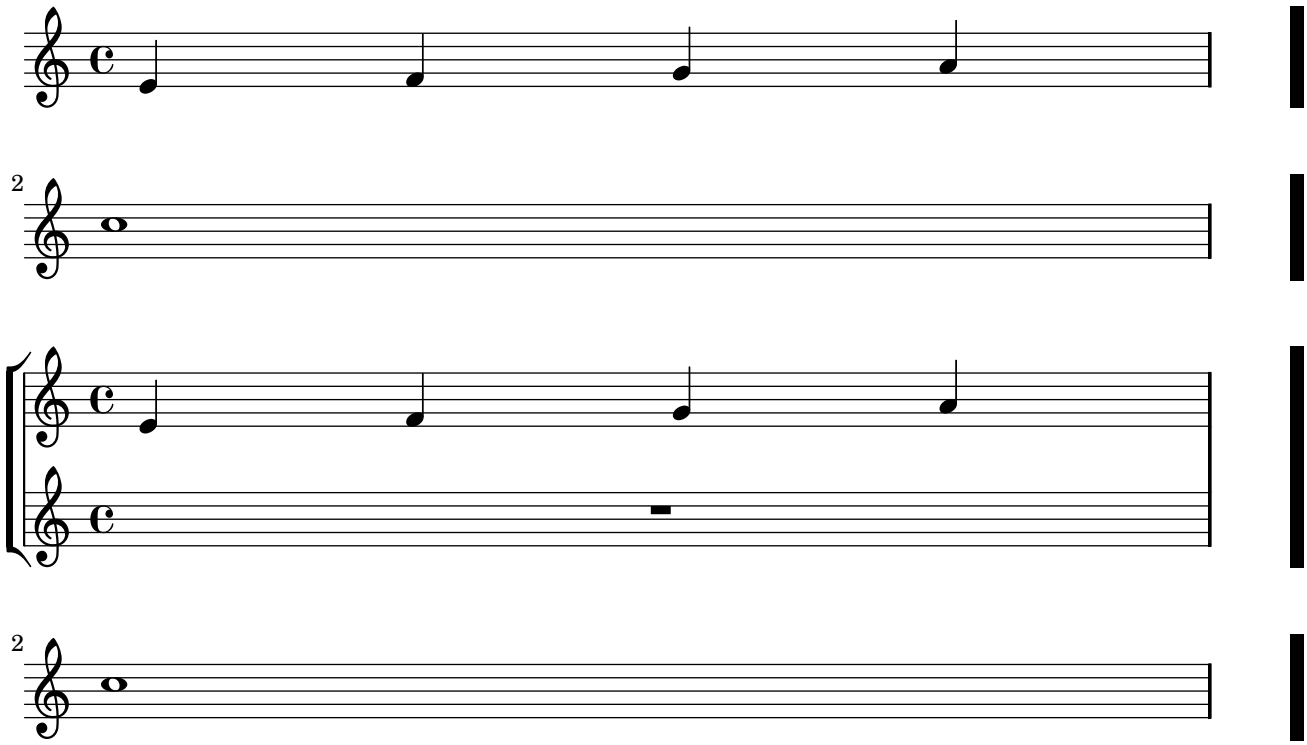
Masquage de la première ligne si elle est vide

Par défaut, le premier système comportera absolument toutes les portées. Si vous voulez masquer les portées vides y compris pour le premier système, vous devrez activer la propriété `remove-first` du `VerticalAxisGroup`. Mentionnée dans un bloc `\layout`, cette commande agira de manière globale. Pour qu'elle ne soit effective que pour une portée individuelle, vous devrez également spécifier le contexte (`Staff` pour qu'il ne concerne que la portée en cours) en préfixe de la propriété.

La première ligne inférieure du deuxième `StaffGroup` est bien présente, pour la simple raison que le réglage en question ne s'applique qu'à la portée dans laquelle il a été inscrit.

```
\layout {
  \context {
    \Staff \RemoveEmptyStaves
    % To use the setting globally, uncomment the following line:
    % \override VerticalAxisGroup #'remove-first = ##t
  }
}
\new StaffGroup <<
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    % To use the setting globally, comment this line,
    % uncomment the line in the \layout block above
    \override Staff.VerticalAxisGroup #'remove-first = ##t
    R1 \break
    R
  }
>>
\new StaffGroup <<
```

```
  \new Staff \relative c' {
    e4 f g a \break
    c1
  }
  \new Staff {
    R1 \break
    R
  }
>>
```



Voir aussi

Glossaire musicologique : Section “Frenched staff” dans *Glossaire*.

Manuel d’initiation: Section “Visibilité et couleur des objets” dans *Manuel d’initiation*.

Manuel de notation : [Symbole de la portée], page 162, [Portées d’ossia], page 164, [Dictée à trous], page 186, [Silences invisibles], page 49, Section 5.4.7 [Visibilité des objets], page 475.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “ChordNames” dans *Référence des propriétés internes*, Section “FiguredBass” dans *Référence des propriétés internes*, Section “Lyrics” dans *Référence des propriétés internes*, Section “Staff” dans *Référence des propriétés internes*, Section “VerticalAxisGroup” dans *Référence des propriétés internes*, Section “Staff_symbol_engraver” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Supprimer le `Staff_symbol_engraver` aura pour effet de masquer les barres de mesure. Forcer leur visibilité peut entraîner des problèmes de formattage. En pareil cas il vaut mieux, au lieu de supprimer le graveur, recourrir aux dérogations suivantes :

```
\override StaffSymbol #'stencil = ##f
\override NoteHead #'no-ledgers = ##t
```

En ce qui concerne les désagréments et messages liés à l'utilisation de l'instruction `\Staff \RemoveEmptyStaves`, consultez Section 5.1.5 [Modification des réglages par défaut d'un contexte], page 466.
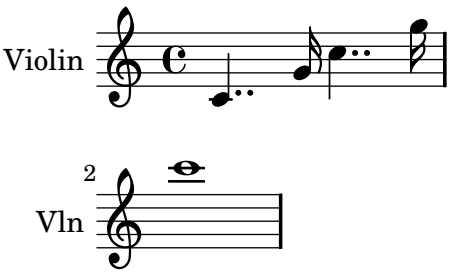
1.6.3 Écriture de parties séparées

Nous allons voir au fil des lignes qui suivent, comment insérer des indications de tempo ou des noms d'instrument dans une partition. Nous aborderons aussi la citation d'autres voix, et comment la mettre en forme.

Noms d'instrument
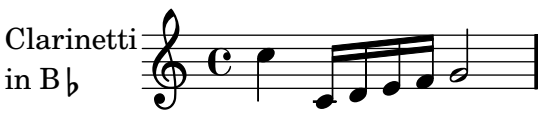
Dans un conducteur, les noms d'instrument sont portés en regard de chacune des portées, qu'il s'agisse d'un contexte `Staff`, `PianoStaff`, `StaffGroup`, `GrandStaff` ou `ChoirStaff` . La première ligne affichera la valeur de `instrumentName`, et les suivantes celle de `shortInstrumentName`.

```
\set Staff.instrumentName = #"Violin "
\set Staff.shortInstrumentName = #"Vln "
c4.. g'16 c4.. g'16
\break
c1
```



Le recours à la commande `\markup` permet de construire des noms d'instruments particuliers, tels que
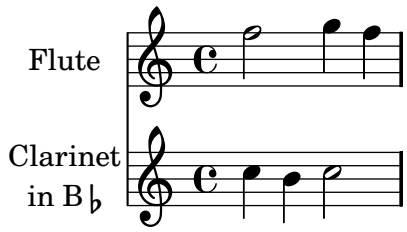
```
\set Staff.instrumentName = \markup {
  \column { "Clarinetti"
            \line { "in B" \smaller \flat } } }
c4 c,16 d e f g2
```



Lorsque plusieurs contextes de portée sont regroupés, les noms d'instrument, que ce soit sous leur forme développée ou abrégée, sont par défaut centrés. Si l'un d'entre eux est libellé sur plusieurs lignes, il faudra recourrir à l'instruction `\center-column` :

```
<<
  \new Staff {
    \set Staff.instrumentName = #"Flute"
    f2 g4 f
  }
  \new Staff {
```
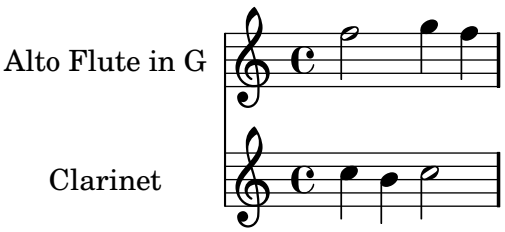
```
    \set Staff.instrumentName = \markup \center-column {
      Clarinet
      \line { "in B" \smaller \flat }
    }
    c4 b c2
  }
>>
```
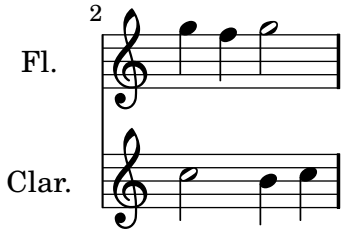


Lorsque le nom d'un instrument est relativement long, il est judicieux d'augmenter les retraits – *indent* – au sein du bloc `\layout` à l'aide des commandes `indent` et `short-indent`. Pour plus de plus amples détails sur ces réglages, reportez-vous au chapitre [Variables d'indentation et de décalage], page 419.

```
\layout {
  indent = 3.0\cm
  short-indent = 1.5\cm
}

\relative c'' <<
  \new Staff {
    \set Staff.instrumentName = #"Alto Flute in G"
    \set Staff.shortInstrumentName = #"Fl."
    f2 g4 f \break
    g4 f g2
  }
  \new Staff {
    \set Staff.instrumentName = #"Clarinet"
    \set Staff.shortInstrumentName = #"Clar."
    c,4 b c2 \break
    c2 b4 c
  }
>>
```

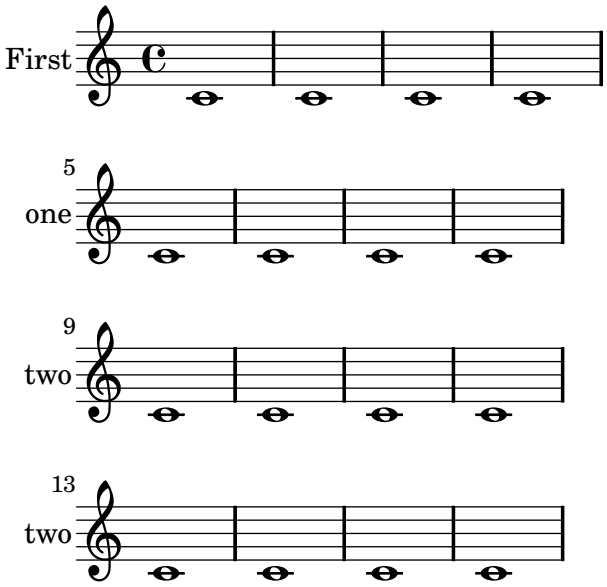




Des noms d'instruments peuvent s'utiliser dans d'autres contextes, tels que `ChordNames` ou `FiguredBass`, dès lors qu'on leur adjoint le graveur `Instrument_name_engraver`. Pour de plus amples informations sur la manière d'activer ou désactiver un graveur, voir Section 5.1.4 [Modification des greffons de contexte], page 464.

Vous pouvez changer le nom d'un instrument en cours de morceau. Notez cependant que la valeur de `instrumentName` ne s'affichera que sur la première portée :

```
\set Staff.instrumentName = #"First"
\set Staff.shortInstrumentName = #"one"
c1 c c c \break
c1 c c c \break
\set Staff.instrumentName = #"Second"
\set Staff.shortInstrumentName = #"two"
c1 c c c \break
c1 c c c \break
```

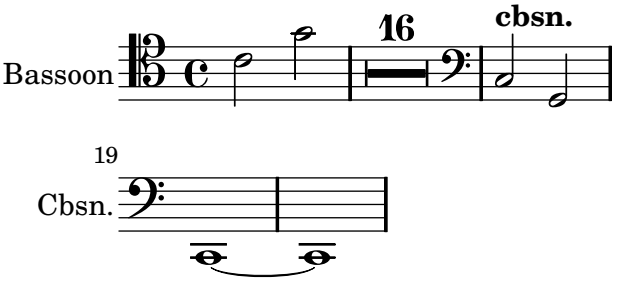


Lorsqu'un musicien doît changer d'instrument – piccolo et flûte, basson et contrebasson – , la commande `\addInstrumentDefinition`, couplée à l'instruction `\instrumentSwitch` permet de spécifier en détail les modifications intervenant lors du changement. La commande `\addInstrumentDefinition` prend deux arguments : une chaîne de caractères qui servira d'identificateur, et une liste d'associations de valeurs aux propriétés de ce nouvel instrument. Ces définitions devront être déclarées avant tout autre élément musical. L'instruction `\instrumentSwitch` se placera dans la musique au moment de la subtitution :

```
\addInstrumentDefinition #"contrabassoon"
  #`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
     (shortInstrumentName . "Cbsn.")
     (clefGlyph . "clefs.F")
     (middleCPosition . 6)
     (clefPosition . 2)
     (instrumentCueName . ,(make-bold-markup "cbsn."))
```

```
      (midiInstrument . "bassoon"))

\new Staff \with {
  instrumentName = #"Bassoon"
}
\relative c' {
  \clef tenor
  \compressFullBarRests
  c2 g'
  R1*16
  \instrumentSwitch "contrabassoon"
  c,,2 g \break
  c,1 ~ | c1
}
```



Voir aussi

Manuel de notation : [Variables d'indentation et de décalage], page 419, Section 5.1.4 [Modification des greffons de contexte], page 464.

Morceaux choisis : Section "Notation sur la portée" dans *Morceaux choisis*.

Référence des propriétés internes : Section "InstrumentName" dans *Référence des propriétés internes*, Section "PianoStaff" dans *Référence des propriétés internes*, Section "Staff" dans *Référence des propriétés internes*.

Citation d'autres voix

Il est assez courant qu'une voix soit doublée par une autre. Par exemple, les premiers et seconds violons peuvent jouer les mêmes notes durant un moment. LilyPond gère parfaitement ces situations où une voix est la réplique d'une autre, et vous évite de ressaisir la musique en question.
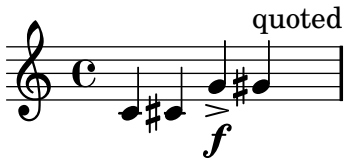
Avant qu'une partie ne puisse être mentionnée ailleurs, elle doit être considérée comme reproductible. C'est le but de l'instruction `\addQuote` qui prend en argument une chaîne d'identification et une expression musicale. Elle se place au niveau le plus haut, c'est à dire en dehors de tout bloc de musique :

```
flute = \relative c'' {
  a4 gis g gis
}
\addQuote "flute" { \flute }
```

Au cours d'une partie, des extraits de répliques peuvent être cités en utilisant la commande `\quoteDuring`. Cette commande prend deux arguments : le nom de la voix reproduite, tel que défini par `\addQuote`, et une expression musicale qui indique la durée de cette citation – silences invisibles ou multimesures. Viendra alors s'insérer dans l'expression musicale le fragment correspondant de la voix originelle, avec tous ses attributs (articulations, nuances, annotations, etc.).

```
flute = \relative c'' {
  a4 gis g->\f gis^\markup{quoted}
}
\addQuote "flute" { \flute }
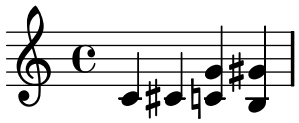
\relative c' {
  c4 cis \quoteDuring #"flute" { s2 }
}
```



Si l'expression musicale utilisée pour `\quoteDuring` contenait autre chose que du silence, il en résulterait une situation polyphonique, ce qui n'est pas le but recherché :

```
flute = \relative c'' {
  a4 gis g gis
}
\addQuote "flute" { \flute }
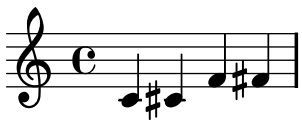
\relative c' {
  c4 cis \quoteDuring #"flute" { c4 b }
}
```



Les citations tiennent compte des transpositions, aussi bien celle de l'instrument d'origine que celle de la partie où elle intervient, dans la mesure où elles sont spécifiées par la commande `\transposition`. Voir [Instruments transpositeurs], page 19 pour plus de détails.

```
clarinet = \relative c'' {
  \transposition bes
  a4 gis g gis
}
\addQuote "clarinet" { \clarinet }

\relative c' {
  c4 cis \quoteDuring #"clarinet" { s2 }
}
```
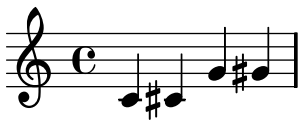


Les citations peuvent être « balisées » par un nom particulier afin de les utiliser de différentes manières. Pour de plus amples détails à ce propos, consultez le chapitre [Utilisation de balises], page 396.

La propriété `quotedEventTypes` permet de définir précisément quels éléments de la voix originelle seront reproduits. Sa valeur par défaut est `#'(StreamEvent)` ; autrement dit, tout

sera recopié. Lui affecter la valeur `#'(note-event rest-event tie-event)` fera que LilyPond reproduira les notes, silences et liaisons de prolongation, mais pas les articulations, annotations ni nuances.

```
clarinet = \relative c'' {
  a4 gis g->\f gis^\markup{quoted}
}
\addQuote "clarinet" { \clarinet }

\relative c' {
  \set Score.quotedEventTypes = #'(note-event rest-event tie-event)
  c4 cis \quoteDuring #"clarinet" { s2 }
}
```



Morceaux choisis

Citation d'une autre voix et transposition

Les citations tiennent compte de la transposition, aussi bien celle de l'instrument d'origine que celle de la partie où elle intervient. Dans l'exemple suivant, tous les instruments sont en tonalité de concert et seront repris par un instrument en fa. Le destinataire de la citation peut à son tour transposer à l'aide de la commande `\transpose`. En pareil cas, toutes les hauteurs, y compris celle de la citation, seront transposées.

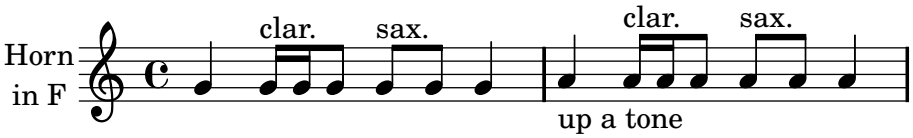
```
\addQuote clarinet {
  \transposition bes
  \repeat unfold 8 { d'16 d' d'8 }
}

\addQuote sax {
  \transposition es'
  \repeat unfold 16 { a8 }
}

quoteTest = {
  % french horn
  \transposition f
  g'4
  << \quoteDuring #"clarinet" { \skip 4 } s4^"clar." >>
  << \quoteDuring #"sax" { \skip 4 } s4^"sax." >>
  g'4
}

{
  \set Staff.instrumentName =
    \markup {
      \center-column { Horn \line { in F } }
    }
  \quoteTest
  \transpose c' d' << \quoteTest s4_"up a tone" >>
```

```
}
```



Citation d'une autre voix
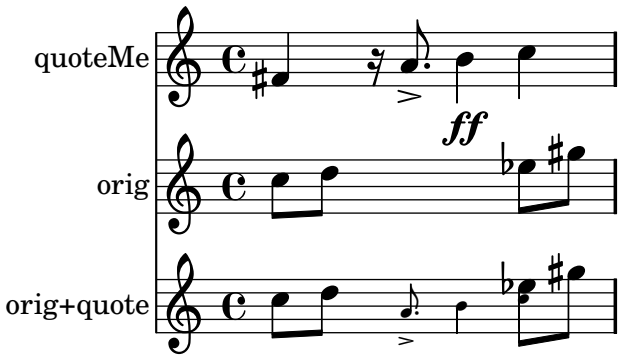
Les types d'événement pris en charge pour la citation peuvent se régler avec la propriété `quotedEventTypes`. Par défaut, sa valeur est fixée à `(note-event rest-event tie-event beam-event tuplet-span-event)`, ce qui signifie que seuls les notes, silences, liaisons, ligatures et nolets seront mentionnés par `\quoteDuring`. Dans l'exemple suivant, le quart de soupir n'est pas reproduit puisqu'il n'est pas mentionné parmi les `quotedEventTypes`.

Pour connaître la liste des types d'événements, reportez-vous au chapitre *Music classes* de la référence des propriétés internes.

```
quoteMe = \relative c' {
  fis4 r16 a8.-> b4\ff c
}
\addQuote quoteMe \quoteMe

original = \relative c'' {
  c8 d s2
  \once \override NoteColumn #'ignore-collision = ##t
  es8 gis8
}

<<
  \new Staff {
    \set Staff.instrumentName = #"quoteMe"
    \quoteMe
  }
  \new Staff {
    \set Staff.instrumentName = #"orig"
    \original
  }
  \new Staff \relative c'' <<
    \set Staff.instrumentName = #"orig+quote"
    \set Staff.quotedEventTypes =
      #'(note-event articulation-event)
    \original
    \new Voice {
      s4
      \set fontSize = #-4
      \override Stem #'length-fraction = #(magstep -4)
      \quoteDuring #"quoteMe" { \skip 2. }
    }
  >>
>>
```



Voir aussi

Manuel de notation : [Instruments transpositeurs], page 19, [Utilisation de balises], page 396.

Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “QuoteMusic” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Seul le contenu de la première `Voice` rencontrée dans la partie marquée d'une commande `\addQuote` pourra être retenu. Par voie de conséquence, ***`musique`*** ne saurait comprendre de `\new` ou une instance `\context Voice` qui la ferait passer à une autre voix.

Citer des notes d'ornement ne fonctionne pas, et peut même entraîner un blocage de LilyPond.

Citer des triolets imbriqués peut entraîner un résultat de piètre qualité.

Dans les versions précédentes de LilyPond (avant 2.11), `addQuote` était écrit entièrement en minuscules : `\addquote`.

Mise en forme d'une citation

La section précédente indiquait comment insérer des notes d'une autre voix. Nous allons maintenant voir une fonction musicale avancée, `\cueDuring`, qui facilite le formatage des petites notes.
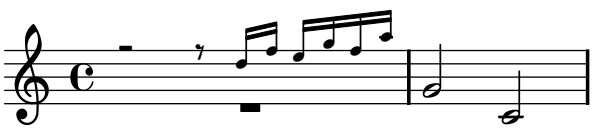
Sa syntaxe est :

```
\cueDuring #origine #voix musique
```

Des mesures issues de la partie d'***`origine`*** seront recopiées dans un contexte de `CueVoice`, créé implicitement, et synchronisées avec ***`musique`*** – habituellement un silence. L'apparition des petites notes initialise une polyphonie temporaire pour la portée concernée. L'argument ***`voix`*** détermine si ces petites notes seront attachées à la première ou à la seconde voix – `UP` pour la première ou `DOWN` pour la seconde.

```
oboe = \relative c'' {
  r2 r8 d16 f e g f a
  g8 g16 g g2.
}
\addQuote "oboe" { \oboe }

\new Voice \relative c'' {
  \cueDuring #"oboe" #UP { R1 }
  g2 c,
}
```
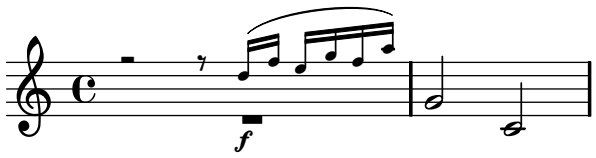


Dans cet exemple, il était nécessaire de déclarer explicitement le contexte `Voice`, sinon l'intégralité de l'expression musicale se serait retrouvée dans le contexte `CueVoice`.

La propriété `quotedCueEventTypes` permet de définir précisément quels éléments de la voix originelle seront reproduits. Sa valeur par défaut est `#'(note-event rest-event tie-event beam-event tuplet-span-event)`. LilyPond reproduira donc les notes, silences, liaisons de prolongation, ligatures et nolets, mais pas les articulations, annotations ni nuances.

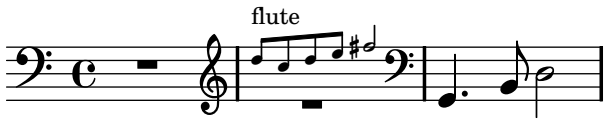
```
oboe = \relative c'' {
  r2 r8 d16(\f f e g f a)
  g8 g16 g g2.
}
\addQuote "oboe" { \oboe }

\new Voice \relative c'' {
  \set Score.quotedCueEventTypes = #'(note-event rest-event tie-event
                                      beam-event tuplet-span-event
                                      dynamic-event slur-event)
  \cueDuring #"oboe" #UP { R1 }
  g2 c,
}
```



Le nom de l'instrument qui est répliqué sera imprimé à l'aide d'une annotation (un *markup*). Par ailleurs, si la citation nécessite l'apparition d'une clef différente, l'originale devrait être rappelée en fin de citation.

```
flute = \relative c'' {
  r2. c4 d8 c d e fis2 g2 d2
}
bassoon = \relative c {
  \clef bass
  R1
  \clef treble
  s1*0^\markup { \tiny "flute" }
  \cueDuring #"flute" #UP { R1 }
  \clef bass
  g4. b8 d2
}
\addQuote "flute" { \flute }
\new Staff {
  \bassoon
}
```
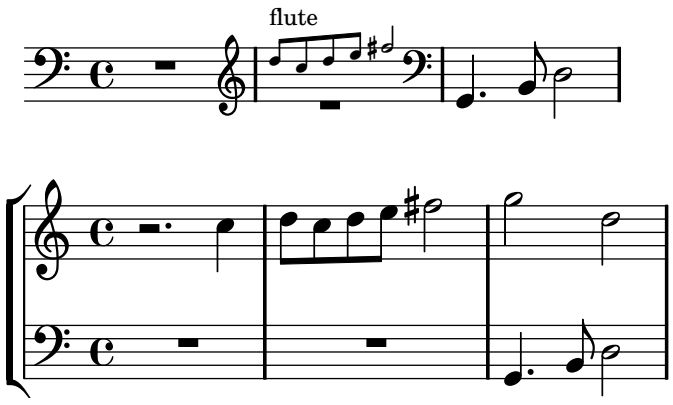


La commande `killCues` permet de supprimer les notes d'une citation. Ceci est utile lorsque cette citation n'est pas imprimée dans le conducteur entre autres. `killCues` supprimera les

notes et autres événements pris en charge par `\cueDuring`. Pour les autres annotations telles que changement de clef ou instrument concerné, faites appel à des balises – voir [Utilisation de balises], page 396 à ce sujet.

```
flute = \relative c'' {
  r2. c4 d8 c d e fis2 g2 d2
}
bassoon = \relative c {
  \clef bass
  R1
  \tag #'part {
    \clef treble
    s1*0^\markup { \tiny "flute" }
  }
  \cueDuring #"flute" #UP { R1 }
  \tag #'part \clef bass
  g4. b8 d2
}
\addQuote "flute" { \flute }

\new Staff {
  \bassoon
}
\new StaffGroup <<
  \new Staff {
    \flute
  }
  \new Staff {
    \removeWithTag #'part { \killCues { \bassoon } }
  }
>>
```



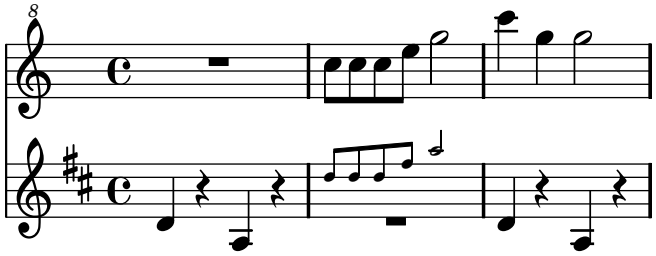
Lorsque la citation comporte une étiquette indiquant l'instrument qui joue et un changement de clef, ces éléments peuvent se regrouper à l'aide de la commande `\addInstrumentDefinition`. Pour plus d'information, reportez-vous au chapitre [Noms d'instrument], page 171.

L'instruction `\transposedCueDuring` est particulièrement adaptée pour des instrument ayant une tessiture éloignée, comme dans le cas d'un piccolo cité dans une partie de clarinette basse. Sa syntaxe est identique à celle de `\cueDuring`, à ceci près qu'elle nécessite un argument supplémentaire afin de spécifier la transposition à effectuer. Pour de plus amples informations sur la transposition, reportez-vous au chapitre [Instruments transpositeurs], page 19.

```
piccolo = \relative c''' {
  \clef "treble^8"
  R1
  c8 c c e g2
  c4 g g2
}
bassClarinet = \relative c' {
  \key d \major
  \transposition bes,
  d4 r a r
  \transposedCueDuring #"piccolo" #UP d { R1 }
  d4 r a r
}
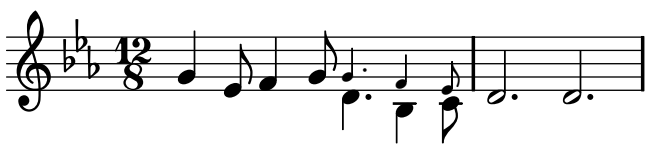
\addQuote "piccolo" { \piccolo }

<<
  \new Staff \piccolo
  \new Staff \bassClarinet
>>
```



Un contexte `CueVoice` créé explicitement permet d'afficher des notes dans une taille plus petite dans le but, par exemple, d'indiquer des notes alternatives pour une voix un peu plus haute ou basse.

```
\time 12/8
\key ees \major
g4 ees8 f4 g8
\stemDown
<<
  { d4. bes4 c8 }
  \new CueVoice
  { g'4. f4 ees8 }
>>
\stemUp
d2. d2.
```



Voir aussi

Manuel de notation : [Instruments transpositeurs], page 19, [Noms d'instrument], page 171, ⟨undefined⟩ [Musical cues], page ⟨undefined⟩, [Utilisation de balises], page 396.
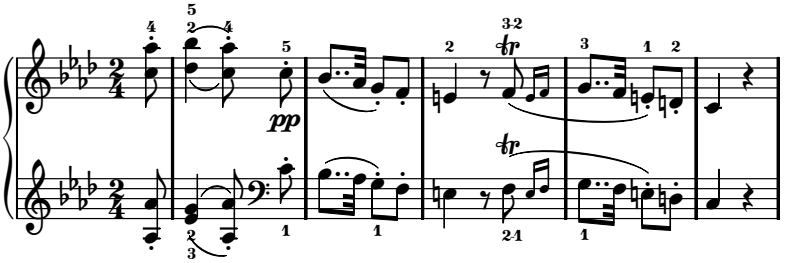
Morceaux choisis : Section “Notation sur la portée” dans *Morceaux choisis*.

Référence des propriétés internes : Section “CueVoice” dans *Référence des propriétés internes*, Section “Voice” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

La commande `\cueDuring` ne sait pas gérer les collisions de silence entre les contextes `Voice` et `CueVoice`.

1.7 Annotations éditoriales



Ce chapitre traite de la manière de modifier l’apparence des notes dans un but pédagogique ou d’analyse.

1.7.1 Dans la portée
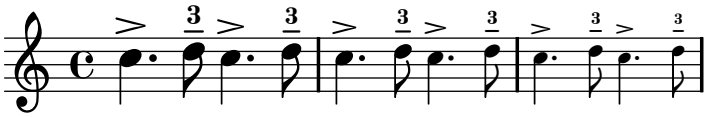
Nous allons voir ici comment mettre en exergue certains éléments à l’intérieur même de la portée.

Indication de la taille de fonte musicale

Le plus sûr moyen de régler la taille de la police, quel que soit le contexte, consiste à définir la propriété `fontSize`. Ceci ne modifiera en rien la taille des différents symboles tels que ligatures ou liaisons.

> **Note :** En matière de taille des éléments textuels, référez-vous au chapitre [Sélection de la fonte et de la taille], page 204.
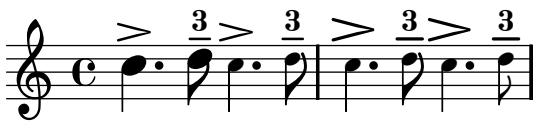
```
\huge
c4.-> d8---3
\large
c4.-> d8---3
\normalsize
c4.-> d8---3
\small
c4.-> d8---3
\tiny
c4.-> d8---3
\teeny
c4.-> d8---3
```



En interne, la propriété `fontSize` d’un contexte aura pour effet de définir la propriété `font-size` pour tous les objets de rendu. La valeur de `font-size` est un entier représentant la taille

proportionnellement à la hauteur standard de la portée en cours. Chaque incrément correspond à une augmentation d'environ 12 % de la taille de la police. Un pas de six aboutit exactement au doublement de la taille. La fonction Scheme `magstep` convertit le nombre affecté à `font-size` en facteur d'échelle. Vous pouvez aussi définir directement la propriété `font-size` de manière à n'affecter seulement que certains objets de rendu.

```
\set fontSize = #3
c4.-> d8---3
\override NoteHead #'font-size = #-4
c4.-> d8---3
\override Script #'font-size = #2
c4.-> d8---3
\override Stem #'font-size = #-5
c4.-> d8---3
```



Pour changer la taille des caractères, LilyPond met à l'échelle la fonte dont la taille est la plus proche de la taille voulue. La taille standard (pour laquelle `font-size = #0`) dépend de la hauteur de la portée. À une portée de 20 points correspond une police de 10 points.

La propriété `font-size` ne peut intervenir que pour les objets qui utilisent des polices, autrement dit ceux qui disposent de l'interface de rendu `font-interface`.

Commandes prédéfinies

`\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`.

Voir aussi
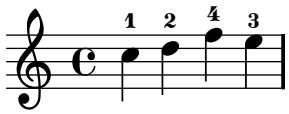
Morceaux choisis : Section "Annotations éditoriales" dans *Morceaux choisis*.

Référence des propriétés internes : Section "font-interface" dans *Référence des propriétés internes*.

Doigtés
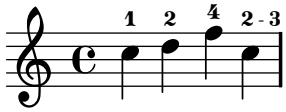
Les doigtés peuvent être indiqués comme suit : *note-chiffre_du_doigt*

```
c4-1 d-2 f-4 e-3
```



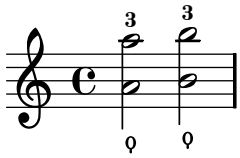
Pour les substitutions de doigts, on a recours à une indication textuelle (commande `\markup`) de doigté (commande `\finger`).

```
c4-1 d-2 f-4 c^\markup { \finger "2 - 3" }
```



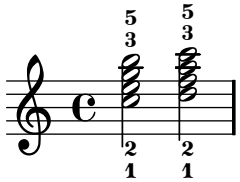
La commande `\thumb` peut être utilisée pour indiquer, par exemple dans une partition de violoncelle, si une note doit être jouée avec le pouce (*thumb* en anglais).

```
<a_\thumb a'-3>2 <b_\thumb b'-3>
```



Les doigtés des accords peuvent être saisis note par note, en les indiquant après chaque hauteur de note.

```
<c-1 e-2 g-3 b-5>2 <d-1 f-2 a-3 c-5>
```



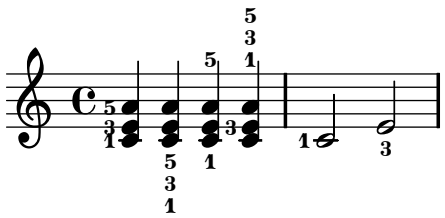
Les indications de doigtés peuvent se placer au-dessus ou en dessous de la portée – voir Section 5.4.2 [Direction et positionnement], page 475 à ce sujet.

Morceaux choisis

Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord < >, même s'il ne s'agit que d'une seule note.

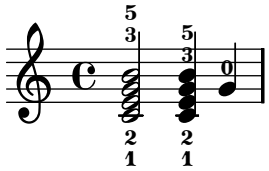
```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```



Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Néanmoins, il est possible d'annuler ce comportement.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering #'staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 <g'-0>
}
```



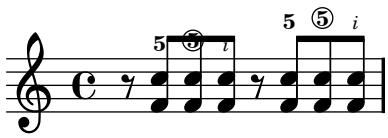
Avoiding collisions with chord fingerings

Fingerings and string numbers applied to individual notes will automatically avoid beams and stems, but this is not true by default for fingerings and string numbers applied to the individual notes of chords. The following example shows how this default behavior can be overridden.

```
\relative c' {
  \set fingeringOrientations = #'(up)
  \set stringNumberOrientations = #'(up)
  \set strokeFingerOrientations = #'(up)

  % Default behavior
  r8
  <f c'-5>8
  <f c'\5>8
  <f c'-\rightHandFinger #2 >8

  % Corrected to avoid collisions
  r8
  \override Fingering #'add-stem-support = ##t
  <f c'-5>8
  \override StringNumber #'add-stem-support = ##t
  <f c'\5>8
  \override StrokeFinger #'add-stem-support = ##t
  <f c'-\rightHandFinger #2 >8
}
```



Voir aussi

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 475

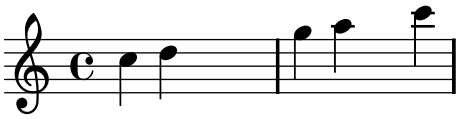
Morceaux choisis : Section “Annotations éditoriales” dans *Morceaux choisis*.

Référence des propriétés internes : Section “FingeringEvent” dans *Référence des propriétés internes*, Section “fingering-event” dans *Référence des propriétés internes*, Section “Fingering_engraver” dans *Référence des propriétés internes*, Section “New_fingering_engraver” dans *Référence des propriétés internes*, Section “Fingering” dans *Référence des propriétés internes*.

Dictée à trous
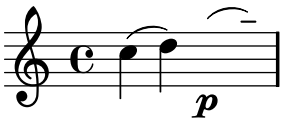
Les notes masquées – ou invisibles ou encore transparentes – sont utiles dans le cadre d'exercices de théorie ou de composition.

```
c4 d
\hideNotes
e4 f
\unHideNotes
g a
\hideNotes
b
\unHideNotes
c
```



Les objets de notation attachés à une note invisible ne seront pas masqués pour autant.

```
c4( d)
\hideNotes
e4(\p f)--
```



Commandes prédéfinies

`\hideNotes`, `\unHideNotes`.

Voir aussi

Manuel d'initiation : Section "Visibilité et couleur des objets" dans *Manuel d'initiation*.

Manuel de notation : [Silences invisibles], page 49, Section 5.4.7 [Visibilité des objets], page 475, [Masquage de portées], page 168.

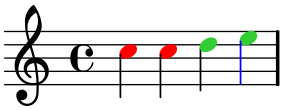
Morceaux choisis : Section "Annotations éditoriales" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Note_spacing_engraver" dans *Référence des propriétés internes*, Section "NoteSpacing" dans *Référence des propriétés internes*.

Coloration d'objets

Des objets peuvent être colorisés individuellement. Une listes des noms des couleurs disponibles se trouvent à l'annexe Section A.5 [Liste des couleurs], page 489.

```
\override NoteHead #'color = #red
c4 c
\override NoteHead #'color = #(x11-color 'LimeGreen)
d
\override Stem #'color = #blue
e
```
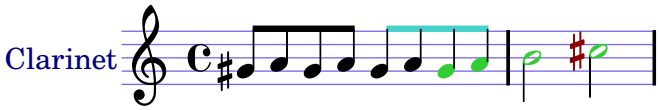


L'intégralité de la palette des couleurs définies pour X11 est accessible par la fonction Scheme `x11-color`. Cette fonction prend en argument une expression symbolique de la forme `'TaraTata` ou bien une chaîne de caractères comme `"TaraTata"`. La première formulation est à la fois plus rapide à écrire et aussi plus efficace. Néanmoins, la deuxième forme permet d'accéder aux noms composés des couleurs de X11.

Lorsque la fonction `x11-color` ne trouve pas le paramètre fourni, elle revient à la couleur par défaut, le noir. Le problème ressort de façon évidente au vu de la partition finale.

```
\override Staff.StaffSymbol #'color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}
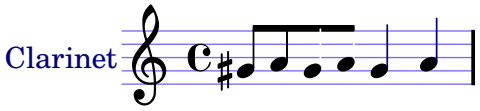
gis8 a
\override Beam #'color = #(x11-color "medium turquoise")
gis a
\override Accidental #'color = #(x11-color 'DarkRed)
gis a
\override NoteHead #'color = #(x11-color "LimeGreen")
gis a
% this is deliberate nonsense; note that the stems remain black
\override Stem #'color = #(x11-color 'Boggle)
b2 cis
```



Un autre moyen consiste à fournir à la fonction Scheme `rgb-color` les composantes de la couleur exacte au format rouge-vert-bleu (*RGB*) – chacune étant exprimée en en fraction de 256 (le 0.5 de l'exemple suivant correspond à 128).

```
\override Staff.StaffSymbol #'color = #(x11-color 'SlateBlue2)
\set Staff.instrumentName = \markup {
  \with-color #(x11-color 'navy) "Clarinet"
}

\override Stem #'color = #(rgb-color 0 0 0)
gis8 a
\override Stem #'color = #(rgb-color 1 1 1)
gis8 a
\override Stem #'color = #(rgb-color 0 0 0.5)
gis4 a
```



Voir aussi

Manuel de notation : Section A.5 [Liste des couleurs], page 489, Section 5.3.4 [La commande d'affinage (tweak)], page 474.

Morceaux choisis : Section "Annotations éditoriales" dans *Morceaux choisis*.

Problèmes connus et avertissements

Une couleur x11 n'aura pas forcément le même rendu qu'une couleur normale ayant un nom similaire.
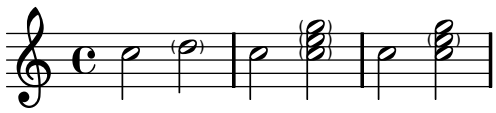
Les couleurs de X11 ne sont pas toutes perceptibles dans un navigateur internet. Aussi nous vous recommandons, dans le cadre d'une présentation multimedia, d'utiliser des couleurs de base `#blue`, `#green`, `#red` – bleu, vert, rouge.

Vous ne pouvez pas coloriser des notes à l'intérieur d'un accord avec `\override`. Si besoin est, utilisez `\tweak`. Pour plus de détails, consultez Section 5.3.4 [La commande d'affinage (tweak)], page 474.

Parenthèses
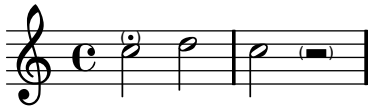
Des objets peuvent être mis entre parenthèses en saisissant `\parenthesize` juste avant l'événement musical. Si l'instruction préfixe un accord, chaque note le composant se présentera entre parenthèses. Vous pouvez aussi mettre individuellement entre parenthèses les notes d'un accord.

```
c2 \parenthesize d
c2 \parenthesize <c e g>
c2 <c \parenthesize e g>
```



Les objets autres que des notes peuvent aussi être entre parenthèses. En ce qui concerne les articulations, l'instruction `\parenthesize` doit cependant être précédée d'un tiret.

```
c2-\parenthesize -. d
c2 \parenthesize r
```



Voir aussi

Morceaux choisis : Section "Annotations éditoriales" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Parenthesis_engraver" dans *Référence des propriétés internes*, Section "ParenthesesItem" dans *Référence des propriétés internes*, Section "parentheses-interface" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Lorsqu'un accord est mis entre parenthèses, celles-ci s'appliquent individuellement à chacune des notes le composant, alors qu'on attendrait une seule paire de parenthèses encadrant tout l'accord.

Hampes

Dès qu'une note est rencontrée, un objet `Stem` se crée automatiquement. Pour les rondes et les silences, ils sont aussi créés, mais en mode invisible.

L'orientation des hampes peut être définie manuellement – voir Section 5.4.2 [Direction et positionnement], page 475 à ce sujet.

Commandes prédéfinies
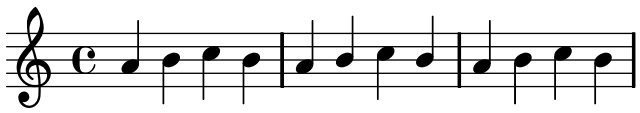
`\stemUp`, `\stemDown`, `\stemNeutral`.

Morceaux choisis

Default direction of stems on the center line of the staff

The default direction of stems on the center line of the staff is set by the `Stem` property `neutral-direction`.

```
\relative c'' {
  a4 b c b
  \override Stem #'neutral-direction = #up
  a4 b c b
  \override Stem #'neutral-direction = #down
  a4 b c b
}
```



Voir aussi

Manuel de notation : Section 5.4.2 [Direction et positionnement], page 475.

Morceaux choisis : Section “Annotations éditoriales” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Stem_engraver” dans *Référence des propriétés internes*, Section “Stem” dans *Référence des propriétés internes*, Section “stem-interface” dans *Référence des propriétés internes*.

1.7.2 Hors de la portée
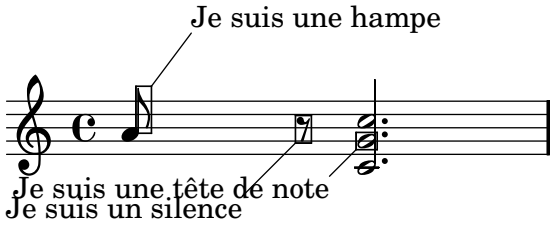
Nous allons nous intéresser ici à souligner des éléments inscrits dans la portée par des éléments qui lui seront externes.

Info-bulle

Vous pouvez marquer et nommer des éléments de notation à l’aide de bulles. L’objectif premier de cette fonctionnalité est d’expliquer la notation.

En voici un exemple :

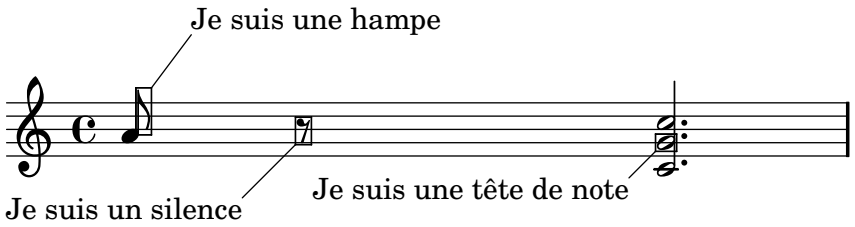
```
\new Voice \with { \consists "Balloon_engraver" }
{
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Je suis une hampe" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Je suis un silence" }
  r
  <c, g'-\balloonText #'(-2 . -2) \markup { "Je suis une tête de note" } c>2.
}
```



Vous disposez de deux fonctions musicales, `balloonGrobText` et `balloonText`. `balloonGrobText` prend en argument l'objet graphique à agrémenter et s'utilise comme `\once \override`. `balloonText`, quant à lui, s'utilise comme une simple articulation et fonctionne comme `\tweak` pour attacher du texte à l'une des notes d'un accord. Les autres arguments sont le décalage et le texte de la bulle.

Les info-bulles ont une influence sur les espacements, que l'on peut gérer ainsi :

```
\new Voice \with { \consists "Balloon_engraver" }
{
  \balloonLengthOff
  \balloonGrobText #'Stem #'(3 . 4) \markup { "Je suis une hampe" }
  a8
  \balloonGrobText #'Rest #'(-4 . -4) \markup { "Je suis un silence" }
  r
  \balloonLengthOn
  <c, g'-\balloonText #'(-2 . -2) \markup { "Je suis une tête de note" } c>2.
}
```



Commandes prédéfinies

`\balloonLengthOn`, `\balloonLengthOff`.

Voir aussi

Morceaux choisis : Section "Annotations éditoriales" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Balloon_engraver" dans *Référence des propriétés internes*, Section "BalloonTextItem" dans *Référence des propriétés internes*, Section "balloon-interface" dans *Référence des propriétés internes*.

Quadrillage temporel

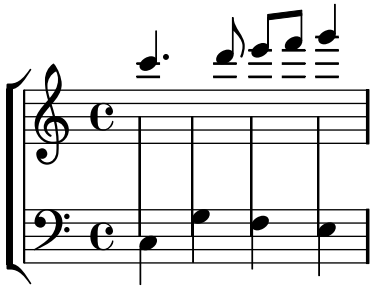
Vous pouvez tracer des lignes entre les portées, synchronisées avec les notes.

LilyPond a recours à deux graveurs distincts afin d'afficher le quadrillage : le `Grid_point_engraver` se charge de déterminer l'envergure du crochet, alors que le `Grid_line_span_engraver` se consacrera au tracé des lignes. Les lignes sont par défaut centrées horizontalement sous les notes et alignées sur la gauche des têtes. La propriété `gridInterval` spécifie quant à elle l'espace de temps entre chaque ligne.

```
\layout {
  \context {
    \Staff
    \consists "Grid_point_engraver" %% active les guides
    gridInterval = #(ly:make-moment 1 4)
  }
  \context {
    \Score
    \consists "Grid_line_span_engraver"
```

```
    %% centre les lignes guides horizontalement sous les notes
  }
}

\score {
  \new ChoirStaff <<
    \new Staff \relative c'' {
      \stemUp
      c'4. d8 e8 f g4
    }
    \new Staff \relative c {
    %% centre les lignes guides verticalement
      \clef bass
      \stemDown
      c4  g'  f  e
    }
  >>
}
```



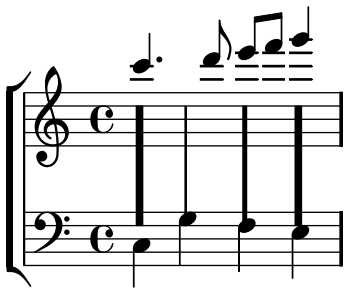
Morceaux choisis

Apparence du quadrillage temporel

Modifier certaines des propriétés du quadrillage temporel aura pour effet d'en changer l'apparence.

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \relative c'' {
        \stemUp
        c'4. d8 e8 f g4
      }
    }
    \new Staff {
      \relative c {
        % this moves them up one staff space from the default position
        \override Score.GridLine #'extra-offset = #'(0.0 . 1.0)
        \stemDown
        \clef bass
        \once \override Score.GridLine #'thickness = #5.0
        c4
        \once \override Score.GridLine #'thickness = #1.0
        g'4
        \once \override Score.GridLine #'thickness = #3.0
```

```
        f4
        \once \override Score.GridLine #'thickness = #5.0
        e4
      }
    }
  >>
  \layout {
    \context {
      \Staff
      % set up grids
      \consists "Grid_point_engraver"
      % set the grid interval to one quarter note
      gridInterval = #(ly:make-moment 1 4)
    }
    \context {
      \Score
      \consists "Grid_line_span_engraver"
      % this moves them to the right half a staff space
      \override NoteColumn #'X-offset = #-0.5
    }
  }
}
```



Voir aussi

Morceaux choisis : Section “Annotations éditoriales” dans *Morceaux choisis*.
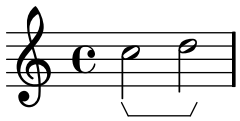
Référence des propriétés internes : Section “Grid_line_span_engraver” dans *Référence des propriétés internes*, Section “Grid_point_engraver” dans *Référence des propriétés internes*, Section “GridLine” dans *Référence des propriétés internes*, Section “GridPoint” dans *Référence des propriétés internes*, Section “grid-line-interface” dans *Référence des propriétés internes*, Section “grid-point-interface” dans *Référence des propriétés internes*.

Crochets d’analyse

On utilise des crochets en analyse musicale, pour indiquer la structure d’une pièce.

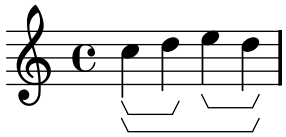
```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
```

```
  c2\startGroup
  d\stopGroup
}
```



Les crochets d'analyses sont susceptibles d'être imbriqués :

```
\layout {
  \context {
    \Voice
    \consists "Horizontal_bracket_engraver"
  }
}
\relative c'' {
  c4\startGroup\startGroup
  d4\stopGroup
  e4\startGroup
  d4\stopGroup\stopGroup
}
```

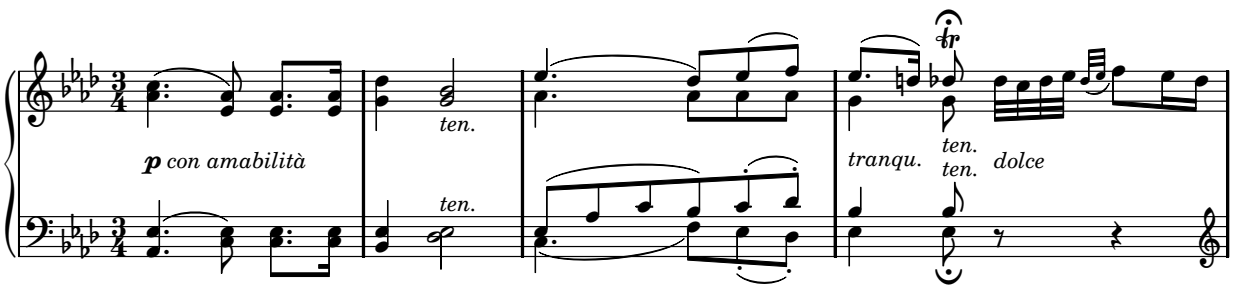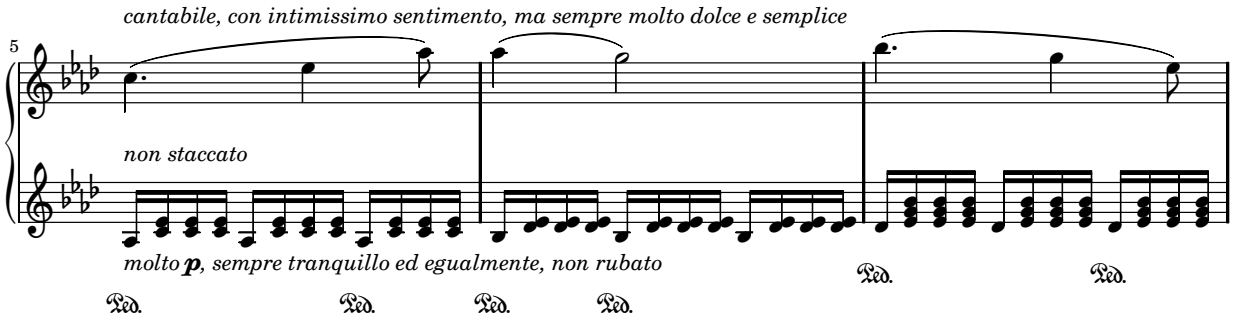


Voir aussi

Morceaux choisis : Section "Annotations éditoriales" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Horizontal_bracket_engraver" dans *Référence des propriétés internes*, Section "HorizontalBracket" dans *Référence des propriétés internes*, Section "horizontal-bracket-interface" dans *Référence des propriétés internes*, Section "Staff" dans *Référence des propriétés internes*.

1.8 Texte





Nous allons voir ici comment insérer du texte dans une partition, ainsi que différentes manières de le mettre en forme.

Certains éléments de texte ne sont pas abordés ici mais dans des chapitres qui leur sont dédiés. C'est le cas de la Section 2.1 [Musique vocale], page 219 et des Section 3.2 [Titres et entêtes], page 386.

1.8.1 Ajout de texte

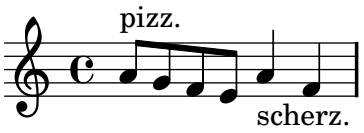
Cette partie constitue une introduction aux différentes manières d'ajouter du texte à une partition.

> **Note :** Pour écrire des accents et autres caractères spéciaux, il suffit de les insérer directement dans votre fichier LilyPond. Ce fichier devra être sauvegardé avec l'encodage UTF-8. Pour plus d'informations, voir Section 3.3.3 [Codage du texte], page 399.

Commentaires textuels
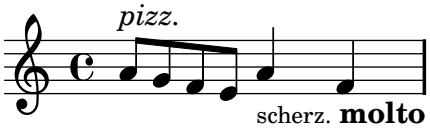
Vous pouvez ajouter à une partition des indications sous forme texuelle, comme dans l'exemple suivant. Ces indications se placeront manuellement au-dessus ou au-dessous de la portée selon la syntaxe utilisée – cf. Section 5.4.2 [Direction et positionnement], page 475.

```
a8^"pizz." g f e a4-"scherz." f
```
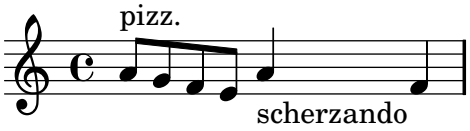


Cette syntaxe est en fait un raccourci. Des constructions plus élaborées d'annotation peuvent être obtenues en ayant recours à un bloc `\markup` et selon les préceptes énoncés dans Section 1.8.2 [Mise en forme du texte], page 202.

```
a8^\markup { \italic pizz. } g f e
a4_\markup { \tiny scherz. \bold molto } f
```



Par défaut, ces indications n'affectent en rien l'espacement des notes. Leur longueur peut néanmoins être prise en considération : dans l'exemple qui suit, le premier commentaire n'influe pas sur l'espacement, à l'inverse du second.

```
a8^"pizz." g f e
\textLengthOn
a4_"scherzando" f
```



En plus d'indications textuelles, les notes peuvent se voir attacher des articulations, comme indiqué au chapitre [Articulations et ornements], page 102.

Pour de plus amples détails sur la manière de combiner indications textuelles et articulations, reportez-vous au chapitre Section "Positionnement des objets" dans *Manuel d'initiation*.

Commandes prédéfinies

`\textLengthOn`, `\textLengthOff`.

Voir aussi

Manuel d'initiation : Section "Positionnement des objets" dans *Manuel d'initiation*.

Manuel de notation : Section 1.8.2 [Mise en forme du texte], page 202, Section 5.4.2 [Direction et positionnement], page 475, [Articulations et ornements], page 102.

Morceaux choisis : Section "Texte" dans *Morceaux choisis*.

Référence des propriétés internes : Section "TextScript" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

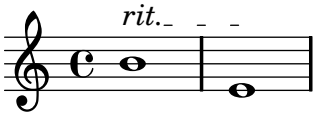
S'assurer que tous les éléments textuels et les paroles respectent les marges du document requiert des calculs relativement lourds ; c'est la raison pour laquelle LilyPond, par défaut, ne s'en préoccupe pas. Vous pouvez cependant l'y forcer en définissant

```
\override Score.PaperColumn #'keep-inside-line = ##t
```

Indication textuelle avec extension

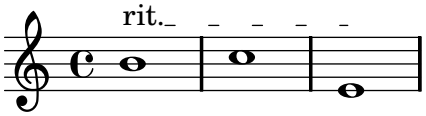
Certaines indications d'interprétation comme *rallentando*, *accelerando* ou *trilles*, s'inscrivent textuellement et se prolongent sur plusieurs notes à l'aide d'une ligne pleine, pointillée ou ondulée. Ces objets, que l'on appelle « extenseurs », se dessinent entre deux notes à l'aide de la syntaxe suivante :

```
\override TextSpanner #'(bound-details left text) = "rit."
b1\startTextSpan
e,\stopTextSpan
```



Le texte à imprimer est spécifié en tant que propriété de l'objet `TextSpanner`. Il apparaîtra par défaut en italique ; cependant, rien ne s'oppose à un autre graphisme dès lors que vous faites appel à un bloc `\markup` – voir Section 1.8.2 [Mise en forme du texte], page 202.

```
\override TextSpanner #'(bound-details left text) =
  \markup { \upright "rit." }
b1\startTextSpan c
e,\stopTextSpan
```



Le style de ligne se définit lui aussi comme une propriété de l'objet. Les détails concernant la syntaxe à utiliser sont expliqués au chapitre Section 5.4.8 [Styles de ligne], page 476.

Commandes prédéfinies

`\textSpannerUp`, `\textSpannerDown`, `\textSpannerNeutral`.

Problèmes connus et avertissements

LilyPond ne peut traiter qu'un seul extenseur à la fois par voix.
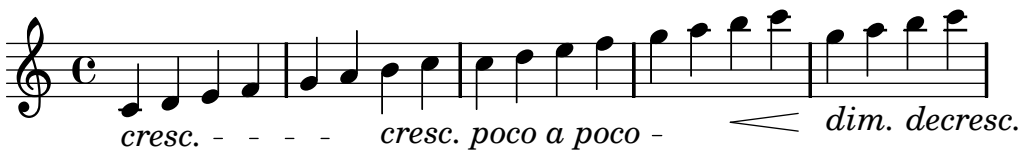
Morceaux choisis

Extensions de nuance postfix

Les lignes d'extension des commandes \cresc, \dim et \decresc peuvent désormais être personnalisées facilement sous forme d'opérateurs postfix. Soufflets et (de)crescendos peuvent cohabiter. `\<` et `\>` produiront par défaut des soufflets, alors que \cresc etc. produiront une indication textuelle avec extension.

```
% Some sample text dynamic spanners, to be used as postfix operators
crpoco =
#(make-music 'CrescendoEvent
             'span-direction START
             'span-type 'text
             'span-text "cresc. poco a poco")

\relative c' {
  c4\cresc d4 e4 f4 |
  g4 a4\! b4\crpoco c4 |
  c4 d4 e4 f4 |
  g4 a4\! b4\< c4 |
  g4\dim a4 b4\decresc c4\!
}
```



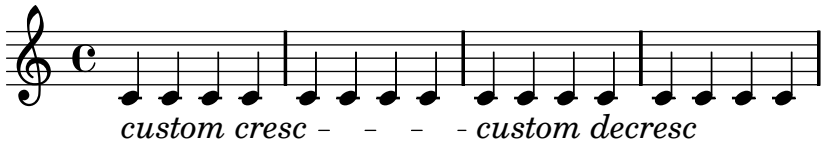
Personnalisation des extenseurs de nuance postfix

Il s'agit de fonctions postfix pour personnaliser l'extension des crescendos textuels. L'extension devrait débuter sur la première notte de la mesure. Il faut utiliser `-\mycresc` – comme une articulation – sous peine que le départ de l'extension n'apparaisse qu'à la note suivante.

```
% Two functions for (de)crescendo spanners where you can explicitly give the
% spanner text.
mycresc =
#(define-music-function (parser location mymarkup) (markup?)
   (make-music 'CrescendoEvent
               'span-direction START
               'span-type 'text
               'span-text mymarkup))
mydecresc =
#(define-music-function (parser location mymarkup) (markup?)
   (make-music 'DecrescendoEvent
               'span-direction START
```

```
                'span-type 'text
                'span-text mymarkup))

\relative c' {
  c4-\mycresc "custom cresc" c4 c4 c4 |
  c4 c4 c4 c4 |
  c4-\mydecresc "custom decresc" c4 c4 c4 |
  c4 c4\! c4 c4
}
```



Voir aussi

Manuel de notation : Section 5.4.8 [Styles de ligne], page 476, [Nuances], page 104, Section 1.8.2 [Mise en forme du texte], page 202.
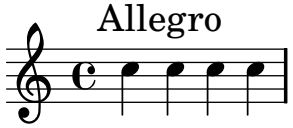
Morceaux choisis : Section “Texte” dans *Morceaux choisis*, Section “Signes d’interprétation” dans *Morceaux choisis*.

Référence des propriétés internes : Section “TextSpanner” dans *Référence des propriétés internes*.

Indications textuelles
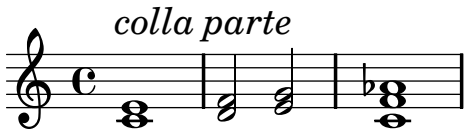
La commande `\mark` est tout d’abord conçue pour les [Indications de repère], page 93.

```
c4
\mark "Allegro"
c c c
```



Cette syntaxe rend possible l’adjonction de n’importe quel texte à une barre de mesure. Ce texte peut être mis en forme de différentes manières dès lors qu’est utilisé un bloc `\markup`, comme indiqué au chapitre Section 1.8.2 [Mise en forme du texte], page 202.
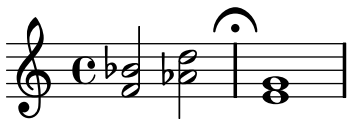
```
<c e>1
\mark \markup { \italic { colla parte } }
<d f>2 <e g>
<c f aes>1
```



Elle peut aussi servir à insérer des signes de *coda* ou de *segno*, ou bien un point d’orgue, au dessus d’une barre de mesure. Couplez-la alors à la commande `\markup` pour avoir accès au

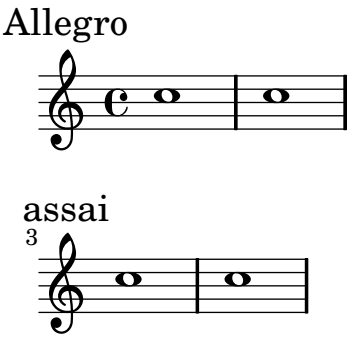
symbole approprié, selon les indications contenues au chapitre [Notation musicale dans du texte formaté], page 212.

```
<bes f>2 <aes d>
\mark \markup { \musicglyph #"scripts.ufermata" }
<e g>1
```



Le résultat de `\mark` n'apparaîtra que sur la portée supérieure d'un système. Si vous introduisez la commande `\mark` au moment d'une barre de mesure, la marque se placera au dessus de la barre. Si vous y faites appel au milieu d'une mesure, cette marque sera positionnée entre les notes. Si elle intervient en début de ligne, elle sera placée juste avant la première note de cette portée. Enfin, une marque qui tomberait sur un saut de ligne sera imprimée au début de la ligne suivante.

```
\mark "Allegro"
c1 c
\mark "assai" \break
c  c
```
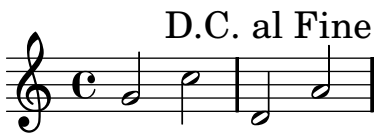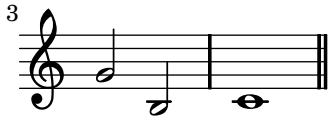


Morceaux choisis

Indication texuelle en fin de ligne

Les indications textuelles peuvent être imprimées à la fin d'une ligne plutôt qu'en tête de la suivante. Pensez alors à aligner l'extrémité droite de l'indication sur la barre de mesure.

```
\relative c'' {
  g2 c
  d,2 a'
  \once \override Score.RehearsalMark #'break-visibility = #end-of-line-visible
  \once \override Score.RehearsalMark #'self-alignment-X = #RIGHT
  \mark "D.C. al Fine"
  \break
  g2 b,
  c1 \bar "||"
}
```





*Alignement des indications par rapport à divers objets*de notation

Les indications textuelles peuvent s'aligner par rapport à d'autres objets que des barres de mesure, tels que `ambitus`, `breathing-sign`, `clef`, `custos`, `staff-bar`, `left-edge`, `key-cancellation`, `key-signature`, ou `time-signature`.

Par défaut, les indications textuelles sont alignées avec le milieu des objets de notation. Bien entendu, vous pouvez modifier les propriétés des objets en question pour obtenir un autre résultat comme l'illustre la deuxième ligne de l'exemple. Dans le cas de portées multiples, ces réglages doivent être faits pour chacune d'entre elles.

```
\relative c' {
  e1

  % the RehearsalMark will be centered above the Clef
  \override Score.RehearsalMark #'break-align-symbols = #'(clef)
  \key a \major
  \clef treble
  \mark ""
  e1

  % the RehearsalMark will be centered above the TimeSignature
  \override Score.RehearsalMark #'break-align-symbols = #'(time-signature)
  \key a \major
  \clef treble
  \time 3/4
  \mark \markup { \char ##x2193 }
  e2.

  % the RehearsalMark will be centered above the KeySignature
  \override Score.RehearsalMark #'break-align-symbols = #'(key-signature)
  \key a \major
  \clef treble
  \time 4/4
  \mark \markup { \char ##x2193 }
  e1

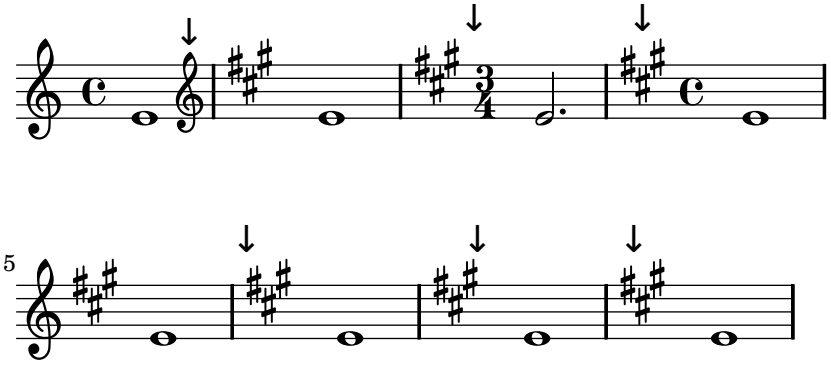
  \break
  e1

  % the RehearsalMark will be aligned with the left edge of the KeySignature
  \once \override Score.KeySignature #'break-align-anchor-alignment = #LEFT
  \mark \markup { \char ##x2193 }
  \key a \major
  e1

  % the RehearsalMark will be aligned with the right edge of the KeySignature
  \once \override Score.KeySignature #'break-align-anchor-alignment = #RIGHT
  \key a \major
  \mark \markup { \char ##x2193 }
  e1

  % the RehearsalMark will be aligned with the left edge of the KeySignature
```

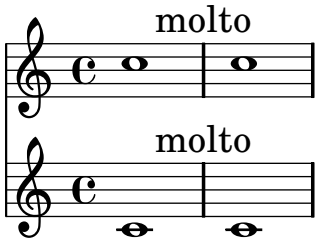
```
  % and then shifted right by one unit.
  \once \override Score.KeySignature #'break-align-anchor = #1
  \key a \major
  \mark \markup { \char ##x2193 }
  e1
}
```



Impression des indications sur toutes les portées d'un système

Bien que ces indications textuelles ne soient habituellement imprimées qu'au niveau de la portée supérieure, vous pouvez forcer leur affectation à chacune des portées.

```
\score {
  <<
    \new Staff { c''1 \mark "molto" c'' }
    \new Staff { c'1 \mark "molto" c' }
  >>
  \layout {
    \context {
      \Score
      \remove "Mark_engraver"
      \remove "Staff_collecting_engraver"
    }
    \context {
      \Staff
      \consists "Mark_engraver"
      \consists "Staff_collecting_engraver"
    }
  }
}
```



Voir aussi

Manuel de notation : [Indications de repère], page 93, Section 1.8.2 [Mise en forme du texte], page 202, [Notation musicale dans du texte formaté], page 212, Section A.6 [La fonte Feta], page 491.

Morceaux choisis: Section “Texte” dans *Morceaux choisis*.

Référence des propriétés internes : Section “MarkEvent” dans *Référence des propriétés internes*, Section “Mark_engraver” dans *Référence des propriétés internes*, Section “RehearsalMark” dans *Référence des propriétés internes*.

Texte indépendant

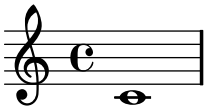
Un bloc `\markup` peut exister en lui-même, indépendamment de tout bloc `\score`, et venir en préambule par exemple – voir le chapitre Section 3.1.5 [Structure de fichier], page 384 à ce propos.

```
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
```

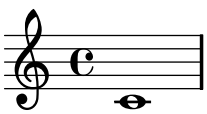
Tomorrow, and tomorrow, and tomorrow...

De cette manière, vous pouvez insérer du texte en dehors de la musique. Ceci devient particulièrement utile lorsque le fichier source contient plusieurs morceaux. Pour plus d’informations à ce propos, reportez-vous au chapitre Section 3.1.2 [Plusieurs partitions dans un même ouvrage], page 381.

```
\score {
  c'1
}
\markup {
  Tomorrow, and tomorrow, and tomorrow...
}
\score {
  c'1
}
```



Tomorrow, and tomorrow, and tomorrow...



Les blocs de textes peuvent s’étendre sur plusieurs pages, ce qui permet de générer des ouvrages complets uniquement grâce à LilyPond. Cette fonctionnalité, ainsi que la syntaxe appropriée, est abordée plus en détail au chapitre [Texte avec sauts de page], page 215.

Commandes prédéfinies

`\markup`, `\markuplines`.

Morceaux choisis

Bloc de texte indépendant sur deux colonnes

L'utilisation de la commande `\markup` permet de distribuer un bloc de texte indépendant sur plusieurs colonnes.

```
\markup {
  \fill-line {
    \hspace #1
    \column {
      \line { O sacrum convivium }
      \line { in quo Christus sumitur, }
      \line { recolitur memoria passionis ejus, }
      \line { mens impletur gratia, }
      \line { futurae gloriae nobis pignus datur. }
      \line { Amen. }
    }
    \hspace #2
    \column {
      \line { \italic { O sacred feast } }
      \line { \italic { in which Christ is received, } }
      \line { \italic { the memory of His Passion is renewed, } }
      \line { \italic { the mind is filled with grace, } }
      \line { \italic { and a pledge of future glory is given to us. } }
      \line { \italic { Amen. } }
    }
    \hspace #1
  }
}
```

O sacrum convivium
in quo Christus sumitur,
recolitur memoria passionis ejus,
mens impletur gratia,
futurae gloriae nobis pignus datur.
Amen.

O sacred feast
in which Christ is received,
the memory of His Passion is renewed,
the mind is filled with grace,
and a pledge of future glory is given to us.
Amen.

Voir aussi

Manuel de notation : Section 1.8.2 [Mise en forme du texte], page 202, Section 3.1.5 [Structure de fichier], page 384, Section 3.1.2 [Plusieurs partitions dans un même ouvrage], page 381, [Texte avec sauts de page], page 215.

Morceaux choisis : Section "Texte" dans *Morceaux choisis*.

Référence des propriétés internes : Section "TextScript" dans *Référence des propriétés internes*.

1.8.2 Mise en forme du texte

Nous allons voir dans les lignes qui suivent la manière de mettre en forme du texte à l'aide de la syntaxe propre au mode `\markup`.

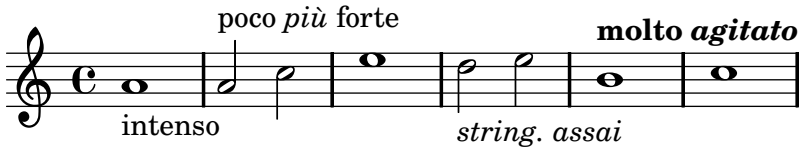
Introduction au formatage de texte

La commande `\markup` permet d'ajouter du texte et dispose de sa propre syntaxe que nous appellerons le « mode *markup* ».

La syntaxe du mode *markup* n'est pas différente de celle des autres modes de LilyPond : une expression `\markup` est bornée par des accolades `{ ... }`. Un mot unique sera considéré comme une expression minimale, et n'aura donc pas besoin d'être mis entre accolades.

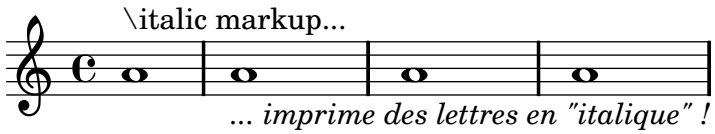
Contrairement aux indications simples, du type "entre guillemets", les blocs `\markup` peuvent contenir des expressions imbriquées ou d'autres commandes *markup*, dès lors qu'elles sont précédées du caractère \. Ces commandes n'affecteront que la première expression qui les suit.

```
a1-\markup intenso
a2^\markup { poco \italic più forte  }
c e1
d2_\markup { \italic "string. assai" }
e
b1^\markup { \bold { molto \italic  agitato } }
c
```



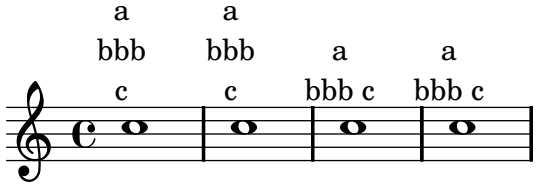
Un bloc `\markup` peut contenir du texte entre guillemets. De telles chaines seront considérées comme des expressions textuelles minimales ; à ce titre, toute commande de type *markup* ou tout caractère spécial – tel un \ ou un # – sera imprimé littéralement et sans influer sur le formatage du texte. Il est de ce fait possible d'imprimer des guillemets informatiques " dès lors qu'ils sont précédés d'une oblique inverse.

```
a1^"\italic markup..."
a_\markup { \italic "... imprime des lettres en \"italique\" !" }
a a
```



Une liste de mots, pour pouvoir être traitée en tant qu'expression distincte, doit être bornée par des " ou précédée d'une commande. La manière de définir les expressions *markup* aura une influence sur la manière dont elles seront empilées, centrées ou alignées. Dans l'exemple qui suit, la deuxième expression `\markup` est traitée tout comme la première :
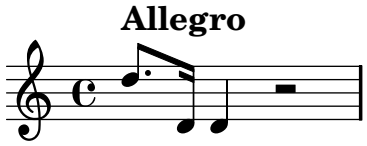
```
c1^\markup { \center-column { a bbb c } }
c1^\markup { \center-column { a { bbb c } } }
c1^\markup { \center-column { a \line { bbb c } } }
c1^\markup { \center-column { a "bbb c" } }
```



Vous pouvez stocker les étiquettes textuelles en tant que variables, et attacher ces identificateurs à des notes, comme ici :

```
allegro = \markup { \bold \large Allegro }

{
  d''8.^\allegro
  d'16 d'4 r2
}
```



Pour une liste des différentes commandes spécifiques au mode `\markup`, consultez l'annexe Section A.8 [Text markup commands], page 510 (en anglais).

Voir aussi

Manuel de notation : Section A.8 [Text markup commands], page 510.

Morceaux choisis : Section "Texte" dans *Morceaux choisis*.

Fichiers d'initialisation : '`scm/markup.scm`'.
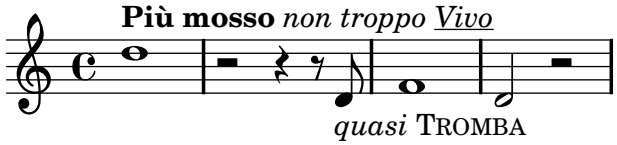
Problèmes connus et avertissements

Les erreurs de syntaxe en mode *markup* sont peu explicites.

Sélection de la fonte et de la taille

Le mode *markup* autorise des changements élémentaires de la fonte :

```
d1^\markup {
  \bold { Più mosso }
  \italic { non troppo \underline Vivo }
}
r2 r4 r8
d,_\markup { \italic quasi \smallCaps Tromba }
f1 d2 r
```
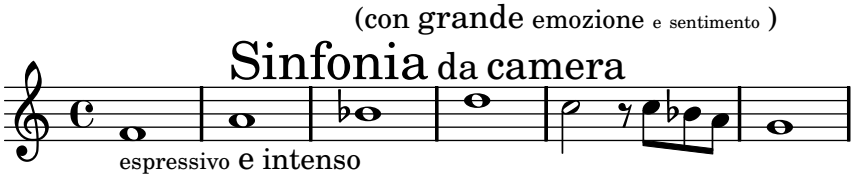


La taille des caractères se modifie de différentes manières :

- à partir de l'une des tailles standard prédéfinies,
- en étant définie en valeur absolue,
- en adoptant une valeur relative à celle précédemment définie.

Voici une illustration de ces trois différentes méthodes :

```
f1_\markup {
  \tiny espressivo
  \large e
  \normalsize intenso
}
a^\markup {
  \fontsize #5 Sinfonia
```

```
  \fontsize #2 da
  \fontsize #3 camera
}
bes^\markup { (con
  \larger grande
  \smaller emozione
  \magnify #0.6 { e sentimento } )
}
d c2 r8 c bes a g1
```



Vous pouvez imprimer du texte en indice ou en exposant. Celui-ci sera dans un taille plus petite, mais rien ne s'oppose à ce que vous lui affectiez un taille normale :

```
\markup {
  \column {
    \line { 1 \super st movement }
    \line { 1 \normal-size-super st movement
      \sub { (part two) }  }
  }
}
```

1st movement
1st movement (part two)

Le mode *markup* vous permet de sélectionner d'autres familles de fontes. Par défaut, LilyPond utilise une police avec empattement, du type roman, et tout changement doit être explicite. Dans la dernière ligne de l'exemple qui suit, vous noterez qu'il n'y a aucune différence entre les premier et deuxième mots.

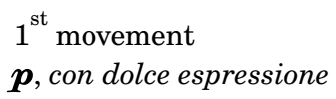
```
\markup {
  \column {
    \line { Act \number 1 }
    \line { \sans { Scene I. } }
    \line { \typewriter { Verona. An open place. } }
    \line { Enter \roman Valentine and Proteus. }
  }
}
```

Act 1
Scene I.
Verona. An open place.
Enter Valentine and Proteus.

Certaines familles de police spécifiques aux nombres ou aux nuances par exemple, ne disposent pas de tous les caractères, comme nous l'avons vu dans les chapitres [Personnalisation des indications de nuance], page 109 et [Indications de reprise manuelles], page 129.

Lorsqu'un changement survient au milieu d'un mot, il se peut qu'un espacement supplémentaire apparaisse. Il suffit en pareil cas de concaténer les différents éléments :

```
\markup {
  \column {
    \line {
      \concat { 1 \super st }
      movement
    }
    \line {
      \concat { \dynamic p , }
      \italic { con dolce espressione }
    }
  }
}
```



Une liste des différentes commandes permettant de changer de fonte ou d'utiliser des fontes personnalisées est disponible à l'annexe Section A.8.1 [Font], page 511.

Pour savoir comment personnaliser des fontes, reportez-vous au chapitre Section 1.8.3 [Fontes], page 216.

Commandes prédéfinies

`\teeny`, `\tiny`, `\small`, `\normalsize`, `\large`, `\huge`, `\smaller`, `\larger`.

Voir aussi

Manuel de notation : Section A.8.1 [Font], page 511, [Personnalisation des indications de nuance], page 109, [Indications de reprise manuelles], page 129, Section 1.8.3 [Fontes], page 216.

Morceaux choisis : Section "Texte" dans *Morceaux choisis*.

Référence des propriétés internes : Section "TextScript" dans *Référence des propriétés internes*.

Fichiers d'initialisation : '`scm/define-markup-commands.scm`'.

Problèmes connus et avertissements

Le recours aux commandes `\teeny`, `\tiny`,`\small`, `\normalsize`, `\large` et `\huge` produiront des espacements nettement moins réguliers que si vous utilisez `\fontsize`.

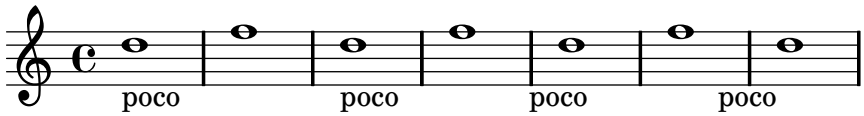
Alignement du texte

Cette partie traite de la manière de positionner du texte en mode *markup*. On déplace l'intégralité d'un objet *markup* en utilisant la syntaxe décrite au chapitre Section "Déplacement d'objets" dans *Manuel d'initiation*.

Les objets de type *markup* peuvent s'aligner de différentes manières. Une indication textuelle est par défaut alignée sur son extrémité gauche. Dans l'exemple qui suit, il n'y a aucune différence entre les deux premiers *markups*.

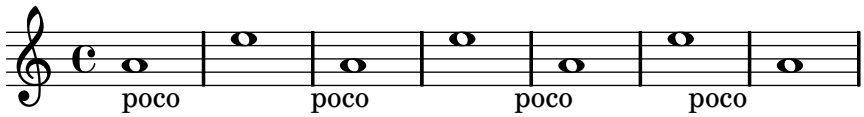
```
d1-\markup { poco }
f
d-\markup { \left-align poco }
f
```

```
d-\markup { \center-align { poco } }
f
d-\markup { \right-align poco }
```



L'alignement horizontal peut être ajusté à l'aide d'une valeur numérique :

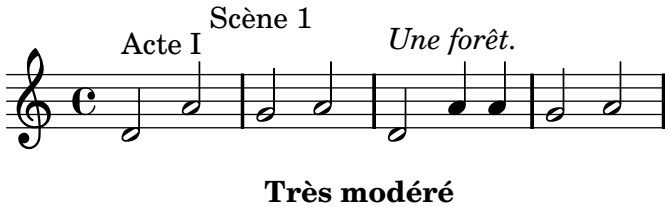
```
a1-\markup { \halign #-1 poco }
e'
a,-\markup { \halign #0 poco }
e'
a,-\markup { \halign #0.5 poco }
e'
a,-\markup { \halign #2 poco }
```



Certains objets possèdent leurs propres procédures d'alignement, qui annuleront toute spécification d'alignement que vous pourriez affecter à un argument de type *markup* que vous leur auriez fourni. La solution consiste alors à déplacer l'intégralité de ces objets *markup*, comme indiqué par exemple au chapitre [Indications textuelles], page 197.

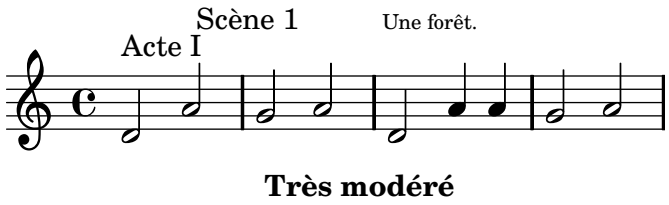
L'alignement vertical est quant à lui un peu plus compliqué. Comme nous l'avons vu ci-avant, les objets *markup* peuvent être déplacés dans leur intégralité. Il est néanmois possible de déplacer certains éléments spécifiques au sein d'un bloc *markup*. En pareil cas, l'élément à déplacer doit être précédé d'un « point d'ancrage » – un autre élément du *markup* ou un objet invisible. L'exemple qui suit illustre ces deux possibilités. Vous noterez par ailleurs que le dernier *markup*, dépourvu de point d'ancrage, n'est de ce fait pas déplacé.

```
d2^\markup {
  Acte I
  \raise #2 { Scène 1 }
}
a'
g_\markup {
  \null
  \lower #4 \bold { Très modéré }
}
a
d,^\markup {
  \raise #4 \italic { Une forêt. }
}
a'4 a g2 a
```



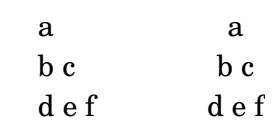
Certaines commandes permettent de régler l'alignement des objets textuels en mode *markup*, tant au niveau horizontal que vertical. Tout objet soumis à ces commandes doit être précédé d'un point d'ancrage.

```
d2^\markup {
  Acte I
  \translate #'(-1 . 2) "Scène 1"
}
a'
g_\markup {
  \null
  \general-align #Y #3.2 \bold "Très modéré"
}
a
d,^\markup {
  \null
  \translate-scaled #'(-1 . 2) \teeny "Une forêt."
}
a'4 a g2 a
```



Un objet de type *markup* peut contenir plusieurs lignes de texte. Dans l'exemple suivant, chaque élément ou expression ira se placer sur sa propre ligne, tantôt alignée à gauche, tantôt centrée.

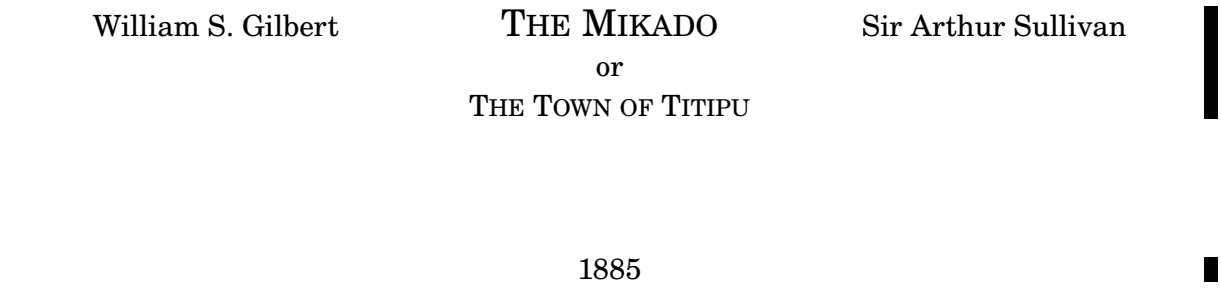
```
\markup {
  \column {
    a
    "b c"
    \line { d e f }
  }
  \hspace #10
  \center-column {
    a
    "b c"
    \line { d e f }
  }
}
```



Pareillement, une liste d'éléments ou d'expressions sera répartie sur une ligne entière, voire même centrée sur toute la page s'il n'y a qu'un seul élément. De telles expressions peuvent à leur tour contenir du texte multi-ligne ou une autre expression *markup*.

```
\markup {
  \fill-line {
```

```
    \line { William S. Gilbert }
    \center-column {
      \huge \smallCaps "The Mikado"
      or
      \smallCaps "The Town of Titipu"
    }
    \line { Sir Arthur Sullivan }
  }
}
\markup {
  \fill-line { 1885 }
}
```



Les indications textuelles, lorsqu'elles sont relativement longues, peuvent se répartir sur plusieurs lignes en fonction de la largeur de ligne. Le texte sera alors soit aligné à gauche, soit justifié, comme le montre l'exemple suivant :

```
\markup {
  \column {
    \line  \smallCaps { La vida breve }
    \line \bold { Acto I }
    \wordwrap \italic {
      (La escena representa el corral de una casa de
      gitanos en el Albaicín de Granada. Al fondo una
      puerta por la que se ve el negro interior de
      una Fragua, iluminado por los rojos resplandores
      del fuego.)
    }
    \hspace #0

    \line \bold { Acto II }
    \override #'(line-width . 50)
    \justify \italic {
      (Calle de Granada. Fachada de la casa de Carmela
      y su hermano Manuel con grandes ventanas abiertas
      a través de las que se ve el patio
      donde se celebra una alegre fiesta)
    }
  }
}
```

LA VIDA BREVE

Acto I

(La escena representa el corral de una casa de gitanos en el Albaicín de Granada. Al fondo una puerta por la que se ve el negro interior de una Fragua, iluminado por los rojos resplandores del fuego.)

Acto II

(Calle de Granada. Fachada de la casa de Carmela y su hermano Manuel con grandes ventanas abiertas a través de las que se ve el patio donde se celebra una alegre fiesta)

Une liste des différentes commandes permettant d'aligner du texte en mode *markup* est disponible à l'annexe Section A.8.2 [Align], page 519.

Voir aussi

Manuel d'initiation : Section "Déplacement d'objets" dans *Manuel d'initiation*.

Manuel de notation : Section A.8.2 [Align], page 519, [Indications textuelles], page 197.

Morceaux choisis : Section "Texte" dans *Morceaux choisis*.

Référence des propriétés internes : Section "TextScript" dans *Référence des propriétés internes*.

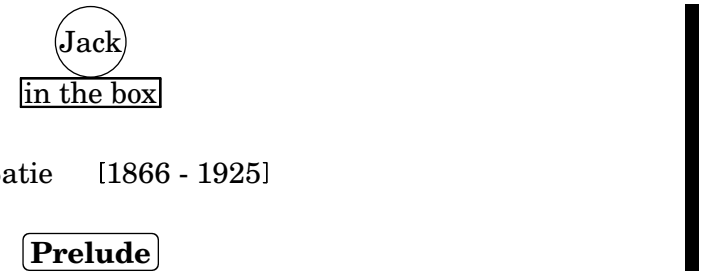
Fichiers d'initialisation : '`scm/define-markup-commands.scm`'.

Éléments graphiques dans du texte formaté

Vous pouvez, grâce aux mode *markup*, ajouter divers objets graphiques à votre partition.

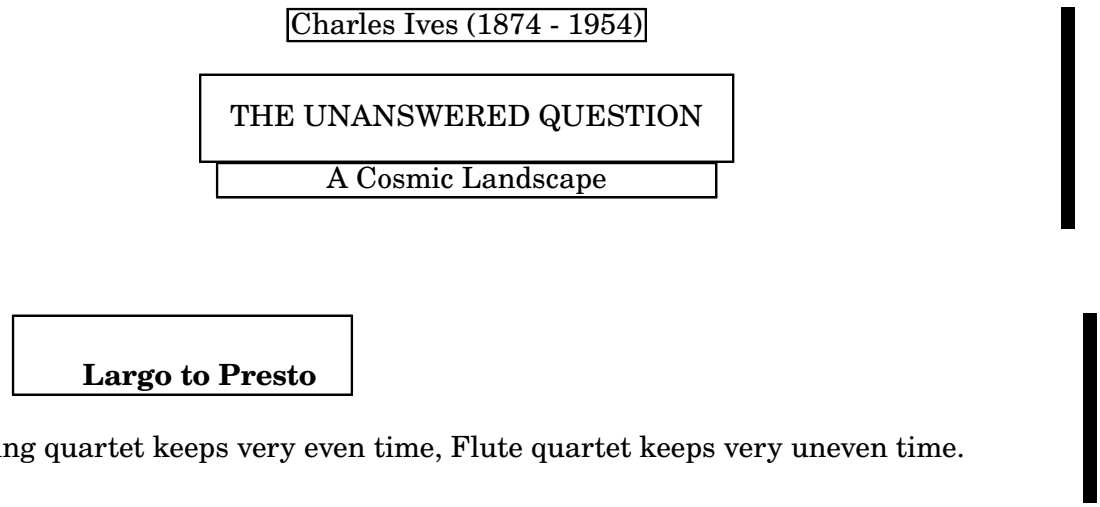
Certaines commandes de *markup* permettent d'ornementer des éléments textuels avec des graphismes, à l'instar de l'exemple suivant :

```
\markup \fill-line {
  \center-column {
    \circle Jack
    \box "in the box"
    \null
    \line {
      Erik Satie
      \hspace #3
      \bracket "1866 - 1925"
    }
    \null
    \rounded-box \bold Prelude
  }
}
```



Certaines directives peuvent nécessiter d'accroître l'espacement autour du texte – voir l'annexe Section A.8.2 [Align], page 519 pour une liste des différentes commandes particulières au mode *markup* ainsi que leur description.

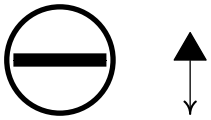
```
\markup \fill-line {
  \center-column {
    \box "Charles Ives (1874 - 1954)"
    \null
    \box \pad-markup #2 "THE UNANSWERED QUESTION"
    \box \pad-x #8 "A Cosmic Landscape"
    \null
  }
}
\markup \column {
  \line {
    \hspace #10
    \box \pad-to-box #'(-5 . 20) #'(0 . 5)
      \bold "Largo to Presto"
  }
  \pad-around #3
      "String quartet keeps very even time,
Flute quartet keeps very uneven time."
}
```



Vous pouvez imprimer certains graphismes ou symboles sans qu'il n'y ait de texte. Ces objets peuvent même se combiner, à l'instar de n'importe quelle expression *markup*.

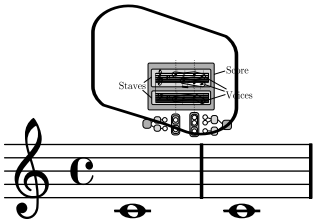
```
\markup {
  \combine
    \draw-circle #4 #0.4 ##f
    \filled-box #'(-4 . 4) #'(-0.5 . 0.5) #1
  \hspace #5

  \center-column {
    \triangle ##t
    \combine
      \draw-line #'(0 . 4)
      \arrow-head #Y #DOWN ##f
  }
}
```



Des fonctionnalités graphiques avancées vous permettent même d'inclure dans une partition un fichier image converti au format PostScript encapsulé (extension `eps`), ou bien de tracer un graphique directement dans le fichier source à partir d'instructions PostScript pures. Nous vous conseillons, en pareil cas, de toujours spécifier les dimensions du dessin, comme dans ce qui suit :

```
c1^\markup {
  \combine
    \epsfile #X #10 #"./context-example.eps"
    \with-dimensions #'(0 . 6) #'(0 . 10)
    \postscript #"
      -2 3 translate
      2.7 2 scale
      newpath
      2 -1 moveto
      4 -2 4 1 1 arct
      4 2 3 3 1 arct
      0 4 0 3 1 arct
      0 0 1 -1 1 arct
      closepath
      stroke"
  }
c
```



L'annexe Section A.8.3 [Graphic], page 534 répertorie les différentes commandes en matière de graphisme.

Voir aussi

Manuel de notation : Section A.8.3 [Graphic], page 534, Section 1.7 [Annotations éditoriales], page 182, Section A.8.2 [Align], page 519.

Morceaux choisis : Section "Texte" dans *Morceaux choisis*.

Référence des propriétés internes : Section "TextScript" dans *Référence des propriétés internes*.

Fichiers d'initialisation : '`scm/define-markup-commands.scm`', '`scm/stencil.scm`'.

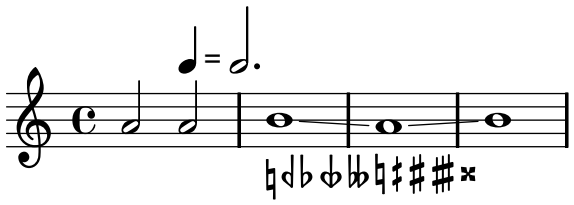
Notation musicale dans du texte formaté

Divers éléments de notation peuvent orner une partition, au moyen d'un objet *markup*.

Notes et altérations sont données à l'aide d'instructions *markup* :
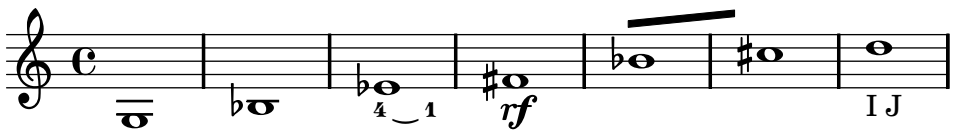
```
a2 a^\markup {
  \note #"4" #1
  =
  \note-by-number #1 #1 #1.5
}
```

```
b1_\markup {
  \natural \semiflat \flat
  \sesquiflat \doubleflat
}
\glissando
a1_\markup {
  \natural \semisharp \sharp
  \sesquisharp \doublesharp
}
\glissando b
```



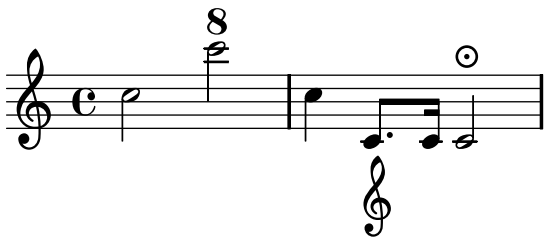
Le mode *markup* permet d'accéder à d'autres objets de notation :

```
g1 bes
ees-\markup {
  \finger 4
  \tied-lyric #"~"
  \finger 1
}
fis_\markup { \dynamic rf }
bes^\markup {
  \beam #8 #0.1 #0.5
}
cis
d-\markup {
  \markalphabet #8
  \markletter #8
}
```



En règle générale, tout symbole musical peut être inclus dans un *markup*, comme le montre l'exemple qui suit. Ces différents symboles sont répertoriés dans l'annexe Section A.6 [La fonte Feta], page 491.

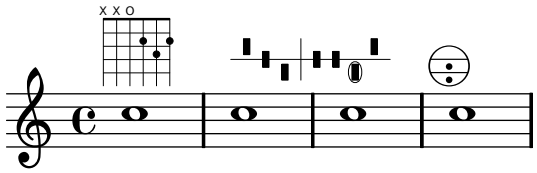
```
c2
c'^\markup { \musicglyph #"eight" }
c,4
c,8._\markup { \musicglyph #"clefs.G_change" }
c16
c2^\markup { \musicglyph #"timesig.neomensural94" }
```



Le sous-chapitre [Tout savoir sur les fontes], page 216 contient d'autres informations sur l'impression de glyphes non alphabétiques, tels que des crochets ou accolades.

Le mode *markup* supporte aussi les diagrammes spécifiques à certains instruments :
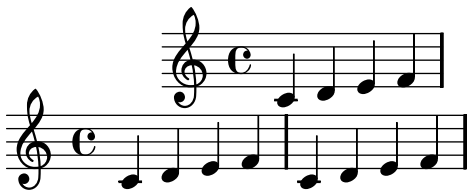
```
c1^\markup {
  \fret-diagram-terse #"x;x;o;2;3;2;"
}
c^\markup {
  \harp-pedal #"^-v|--ov^"
}
c
c^\markup {
  \combine
    \musicglyph #"accordion.discant"
    \combine
      \raise #0.5 \musicglyph #"accordion.dot"
      \raise #1.5 \musicglyph #"accordion.dot"
}
```



La documentation sur ces diagrammes se trouvent à l'annexe Section A.8.5 [Instrument Specific Markup], page 543.

Rien ne s'oppose à ce qu'une étiquette ne comporte une partition. Il suffit que l'expression en question contienne un bloc `\score` incluant un bloc `\layout`.

```
c4 d^\markup {
  \score {
    \relative c' { c4 d e f }
    \layout { }
  }
}
e f |
c d e f
```



Les différentes commandes *markup* relatives à la notation musicale sont répertoriées à l'annexe Section A.8.4 [Music], page 540.

Voir aussi

Manuel de notation : Section A.8.4 [Music], page 540, Section A.6 [La fonte Feta], page 491, [Tout savoir sur les fontes], page 216.

Morceaux choisis : Section “Texte” dans *Morceaux choisis*.

Référence des propriétés internes : Section “TextScript” dans *Référence des propriétés internes*.

Fichiers d’initialisation : ‘`scm/define-markup-commands.scm`’, ‘`scm/fret-diagrams.scm`’, ‘`scm/harp-pedals.scm`’.

Texte avec sauts de page

Alors que `\markup` s’utilise pour traiter un bloc de texte insécable, `\markuplines` permet, employé en tête de partition, d’obtenir un bloc de lignes réparties différemment et, le cas échéant, sur plusieurs pages.

```
\markuplines {
  \justified-lines {
    Un long texte constitué de lignes justifiées.
    ...
  }
  \wordwrap-lines {
    Un autre grand paragraphe.
    ...
  }
  ...
}
```

Un long texte constitué de lignes justifiées. ...

Un autre grand paragraphe. ...

...

Cette syntaxe prend en charge une liste de *markups* ; il peut s’agir

- d’une suite de commandes générant à leur tour des lignes de texte,
- d’une liste de lignes de texte,
- d’une liste d’étiquettes.

Les différentes commandes permettant de générer des listes de lignes se trouve dans l’annexe Section A.9 [Text markup list commands], page 552.

Voir aussi

Manuel de notation : Section A.9 [Text markup list commands], page 552,

Morceaux choisis : Section “Texte” dans *Morceaux choisis*.

Référence des propriétés internes : Section “TextScript” dans *Référence des propriétés internes*.

Fichiers d’initialisation : ‘`scm/define-markup-commands.scm`’.

Commandes prédéfinies

`\markuplines`.

1.8.3 Fontes

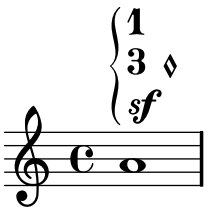
Ce chapitre est consacré aux fontes et polices de caractère, à leur gestion. Vous y apprendrez aussi comment changer de fonte en cours de partition.

Tout savoir sur les fontes

La gestion des fontes est assurée par plusieurs bibliothèques : FontConfig se charge de répertorier les différentes fontes installées sur votre système ; quant à Pango, elle se charge plus particulièrement de leur rendu.

Les fontes musicales peuvent se décrire comme un jeu de glyphes spécifiques regroupés dans plusieurs familles. L'exemple qui suit montre la syntaxe à utiliser pour accéder, en mode *markup*, aux différentes fontes `feta` non textuelles de LilyPond.

```
a1^\markup {
  \vcenter {
    \override #'(font-encoding . fetaBraces)
    \lookup #"brace120"
    \override #'(font-encoding . fetaText)
    \column { 1 3 sf }
    \override #'(font-encoding . fetaMusic)
    \lookup #"noteheads.s0petrucci"
  }
}
```



Tous ces glyphes, à l'exception des accolades qui sont regroupées dans `fetaBraces`, sont accessibles avec la syntaxe indiquée dans [Notation musicale dans du texte formaté], page 212.
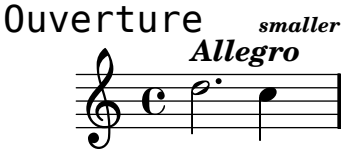
Une remarque s'impose au sujet des glyphes contenus dans `fetaBraces` : chacun d'eux est formé du mot *brace* auquel est accolé un numéro d'ordre. Lorsque vous désirez imprimer une accolade, vous devez la « chercher » par son numéro d'ordre – d'où la fonction `\lookup` de l'exemple ci-dessus –, tout en sachant qu'il est compris entre `0` (la plus petite) et `575` (la plus grande). Vous serez souvent amené à procéder par tatonnement pour arriver au résultat optimal. Par ailleurs, `fetaBraces` ne comporte que des accolades ouvrantes. Pour obtenir une accolade fermante, il suffit d'appliquer une rotation au glyphe sélectionné, comme indiqué au chapitre Section 5.4.9 [Rotation des objets], page 476.

Vous disposez de trois familles de fontes textuelles : *roman* pour la police sérif ou avec empattement – fixée par défaut à New Century Schoolbook –, une police sans empattement (*sans* sérif) et une à chasse fixe (ou monospace) – les deux dernières étant déterminées par l'installation de Pango.

Chaque famille dispose en principe de différents styles et niveaux de gras. L'exemple qui suit illustre la manière de changer la famille, le style, la graisse ou la taille. Notez bien que l'argument fourni à `font-size` correspond à la correction à apporter à la taille par défaut.

```
\override Score.RehearsalMark #'font-family = #'typewriter
\mark \markup "Ouverture"
\override Voice.TextScript #'font-shape = #'italic
\override Voice.TextScript #'font-series = #'bold
```

```
d2.^\markup "Allegro"
\override Voice.TextScript #'font-size = #-3
c4^smaller
```



Cette syntaxe s'applique aussi en mode *markup*, bien que celui-ci dispose d'une syntaxe allégée comme nous l'avons vu dans [Sélection de la fonte et de la taille], page 204 :

```
\markup {
  \column {
    \line {
      \override #'(font-shape . italic)
      \override #'(font-size . 4)
      Idomeneo,
    }
    \line {
      \override #'(font-family . typewriter)
      {
        \override #'(font-series . bold)
        re
        di
      }
      \override #'(font-family . sans)
      Creta
    }
  }
}
```



En plus de pouvoir jongler entre les différentes fontes prédéfinies, LilyPond vous permet d'en utiliser d'autres, ce qui fait l'objet des deux prochaines parties : [Attribution d'une fonte en particulier], page 217 et [Choix des fontes par défaut], page 218.

Voir aussi

Manuel de notation : Section A.6 [La fonte Feta], page 491, [Notation musicale dans du texte formaté], page 212, Section 5.4.9 [Rotation des objets], page 476, [Sélection de la fonte et de la taille], page 204, Section A.8.1 [Font], page 511.
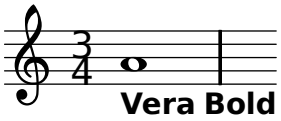
Attribution d'une fonte en particulier

Vous pouvez utiliser n'importe quelle police installée sur votre système, dès lors qu'elle est accessible par Fontconfig et que vous respectez la syntaxe suivante :

```
\override Staff.TimeSignature #'font-name = #"Bitstream Charter"
\override Staff.TimeSignature #'font-size = #2
\time 3/4

a1_\markup {
```

```
  \override #'(font-name . "Vera Bold")
    { Vera Bold }
}
```



Pour obtenir la liste de toutes les polices disponibles sur votre machine, lancez

```
lilypond -dshow-available-fonts toto
```

(quel qu'il soit, le dernier argument est obligatoire).

Voir aussi

Manuel de notation : [Tout savoir sur les fontes], page 216, [Choix des fontes par défaut], page 218.

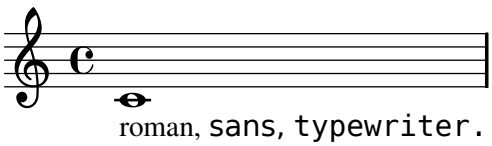
Morceaux choisis : Section "Texte" dans *Morceaux choisis*.

Choix des fontes par défaut

Vous pouvez tout à fait préférer un autre jeu de polices par défaut que celui de LilyPond. Il vous faudra alors spécifier les différentes familles, en respectant l'ordre *roman*, *sans empattement* et *monospace*, comme dans l'exemple suivant. Pour plus d'explications sur les fontes, relisez [Tout savoir sur les fontes], page 216.

```
\paper  {
  myStaffSize = #20
  #(define fonts
    (make-pango-font-tree "Times New Roman"
                          "Nimbus Sans"
                          "Luxi Mono"
                           (/ myStaffSize 20)))
}

\relative c'{
  c1-\markup {
    roman,
    \sans sans,
    \typewriter typewriter. }
}
```



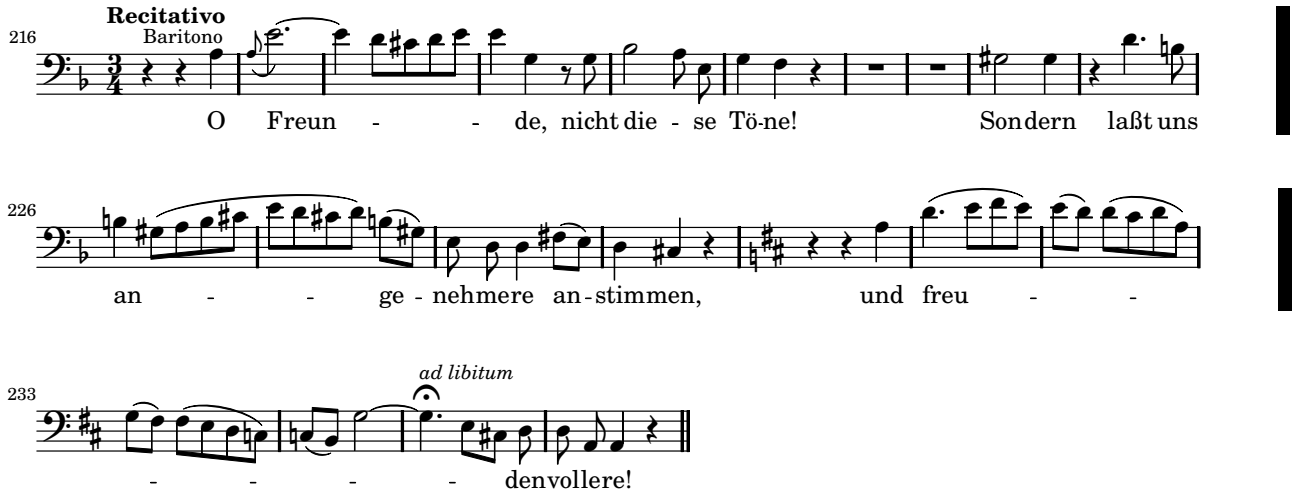
Voir aussi

Manuel de notation : [Tout savoir sur les fontes], page 216, [Attribution d'une fonte en particulier], page 217, [Sélection de la fonte et de la taille], page 204, Section A.8.1 [Font], page 511.

2 Notation spécialisée

Ce chapitre explique comment créer la notation musicale spécifique à certains intruments ou certaines époques.

2.1 Musique vocale



Ce chapitre traite de la musique vocale : comment la saisir et comment s'assurer que les paroles s'alignent avec les notes de la mélodie correspondante.

2.1.1 Vue d'ensemble de la musique vocale

En complément de généralités, ce sous-chapitre aborde quelques styles particuliers en terme de musique vocale.

Références en matière de musique vocale

Graver de la musique vocale soulève plusieurs problèmes ; ils sont abordés soit dans ce chapitre, soit dans d'autres parties de la documentation de LilyPond.

- La plupart du temps, les paroles ne sont constituées que de texte simple. Cette forme de notation est abordée dans Section "Écriture de chants simples" dans *Manuel d'initiation*.
- La musique vocale nécessite souvent de recourir au mode `markup`, aussi bien pour des paroles que pour d'autres éléments textuels comme le nom des personnages. Cette syntaxe est expliquée dans [Introduction au formatage de texte], page 202.
- L'impression d'un *ambitus* – ou tessiture – que l'on trouve en tête de certaines partitions, est abordée dans [Ambitus], page 27.
- Les indications de nuance viennent, par défaut, se placer sous la portée. Il en va différemment pour la musique vocale, de telle sorte qu'elles ne soient pas mélangées avec les paroles. Ceci fait l'objet de la rubrique [Mise en forme d'une partition chorale], page 255.

Voir aussi

Glossaire musicologique : Section "ambitus" dans *Glossaire*.

Manuel d'initiation : Section "Écriture de chants simples" dans *Manuel d'initiation*.

Manuel de notation : [Introduction au formatage de texte], page 202, [Ambitus], page 27, [Mise en forme d'une partition chorale], page 255.

Morceaux choisis : Section "Musique vocale" dans *Morceaux choisis*.

Saisie des paroles

Il existe un mode de saisie spécialement adapté aux paroles. On l'introduit avec le mot-clé `\lyricmode`, ou en utilisant `\addlyrics` ou `\lyricsto`. Ce mode vous permet de saisir des paroles ainsi que leur ponctuation, de telle sorte que le caractère `a` ne sera plus interprété comme une note, un *la* pour les non latinistes, mais comme une syllabe. Les syllabes sont saisies comme des notes, mais les hauteurs sont alors remplacées par du texte. Exemple avec une comptine anglaise :

```
\lyricmode { Three4 blind mice,2 three4 blind mice2 }
```

Il y a deux manières principales de préciser la place exacte des syllabes : soit en spécifiant explicitement la durée de chaque syllabe – comme dans l'exemple ci-dessus – soit en alignant automatiquement les paroles sur les notes d'une mélodie ou d'une voix en utilisant `\addlyrics` ou `\lyricsto`. La première méthode est abordée plus en détail à la rubrique [Durée explicite des syllabes], page 225, et la deuxième à la rubrique [Durée automatique des syllabes], page 223.

Dans les paroles, un mot ou une syllabe commence par une lettre de l'alphabet, et se termine par une espace. Toute syllabe doit donc être séparée d'une autre par une espace, tout autre caractère – chiffre ou ponctuation – étant considéré comme partie intégrante de cette même syllabe. L'exemple suivant comporte une faute de frappe évidente :

```
\lyricmode { lah lah lah}
```

la dernière syllabe contient une `}` ; il y a de fait un défaut de parité avec l'accolade ouvrante, et la compilation échouera fort probablement. Prenez dès à présent l'habitude de toujours encadrer d'espaces une accolade :

```
\lyricmode { lah lah lah }
```

De la même manière, un point concluant une suite de lettres sera partie intégrante de la chaîne résultante. Par conséquent, il est impératif d'insérer des espaces lorsque vous modifiez les propriétés d'une commande. En d'autres termes, **ne saisissez pas**
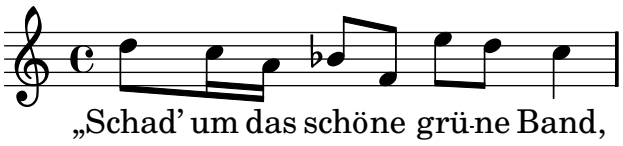
```
\override Score.LyricText #'font-shape = #'italic
```

mais plutôt

```
\override Score . LyricText #'font-shape = #'italic
```
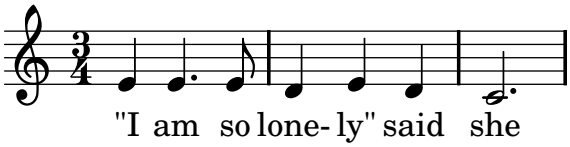
Pour utiliser des lettres accentuées ou des caractères spéciaux – cœurs ou guillemets inversés par exemple – il suffit de les insérer dans le fichier et de veiller à le sauvegarder avec le codage UTF-8. Voir à ce sujet Section 3.3.3 [Codage du texte], page 399 pour plus de détails.

```
\relative c'' { d8 c16 a bes8 f e' d c4 }
\addlyrics { „Schad' um das schö -- ne grü -- ne Band, }
```



Pour utiliser des guillemets informatiques standard, faites-les précéder d'une barre oblique inverse et encadrez d'une paire de guillemets la syllabe ainsi composée :

```
\relative c' { \time 3/4 e4 e4. e8 d4 e d c2. }
\addlyrics { "\"I" am so lone- "ly\"" said she }
```



Expliquer exactement comment LilyPond repère le début d'un mot en mode paroles (*Lyrics*) est quelque peu compliqué. En mode *Lyrics*, un mot peut commencer par : tout caractère alphabétique, `_`, `?`, `!`, `:`, `'`, un des codes de contrôle `^A` à `^F` et `^Q` à `^W`, `^Y`, `^^`, tout caractère ASCII de code strictement supérieur à 127, ou enfin l'un des symboles `` ` ``, `'`, `"`, ou `^`, s'il est précédé d'une barre oblique inverse.

Voir aussi

Manuel d'initiation : Section "Chansons" dans *Manuel d'initiation*

Manuel de notation : [Durée automatique des syllabes], page 223, [Durée explicite des syllabes], page 225, Section 1.8.3 [Fontes], page 216, Section "Modes de saisie" dans *Manuel de notation*.

Référence des propriétés internes : Section "LyricText" dans *Référence des propriétés internes*.

Alignement des paroles sur la mélodie

L'impression des paroles est réalisée à partir de l'interprétation d'un contexte `Lyrics` – voir Section 5.1.1 [Tout savoir sur les contextes], page 463.

```
\new Lyrics \lyricmode { ... }
```

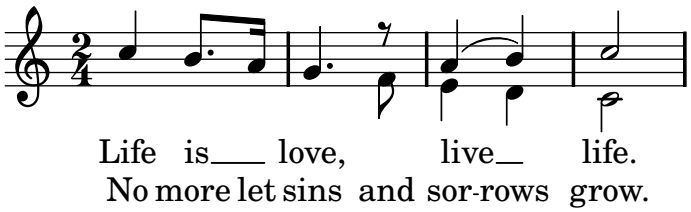
Vous disposez de deux méthodes pour aligner des paroles sur une mélodie :

- Les paroles peuvent s'aligner automatiquement, la durée des syllabes étant déterminée à partir d'un contexte de voix ou, dans certaines circonstance, une mélodie associée, grâce aux commandes `\addlyrics` et `\lyricsto` ou en définissant la propriété `associatedVoice`. Ceci est détaillé à la rubrique [Durée automatique des syllabes], page 223.

```
<<
  \new Staff <<
    \time 2/4
    \new Voice = "one" \relative c'' {
      \voiceOne
      c4 b8. a16 g4. r8 a4 ( b ) c2
    }
    \new Voice = "two" \relative c' {
      \voiceTwo
       s2 s4. f8 e4 d c2
    }
  >>

% takes durations and alignment from notes in "one"
  \new Lyrics \lyricsto "one" {
    Life is __ _ love, live __ life.
  }

% takes durations and alignment from notes in "one" initially
% then switches to "two"
  \new Lyrics \lyricsto "one" {
    No more let
    \set associatedVoice = "two"  % must be set one syllable early
    sins and sor -- rows grow.
  }
>>
```



La première ligne de paroles est saisie selon la méthode la plus simple.

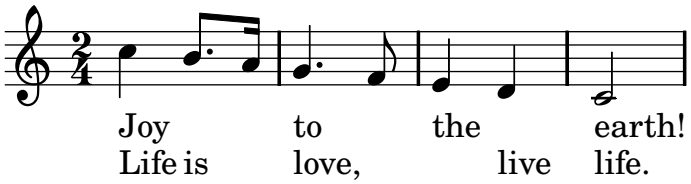
Vous pouvez constater, dans la deuxième ligne, que les paroles s'alignent selon les durées d'une voix différente. Ceci est particulièrement utile lorsque le texte s'agence différemment selon les couplets et que les durées sont accessibles grâce à des contextes `Voice` particuliers. Pour de plus amples détails, rendez-vous à la rubrique Section 2.1.3 [Couplets], page 247.

- Les paroles s'aligneront indépendamment de la valeur des notes dès lors que vous utiliserez le mode `\lyricmode` et affecterez explicitement leur durée à chaque syllabe.

```
<<
  \new Voice = "one" \relative c'' {
    \time 2/4
    c4 b8. a16 g4. f8 e4 d c2
  }

% uses previous explicit duration of 2;
  \new Lyrics \lyricmode {
    Joy to the earth!
  }

% explicit durations, set to a different rhythm
  \new Lyrics \lyricmode {
    Life4 is love,2. live4 life.2
  }
>>
```



La première ligne de paroles ne s'aligne pas vraiment sur les notes parce qu'aucune durée n'a été spécifiée. En fait, Lilypond adopte la dernière durée mentionnée, un 2, et l'applique à chaque mot.

La deuxième ligne illustre la manière d'aligner des paroles sans tenir compte de la durée des notes. Cette façon de procéder permet de traiter un alignement différent selon les couplets lorsqu'il n'y a pas moyen de déduire les durées à partir d'un contexte musical ; la rubrique [Durée explicite des syllabes], page 225 aborde ceci plus en détails. Cette technique permet aussi d'ajouter des dialogues, comme indiqué à la rubrique [Dialogue et musique], page 264.

Des paroles saisies de cette manière s'aligneront par défaut sur la gauche des notes ; elles seront centrées sur les notes de la mélodie dès lors que vous pourrez les associer à une voix. Tout ceci est abordé plus en détails à la rubrique [Durée explicite des syllabes], page 225.

Voir aussi

Manuel d'initiation : Section "Alignement des paroles sur une mélodie" dans *Manuel d'initiation*.

Référence des propriétés internes : Section "Lyrics" dans *Référence des propriétés internes*.

Durée automatique des syllabes

Les paroles peuvent être automatiquement alignées sous une mélodie, de trois manières différentes :

- en utilisant la commande `\lyricsto` pour spécifier le contexte de voix qui contient la mélodie,
- en introduisant les paroles par la commande `\addlyrics`, placée juste après le contexte `Voice` qui contient la mélodie,
- en définissant la propriété `associatedVoice` pour synchroniser les paroles avec un autre contexte de voix, ce à n'importe quel moment.
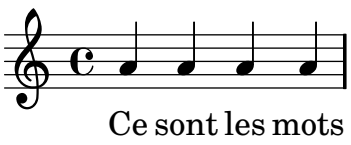
Ces trois méthodes permettent d'ajouter les traits d'union séparant les syllabes d'un même mot ainsi que d'indiquer la tenue de la dernière syllabe. Ceci fait l'objet de la rubrique [Traits d'union et de prolongation], page 230.

Le contexte `Voice` contenant la mélodie sur laquelle les paroles vont s'aligner doit rester actif, au risque de voir la suite du texte disparaître. Ceci peut se produire lorsqu'il y a des moments où l'on ne chante pas. La rubrique Section "Conservation d'un contexte" dans *Manuel de notation* vous indiquera comment maintenir un contexte actif.

Utilisation de `\lyricsto`

Vous pouvez aligner automatiquement des paroles sous une mélodie en spécifiant à l'aide de la commande `\lyricsto` le contexte de voix qui contient cette mélodie :

```
<<
  \new Voice = "melodie" {
    a4 a a a
  }
  \new Lyrics \lyricsto "melodie" {
    Ce sont les mots
  }
>>
```



Cette commande adapte les paroles aux notes de la voix (contexte `Voice` dans le jargon LilyPond) *melodie*. Ce contexte `Voice` doit exister avant l'affectation des paroles par `\lyricsto`. La commande `\lyricsto` introduit automatiquement le mode `\lyricmode` ; il n'est alors pas nécessaire de rajouter `\lyricmode`. Les paroles viendront par défaut se placer en dessous des notes ; la rubrique [Positionnement vertical des paroles], page 232 vous donnera des indications pour d'autres options.

Utilisation de `\addlyrics`

La commande `\addlyrics` n'est en fait qu'une manière plus aisée d'écrire de la musique vocale dans une structure Lilypond plus complexe.
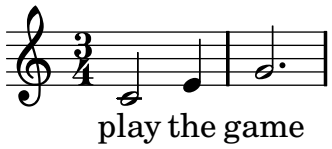
```
{ MUSIQUE }
\addlyrics { PAROLES }
```

revient au même que

```
\new Voice = "blah" { MUSIQUE }
\new Lyrics \lyricsto "blah" { PAROLES }
```

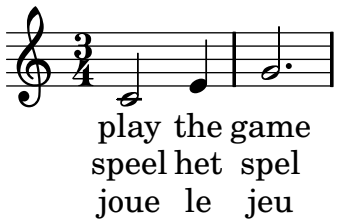
En voici un exemple :

```
{
  \time 3/4
  \relative c' { c2 e4 g2. }
  \addlyrics { play the game }
}
```



On peut ajouter davantage de couplets en multipliant le nombre de sections `\addlyrics`.

```
{
  \time 3/4
  \relative c' { c2 e4 g2. }
  \addlyrics { play the game }
  \addlyrics { speel het spel }
  \addlyrics { joue le jeu }
}
```



Cependant, la commande `\addlyrics` ne peut pas gérer les constructions polyphoniques. Dans ce cas, mieux vaut employer `\lyricsto`.

Utilisation de associatedVoice

La propriété `associatedVoice` permet de basculer de mélodie pour la synchronisation des paroles. Elle s'emploie de la manière suivante :
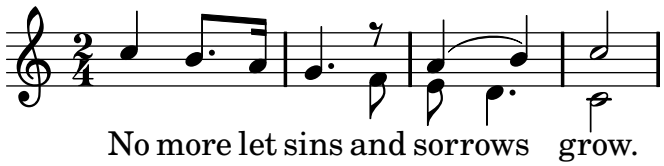
```
\set associatedVoice = #"lala"
```

La valeur que vous attribuez à cette propriété (ici `"lala"`) doit désigner un contexte `Voice` nommé, sans quoi les mélismes ne seront pas imprimés correctement.

Voici un exemple de cette manière de procéder :

```
<<
  \new Staff <<
    \time 2/4
    \new Voice = "one" \relative c'' {
      \voiceOne
      c4 b8. a16 g4. r8 a4 ( b ) c2
    }
    \new Voice = "two" \relative c' {
      \voiceTwo
       s2 s4. f8 e8 d4. c2
    }
  >>
% takes durations and alignment from notes in "one" initially
% then switches to "two"
```

```
  \new Lyrics \lyricsto "one" {
    No more let
    \set associatedVoice = "two"  % must be set one syllable early
    sins and sor -- rows grow.
  }
>>
```



Voir aussi

Manuel de notation : [Traits d'union et de prolongation], page 230, Section 5.1.3 [Conservation d'un contexte], page 464.
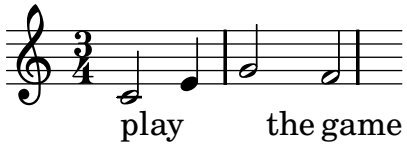
Durée explicite des syllabes

On peut aussi se passer de `\addlyrics`, `\lyricsto` et `associatedVoice` pour saisir des paroles. Dans ce cas, les syllabes sont entrées comme des notes – du texte remplaçant les hauteurs – ce qui veut dire que vous devez définir leur durée explicitement.

Par défaut, les syllabes seront alignées par la gauche sur l'instant musical. Les traits d'union seront imprimés entre les syllabes, à l'inverse des mélismes puisqu'il n'y a pas de voix associée.

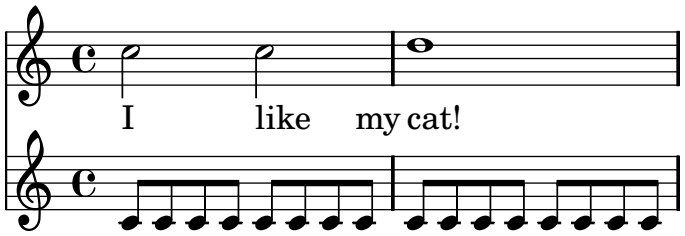
Voici deux illustrations de cette technique :

```
<<
  \new Voice = "melody" {
    \time 3/4
    c2 e4 g2 f
  }
  \new Lyrics \lyricmode {
    play1 the4 game4
  }
>>
```




```
<<
  \new Staff {
    \relative c'' {
      c2 c2
      d1
    }
  }
  \new Lyrics {
    \lyricmode {
      I2 like4. my8 cat!1
    }
  }
```
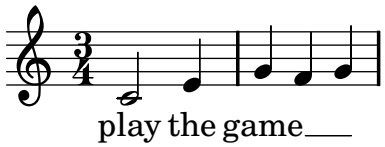
```
  \new Staff {
    \relative c' {
      c8 c c c c c c c
      c8 c c c c c c c
    }
  }
>>
```



Cette manière de procéder est tout à fait adaptée lorsqu'un fond musical accompagne des dialogues – voir [Dialogue et musique], page 264.

Les syllabes seront centrées par rapport à l'instant musical dès lors que vous aurez attribué à la propriété `associatedVoice` le nom du contexte `Voice` contenant les notes. Grâce à l'utilisation de `associatedVoice`, les doubles tirets `--` ou soulignés `__` seront rendus correctement en trait d'union ou indication de mélisme.

```
<<
  \new Voice = "melody" {
    \time 3/4
    c2 e4 g f g
  }
  \new Lyrics \lyricmode {
    \set associatedVoice = #"melody"
    play2 the4 game2. __
  }
>>
```



Voir aussi

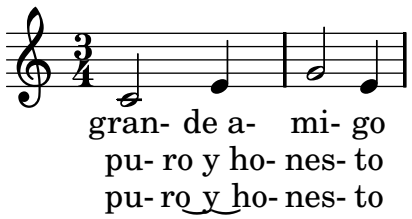
Manuel de notation : [Dialogue et musique], page 264.

Référence des propriétés internes : Section "Lyrics" dans *Référence des propriétés internes*, Section "Voice" dans *Référence des propriétés internes*.

Plusieurs syllabes sur une note

Pour attribuer plus d'une syllabe à une même note, vous pouvez soit les mettre entre guillemets, soit utiliser le caractère souligné (_) pour obtenir une espace, ou bien encore utiliser un tilde (~) pour obtenir une liaison entre les syllabes. Cette liaison adaptée aux paroles correspond au caractère Unicode `U+203F`, et n'apparaîtra dans la partition que si le système dispose d'une police installée qui contient ce symbole. Un certain nombre de fontes librement disponibles en disposent, comme FreeSerif (un clone de Times), 'DejaVuSans' (mais pas DejaVuSerif) ou TeXGyreSchola (un clone de Century Schoolbook).

```
{
  \time 3/4
  \relative c' { c2 e4 g2 e4 }
  \addlyrics { gran- de_a- mi- go }
  \addlyrics { pu- "ro y ho-" nes- to }
  \addlyrics { pu- ro~y~ho- nes- to }
}
```



Voir aussi

Référence des propriétés internes : Section “LyricCombineMusic” dans *Référence des propriétés internes*.

Plusieurs notes pour une même syllabe

Parfois, tout particulièrement dans la musique médiévale, plusieurs notes correspondent à une même syllabe. Ces vocalises sont appelées Section “mélismes” dans *Glossaire*. La syllabe à vocaliser est traditionnellement alignée sur la gauche de la première note du mélisme.
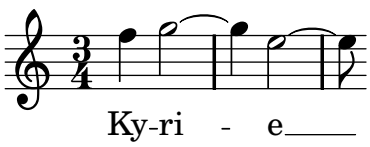
Lorsqu’un mélisme tombe sur une syllabe autre que la dernière d’un mot, un trait d’union étiré, indiqué par un double tiret `--` dans le fichier source, reliera cette syllabe à la suivante.

Lorsqu’un mélisme tombe sur la dernière syllabe d’un mot ou que ce mot n’en comporte qu’une, l’usage est d’indiquer la « tenue » jusqu’à la dernière note de la vocalise. Ceci s’obtient en ajoutant un double caractère souligné `__` après cette syllabe.

Vous disposez de cinq méthodes pour indiquer la présence d’un mélisme :
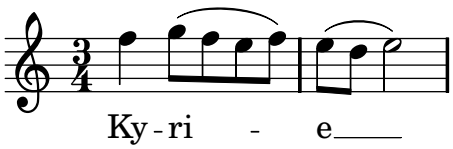
- Une liaison de prolongation constitue de fait un mélisme :

```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g2 ~ |
    g4 e2 ~ |
    e8
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e __
  }
>>
```
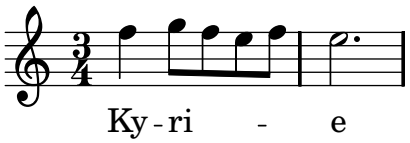


- LilyPond considère une liaison d’articulation comme un mélisme – il s’étendra de la première à la dernière note couvertes par cette liaison. Il s’agit là de la façon traditionnelle de saisir des paroles :

```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g8 ( f e f )
    e8 ( d e2 )
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e __
  }
>>
```



- LilyPond considère des notes regroupées par une ligature manuelle comme un mélisme, si tant est que la procédure de ligature automatique a été désactivée – voir [Définition des règles de ligature automatique], page 73.
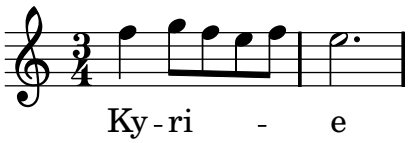
```
<<
  \new Voice = "melody" {
    \time 3/4
    \autoBeamOff
    f4 g8[ f e f]
    e2.
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e
  }
>>
```



Ceci ne peut, vous en conviendrez, prendre en compte des durées plus longues que la croche.
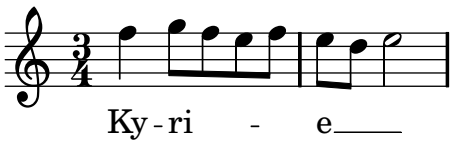
- LilyPond considère un groupe de notes non liées, mais encadrées par `\melisma` et `\melismaEnd`, comme constituant un mélisme :

```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g8
    \melisma
    f e f
    \melismaEnd
    e2.
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e
  }
>>
```
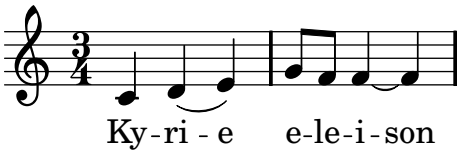


- Vous pouvez indiquer un mélisme directement dans les paroles, à l'aide d'un caractère souligné simple _, pour chaque note faisant partie de la vocalise :

```
<<
  \new Voice = "melody" {
    \time 3/4
    f4 g8 f e f
    e8 d e2
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- _ _ _ e __ _ _
  }
>>
```



Vous pouvez totalement désactiver l'interprétation des liaisons de prolongation ou d'articulation et des ligatures apparaissant dans une mélodie comme fait générateur d'un mélisme. Il suffit en ce cas de définir `melismaBusyProperties` :

```
<<
  \new Voice = "melody" {
    \time 3/4
    \set melismaBusyProperties = #'()
    c4 d ( e )
    g8 [ f ] f4 ~ f
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- e e -- le -- i -- son
  }
>>
```



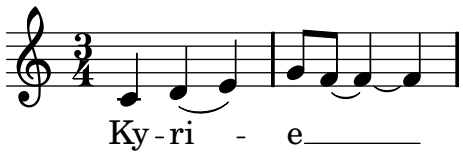
Certains réglages de `melismaBusyProperties` permettent de prendre en compte ou non les liaisons de tenue, les liaisons d'articulation et les ligatures dans la détection automatique des mélismes – voir `melismaBusyProperties` à la rubrique Section "Tunable context properties" dans *Référence des propriétés internes*.

Dans le cas où les indications de mélisme doivent être totalement ignorées, il vous faudra alors activer `ignoreMelismata` – voir [Rythme différent selon le couplet], page 248.

Lorsque, dans un passage où la propriété `melismaBusyProperties` est active, survient un mélisme, vous pouvez l'indiquer dans les paroles par un simple caractère souligné pour chaque note à inclure :

```
<<
  \new Voice = "melody" {
```

```
    \time 3/4
    \set melismaBusyProperties = #'()
    c4 d ( e )
    g8 [ f ] ~ f4 ~ f
  }
  \new Lyrics \lyricsto "melody" {
    Ky -- ri -- _ e __ _ _ _
  }
>>
```



Commandes prédéfinies

`\autoBeamOff`, `\autoBeamOn`, `\melisma`, `\melismaEnd`.

Voir aussi

Glossaire musicologique : Section “melisma” dans *Glossaire*.

Manuel d’initiation : Section “Alignement des paroles sur une mélodie” dans *Manuel d’initiation*.

Manuel de notation : [Alignement des paroles sur la mélodie], page 221, [Durée automatique des syllabes], page 223, [Définition des règles de ligature automatique], page 73, [Rythme différent selon le couplet], page 248.

Référence des propriétés internes : Section “Tunable context properties” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Certains mélismes ne sont pas détectés automatiquement ; vous devrez alors prolonger vous-même les syllabes concernées à l’aide d’un double caractère souligné.

Traits d’union et de prolongation

Un mélisme sur la dernière syllabe d’un mot est indiqué par une longue ligne horizontale basse s’étirant jusqu’à syllabe suivante. Une telle ligne, que nous appellerons prolongateur ou extenseur, s’obtient en saisissant ‘ `__` ’ (notez les espaces entourant le double caractère souligné).

> **Note :** Dans une partition, les mélismes, ou vocalises, sont matérialisés par une ligne de prolongation. On l’indique par un double caractère souligné. Lorsqu’ils sont assez courts, ces mélismes peuvent s’indiquer par un souligné unique, ce qui aura pour effet de sauter une note à chaque fois et de ne pas imprimer de ligne.

Un trait d’union séparant deux syllabes d’un même mot s’obtient en saisissant ‘ `--` ’ (notez les espaces entourant le tiret double). Ce trait d’union sera centré entre les deux syllabes et sa longueur sera proportionnelle à l’espace les séparant.

Dans les partitions très serrées, les traits d’union peuvent ne pas être imprimés. Cet inconvénient peut être contrôlé par les propriétés `minimum-distance` pour gérer l’espace minimum entre deux syllabes, et `minimum-length`, seuil en deçà duquel il n’y a pas de trait d’union, toutes deux attachées à l’objet `LyricHyphen`.

Voir aussi
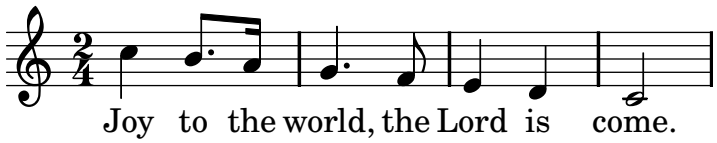
Référence des propriétés internes : Section “LyricExtender” dans *Référence des propriétés internes*, Section “LyricHyphen” dans *Référence des propriétés internes*

2.1.2 Situations particulières en matière de paroles

Travail avec des paroles et variables

Vous pouvez créer des variables pour contenir les paroles, dès lors que vous faites appel au mode approprié :

```
musicOne = \relative c'' {
  c4 b8. a16 g4. f8 e4 d c2
}
verseOne = \lyricmode {
  Joy to the world, the Lord is come.
}
\score {
  <<
    \new Voice = "one" {
      \time 2/4
      \musicOne
    }
    \new Lyrics \lyricsto "one" {
      \verseOne
    }
  >>
}
```
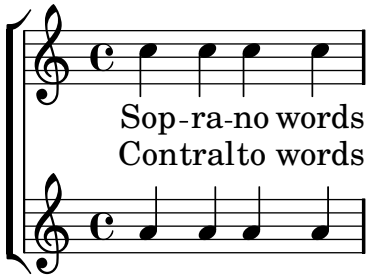


La fonction `\lyricmode` permet de définir une variable pour les paroles. Point n’est besoin de spécifier les durées si vous utilisez `\addlyrics` ou `\lyricsto` lorsque vous y faites référence.

Pour une organisation différente ou plus complexe, mieux vaut commencer par créer et alimenter les variables contenant mélodies et paroles, puis définir la hiérarchie des portées et des lignes de paroles, et enfin combiner correctement mélodies et paroles à l’aide de la commande `\context`. Vous serez ainsi assuré que la voix à laquelle il est fait référence par `\lyricsto` aura bien été préalablement définie, comme dans l’exemple suivant :

```
sopranoMusic = \relative c'' { c4 c c c }
contraltoMusic = \relative c'' { a4 a a a }
sopranoWords = \lyricmode { Sop -- ra -- no words }
contraltoWords = \lyricmode { Con -- tral -- to words }

\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \sopranoMusic
      }
    }
```

```
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos"
    \new Staff {
      \new Voice = "contraltos" {
        \contraltoMusic
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        \sopranoWords
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        \contraltoWords
      }
    }
  >>
}
```



Voir aussi

Manuel de notation : [Positionnement vertical des paroles], page 232.

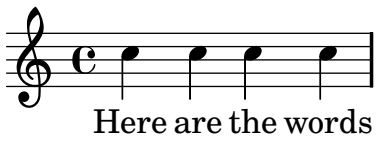
Référence des propriétés internes : Section “LyricCombineMusic” dans *Référence des propriétés internes*, Section “Lyrics” dans *Référence des propriétés internes*.

Positionnement vertical des paroles

Selon le type de musique, les paroles apparaîtront au-dessus ou au-dessous d’une portée ou bien entre deux portées. Positionner des paroles en dessous de la portée à laquelle elles se rattachent est de loin la chose la plus simple : il suffit de mentionner le contexte de paroles après le contexte de portée :

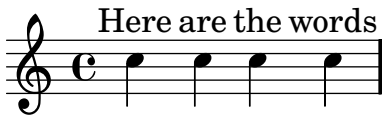
```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here are the words
      }
    }
```

```
  >>
}
```



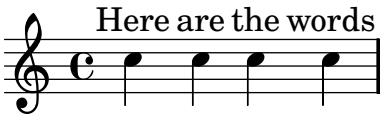
Positionner les paroles au-dessus de la portée se fait de deux manières différentes, le plus simple étant d'utiliser la même syntaxe que ci-dessus, à ceci près que la ligne de paroles sera positionnée de manière explicite :

```
\score {
  <<
    \new Staff = "staff" {
      \new Voice = "melody" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics \with { alignAboveContext = "staff" } {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```
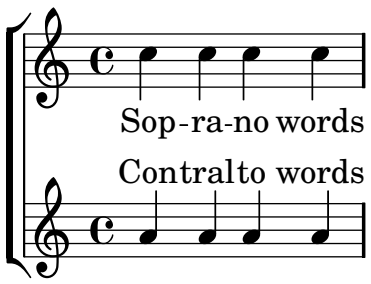


Autre façon de procéder, cette fois-ci en deux étapes. Nous commençons par déclarer un contexte `Lyrics` que nous laissons vide, puis les contextes `Staff` et `Voice`. Dans un deuxième temps, nous ajoutons l'instruction `\context` et la commande `\lyricsto` pour affecter les paroles au contexte de voix en question. Voici comment cela se présente :

```
\score {
  <<
    \new Lyrics = "lyrics" \with {
      % lyrics above a staff should have this override
      \override VerticalAxisGroup #'staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "melody" {
        \relative c'' { c4 c c c }
      }
    }
    \context Lyrics = "lyrics" {
      \lyricsto "melody" {
        Here are the words
      }
    }
  >>
}
```



Lorsque deux voix sont isolées chacune sur une portée, vous pouvez placer les paroles entre les deux portées en utilisant l'une des méthodes que nous venons de voir. En voici un exemple, basé sur la deuxième méthode :

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice = "sopranos" {
        \relative c'' { c4 c c c }
      }
    }
    \new Lyrics = "sopranos"
    \new Lyrics = "contraltos" \with {
      % lyrics above a staff should have this override
      \override VerticalAxisGroup #'staff-affinity = #DOWN
    }
    \new Staff {
      \new Voice = "contraltos" {
        \relative c'' { a4 a a a }
      }
    }
    \context Lyrics = "sopranos" {
      \lyricsto "sopranos" {
        Sop -- ra -- no words
      }
    }
    \context Lyrics = "contraltos" {
      \lyricsto "contraltos" {
        Con -- tral -- to words
      }
    }
  >>
}
```



Vous pouvez générer d'autres combinaisons de paroles et portées à partir de ces exemples, ou en examinant les Section "modèles" dans *Manuel d'initiation* inclus dans le manuel d'initiation.

Morceaux choisis

Espacement des paroles selon les pratiques de la version 2.12

La version 2.14 a donné naissance à un nouveau moteur pour l'espacement vertical des paroles. Celles-ci peuvent donc se retrouver positionnées différemment. Le moteur adoptera les usages de la version 2.12 une fois que vous aurez réglé certaines propriétés des contextes `Lyric` et `Staff`.

```
global = {
  \key d \major
  \time 3/4
}

sopMusic = \relative c' {
  % VERSE ONE
  fis4 fis fis | \break
  fis4. e8 e4
}

altoMusic = \relative c' {
  % VERSE ONE
  d4 d d |
  d4. b8 b4 |
}

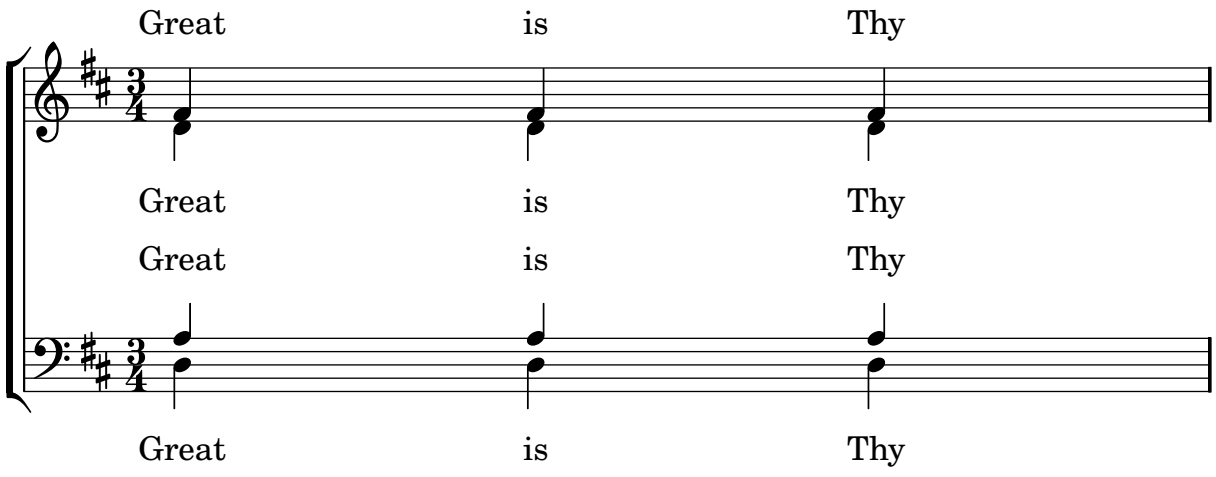
tenorMusic = \relative c' {
  a4 a a |
  b4. g8 g4 |
}

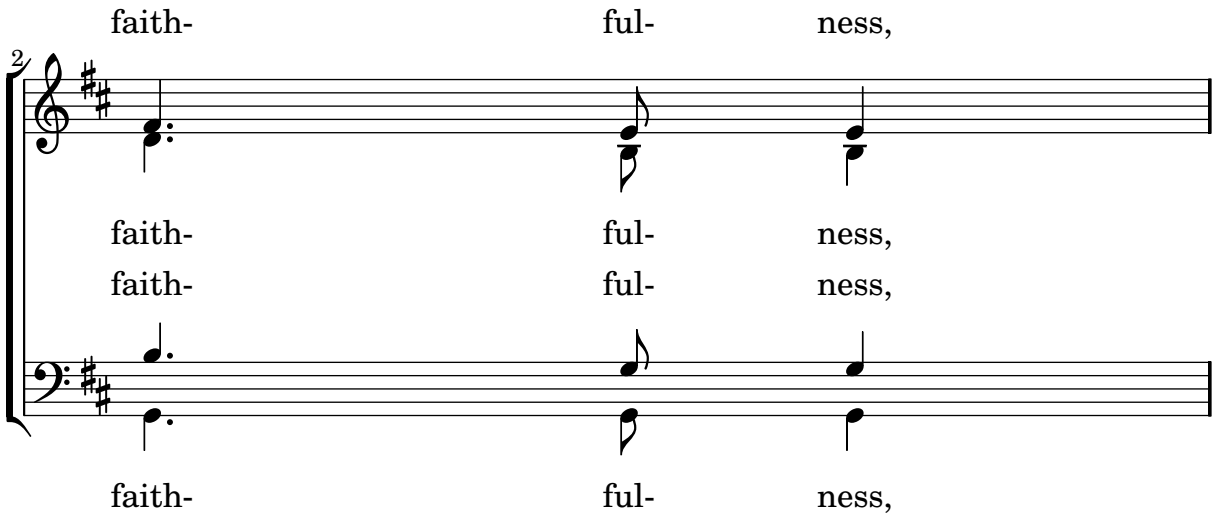
bassMusic = \relative c {
  d4 d d |
  g,4. g8 g4 |
}

words = \lyricmode {
  Great is Thy faith- ful- ness,
}

\score {
  \new ChoirStaff <<
    \new Lyrics = sopranos
    \new Staff = women <<
      \new Voice = "sopranos" {
        \voiceOne
        \global \sopMusic
      }
      \new Voice = "altos" {
        \voiceTwo
        \global \altoMusic
      }
    >>
    \new Lyrics = "altos"
    \new Lyrics = "tenors"
    \new Staff = men <<
      \clef bass
      \new Voice = "tenors" {
        \voiceOne
        \global \tenorMusic
      }
```

```
      \new Voice = "basses" {
        \voiceTwo  \global \bassMusic
      }
    >>
    \new Lyrics = basses
    \context Lyrics = sopranos \lyricsto sopranos \words
    \context Lyrics = altos \lyricsto altos \words
    \context Lyrics = tenors \lyricsto tenors \words
    \context Lyrics = basses \lyricsto basses \words
  >>
  \layout {
    \context {
      \Lyrics
      \override VerticalAxisGroup #'staff-affinity = ##f
      \override VerticalAxisGroup #'staff-staff-spacing =
        #'((basic-distance . 0)
           (minimum-distance . 2)
           (padding . 2))
    }
    \context {
      \Staff
      \override VerticalAxisGroup #'staff-staff-spacing =
        #'((basic-distance . 0)
           (minimum-distance . 2)
           (padding . 2))
    }
  }
}
```





Voir aussi

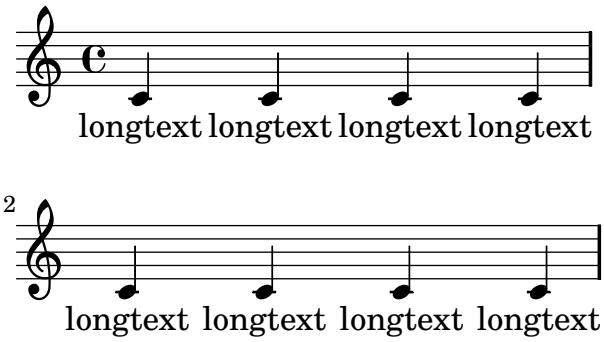
Manuel d'initiation : Section "Ensemble vocal" dans *Manuel d'initiation*.

Manuel de notation : Section 5.1.7 [Alignement des contextes], page 468, Section 5.1.2 [Création d'un contexte], page 464.

Positionnement horizontal des syllabes

La propriété `#'minimum-distance` de l'objet `LyricSpace` permet d'accroître l'espacement des paroles.

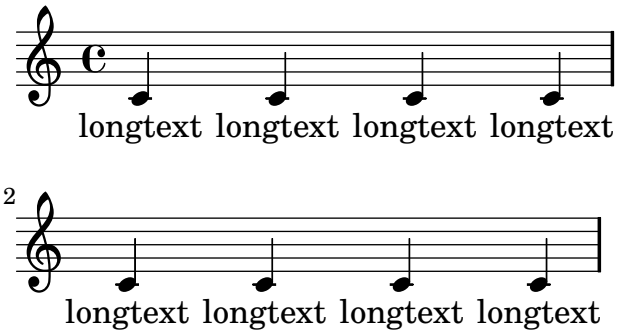
```
{
  c c c c
  \override Lyrics.LyricSpace #'minimum-distance = #1.0
  c c c c
}
\addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
}
```



Pour que ce réglage s'applique à toute la partition, définissez-le dans le bloc `\layout`.

```
\score {
  \relative c' {
  c c c c
  c c c c
  }
  \addlyrics {
  longtext longtext longtext longtext
  longtext longtext longtext longtext
```

```
  }
  \layout {
    \context {
      \Lyrics
      \override LyricSpace #'minimum-distance = #1.0
    }
  }
}
```

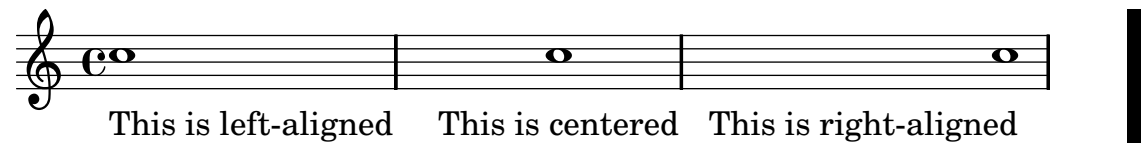


Morceaux choisis

Alignement des syllabes

L'alignement horizontal des paroles peut se gérer à l'aide de la propriété `self-alignment-X` de l'objet `LyricText`. Les valeurs `#-1` ou `#LEFT` produiront un alignement par la gauche, les valeurs `#0` ou `#CENTER` un alignement centré, et les valeurs `#1` ou `#RIGHT` un alignement par la droite.

```
\layout { ragged-right = ##f }
\relative c'' {
  c1
  c1
  c1
}
\addlyrics {
  \once \override LyricText #'self-alignment-X = #LEFT
  "This is left-aligned"
  \once \override LyricText #'self-alignment-X = #CENTER
  "This is centered"
  \once \override LyricText #'self-alignment-X = #1
  "This is right-aligned"
}
```



L'assurance que tous les scripts textuels et les paroles resteront bien à l'intérieur des marges requiert des ressources non négligeables. Afin de réduire le temps de traitement, LilyPond n'effectue pas ces calculs par défaut ; pour l'y obliger, vous devrez ajouter

```
\override Score.PaperColumn #'keep-inside-line = ##t
```

Pour s'assurer que les paroles ne seront pas traversées par des barres de mesure, il faut ajouter

```
\layout {
  \context {
    \Lyrics
      \consists "Bar_engraver"
      \consists "Separating_line_group_engraver"
      \override BarLine #'transparent = ##t
  }
}
```
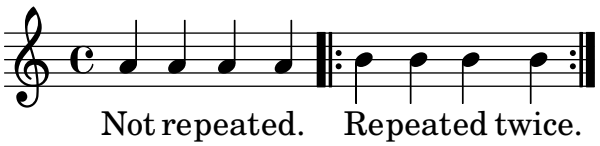
Paroles et reprises

La répétition de *fragments musicaux* est abordée de manière détaillée dans un Section "chapitre spécifique" dans *Manuel de notation*. Nous nous intéresserons ici aux moyens d'ajouter des paroles à des reprises.

Reprises simples

Les paroles attachées à un fragment musical répété devraient adopter rigoureusement la même construction que la musique, si tant est qu'elles ne diffèrent pas d'une fois sur l'autre.
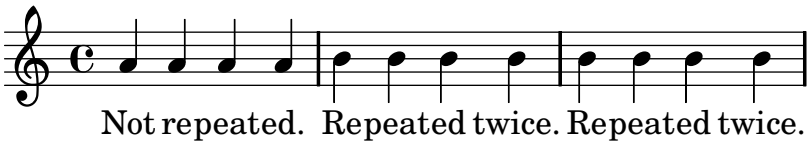
```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- ed twice. }
      }
    }
  >>
}
```



Les mots seront alors correctement répétés si la reprise est développée.

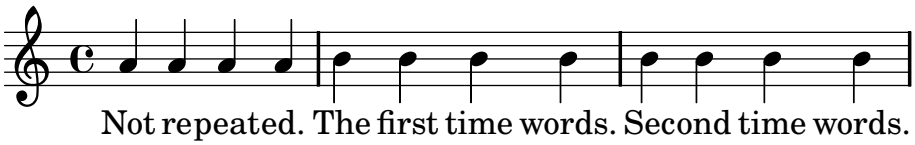
```
\score {
  \unfoldRepeats {
    <<
      \new Staff {
        \new Voice = "melody" {
          \relative c'' {
            a4 a a a
            \repeat volta 2 { b4 b b b }
          }
```

```
        }
      }
      \new Lyrics {
        \lyricsto "melody" {
          Not re -- peat -- ed.
          \repeat volta 2 { Re -- peat -- ed twice. }
        }
      }
    >>
  }
}
```



Lorsque la reprise est développée et que les paroles diffèrent, saisissez le texte normalement :
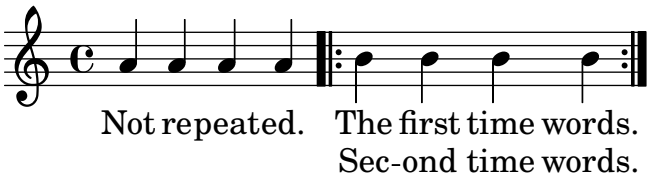
```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat unfold 2 { b4 b b b }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        The first time words.
        Sec -- ond time words.
      }
    }
  >>
}
```



Lorsque les paroles diffèrent pour une reprise non développée – utiliation de `volta` au lieu de `unfold` – les paroles en question doivent être saisies dans des contextes `Lyrics` séparés ; ils seront combinés dans une section parallèle :

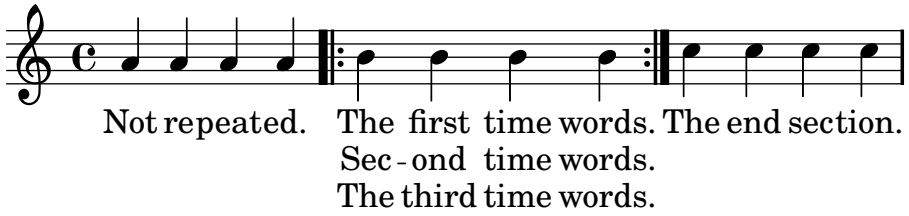
```
\score {
  <<
    \new Staff {
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
```

```
          \repeat volta 2 { b4 b b b }
        }
      }
    }
    \new Lyrics \lyricsto "melody" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics {
          \set associatedVoice = "melody"
          Sec -- ond time words.
        }
      >>
    }
  >>
}
```



Et ce quel que soit le nombre de « couplets » :
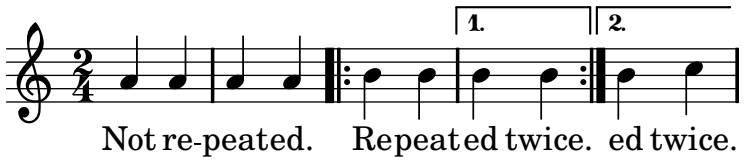
```
\score {
  <<
    \new Staff {
      \new Voice = "singleVoice" {
        \relative c'' {
          a4 a a a
          \repeat volta 3 { b4 b b b }
          c4 c c c
        }
      }
    }
    \new Lyrics \lyricsto "singleVoice" {
      Not re -- peat -- ed.
      <<
        { The first time words. }
        \new Lyrics {
          \set associatedVoice = "singleVoice"
          Sec -- ond time words.
        }
        \new Lyrics {
          \set associatedVoice = "singleVoice"
          The third time words.
        }
      >>
      The end sec -- tion.
    }
  >>
}
```



Reprises avec alternative

Les paroles d'un fragment répété, lorsqu'elles sont identiques, peuvent adopter la même construction que la musique :

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          a4 a a a
          \repeat volta 2 { b4 b }
          \alternative { { b b } { b c } }
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Not re -- peat -- ed.
        \repeat volta 2 { Re -- peat -- }
        \alternative { { ed twice. } { ed twice. } }
      }
    }
  >>
}
```
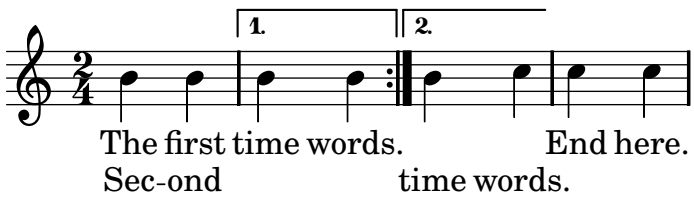


Cette identité de structure n'est toutefois pas possible lorsque les paroles sont différentes. Des instructions `\skip` devront venir s'insérer dans les paroles pour « sauter » les notes des alternatives qui ne les concernent pas.

N'utilisez pas de simple caratère souligné pour sauter une note. N'oubliez pas qu'un caractère souligné indique un mélisme ; la syllabe précédente sera donc alignée à gauche.

> **Note :** La commande `\skip` doit comporter une durée quelle qu'elle soit – elle sera toujours ignorée lorsque les paroles sont associées à une mélodie à l'aide de `\addlyrics` ou `\lyricsto`. Chaque `\skip` correspond à une seule note quelle qu'en soit la durée.

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
```

```
        \relative c'' {
          \repeat volta 2 { b4 b }
          \alternative { { b b } { b c } }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        The first time words.
        \repeat unfold 2 { \skip 1 }
        End here.
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Sec -- ond
        \repeat unfold 2 { \skip 1 }
        time words.
      }
    }
  >>
}
```

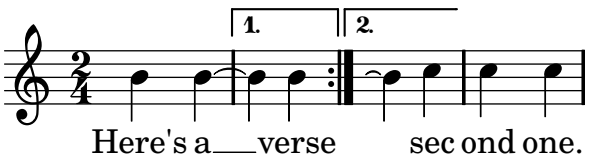


Lorsqu'une note se prolonge sur les alternatives, la tenue est indiquée normalement pour la première alternative, et à l'aide de l'instruction `\repeatTie` pour les suivantes. Cette liaison « de répétition » pose problème en matière d'alignement des paroles puisque la longueur de l'alternative est accrue en raison de la liaison.

D'autre part, une liaison de prolongation crée un mélisme qui sera effectif pour la première alternative, mais pas pour les autres. La solution pour recaler les paroles, consiste à désactiver temporairement la détection automatique de mélismes et insérer des « blancs ».

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \set melismaBusyProperties = #'()
          \repeat volta 2 { b4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          \unset melismaBusyProperties
          c4 c
        }
      }
    }
```

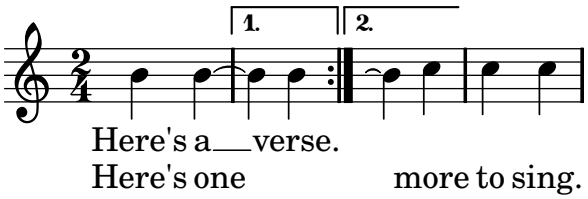
```
    \new Lyrics {
      \lyricsto "melody" {
        \repeat volta 2 { Here's a __ }
        \alternative {
          { \skip 1 verse }
          { \skip 1 sec }
        }
        ond one.
      }
    }
  >>
}
```



Notez bien que l'utilisation conjointe de `\unfoldRepeats` et de `\repeatTie` entraîne l'impression d'une double liaison, sauf à supprimer les `\repeatTie`.

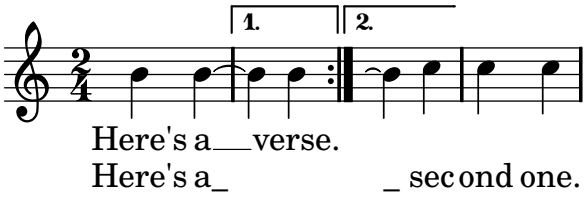
Lorsque les paroles sont différentes sur le fragment répété, la construction avec `\repeat` est inefficace ; vous devrez alors insérer des blancs :

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \repeat volta 2 { b4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's a __ verse.
        \repeat unfold 2 { \skip 1 }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's one
        \repeat unfold 2 { \skip 1 }
        more to sing.
      }
    }
  >>
}
```



Les indications de mélisme et traits d'union en début d'alternative doivent être insérés manuellement :

```
\score {
  <<
    \new Staff {
      \time 2/4
      \new Voice = "melody" {
        \relative c'' {
          \repeat volta 2 { b4 b ~}
          \alternative { { b b } { b \repeatTie c } }
          c4 c
        }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's a __ verse.
        \repeat unfold 2 { \skip 1 }
      }
    }
    \new Lyrics {
      \lyricsto "melody" {
        Here's "a_"
        \skip 1
        "_" sec -- ond one.
      }
    }
  >>
}
```



Voir aussi

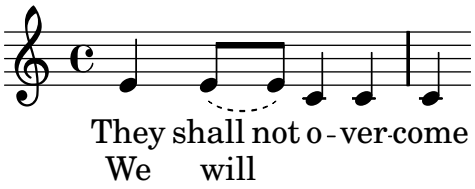
Manuel de notation : Section 5.1.3 [Conservation d'un contexte], page 464, Section 1.4 [Répétitions et reprises], page 124.

Paroles alternatives

Il arrive parfois, dans un fragment répété, qu'une note soit divisée pour répondre au texte. Vous pouvez indiquer cette adaptation rythmique en désactivant temporairement la détection automatique des mélismes tout en spécifiant ces mélismes au niveau des paroles :

```
\score {
  <<
```
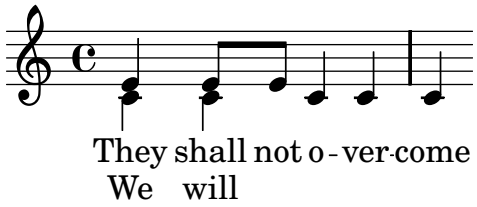
```
    \new Voice = "melody" {
      \relative c' {
        \set melismaBusyProperties = #'()
        \slurDown
        \slurDashed
        e4 e8 ( e ) c4 c |
        \unset melismaBusyProperties
        c
      }
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "melody" {
      We will _
    }
  >>
}
```



En donnant un nom à chaque voix et en leur attribuant spécifiquement des paroles, vous pourez traiter le cas où notes et rythme diffèrent d'une fois sur l'autre :

```
\score {
  <<
    \new Voice = "melody" {
      \relative c' {
        <<
          {
            \voiceOne
            e4 e8 e
          }
          \new Voice = "splitpart" {
            \voiceTwo
            c4 c
          }
        >>
        \oneVoice
        c4 c |
        c
      }
    }
    \new Lyrics \lyricsto "melody" {
      They shall not o -- ver -- come
    }
    \new Lyrics \lyricsto "splitpart" {
      We will
    }
```

```
  >>
}
```

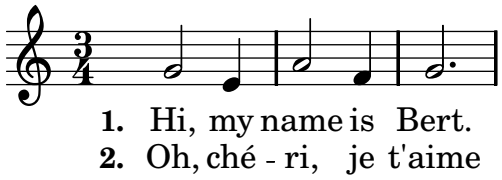


2.1.3 Couplets

Numérotation des couplets

On peut ajouter un numéro aux couplets en définissant la variable `stanza` :

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set stanza = #"1. "
  Hi, my name is Bert.
} \addlyrics {
  \set stanza = #"2. "
  Oh, ché -- ri, je t'aime
}
```
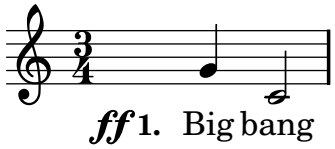


Ces numéros se placeront juste avant le début de la première syllabe.

Indication de nuance dans les couplets

Lorsque des couplets ont des nuances différentes, vous pouvez ajouter une nuance en regard de chacun d'eux. L'objet `StanzaNumber` contient tout ce qui se place avant les paroles du couplet. Pour des raisons techniques, vous devrez définir la variable `stanza` en dehors du mode `\lyricmode`.

```
text = {
  \set stanza = \markup { \dynamic "ff" "1. " }
  \lyricmode {
    Big bang
  }
}

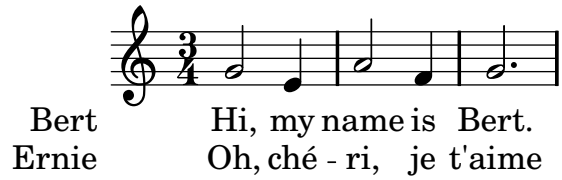
<<
  \new Voice = "tune" {
    \time 3/4
    g'4 c'2
  }
\new Lyrics \lyricsto "tune" \text
>>
```



Indication du personnage et couplets

On peut également ajouter le nom de chaque rôle ; ils s'imprimeront au début de chaque ligne comme les noms d'instrument. Il faut pour cela définir `vocalName`, et `shortVocalName` pour une version abrégée.

```
\new Voice {
  \time 3/4 g2 e4 a2 f4 g2.
} \addlyrics {
  \set vocalName = #"Bert "
  Hi, my name is Bert.
} \addlyrics {
  \set vocalName = #"Ernie "
  Oh, ché -- ri, je t'aime
}
```



Rythme différent selon le couplet

Il arrive assez souvent que le paroles de différents couplets, bien qu'attachées à une même mélodie, ne s'articulent pas de la même manière. La commande `\lyricsto` est cependant capable de gérer de telles situations.

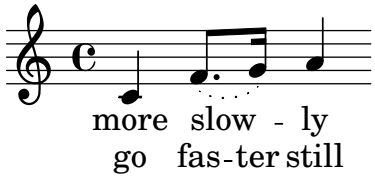
Mélismes dans certaines strophes seulement

Il peut survenir que les paroles comportent un mélisme pour l'un des couplets, mais plusieurs syllabes pour d'autres. Une solution consiste à ignorer temporairement les mélismes dans le couplet ayant le plus de syllabes. Il suffit pour cela de définir la propriété `ignoreMelismata` à l'intérieur du contexte `Lyrics`.

Petit détail qui a son importance : l'activation de `ignoreMelismata` doit **précéder** la syllabe à partir de laquelle elle s'appliquera :

```
<<
  \relative c' \new Voice = "lahlah" {
    \set Staff.autoBeaming = ##f
    c4
    \slurDotted
    f8.[( g16])
    a4
  }
  \new Lyrics \lyricsto "lahlah" {
    more slow -- ly
  }
  \new Lyrics \lyricsto "lahlah" {
    go
    \set ignoreMelismata = ##t
    fas -- ter
    \unset ignoreMelismata
```

```
    still
  }
>>
```
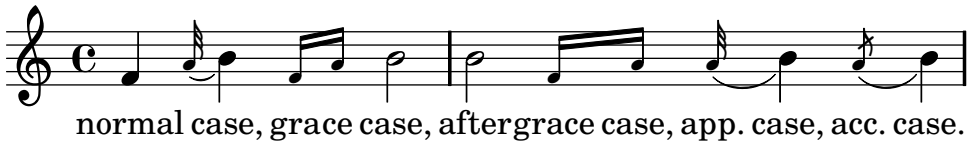


Problèmes connus et avertissements

Contrairement aux autres utilisations de l'instruction `\set`, il n'est pas possible de la faire précéder d'un `\once` dans le cas de `\set ignoreMelismata`. Les paroles affectées par la propriété `ignoreMelismata` **doivent** être encadrées respectivement d'un `\set` et d'un `\unset`.

Syllabe sur note de passage

L'utilisation de la commande `\lyricsto` ne permet pas, par défaut, d'assigner une syllabe à des notes d'ornement – introduites par la commande `\grace`. Vous pouvez cependant y parvenir grâce à la propriété `includeGraceNotes` :
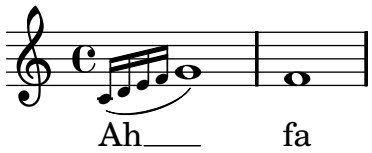
```
<<
  \new Voice = melody \relative c' {
    f4 \appoggiatura a32 b4
    \grace { f16[ a16] } b2
    \afterGrace b2 { f16[ a16] }
    \appoggiatura a32 b4
    \acciaccatura a8 b4
  }
  \new Lyrics
  \lyricsto melody {
    normal
    \set includeGraceNotes = ##t
    case,
    gra -- ce case,
    after -- grace case,
    \set ignoreMelismata = ##t
    app. case,
    acc. case.
  }
>>
```



Problèmes connus et avertissements

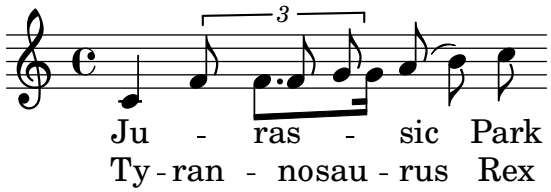
Tout comme pour la propriété `associatedVoice`, la propriété `includeGraceNotes` doit être activée au moins une syllabe avant celle qui viendra s'attacher à la note d'ornement. Dans le cas où cette note se trouve être la première de la pièce, vous devrez recourir à une clause `\with` ou `\context` :

```
<<
  \new Voice = melody \relative c' {
    \grace { c16[( d e f] }
    g1) f
  }
  \new Lyrics \with { includeGraceNotes = ##t }
  \lyricsto melody {
    Ah __ fa
  }
>>
```



Basculer vers une mélodie alternative

On peut créer des variations plus complexes à partir d'une mélodie à plusieurs voix. Les paroles peuvent suivre l'une ou l'autre des lignes mélodiques, et même basculer de l'une à l'autre si l'on modifie la propriété `associatedVoice`. Dans cet exemple,



le texte du premier couplet s'aligne de manière habituelle sur la mélodie nommée « lahlah ». Mais le second couplet, tout d'abord rattaché au contexte `lahlah`, bascule sur la mélodie `alternative` pour les syllabes « ran » à « sau » grâce aux lignes

```
\new Lyrics \lyricsto "lahlah" {
  \set associatedVoice = alternative % s'applique à "ran"
  Ty --
  ran --
  no --
  \set associatedVoice = lahlah % s'applique à "rus"
  sau -- rus Rex
}
```

où `alternative` désigne le nom du contexte `Voice` qui contient le triolet.

notez bien où apparaît la commande `\set associatedVoice` – une syllabe en avance, ce qui est tout à fait correct.

> **Note :** La commande `\set associatedVoice` **doit** intervenir une syllabe *avant* celle qui sera suivie par la bascule. Autrement dit, une modification de la voix associée sera effective une syllabe plus tard que prévu. Il ne s'agit en aucun cas d'un bogue, la raison en est purement technique.

Paroles en fin de partition

Il peut parfois s'avérer opportun d'aligner un seul couplet sur la mélodie, et de présenter tous les autres en bloc à la fin du morceau. Ces couplets additionnels peuvent être inclus dans une

section `\markup` en dehors du bloc `\score` principal. Vous en trouverez un exemple ci-dessous ; notez également les deux méthodes différentes employées pour indiquer les sauts de ligne, entre les couplets (*verses* en anglais) 2 et 3.
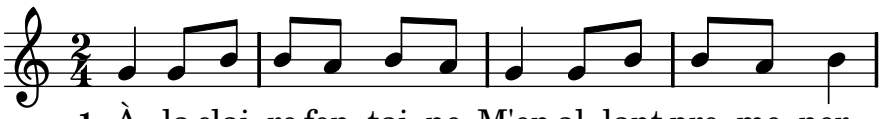
```
melody = \relative c'' {
  \time 2/4
  g4 g8 b | b a b a |
  g4 g8 b | b a b4 |
}

text = \lyricmode {
\set stanza = #"1." À la clai- re fon- tai- ne,
M'en al- lant pro- me- ner...
}

\score{ <<
  \new Voice = "one" { \melody }
  \new Lyrics \lyricsto "one" \text
>>
  \layout { }
}
\markup { \column{
  \line{ Verse 2. }
  \line{ Sous les feuilles d'un chêne }
  \line{ Je me suis fait sécher... }
  }
}
\markup{
  \wordwrap-string #"
  Verse 3.

  Chante, rossignol, chante,

  Toi qui as le cœur gai..."
}
```



1. À la clai- re fon- tai- ne, M'en al- lant pro- me- ner...

Verse 2.
Sous les feuilles d'un chêne
Je me suis fait sécher...

Verse 3.
Chante, rossignol, chante,
Toi qui as le cœur gai...

Paroles sur plusieurs colonnes en fin de partition

Si les couplets sont vraiment nombreux, il est possible de les imprimer sur plusieurs colonnes. L'exemple suivant vous montre comment procéder pour que le numéro du couplet soit en retrait à gauche, comme c'est traditionnellement le cas.
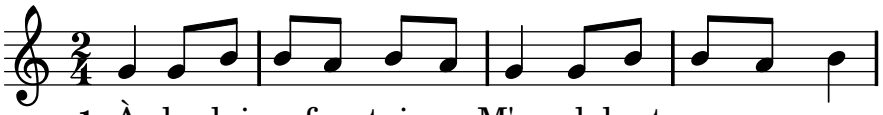
```
melody = \relative c'' {
  \time 2/4
  g4 g8 b | b a b a |
  g4 g8 b | b a b4 |
}

text = \lyricmode {
  \set stanza = #"1." À la clai- re fon- tai- ne,
  M'en al- lant pro- me- ner...
}

\score{ <<
    \new Voice = "one" { \melody }
    \new Lyrics \lyricsto "one" \text
   >>
  \layout { }
}

\markup {
  \fill-line {
    \hspace #0.1 % décalage par rapport à la marge de gauche
        % peut être supprimé si l'espace sur la page est réduit
     \column {
      \line { \bold "2."
        \column {
          "Sous les feuilles d'un chêne"
          "Je me suis fait sécher..."
        }
      }
      \hspace #0.1 % ajout d'espace vertical entre les couplets
      \line { \bold "3."
        \column {
          "Chante, rossignol, chante,"
          "Toi qui as le cœur gai..."
        }
      }
    }
    \hspace #0.1  % ajout d'espace horizontal entre les colonnes
        % si elles sont toujours trop proches, ajouter d'autres paires de " "
        % jusqu'à ce que le résultat soit acceptable.
     \column {
      \line { \bold "4."
        \column {
          "J'ai perdu mon ami"
          "Sans l'avoir mérité..."
        }
      }
      \hspace #0.1 % ajout d'espace vertical entre les couplets
      \line { \bold "5."
        \column {
          "Je voudrais que la rose"
          "Fût encore au rosier..."
```

```
        }
      }
    }
  \hspace #0.1 % décalage par rapport à la marge de droite
      % peut être supprimé si l'espace sur la page est réduit
  }
}
```



1. À la clai- re fon- tai- ne, M'en al- lant pro- me- ner...

2. Sous les feuilles d'un chêne
Je me suis fait sécher...
3. Chante, rossignol, chante,
Toi qui as le cœur gai...
4. J'ai perdu mon ami
Sans l'avoir mérité...
5. Je voudrais que la rose
Fût encore au rosier...

Voir aussi

Référence des propriétés internes : Section “LyricText” dans *Référence des propriétés internes*, Section “StanzaNumber” dans *Référence des propriétés internes*.

2.1.4 Chansons

Références en matière de chanson

Une chanson se présente la plupart du temps sous la forme de trois portées : une pour la mélodie surmontant un système pianistique pour l'accompagnement ; les paroles du premier couplet s'accolent sous la mélodie. S'il n'y a que deux ou trois couplets, et que cela n'est pas génant au niveau de l'aspect général, tous peuvent prendre place entre la mélodie et l'accompagnement. Dans le cas contraire, le premier couplet sera imprimé sous la mélodie et les suivants après la partition, sous forme de blocs de texte indépendants.

Tous les éléments qui permettent d'imprimer des chansons sont examinés à différents endroits de la documentation de LilyPond :

- L'agencement des portées est abordé au chapitre Section 1.6.1 [Gravure des portées], page 154.
- Les spécificités du piano sont abordées au chapitre Section 2.2 [Instruments utilisant des portées multiples], page 278.
- L'affectation de paroles à une ligne mélodique est abordée au chapitre Section 2.1.1 [Vue d'ensemble de la musique vocale], page 219.
- Le positionnement des paroles fait l'objet d'une Section “rubrique dédiée” dans *Manuel de notation*.
- La gestion des couplets est abordée dans un Section “chapitre spécifique” dans *Manuel de notation*.
- L'harmonisation d'une chanson est souvent indiquée par des noms d'accord en surplomb de la mélodie. Ceci est abordé au chapitre Section 2.7.2 [Gravure des accords], page 346.
- L'impression de diagrammes d'accord lorsque l'accompagnement est fait à la guitare est expliqué à la rubrique « Tablatures sous forme d'étiquette », au chapitre Section 2.4.1 [Vue d'ensemble des cordes frettées], page 294.

Voir aussi

Manuel d'initiation : Section "Chansons" dans *Manuel d'initiation*.

Manuel de notation : Section 2.1.1 [Vue d'ensemble de la musique vocale], page 219, Section 2.7.2 [Gravure des accords], page 346, Section 1.6.1 [Gravure des portées], page 154, Section 2.2 [Instruments utilisant des portées multiples], page 278, [Positionnement vertical des paroles], page 232, Section 2.1.3 [Couplets], page 247.

Morceaux choisis : Section "Musique vocale" dans *Morceaux choisis*.

Feuille de chant

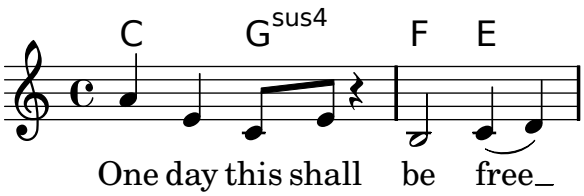
Une simple feuille de chant s'obtient en combinant une partie vocale et son harmonisation. La syntaxe appropriée est expliquée en détails au chapitre Section 2.7 [Notation des accords], page 343.

Morceaux choisis

Chanson simple

Assembler des noms d'accords, une mélodie et des paroles permet d'obtenir la partition d'un chanson :

```
<<
  \chords { c2 g:sus4 f e }
  \relative c'' {
    a4 e c8 e r4
    b2 c4( d)
  }
  \addlyrics { One day this shall be free __ }
>>
```



Voir aussi

Manuel de notation : Section 2.7 [Notation des accords], page 343.

2.1.5 Chorale

Nous allons voir, dans les paragraphes qui suivent, les particularités de la musique chorale, qu'il s'agisse de motet, d'oratorio ou de simple partie de chœur.

Références en matière de chorale

Une partition pour chœur comporte habituellement de deux à quatre portées regroupées dans un `ChoirStaff`. L'accompagnement, s'il y en a un, se présente sous la forme d'un système pianistique – un `PianoStaff` – en dessous du chœur ; il s'agira d'une simple réduction dans le cas d'une œuvre *a capella*. Les notes de chaque pupitre font l'objet d'un contexte `Voice` distinct. Ces contextes `Voice` peuvent se voir groupés ou non sur une même portée.

Les paroles sont traitées dans des contextes `Lyrics` qui viendront se placer tantôt sous la portée, tantôt au-dessus et au-dessous de la portée si elle contient deux voix.

Un certain nombre de composantes d'une partition pour chœur sont examinés à différents endroits de la documentation de LilyPond :

- La création pas à pas d'une partition pour chœur se trouve dans la manuel d'initiation, au chapitre Section "Partition pour chœur à quatre voix mixtes" dans *Manuel d'initiation*.
- Plusieurs exemples et canevas sont regroupés dans le manuel d'initiation, à la rubrique Section "Ensemble vocal" dans *Manuel d'initiation*.
- Des informations détaillées sur les contextes `ChoirStaff` et `PianoStaff` sont disponibles au chapitre [Regroupement de portées], page 155.
- Les formes de notation particulière, telles que celle utilisée en *Sacred Harp* et assimilées, sont abrodées au chapitre [Têtes de note à forme variable], page 33.
- Lorsque plusieurs pupitres sont regroupés sur la même portée, les hampes, liaisons etc. de la voix supérieure sont orientées vers le haut, et inversement pour la voix inférieure. L'utilisation de `\voiceOne` et `\voiceTwo` est expliquée au chapitre [Polyphonie sur une portée], page 141.

Commandes prédéfinies

`\oneVoice`, `\voiceOne`, `\voiceTwo`.

Voir aussi

Manuel d'initiation : Section "Partition pour chœur à quatre voix mixtes" dans *Manuel d'initiation*, Section "Ensemble vocal" dans *Manuel d'initiation*.

Manuel de notation : Section 5.4.3 [Ordre des contextes de mise en forme], page 475, [Regroupement de portées], page 155, [Têtes de note à forme variable], page 33, [Polyphonie sur une portée], page 141.

Morceaux choisis : Section "Musique vocale" dans *Morceaux choisis*.

Référence des propriétés internes : Section "ChoirStaff" dans *Référence des propriétés internes*, Section "Lyrics" dans *Référence des propriétés internes*, Section "PianoStaff" dans *Référence des propriétés internes*.

Mise en forme d'une partition chorale

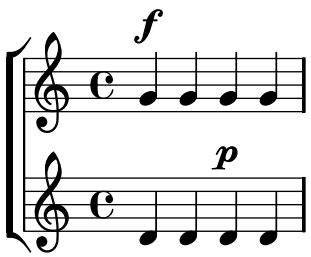
Une partition pour chœur sur quatre portées, avec ou sans accompagnement, présente traditionnellement deux systèmes par page. Selon la taille du papier, vous pourrez être amené à effectuer quelques ajustements aux réglages par défaut, notamment en raison des points suivants :

- La taille des portées a des répercussions sur l'ensemble des éléments de la partition. Voir Section 4.2.2 [Définition de la taille de portée], page 423.
- La distance séparant les systèmes, les portées et les paroles peut s'ajuster de manière séparée, comme expliqué au chapitre Section 4.4 [Espacement vertical], page 432.
- La mise en évidence des différentes dimensions permet d'appréhender avec finesse le réglage des variables d'espacement vertical et, pourquoi pas, de faire tenir la partition sur moins de pages, comme l'explique la rubrique Section 4.6 [Réduction du nombre de pages de la partition], page 459.
- Lorsque le nombre de systèmes varie d'une page à l'autre, il est judicieux de l'indiquer visuellement, en suivant les instructions de la rubrique [Séparation des systèmes], page 160.
- Pour de plus amples détails quant aux propriétés liées au formatage, consultez le chapitre Section 4.1 [Mise en forme de la page], page 412.

Les indications de nuance se placent traditionnellement sous la portée, ce qui n'est pas le cas en matière de musique vocale dans le but d'éviter toute collision avec les paroles. La commande prédéfinie `\dynamicUp` attachée à un contexte `Voice` permet de positionner les nuances au-dessus

de la portée. Dans le cas où il y en aurait plusieurs, cette commande devra apparaître dans chacun des contextes `Voice` qui le requiert. Vous pouvez aussi opter pour la forme développée, comme dans l'exemple ci-dessous, pour que cela s'applique à toutes les portées de la partition – changez `\Score` en `\ChoirStaff` s'il y a d'autres parties que celles du chœur.

```
\score {
  \new ChoirStaff <<
    \new Staff {
      \new Voice {
        \relative c'' { g4\f g g g }
      }
    }
    \new Staff {
      \new Voice {
        \relative c' { d4 d d\p d }
      }
    }
  >>
  \layout {
    \context {
      \Score
      \override DynamicText #'direction = #UP
      \override DynamicLineSpanner #'direction = #UP
    }
  }
}
```



Commandes prédéfinies

`\dynamicUp`, `\dynamicDown`, `\dynamicNeutral`.

Voir aussi

Manuel de notation : Section 4.6.2 [Modification de l'espacement], page 461, Section 4.6.1 [Mise en évidence de l'espacement], page 460, Section 4.6 [Réduction du nombre de pages de la partition], page 459, Section 4.1 [Mise en forme de la page], page 412, Section 4.2 [Mise en forme de la partition], page 422, [Séparation des systèmes], page 160, Section 4.2.2 [Définition de la taille de portée], page 423, Section 4.3.7 [Recours à une voix supplémentaire pour gérer les sauts], page 430, Section 4.4 [Espacement vertical], page 432.

Référence des propriétés internes : Section "VerticalAxisGroup" dans *Référence des propriétés internes*, Section "StaffGrouper" dans *Référence des propriétés internes*.

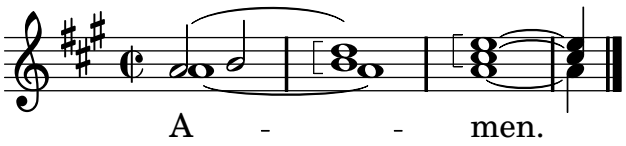
Division de voix

*Utilisation d'un arpeggioBracket pour rendre les*divisions plus évidentes

Les crochets d'arpegio (`arpeggioBracket`) permettent de mettre en évidence les divisions d'un pupitre en l'absence de hampe, comme on le voit régulièrement dans les partitions pour chœur.

```
\include "english.ly"

\score {
  \relative c'' {
    \key a \major
    \time 2/2
    <<
      \new Voice = "upper"
      <<
        { \voiceOne \arpeggioBracket
          a2( b2
          <b d>1\arpeggio)
          <cs e>\arpeggio ~
          <cs e>4
        }
        \addlyrics { \lyricmode { A -- men. } }
      >>
      \new Voice = "lower"
      { \voiceTwo
        a1 ~
        a
        a ~
        a4 \bar "|."
      }
    >>
  }
  \layout { ragged-right = ##t }
}
```



Voir aussi

Manuel de notation : Section 1.3.3 [Signes d'interprétation sous forme de ligne], page 117.

2.1.6 Opéras et musiques de scène

Tout ce qui permet d'exécuter un opéra ou une œuvre scénique accompagnée de musique se présente généralement sous l'une ou plusieurs des formes suivantes :

- Un *Conducteur* destiné au chef d'orchestre. Il comprend l'intégralité des parties d'orchestre et des chanteurs, ainsi que des citations du livret pour les passages déclamés.
- Un *matériel d'orchestre*, autrement dit une partition pour chacun des pupitres de l'orchestre ou de l'ensemble instrumental.
- Une *partition de chœur* regroupant toutes les parties vocales avec accompagnement au piano. Cet accompagnement est souvent une réduction d'orchestre où les différents instruments sont

indiqués. Les partitions de chœur comprennent parfois des indications de mise en scène ainsi que des extraits du livret.

- Une *partition de choriste* qui ne comprend que les parties vocales – donc sans accompagnement. Elle peut être augmentée du livret.
- Un *livret* contenant l'intégralité des dialogues et le texte des passsages chantés. On y trouve aussi très souvent les indications de mise en scène. Bien que LilyPond soit capable de « typographier » un livret, n'oubliez pas qu'il n'y a dans ce cas pas de musique, et que d'autres outils pourraient être mieux appropriés.

La plupart de ce qui est nécessaire à la mise en forme d'un opéra ou d'une musique de scène est disséminé dans la somme documentaire de LilyPond. Nous commencerons par rappeler ces différents éléments, avant que d'en examiner certaines particularités adaptées aux styles opératique et scénique.

Références en matière d'opéra et musique de scène

- Un conducteur contient un certain nombre de portées et de nombreuses paroles. Les manières d'agencer les portées sont indiquées à la rubrique [Regroupement de portées], page 155, et les façons de les combiner à la rubrique [Imbrication de regroupements de portées], page 159.
- Les portées vides sont la plupart du temps éliminées d'un conducteur ou d'une partition de chœur. La réalisation d'une telle partition – les anglophones la disent « à la française » – est expliquée à la rubrique [Masquage de portées], page 168.
- La génération d'un matériel d'orchestre fait l'objet de la rubrique Section 1.6.3 [Écriture de parties séparées], page 171. D'autres parties du chapitre consacré à la notation spécialisée vous seront utiles selon l'orchestration de la pièce. Tous les instruments ne sont pas accordés pareil ; vous trouverez des informations à ce sujet à la rubrique [Instruments transpositeurs], page 19.
- Lorsque le nombre de systèmes varie d'une page à l'autre, il peut être judicieux de les mettre en évidence, en suivant les indications de la rubrique [Séparation des systèmes], page 160.
- Les différentes propriétés impliquées dans la mise en page sont répertoriées au chapitre Section 4.1 [Mise en forme de la page], page 412.
- L'insertion de dialogues et d'indications de mise en scène peuvent se réaliser à l'aide de *markups*, en suivant les directives fournies au chapitre Section 1.8 [Texte], page 193. Les indications de mise en scène peuvent s'intercaler entre deux blocs `\score` selon les préceptes de la rubrique [Texte indépendant], page 201.

Voir aussi

Glossaire musicologique : Section "Partition à la française" dans *Glossaire*, Section "Frenched staves" dans *Glossaire*, Section "instrument transpositeur" dans *Glossaire*.

Manuel de notation : [Regroupement de portées], page 155, [Masquage de portées], page 168, [Instruments transpositeurs], page 19, [Imbrication de regroupements de portées], page 159, Section 4.1 [Mise en forme de la page], page 412, [Séparation des systèmes], page 160, [Transposition], page 10, Section 1.6.3 [Écriture de parties séparées], page 171, Section 1.8.1 [Ajout de texte], page 194.

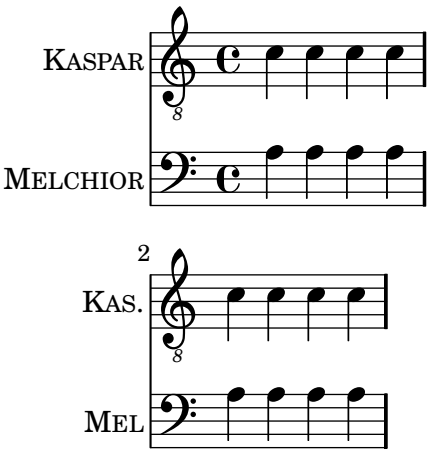
Morceaux choisis : Section "Musique vocale" dans *Morceaux choisis*.

Indication du rôle

Lorsqu'un rôle est distribué sur une portée spécifique, vous pouvez l'indiquer en regard de cette portée :
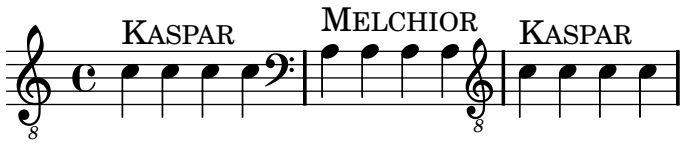
```
\score {
  <<
```

```
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Kaspar
      \set Staff.shortVocalName = \markup \smallCaps Kas.
      \relative c' {
        \clef "G_8"
        c4 c c c
        \break
        c4 c c c
      }
    }
    \new Staff {
      \set Staff.vocalName = \markup \smallCaps Melchior
      \set Staff.shortVocalName = \markup \smallCaps Mel
      \clef "bass"
      \relative c' {
        a4 a a a
        a4 a a a
      }
    }
  >>
}
```



Lorsque la même portée sert à plusieurs personnages, leur nom est généralement imprimé en surplomb de la portée, à chaque changement de rôle. L'utilisation d'un *markup* – dans une fonte réservée à cet effet – vous permettra de générer ces indications :

```
\clef "G_8"
c4^\markup \fontsize #1 \smallCaps Kaspar
c c c
\clef "bass"
a4^\markup \fontsize #1 \smallCaps Melchior
a a a
\clef "G_8"
c4^\markup \fontsize #1 \smallCaps Kaspar
c c c
```
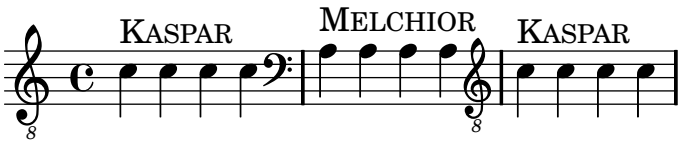


Dans le cas où les changements de personnage se multiplient, il peut s'avérer pratique de détourner l'utilisation de la fonction `\instrumentSwitch` pour gérer les différentes interventions de l'un ou de l'autre ; vous devrez auparavant avoir établi la définition d'un « instrument » pour chacun des rôles en question :

```
\addInstrumentDefinition #"kaspar"
  #`((instrumentTransposition . ,(ly:make-pitch -1 0 0))
     (shortInstrumentName . "Kas.")
     (clefGlyph . "clefs.G")
     (clefOctavation . -7)
     (middleCPosition . 1)
     (clefPosition . -2)
     (instrumentCueName . ,(markup #:fontsize 1 #:smallCaps "Kaspar"))
     (midiInstrument . "voice oohs"))

\addInstrumentDefinition #"melchior"
  #`((instrumentTransposition . ,(ly:make-pitch 0 0 0))
     (shortInstrumentName . "Mel.")
     (clefGlyph . "clefs.F")
     (clefOctavation . 0)
     (middleCPosition . 6)
     (clefPosition . 2)
     (instrumentCueName . ,(markup #:fontsize 1 #:smallCaps "Melchior"))
     (midiInstrument . "voice aahs"))

\relative c' {
  \instrumentSwitch "kaspar"
  c4 c c c
  \instrumentSwitch "melchior"
  a4 a a a
  \instrumentSwitch "kaspar"
  c4 c c c
}
```



Voir aussi

Manuel de notation : [Noms d'instrument], page 171, Section A.17 [Fonctions Scheme], page 591, Section 1.8 [Texte], page 193, Section A.8 [Text markup commands], page 510.

Manuel d'extension des fonctionnalités : Section "Construction d'un markup en Scheme" dans *Extension de LilyPond*.

Citation-repère

Les citations d'instruments insérées dans les parties vocales, les partitions de chœur ou les partitions d'un pupitre permettent d'indiquer ce qui se passe ailleurs juste avant une entrée. On les retrouve souvent dans la réduction pour piano, ce qui fournit au chef de chœur de précieuses indications sur qui joue quoi, lorsqu'il ne dispose pas d'un conducteur en bonne et due forme.

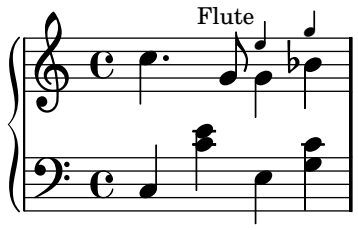
Les mécanismes de base permettant d'insérer des citations sont expliqués en détail aux rubriques [Citation d'autres voix], page 174 et [Mise en forme d'une citation], page 178. Dans

le cas où les citations concernent différents instruments, faire mention de celui qui intervient devient une nécessité ; voici une illustration de la manière de procéder en pareil cas :

```
flute = \relative c'' {
  s4 s4 e g
}
\addQuote "flute" { \flute }

pianoRH = \relative c'' {
  c4. g8
  % position name of cue-ing instrument just before the cue notes,
  % and above the staff
  s1*0^\markup { \right-align { \tiny "Flute" } }
  \cueDuring "flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  \new PianoStaff <<
    \new Staff {
      \pianoRH
    }
    \new Staff {
      \clef "bass"
      \pianoLH
    }
  >>
}
```



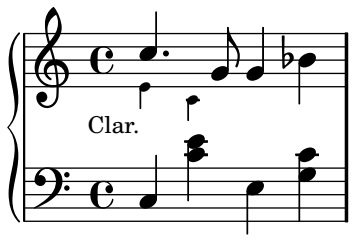
La citation peut concerner un instrument transpositeur, auquel cas il faudra mentionner sa tonalité dans sa définition, afin que ses hauteurs soient automatiquement converties dans la réplique. Ceci est illustré par l'exemple ci-dessous, dans lequel il est fait appel à une clarinette en si bémol. Dans la mesure où les notes citées se trouvent vers le bas de la portée, nous ajoutons un `#DOWN` à la commande `\cueDuring`, de telle sorte que les hampes aillent vers le bas et que le nom de l'instrument cité soit en dessous de la portée. Vous noterez que la voix contenant la main droite du piano est explicitement déclarée ; ceci tient au fait que la citation se produit dès le début de la première mesure – si nous ne le faisions pas, l'intégralité de la main droite se verrait placée dans un contexte `CueVoice`.

```
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

pianoRH = \relative c'' {
```

```
  \transposition c'
  % position name of cue-ing instrument below the staff
  s1*0_\markup { \right-align { \tiny "Clar." } }
  \cueDuring "clarinet" #DOWN { c4. g8 }
  g4 bes4
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
    >>
  >>
}
```



Il est clair, au vu de ces deux exemples, que multiplier le nombre de citations dans une partition vocale demande un travail fastidieux et que relire la partie de piano deviendrait vite un cauchemar. Vous pouvez néanmoins, comme l'illustre l'exemple suivant, définir une fonction musicale dans le but de vous épargner de la saisie tout en améliorant la lisibilité des notes du piano.

Morceaux choisis

*Indication de l'instrument cité dans l'accompagnement d'une*partition pour chœur

Lorsque le nombre d'instruments cités dans la réduction pour piano se multiplie, vous pourriez avoir intérêt à créer votre propre fonction pour gérer ces repères. La fonction musicale `\cueWhile` prend quatre arguments : la musique d'où provient la citation, telle que définie par `\addQuote`, le nom qui sera mentionné en regard de cette citation, son positionnement – `#UP` ou `#DOWN` selon qu'il sera attribué à `\voiceOne` et placé au-dessus ou `\voiceTwo` et placé en dessous – et enfin la musique du piano qui interviendra en parallèle. Le nom de l'instrument en question viendra s'aligner sur la gauche de la citation. Bien que vous puissiez effectuer plusieurs citations, elle ne peuvent être simultanées.

```
cueWhile =
#(define-music-function
   (parser location instrument name dir music)
   (string? string? ly:dir? ly:music?)
```

```
   #{
     \cueDuring $instrument #$dir {
       \once \override TextScript #'self-alignment-X = #RIGHT
       \once \override TextScript #'direction = $dir
       s1*0-\markup { \tiny $name }
       $music
     }
   #})

flute = \relative c'' {
  \transposition c'
  s4 s4 e g
}
\addQuote "flute" { \flute }
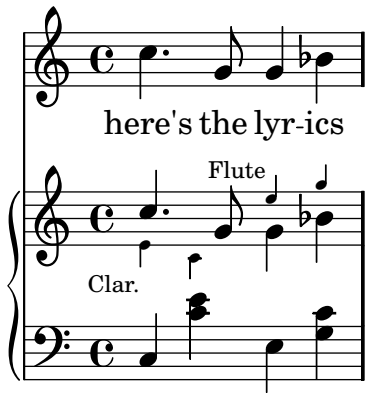
clarinet = \relative c' {
  \transposition bes
  fis4 d d c
}
\addQuote "clarinet" { \clarinet }

singer = \relative c'' { c4. g8 g4 bes4 }
words = \lyricmode { here's the lyr -- ics }

pianoRH = \relative c'' {
  \transposition c'
  \cueWhile "clarinet" "Clar." #DOWN { c4. g8 }
  \cueWhile "flute" "Flute" #UP { g4 bes4 }
}
pianoLH = \relative c { c4 <c' e> e, <g c> }

\score {
  <<
    \new Staff {
      \new Voice = "singer" {
        \singer
      }
    }
    \new Lyrics {
      \lyricsto "singer"
      \words
    }
    \new PianoStaff <<
      \new Staff {
        \new Voice {
          \pianoRH
        }
      }
      \new Staff {
        \clef "bass"
        \pianoLH
      }
```

```
    >>
  >>
}
```



Voir aussi

Glossaire musicologique : Section “cue-notes” dans *Glossaire*.

Manuel de notation : Section 5.5.1 [Alignement des objets], page 476, Section 5.4.2 [Direction et positionnement], page 475, [Mise en forme d’une citation], page 178, [Citation d’autres voix], page 174, Section 5.6 [Utilisation de fonctions musicales], page 476.

Morceaux choisis : Section “Musique vocale” dans *Morceaux choisis*.

Référence des propriétés internes : Section “InstrumentSwitch” dans *Référence des propriétés internes*, Section “CueVoice” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

`\cueDuring` crée automatiquement un contexte `CueVoice` qui accueillera toutes les notes répliquées. Il est par conséquenet impossible de faire se superposer des citations à l’aide de la technique simplifiée telle que nous venons de le voir. La superposition de fragments cités requiert que les contextes `CueVoice` soient explicitement déclarés, ainsi que l’utilisation de la commande `\quoteDuring` pour extraire et insérer les notes répliquées.

Musique parlée
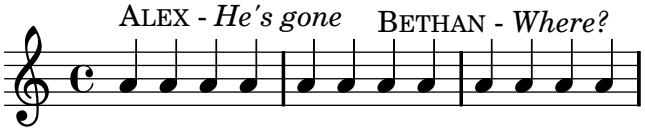
Le *parlato* – ou *Sprechgesang* pour les germanistes – est du texte scandé en rythme, mais sans hauteurs définies ; il est indiqué par des notes en croix, à l’instar des percussions – voir [Têtes de note spécifiques], page 30.

Dialogue et musique

Les dialogues que l’on ajoute à la musiqua sont traditionnellement imprimés en italique au-dessus des portées, au moment même où ils surviennent.

Une courte intervention peut se formuler à l’aide d’un simple *markup* :

```
a4^\markup { \smallCaps { Alex - } \italic { He's gone } } a a a
a4 a a^\markup { \smallCaps { Bethan - } \italic Where? } a
a4 a a a
```



Une intervention un peu plus longue peut nécessiter d'étirer la musique de telle sorte que le texte ait suffisamment de place. LilyPond ne disposant d'aucun mécanisme permettant d'automatiser l'étirement, vous devrez probablement ajuster vous même la mise en forme.

Dans le cas d'une phrase entière ou de passages relativement denses, le recours à un contexte `Lyrics` peut donner de meilleurs résultats. Le contexte `Lyrics` en question ne doit être rattaché à aucune voix musicale ; chaque fragment de dialogue devra donc comporter des durées explicites. Lorsque les dialogues comportent des pauses, le dernier mot devra être séparé du reste et les durées individualisées pour obtenir un espacement harmonieux de la musique.

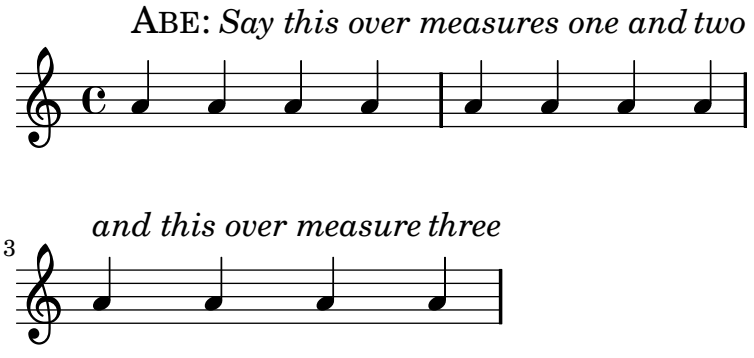
Des dialogues qui s'étendent sur plus d'une ligne vous obligeront à insérer des `\break` et ajuster leur placement pour éviter qu'ils ne débordent dans la marge droite. Le dernier mot de la dernière mesure d'une ligne doit être saisi sur une ligne à part.

Voici une illustration de tout ce que nous venons de voir :

```
music = \relative c'' {
  \repeat unfold 3 { a4 a a a }
}

dialogue = \lyricmode {
  \markup {
    \fontsize #1 \upright \smallCaps Abe:
    "Say this over measures one and"
  }4*7
  "two"4 |
  \break
  "and this over measure"4*3
  "three"4 |
}

\score {
  <<
    \new Lyrics \with {
      \override LyricText #'font-shape = #'italic
      \override LyricText #'self-alignment-X = #LEFT
    }
    { \dialogue }
    \new Staff {
      \new Voice { \music }
    }
  >>
}
```



Voir aussi

Manuel de notation : [Durée explicite des syllabes], page 225, Section 1.8 [Texte], page 193.

Référence des propriétés internes : Section “LyricText” dans *Référence des propriétés internes*.

2.1.7 Chants liturgiques

Selon les chapelles, la mise en forme des cantiques, psaumes et hymnes répond à des canons bien établis. Bien que différents de par leur présentation, nous verrons dans ce qui suit que les problèmes qui surviennent en matière de typographie se ressemblent, quelle que soit l'obédience.

Références en matière de chant liturgique

La présentation du plain chant et du grégorien selon différents styles est abordée au chapitre Section 2.9 [Notations anciennes], page 354.

Voir aussi

Manuel de notation : Section 2.9 [Notations anciennes], page 354.

Morceaux choisis : Section “Musique vocale” dans *Morceaux choisis*.

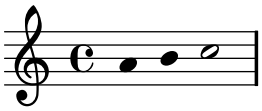
Cantiques et hymnes

La mise en forme contemporaine de cantiques utilise à la fois la notation moderne et un certain nombre d'éléments propres aux notations anciennes. Nous allons examiner quelques uns de ces éléments et la méthode consacrée pour les mettre en œuvre.

Les cantiques utilisent souvent des noires dépourvues de hampe pour indiquer les hauteurs ; le rythme de la mélodie est donné par le rythme et l'accentuation des paroles elles-même.

```
stemOff = { \override Staff.Stem #'transparent = ##t }
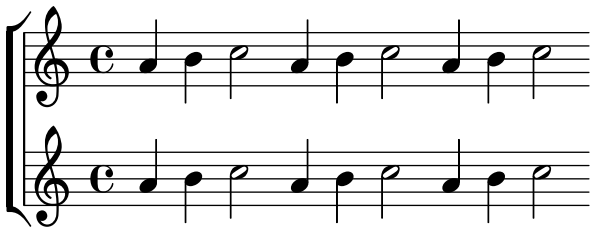
\relative c' {
  \stemOff
  a'4 b c2 |
}
```



Les barres de mesure sont absentes dans la plupart des cas ; celles que vous rencontrerez seront raccourcies ou en pointillé, dans le but d'indiquer une « respiration ». Le fait de supprimer le graveur de barres de mesure produira des portées sans barre :
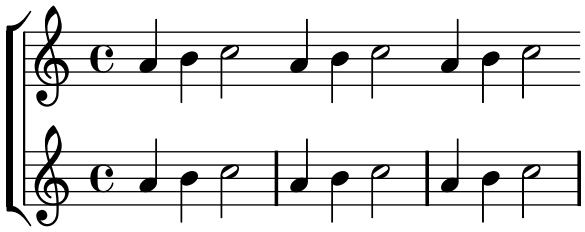
```
\score {
  \new StaffGroup <<
    \new Staff {
      \relative c'' {
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
    \new Staff {
      \relative c'' {
```

```
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
  >>
  \layout {
    \context {
      \Staff
      \remove Bar_engraver
    }
  }
}
```
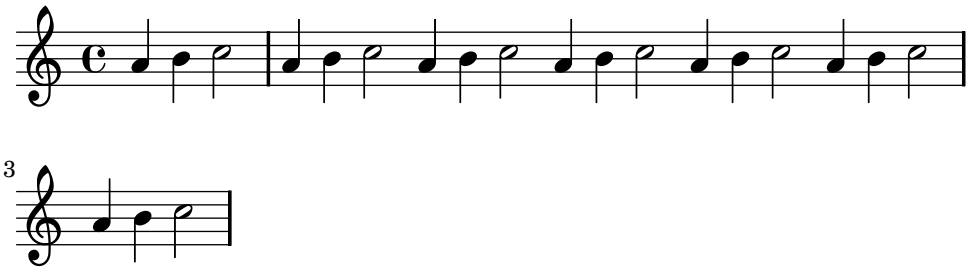


L'absence de barre de mesure peut ne concerner que certaines portées :

```
\score {
  \new ChoirStaff <<
    \new Staff
    \with { \remove Bar_engraver } {
      \relative c'' {
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
    \new Staff {
      \relative c'' {
        a4 b c2 |
        a4 b c2 |
        a4 b c2 |
      }
    }
  >>
}
```
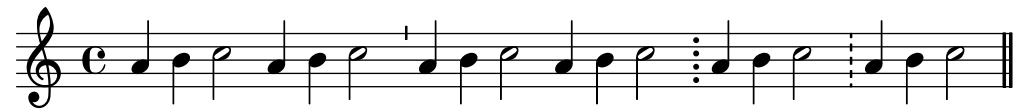


L'absence de barre de mesure sur un fragment seulement s'obtient en traitant ce fragment comme une cadence. S'il est relativement long, pensez à y insérer des barres fantômes – un simple `\bar ""` – pour indiquer à LilyPond où serait susceptible de se produire un saut de ligne.

```
a4 b c2 |
\cadenzaOn
a4 b c2
a4 b c2
\bar ""
a4 b c2
a4 b c2
\cadenzaOff
a4 b c2 |
a4 b c2 |
```



Dans la mélodie d'un cantique, les silences ou pauses s'indiquent à l'aide de barres de mesure spécifiques :

```
a4
\cadenzaOn
b c2
a4 b c2
\bar "'"
a4 b c2
a4 b c2
\bar ":"
a4 b c2
\bar "dashed"
a4 b c2
\bar "||"
```
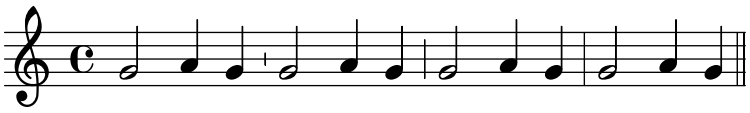


Vous pouvez accessoirement, bien qu'il s'agisse de notation moderne, emprunter au grégorien des indications de pause et silence. Il vous suffit pour cela d'adapter la commande `\breathe` selon vos besoins :

```
divisioMinima = {
  \once \override BreathingSign  #'stencil = #ly:breathing-sign::divisio-minima
  \once \override BreathingSign  #'Y-offset = #0
  \breathe
}
divisioMaior = {
  \once \override BreathingSign  #'stencil = #ly:breathing-sign::divisio-maior
  \once \override BreathingSign  #'Y-offset = #0
  \breathe
}
```
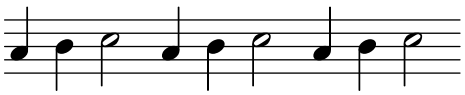
```
divisioMaxima = {
  \once \override BreathingSign  #'stencil = #ly:breathing-sign::divisio-maxima
  \once \override BreathingSign  #'Y-offset = #0
  \breathe
}
finalis = {
  \once \override BreathingSign  #'stencil = #ly:breathing-sign::finalis
  \once \override BreathingSign  #'Y-offset = #0
  \breathe
}

\score {
  \relative c'' {
    g2 a4 g
    \divisioMinima
    g2 a4 g
    \divisioMaior
    g2 a4 g
    \divisioMaxima
    g2 a4 g
    \finalis
  }
  \layout {
    \context {
      \Staff
      \remove Bar_engraver
    }
  }
}
```



De nombreux cantiques sont dépourvus de métrique, voire même de clef.

```
\score {
  \new Staff {
    \relative c'' {
      a4 b c2 |
      a4 b c2 |
      a4 b c2 |
    }
  }
  \layout {
    \context {
      \Staff
      \remove Bar_engraver
      \remove Time_signature_engraver
      \remove Clef_engraver
    }
  }
}
```



L'une des traditions anglicanes est de chanter les psaumes sur la base d'un fragment de sept mesures – forme *single* ou simple – ou de deux fragments toujours de sept mesures – forme *double*. Chaque fragment est divisé en deux parties correspondant aux deux moitiés de chaque verset et généralement séparées par une double barre. Il n'est fait usage que de rondes et de blanches, et la première mesure de chaque moitié contient un simple accord de rondes. Il s'agit donc des notes correspondant au « récitatif ». Ces cantiques sont traditionnellement centrés sur la page.

```
SopranoMusic = \relative g' {
  g1 | c2 b | a1 | \bar "||"
  a1 | d2 c | c b | c1 | \bar "||"
}

AltoMusic = \relative c' {
  e1 | g2 g | f1 |
  f1 | f2 e | d d | e1 |
}

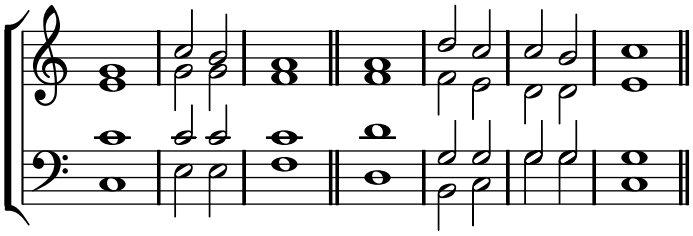
TenorMusic = \relative a {
  c1 | c2 c | c1 |
  d1 | g,2 g | g g | g1 |
}

BassMusic =  \relative c {
  c1 | e2 e | f1 |
  d1 | b2 c | g' g | c,1 |
}

global = {
  \time 2/2
}

% Use markup to center the chant on the page
\markup {
  \fill-line {
    \score {  % centered
      <<
        \new ChoirStaff <<
          \new Staff <<
            \global
            \clef "treble"
            \new Voice = "Soprano" <<
              \voiceOne
              \SopranoMusic
            >>
            \new Voice = "Alto" <<
              \voiceTwo
              \AltoMusic
            >>
          >>
          \new Staff <<
```

```
            \clef "bass"
            \global
            \new Voice = "Tenor" <<
              \voiceOne
              \TenorMusic
            >>
            \new Voice = "Bass" <<
              \voiceTwo
              \BassMusic
            >>
          >>
        >>
      >>
      \layout {
        \context {
          \Score
          \override SpacingSpanner
          #'base-shortest-duration = #(ly:make-moment 1 2)
        }
        \context {
          \Staff
          \remove "Time_signature_engraver"
        }
      }
    }  % End score
  }
}  % End markup
```



D'autres approches d'une telle mise en forme font l'objet du premier des exemples qui suivent.

Morceaux choisis

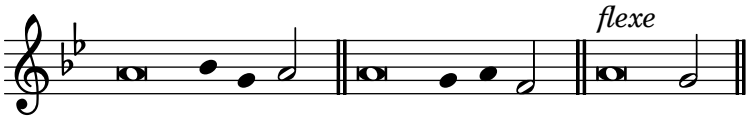
Notation pour psalmodie

Ce style de notation permet d'indiquer la mélodie d'une psalmodie lorsque les strophes sont de longueur inégale.

```
stemOn = { \revert Staff.Stem #'transparent }
stemOff = { \override Staff.Stem #'transparent = ##t }

\score {
  \new Staff \with { \remove "Time_signature_engraver" }
  {
    \key g \minor
    \cadenzaOn
    \stemOff a'\breve bes'4 g'4
    \stemOn a'2 \bar "||"
```

```
    \stemOff a'\breve g'4 a'4
    \stemOn f'2 \bar "||"
    \stemOff a'\breve^\markup { \italic flexe }
    \stemOn g'2 \bar "||"
  }
}
```



Cantiques et autres textes liturgiques peuvent être mis en forme avec une grande liberté et parfois emprunter des éléments de notation ancienne. Le texte apparaît souvent sous la mélodie, les mots alors alignés sur les notes. En pareil cas, les notes sont espacées selon les syllabes et non leur durée.

Exemples de notation ancienne – transcription moderne de musique grégorienne

Voici comment vous pourriez transcrire du grégorien. Pour mémoire, il n'y a en grégorien ni de découpage en mesure, ni de hampe ; seules sont utilisées des têtes de note blanches ou noires, ainsi que des signes spécifiques permettant d'indiquer des silences de différentes durées.
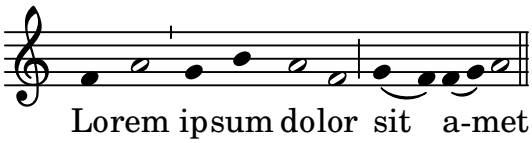
```
\include "gregorian.ly"

chant = \relative c' {
  \set Score.timing = ##f
  f4 a2 \divisioMinima
  g4 b a2 f2 \divisioMaior
  g4( f) f( g) a2 \finalis
}

verba = \lyricmode {
  Lo -- rem ip -- sum do -- lor sit a -- met
}

\score {
  \new Staff <<
    \new Voice = "melody" \chant
    \new Lyrics = "one" \lyricsto melody \verba
  >>
  \layout {
    \context {
      \Staff
      \remove "Time_signature_engraver"
      \remove "Bar_engraver"
      \override Stem #'transparent = ##t
    }
    \context {
      \Voice
      \override Stem #'length = #0
    }
    \context {
      \Score
```

```
      barAlways = ##t
    }
  }
}
```



Voir aussi

Manuel d'initiation : Section "Visibilité et couleur des objets" dans *Manuel d'initiation*, Section "Ensemble vocal" dans *Manuel d'initiation*.

Manuel de notation : Section 2.9 [Notations anciennes], page 354, [Barres de mesure], page 84, Section 5.1.4 [Modification des greffons de contexte], page 464, ⟨undefined⟩ [Typesetting Gregorian chant], page ⟨undefined⟩, [Musique sans métrique], page 63, Section 5.4.7 [Visibilité des objets], page 475.

Psalmodie

Les versets d'un psaume anglican sont habituellement centrées sous la mélodie.

Dans le cas d'un chant simple, les sept mesures sui le composent sont répétées pour chaque verset. Dans le cas d'un chant double, les quatorze mesures se répètent par couple de versets. Des marques insérées dans le texte indiquent comment il s'articule par rapport à la mélodie. Chaque verset est séparé en deux, et la rupture est indiquée par un caractère deux points (`:`) correspondant à la double barre de la mélodie. Le texte précédant les deux points se chante sur les trois premières mesures, celui qui suit sur les quatre dernières mesures.

De simples barres verticales – remplacées par des virgules inversées dans certains psautiers – représentent les barres de mesures portées sur la mélodie. En mode *markup*, ces barres s'obtiennent en saisissant le même caractère `|` qui sert pour les contrôles de mesure.

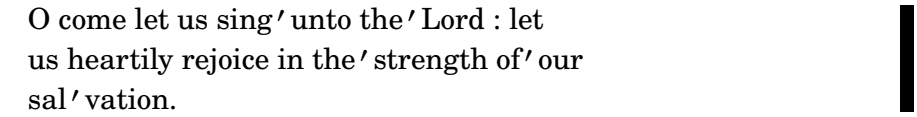
```
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { O come let us sing | unto the | Lord : let }
        \line { us heartily rejoice in the | strength of | our }
        \line { sal- | -vation. }
      }
    }
  }
}
```

O come let us sing | unto the | Lord : let
us heartily rejoice in the | strength of | our
sal- | -vation.

Vous pourriez tout à fait utiliser d'autres symboles disponibles au travers des glyphes de la fonte `fetaMusic` – voir le chapitre Section 1.8.3 [Fontes], page 216 pour plus de détatils.

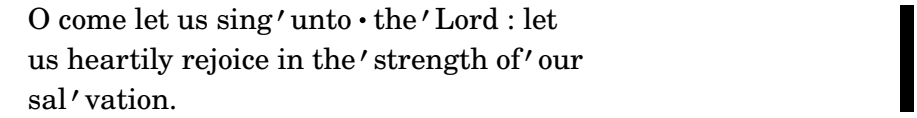
```
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
```

```
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { O come let us sing \tick unto the \tick Lord : let }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
        \line { sal \tick vation. }
      }
    }
  }
}
```



Lorsqu'une mesure ne comporte qu'une ronde, le texte correspondant à cette mesure est chanté sur cette même note, selon le rythme naturel de la phrase. Lorsque la mesure comporte deux notes, celles-ci correspondent en général à une ou deux syllabes ; dans le cas contraire, le changement de note est indiqué par un point.

```
dot = \markup {
  \raise #0.7 \musicglyph #"dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          O come let us sing \tick unto \dot the \tick Lord : let
        }
        \line {
          us heartily rejoice in the \tick strength of \tick our
        }
        \line { sal \tick vation. }
      }
    }
  }
}
```



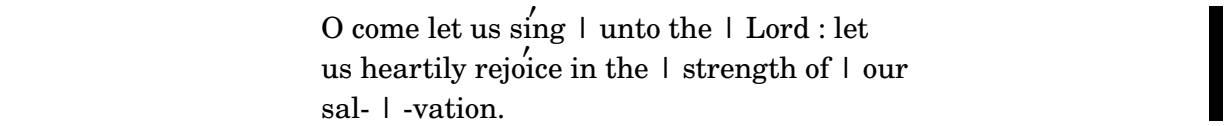
Certains psautiers font apparaître, pour indiquer une césure, une astérisque au lieu d'une virgule, ainsi que des caractères gras pour les syllabes accentuées ou allongées.

```
dot = \markup {
  \raise #0.7 \musicglyph #"dots.dot"
}
tick = \markup {
  \raise #1 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line { Today if ye will hear his voice * }
        \line {
          \concat { \bold hard en }
          | not your | hearts : as in the pro-
        }
        \line { vocation * and as in the \bold day of tempt- | }
        \line { -ation | in the | wilderness. }
      }
    }
  }
}
```

Today if ye will hear his voice *
harden | not your | hearts : as in the pro-
vocation * and as in the **day** of tempt- |
-ation | in the | wilderness.

D'autres psautiers indiquent une syllabe accentuées en la surchargeant d'un accent.

```
tick = \markup {
  \raise #2 \fontsize #-5 \musicglyph #"scripts.rvarcomma"
}
\markup {
  \fill-line {
    \column {
      \left-align {
        \line {
          O come let us \concat {
            si \combine \tick ng
          }
          | unto the | Lord : let
        }
        \line {
          us heartily \concat {
            rejo \combine \tick ice
          }
          in the | strength of | our
        }
        \line { sal- | -vation. }
      }
    }
  }
}
```



L'utilisation du mode *markup* pour centrer le texte et agencer les lignes est abordée en détails au chapitre Section 1.8.2 [Mise en forme du texte], page 202.

La plupart de ces éléments sont regroupés dans l'un des versets du modèle « Psalmodie », à la rubrique Section "Ensemble vocal" dans *Manuel d'initiation*.

Voir aussi

Manuel d'initiation : Section "Ensemble vocal" dans *Manuel d'initiation*.

Manuel de notation : Section 1.8.3 [Fontes], page 216, Section 1.8.2 [Mise en forme du texte], page 202.

Mesure incomplète et musique liturgique

Il arrive fréquemment que les chants liturgiques coportent des mesures incomplètes aussi bien en début qu'en fin de ligne, de telle sorte qu'à une portée corresponde une ligne de texte. Ceci requiert donc l'utilisation de la commande `\partial` en début de partition et d'une commande `\bar "|"` ou `\baràtie"||"` à la fin de chaque ligne.

Modèle pour cantique

Le code ci-dessous illustre la manière d'agencer un cantique liturgique dans lequel chaque ligne débute et se termine par une mesure incomplète. Vous noterez par ailleurs l'affichage des paroles indépendamment de la musique.

```
Timeline = {
  \time 4/4
  \tempo 4=96
  \partial 2
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||" \break
  s2 | s1 | s2 \breathe s2 | s1 | s2 \bar "||"
}

SopranoMusic = \relative g' {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

AltoMusic = \relative c' {
  d4 d | d d d d | d d d d | d d d d | d2
  d4 d | d d d d | d d d d | d d d d | d2
}

TenorMusic = \relative a {
  b4 b | b b b b | b b b b | b b b b | b2
  b4 b | b b b b | b b b b | b b b b | b2
}

BassMusic =  \relative g {
  g4 g | g g g g | g g g g | g g g g | g2
  g4 g | g g g g | g g g g | g g g g | g2
}

global = {
 \key g \major
```

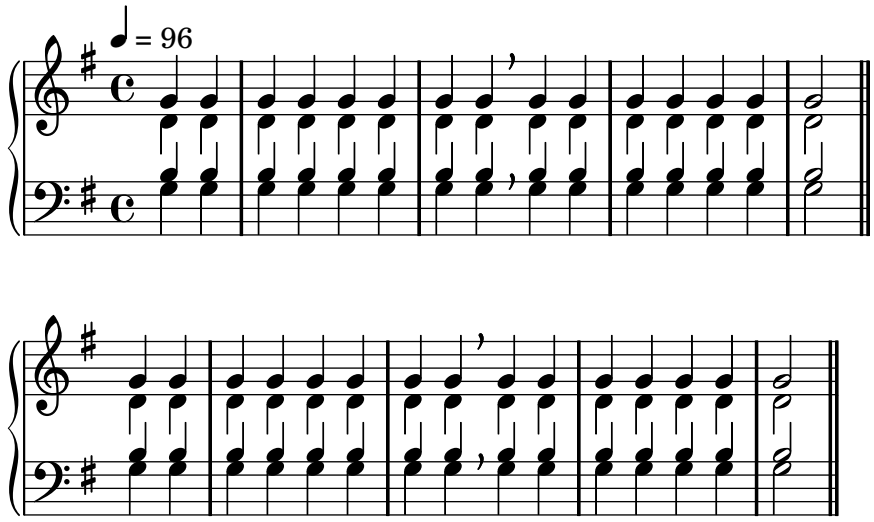
```
}

\score {  % Start score
  <<
    \new PianoStaff <<  % Start pianostaff
      \new Staff <<  % Start Staff = RH
        \global
        \clef "treble"
        \new Voice = "Soprano" <<  % Start Voice = "Soprano"
          \Timeline
          \voiceOne
          \SopranoMusic
        >>  % End Voice = "Soprano"
        \new Voice = "Alto" <<  % Start Voice = "Alto"
          \Timeline
          \voiceTwo
          \AltoMusic
        >>  % End Voice = "Alto"
      >>  % End Staff = RH
      \new Staff <<  % Start Staff = LH
        \global
        \clef "bass"
        \new Voice = "Tenor" <<  % Start Voice = "Tenor"
          \Timeline
          \voiceOne
          \TenorMusic
        >>  % End Voice = "Tenor"
        \new Voice = "Bass" <<  % Start Voice = "Bass"
          \Timeline
          \voiceTwo
          \BassMusic
        >>  % End Voice = "Bass"
      >>  % End Staff = LH
    >>  % End pianostaff
  >>
}  % End score

\markup {
  \fill-line {
    ""
    {
      \column {
        \left-align {
          "This is line one of the first verse"
          "This is line two of the same"
          "And here's line three of the first verse"
          "And the last line of the same"
        }
      }
    }
    ""
  }
```

```
}

\paper {  % Start paper block
  indent = 0     % don't indent first system
  line-width = 130   % shorten line length to suit music
}  % End paper block
```



This is line one of the first verse
This is line two of the same
And here's line three of the first verse
And the last line of the same

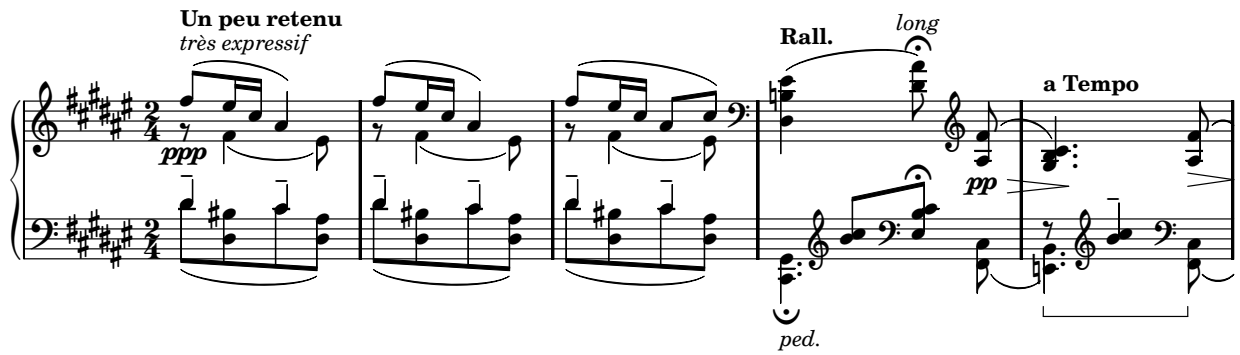
2.1.8 Musique vocale ancienne
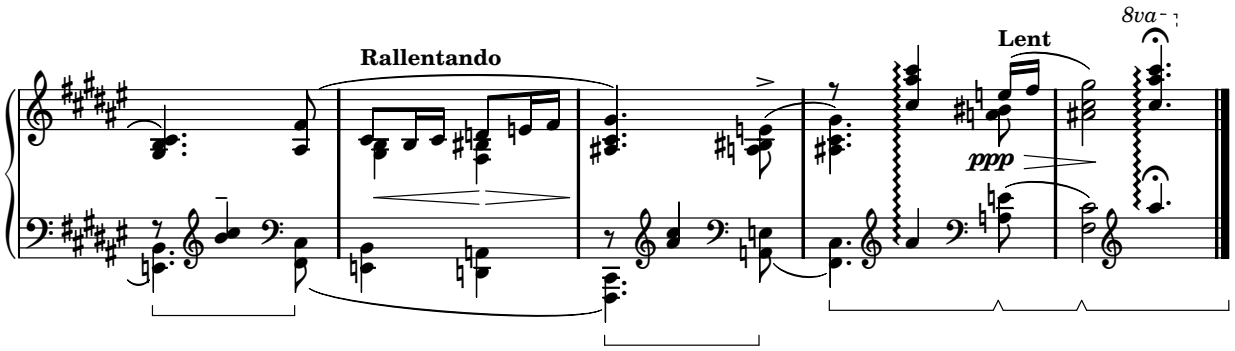
LilyPond prend en charge la musique vocale ancienne. Elle est abordée en détails au chapite Section 2.9 [Notations anciennes], page 354.

Voir aussi

Manuel de notation : Section 2.9 [Notations anciennes], page 354.

2.2 Instruments utilisant des portées multiples





Ce chapitre traite des différents aspects de la notation que l'on rencontre particulièrement avec les instruments qui ont recours à plusieurs portées, tels que ceux disposant de claviers, la harpe ou le vibraphone. Pour les besoins du discours, et pour simplifier, nous parlerons ici de « clavier » bien que le recours à des portées multiples concerne aussi des instruments qui en sont dépourvus.

2.2.1 Vue d'ensemble des claviers

Nous allons examiner ici les problèmes qui peuvent survenir en matière de notation pour la plupart des instrument à cordes multiples.

Généralités sur les instruments à clavier

La notation pour instrument à clavier est en règle générale présentée sous la forme d'un système pour piano, autrement dit deux portées normales ou plus réunies par une accolade. Cette notation sert également à la harpe ou à d'autres instruments à clés. L'organiste, quant à lui, lira une partition composée de deux portées au sein d'un système pianistique auquel vient s'adjoindre une portée normale pour le pédalier.

Les portées sont largement autonomes, mais il arrive que des voix passent de l'une à l'autre. Le contexte `PianoStaff` est précisément conçu pour gérer la notation spécifique au piano et autres instruments à clavier, notamment ces croisements.

Certaines particularités de la notation pour claviers sont abordées dans d'autres chapitres :

- Les claviers ont régulièrement recours à plusieurs voix dont le nombre peut varier. Voir à ce sujet [Résolution des collisions], page 144.
- On peut écrire la musique pour claviers de façon parallèle, comme l'explique [Saisie de musique en parallèle], page 151.
- Les nuances peuvent se gérer dans un contexte `Dynamics` qu'il suffira d'insérer entre les deux contextes `Staff` pour qu'elles apparaissent sur leur propre ligne centrée entre les deux portées. Voir à ce sujet [Nuances], page 104.
- Les indications de doigté sont abrodées dans [Doigtés], page 183.
- Les indications en matière de pédalier d'orgue sont traitées comme des articulations. Voir à ce sujet Section A.10 [Liste des signes d'articulation], page 553.
- Pour générer des repères verticaux, voir [Quadrillage temporel], page 190.
- En plus des *Laissez vibrer*, les liaisons en matière de clavier peuvent intervenir sur des accords arpégés ou des trémolos. Reportez-vous au chapitre [Liaisons de prolongation], page 44.
- Le traitement des arpèges couvrant plusieurs voix ou portées est abordé au chapitre [Arpèges], page 119.
- Une description des indications de tremolo est disponible au chapitre [Répétitions en trémolo], page 135.
- Certaines retouches particulières au monde des claviers sont abordées au chapitre Section "Exemple concret" dans *Manuel d'initiation*.
- Des notes fantômes permettent d'introduire des liaisons de tenue qui passent d'une voix à l'autre, comme le montre Section "Autres utilisations des retouches" dans *Manuel d'initiation*.

Voir aussi

Manuel d'initiation : Section "Exemple concret" dans *Manuel d'initiation*, Section "Autres utilisations des retouches" dans *Manuel d'initiation*.

Manuel de notation : [Regroupement de portées], page 155, [Noms d'instrument], page 171, [Résolution des collisions], page 144, [Saisie de musique en parallèle], page 151, [Doigtés], page 183, ⟨undefined⟩ [List of articulations], page ⟨undefined⟩, [Quadrillage temporel], page 190, [Liaisons de prolongation], page 44, [Arpèges], page 119, [Répétitions en trémolo], page 135.

Référence des propriétés internes : Section "PianoStaff" dans *Référence des propriétés internes*.

Morceaux choisis : Section "Claviers" dans *Morceaux choisis*.

Changement de portée manuel

Il est possible de passer d'une portée à l'autre de façon manuelle, au moyen de la commande

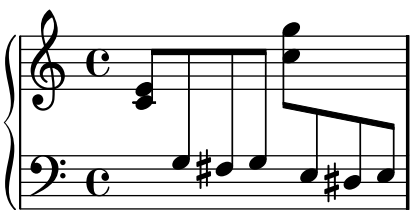
```
\change Staff = nomDeLaPortee
```

La valeur *nomDeLaPortee* est le nom de la portée sur laquelle va se déplacer la voix courante. Pour des raisons pratiques, on nomme la portée supérieure `"haut"` et la portée inférieure `"bas"`, donc *nomDeLaPortee* désigne habituellement `"haut"`, `"bas"`, `"MD"` ou `"MG"`.

Dans tous les cas, le contexte de portée ainsi utilisé doit exister au préalable. Le cas échéant, vous devrez « garder actives » certaines portées – voir Section 5.1.3 [Conservation d'un contexte], page 464 à ce propos.

Pour ligaturer automatiquement des notes entre deux portées, procédez ainsi :

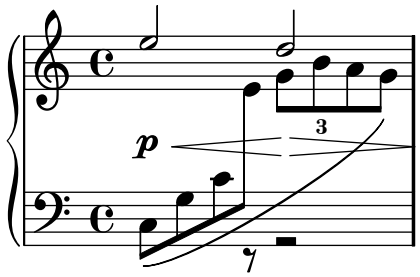
```
\new PianoStaff <<
  \new Staff = "up" {
    <e' c'>8
    \change Staff = "down"
    g8 fis g
    \change Staff = "up"
    <g'' c''>8
    \change Staff = "down"
    e8 dis e
    \change Staff = "up"
  }
  \new Staff = "down" {
    \clef bass
    % keep staff alive
    s1
  }
>>
```



Si les ligatures demandaient à être retouchées, commencez par modifier la direction des hampes. L'emplacement des ligatures sera alors calculé à partir du centre de la portée la plus proche – voir Section "Correction des collisions d'objets" dans *Manuel d'initiation* pour un exemple de retouche sur des ligatures.

Une voix qui change de portée peut entraîner des collisions :

```
\new PianoStaff <<
  \new Staff = "up" {
    \voiceOne
    % Make space for fingering in the cross-staff voice
    \once\override DynamicLineSpanner #'staff-padding = #3.4
    e''2\p\< d''\> s1*0\!
  }
  \new Staff = "down" <<
    {
      \clef bass
      s4. e,8\rest g,2\rest
    } \\ {
      c8\( g c'
      \change Staff = "up"
      e' g' b'-3 a' g'\)
    }
  >>
>>
```



Hampes et liaisons viennent en surimpression sur la ligne des nuances parce que la résolution automatique des collisions est inactivée pour ce qui relie les notes de différentes portées ainsi que pour les hampes ou extenseurs affectés à des notes incluses dans un changement de portée. Lorsque des collisions surviennent en pareil cas, il vous faudra les résoudre, en suivant les directives du chapitre Section “Correction des collisions d’objets” dans *Manuel d’initiation*.

Voir aussi

Manuel d’initiation : Section “Correction des collisions d’objets” dans *Manuel d’initiation*.

Manuel de notation : [Hampes], page 188, [Barres de ligature automatiques], page 70, Section 5.1.3 [Conservation d’un contexte], page 464.

Morceaux choisis : Section “Keyboards” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Beam” dans *Référence des propriétés internes*, Section “ContextChange” dans *Référence des propriétés internes*.
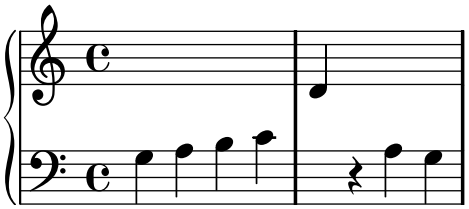
Changement de portée automatique

Les voix peuvent passer automatiquement d’une portée à l’autre, au moyen de la syntaxe suivante :

```
\autochange ...musique...
```

Deux portées seront alors créées au sein du contexte `PianoStaff`, nommées respectivement `"up"` et `"down"`. La portée du bas sera par défaut en clé de fa. La commande `\autochange` bascule les notes d’une portée à l’autre en fonction de leur hauteur (le do du milieu servant de pivot), et place les silences en fonction des notes qui les suivront. Ainsi :

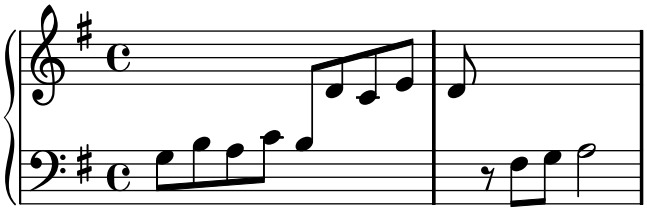
```
\new PianoStaff {
  \autochange {
    g4 a b c'
    d'4 r a g
  }
}
```



Une section en mode `\relative` se situant en dehors de la commande `\autochange` n'aura pas d'effet sur les hauteurs de l'expression *musique*. Il est donc préférable d'introduire la directive `\relative` **après** `\autochange`.

Lorsque des contrôles particuliers doivent s'appliquer aux portées, mieux vaut les nommer explicitement – attention : sous peine d'effet indésirable quant au résultat, la portée supérieure doit s'appeler `"up"` et l'inférieure `"down"` ; *ceci est à notre connaissance le seul cas où ces noms de variable sont figés*. Cette procédure sert, entre autres, à indiquer l'armure sur la portée inférieure :

```
\new PianoStaff <<
  \new Staff = "up" {
    \new Voice = "melodieUn" {
      \key g \major
      \autochange \relative c' {
        g8 b a c b d c e
        d8 r fis, g a2
      }
    }
  }
  \new Staff = "down" {
    \key g \major
    \clef bass
  }
>>
```



Voir aussi

Manuel de notation : [Changement de portée manuel], page 280.

Morceaux choisis : Section "Claviers" dans *Morceaux choisis*.

Référence des propriétés internes : Section "AutoChangeMusic" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les changements de portée automatiques n'interviennent pas toujours à l'endroit le plus opportun. Pour un résultat de meilleure qualité, il vaut mieux indiquer soi-même ces changements.

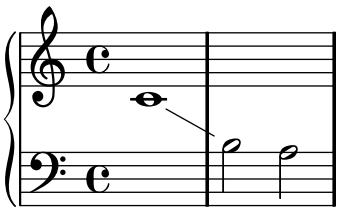
Un accord ne peut se répartir sur plusieurs portées ; sa portée d'affectation sera déterminée par la première hauteur mentionnée dans la construction de cet accord.

`\autochange` ne peut intervenir à l'intérieur d'une commande `\times`.

Lignes de changement de portée

Lorsqu'une voix change de portée, il est possible d'imprimer automatiquement une ligne reliant les notes, en faisant appel à la commande `\showStaffSwitch` :

```
\new PianoStaff <<
  \new Staff = "one" {
    \showStaffSwitch
    c1
    \change Staff = "two"
    b2 a
  }
  \new Staff = "two" {
    \clef bass
    s1*2
  }
>>
```



Commandes prédéfinies

`\showStaffSwitch`, `\hideStaffSwitch`.

Voir aussi

Morceaux choisis : Section "Claviers" dans *Morceaux choisis*.

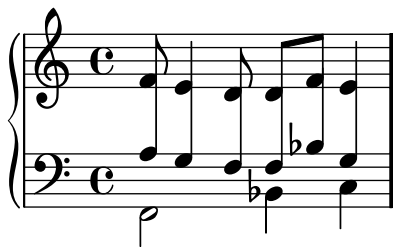
Référence des propriétés internes : Section "Note_head_line_engraver" dans *Référence des propriétés internes*, Section "VoiceFollower" dans *Référence des propriétés internes*.

Hampes et changements de portée

Pour écrire des accords qui enjambent deux portées, on allonge la hampe de l'accord de l'une des deux portées de façon à ce qu'elle rejoigne celle de l'autre portée.

```
\new PianoStaff <<
  \new Staff {
    \relative c' {
      f8 e4 d8 d f e4
    }
  }
  \new Staff {
    \relative c' {
```

```
      << {
        \clef bass
        % stems may overlap the other staff
        \override Stem #'cross-staff = ##t
        % extend the stems to reach the other staff
        \override Stem #'length = #12
        % do not print extra flags
        \override Stem #'flag-style = #'no-flag
        % prevent beaming as needed
        a8 g4 f8 f bes\noBeam g4
      }
      \\
      {
        f,2 bes4 c
      } >>
    }
  }
>>
```



Morceaux choisis

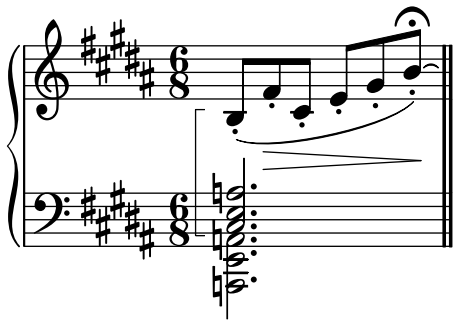
Indicating cross-staff chords with arpeggio bracket

An arpeggio bracket can indicate that notes on two different staves are to be played with the same hand. In order to do this, the `PianoStaff` must be set to accept cross-staff arpeggios and the arpeggios must be set to the bracket shape in the `PianoStaff` context.

(Debussy, Les collines d'Anacapri, m. 65)

```
\new PianoStaff <<
  \set PianoStaff.connectArpeggios = ##t
  \override PianoStaff.Arpeggio #'stencil = #ly:arpeggio::brew-chord-bracket
  \new Staff {
    \relative c' {
      \key b \major
      \time 6/8
      b8-.(\arpeggio fis'-.\> cis-. e-. gis-. b-.)\!\fermata^\laissezVibrer
      \bar "||"
    }
  }
  \new Staff {
    \relative c' {
      \clef bass
      \key b \major
      <<
        {
          <a e cis>2.\arpeggio
```

```
        }
        \\
        {
          <a, e a,>2.
        }
      >>
    }
  }
>>
```



Voir aussi

Morceaux choisis : Section “Claviers” dans *Morceaux choisis*.

Référence des propriétés internes : Section “Stem” dans *Référence des propriétés internes*.
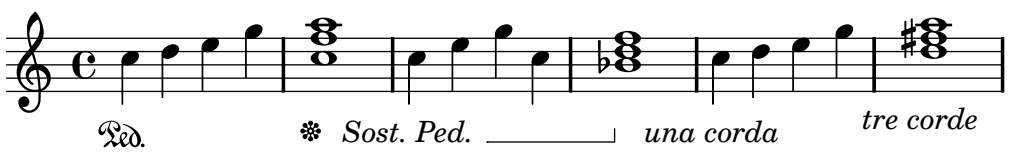
2.2.2 Piano

Ce chapitre traîte des aspects de la notation directement liés au piano.

Pédales de piano

Le piano possède deux pédales, parfois trois, permettant de modifier l’émission du son : une pédale de *tenue* (*sustain*), une pédale de *sourdine* (*una corda* ou *U.C.*) et une pédale *tonale* (*sostenuto* ou *sos.*). La pédale *sustain* se rencontre aussi sur les vibraphones et celestas.

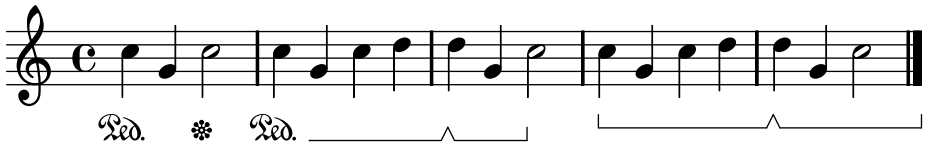
```
c4\sustainOn d e g
<c, f a>1\sustainOff
c4\sostenutoOn e g c,
<bes d f>1\sostenutoOff
c4\unaCorda d e g
<d fis a>1\treCorde
```



Trois styles sont à votre disposition pour imprimer les indications de pédale : sous forme de texte, de crochet, ou une combinaison des deux. `text` est le style de notation par défaut pour les pédales de tenue ou de sourdine — le traditionnel ‘*Ped.’. La pédale tonale, en revanche, utilise `mixed` par défaut.

```
c4\sustainOn g c2\sustainOff
\set Staff.pedalSustainStyle = #'mixed
```

```
c4\sustainOn g c d
d\sustainOff\sustainOn g, c2\sustainOff
\set Staff.pedalSustainStyle = #'bracket
c4\sustainOn g c d
d\sustainOff\sustainOn g, c2
\bar "|."
```



Le placement des commandes de pédale correspond au mouvement de la pédale de tenue pendant le jeu. Garder une pédale enclanchée jusqu'à la barre finale s'obtient tout simplement en omettant la commande de relâcher.

Les indication de pédale peuvent s'inscrire dans un contexte `Dynamics`, ce qui aura pour effet de leur attribuer une ligne en propre.

Voir aussi

Manuel de notation : [Liaisons de prolongation], page 44.

Morceaux choisis : Section "Claviers" dans *Morceaux choisis*.

Référence des propriétés internes : Section "SustainPedal" dans *Référence des propriétés internes*, Section "SustainPedalLineSpanner" dans *Référence des propriétés internes*, Section "SustainEvent" dans *Référence des propriétés internes*, Section "SostenutoPedal" dans *Référence des propriétés internes*, Section "SostenutoPedalLineSpanner" dans *Référence des propriétés internes*, Section "SostenutoEvent" dans *Référence des propriétés internes*, Section "UnaCordaPedal" dans *Référence des propriétés internes*, Section "UnaCordaPedalLineSpanner" dans *Référence des propriétés internes*, Section "UnaCordaEvent" dans *Référence des propriétés internes*, Section "PianoPedalBracket" dans *Référence des propriétés internes*, Section "Piano_pedal_engraver" dans *Référence des propriétés internes*.

2.2.3 Accordéon

Symboles de jeux

De nombreux accordéons possèdent plusieurs jeux d'anches capables de jouer à l'unisson ou bien à l'octave inférieure ou supérieure par rapport aux notes écrites. Chaque facteur d'accordéon donne des noms différents aux *tirettes* (*shifts*) qui permettent de sélectionner les différents jeux d'anches tels que *hautbois*, *musette* ou *bandonéon*, de telle sorte qu'un système de symbole a fini par voir le jour afin de faciliter les instructions pour exécuter un morceau.

Morceaux choisis

Accordion-discant symbols

Accordion discant-specific symbols are added using `\markup`. The vertical placement of the symbols can be tweaked by changing the `\raise` arguments.

```
discant = \markup {
  \musicglyph #"accordion.discant"
}
dot = \markup {
  \musicglyph #"accordion.dot"
}
```

```
\layout { ragged-right = ##t }

% 16 voets register
accBasson = ^\markup {
  \combine
  \discant
  \raise #0.5 \dot
}

% een korig 8 en 16 voets register
accBandon = ^\markup {
  \combine
    \discant
    \combine
      \raise #0.5 \dot
      \raise #1.5 \dot
}

accVCello = ^\markup {
  \combine
    \discant
    \combine
      \raise #0.5 \dot
      \combine
        \raise #1.5 \dot
        \translate #'(1 . 0) \raise #1.5 \dot
}

% 4-8-16 voets register
accHarmon = ^\markup {
  \combine
    \discant
    \combine
      \raise #0.5 \dot
      \combine
        \raise #1.5 \dot
        \raise #2.5 \dot
}

accTrombon = ^\markup {
  \combine
    \discant
    \combine
      \raise #0.5 \dot
      \combine
        \raise #1.5 \dot
        \combine
          \translate #'(1 . 0) \raise #1.5 \dot
          \translate #'(-1 . 0) \raise #1.5 \dot
}

% eenkorig 4 en 16 voets register
```

```
accOrgan = ^\markup {
  \combine
    \discant
    \combine
      \raise #0.5 \dot
      \raise #2.5 \dot
}

accMaster = ^\markup {
  \combine
    \discant
    \combine
      \raise #0.5 \dot
      \combine
        \raise #1.5 \dot
        \combine
          \translate #'(1 . 0) \raise #1.5 \dot
          \combine
            \translate #'(-1 . 0) \raise #1.5 \dot
            \raise #2.5 \dot
}

accAccord = ^\markup {
  \combine
    \discant
    \combine
      \raise #1.5 \dot
      \combine
        \translate #'(1 . 0) \raise #1.5 \dot
        \combine
          \translate #'(-1 . 0) \raise #1.5 \dot
          \raise #2.5 \dot
}

accMusette = ^\markup {
  \combine
    \discant
    \combine
      \raise #1.5 \dot
      \combine
        \translate #'(1 . 0) \raise #1.5 \dot
        \translate #'(-1 . 0) \raise #1.5 \dot
}

accCeleste = ^\markup {
  \combine
    \discant
    \combine
      \raise #1.5 \dot
      \translate #'(-1 . 0) \raise #1.5 \dot
}
```

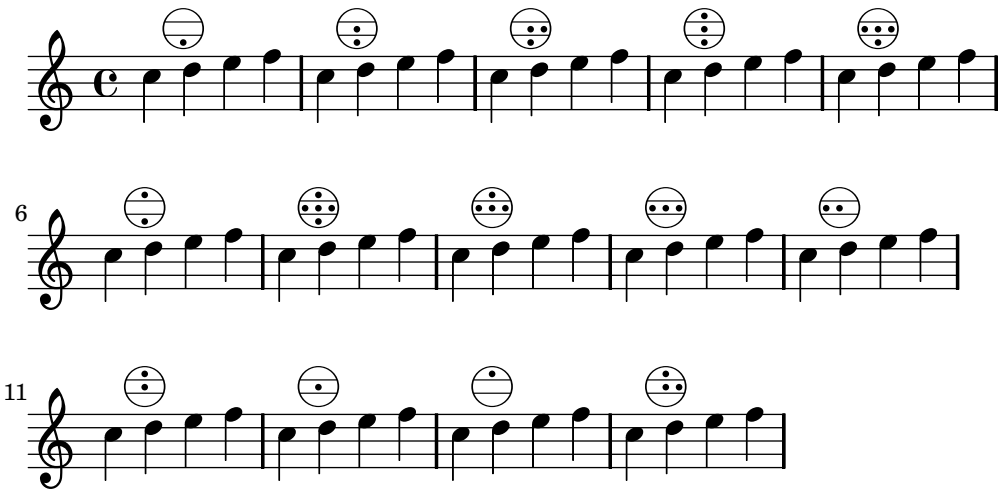
```
accOboe = ^\markup {
  \combine
    \discant
    \combine
      \raise #1.5 \dot
      \raise #2.5 \dot
}

accClarin = ^\markup {
  \combine
    \discant
    \raise #1.5 \dot
}

accPiccolo = ^\markup {
    \combine
       \discant
       \raise #2.5 \dot
}

accViolin = ^\markup {
  \combine
    \discant
    \combine
      \raise #1.5 \dot
      \combine
        \translate #'(1 . 0) \raise #1.5 \dot
        \raise #2.5 \dot
}

\relative c'' {
  c4 d\accBasson e f
  c4 d\accBandon e f
  c4 d\accVCello e f
  c4 d\accHarmon e f
  c4 d\accTrombon e f
  \break
  c4 d\accOrgan e f
  c4 d\accMaster e f
  c4 d\accAccord e f
  c4 d\accMusette e f
  c4 d\accCeleste e f
  \break
  c4 d\accOboe e f
  c4 d\accClarin e f
  c4 d\accPiccolo e f
  c4 d\accViolin e f
}
```



Voir aussi

Morceaux choisis : Section “Keyboards” dans *Morceaux choisis*.

2.2.4 Harpe

This section discusses notation issues that are unique to the harp.

Généralités sur la harpe

Certaines caractéritiques de la musique pour harpes sont abordées dans d’autres chapitres, telles que :

- Les glissandos, l’une des techniques spécifique à la harpe, [Glissando], page 118.
- Le *bisbigliando*, qui s’écrit comme un trémolo, [Répétitions en trémolo], page 135.
- Les harmoniques naturelles sont présentées dans [Harmoniques], page 293.
- For directional arpeggios and non-arpeggios, see [Arpèges], page 119.
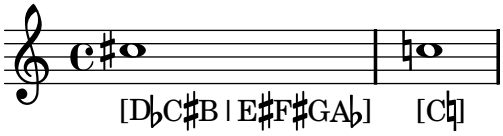
Voir aussi

Manuel de notation : [Répétitions en trémolo], page 135, [Glissando], page 118, [Arpèges], page 119, [Harmoniques], page 293.

Pédales de harpe

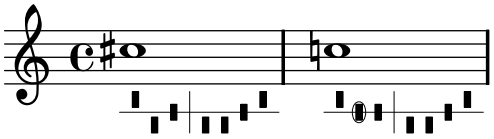
Les harpes comportent sept cordes par octave qui peuvent sonner naturel, dièse ou bémol. Si chacune des cordes de la harpe celtique (*lever harp*) est accordée individuellement, celles d’une harpe à pédalier ayant la même note de base sont contrôlées par une seule pédale. De gauche à droite, elles correspondent aux notes ré, do, si, et mi, fa, sol, la pour la grande harpe. Les trois premières pédales sont réservées au pied gauche, les quatre dernières au pied droit. Leur position peut être indiquée par une marque textuelle :

```
\textLengthOn
cis1_\markup \concat \vcenter {
  [D \flat C \sharp B|E \sharp F \sharp G A \flat] }
c!1_\markup \concat \vcenter {
  [ C \natural ]}
```



ou bien sous forme de diagrammes :

```
\textLengthOn
cis1_\markup { \harp-pedal #"^v-|vv-^" }
c!1_\markup { \harp-pedal #"^o--|vv-^" }
```
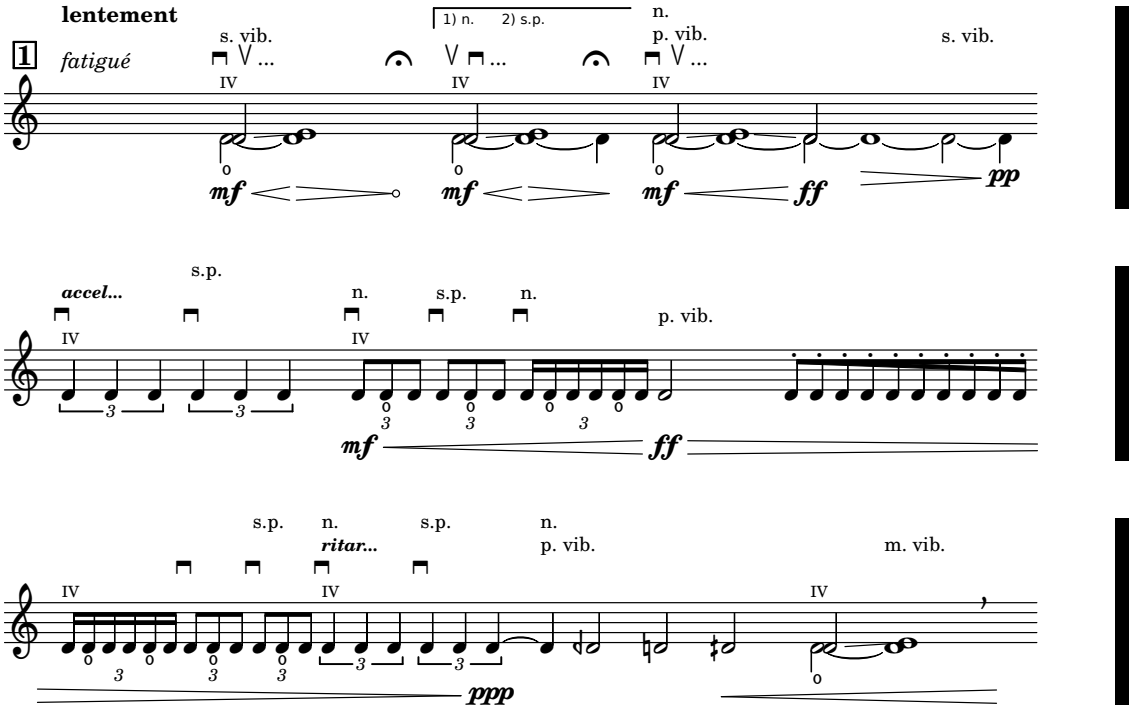


bémol si la pédale est relâchée (ou en haut), bécarre si elle est bloquée sur le cran du milieu, et dièse si elle est tout à fait enfoncée. La commande `\harp-pedal` prend en argument une chaîne de caractères, où `^` indique que la pédale est relâchée ou en haut (bémol), `-` qu'elle est bloquée sur le cran du milieu (bécarre), `v` qu'elle est tout à fait enfoncée (dièse) et `|` représente le séparateur (entre gauche et droite de l'instrumentiste). Faire précéder un symbole par un `o` permet de l'inscrire dans un cercle.

Voir aussi

Manuel de notation : [Commentaires textuels], page 194, Section A.8.5 [Instrument Specific Markup], page 543.

2.3 Cordes non frettées



Cette section dispense des informations supplémentaires et utiles à l'écriture pour les cordes non frettées, et tout spécialement pour les cordes d'orchestre.

2.3.1 Vue d'ensemble de la notation pour cordes non frettées

Il existe peu d'éléments spécifiques à l'écriture pour instruments à cordes non frettées. La musique est notée sur une portée et ne nécessite généralement qu'une seule voix. Le recours à deux voix distinctes peut cependant permettre de traiter efficacement des doubles notes ou des divisions.

Références en matière de cordes non frettées

La majorité des éléments de notation nécessaires en matière de cordes d'orchestre et autres instruments à archet est abordée dans d'autres chapitres de la documentation :

- Les indications textuelles telles que « pizz. » ou « arco » sont ajoutées comme du texte simple – voir à ce sujet [Commentaires textuels], page 194.
- Les indications de doigtés, incluant les indications du pouce, sont décrites dans [Doigtés], page 183.
- Les doubles notes sont généralement indiquées en écrivant un accord, voir [Notes en accords], page 137. Les précisions pour jouer des accords peuvent être ajoutées, comme l'indique [Arpèges], page 119.
- Un modèle pour quatuor à cordes est disponible au chapitre Section "Quatuor à cordes" dans *Manuel d'initiation*. D'autres informations se trouvent dans les exemples de code.

Voir aussi

Manuel d'initiation : Section "Quatuor à cordes" dans *Manuel d'initiation*.

Manuel de notation : [Commentaires textuels], page 194, [Doigtés], page 183, [Notes en accords], page 137, [Arpèges], page 119.
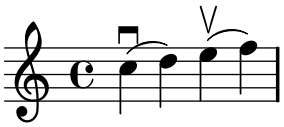
Morceaux choisis : Section "Unfretted strings" dans *Morceaux choisis*.

Indications d'archet

Les indications d'archet se créent comme des articulations, elles sont décrites dans [Articulations et ornements], page 102.

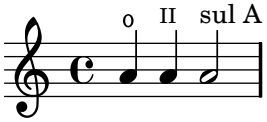
Les indications d'archet, poussé (`\upbow`) et tiré (`\downbow`), peuvent se combiner à des liaisons comme ici :

```
c4(\downbow d) e(\upbow f)
```



ou dans l'exemple suivant qui montre trois manières d'indiquer la corde de La, à vide, sur un violon :

```
a4 \open
a^\markup { \teeny "II" }
a2^\markup { \small "sul A" }
```



Commandes prédéfinies

`\downbow`, `\upbow`, `\open`.

Voir aussi
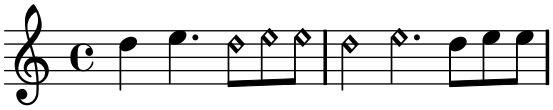
Manuel de notation : [Articulations et ornements], page 102, [Liaisons d'articulation], page 111.

Harmoniques

Harmoniques naturels
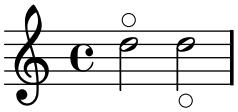
Les harmoniques naturels sont indiqués de différentes manières : une tête de note en forme de losange signifie généralement que vous devez effleurer la corde à l'endroit où vous l'auriez pincée si la note avait été normale.

```
d4 e4.
\harmonicsOn
d8 e e
d4 e4.
\harmonicsOff
d8 e e
```



Une autre façon de procéder consiste à faire surmonter la note normale d'un petit cercle. Ceci indique que la note écrite doît être jouée en harmonique :

```
d2^\flageolet d_\flageolet
```
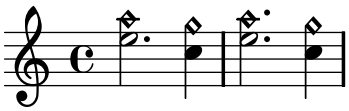


Un plus petit cercle peut être créé, comme indiqué dans les exemples de code contenus dans [Références en matière de cordes non frettées], page 292.

Harmoniques artificiels

Les harmoniques artificiels sont indiqués par une double tête de note : l'une, normale, indique la note à pincer, et l'autre, en forme de losange, indique l'endroit où la corde doit être effleurée.

La propriété `harmonicDots`, lorsqu'elle est activée, permet d'ajouter un point aux notes pointées affublées d'un `\harmonic`.

```
<e a\harmonic>2.  <c g'\harmonic>4
\set harmonicDots = ##t
<e a\harmonic>2.  <c g'\harmonic>4
```



Note : La commande `\harmonic` **doit** être placée dans une construction de type accord même s'il n'y a qu'une seule note. On utiliserait plutôt la commande `\harmonicsOn` en pareil cas.

Voir aussi

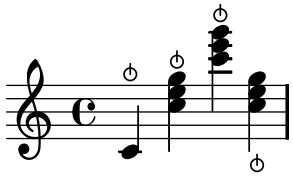
Glossaire musicologique : Section "harmonics" dans *Glossaire*.

Manuel de notation : [Têtes de note spécifiques], page 30, [Références en matière de cordes non frettées], page 292.

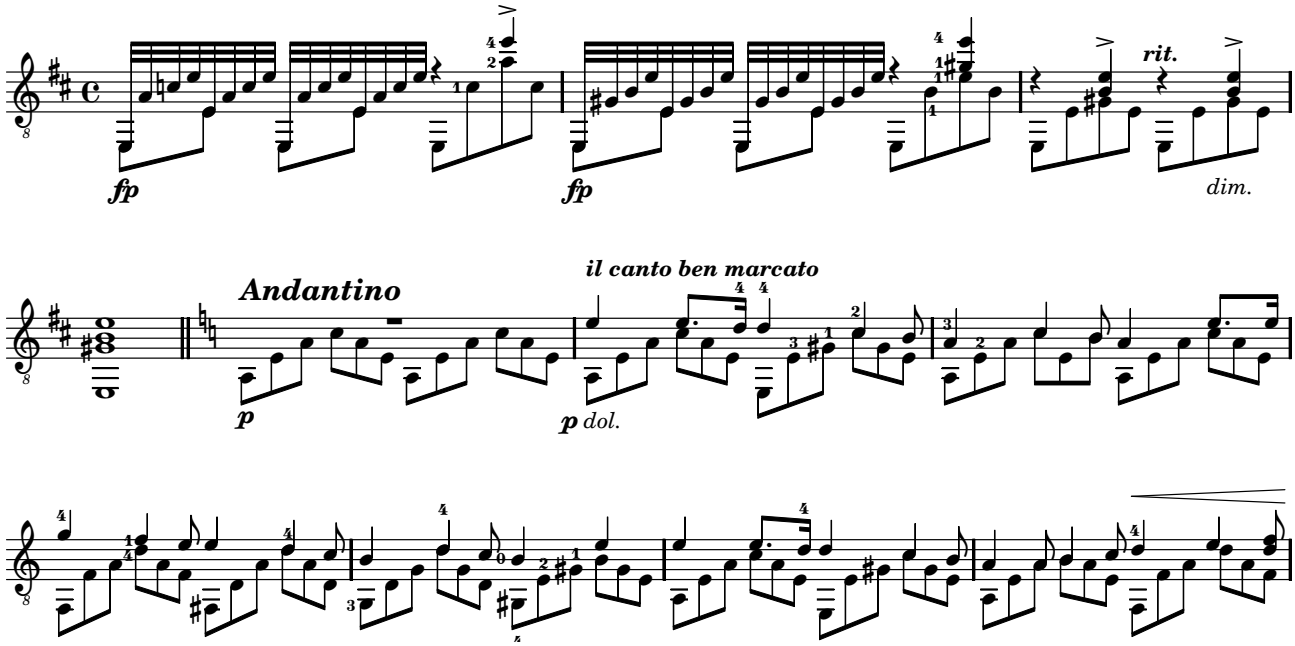
Snap (Bartók) pizzicato

Un *snap pizzicato*, aussi appelé « Bartok pizz » est un type de pizzicato pour lequel la corde est tirée vers le haut (plutôt que sur le côté) de telle sorte qu'elle vienne frapper le manche.

```
c4\snappizzicato
<c' e g>4\snappizzicato
<c' e g>4^\snappizzicato
<c, e g>4_\snappizzicato
```



2.4 Instruments à cordes frettées



Cette section traite de différents aspects de la notation propre aux instruments à cordes frettées.

2.4.1 Vue d'ensemble des cordes frettées

Vue d'ensemble de la notation pour les instruments à cordes frettées.

Références en matière de cordes frettées

La musique pour instruments à cordes frettées est généralement notée sur une seule portée, en notation traditionnelle ou en tablature, les deux étant parfois combinées. Il est aussi courant en musique populaire d'utiliser des diagrammes d'accord au-dessus de la portée traditionnelle. La guitare et le banjo sont des instruments transpositeurs, sonnant une octave au-dessous de ce qui est écrit. Les partitions pour ces instruments utilisent donc la clé de sol ottava bassa `"treble_8"`. Vous pourrez trouver ailleurs dans la documentation d'autres éléments aussi utilisés par les instruments à cordes frettées :

- Les doigtés s'obtiennent comme indiqué au chapitre [Doigtés], page 183.

- En plus des *Laissez vibrer*, les liaisons peuvent intervenir sur des accords arpégés ou des trémolos. Reportez-vous au chapitre [Liaisons de prolongation], page 44.
- Des indications quant au support polyphonique se trouvent au chapitre [Résolution des collisions], page 144.
- La notation des sons harmoniques se trouve à la section [Harmoniques], page 293.

Voir aussi

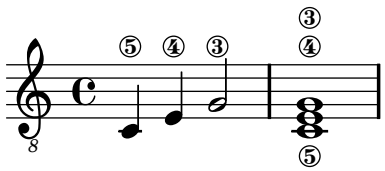
Manuel de notation : [Doigtés], page 183, [Liaisons de prolongation], page 44, [Résolution des collisions], page 144, [Noms d'instrument], page 171, [Saisie de musique en parallèle], page 151, [Arpèges], page 119, Section A.10 [Liste des signes d'articulation], page 553, [Clefs], page 13.

Indications du numéro de corde

Une corde sur laquelle une note doit être jouée peut être indiquée en attachant `\numéro` à cette note prise dans une construction de type accord `<>`.

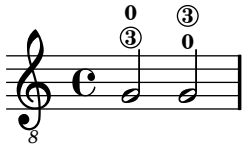
> **Note :** Les numéros de cordes **doivent** être définis dans une construction de type accord même s'il n'y a qu'une seule note.

```
\clef "treble_8"
<c\5>4 <e\4> <g\3>2
<c,\5 e\4 g\3>1
```



Quand les indications de doigté et de numéro de corde sont attachées à une même note, leur positionnement se fera en fonction de l'ordre dans lequel ils apparaissent dans le code :

```
\clef "treble_8"
<g\3-0>2
<g-0\3>
```



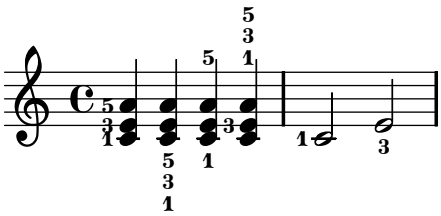
Morceaux choisis

Contrôler la position des doigtés dans un accord

Le positionnement des doigtés peut être contrôlé de manière très précise. Afin que l'orientation soit prise en compte, il est nécessaire d'utiliser une syntaxe d'accord `< >`, même s'il ne s'agit que d'une seule note.

```
\relative c' {
  \set fingeringOrientations = #'(left)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(down)
  <c-1 e-3 a-5>4
```

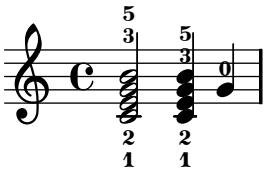
```
  \set fingeringOrientations = #'(down right up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(up)
  <c-1 e-3 a-5>4
  \set fingeringOrientations = #'(left)
  <c-1>2
  \set fingeringOrientations = #'(down)
  <e-3>2
}
```



Impression des doigtés à l'intérieur de la portée

L'empilement des indications de doigté se fait par défaut à l'extérieur de la portée. Néanmoins, il est possible d'annuler ce comportement.

```
\relative c' {
  <c-1 e-2 g-3 b-5>2
  \override Fingering #'staff-padding = #'()
  <c-1 e-2 g-3 b-5>4 <g'-0>
}
```



Voir aussi

Manuel de notation : [Doigtés], page 183.

Morceaux choisis : Section "Cordes frettées" dans *Morceaux choisis*.

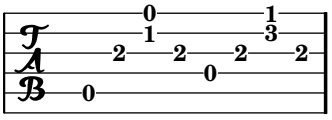
Référence des propriétés internes : Section "StringNumber" dans *Référence des propriétés internes*, Section "Fingering" dans *Référence des propriétés internes*.

Tablatures par défaut

La notation en tablature est utilisée pour certains instruments à cordes pincées. Les hauteurs n'y sont pas indiquées par des têtes de note, mais par des chiffres ou autres symboles, qui indiquent sur quelle corde, et à quelle case chaque note doit être jouée. Dans certaines limites, LilyPond vous permet d'écrire des tablatures.

Chaque numéro de corde associé à une note doit être précédé d'une barre oblique inverse (ou « antislash ») suivi d'un nombre. Par défaut, la première corde est la plus aiguë, et les cordes suivent l'accordage traditionnel d'une guitare (à six cordes). Les notes sont imprimées sous forme de tablature, dans les contextes `TabStaff` et `TabVoice`.

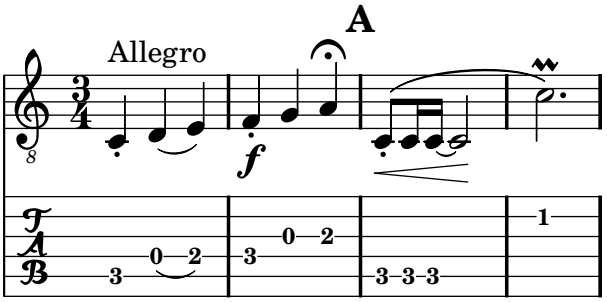
```
\new TabStaff \relative c' {
        a,8 a' <c e> a
        d, a' <d f> a
}
```



Par défaut, les tablatures ne comportent aucune marque de durée ni de symbole musical tel que des nuances.

```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-.\f g a^\fermata
  \mark \default
  c8_.\<\( c16 c ~ c2\!
  c'2.\prall\)
}

\score {
  <<
    \new Staff { \clef "G_8" \symbols }
    \new TabStaff   { \symbols }
  >>
}
```

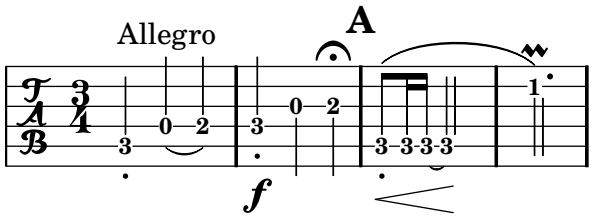


Pour obtenir les symboles de notation habituelle dans une tablature, il faut appliquer la commande `\tabFullNotation` au contexte `TabStaff`. Vous noterez par ailleurs que les blanches sont affublées d'une double hampe afin de ne pas les confondre avec des noires.

```
symbols = {
  \time 3/4
  c4-.^"Allegro" d( e)
  f4-.\f g a^\fermata
  \mark \default
  c8_.\<\( c16 c ~ c2\!
  c'2.\prall\)
}

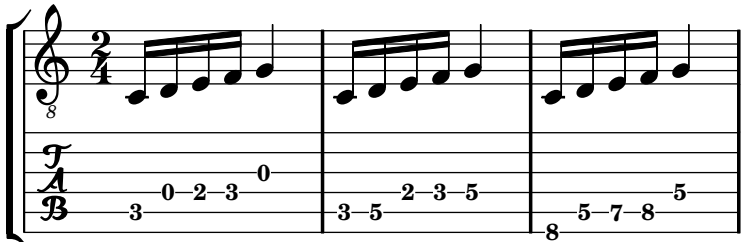
\score {
  \new TabStaff {
```

```
    \tabFullNotation
    \symbols
  }
}
```



Quand aucune corde n'est précisée, LilyPond choisit automatiquement la corde pour laquelle la position est la moins élevée et qui donne un numéro de case supérieur ou égal à la valeur de la propriété `minimumFret`. La valeur par défaut de cette propriété est fixée à 0, ce qui correspond aux cordes à vide.

```
\new StaffGroup <<
   \new Staff \relative c {
     \clef "treble_8"
     \time 2/4
     c16 d e f g4
     c,16\5 d\5 e\4 f\4 g4\4
     c,16 d e f g4
   }
   \new TabStaff \relative c {
     c16 d e f g4
     c,16\5 d\5 e\4 f\4 g4\4
     \set TabStaff.minimumFret = #5
     c,16 d e f g4
   }
>>
```

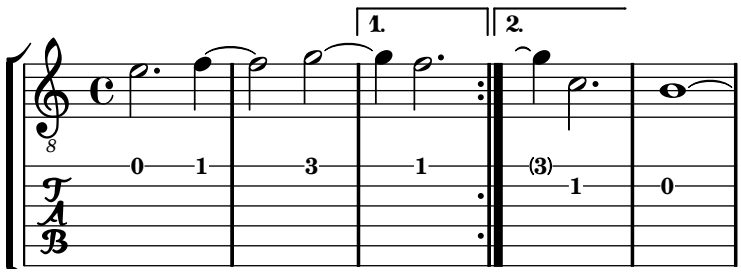


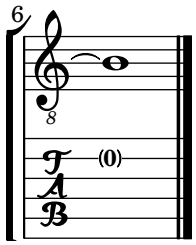
Lorsqu'un liaison de prolongation intervient lors d'un saut de ligne, la note est répétée, entre parenthèses. Il en va de même pour la seconde alternative d'une répétition.

```
ties = \relative c' {
        \repeat volta 2 {
        e2. f4~
        f2 g2~ }
        \alternative {
                { g4 f2. }
                { g4\repeatTie c,2. }
        }
        b1~\break
        b1  \bar "|."
```

```
}

\score {
  <<
    \new StaffGroup  <<
    \context Staff  {
        \clef "treble_8"
        \ties
    }
      \context TabStaff  {
        \ties
      }
    >>
  >>
  \layout {
  indent = #0
  ragged-right = ##t
  }
}
```



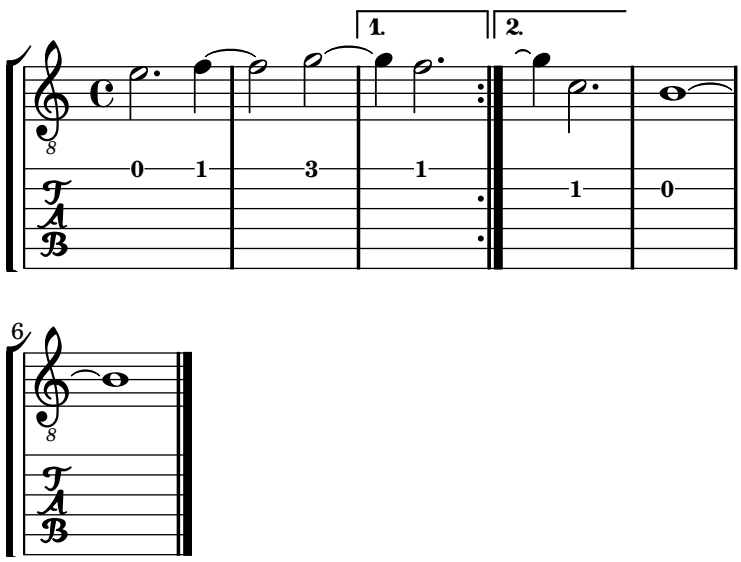


La commande `\hideSplitTiedTabNotes` permet d'éviter d'imprimer ces cases entre parenthèses.

```
ties = \relative c' {
        \repeat volta 2 {
        e2. f4~
        f2 g2~ }
        \alternative {
                { g4 f2. }
                { g4\repeatTie c,2. }
        }
        b1~\break
        b1  \bar "|."
}

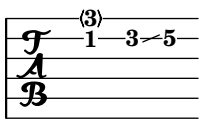
\score {
```

```
  <<
    \new StaffGroup  <<
    \context Staff  {
        \clef "treble_8"
        \ties
    }
      \context TabStaff  {
      \hideSplitTiedTabNotes
        \ties
      }
    >>
  >>
  \layout {
  indent = #0
  ragged-right = ##t
  }
}
```



Les indications de sons harmoniques ainsi que les glissandi peuvent être ajoutés aux tablatures.

```
\new TabStaff {
  \new TabVoice {
    <c g'\harmonic> d\2\glissando e\2
  }
}
```

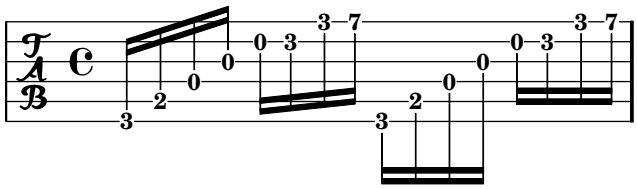


Morceaux choisis

Hampes et ligatures en mode tablature

La direction des hampes se gère dans les tablatures tout comme en notation traditionnelle. Les ligatures peuvent être mises à l'horizontale comme le montre cet exemple.

```
\new TabStaff {
  \relative c {
    \tabFullNotation
    g16 b d g b d g b
    \stemDown
    \override Beam #'damping = #+inf.0
    g,,16 b d g b d g b
  }
}
```



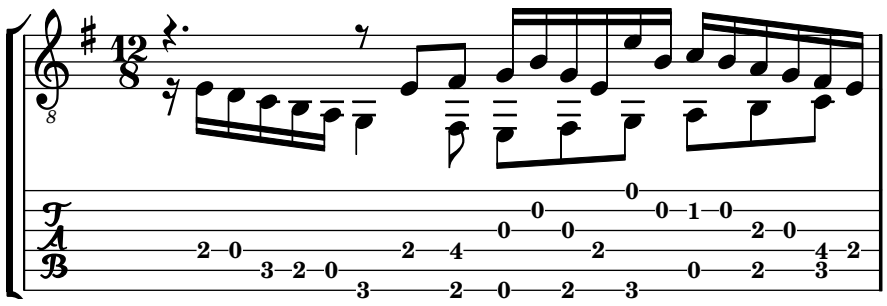
Polyphonie en mode tablature

Une section polyphonique s'obtient dans un `TabStaff` de la même manière que dans une portée normale.

```
upper = \relative c' {
  \time 12/8
  \key e \minor
  \voiceOne
  r4. r8 e, fis g16 b g e e' b c b a g fis e
}

lower = \relative c {
  \key e \minor
  \voiceTwo
  r16 e d c b a g4 fis8 e fis g a b c
}

\score {
  <<
    \new StaffGroup = "tab with traditional" <<
      \new Staff = "guitar traditional" <<
        \clef "treble_8"
        \context Voice = "upper" \upper
        \context Voice = "lower" \lower
      >>
      \new TabStaff = "guitar tab" <<
        \context TabVoice = "upper" \upper
        \context TabVoice = "lower" \lower
      >>
    >>
  >>
}
```



Voir aussi

Manuel de notation : [Hampes], page 188.

Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*.

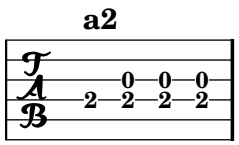
Référence des propriétés internes : Section “TabNoteHead” dans *Référence des propriétés internes*, Section “TabStaff” dans *Référence des propriétés internes*, Section “TabVoice” dans *Référence des propriétés internes*, Section “Beam” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

Les accords ne subissent aucun traitement particulier ; de ce fait, la sélection automatique des cordes peut attribuer une même corde pour deux notes différentes de l’accord.

Afin que `\partcombine` fonctionne avec des tablatures, on doit ajouter au contexte `TabStaff` des voix fantômes :

```
melodia = \partcombine { e4 g g g }{ e4 e e e }
<<
  \new TabStaff <<
    \new TabVoice = "one" s1
    \new TabVoice = "two" s1
    \new TabVoice = "shared" s1
    \new TabVoice = "solo" s1
    { \melodia }
  >>
>>
```



Le support des modes de jeu propres à la guitare se limite aux sons harmoniques et aux glissandi.

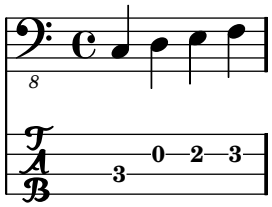
Tablatures personnalisées

Sous LilyPond, la case correspondant à une note jouée sur une corde donnée est calculée automatiquement. Pour ce faire, l’accordage doit être spécifié. L’accordage des cordes est donné par la propriété `StringTunings`.

LilyPond possède des accordages prédéfinis pour banjo, mandoline, guitare et guitare basse. Lilypond calcule automatiquement la transposition correspondant à ces accordages. L’exemple ci-dessous est pour guitare basse, il sonnera donc une octave en dessous de ce qui est écrit.

```
<<
  \new Staff {
    \clef "bass_8"
    \relative c, {
      c4 d e f
```

```
    }
  }
  \new TabStaff {
    \set TabStaff.stringTunings = #bass-tuning
    \relative c, {
      c4 d e f
    }
  }
>>
```



L'accordage par défaut est `guitar-tuning` ; il correspond à l'accordage standard d'une guitare : mi la ré sol si mi (EADGBE). D'autres accordages prédéfinis sont disponibles : `guitar-open-g-tuning`, `mandolin-tuning` et `banjo-open-g-tuning`. Les accordages prédéfinis sont répertoriés dans le fichier '`scm/output-lib.scm`'.

L'accordage est constitué par une liste Scheme des hauteurs de notes correspondant aux cordes à vide, une note pour chaque corde, classée par numéro de corde de 1 à n, où la corde 1 est la plus haute dans la tablature et n la plus basse. Cela revient généralement à classer les cordes de la plus aiguë à la plus grave, mais certains instruments (comme le ukulele) n'ont pas les cordes classées par hauteur.

Dans cette liste, la hauteur des notes est représentée par l'intervalle, exprimé en demi-tons, du do central jusqu'à la note de la corde à vide. Elle doit donc être représentée par un nombre entier. LilyPond calcule la note correspondant à chaque corde en ajoutant cet intervalle au do central.

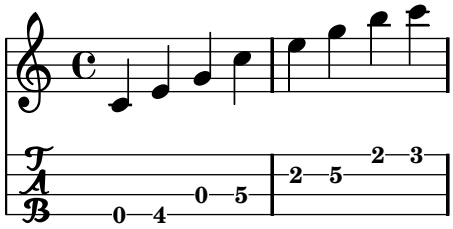
Lilypond calcule automatiquement le nombre de cordes à représenter dans la tablature (`TabStaff`) en comptant le nombre d'éléments définis dans le `stringTunings`.

N'importe quel accordage peut donc être créé. Par exemple, nous pouvons définir l'accordage d'un instrument à quatre cordes accordées do sol ré la, donc de la plus aiguë à la plus grave et en anglais : `a''`, `d''`, `g'`, et `c'` :

```
mynotes = {
    c'4 e' g' c'' |
    e'' g'' b'' c'''
}

<<
  \new Staff {
    \clef treble
    \mynotes
  }
  \new TabStaff {
    \set TabStaff.stringTunings = #`(,(ly:make-pitch 1 5 0)
                                     ,(ly:make-pitch 1 1 0)
                                     ,(ly:make-pitch 0 4 0)
                                     ,(ly:make-pitch 0 0 0))
    \mynotes
```

```
  }
>>
```



Voir aussi

Fichiers d'initialisation : '`scm/output-lib.scm`'.

Morceaux choisis : Section "Cordes frettées" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Tab_note_heads_engraver" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

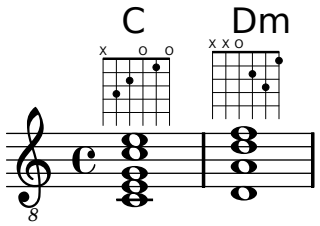
Le calcul automatique des tablatures se révèle inapproprié pour les instruments dont l'ordre des cordes ne correspond pas à l'ordre des hauteurs, comme le ukulele.

Tablatures sous forme d'étiquette

On peut ajouter des diagrammes d'accords au-dessus de n'importe quelle note, en tant qu'objets `\markup`. Ces objets contiennent toutes les informations du diagramme d'accord. Il y a pour les définir trois interfaces "diagramme d'accord" (*fret-diagram* en anglais) : *standard*, *terse* et *verbose*. Ces trois interfaces produiront des diagrammes similaires mais permettent des niveaux de personnalisation différents. Des détails à propos des interfaces de type `\markup` peuvent être trouvés dans Section A.8 [Text markup commands], page 510.

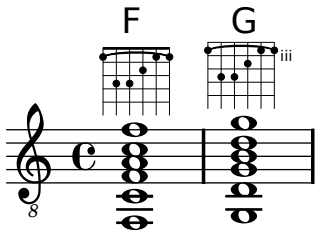
Dans l'interface standard des diagrammes d'accords, on indiquera le numéro de corde et le numéro de case pour chaque point placé sur une corde. Les cordes à vide et étouffées peuvent aussi être indiquées.

```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram #"6-x;5-3;4-2;3-o;2-1;1-o;"
    < d a d' f'> ^\markup
      \fret-diagram #"6-x;5-x;4-o;3-2;2-3;1-1;"
  }
>>
```



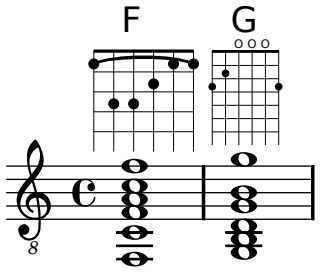
Les indications de barré peuvent aussi être ajoutées au diagramme d'accord dans l'interface standard :

```
<<
  \context ChordNames {
     \chordmode {
       f1 g
     }
  }
  \context Staff {
    \clef "treble_8"
    < f, c f a c' f'>1 ^\markup
      \fret-diagram #"c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    < g, d g b d' g'> ^\markup
      \fret-diagram #"c:6-1-3;6-3;5-5;4-5;3-4;2-3;1-3;"
  }
>>
```



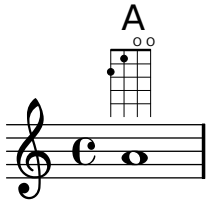
La taille du diagramme d'accord ainsi que le nombre de cases représentées peuvent aussi être modifiés dans l'interface standard.

```
<<
  \context ChordNames {
     \chordmode {
       f1 g
     }
  }
  \context Staff {
    \clef "treble_8"
    < f, c f a c' f'>1 ^\markup
      \fret-diagram #"s:1.5;c:6-1-1;6-1;5-3;4-3;3-2;2-1;1-1;"
    < g, b, d g b g'> ^\markup
      \fret-diagram #"h:6;6-3;5-2;4-o;3-o;2-o;1-3;"
  }
>>
```



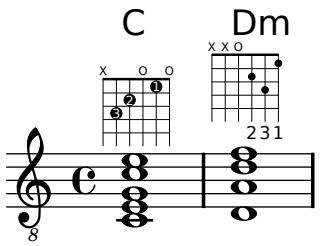
Le nombre de cordes dans les diagrammes d'accord peut être modifié dans l'interface standard pour s'adapter à différents instruments tels que le banjo et le ukulele.

```
<<
  \context ChordNames {
     \chordmode {
       a1
     }
  }
  \context Staff {
        %% A chord for ukulele
    a'1 ^\markup \fret-diagram #"w:4;4-2-2;3-1-1;2-o;1-o;"
  }
>>
```



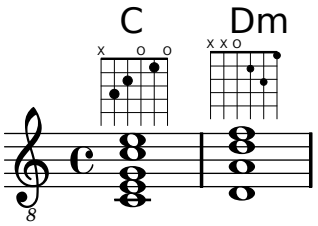
Des indications de doigtés peuvent être ajoutées, et le positionnement de ces doigtés peut être modifié dans l'interface standard.

```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram #"f:1;6-x;5-3-3;4-2-2;3-o;2-1-1;1-o;"
    < d a d' f'> ^\markup
      \fret-diagram #"f:2;6-x;5-x;4-o;3-2-2;2-3-3;1-1-1;"
  }
>>
```



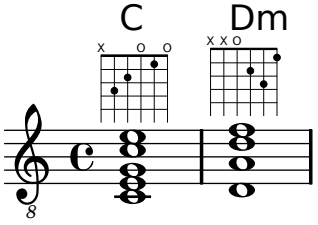
La taille ainsi que la position des points peuvent être contrôlées dans l'interface standard.

```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram #"d:0.35;6-x;5-3;4-2;3-o;2-1;1-o;"
    < d a d' f'> ^\markup
      \fret-diagram #"p:0.2;6-x;5-x;4-o;3-2;2-3;1-1;"
  }
>>
```



Dans l'interface `fret-diagram-terse`, les numéros de cordes sont omis ; les numéros de cordes sont induits par la présence de points-virgules. Il y a un point-virgule pour chaque corde du diagramme. Le premier point-virgule correspondant au plus haut numéro de corde, le dernier à la première corde. Les cordes étouffées, les cordes à vide ainsi que les numéros de case peuvent y être indiqués.

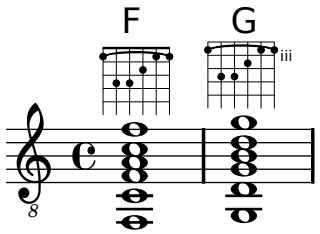
```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram-terse #"x;3;2;o;1;o;"
    < d a d' f'> ^\markup
      \fret-diagram-terse #"x;x;o;2;3;1;"
  }
>>
```



Les indications de barré peuvent être incluses dans l'interface `fret-diagram-terse`.

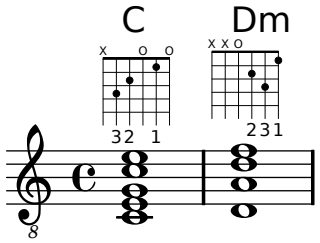
```
<<
```

```
  \context ChordNames {
     \chordmode {
       f1 g
     }
  }
  \context Staff {
    \clef "treble_8"
    < f, c f a c' f'>1 ^\markup
      \fret-diagram-terse #"1-(;3;3;2;1;1-);"
    < g, d g b d' g'> ^\markup
      \fret-diagram-terse #"3-(;5;5;4;3;3-);"
  }
>>
```



Les indications de doigtés peuvent être inclus dans l'interface `fret-diagram-terse`.

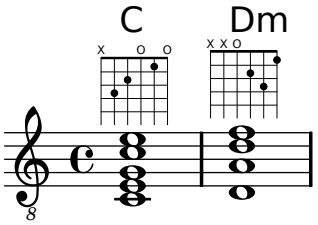
```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \override Voice.TextScript
      #'(fret-diagram-details finger-code) = #'below-string
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram-terse #"x;3-3;2-2;o;1-1;o;"
    < d a d' f'> ^\markup
      \fret-diagram-terse #"x;x;o;2-2;3-3;1-1;"
  }
>>
```



Les autres propriétés des diagrammes d'accord doivent être ajustées en utilisant la commande `\override` dans l'interface `fret-diagram-terse`.

L'interface `fret-diagram-verbose` est au format d'une liste Scheme. Chaque élément de la liste décrit un objet devant être placé dans le diagramme d'accord.

```
<<  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1 ^\markup
      \fret-diagram-verbose #'(
        (mute 6)
        (place-fret 5 3)
        (place-fret 4 2)
        (open 3)
        (place-fret 2 1)
        (open 1)
      )
    < d a d' f'> ^\markup
      \fret-diagram-verbose #'(
        (mute 6)
        (mute 5)
        (open 4)
        (place-fret 3 2)
        (place-fret 2 3)
        (place-fret 1 1)
      )
  }
>>
```

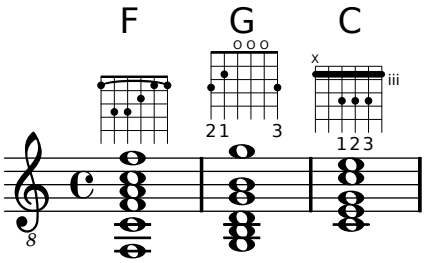


Les indications de doigté et de barré peuvent être décrits dans l'interface `fret-diagram-verbose`. Particularité propre à l'interface `fret-diagram-verbose` : l'indication de capodastre dans le diagramme d'accord. L'indication de capodastre est une petite ligne transversale aux cordes. La case avec le capodastre est la case la plus basse du diagramme d'accord.

```
<<
  \context ChordNames {
     \chordmode {
       f1 g c
     }
  }
  \context Staff {
    \clef "treble_8"
    \override Voice.TextScript
      #'(fret-diagram-details finger-code) = #'below-string

    < f, c f a c' f'>1 ^\markup
      \fret-diagram-verbose #'(
```

```
        (place-fret 6 1)
        (place-fret 5 3)
        (place-fret 4 3)
        (place-fret 3 2)
        (place-fret 2 1)
        (place-fret 1 1)
        (barre 6 1 1)
      )
    < g, b, d g b g'> ^\markup
      \fret-diagram-verbose #'(
        (place-fret 6 3 2)
        (place-fret 5 2 1)
        (open 4)
        (open 3)
        (open 2)
        (place-fret 1 3 3)
      )
    < c e g c' e'> ^\markup
      \fret-diagram-verbose #'(
        (capo 3)
        (mute 6)
        (place-fret 4 5 1)
        (place-fret 3 5 2)
        (place-fret 2 5 3)
      )
  }
>>
```



Toutes les autres propriétés du diagramme d'accord doivent être indiquées en utilisant la commande `\override` lorsque l'on utilise l'interface `fret-diagram-verbose`.

La disposition graphique d'un diagramme d'accord peut être modifiée suivant les préférences de l'utilisateur grâce aux propriétés de l'interface `fret-diagram-interface`. Des détails se trouvent dans Section "fret-diagram-interface" dans *Référence des propriétés internes*. Pour un diagramme d'accord, les propriétés de l'interface dépendent de `Voice.TextScript`.

Morceaux choisis

Personnalisation des diagrammes de fret

Les propriétés d'un diagramme de fret sont modifiables grâce au `'fret-diagram-details`. Lorsqu'ils sont générés sous forme de `\markup`, rien n'empêche de les modifier en jouant sur les réglages de l'objet `Voice.TextScript` ou bien directement sur le « markup ».

```
<<
  \chords { c1 | c | c | d }
```

```
\new Voice = "mel" {
  \textLengthOn
  % Set global properties of fret diagram
  \override TextScript #'size = #'1.2
  \override TextScript
    #'(fret-diagram-details finger-code) = #'in-dot
  \override TextScript
    #'(fret-diagram-details dot-color) = #'white

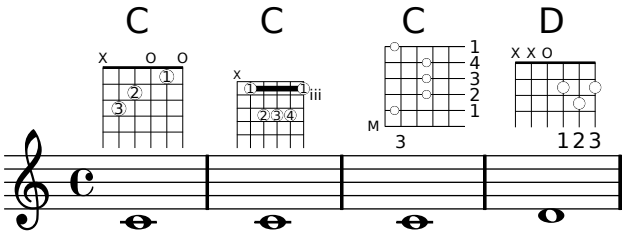
  %% C major for guitar, no barre, using defaults
     % terse style
  c'1^\markup { \fret-diagram-terse #"x;3-3;2-2;o;1-1;o;" }

  %% C major for guitar, barred on third fret
     % verbose style
     % size 1.0
     % roman fret label, finger labels below string, straight barre
  c'1^\markup {
    % standard size
    \override #'(size . 1.0) {
      \override #'(fret-diagram-details . (
                   (number-type . roman-lower)
                   (finger-code . in-dot)
                   (barre-type . straight))) {
        \fret-diagram-verbose #'((mute 6)
                                 (place-fret 5 3 1)
                                 (place-fret 4 5 2)
                                 (place-fret 3 5 3)
                                 (place-fret 2 5 4)
                                 (place-fret 1 3 1)
                                 (barre 5 1 3))
      }
    }
  }

  %% C major for guitar, barred on third fret
     % verbose style
     % landscape orientation, arabic numbers, M for mute string
     % no barre, fret label down or left, small mute label font
  c'1^\markup {
    \override #'(fret-diagram-details . (
                 (finger-code . below-string)
                 (number-type . arabic)
                 (label-dir . -1)
                 (mute-string . "M")
                 (orientation . landscape)
                 (barre-type . none)
                 (xo-font-magnification . 0.4)
                 (xo-padding . 0.3))) {
      \fret-diagram-verbose #'((mute 6)
                               (place-fret 5 3 1)
                               (place-fret 4 5 2)
```

```
                                    (place-fret 3 5 3)
                                    (place-fret 2 5 4)
                                    (place-fret 1 3 1)
                                    (barre 5 1 3))
      }
    }

    %% simple D chord
       % terse style
       % larger dots, centered dots, fewer frets
       % label below string
    d'1^\markup {
      \override #'(fret-diagram-details . (
                  (finger-code . below-string)
                  (dot-radius . 0.35)
                  (dot-position . 0.5)
                  (fret-count . 3))) {
        \fret-diagram-terse #"x;x;o;2-1;3-2;2-3;"
      }
    }
  }
>>
```



Voir aussi

Manuel de notation : Section A.8 [Text markup commands], page 510.

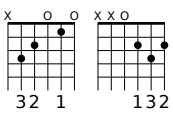
Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*.

Référence des propriétés internes : Section “fret-diagram-interface” dans *Référence des propriétés internes*.

Tablatures prédéfinies

Les diagrammes d’accords peuvent être affichés en utilisant le contexte `FretBoards`. Par défaut le contexte `FretBoards` affichera des diagrammes d’accords stockés dans une table de correspondance :

```
\include "predefined-guitar-fretboards.ly"
\context FretBoards {
  \chordmode {
    c1 d
  }
}
```



Les diagrammes d'accord définis par défaut sont dans le fichier '`predefined-guitar-fretboards.ly`'. Les diagrammes d'accord sont stockés en fonction des notes de l'accord ainsi que de l'accordage (`stringTunings`) utilisé. '`predefined-guitar-fretboards.ly`' contient les diagrammes d'accord prédéfinis uniquement pour l'accordage standard (`guitar-tuning`). Des diagrammes d'accords peuvent être définis pour d'autres instruments ou d'autres accordages en suivant les exemples du fichier '`predefined-guitar-fretboards.ly`'.

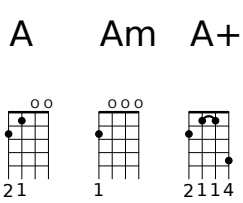
Les diagrammes de fret propres au ukulele se trouvent dans le fichier '`predefined-ukulele-fretboards.ly`'.

```
\include "predefined-ukulele-fretboards.ly"

myChords = \chordmode { a1 a:m a:aug }

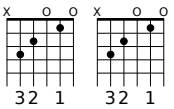
\new ChordNames {
  \myChords
}

\new FretBoards {
  \set stringTunings = #ukulele-tuning
  \myChords
}
```



Les notes des accords peuvent être entrées aussi bien comme musique simultanée qu'en utilisant le mode accord (voir [Généralités sur le mode accords], page 343).

```
\include "predefined-guitar-fretboards.ly"
\context FretBoards {
  \chordmode {c1}
  <c' e' g'>1
}
```

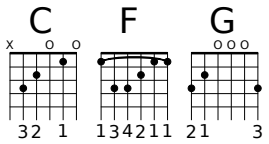


Il est courant d'afficher simultanément le nom des accords et les diagrammes d'accords correspondants. Ceci s'obtient en mettant en parallèle un contexte `ChordNames` avec un contexte `FretBoards` et en affectant aux deux la même musique.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}

<<
  \context ChordNames {
    \mychords
```

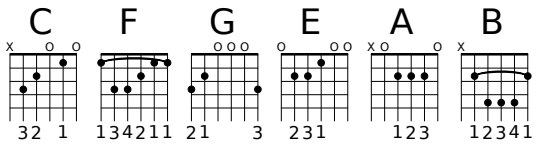
```
  }
  \context FretBoards {
    \mychords
  }
>>
```



Les diagrammes d'accords prédéfinis sont transposables tant qu'il y a un diagramme correspondant à l'accord transposé dans la base des diagrammes d'accords.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 f g
}

mychordlist = {
  \mychords
  \transpose c e { \mychords}
}
<<
  \context ChordNames {
    \mychordlist
  }
  \context FretBoards {
    \mychordlist
  }
>>
```

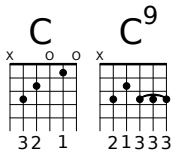


La table des diagrammes d'accord contient sept types d'accord (majeur, mineur, augmenté, diminué, septième de dominante, majeur sept, mineur sept) pour chacune des 17 fondamentales possibles. Une liste complète des diagrammes d'accords prédéfinis se trouve dans [Tablatures prédéfinies], page 312. S'il n'y a pas d'entrée dans la table pour un accord donné, le graveur `Fretboard_engraver` calculera le diagramme d'accord en utilisant la fonctionnalité automatique décrite dans [Tablatures automatiques], page 320.

```
\include "predefined-guitar-fretboards.ly"
mychords = \chordmode{
  c1 c:9
}

<<
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
```

```
    \mychords
  }
>>
```



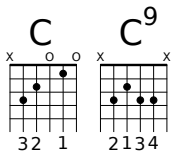
Des diagrammes d'accord peuvent être ajoutés à la table des diagrammes d'accord. Pour ajouter un diagramme d'accord, il faut spécifier l'accord correspondant au diagramme, l'accord utilisé et la définition du diagramme. Cette définition de diagramme peut être aussi bien de type *terse* que *verbose*.

```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table \chordmode {c:9}
                        #guitar-tuning
                        #"x;3-2;2-1;3-3;3-4;x;"

mychords = \chordmode{
  c1 c:9
}

<<
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
    \mychords
  }
>>
```



On peut enregistrer différents diagrammes pour un même accord en les définissant à des octaves différentes. Notez qu'il faut un intervalle de deux octaves, le premier servant à la transposition.

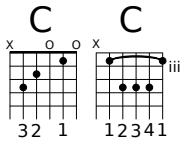
```
\include "predefined-guitar-fretboards.ly"

\storePredefinedDiagram #default-fret-table \chordmode { c'' }
                        #guitar-tuning
                        #(offset-fret 2 (chord-shape 'bes guitar-tuning))

mychords = \chordmode{
  c1 c''
}

<<
```

```
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
    \mychords
  }
>>
```



En plus des diagrammes d'accord, LilyPond possède une liste interne de carrures d'accord. Les carrures d'accords sont des diagrammes d'accord qui peuvent être transposés le long du manche. Les carrures d'accords peuvent être ajoutées à la liste interne et être ensuite utilisées pour définir des accords prédéfinis. Puisqu'elles sont transposables le long du manche, les carrures d'accord ne contiennent généralement pas de corde à vide. Tout comme les diagrammes d'accord, les carrures d'accord sont définies grâce aux interfaces `fret-diagram-terse` ou `fret-diagram-verbose`.

```
\include "predefined-guitar-fretboards.ly"

% add a new chord shape

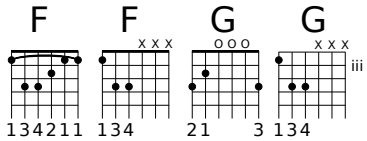
\addChordShape #'powerf #guitar-tuning #"1-1;3-3;3-4;x;x;x;"

% add some new chords based on the power chord shape

\storePredefinedDiagram #default-fret-table \chordmode {f'}
                        #guitar-tuning
                        #(chord-shape 'powerf guitar-tuning)
\storePredefinedDiagram #default-fret-table \chordmode {g'}
                        #guitar-tuning
                        #(offset-fret 2 (chord-shape 'powerf guitar-tuning))

mychords = \chordmode{
  f1 f' g g'
}

<<
  \context ChordNames {
    \mychords
  }
  \context FretBoards {
    \mychords
  }
>>
```



La disposition graphique d'un diagramme d'accord peut être modifiée suivant les préférences de l'utilisateur grâce aux propriétés de l'interface `fret-diagram-interface`. Pour plus d'information, consultez Section "fret-diagram-interface" dans *Référence des propriétés internes*. Pour un diagramme d'accord donné, les propriétés de l'interface dépendent de `FretBoards.FretBoard`.

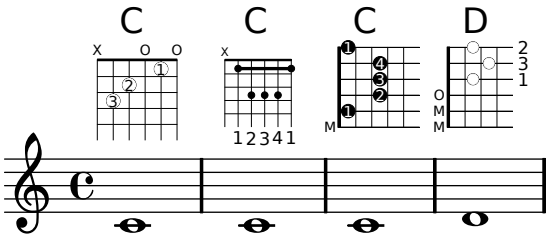
Morceaux choisis

Customizing fretboard fret diagrams

Fret diagram properties can be set through `'fret-diagram-details`. For FretBoard fret diagrams, overrides are applied to the `FretBoards.FretBoard` object. Like `Voice`, `FretBoards` is a bottom level context, therefore can be omitted in property overrides.

```
\include "predefined-guitar-fretboards.ly"
\storePredefinedDiagram #default-fret-table \chordmode { c' }
                        #guitar-tuning
                        #"x;1-1-(;3-2;3-3;3-4;1-1-);"
<<
  \new ChordNames {
    \chordmode { c1 | c | c | d }
  }
  \new FretBoards {
    % Set global properties of fret diagram
    \override FretBoards.FretBoard #'size = #'1.2
    \override FretBoard
      #'(fret-diagram-details finger-code) = #'in-dot
    \override FretBoard
      #'(fret-diagram-details dot-color) = #'white
    \chordmode {
      c
      \once \override FretBoard #'size = #'1.0
      \once \override FretBoard
        #'(fret-diagram-details barre-type) = #'straight
      \once \override FretBoard
        #'(fret-diagram-details dot-color) = #'black
      \once \override FretBoard
        #'(fret-diagram-details finger-code) = #'below-string
      c'
      \once \override FretBoard
        #'(fret-diagram-details barre-type) = #'none
      \once \override FretBoard
        #'(fret-diagram-details number-type) = #'arabic
      \once \override FretBoard
        #'(fret-diagram-details orientation) = #'landscape
      \once \override FretBoard
        #'(fret-diagram-details mute-string) = #"M"
      \once \override FretBoard
        #'(fret-diagram-details label-dir) = #LEFT
      \once \override FretBoard
        #'(fret-diagram-details dot-color) = #'black
      c'
      \once \override FretBoard
        #'(fret-diagram-details finger-code) = #'below-string
```

```
      \once \override FretBoard
        #'(fret-diagram-details dot-radius) = #0.35
      \once \override FretBoard
        #'(fret-diagram-details dot-position) = #0.5
      \once \override FretBoard
        #'(fret-diagram-details fret-count) = #3
      d
    }
  }
  \new Voice {
    c'1 | c' | c' | d'
  }
>>
```



Création de diagrammes de fret prédéfinis pour d'autres instruments

La liste des diagrammes standards prédéfinis pour la guitare peut être augmentée d'autres définitions spécifiques à d'autres instruments. Voici comment définir un nouvel accordage ainsi que quelques diagrammes prédéfinis pour le « cuatro vénézuélien ».

Cet exemple illustre aussi la manière d'ajouter des doigtés aux accords ; ils serviront de référence pour la boucle d'accord et seront indiqués dans les diagrammes et le `TabStaff`, mais pas dans la musique.

Ces diagrammes ne peuvent pas être transposés, dans la mesure où ils contiennent des informations sur les cordes. Ceci est amené à évoluer.

```
% add FretBoards for the Cuatro
%   Note: This section could be put into a separate file
%      predefined-cuatro-fretboards.ly
%      and \included into each of your compositions

cuatroTuning = #`(,(ly:make-pitch 0 6 0)
                  ,(ly:make-pitch 1 3 SHARP)
                  ,(ly:make-pitch 1 1 0)
                  ,(ly:make-pitch 0 5 0))

dSix = { <a\4 b\1 d\3 fis\2> }
dMajor = { <a\4 d\1 d\3 fis \2> }
aMajSeven = { <a\4 cis\1 e\3 g\2> }
dMajSeven = { <a\4 c\1 d\3 fis\2> }
gMajor = { <b\4 b\1 d\3 g\2> }

\storePredefinedDiagram #default-fret-table \dSix
                        #cuatroTuning
                        #"o;o;o;o;"
\storePredefinedDiagram #default-fret-table \dMajor
                        #cuatroTuning
```

```
                          #"o;o;o;3-3;"
\storePredefinedDiagram #default-fret-table \aMajSeven
                          #cuatroTuning
                          #"o;2-2;1-1;2-3;"
\storePredefinedDiagram #default-fret-table \dMajSeven
                          #cuatroTuning
                          #"o;o;o;1-1;"
\storePredefinedDiagram #default-fret-table \gMajor
                          #cuatroTuning
                          #"2-2;o;1-1;o;"

% end of potential include file /predefined-cuatro-fretboards.ly


#(set-global-staff-size 16)

primerosNames = \chordmode {
  d:6 d a:maj7 d:maj7
  g
}
primeros = {
  \dSix \dMajor \aMajSeven \dMajSeven
  \gMajor
}

\score {
  <<
    \new ChordNames {
      \set chordChanges = ##t
      \primerosNames
    }

    \new Staff {
      \new Voice \with {
        \remove "New_fingering_engraver"
      }
      \relative c'' {
        \primeros
      }
    }

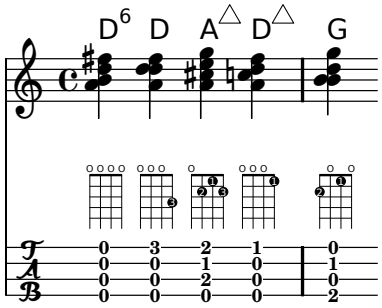
    \new FretBoards {
      \set stringTunings = #cuatroTuning
%      \override FretBoard
%        #'(fret-diagram-details string-count) = #'4
      \override FretBoard
        #'(fret-diagram-details finger-code) = #'in-dot
      \primeros
    }

    \new TabStaff \relative c'' {
      \set TabStaff.stringTunings = #cuatroTuning
```

```
      \primeros
    }

  >>

  \layout {
    \context {
      \Score
      \override SpacingSpanner
        #'base-shortest-duration = #(ly:make-moment 1 16)
    }
  }
  \midi { }
}
```



Voir aussi

Manuel de notation : [Tablatures personnalisées], page 302, [Tablatures automatiques], page 320, [Généralités sur le mode accords], page 343, [Tablatures prédéfinies], page 312.

Fichiers d'initialisation : '`ly/predefined-guitar-fretboards.ly`', '`ly/predefined-guitar-ninth-fretboards.ly`'.

Morceaux choisis : Section "Cordes frettées" dans *Morceaux choisis*.

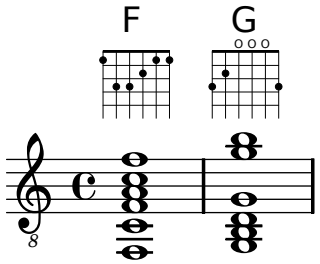
Référence des propriétés internes : Section "fret-diagram-interface" dans *Référence des propriétés internes*.

Tablatures automatiques

Les diagrammes d'accord peuvent être créés automatiquement ; il suffit d'affecter les notes à un contexte `FretBoards`. Si aucun diagramme prédéfini n'est disponible pour les notes entrées avec l'accordage actuel (`stringTunings`), les cordes et cases correspondant aux notes seront automatiquement calculées.

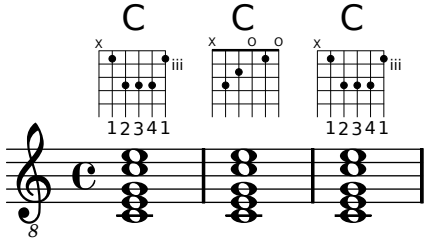
```
<<
  \context ChordNames {
     \chordmode {
       f1 g
     }
  }
  \context FretBoards {
    < f, c f a c' f'>1
    < g,\6 b, d g b g'>
  }
  \context Staff {
```

```
    \clef "treble_8"
    < f, c f a c' f'>1
    < g, b, d g b' g'>
  }
>>
```



Dans la mesure où aucun diagramme prédéfini n'est chargé par défaut, le calcul automatique des diagrammes d'accord est le comportement par défaut. Dès que les diagrammes par défaut sont chargés, le calcul automatique peut être activé ou désactivé par des commandes prédéfinies :

```
\storePredefinedDiagram #default-fret-table <c e g c' e'>
                        #guitar-tuning
                        #"x;3-1-(;5-2;5-3;5-4;3-1-1);"
<<
  \context ChordNames {
     \chordmode {
       c1 c c
     }
  }
  \context FretBoards {
    <c e g c' e'>1
    \predefinedFretboardsOff
    <c e g c' e'>
    \predefinedFretboardsOn
    <c e g c' e'>
  }
  \context Staff {
    \clef "treble_8"
    <c e g c' e'>1
    <c e g c' e'>
    <c e g c' e'>
  }
>>
```

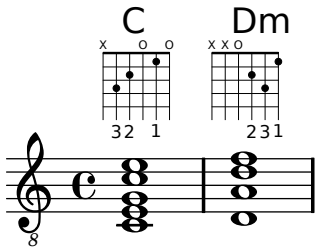


Parfois le calculateur se trouvera incapable de trouver un diagramme d'accord convenable. On peut souvent y remédier en assignant les notes aux cordes. Dans bien des cas, il suffit de placer

manuellement une seule note pour que les autres soient alors placées de maniére appropriée par le contexte `FretBoards`.

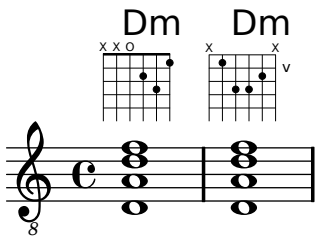
Il est possible d'ajouter des indications de doigté aux diagrammes de fret.

```
<<
  \context ChordNames {
     \chordmode {
       c1 d:m
     }
  }
  \context FretBoards {
    < c-3 e-2 g c'-1 e' > 1
    < d a-2 d'-3 f'-1>
  }
  \context Staff {
    \clef "treble_8"
    < c e g c' e' > 1
    < d a d' f'>
  }
>>
```



La propriété `minimumFret` permet de définir la case minimale qui servira à calculer les cordes et les cases du contexte `FretBoard`.

```
<<
  \context ChordNames {
     \chordmode {
       d1:m d:m
     }
  }
  \context FretBoards {
    < d a d' f'>
    \set FretBoards.minimumFret = #5
    < d a d' f'>
  }
  \context Staff {
    \clef "treble_8"
    < d a d' f'>
    < d a d' f'>
  }
>>
```



Les cordes et les cases du contexte `FretBoards` sont liées à la propriété `stringTunings`, qui a là même signification que dans le contexte `TabStaff`. Voir [Tablatures personnalisées], page 302 pour plus d'information sur la propriété `stringTunings`.

La disposition graphique d'un diagramme d'accord peut être modifiée suivant les préférences de l'utilisateur au travers des propriétés de l'interface `fret-diagram-interface`. Pour un diagramme d'accord, les propriétés de l'interface dépendent de `FretBoards.FretBoard`.

Commandes prédéfinies

`\predefinedFretboardsOff`, `\predefinedFretboardsOn`.

Voir aussi

Manuel de notation : [Tablatures personnalisées], page 302.

Morceaux choisis : Section “Cordes frettées” dans *Morceaux choisis*.

Référence des propriétés internes : Section “fret-diagram-interface” dans *Référence des propriétés internes*.

Problèmes connus et avertissements

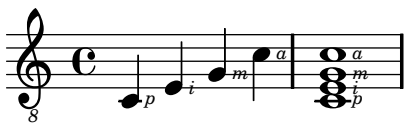
Le calcul automatique des diagrammes de fret se révèle inapproprié pour les instruments dont l'ordre des cordes ne correspond pas à l'ordre des hauteurs.

Doigtés pour la main droite

Les doigtés de main droite *p-i-m-a* doivent être entrés dans une construction de type accord `<>` même s'il n'y a qu'une seule note.

> **Note :** Un trait d'union **doit** suivre la note et une espace **doit** précéder la fermeture de la construction par un `>`.

```
\clef "treble_8"
<c-\rightHandFinger #1 >4
<e-\rightHandFinger #2 >
<g-\rightHandFinger #3 >
<c-\rightHandFinger #4 >
<c,-\rightHandFinger #1 e-\rightHandFinger #2
  g-\rightHandFinger #3 c-\rightHandFinger #4 >1
```



Pour plus de clarté, vous pouvez traduire ou abréger la commande `\rightHandFinger`, par exemple en `\doigtMainDroite` ou même `\MD` :

```
#(define MD rightHandFinger)
```

Morceaux choisis

Positionnement des doigtés main droite

Vous disposez d'une propriété spécifique qui permet de contrôler plus finement le positionnement des doigtés main droite, comme l'indique l'exemple suivant.

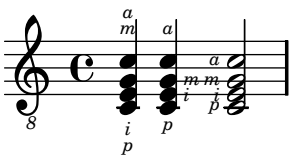
```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"

  \set strokeFingerOrientations = #'(up down)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >4

  \set strokeFingerOrientations = #'(up right down)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >4

  \set strokeFingerOrientations = #'(left)
  <c-\RH #1 e-\RH #2 g-\RH #3 c-\RH #4 >2
}
```

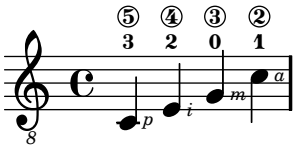


Doigtés indications de cordeet doigtés main droite

L'exemple suivant illustre comment combiner des doigtés pour la main gauche, des indications de corrde et des doigtés pour la main droite.

```
#(define RH rightHandFinger)

\relative c {
  \clef "treble_8"
  <c-3\5-\RH #1 >4
  <e-2\4-\RH #2 >4
  <g-0\3-\RH #3 >4
  <c-1\2-\RH #4 >4
}
```



Voir aussi

Morceaux choisis : Section "Cordes frettées" dans *Morceaux choisis*.

Référence des propriétés internes : Section "StrokeFinger" dans *Référence des propriétés internes*.
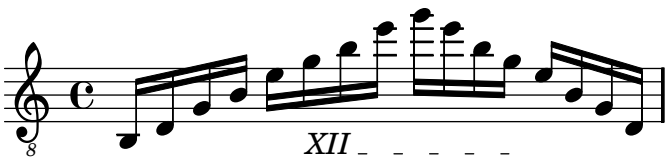
2.4.2 Guitare

La plupart des aspects en matière de notation pour guitare sont traités dans la partie commune aux instruments frettés. Il subsiste cependant quelques particularités que nous allons maintenant examiner. Parfois l'utilisateur aimerait créer des documents de type recueil de chansons, où l'on ne trouve que des accords au-dessus des paroles. Dans la mesure où LilyPond est un éditeur de partitions, il n'est pas l'outil optimal pour des documents sans partition. Une meilleure alternative serait de recourrir à un traitement de texte, un éditeur de texte ou, pour les utilisateurs experimentés, un logiciel typographique tel que GuitarTex.

Indication de la position et du barré

Cet exemple montre comment indiquer les positions et les barrés :

```
\clef "treble_8"
b16 d g b e
\textSpannerDown
\override TextSpanner #'(bound-details left text) = #"XII "
  g16\startTextSpan
  b16 e g e b g\stopTextSpan
e16 b g d
```



Voir aussi

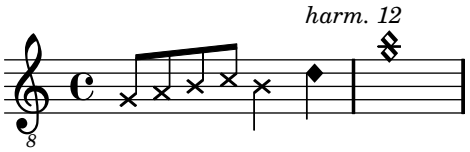
Manuel de notation : [Indication textuelle avec extension], page 195.

Morceaux choisis : Section "Cordes frettées" dans *Morceaux choisis*, Section "Signes d'interprétation" dans *Morceaux choisis*.

Indication des harmoniques et notes étouffées

Des têtes de note spéciales peuvent servir à indiquer les notes étouffées et les sons harmoniques. Les sons harmoniques sont souvent détaillés grâce à des indications textuelles.

```
\relative c' {
  \clef "treble_8"
  \override Staff.NoteHead #'style = #'cross
  g8 a b c b4
  \override Staff.NoteHead #'style = #'harmonic-mixed
  d^\markup { \italic { \fontsize #-2 { "harm. 12" }}} <g b>1
}
```



Voir aussi

Morceaux choisis : Section "Cordes frettées" dans *Morceaux choisis*.

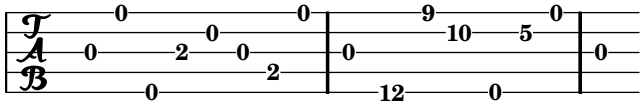
Manuel de notation : [Têtes de note spécifiques], page 30, Section 1.1.4 [Têtes de note], page 30.

2.4.3 Banjo

Tablatures pour banjo

LilyPond permet d'écrire des tablatures de base pour le banjo à cinq cordes. Pour ce faire, pensez à utiliser le format de tablature pour banjo, afin d'avoir le bon nombre de cordes et le bon accordage :

```
\new TabStaff <<
  \set TabStaff.tablatureFormat = #fret-number-tablature-format-banjo
  \set TabStaff.stringTunings = #banjo-open-g-tuning
  {
    \stemDown
    g8 d' g'\5 a b g e d' |
    g4 d''8\5 b' a'\2 g'\5 e'\2 d' |
    g4
  }
>>
```



LilyPond prend en charge un certain nombre d'accordages courants pour banjo : `banjo-c-tuning` sol,do,sol,si,ré (gCGBD), `banjo-modal-tuning` sol,ré,sol,do,ré (gDGCD), `banjo-open-d-tuning` la,ré,fa#,la,ré (aDF#AD) et `banjo-open-dm-tuning` la,ré,fa,la,ré (aDFAD).

Ces accordages peuvent être convertis pour banjo à quatre cordes au moyen de la fonction `four-string-banjo` :

```
\set TabStaff.stringTunings = #(four-string-banjo banjo-c-tuning)
```

Voir aussi

Morceaux choisis : Section "Cordes frettées" dans *Morceaux choisis*.

Les accordages prédéfinis du banjo sont contenus dans le fichier '`scm/output-lib.scm`'.

2.5 Percussions

2.5.1 Vue d'ensemble des percussions

La notation rythmique sert avant tout aux parties de percussions ou de batterie, mais on peut aussi s'en servir à des fins pédagogiques, pour montrer le rythme d'une mélodie.

Références en matière de notation pour percussions

Notation de base pour percussions

Les parties de percussions peuvent être saisies avec le mode `\drummode`, qui est l'équivalent du mode standard utilisé pour les notes à hauteur déterminée. Chaque instrument de percussion peut avoir, dans le fichier LilyPond, un nom complet et un nom raccourci.

```
\drums {
  hihat hh bassdrum bd
}
```



Ces noms sont inventoriés dans le fichier d'initialisation '`ly/drumpitch-init.ly`'.

Voir aussi

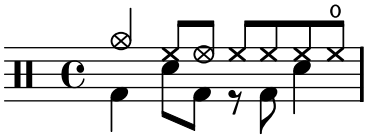
Référence du Programme : Section “note-event” dans *Référence des propriétés internes*.

Portées de percussion

Une partie de percussions utilisant plusieurs instruments requiert en général une portée de plusieurs lignes, où chaque hauteur sur la portée représente un instrument à percussion.

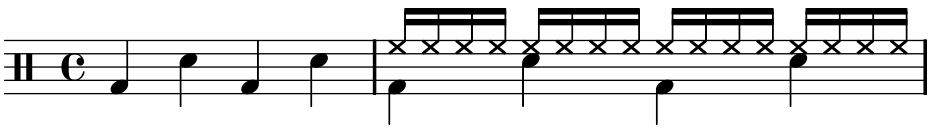
Pour saisir cette musique, il faut que les notes soient situées dans des contextes Section “DrumStaff” dans *Référence des propriétés internes* et Section “DrumVoice” dans *Référence des propriétés internes*.

```
up = \drummode { crashcymbal4 hihat8 halfopenhihat hh hh hh openhihat }
down = \drummode { bassdrum4 snare8 bd r bd sn4 }
  \new DrumStaff <<
    \new DrumVoice { \voiceOne \up }
    \new DrumVoice { \voiceTwo \down }
  >>
```



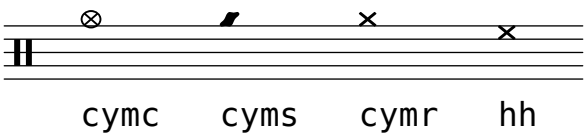
L’exemple ci-dessus montre une notation polyphonique détaillée. La notation polyphonique abrégée peut être employée lorsque le contexte Section “DrumVoice” dans *Référence des propriétés internes* est spécifié explicitement :

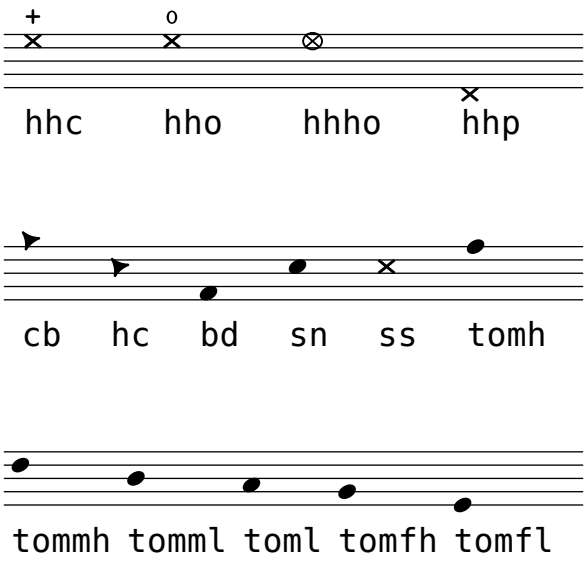
```
\new DrumStaff <<
  \new DrumVoice = "1" { s1 *2 }
  \new DrumVoice = "2" { s1 *2 }
  \drummode {
    bd4 sn4 bd4 sn4
    <<
      { \repeat unfold 16 hh16 }
      \\
      { bd4 sn4 bd4 sn4 }
    >>
  }
>>
```



On peut choisir d’autres mises en forme si l’on définit la propriété `drumStyleTable` dans le contexte Section “DrumVoice” dans *Référence des propriétés internes*. Quelques variables prédéfinies sont disponibles :

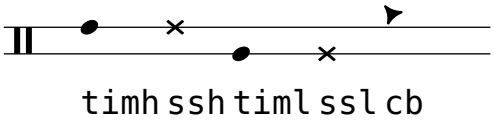
`drums-style`
: La notation par défaut : une batterie standard, sur une portée de cinq lignes.



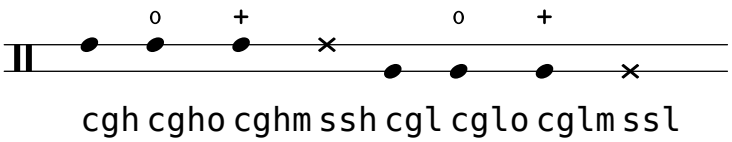


Le plan de la batterie peut inclure jusqu'à six toms différents. Bien sûr, vous n'êtes pas obligé de tous les utiliser si la musique en prévoit moins ; par exemple, les trois toms des lignes du milieu sont `tommh`, `tomml`, et `tomfh`.

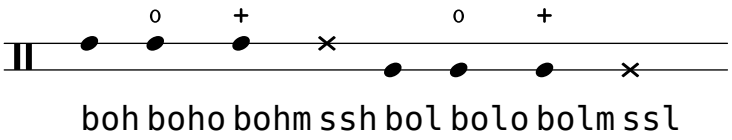
`timbales-style`
: Ce style permet de saisir une partie de timbales, sur une portée à deux lignes.



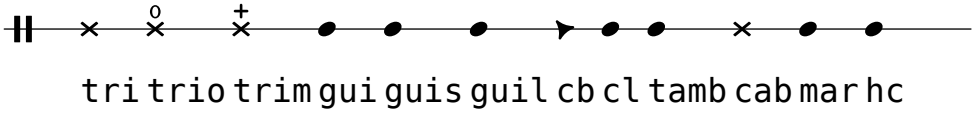
`congas-style`
: Ce style produit une portée à deux lignes pour une partie de congas.



`bongos-style`
: Ce style produit une portée à deux lignes pour une partie de bongos.



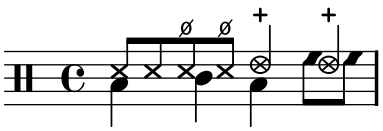
`percussion-style`
: Ce style permet de saisir toute sorte de percussions sur des portées d'une ligne.



Cependant, si aucun des styles prédéfinis ne vous convient, il est aisé de définir le vôtre en début de fichier.

```
#(define mydrums '(
         (bassdrum        default   #f           -1)
         (snare           default   #f           0)
         (hihat           cross     #f           1)
         (halfopenhihat   cross     "halfopen"   1)
         (pedalhihat      xcircle   "stopped"    2)
         (lowtom          diamond   #f           3)))
up = \drummode { hh8 hh hhho hhho hhp4 hhp }
down = \drummode { bd4 sn bd toml8 toml }

\new DrumStaff <<
  \set DrumStaff.drumStyleTable = #(alist->hash-table mydrums)
  \new DrumVoice { \voiceOne \up }
  \new DrumVoice { \voiceTwo \down }
>>
```



Voir aussi

Fichier d'initialisation : '`ly/drumpitch-init.ly`'.

Référence du programme : Section "DrumStaff" dans *Référence des propriétés internes*, Section "DrumVoice" dans *Référence des propriétés internes*.

Problèmes connus et avertissements

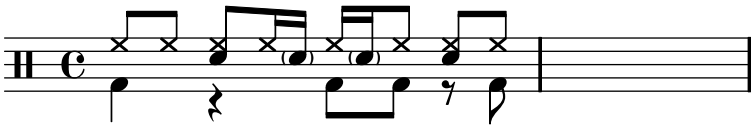
La bibliothèque MIDI générale ne contient pas les *rimshots* — coups sur le bord de la caisse claire — donc ils sont remplacés par des *sidesticks,* largement équivalents.

Notes fantômes

Des notes fantômes peuvent être créées pour les parties de percussion, grâce à la commande `\parenthesize` décrite dans [Parenthèses], page 188. Cependant, le mode `\drummode` n'inclut pas par défaut le graveur `Parenthesis_engraver` qui permet d'imprimer ces signes. Il faut donc l'ajouter explicitement dans la définition du contexte, suivant la manœuvre indiquée dans ⟨undefined⟩ [La commande set], page ⟨undefined⟩.

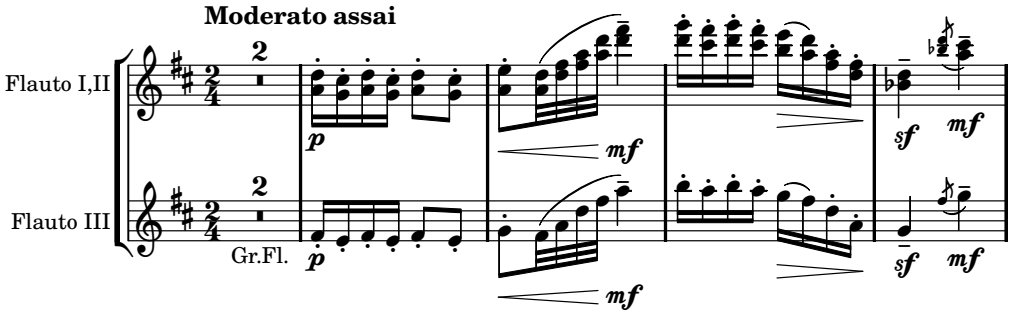
```
\new DrumStaff \with {
  \consists "Parenthesis_engraver"
} <<
  \context DrumVoice  = "1"  { s1 *2 }
  \context DrumVoice  = "2" { s1 *2 }
  \drummode {
    <<
      {
        hh8[ hh] <hh sn> hh16
        < \parenthesize sn > hh < \parenthesize
        sn > hh8 <hh sn> hh
      } \\ {
        bd4 r4 bd8 bd r8 bd
      }
    >>
```

```
  }
>>
```



Notez que les commandes `\parenthesize` obligent à ajouter des accords — sous la forme `< >` — autour de chaque élément.

2.6 Instruments à vent



Ce chapitre traite de certains aspects particuliers en matière de notation pour instruments à vent.

2.6.1 Vue d'ensemble des instruments à vent

Nous allons aborder ici quelques aspects communs à la plupart des instruments à vent.

Références en matière d'instruments à vent

Ce qui caractérise les partitions pour instruments à vent a trait principalement à la respiration et à l'attaque :

- Les respirations s'indiquent par des silences ou des [Signes de respiration], page 116.
- Un jeu *legato* s'indique par des [Liaisons d'articulation], page 111.
- Les différents types d'attaque – *legato*, détaché ou piqué – s'indiquent en principe par des signes d'articulation, parfois agrémentés de liaisons. Voir à ce sujet [Articulations et ornements], page 102 et Section A.10 [Liste des signes d'articulation], page 553.
- Un *Flatterzunge* (trémolo dental ou trémolo en roulant les r) s'indique par une marque de trémolo et une étiquette textuelle attachée à la note concernée. Voir à ce sujet [Répétitions en trémolo], page 135.

D'autres aspects de la notation s'appliquent aussi aux instruments à vent :

- De nombreux instruments à vent sont transpositeurs ; voir [Instruments transpositeurs], page 19.
- Les glissandos sont l'une des caractéristiques du trombone à coulisse, bien que d'autres instruments puisssent y parvenir en jouant sur les pistons ou des clés ; consulter [Glissando], page 118.
- Des glissandos harmoniques sont réalisables par les cuivres. Ils sont traditionnellement indiqués par des [Notes d'ornement], page 95.
- Les inflections en fin de note sont abordées au chapitre [Chutes et sauts], page 117.

- Les « bruitages » de clé ou de piston s'indiquent souvent par le style `cross` ou des [Têtes de note spécifiques], page 30.
- Les bois peuvent émettre des harmoniques dans le bas de leur registre. On les indique avec un `flageolet` – voir Section A.10 [Liste des signes d'articulation], page 553.
- En ce qui concerne les cuivres, la sourdine s'indique en pricipe par une étiquette textuelle. Cependant, lorsque les changements sont nombreux et rapides, il est d'usage de recourrir aux articulations `stopped` et `open`. Pour de plus amples détails, voir [Articulations et ornements], page 102 et Section A.10 [Liste des signes d'articulation], page 553.
- La sourdine du cor d'harmonie s'indique par un `stopped`. Voir le chapitre [Articulations et ornements], page 102.

Morceaux choisis
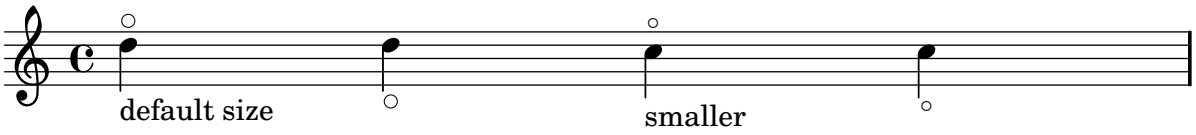
Modifier la taille d'un \flageolet

Il est possible de rapetisser le cercle d'un `\flageolet` grâce à une fonction Scheme.

```
smallFlageolet =
#(let ((m (make-articulation "flageolet")))
   (set! (ly:music-property m 'tweaks)
         (acons 'font-size -3
                (ly:music-property m 'tweaks)))
   m)

\layout { ragged-right = ##f }

\relative c'' {
  d4^\flageolet_\markup { default size } d_\flageolet
  c4^\smallFlageolet_\markup { smaller } c_\smallFlageolet
}
```



Voir aussi

Manuel de notation : [Signes de respiration], page 116, [Liaisons d'articulation], page 111, [Articulations et ornements], page 102, Section A.10 [Liste des signes d'articulation], page 553, [Répétitions en trémolo], page 135, [Instruments transpositeurs], page 19, [Glissando], page 118, [Notes d'ornement], page 95, [Chutes et sauts], page 117, [Têtes de note spécifiques], page 30,

Morceaux choisis : Section "Vents" dans *Morceaux choisis*.

Doigtés pour vents

Tout instrument à vent, hormis le trombone à coulisse, fait appel à plusieurs doigts pour produire un son. Les exemples ci-dessous vous donnent un apperçu de différentes manières d'indiquer des doigtés.

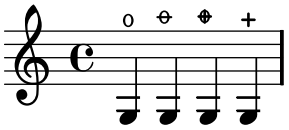
La gestion des diagrammes de doigté spécifiques aux bois est abordée plus en profondeur au chapitre Section 2.6.3.1 [Diagrammes pour bois], page 335.

Morceaux choisis

Symboles de doigtés pour instruments à vent

Des sumboles spécifiques peuvent être obtenus en combinant les glyphes disponibles, ce qui est tout à fait indiqué en matière d'instrument à vent.

```
centermarkup = {
  \once \override TextScript #'self-alignment-X = #CENTER
  \once \override TextScript #'X-offset =#(ly:make-simple-closure
    `(,+
      ,(ly:make-simple-closure (list
        ly:self-alignment-interface::centered-on-x-parent))
      ,(ly:make-simple-closure (list
        ly:self-alignment-interface::x-aligned-on-self))))
}
\score
{\relative c'
  {
    g\open
    \once \override TextScript #'staff-padding = #-1.0 \centermarkup
    g^\markup{\combine \musicglyph #"scripts.open" \musicglyph
    #"scripts.tenuto"}
    \centermarkup g^\markup{\combine \musicglyph #"scripts.open"
    \musicglyph #"scripts.stopped"}
    g\stopped
  }
}
```



Doigtés pour flûte à bec

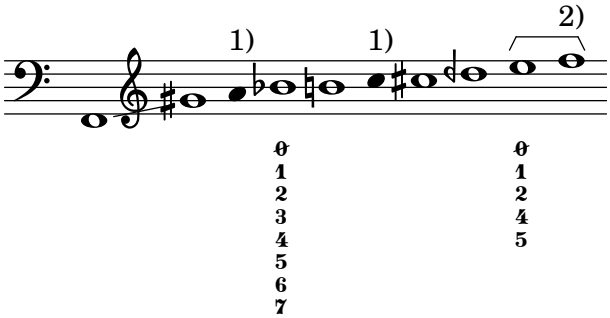
Cet exemple illustre la manière de créer et afficher des indications de doigté pour instrument à vent.

```
% range chart for paetzold contrabass recorder

centermarkup = {
  \once \override TextScript #'self-alignment-X = #CENTER
  \once \override TextScript #'X-offset =#(ly:make-simple-closure
  `(,+
  ,(ly:make-simple-closure (list
  ly:self-alignment-interface::centered-on-x-parent))
  ,(ly:make-simple-closure (list
  ly:self-alignment-interface::x-aligned-on-self))))
}

\score {
  \new Staff \with {
    \remove "Time_signature_engraver"
    \override Stem #'stencil = ##f
```

```
    \consists "Horizontal_bracket_engraver"
  }
  {
    \clef bass
    \set Score.timing = ##f
    f,1*1/4 \glissando
    \clef violin
    gis'1*1/4
    \stemDown a'4^\markup{1)}
    \centermarkup
    \once \override TextScript #'padding = #2
    bes'1*1/4_\markup{\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2 \finger 3 \finger 4
    \finger 5 \finger 6 \finger 7} }
    b'1*1/4
    c''4^\markup{1)}
    \centermarkup
    \once \override TextScript #'padding = #2
    cis''1*1/4
    deh''1*1/4
    \centermarkup
    \once \override TextScript #'padding = #2
    \once \override Staff.HorizontalBracket #'direction = #UP
    e''1*1/4_\markup{\override #'(baseline-skip . 1.7) \column
      { \fontsize #-5 \slashed-digit #0 \finger 1 \finger 2 \finger 4
    \finger 5} }\startGroup
    f''1*1/4^\markup{2)}\stopGroup
  }
}
```



Voir aussi

Manuel de notation : Section 2.6.3.1 [Diagrammes pour bois], page 335.

Snippets: Section “Vents” dans *Morceaux choisis*.

2.6.2 Cornemuse

Voici quelques informations spécifiques à la cornemuse.

Définitions pour la cornemuse

LilyPond inclut des définitions spécifiques destinées à la notation pour cornemuse écossaise ; pour les utiliser, il suffit d'ajouter
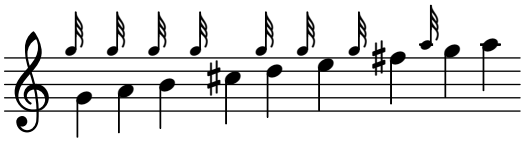
```
\include "bagpipe.ly"
```

en début de fichier. Ainsi, vous bénéficierez de commandes courtes pour les appogiatures spéciales et idiomatiques de la cornemuse. Par exemple, `\taor` est un raccourci pour

```
\grace { \small G32[ d G e] }
```

'`bagpipe.ly`' prend également en charge les définitions de hauteurs pour la cornemuse ; vous n'avez donc pas à vous soucier d'employer `\relative` ou `\transpose`.

```
\include "bagpipe.ly"
{ \grg G4 \grg a \grg b \grg c \grg d \grg e \grg f \grA g A }
```



La musique pour cornemuse est traditionnellement écrite en ré majeur. Dans la mesure où c'est la seule tonalité possible, l'usage est de ne pas indiquer l'armure explicitement. À cet effet, pensez à commencer votre partition par `\hideKeySignature` – ou bien `\showKeySignature` si, pour une raison ou pour une autre, vous tenez à afficher l'armure.

Des compositions actuelles peuvent ne pas respecter cette tonalité traditionnelle, auquel cas les do et fa devraient être abaissés en utilisant `cflat` ou `fflat` ; ils seront représentés par une note en forme de croix.

Lorsqu'on joue des œuvres moins cérémonieuses que pour une parade ou un défilé, peut survenir un sol aigu, au doux nom de « piobaireachd », et que l'on indiquera par `gflat`.

Voir aussi

Section "Vents" dans *Morceaux choisis*.

Exemple pour la cornemuse
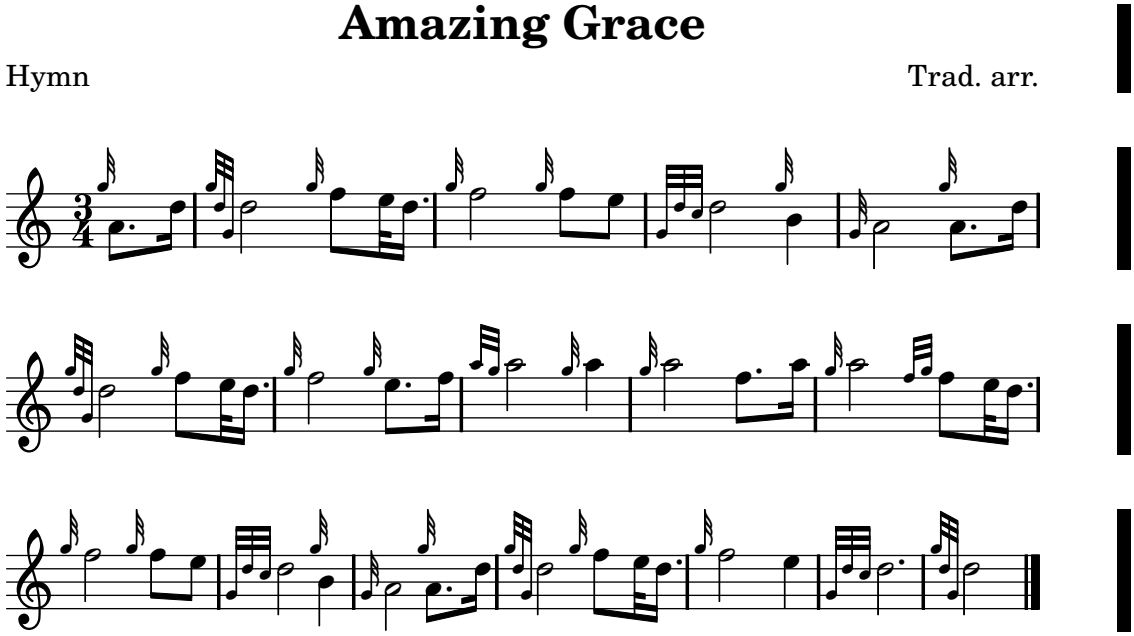
Et voici en guise d'exemple, à quoi ressemble le chant populaire *Amazing Grace*, noté dans l'idiome de la cornemuse.

```
\include "bagpipe.ly"
\layout {
  indent = 0.0\cm
  \context { \Score \remove "Bar_number_engraver" }
}

\header {
  title = "Amazing Grace"
  meter = "Hymn"
  arranger = "Trad. arr."
}

{
  \hideKeySignature
  \time 3/4
  \grg \partial 4 a8. d16
  \slurd d2 \grg f8[ e32 d16.]
```

```
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 \grg e8. f16
  \dblA A2 \grg A4
  \grg A2 f8. A16
  \grg A2 \hdblf f8[ e32 d16.]
  \grg f2 \grg f8 e
  \thrwd d2 \grg b4
  \grG a2 \grg a8. d16
  \slurd d2 \grg f8[ e32 d16.]
  \grg f2 e4
  \thrwd d2.
  \slurd d2
  \bar "|."
}
```



Voir aussi

Section “Vents” dans *Morceaux choisis*

2.6.3 Bois

Nous allons ici nous intéresser aux spécificités de la section des bois.

2.6.3.1 Diagrammes pour bois

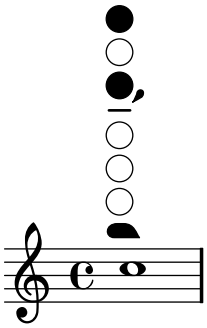
Les doigtés pour obtenir une note particulière peuvent s’afficher sous forme graphique. LilyPond dispose de diagrammes pour la plupart des bois, et tout particulièrement les instruments suivants :

- piccolo
- flûte
- hautbois

- clarinette
- clarinette basse
- saxophone
- basson
- contrebasson

Les diagrammes sont générés en tant qu'objet de type *markup* :

```
c1^\markup
  \woodwind-diagram #'piccolo #'((lh . (gis))
                                 (cc . (one three))
                                 (rh . (ees)))
```



Les clés ou trous peuvent être partiellement enfoncés ou bouchés :

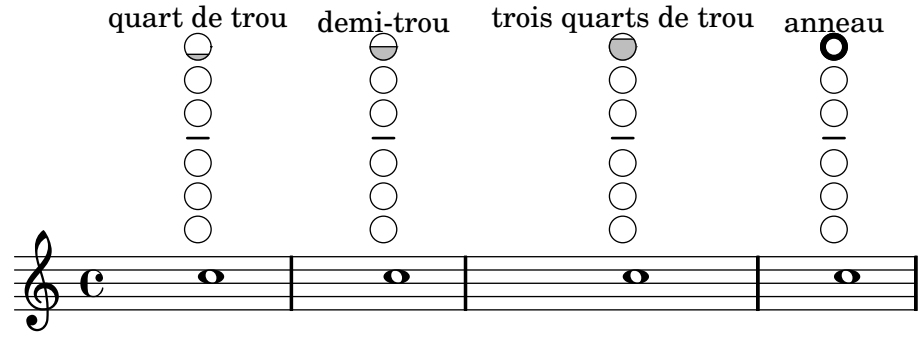
```
\textLengthOn
c1^\markup {
  \center-column {
   "quart de trou"
   \woodwind-diagram #'flute #'((cc . (one1q))
                                (lh . ())
                                (rh . ()))
  }
}

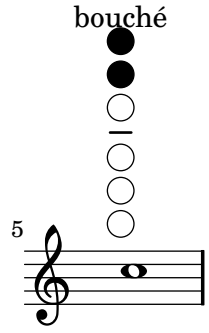
c1^\markup {
  \center-column {
   "demi-trou"
   \woodwind-diagram #'flute #'((cc . (one1h))
                                (lh . ())
                                (rh . ()))
  }
}

c1^\markup {
  \center-column {
   "trois quarts de trou"
   \woodwind-diagram #'flute #'((cc . (one3q))
                                (lh . ())
                                (rh . ()))
  }
}
```

```
c1^\markup {
  \center-column {
   "anneau"
   \woodwind-diagram #'flute #'((cc . (oneR))
                                (lh . ())
                                (rh . ()))
  }
}

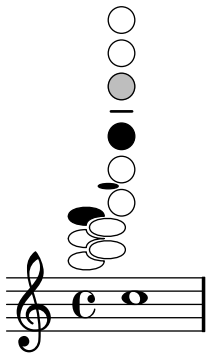
c1^\markup {
  \center-column {
   "bouché"
   \woodwind-diagram #'flute #'((cc . (oneF two))
                                (lh . ())
                                (rh . ()))
  }
}
```





L'indication du doigté permettant de triller s'obtient en grisant une position :

```
c1^\markup {
  \woodwind-diagram #'bass-clarinet
                    #'((cc . (threeT four))
                       (lh . ())
                       (rh . (b fis)))
}
```



Certaines combinaisons particulières en matière de trille sont possibles :

```
\textLengthOn
c1^\markup {
  \center-column {
   "quart de trou et anneau"
   \woodwind-diagram #'flute #'((cc . (one1qTR))
                                (lh . ())
                                (rh . ()))
  }
}

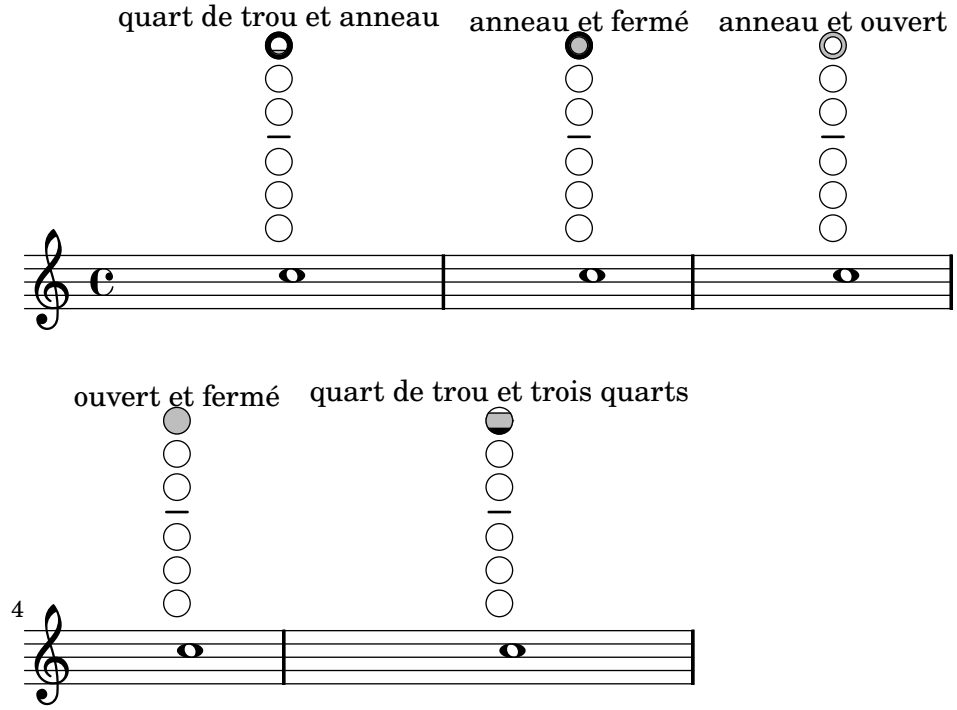
c1^\markup {
  \center-column {
   "anneau et fermé"
   \woodwind-diagram #'flute #'((cc . (oneTR))
                                (lh . ())
                                (rh . ()))
  }
}

c1^\markup {
  \center-column {
   "anneau et ouvert"
   \woodwind-diagram #'flute #'((cc . (oneRT))
                                (lh . ())
                                (rh . ()))
  }
}

c1^\markup {
  \center-column {
   "ouvert et fermé"
   \woodwind-diagram #'flute #'((cc . (oneT))
                                (lh . ())
                                (rh . ()))
  }
}

c1^\markup {
  \center-column {
   "quart de trou et trois quarts"
   \woodwind-diagram #'flute #'((cc . (one1qT3q))
                                (lh . ())
```

```
                                (rh . ()))
  }
}
```



Bien que cela ne produise pas de notation, vous pouvez obtenir à l'écran ou dans un fichier de journalisation la liste de toutes les possibilités pour un instrument donné, en utilisant dans un fichier

```
#(print-keys-verbose 'flute)
```

De nouveaux diagrammes sont réalisables, en suivant les directives contenues dans les fichiers '`scm/define-woodwind-diagrams.scm`' et '`scm/display-woodwind-diagrams.scm`'. Ceci requiert toutefois la maîtrise du langage Scheme.

Morceaux choisis

Liste des diagrammes de doigtés pour bois

Voici la liste des différents instruments à vent de la section des bois dont les doigtés sont disponibles à ce jour.

```
\relative c' {
  \textLengthOn
  c1^
  \markup {
    \center-column {
      'piccolo
      " "
       \woodwind-diagram
                  #'piccolo
                  #'()
    }
  }
```

```
c1^
\markup {
  \center-column {
     'flute
     " "
     \woodwind-diagram
        #'flute
        #'()
  }
}
c1^\markup {
  \center-column {
    'oboe
    " "
    \woodwind-diagram
      #'oboe
      #'()
  }
}

c1^\markup {
  \center-column {
    'clarinet
    " "
    \woodwind-diagram
      #'clarinet
      #'()
  }
}

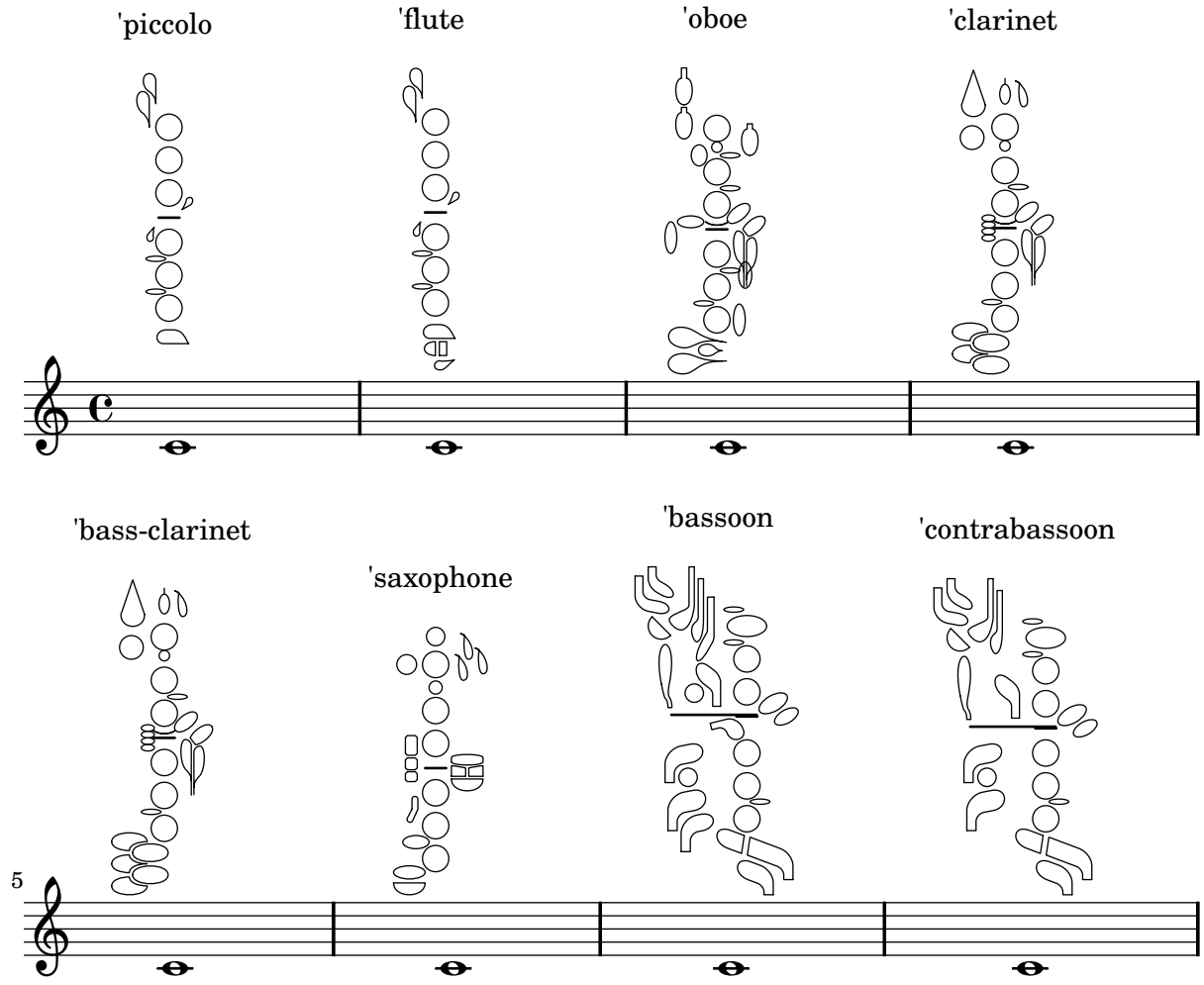
c1^\markup {
  \center-column {
    'bass-clarinet
    " "
    \woodwind-diagram
      #'bass-clarinet
      #'()
  }
}

c1^\markup {
  \center-column {
    'saxophone
    " "
    \woodwind-diagram
      #'saxophone
      #'()
  }
}

c1^\markup {
```

```
    \center-column {
      'bassoon
      " "
      \woodwind-diagram
        #'bassoon
        #'()
    }
  }

  c1^\markup {
    \center-column {
      'contrabassoon
      " "
      \woodwind-diagram
        #'contrabassoon
        #'()
    }
  }
}
```

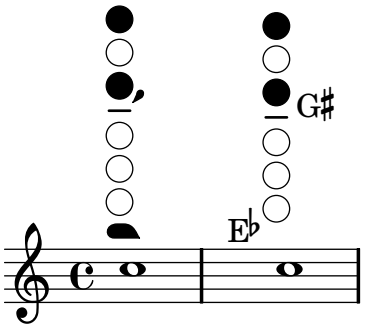


Ajout de texte à un diagramme de doigté

Dans certains cas, vous pouvez opter pour l'affichage textuel d'une clé située à côté d'un trou plutôt que sa représentation graphique.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram
      #'piccolo
      #'((cc . (one three))
         (lh . (gis))
         (rh . (ees)))

  c^\markup
    \override #'(graphical . #f) {
      \woodwind-diagram
        #'piccolo
        #'((cc . (one three))
           (lh . (gis))
           (rh . (ees)))
    }
}
```

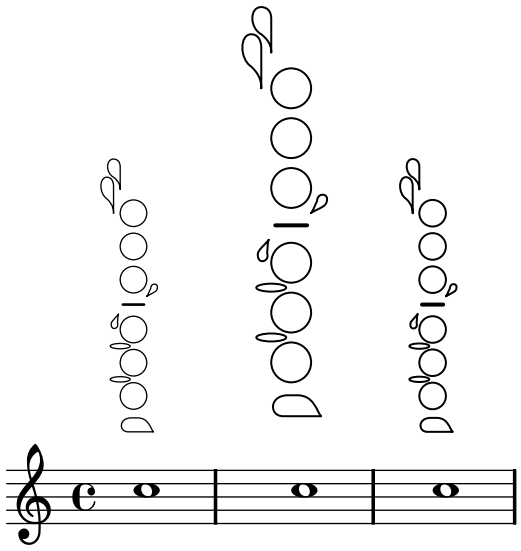


Modification de la taille d'un diagramme pour bois

La taille et l'épaisseur des diagrammes de doigté pour bois est modifiable à souhait.

```
\relative c'' {
  \textLengthOn
  c1^\markup
    \woodwind-diagram
      #'piccolo
      #'()

  c^\markup
    \override #'(size . 1.5) {
      \woodwind-diagram
        #'piccolo
        #'()
    }
  c^\markup
    \override #'(thickness . 0.15) {
      \woodwind-diagram
        #'piccolo
        #'()
    }
}
```



Liste des différents diagrammes de doigtés pour bois

Le code suivant vous permettra d'obtenir une liste de toutes les possibilités en matière de doigtés pour bois, tels qu'ils sont définis dans le fichier '`scm/define-woodwind-diagrams.scm`'. Cette liste sera produite en console et dans le fichier de journalisation, mais pas sous forme de musique.

```
#(print-keys-verbose 'piccolo)
#(print-keys-verbose 'flute)
#(print-keys-verbose 'flute-b-extension)
#(print-keys-verbose 'oboe)
#(print-keys-verbose 'clarinet)
#(print-keys-verbose 'bass-clarinet)
#(print-keys-verbose 'low-bass-clarinet)
#(print-keys-verbose 'saxophone)
#(print-keys-verbose 'baritone-saxophone)
#(print-keys-verbose 'bassoon)
#(print-keys-verbose 'contrabassoon)
```

Voir aussi

Fichiers d'initialisation : '`scm/define-woodwind-diagrams.scm`', '`scm/display-woodwind-diagrams.scm`'.

Morceaux choisis : Section "Vents" dans *Morceaux choisis*.

Référence des propriétés internes : Section "TextScript" dans *Référence des propriétés internes*, Section "instrument-specific-markup-interface" dans *Référence des propriétés internes*.

2.7 Notation des accords

2.7.1 Mode accords

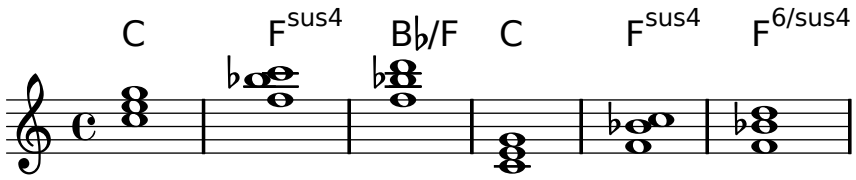
Généralités sur le mode accords

LilyPond permet de désigner les accords par leur chiffrage jazz. S'il est possible d'entrer un accord sous sa forme musicale, avec `<...>`, on peut aussi le saisir par son nom. Le logiciel traite les accords comme des ensembles de hauteurs, donc les accords peuvent être transposés.

```
twoWays = \transpose c c' {
  \chordmode {
    c1 f:sus4 bes/f
```

```
  }
  <c e g>
  <f bes c'>
  <f bes d'>
}

<< \new ChordNames \twoWays
   \new Voice \twoWays >>
```



Cet exemple montre également que les jeux d'instructions qui permettent à LilyPond d'imprimer des accords ne cherchent pas à se montrer intelligents. Ici, le dernier accord n'est pas interprété comme étant renversé.
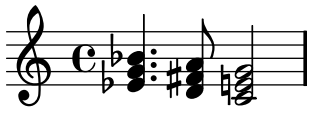
Notez bien que la valeur rythmique des accords doit être indiquée *à l'extérieur* des symboles `<>`.

```
<c e g>2
```

Accords courants

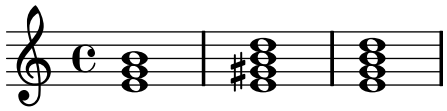
Dans le mode accords, introduit par la commande `\chordmode`, les accords ne sont indiqués que par leur note fondamentale.

```
\chordmode { es4. d8 c2 }
```



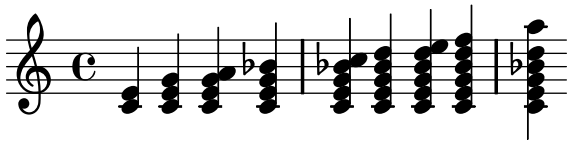
On peut cependant obtenir des accords différents, en utilisant le caractère deux points (`:`).

```
\chordmode { e1:m e1:7 e1:m7 }
```
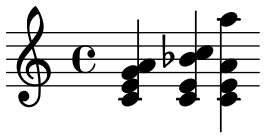


Le nombre accolé à la note fondamentale est considéré comme chiffrage jazz de l'accord, qui est de ce fait construit par un empilement de tierces. Notez l'exception que constitue `c:13` dans lequel la onzième est délibérément omise.

```
\chordmode { c:3 c:5 c:6 c:7 c:8 c:9 c:10 c:11 c:13 }
```



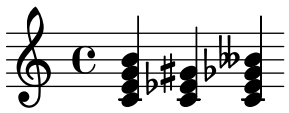
Des accords encore plus complexes peuvent être élaborés en plusieurs étapes séparées. Ainsi, on peut enrichir l'accord par des notes ajoutées, que l'on indique après le chiffrage principal et que l'on sépare par des points :

```
\chordmode { c:5.6 c:3.7.8 c:3.6.13 }
```
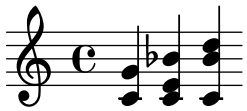


On peut augmenter ou diminuer certains intervalles au moyen des signes `-` ou `+` :

```
\chordmode { c:7+ c:5+.3- c:3-.5-.7- }
```



On peut aussi enlever certaines notes de l'accord, en les spécifiant après un signe `^` – les notes ajoutées doivent être indiquées *avant* les notes à enlever.

```
\chordmode { c^3 c:7^5 c:9^3.5 }
```



Voici les différents chiffrages disponibles, en plus des nombres que nous venons de voir :

`m` Accord mineur. Ce chiffrage minorise la tierce, et la septième s'il y en a une.

`dim` Accord diminué. Ce chiffrage minorise la tierce, diminue la quinte et la septième s'il y en a.
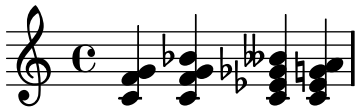
`aug` Accord augmenté. Ce chiffrage augmente la quinte.

`maj` Accord de septième majeure. Ce chiffrage majorise la septième s'il y en a une (dans le cas d'un accord parfait, ce chiffrage est facultatif).

`sus` Accord de suspension. Ce chiffrage supprime la tierce, et y ajoute, suivant que vous spécifiez `2` ou `4`, la seconde ou la quarte.
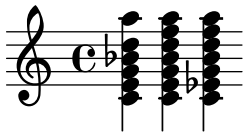
Il est bien sûr possible de mélanger ces chiffrages avec des notes ajoutées.

```
\chordmode { c:sus4 c:7sus4 c:dim7 c:m6 }
```



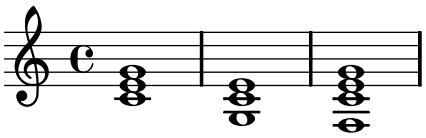
Dans la mesure où un accord de treizième majeure ne sonne pas très bien avec la onzième, la onzième est enlevée automatiquement, à moins que vous ne le spécifiez explicitement.

```
\chordmode { c:13 c:13.11 c:m13 }
```



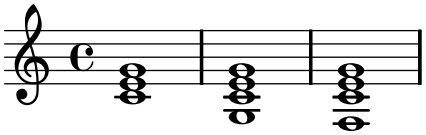
Les accords peuvent être renversés ou combinés avec une note étrangère à la basse, au moyen de *accord*`/`*note*

```
\chordmode { c1 c/g c/f }
```



Si la note de basse précisée appartient à l'accord, la doublure supérieure sera supprimée. Pour l'éviter, utilisez la syntaxe `/+`*note*.

```
\chordmode { c1 c/+g c/+f }
```



Le mode accords est semblable à `\lyricmode` ou autre, c'est-à-dire que la plupart des commandes sont encore disponibles ; ainsi, `r` ou `\skip` peuvent servir à insérer des silences ou des silences invisibles. De plus, les propriétés des objets peuvent être ajustées ou modifiées.

Problèmes connus et avertissements

Aucun nom de note ne peut être indiqué deux fois dans un accord. Ainsi, dans l'accord suivant, seule la quinte augmentée est prise en compte, car elle est indiquée en dernier :

```
\chordmode { c:5.5-.5+ }
```



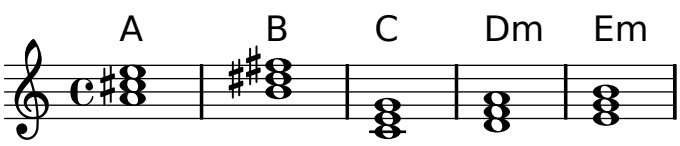
Extensions et altération d'accords

2.7.2 Gravure des accords

Impression des noms d'accord

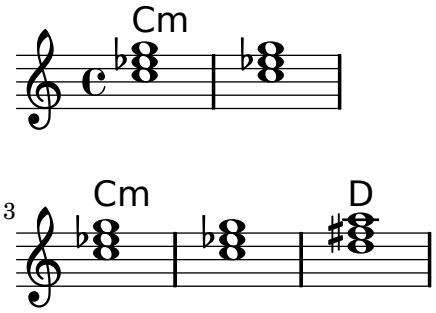
Les chiffrages d'accords sont liés au contexte Section "ChordNames" dans *Référence des propriétés internes*. Les accords peuvent être saisis soit au moyen de la notation indiquée ci-dessus, soit directement avec les symboles `<` et `>`.

```
harmonies = {
  \chordmode {a1 b c} <d' f' a'> <e' g' b'>
}
<<
  \new ChordNames \harmonies
  \new Staff \harmonies
>>
```



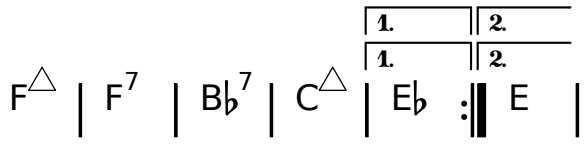
Vous pouvez faire ressortir les chiffrages d'accords en assignant la valeur vrai à la propriété `chordChanges` dans le contexte Section "ChordNames" dans *Référence des propriétés internes*. De cette façon, les chiffrages ne sont imprimés qu'aux changements d'accord ou en début de ligne.

```
harmonies = \chordmode {
  c1:m c:m \break c:m c:m d
}
<<
  \new ChordNames {
    \set chordChanges = ##t
    \harmonies }
  \new Staff \transpose c c' \harmonies
>>
```



Les exemples précédents montrent des chiffrages placés au-dessus de la portée, mais ce n'est pas obligatoire : les accords peuvent également être imprimés séparément – auquel cas vous aurez peut-être besoin d'ajouter le Section "Volta_engraver" dans *Référence des propriétés internes* et le Section "Bar_engraver" dans *Référence des propriétés internes* afin que les éventuelles barres de reprises s'affichent correctement.

```
\new ChordNames \with {
  \override BarLine #'bar-extent = #'(-2 . 2)
  \consists Bar_engraver
  \consists "Volta_engraver"
}
\chordmode { \repeat volta 2 {
  f1:maj7 f:7 bes:7
  c:maj7
} \alternative {
  es e
}
}
```



Le modèle par défaut des chiffrages d'accord est celui de Klaus Ignatzek pour le jazz (cf. Section "Références bibliographiques" dans *Essai*). Il s'agit d'une notation anglo-saxonne ; cependant vous pouvez indiquer vos chiffrages en notation française au moyen de la commande `\frenchChords` (voir plus bas).

Il est possible de créer votre propre modèle de chiffrages en réglant les propriétés suivantes :

`chordNameExceptions`

C'est la liste des accords mis en forme de manière particulière.

Cette liste d'exceptions s'indique de la façon suivante. On commence par créer une expression musicale telle que

```
chExceptionMusic = { <c f g bes>1 \markup { \super "7" "wahh" } }
```

puis on la transforme en liste d'exceptions au moyen du code

```
(sequential-music-to-chord-exceptions chExceptionMusic #t)
```

Pour qu'elles soient effectives, on ajoute ces exceptions aux exceptions par défaut définies dans '`ly/chord-modifier-init.ly`' :

```
(append
 (sequential-music-to-chord-exceptions chExceptionMusic #t)
 ignatzekExceptions)
```

Chord name exceptions

The property `chordNameExceptions` can be used to store a list of special notations for specific chords.

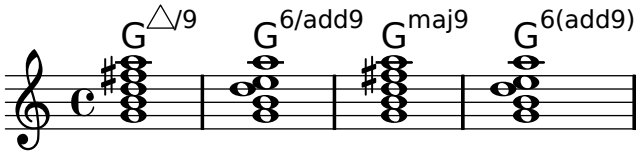
```
% modify maj9 and 6(add9)
% Exception music is chords with markups
chExceptionMusic = {
  <c e g b d'>1-\markup { \super "maj9" }
  <c e g a d'>1-\markup { \super "6(add9)" }
}

% Convert music to list and prepend to existing exceptions.
chExceptions = #( append
  ( sequential-music-to-chord-exceptions chExceptionMusic #t)
  ignatzekExceptions)

theMusic = \chordmode {
  g1:maj9 g1:6.9
  \set chordNameExceptions = #chExceptions
  g1:maj9 g1:6.9
}

\layout {
  ragged-right = ##t
}

<< \context ChordNames \theMusic
   \context Voice \theMusic
>>
```



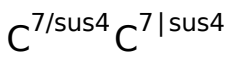
`majorSevenSymbol`

Cette propriété définit l'objet employé pour indiquer une septième majeure. Les options prédéfinies sont `whiteTriangleMarkup` (triangle blanc) et `blackTriangleMarkup` (triangle noir).

`chordNameSeparator`

Les différents termes d'un chiffrage jazz (par exemple les notes de l'accord et la basse) sont habituellement séparés par une barre oblique. La propriété `chordNameSeparator` permet d'indiquer un autre séparateur, par exemple

```
\new ChordNames \chordmode {
  c:7sus4
  \set chordNameSeparator
    = \markup { \typewriter "|" }
  c:7sus4
}
```

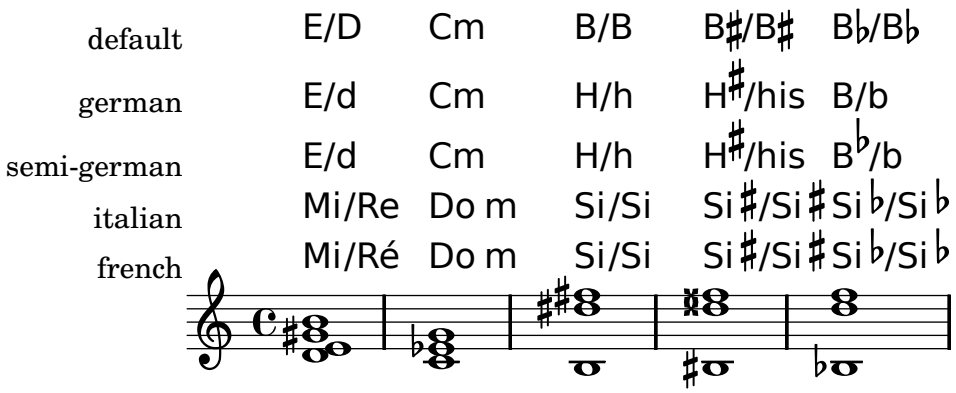


`chordRootNamer`
: Dans les chiffrages d'accord jazz, la note fondamentale de chaque accord est exprimée par une lettre capitale, parfois suivie d'une altération, correspondant à la notation anglo-saxonne de la musique. Cette propriété a pour valeur la fonction qui transforme la hauteur de la note foncdamentale en nom de note ; c'est donc en assignant une nouvelle fonction à cette propriété que l'on peut produire des noms de note spéciaux, adaptés par exemple aux systèmes de notation d'autres pays.

`chordNoteNamer`
: Lorsqu'un chiffrage mentionne une note ajoutée (par exemple la basse), les règles utilisées sont par défaut celles définies par la propriété `chordRootNamer` ci-dessus. Cependant, la propriété `chordNoteNamer` permet de régler cet élément indépendamment, par exemple pour imprimer la basse en caractères minuscules.

`chordPrefixSpacer`
: Le petit `m` qui indique un accord mineur est, dans la notation anglo-saxonne, attaché à la lettre de l'accord. Cependant il est possible d'ajouter un espace en assignant la valeur vrai à la propriété `chordPrefixSpacer`. Cet espace sera omis si une altération est présente.

Les propriétés ci-dessus font l'objet de commandes prédéfinies adaptées aux notations de différents pays : `\germanChords` et `\semiGermanChords` pour la notation allemande, `\italianChords` pour la notation italienne et enfin `\frenchChords` pour la notation française.



LilyPond intègre également deux autres modèles de chiffrages : une notation Jazz alternative, et un modèle systématique appelé système Banter. Pour la mise en œuvre de ces modèles, voir Section A.1 [Table des noms d'accord], page 477.

Commandes prédéfinies

`\germanChords`, `\semiGermanChords`, `\italianChords`, `\frenchChords`.

Voir aussi

Morceaux choisis : Section “Accords” dans *Morceaux choisis*.

Fichiers internes : ‘`scm/chords-ignatzek.scm`’, ‘`scm/chord-entry.scm`’.

Problèmes connus et avertissements

Les chiffrages d’accords ne sont déterminés que par la succession des hauteurs de notes. En d’autres termes, les accords inversés ne sont pas reconnus, ni les notes ajoutées à la basse. C’est pourquoi les accords saisis au moyen de la syntaxe `<...>` peuvent produire des chiffrages étranges.

Personnalisation des noms d’accord

2.7.3 Basse chiffrée

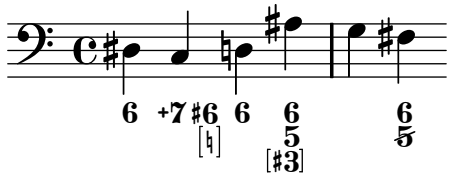
Introduction à la basse chiffrée

Les parties de basse continue étaient très répandues dans la musique baroque et jusqu’à la fin du XVIIIe siècle. Comme son nom l’indique, le *continuo* constitue une partie à lui seul, qui se déroule tout au long de l’œuvre pour en donner la structure harmonique.

Les musiciens du *continuo* jouent des claviers (clavecin, orgue) ou de tout autre instrument pouvant réaliser des accords. Leur partie est constituée d’une portée de basse aux notes agrémentées de combinaisons de chiffres et signes indiquant le développement des accords à jouer, ainsi que leur éventuel renversement. Cette notation était avant tout un guide, invitant le musicien à improviser de lui-même l’accompagnement.

LilyPond gère la basse chiffrée.

```
<<
  \new Voice { \clef bass dis4 c d ais g fis}
  \new FiguredBass \figuremode {
    < 6 >4 < 7\+ >8 < 6+ [_!] >
    < 6 >4 <6 5 [3+] >
    < _ >4 < 6 5/>4
  }
>>
```



La gestion de la basse chiffrée se décompose en deux parties. Dans un premier temps, le mode `\figuremode` permet de saisir les accords sous forme chiffrée. Le contexte Section “FiguredBass” dans *Référence des propriétés internes* s’occupera ensuite de gérer les objets Section “BassFigure” dans *Référence des propriétés internes*.

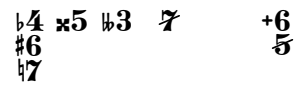
En mode de saisie, un chiffrage est délimité par `<` et `>`. La durée est indiquée après le `>` :

```
<4 6>
```



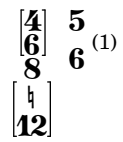
Les altérations s’obtiennent en ajoutant aux chiffres les caractères `-`, `!` ou `+`. Un signe plus s’obtient grâce à `\+`, et une quinte ou septième diminuée par `5/` ou `7/` respectivement.

```
<4- 6+ 7!> <5++> <3--> <7/> r <6\+ 5/>
```



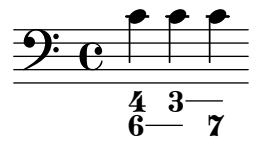
Le caractère _ insère un espace, et l'on peut imprimer des crochets avec [et]. Vous pouvez aussi ajouter des chaînes de caractères ou des étiquettes — cf. Section A.8 [Text markup commands], page 510.

```
<[4 6] 8 [_! 12] > <5 \markup { \number 6 \super (1) } >
```



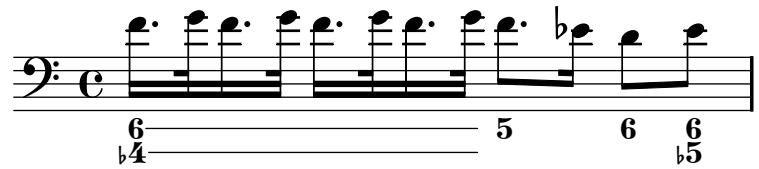
Lorsque des chiffrages se répètent, vous pouvez utiliser des lignes de prolongation.

```
<<
  \new Staff {
    \clef bass
    c4 c c
  }
  \figures {
    \set useBassFigureExtenders = ##t
    <4 6> <3 6> <3 7>
  }
>>
```



En pareil cas, la ligne de prolongation masquera toujours le chiffre qu'elle rappelle dans le chiffrage suivant.

Le contexte `FiguredBass` ne tient aucun compte de la ligne de basse. Par conséquent, il vous faudra peut être insérer des chiffrages supplémentaires pour imprimer les prolongations, ou utiliser des `\!` pour les éviter, comme dans l'exemple suivant :



Lorsque vous utilisez des lignes de prolongation, les chiffres communs seront verticalement alignés. Pour l'éviter, insérez un silence avec `r` afin de réinitialiser l'alignement. Par exemple, saisissez

```
<4 6>8 r8
```

au lieu de

```
<4 6>4
```

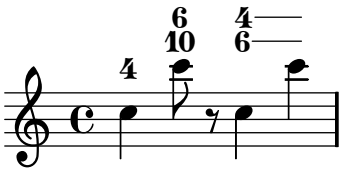
On peut choisir d'impriemr les altérations et signes plus aussi bien avant qu'après les chiffres, en réglant les propriétés `figuredBassAlterationDirection` et `figuredBassPlusDirection`.



Bien que la gestion de la basse chiffrée ressemble beaucoup à celle des accords, elle est beaucoup plus simpliste. Le mode `\figuremode` ne fait que stocker des chiffres que le contexte Section "FiguredBass" dans *Référence des propriétés internes* se chargera d'imprimer tels quels. En aucune manière ils ne sont transformés en son, et ils ne sont pas rendus dans un fichier MIDI.

En interne, ce code produit des étiquettes de texte que vous pouvez formater à votre convenance grâce aux propriétés des étiquettes. Par exemple, l'espacement vertical des chiffrages est déterminé par la propriété `baseline-skip`.

On peut également ajouter une basse chiffrée directement à un contexte `Staff`. L'alignement vertical est alors automatiquement ajusté.



Propriétés couramment modifiées

Par défaut, les chiffres sont imprimés au-dessus de la portée. Pour les imprimer dessous, ajoutez `\override Staff.BassFigureAlignmentPositioning #'direction = #DOWN`

Problèmes connus et avertissements

Si vous positionnez la basse chiffrée au dessus de la portée en ayant recours aux lignes d'extension et `implicitBassFigures`, les lignes peuvent se mélanger. Préserver l'ordre des prolongateurs peut s'avérer impossible lorsque plusieurs chiffrages qui se chevauchent en possèdent. Ce problème peut être contourné en jouant sur l'empilement, avec la propriété `stacking-dir` de l'objet `BassFigureAlignment`.

Voir aussi

Référence des pripriétés internes : les objets Section "BassFigure" dans *Référence des propriétés internes*, Section "BassFigureAlignment" dans *Référence des propriétés internes*, Section "BassFigureLine" dans *Référence des propriétés internes*, Section "BassFigureBracket" dans *Référence des propriétés internes* et Section "BassFigureContinuation" dans *Référence des propriétés internes*, ainsi que le contexte Section "FiguredBass" dans *Référence des propriétés internes*.

Saisie de la basse chiffrée

Gravure de la basse chiffrée

2.8 Contemporary music

From the beginning of the 20th Century there has been a massive expansion of compositional style and technique. New harmonic and rhythmic developments, an expansion of the pitch spectrum and the development of a wide range of new instrumental techniques have been accompanied by a parallel evolution and expansion of musical notation. The purpose of this section is to provide references and information relevant to working with these new notational techniques.

2.8.1 Pitch and harmony in contemporary music

This section highlights issues that are relevant to notating pitch and harmony in contemporary music.

References for pitch and harmony in contemporary music

- Standard quarter-tone notation is addressed in ⟨undefined⟩ [Note names in other languages], page ⟨undefined⟩.
- Non-standard key signatures are addressed in ⟨undefined⟩ [Key signature], page ⟨undefined⟩.
- Contemporary practises in displaying accidentals are addressed in ⟨undefined⟩ [Automatic accidentals], page ⟨undefined⟩.

Microtonal notation

Contemporary key signatures and harmony

2.8.2 Contemporary approaches to rhythm

This section highlights issues that are relevant to the notation of rhythm in contemporary music.

References for contemporary approaches to rhythm

- Compound time signatures are addressed in ⟨undefined⟩ [Time signature], page ⟨undefined⟩.
- Basic polymetric notation is addressed in ⟨undefined⟩ [Polymetric notation], page ⟨undefined⟩.
- Feathered beams are addressed in ⟨undefined⟩ [Feathered beams], page ⟨undefined⟩.
- Mensurstriche bar lines (bar lines between staves only) are addressed in ⟨undefined⟩ [Grouping staves], page ⟨undefined⟩.

Tuplets in contemporary music

Contemporary time signatures

Extended polymetric notation

Beams in contemporary music

Bar lines in contemporary music

2.8.3 Graphical notation

2.8.4 Contemporary scoring techniques

2.8.5 New instrumental techniques

2.8.6 Further reading and scores of interest

This section suggests books, musical examples and other resources useful in studying contemporary musical notation.

Books and articles on contemporary musical notation

- *Music Notation in the Twentieth Century: A Practical Guidebook* by Kurt Stone [W. W. Norton, 1980]
- *Music Notation: A Manual of Modern Practice* by Gardner Read [Taplinger, 1979]
- *Instrumentation and Orchestration* by Alfred Blatter [Schirmer, 2nd ed. 1997]

Scores and musical examples

2.9 Notations anciennes

2.9.1 Introduction aux notations anciennes

La gestion par LilyPond des formes de notation ancienne inclut des fonctionnalités spécifiques à la notation mensurale et au chant grégorien. La basse chiffrée est également partiellement prise en charge.

De nombreux objets graphiques — « grobs » dans le jargon de LilyPond — disposent d'une propriété `style`, comme nous le verrons dans

- [Têtes de note anciennes], page 355,
- [Altérations anciennes], page 355,
- [Silences anciens], page 356,
- [Clefs anciennes], page 357,
- [Crochets anciens], page 359,
- [Métriques anciennes], page 359.

Manipuler cette propriété permet d'adapter l'aspect typographique des grobs à une forme de notation particulière, ce qui évite la création de nouveaux concepts de notation.

En plus des signes d'articulation standards décrits à la section [Articulations et ornements], page 102, la notation ancienne dispose de signes particuliers.

- [Articulations anciennes], page 360

D'autres aspects de la notation ancienne ne peuvent pas être gérés aussi simplement qu'en jouant sur les propriétés d'un style appliqué à un objet graphique ou en lui ajoutant des articulations. Certains concepts sont spécifiques à la notation ancienne.

- [Guidons], page 361,
- [Divisions], page 362,
- [Ligatures], page 362.

Si tout cela vous dépasse et que vous désirez plonger dans le vif du sujet sans trop vous préoccuper d'ajuster des contextes, consultez les pages dédiées aux contextes prédéfinis. Ils vous permettront d'adapter vos contextes de voix et de portée, et vous n'aurez plus qu'à saisir les notes.

- [Contextes du chant grégorien], page 370,
- [Les contextes de la musique mensurale], page 370.

LilyPond gère partiellement la représentation de basses chiffrées, typiques de l'époque baroque, mais également employées de nos jours en harmonie et en analyse.

- Section 2.7.3 [Basse chiffrée], page 350

Voici les points que nous allons aborder :

Formes de notation ancienne prises en charge

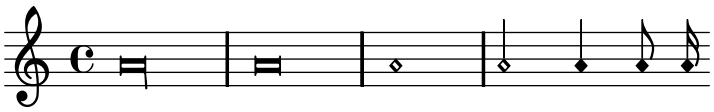
2.9.2 Signes de note alternatifs

Têtes de note anciennes

Pour de la musique ancienne, vous disposez de plusieurs styles de tête de note, en plus du style par défaut `default`. Vous pouvez affecter à la propriété `style` de l'objet Section "NoteHead" dans *Référence des propriétés internes* les valeurs `baroque`, `neomensural`, `mensural` ou `petrucci`. En style `baroque`, la seule différence par rapport au style `default` concerne la `\breve` qui sera carrée et non pas ovoïde. Le style `neomensural` ajoute au `baroque` le fait que les notes de durée inférieure ou égale à une ronde sont en forme de losange, et les hampes centrées sur la tête. Ce style est particulièrement adapté à la transcription de la musique mesurée dans les incipits. Le style `mensural` permet de reproduire les têtes de note telles qu'imprimées au XVIe siècle. Enfin, le style `petrucci` imite des partitions historiques, bien qu'il utilise de plus grosses têtes de note.

L'exemple suivant illustre le style `neomensural`.

```
\set Score.skipBars = ##t
\override NoteHead #'style = #'neomensural
a'\longa a'\breve a'1 a'2 a'4 a'8 a'16
```



Si vous écrivez en notation grégorienne, le Section "Vaticana_ligature_engraver" dans *Référence des propriétés internes* se chargera de sélectionner les têtes de note appropriées ; il est donc inutile de spécifier le style à utiliser. Vous pouvez cependant spécifier par exemple le style `vaticana_punctum` pour obtenir des neumes punctums. De même, c'est le Section "Mensural_ligature_engraver" dans *Référence des propriétés internes* qui se chargera des ligatures mensurales. Consultez la section [Ligatures], page 362 pour savoir comment fonctionnent les graveurs de ligature.

Voir aussi

Manuel de notation : Section 1.1.4 [Têtes de note], page 30.

Altérations anciennes

Pour utiliser les formes anciennes d'altération, utilisez la propriété `glyph-name-alist` des objets graphiques Section "Accidental" dans *Référence des propriétés internes* et Section "KeySignature" dans *Référence des propriétés internes*.

vaticana medicaea hufnagel mensural



Vous noterez que chacun de ces styles ne comporte pas toutes les altérations. LilyPond changera de style s'il y a besoin d'une altération indisponible dans le style utilisé.

À l'instar des altérations accidentelles, le style d'armure est géré par la propriété `glyph-name-alist` de l'objet Section "KeySignature" dans *Référence des propriétés internes*.

Voir aussi

Manuel de notation: Section 1.1 [Hauteurs], page 1, [Altérations], page 5, [Altérations accidentelles automatiques], page 20,

Référence des propriétés internes : Section "KeySignature" dans *Référence des propriétés internes*.
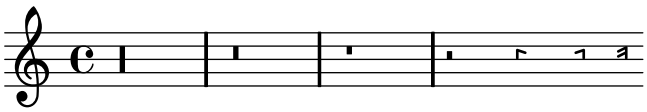
Morceaux choisis : Section "Notations anciennes" dans *Morceaux choisis*.

Silences anciens

La propriété `style` de l'objet Section "Rest" dans *Référence des propriétés internes* permet d'obtenir des silences de type ancien. Vous disposez des styles `classical`, `neomensural` et `mensural`. Le style `classical` ne se distingue du style `default` que par le soupir (demi-soupir en miroir). Le style `neomensural` convient tout à fait à l'incipit lors de la transcription de musique mensurale. Le style `mensural`, enfin, imite la gravure des silences dans certaines éditions du XVIe siècle.

L'exemple suivant illustre le style `neomensural`.

```
\set Score.skipBars = ##t
\override Rest #'style = #'neomensural
r\longa r\breve r1 r2 r4 r8 r16
```



Les styles `mensural` et `neomensural` ne disposent pas des 8e et 16e de soupir ; LilyPond utilise dans de tels cas le style par défaut. Voici une liste des styles de silences disponibles.

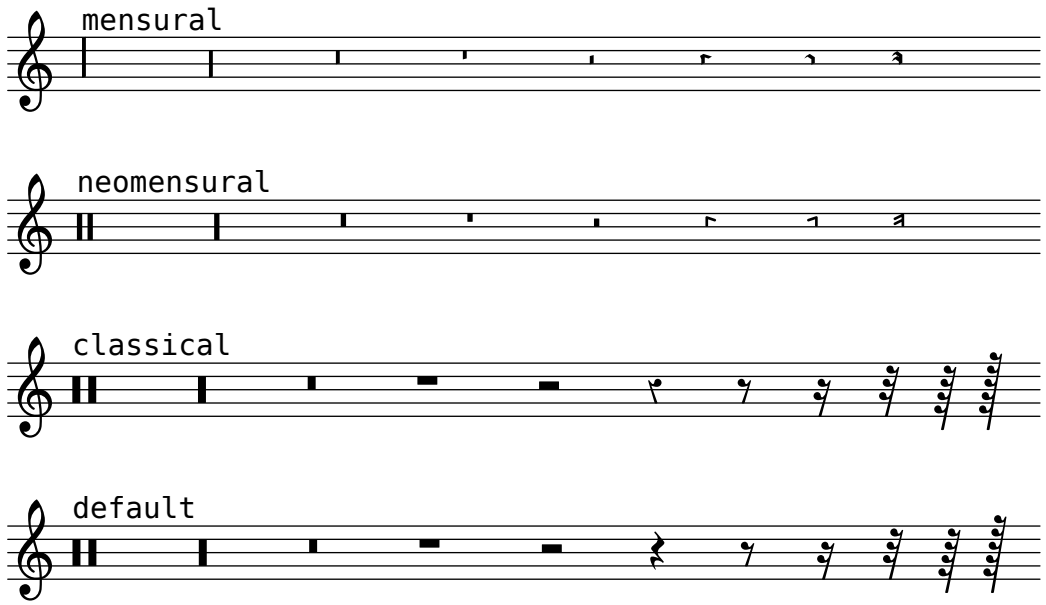
```
\layout {
  indent = 0
  \context {
    \Staff
    \remove "Time_signature_engraver"
  }
}

\new Staff \relative c {
  \cadenzaOn
  \override Staff.Rest #'style = #'mensural
  r\maxima^\markup \typewriter { mensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest #'style = #'neomensural
  r\maxima^\markup \typewriter { neomensural }
  r\longa r\breve r1 r2 r4 r8 r16 s32 s64 s128 s128
  \bar ""

  \override Staff.Rest #'style = #'classical
  r\maxima^\markup \typewriter { classical }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
  \bar ""

  \override Staff.Rest  #'style = #'default
  r\maxima^\markup \typewriter { default }
  r\longa r\breve r1 r2 r4 r8 r16 r32 r64 r128 s128
}
```



Les silences sont absents de la notation grégorienne ; par contre, cette notation utilise des [Divisions], page 362.
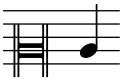
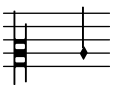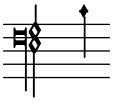
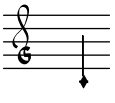
Voir aussi
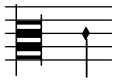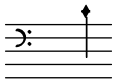
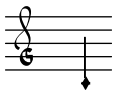
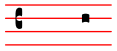
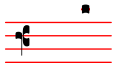
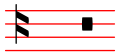
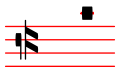
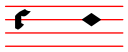
Manuel de notation : les principes généraux sur l'utilisation des silences sont exposés dans [Silences], page 47.

Clefs anciennes

Avec LilyPond, de nombreuses clés sont disponibles, dont certaines sont dédiées à la musique ancienne.

Le tableau suivant présente les différentes clés anciennes que vous pouvez sélectionner avec la commande `\clef`. Certaines de ces clés utilisent le même glyphe, attaché à l'une ou l'autre des lignes de la portée. Le chiffre porté en suffixe permet alors de les différencier. Vous pouvez néanmoins forcer le positionnement du glyphe sur une ligne, comme expliqué à la section [Clefs], page 13. Dans la colonne exemple, la note suivant la clé montre le do médium.

Description	**Clé disponible**	**Exemple**
Clé d'ut, style mensural moderne	`neomensural-c1`, `neomensural-c2`, `neomensural-c3`, `neomensural-c4`	
Clé d'ut, style mensural Petrucci, positionnable sur différentes lignes (clé d'ut seconde pour l'exemple)	`petrucci-c1`, `petrucci-c2`, `petrucci-c3`, `petrucci-c4`, `petrucci-c5`	
Clé de fa, style mensural Petrucci	`petrucci-f`	
Clé de sol, style mensural Petrucci	`petrucci-g`	

Clé d'ut, style mensural historique	`mensural-c1`, `mensural-c2`, `mensural-c3`, `mensural-c4`	
Clé de fa, style mensural historique	`mensural-f`	
Clé de sol, style mensural historique	`mensural-g`	
Clé d'ut, style Editio Vaticana	`vaticana-do1`, `vaticana-do2`, `vaticana-do3`	
Clé de fa, style Editio Vaticana	`vaticana-fa1`, `vaticana-fa2`	
Clé d'ut, style Editio Medicaea	`medicaea-do1`, `medicaea-do2`, `medicaea-do3`	
Clé de fa, style Editio Medicaea	`medicaea-fa1`, `medicaea-fa2`	
Clé d'ut, style historique Hufnagel	`hufnagel-do1`, `hufnagel-do2`, `hufnagel-do3`	
Clé de fa, style historique Hufnagel	`hufnagel-fa1`, `hufnagel-fa2`	
Clé combinée ut/fa, style historique Hufnagel	`hufnagel-do-fa`	

Moderne signifie « gravé comme dans les transcriptions contemporaines de musique mesurée. »

Petrucci signifie « inspiré des éditions réalisées par le maître graveur vénitien Petrucci (1466-1539). »

Historique signifie « gravé comme dans les éditions historiques, manuscrites ou non, autres que celles de Petrucci. »

Editio XXX signifie « gravé comme dans les ouvrages estampillés Editio XXX. »

Les clés d'ut de Petrucci avaient une hampe gauche différente selon leur ligne de rattachement.

Voir aussi

Manuel de notation : [Clefs], page 13.

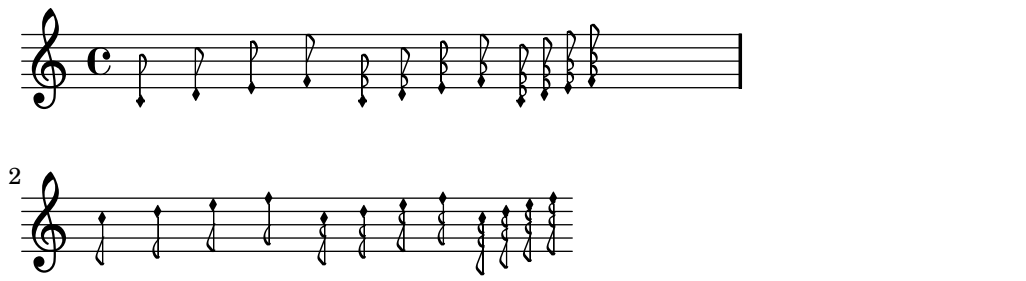
Problèmes connus et avertissements

La clé de sol mensurale est calquée sur celle de Petrucci.

Crochets anciens

Le réglage de la propriété `flag-style` de l'objet hampe (Section "Stem" dans *Référence des propriétés internes*) donne accès aux crochets de style ancien. Les seuls styles actuellement pris en charge sont `default` et `mensural`.

```
\override Stem #'flag-style = #'mensural
\override Stem #'thickness = #1.0
\override NoteHead #'style = #'mensural
\autoBeamOff
c'8 d'8 e'8 f'8 c'16 d'16 e'16 f'16 c'32 d'32 e'32 f'32 s8
c''8 d''8 e''8 f''8 c''16 d''16 e''16 f''16 c''32 d''32 e''32 f''32
```



Notez que pour chaque crochet mensural, l'extrémité la plus proche de la tête de note sera toujours attachée à une ligne de la portée.

Il n'existe pas de crochet spécifique au style néo-mensural. Nous vous conseillons donc, lorsque vous réalisez l'incipit d'une transcription, d'utiliser le style par défaut. Les crochets n'existent pas en notation grégorienne.

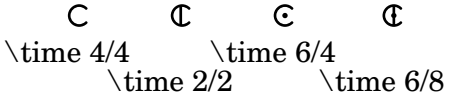
Problèmes connus et avertissements

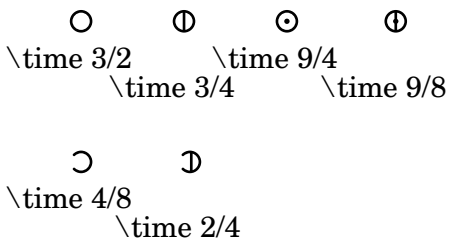
Les crochets anciens s'attachent aux hampes avec un léger décalage, suite à des modifications intervenues au début de la série 2.3.

L'alignement vertical des crochets par rapport aux lignes de la portée sous-entend que les hampes se terminent toujours soit sur une ligne, soit à l'exact milieu d'un interligne. Ceci n'est pas toujours réalisable, surtout si vous faites appel à des fonctionnalités avancées de présentation de la notation classique, qui, par définition, ne sont pas prévues pour être appliquées à la notation mensurale.

Métriques anciennes

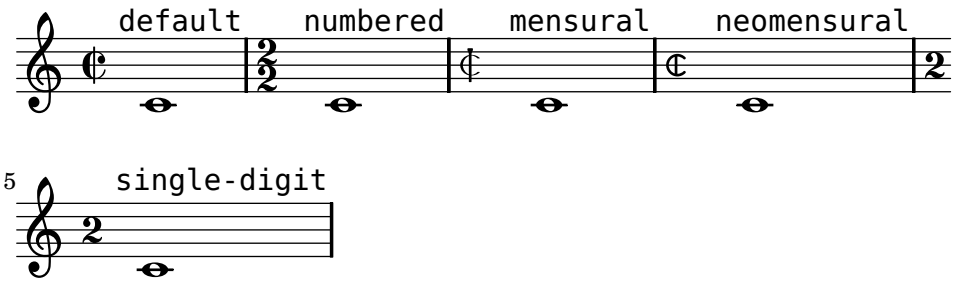
Les chiffrages de métrique mensurale sont partiellement pris en charge. Les glyphes ne font que représenter des métriques particulières. En d'autres termes, pour obtenir le glyphe correspondant à une métrique mensurale particulière à l'aide de la commande `\time n/m`, vous devez choisir la paire `(n,m)` parmi les valeurs suivantes :





La propriété `style` de l'objet Section "TimeSignature" dans *Référence des propriétés internes* permet d'accéder aux indicateurs de métrique anciens. Les styles `neomensural` et `mensural` sont disponibles. Vous avez vu ci-dessus le style `neomensural`, particulièrement utilisé pour l'incipit des transcriptions. Le style `mensural` imite l'aspect de certaines éditions du XVIe siècle.

Voici les différences entre les styles :



Voir aussi

Dans ce manuel : [Métrique], page 54 expose les principes généraux sur l'utilisation des indications de métrique.

Problèmes connus et avertissements

Les équivalences de durées de note ne sont pas modifiées par un changement de métrique. Par exemple, l'équivalence 1 brève pour 3 semi-brèves (*tempus perfectum*) doit s'effectuer à la main en entrant :

```
breveTP = #(ly:make-duration -1 0 3 2)
...
{ c\breveTP f1 }
```

Ce qui définira `breveTP` à 3/2 fois 2 = 3 fois une ronde.

Le symbole `old6/8alt` — symbole alternatif pour la métrique 6/8 – ne peut être utilisé grâce à une commande `\time`. Utilisez plutôt un `\markup`.

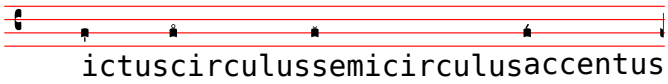
2.9.3 Signes de note supplémentaires

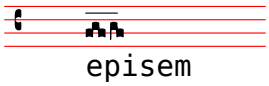
Articulations anciennes

En plus des signes d'articulation standards décrits à la section [Articulations et ornements], page 102, LilyPond fournit des articulations pour la musique ancienne. Elles sont spécifiquement destinées au style Editio Vaticana.

```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \override TextScript #'font-family = #'typewriter
    \override TextScript #'font-shape = #'upright
    \override Script #'padding = #-0.1
    a\ictus_"ictus" \break
    a\circulus_"circulus" \break
```

```
    a\semicirculus_"semicirculus" \break
    a\accentus_"accentus" \break
    \[ a_"episem" \episemInitium \pes b \flexa a b \episemFinis \flexa a \]
  }
}
```





Problèmes connus et avertissements

Certaines articulations sont verticalement trop proches de leurs têtes de note.

Le trait d'un episem n'est bien souvent pas apparent et, lorsqu'il l'est, son extension à droite est trop longue.

Guidons

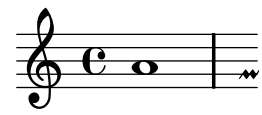
Un guidon — *custos*, pluriel *custodes* en latin — est un symbole qui apparaît à la fin d'une portée. Il montre la hauteur de la ou des premières notes de la portée suivante, donnant une indication judicieuse à l'exécutant.
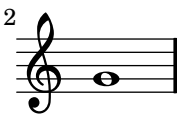
Les guidons étaient couramment utilisés jusqu'au XVIIe siècle. De nos jours, on les retrouve uniquement dans quelques formes particulières de notation telles que les éditions contemporaines de chant grégorien comme les *editio vaticana*. Différents glyphes existent selon le style de notation.

L'impression de guidons s'obtient en affectant, dans un bloc `\layout`, le Section "Custos_engraver" dans *Référence des propriétés internes* au contexte `Staff`, comme le montre l'exemple suivant.

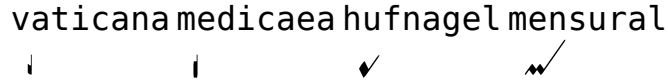
```
\layout {
  \context {
    \Staff
    \consists Custos_engraver
    Custos \override #'style = #'mensural
  }
}
```

Le résultat ressemblera à





Le glyphe du guidon est déterminé par la propriété `style`. Les styles disponibles sont `vaticana`, `medicaea`, `hufnagel` et `mensural`. En voici un aperçu :



Voir aussi

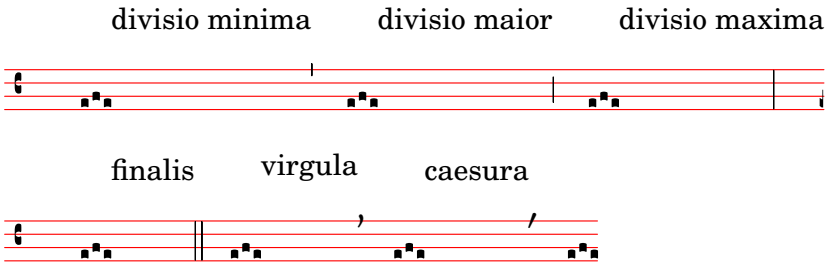
Référence des propriétés internes : Section “Custos” dans *Référence des propriétés internes*.

Morceaux choisis : Section “Notations anciennes” dans *Morceaux choisis*.

Divisions

Une division – *divisio*, pluriel *divisiones* en latin – est un symbole ajouté à la portée et utilisé en chant grégorien pour séparer les phrases ou parties. *Divisio minima*, *divisio maior* et *divisio maxima* peuvent respectivement s'interpréter comme une pauses courte, moyenne ou longue, à l'image des marques de respiration – cf. [Signes de respiration], page 116. Le signe *finalis* n'est pas uniquement une marque de fin de chant ; il sert aussi à indiquer la fin de chaque partie dans une structure verset/répons.

Les divisions sont disponibles après inclusion du fichier ‘`gregorian-init.ly`’. Ce fichier définit les commandes `\divisioMinima`, `\divisioMaior`, `\divisioMaxima` et `\finalis`. Certaines éditions utilisent *virgula* ou *caesura* en lieu et place de *divisio minima* ; c'est pourquoi ‘`gregorian-init.ly`’ définit aussi `\virgula` et `\caesura`.



Commandes prédéfinies

`\virgula`, `\caesura`, `\divisioMinima`, `\divisioMaior`, `\divisioMaxima`, `\finalis`.

Voir aussi

Manuel de notation : [Signes de respiration], page 116.

Référence des propriétés internes : Section “BreathingSign” dans *Référence des propriétés internes*.
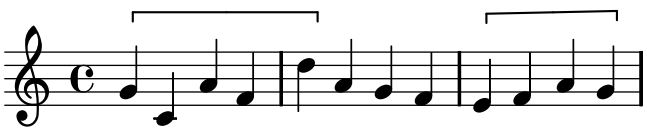
Morceaux choisis : Section “Instruments à vent” dans *Morceaux choisis*.

Ligatures

Une ligature est un symbole graphique qui représente un groupe d'au moins deux notes. Les ligatures ont commencé à apparaître dans les manuscrits de chant grégorien, pour indiquer des suites ascendantes ou descendantes de notes.

Les ligatures s'indiquent par une inclusion entre `\[` et `\]`. Certains styles de ligature peuvent demander un complément de syntaxe spécifique. Par défaut, le graveur Section “LigatureBracket” dans *Référence des propriétés internes* place un simple crochet au dessus de la ligature :

```
\transpose c c' {
  \[ g c a f d' \]
  a g f
  \[ e f a g \]
}
```



Selon le style de ligature désiré, il faut ajouter au contexte Section “Voice” dans *Référence des propriétés internes* le graveur de ligature approprié, comme nous le verrons plus loin. Seules sont disponibles les ligatures mensurales blanches, avec quelques limitations.

Problèmes connus et avertissements

La gestion de l’espacement spécifique aux ligatures n’est à ce jour pas implémentée. En conséquence, les ligatures sont trop espacées les unes des autres et les sauts de ligne mal ajustés. De plus, les paroles ne s’alignent pas de manière satisfaisante en présence de ligatures.

Les altérations ne pouvant être imprimées à l’intérieur d’une ligature, il faut les rassembler et les imprimer juste avant.

La syntaxe utilisée correspond à l’ancienne convention de préfixage `\[ expr_musicale\]`. Pour des raisons d’uniformité, nous opterons probablement pour le style en suffixe (postfix) `note\[ ... note\]`. En attendant, vous pouvez inclure le fichier ‘`gregorian-init.ly`’, qui fournit une fonction Scheme

`\ligature` *`expr_musicale`*

qui produit le même résultat, et dont la pérennité est assurée.

* Ligatures mensurales:: * Neumes ligaturés grégoriens::

Ligatures mensurales

Les ligatures mensurales blanches sont prises en charge, avec des limitations.

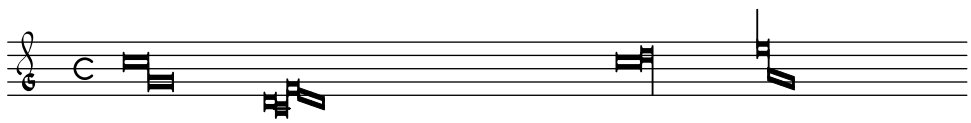
La gravure des ligatures mensurales blanches s’obtient après avoir ajouté le Section “Mensural_ligature_engraver” dans *Référence des propriétés internes* et enlevé le Section “Ligature_bracket_engraver” dans *Référence des propriétés internes* dans le contexte Section “Voice” dans *Référence des propriétés internes*, comme ici :

```
\layout {
  \context {
    \Voice
    \remove Ligature_bracket_engraver
    \consists Mensural_ligature_engraver
  }
}
```
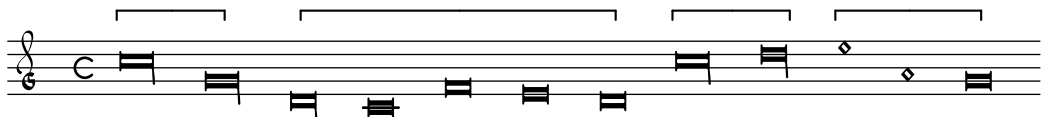
Lorsque le code ci-dessus est employé, l’aspect d’une ligature mensurale blanche est déterminé à partir des hauteurs et durées des notes qui la composent. Bien que cela demande un temps d’adaptation au nouvel utilisateur, cette méthode offre l’avantage que toute l’information musicale incluse dans la ligature est connue en interne. Ceci est non seulement important pour le rendu MIDI, mais aussi pour des questions de transcription automatisée d’une ligature.

Par exemple,

```
\set Score.timing = ##f
\set Score.defaultBarType = "empty"
\override NoteHead #'style = #'neomensural
\override Staff.TimeSignature #'style = #'neomensural
\clef "petrucci-g"
\[ c'\maxima g \]
\[ d\longa c\breve f e d \]
\[ c'\maxima d'\longa \]
\[ e'1 a g\breve \]
```



Si on ne remplace pas le Section “Ligature_bracket_engraver” dans *Référence des propriétés internes* par le Section “Mensural_ligature_engraver” dans *Référence des propriétés internes*, on obtient



Problèmes connus et avertissements

L’espacement horizontal n’est pas des meilleurs.

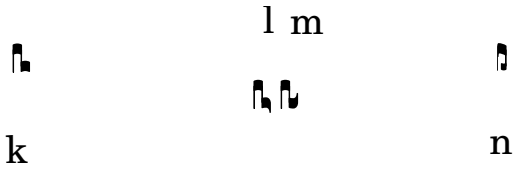
Neumes ligaturés grégoriens

Les neumes grégoriens conformément au style des Editio Vaticana sont pris en charge de façon assez limitée. Les ligatures élémentaires sont déjà disponibles, mais beaucoup de règles typographiques ne sont pas encore implémentées, notamment l’espacement horizontal des enchaînements de ligatures, l’alignement des paroles ou une gestion convenable des altérations.

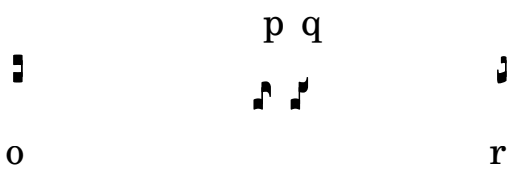
Le tableau ci-dessous inventorie les différents neumes contenus dans le second tome de l’Antiphonale Romanum (*Liber Hymnarius*) publié par l’abbaye de Solesmes en 1983.

Neuma aut Neumarum Elementa	**Figurae Rectae**	**Figurae Liquescentes Auctae**	**Figurae Liquescentes Deminutae**
`1. Punctum`	a b	c d e	f
`2. Virga`	g		
`3. Apostropha vel Stropha`	h	i	
`4. Oriscus`	j		

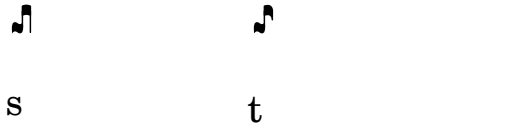
5. Clivis vel Flexa



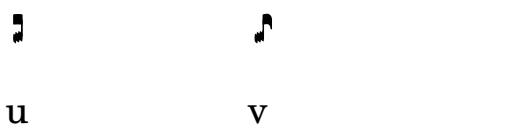
6. Podatus vel Pes
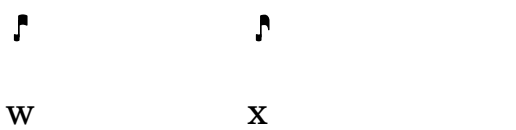


7. Pes Quassus



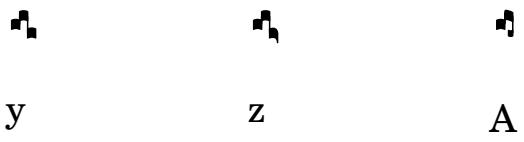
8. Quilisma Pes



9. Podatus Initio Debilis

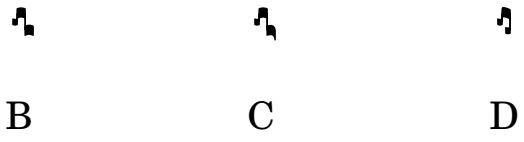


10. Torculus



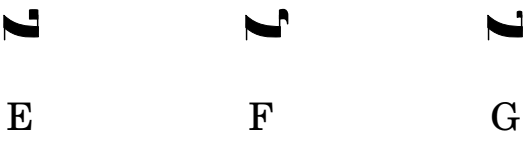
y z A

11. Torculus Initio Debilis



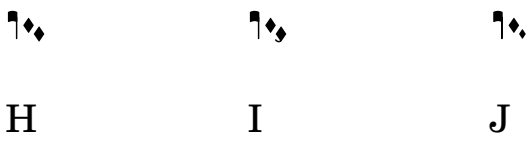
B C D

12. Porrectus



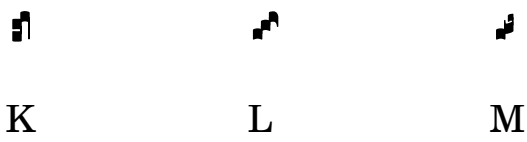
E F G

13. Climacus



H I J

14. Scandicus



K L M

15. Salicus



N O

16. Trigonus



P

Contrairement à la majorité des autres systèmes de notation neumatique, la manière de saisir les neumes n'a rien à voir avec leur apparence typographique ; elle se concentre plutôt sur le sens musical. Ainsi, `\[ a \pes b \flexa g \]` produit un *torculus* constitué de trois *punctums*, alors que `\[ a \flexa g \pes b \]` produit un *porrectus* avec une flexe incurvée et un seul *punctum*. Il n'existe pas de commande à proprement parler qui permette de spécifier la courbe d'une flexe ; c'est la source musicale qui va le déterminer. Le fondement d'une telle approche réside dans la distinction que nous faisons entre les aspects musicaux de la source et le style de notation que nous voulons obtenir. De ce fait, la même source pourra être utilisée pour imprimer dans un autre style de notation grégorienne.

Le tableau suivant présente les fragments de code qui ont permis de générer les neumes ligaturés du tableau précédent. Les lettres de la première colonne renvoient aux ligatures ci-dessus. La seconde colonne énumère le nom des ligatures, et la troisième le code ayant permis de les générer, se basant ici sur sol, la, si.

#	**Nom**	**Code source**
a	Punctum	`\[ b \]`
b	Punctum Inclinatum	`\[ \inclinatum b \]`
c	Punctum Auctum Ascendens	`\[ \auctum \ascendens b \]`
d	Punctum Auctum Descendens	`\[ \auctum \descendens b \]`
e	Punctum Inclinatum Auctum	`\[ \inclinatum \auctum b \]`
f	Punctum Inclinatum Parvum	`\[ \inclinatum \deminutum b \]`
g	Virga	`\[ \virga b \]`
h	Stropha	`\[ \stropha b \]`
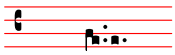
i	Stropha Aucta	`\[ \stropha \auctum b \]`
j	Oriscus	`\[ \oriscus b \]`
k	Clivis vel Flexa	`\[ b \flexa g \]`
l	Clivis Aucta Descendens	`\[ b \flexa \auctum \descendens g \]`
m	Clivis Aucta Ascendens	`\[ b \flexa \auctum \ascendens g \]`
n	Cephalicus	`\[ b \flexa \deminutum g \]`
o	Podatus vel Pes	`\[ g \pes b \]`
p	Pes Auctus Descendens	`\[ g \pes \auctum \descendens b \]`
q	Pes Auctus Ascendens	`\[ g \pes \auctum \ascendens b \]`
r	Epiphonus	`\[ g \pes \deminutum b \]`
s	Pes Quassus	`\[ \oriscus g \pes \virga b \]`
t	Pes Quassus Auctus Descendens	`\[ \oriscus g \pes \auctum \descendens b \]`
u	Quilisma Pes	`\[ \quilisma g \pes b \]`
v	Quilisma Pes Auctus Descendens	`\[ \quilisma g \pes \auctum \descendens b \]`
w	Pes Initio Debilis	`\[ \deminutum g \pes b \]`
x	Pes Auctus Descendens Initio Debilis	`\[ \deminutum g \pes \auctum \descendens b \]`
y	Torculus	`\[ a \pes b \flexa g \]`
z	Torculus Auctus Descendens	`\[ a \pes b \flexa \auctum \descendens g \]`
A	Torculus Deminutus	`\[ a \pes b \flexa \deminutum g \]`
B	Torculus Initio Debilis	`\[ \deminutum a \pes b \flexa g \]`
C	Torculus Auctus Descendens Initio Debilis	`\[ \deminutum a \pes b \flexa \auctum \descendens g \]`
D	Torculus Deminutus Initio Debilis	`\[ \deminutum a \pes b \flexa \deminutum g \]`
E	Porrectus	`\[ a \flexa g \pes b \]`
F	Porrectus Auctus Descendens	`\[ a \flexa g \pes \auctum \descendens b \]`
G	Porrectus Deminutus	`\[ a \flexa g \pes \deminutum b \]`

H	Climacus	`\[ \virga b \inclinatum a \inclinatum g \]`
I	Climacus Auctus	`\[ \virga b \inclinatum a \inclinatum \auctum g \]`
J	Climacus Deminutus	`\[ \virga b \inclinatum a \inclinatum \deminutum g \]`
K	Scandicus	`\[ g \pes a \virga b \]`
L	Scandicus Auctus Descendens	`\[ g \pes a \pes \auctum \descendens b \]`
M	Scandicus Deminutus	`\[ g \pes a \pes \deminutum b \]`
N	Salicus	`\[ g \oriscus a \pes \virga b \]`
O	Salicus Auctus Descendens	`\[ g \oriscus a \pes \auctum \descendens b \]`
P	Trigonus	`\[ \stropha b \stropha b \stropha a \]`

Les ligatures que nous venons de voir, bien que rudimentaires, donnent un aperçu des possibilités de former des ligatures grégoriennes. En théorie, vous pouvez inclure entre les délimiteurs `\[` et `\]`, autant de sons que nécessaires à la ligature, ainsi que de préfixes tels que `\pes`, `\flexa`, `\virga`, `\inclinatum`, ... Bien sûr, les règles de construction présentées ci-dessus peuvent se combiner, ce qui permet la création d'une infinité de ligatures.

Les points d'*augmentum*, ou *morae*, s'obtiennent avec la fonction `\augmentum`. Notez que cette fonction `\augmentum` est implémentée en tant que fonction unaire plutôt que comme un préfixe de note. Par conséquent, `\augmentum \virga c` ne donnera rien de particulier. Il faut l'utiliser avec la syntaxe `\virga \augmentum c` ou `\augmentum {\virga c}`. Par ailleurs, l'expression `\augmentum {a g}` constitue une forme abrégée de `\augmentum a \augmentum g`.

```
\include "gregorian.ly"
\score {
  \new VaticanaVoice {
    \[ \augmentum a \flexa \augmentum g \]
    \augmentum g
  }
}
```



Commandes prédéfinies

LilyPond dispose des préfixes suivants : `\virga`, `\stropha`, `\inclinatum`, `\auctum`, `\descendens`, `\ascendens`, `\oriscus`, `\quilisma`, `\deminutum`, `\cavum`, `\linea`.

Les préfixes de note peuvent s'agglutiner, modulo quelques restrictions. Par exemple, on peut appliquer un `\descendens` ou un `\ascendens` à une note, mais pas les deux simultanément à une même note.

Deux notes adjacentes peuvent être reliées grâce aux commandes `\pes` ou `\flexa` pour marquer une ligne mélodique respectivement ascendante ou descendante.

Utilisez la fonction musicale unaire `\augmentum` pour ajouter des points d'augmentum.

Problèmes connus et avertissements

Lorsqu'un `\augmentum` apparaît dans une ligature en fin de portée, son placement vertical peut être erroné. Pour y remédier, ajoutez un silence invisible, `s8` par exemple, comme dernière note de cette portée.

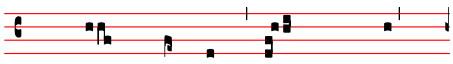
L'`\augmentum` devrait être implémenté en tant que préfixe plutôt qu'en tant que fonction unaire, afin qu'`\augmentum` puisse s'intégrer avec d'autres préfixes dans n'importe quel ordre.

2.9.4 Contextes prédéfinis

Contextes du chant grégorien

Les contextes `VaticanaVoiceContext` et `VaticanaStaffContext` permettent de graver le chant grégorien dans le style des éditions vaticanes. Ces contextes initialisent les propriétés de tous les autres contextes et objets graphiques à des valeurs adéquates, de telle sorte que vous pouvez tout de suite vous lancer dans la saisie de votre chant, comme ci-dessous :

```
\include "gregorian.ly"
\score {
  <<
    \new VaticanaVoice = "cantus" {
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \]
      f \divisioMinima
      \[ f\melisma \pes a c' c' \pes d'\melismaEnd \]
      c' \divisioMinima \break
      \[ c'\melisma c' \flexa a \]
      \[ a \flexa \deminutum g\melismaEnd \] f \divisioMinima
    }
    \new Lyrics \lyricsto "cantus" {
      San- ctus, San- ctus, San- ctus
    }
  >>
}
```



San- ctus, San- ctus,



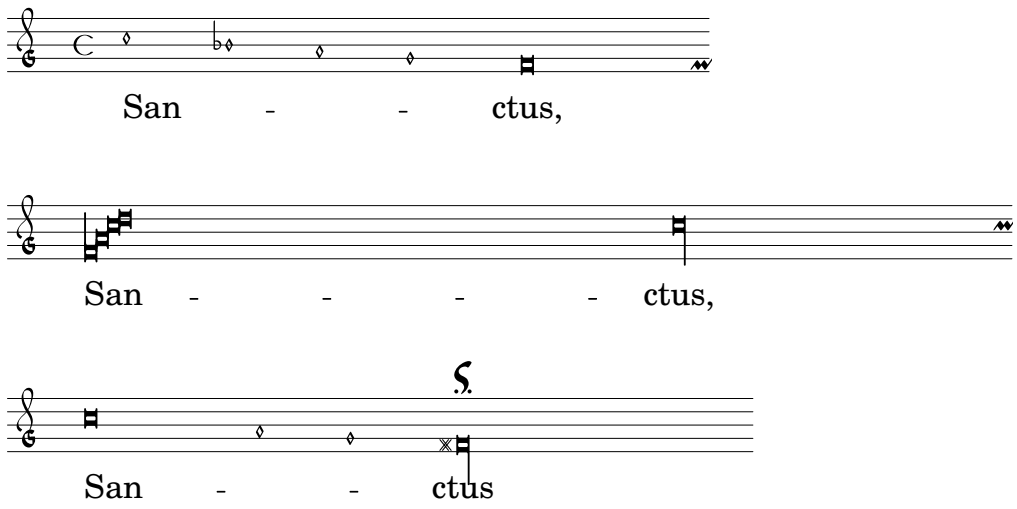
San- ctus

Les contextes de la musique mensurale

Les contextes `MensuralVoiceContext` et `MensuralStaffContext` permettent de graver des chants dans le style mesuré. Ces contextes initialisent les propriétés de tous les autres contextes et objets graphiques à des valeurs adéquates, de telle sorte que vous pouvez tout de suite vous lancer dans la saisie de votre chant comme ci-après :

```
\score {
  <<
    \new MensuralVoice = "discantus" \transpose c c' {
      \override Score.BarNumber #'transparent = ##t {
        c'1\melisma bes a g\melismaEnd
```

```
        f\breve
        \[ f1\melisma a c'\breve d'\melismaEnd \]
        c'\longa
        c'\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```



2.9.5 Transcription de musique mensurale

Éditions ancienne et moderne à partir d'une même source

Des incipits

Mise en forme de la musique mensurale

Transcription de chant grégorien

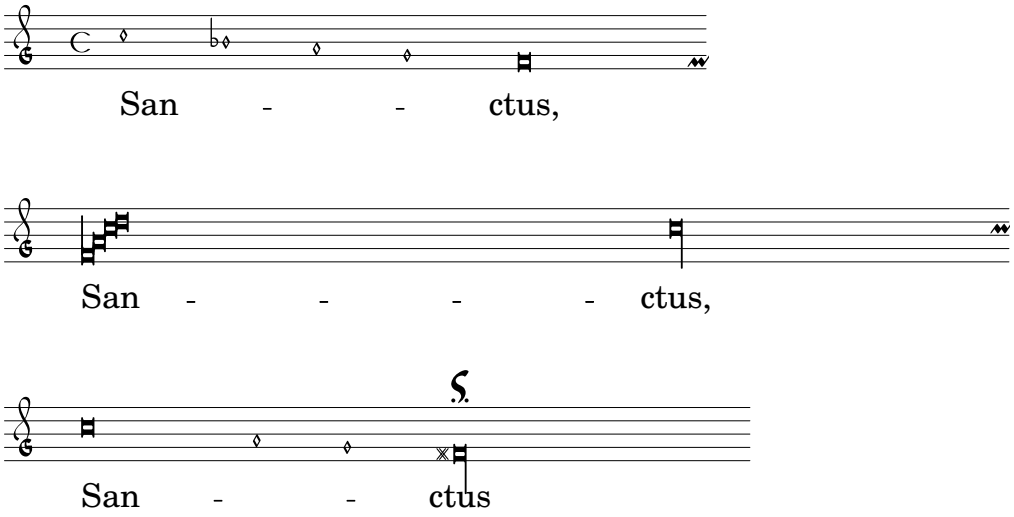
2.9.6 Notation éditoriale

Altérations accidentelles suggérées

Les contextes `MensuralVoiceContext` et `MensuralStaffContext` permettent de graver des chants dans le style mesuré. Ces contextes initialisent les propriétés de tous les autres contextes et objets graphiques à des valeurs adéquates, de telle sorte que vous pouvez tout de suite vous lancer dans la saisie de votre chant comme ci-après :

```
\score {
  <<
    \new MensuralVoice = "discantus" \transpose c c' {
      \override Score.BarNumber #'transparent = ##t {
        c'1\melisma bes a g\melismaEnd
        f\breve
        \[ f1\melisma a c'\breve d'\melismaEnd \]
```

```
        c'\longa
        c'\breve\melisma a1 g1\melismaEnd
        fis\longa^\signumcongruentiae
      }
    }
    \new Lyrics \lyricsto "discantus" {
      San -- ctus, San -- ctus, San -- ctus
    }
  >>
}
```



Notation du rythme dans la musique baroque

2.10 Musiques du monde

Ce chapitre a pour objet la notation des musiques traditionnelles autres qu'occidentales.

2.10.1 Noms de note et altérations non-occidentaux

De nombreuses musiques autres qu'occidentales – et même certaines formes de musique traditionnelle occidentales – ont recours à des systèmes de notation alternatifs ou étendus, qui ne s'intègrent pas forcément dans notre système standard.

Sans certains cas où la notation standard est utilisée, ces différences de hauteurs seront implicites. Par exemple, la musique arabe est reproduite en notation standard, et utilise des quarts de ton, l'altération réelle dépendant du contexte. D'autres, par contre, font appel à une notation étendue, voire toute particulière.

La *musique classique turque*, ou musique ottomane, utilise des formes mélodiques appelées *makamlar*, dans laquelle les tons sont divisés en 9 intervalles. Du point de vue actuel des pratiques de notation, il est possible d'utiliser les notes occidentales (do, ré, mi. . .) auxquelles on ajoutera l'atlération spécifique à la musique turque. Ces différentes altérations sont définies dans le fichier '`makam.ly`' – reportez vous au chapitre Section "Autres sources de documentation" dans *Manuel d'initiation* pour le localiser sur votre système. Vous trouverez, dans le tableau suivant, le nom de ces altérations, le suffixe à utiliser, ainsi que la fraction de ton entier à laquelle elles conrrespondent.

Nom d'altération	**suffixe**	**altération**

büyük mücenneb (dièse)	-bm	+8/9
kücük mücenneb (dièse)	-k	+5/9
bakiye (dièse)	-b	+4/9
koma (dièse)	-c	+1/9
koma (bémol)	-fc	-1/9
bakiye (bémol)	-fb	-4/9
kücük mücenneb (bémol)	-fk	-5/9
büyük mücenneb (bémol)	-fbm	-8/9

Pour plus d'information sur la musique classique turque et les makamlar, reportez-vous au chapitre Section 2.10.3 [Musique classique turque], page 378.
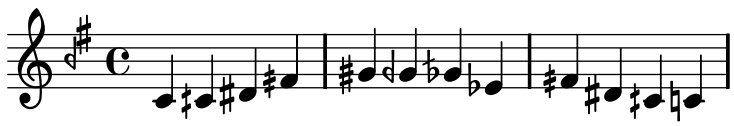
Morceaux choisis

Exemple de musique « Makam »

Le « makam » est une forme de mélodie turque qui utilise des altérations d'un neuvième de ton. Consultez le fichier d'initialisation '`makam.ly`' pour plus de détails sur les hauteurs et altérations utilisées (voir le chapitre 4.6.3 - Autres sources d'information du manuel d'initiation pour le localiser).

```
% Initialize makam settings
\include "makam.ly"

\relative c' {
  \set Staff.keySignature = #`((6 . ,(- KOMA)) (3 . ,BAKIYE))
  c4 cc db fk
  gbm4 gfc gfb efk
  fk4 db cc c
}
```



2.10.2 Musique arabe

Ce chapitre souligne les questions propres à la notation de la musique arabe.

Références pour la musique arabe

Jusqu'à nos jours, la musique arabe a principalement été transmise comme une tradition orale. Lorsqu'elle était transcrite, c'était en général sous forme de canevas sur lequel le rôle des interprètes était d'improviser substantiellement. La notation occidentale, cependant, est de plus en plus utilisée, avec quelques variations, pour transmettre et préserver la musique arabe.

Certains éléments de notation musicale occidentale, tels que les transcriptions d'accords ou de parties indépendantes, ne sont pas nécessaires pour retranscrire les pièces arabes les plus traditionnelles. Il y a cependant quelques besoins spécifiques, tels que des intervalles se trouvant entre le demi-ton et le ton qui s'ajoutent aux intervalles mineurs ou majeurs utilisés dans la musique occidentale. Il est également nécessaire de regrouper et de noter un grand nombre de maqams (modes) différents qui font partie de la musique arabe.

En général, la notation de la musique arabe n'essaie pas d'indiquer précisément les micro-intervalles intervenant dans la pratique musicale.

Plusieurs particularités propres à la musique arabe sont traitées ailleurs :

- Les noms des notes et altérations (y compris les quarts de tons) peuvent être adaptés comme l'explique [Nom des notes dans d'autres langues], page 7.
- Les armures peuvent également être adaptées comme expliqué dans [Armure], page 16.
- Des métriques complexes peuvent nécessiter de grouper les notes manuellement, comme décrit dans [Barres de ligature manuelles], page 80.
- Les *Takasim*, qui sont des improvisations rythmiquement libres, peuvent être écrites en omettant les barres de mesures, de la façon indiquée dans [Musique sans métrique], page 63.

Voir aussi

Manuel de notation : [Nom des notes dans d'autres langues], page 7, [Armure], page 16, [Barres de ligature manuelles], page 80.
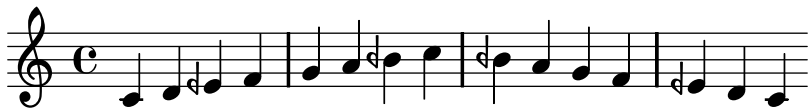
Morceaux choisis : Section "Musiques du monde" dans *Morceaux choisis*.

Noms des notes en arabe

Les noms de note les plus traditionnels en arabe peuvent être très longs et ne conviennent pas à l'écriture de la musique, aussi ne sont ils pas utilisés. Les noms de note anglais ne sont pas très courants dans l'éducation musicale en arabe, c'est pourquoi on utilise plus volontiers les noms italiens (`do, re, mi, fa, sol, la, si`). On peut également utiliser des altérations, comme cela est expliqué dans [Nom des notes dans d'autres langues], page 7.
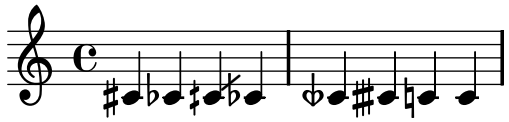
Par exemple, voici comment on peut écrire la gamme arabe *rast* :

```
\include "arabic.ly"
\relative do' {
  do re misb fa sol la sisb do sisb la sol fa misb re do
}
```



Le symbole indiquant un demi-bémol ne correspond pas au symbole utilisé dans la notation arabe. Si le symbole particulier du demi-bémol arabe doit absolument être utilisé, il est possible de s'en approcher en faisant précéder la note par la commande `\dwn` définie dans le fichier '`arabic.ly`'. Cette méthode ne peut toutefois pas être utilisée pour modifier l'aspect du demi-bémol dans l'armure.

```
\include "arabic.ly"
\relative do' {
  \set Staff.extraNatural = ##f
  dod dob dosd \dwn dob dobsb dodsd do do
}
```



Voir aussi

Manuel de notation : [Nom des notes dans d'autres langues], page 7

Morceaux choisis : Section "Musiques du monde" dans *Morceaux choisis*.

Armures arabes

Outre les armures mineures et majeures, les armures suivantes sont définies dans la fichier '`arabic.ly`' : *bayati*, *rast*, *sikah*, *iraq* et *kurd*. Ces armatures définissent un petit nombre de groupes de maqams plutôt que le grand nombre de maqams habituellement utilisés.

En général, un maqam utilise l'armure de son groupe ou d'un groupe voisin et diverses altérations accidentelles sont indiquées tout au long de la musique.

Par exemple, pour indiquer l'armure d'une pièce en maqam muhayer :

```
\key re \bayati
```

Ici, *re* est le nom de la tonalité par défaut de la base maqam dans le groupe.

Alors que l'armure correspond à un groupe, il est courant que le titre mentionne un maqam en particulier. Ainsi, dans cet exemple, le titre devrait faire apparaître le nom du maqam muhayer.

D'autres maqams du même groupe bayati, comme l'explique le tableau ci-dessous (bayati, hussaini, saba, et ushaq), peuvent être indiqués de la même manière. Ils sont autant de variations du maqam de base, le plus courant, du groupe (en l'occurrence, bayati). En général, c'est dans les tétracordes supérieurs que ces modes apparentés diffèrent, ou dans certains détails de disposition qui ne changent pas fondamentalement leur nature.

Certains maqams ne sont qu'une modulation de leur maqam de base. Ainsi, dans ce même groupe de bayati, du maqam Nawa, dont la modulation est indiquée entre parenthèses dans le tableau. Les maqams arabes n'admettent que des modulations limitées, en raison de la nature des instruments de musique arabes. Le Nawa peut être indiqué comme suit :

```
\key sol \bayati
```

En musique arabe, le terme utilisé pour désigner un groupe maqam, tel que bayati, est également lui-même un maqam, généralement le plus important dans le groupe ; on peut le considérer comme un maqam de base.

Voici une suggestion de groupement qui relie les maqams les plus courants à leur armure :

groupe maqam	**Armure**	**Tonique**	**Autres maqams dans le groupe (tonique)**
ajam	major	sib	jaharka (fa)
bayati	bayati	re	hussaini, muhayer, saba, ushaq, nawa (sol)
hijaz	kurd	re	shahnaz, shad arban (sol), hijazkar (do)
irakien	iraq	sisb	-
kurde	kurd	re	kurde hijazkar (do)
nahawand	minor	do	busalik (re), farah faza (sol)
nakriz	mineur	do	nawa athar, hisar (re)
rast	rast	do	mahur, yakah (sol)
sikah	sikah	misb	huzam

Morceaux choisis

Armures inhabituelles

La commande `\key` détermine la propriété `keySignature` d'un contexte `Staff`.

Des armures inhabituelles peuvent être spécifiées en modifiant directement cette propriété. Il s'agit en l'occurence de définir une liste :
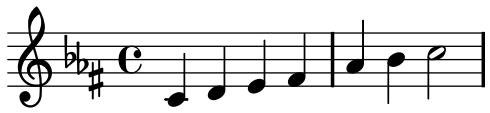
```
\set Staff.keySignature = #`(((octave . pas) . altération) ((octave . pas) .
altération) ...)
```

dans laquelle, et pour chaque élément, `octave` spécifie l'octave (0 pour celle allant du do médium au si supérieur), `pas` la note dans cette octave (0 pour do et 6 pour si), et `altération` sera `,SHARP ,FLAT ,DOUBLE-SHARP` etc. (attention à la virgule en préfixe).

Une formulation abrégée – `(pas . altération)` – signifie que l'altération de l'élément en question sera valide quel que soit l'octave.

Voici, par exemple, comment générer une gamme par ton :

```
\relative c' {
  \set Staff.keySignature = #`(((0 . 6) . ,FLAT)
                              ((0 . 5) . ,FLAT)
                              ((0 . 3) . ,SHARP))
  c4 d e fis
  aes4 bes c2
}
```



Voir aussi

Manuel de notation : [Armure], page 16.

Manuel d'initiation : Section "Altérations et armure" dans *Manuel d'initiation*.

Référence des propriétés internes : Section "KeySignature" dans *Référence des propriétés internes*.

Morceaux choisis : Section "Musiques du monde" dans *Morceaux choisis*, Section "Hauteurs" dans *Morceaux choisis*.

Métriques arabes

Quelques formes de musique classique arabes et turques telles que *Semai* utilisent des métriques inhabituelles comme le 10/8. Ceci peut impliquer une manière de grouper les notes fort différente de la musique écrite existante, où les notes ne sont pas groupées par temps mais d'une façon difficile à reproduire automatiquement. Il est possible d'y remédier en désactivant la ligature automatique et en groupant les notes manuellement. Lorsque l'enjeu n'est pas de reproduire exactement un texte existant, il est toujours possible d'ajuster le comportement de ligature automatique ou d'utiliser des chiffres de mesure composés.

Morceaux choisis
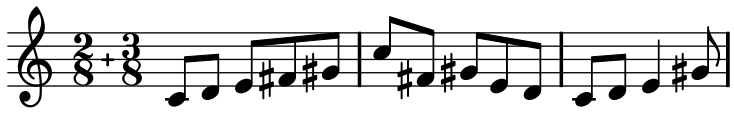
Métrique décomposée

Des métriques telles que « 5/8 » peuvent s'interpréter sous une forme décomposée — « 3/8 + 2/8 » par exemple — qui combine plusieurs métriques. LilyPond est capable de rendre ce type de notation, plus facile à lire et à interpréter, en imprimant cette métrique composite et en adaptant les règles de ligature automatique en conséquence.

```
#(define ((compound-time one two num) grob)
   (grob-interpret-markup grob
                          (markup #:override '(baseline-skip . 0) #:number
                                  (#:line ((#:column (one num))
                                           #:vcenter "+"
                                           (#:column (two num)))))))

\relative c' {
  \override Staff.TimeSignature #'stencil = #(compound-time "2" "3" "8")
  \time 5/8
```

```
  \set Staff.beatStructure = #'(2 3)
  c8 d e fis gis
  c8 fis, gis e d
  c8 d e4 gis8
}
```

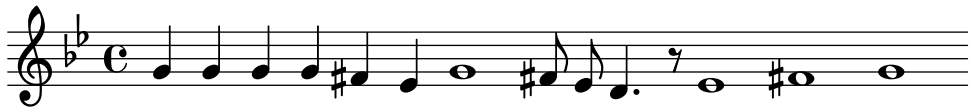


Arabic improvisation

For improvisations or taqasim which are temporarily free, the time signature can be omitted and `\cadenzaOn` can be used. Adjusting the accidental style might be required, since the absence of bar lines will cause the accidental to be marked only once. Here is an example of what could be the start of a hijaz improvisation:

```
\include "arabic.ly"

\relative sol' {
  \key re \kurd
  #(set-accidental-style 'forget)
  \cadenzaOn
  sol4 sol sol sol fad mib sol1 fad8 mib re4. r8 mib1 fad sol
}
```



Voir aussi

Manuel de notation : [Barres de ligature manuelles], page 80, [Barres de ligature automatiques], page 70, [Musique sans métrique], page 63, [Altérations accidentelles automatiques], page 20, [Définition des règles de ligature automatique], page 73, [Métrique], page 54.

Morceaux choisis : Section “Musiques du monde” dans *Morceaux choisis*.
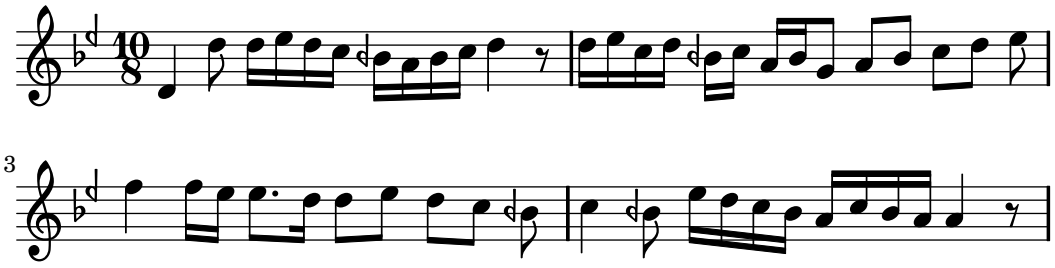
Exemple de musique arabe

Voici un modèle qui utilise également le début d’un Semai turc courant dans l’éducation musicale arabe, pour illustrer quelques unes des particularités de la notation musicale arabe, comme des intervalles intermédiaires et des modes inhabituels traités dans ce chapitre.

```
\include "arabic.ly"
\score {
  \relative re' {
    \set Staff.extraNatural = ##f
    \set Staff.autoBeaming = ##f
    \key re \bayati
    \time 10/8

    re4 re'8 re16 [misb re do] sisb [la sisb do] re4 r8
    re16 [misb do re] sisb [do] la [sisb sol8] la [sisb] do [re] misb
    fa4 fa16 [misb] misb8. [re16] re8 [misb] re  [do] sisb
    do4 sisb8 misb16 [re do sisb] la [do sisb la] la4 r8
```

```
  }
  \header {
    title = "Semai Muhayer"
    composer = "Jamil Bek"
  }
}
```



Voir aussi

Morceaux choisis : Section “Musiques du mondes” dans *Morceaux choisis*.

Lectures complémentaires pour la musique arabe

1. *La musique des Arabes* par Habib Hassan Touma [Amadeus Press, 1996], contient une étude des maqams et leur méthode de classification.

   Il existe également de nombreux sites web qui expliquent les maqams, dont quelques uns s’accompagnent d’exemples audio :

   - http://www.maqamworld.com/
   - http://www.turath.org/

   Si tout le monde s’accorde à apparenter les maqams d’après leur tétracorde inférieur, parfois transposé, les méthodes de classification varient dans certains détails.

2. Les sources ne sont pas entièrement cohérentes (parfois dans un même texte) quant à la manière d’indiquer l’armure de certains maqams. Il est courant, cependant, d’utiliser une armure par groupe plutôt qu’une armure différente pour chaque maqam.

   Des méthodes de luth arabe, l’*Oud*, par les auteurs suivants, contiennent des exemples de compositions principalement turques et arabes,

   - Charbel Rouhana
   - George Farah
   - Ibrahim Ali Darwish Al-masri

2.10.3 Musique classique turque

Ce chapitre met en évidence des questions propres à la notation de la musique classique turque.

Références pour la musique classique turque

La musique classique turque s’est développée dans l’Empire Ottoman à peu près à la même période que la musique classique en Europe, et a continué jusqu’au XXe et XXIe siècle comme une tradition vibrante et distincte avec sa propre théorie, ses propres formes, et styles d’interprétation. Parmi ses caractéristiques remarquables, se trouve l’usage de micro-intervalles fondés sur des « commas » d’un neuvième de ton, dont sont dérivées les formes mélodiques *makam* (pluriel *makamlar*).

Quelques questions relatives à la musique classique turque sont traitées dans d’autres chapitres :

- Les noms de notes et altérations sont mentionnés dans Section 2.10.1 [Noms de note et altérations non-occidentaux], page 372.

Noms des notes en turc

La musique classique turque attribue traditionnellement un nom unique à chaque hauteur, et du fait de la division du ton en neuf parts, les makamlar emploient une échelle de hauteurs complètement différente des gammes et modes d'occident : *koma* de 1/9 de ton entier, *eksik bakiye* (3/9), *bakiye* (4/9), *kücük mücenneb* (5/9), *büyük mücenneb* (8/9), *tanîni* (un ton entier) et *artık ikili* (12/9 ou 13/9 de ton).

D'un point de vue de notation moderne, il est pratique d'utiliser positions occidentales des notes sur la portée (do, ré, mi...) avec des altérations spéciales qui haussent ou baissent les notes par intervalles de 1/9, 4/9, 5/9 et 8/9 de ton. Ces altérations sont définies dans le dossier '`makam.ly`' (pour trouver l'emplacement de ce dossier dans votre système, voir Section "Autres sources de documentation" dans *Manuel d'initiation*). Une description plus détaillée se trouve dans Section 2.10.1 [Noms de note et altérations non-occidentaux], page 372.

3 Généralités en matière d'entrée et sortie

Nous n'allons pas, dans ce chapitre, parler directement de notation, mais plutôt du contenu des fichiers source et du résultat produit par LilyPond.

3.1 Agencement du code

LilyPond traite des fichiers textuels. Ces fichiers portent par convention une extension `.ly`.

3.1.1 Structure d'une partition

Un bloc `\score` contient obligatoirement une seule expression musicale délimitée par des accolades :
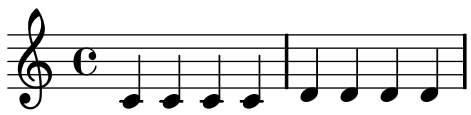
```
\score {
...
}
```

Note : Il ne doit y avoir qu'**une seule** expression musicale globale dans un bloc `\score`, et elle **doit** être bornée par une paire d'accolades.

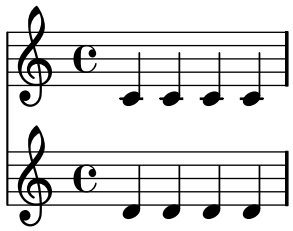
Cette unique expression musicale peut être de n'importe quelle taille et contenir d'autres expressions musicales aussi complexes soient elles. Voici quelques exemples d'expression musicale :

```
{ c'4 c' c' c' }
```

```
{
  { c'4 c' c' c' }
  { d'4 d' d' d' }
}
```



```
<<
  \new Staff { c'4 c' c' c' }
  \new Staff { d'4 d' d' d' }
>>
```



```
{
  \new GrandStaff <<
    \new StaffGroup <<
      \new Staff { \flute }
      \new Staff { \hautbois }
    >>
    \new StaffGroup <<
      \new Staff { \violonI }
      \new Staff { \violonII }
```

```
    >>
  >>
}
```

Les commentaires constituent l'une des rares exceptions à cette règle immuable – voir Section 3.1.5 [Structure de fichier], page 384 pour les autres. Qu'il s'agisse d'une seule ligne ou de tout un bloc – délimité par `%{ .. %}` – un commentaire peut se placer n'importe où dans le fichier source, aussi bien à l'intérieur qu'à l'extérieur du bloc `\score`, ou encore à l'intérieur ou à l'extérieur de l'expression musicale contenue dans un bloc `\score`.

Lorsqu'un fichier ne comprend qu'un bloc `\score`, celui-ci est implicitement inclus dans un bloc `\book`. Le bloc `\book` d'un fichier source permet la production d'au moins un fichier dont le nom sera, par défaut, déduit du fichier source : le traitement de '`fandangopourelephants.ly`' produira donc '`fandangopourelephants.pdf`'. Pour de plus amples informations à propos du bloc `\book`, lisez Section 3.1.2 [Plusieurs partitions dans un même ouvrage], page 381, Section 3.1.3 [Plusieurs éditions pour une même source], page 382 et Section 3.1.5 [Structure de fichier], page 384.

Voir aussi

Manuel d'initiation : Section "Travail sur les fichiers d'entrée" dans *Manuel d'initiation*, Section "Les expressions musicales en clair" dans *Manuel d'initiation*, Section "La partition est une (unique) expression musicale composée" dans *Manuel d'initiation*.

3.1.2 Plusieurs partitions dans un même ouvrage

Un ouvrage peut se composer de plusieurs morceaux et de texte. C'est le cas des cahiers d'exercices ou d'une partie d'orchestre avec ses différents mouvements. Chaque mouvement fait l'objet d'un bloc `\score`,

```
\score {
  ..musique..
}
```

et le texte est contenu dans un bloc `\markup`,

```
\markup {
  ..texte..
}
```

Les différents mouvements et textes qui apparaissent dans un même fichier '`.ly`' ne composeront en principe qu'un seul fichier résultant.

```
\score {
  ..
}
\markup {
  ..
}
\score {
  ..
}
```

Attention cependant si vous travaillez avec lilypond-book : il vous faudra explicitement mentionner le bloc `\book`, en l'absence de quoi seul le premier `\score` ou `\markup` apparaîtra après traitement.

L'entête de chaque pièce peut se placer au sein du bloc `\score` ; le contenu du champ `piece` viendra s'imprimer avant chaque mouvement. De même, le titre de l'ouvrage peut se placer au sein du bloc `\book`. Dans le cas contraire, le contenu du bloc `\header` placé en début de fichier sera utilisé.

```
\header {
  title = "Huit miniatures"
  composer = "Igor Stravinsky"
}
\score {
  ...
  \header { piece = "Romance" }
}
\markup {
   ..texte du second couplet..
}
\markup {
   ..texte du troisième couplet..
}
\score {
  ...
  \header { piece = "Menuet" }
}
```

Plusieurs pièces seront regroupées dans un même « chapitre » à l'aide d'un bloc `\bookpart`. Les différentes parties sont séparées par un saut de page et peuvent comporter un titre à l'instar de l'ouvrage dès lors que vous y insérez un bloc `\header`.

```
\bookpart {
  \header {
    title = "Titre de l'ouvrage"
    subtitle = "Première partie"
  }
  \score { ... }
  ...
}
\bookpart {
  \header {
    subtitle = "Deuxième partie"
  }
  \score { ... }
  ...
}
```

3.1.3 Plusieurs éditions pour une même source

Dès lors que vous inscrivez plusieurs blocs `\book` dans un même fichier '`.ly`', chacun d'eux donnera lieu à un résultat indépendant. Lorsqu'aucun bloc `\book` n'est spécifié dans le fichier source, LilyPond considère que l'intégralité du fichier constitue un bloc `\book` unique, comme indiqué à la rubrique Section 3.1.5 [Structure de fichier], page 384.

LilyPond fait en sorte, lorsque plusieurs fichiers sont produits à partir d'une même source, qu'aucun résultat d'un bloc `\book` n'écrase celui qui a été généré pour un bloc `\book` précédent.

Dans les faits, et si le nom du fichier produit est repris de sa source – comportement par défaut –, un suffixe lui sera ajouté pour chaque `\book`. Il s'agit en principe d'un pseudo numéro de version. Ainsi, le fichier '`huitminiatures.ly`' qui contiendrait

```
\book {
  \score { ... }
  \layout { ... }
}
```

```
\book {
  \score { ... }
  \layout { ... }
}
\book {
  \score { ... }
  \layout { ... }
}
```

génèrera

- ‘huitminiatures.pdf’,
- ‘huitminiatures-1.pdf’ et
- ‘huitminiatures-2.pdf’.

3.1.4 Nom des fichiers de sortie

LilyPond vous permet de prendre le contrôle dans la dénomination des fichiers que vous voulez générer, quel que soit le moteur de rendu utilisé.

Nous avons vu dans la rubrique précédente que LilyPond évite les conflits de nom des fichiers qu'il génère à partir d'une même source. Vous pouvez même définir vous-même le suffixe qui sera appliqué à chacun des blocs `\book`. Ainsi, en reprenant l'exemple ci-avant, vous obtiendrez les fichiers ‘`huitminiatures-Romance.pdf`’, ‘`huitminiatures-Menuet.pdf`’ et ‘`huitminiatures-Nocturne.pdf`’ en ajoutant simplement une déclaration `\bookOutputSuffix` au sein de chaque bloc `\book`.

```
\book {
  \bookOutputSuffix "Romance"
  \score { ... }
  \layout { ... }
}
\book {
  \bookOutputSuffix "Menuet"
  \score { ... }
  \layout { ... }
}
\book {
  \bookOutputSuffix "Nocturne"
  \score { ... }
  \layout { ... }
}
```

La déclaration `\bookOutputName` vous permet de définir vous-même le nom du fichier généré pour un bloc `\book` :

```
\book {
  \bookOutputName "Romance"
  \score { ... }
  \layout { ... }
}
\book {
  \bookOutputName "Menuet"
  \score { ... }
  \layout { ... }
}
\book {
```

```
  \bookOutputName "Nocturne"
  \score { ... }
  \layout { ... }
}
```

Le traitement de ce fichier produira :

- 'Romance.pdf',
- 'Menuet.pdf' et
- 'Nocturne.pdf'.

3.1.5 Structure de fichier

Un fichier .ly peut contenir un certain nombre d'expression de haut niveau. Les expressions de haut niveau sont les suivantes :

- Une définition de sortie, comme \paper, \midi et \layout. Ces définitions, lorsqu'elles se trouvent à un niveau supérieur, s'appliqueront à l'intégralité de l'ouvrage. Si l'une de ces expression apparaît à plusieurs reprises, la dernière aura préséance.
- Une expression scheme pure, telle que #(set-default-paper-size "a7" 'landscape) ou #(ly:set-option 'point-and-click #f).
- Un bloc \header, dont le contenu sera de portée globale – ce qui est le cas en général pour le titre ou l'auteur entre autres.
- Un bloc \score pour la partition. Cette partition sera assemblée avec les autres partitions se trouvant au même niveau pour composer le \book. Vous pouvez modifier ce comportement à l'aide de la variable toplevel-score-handler placée en tête.
- Un bloc \book permet de regrouper naturellement plusieurs mouvements – autrement dit plusieurs blocs \score – dans un même document. Lorsqu'il y a plusieurs \scores, LilyPond génère un seul fichier dans lequel les mouvements sont mis les uns à la suite des autres, ce pour chacun des blocs \book rencontrés. La seule raison qui peut vous demander d'expliciter plusieurs blocs \book dans un fichier '.ly' est lorsque vous avez besoin de générer différents documents à partir d'une même source. La présence explicite d'un bloc \book est aussi nécessaire lorsque vous travaillez sur un document lilypond-book qui reprendrait plusieurs \scores ou \markups dans un même extrait. Vous pouvez modifier ce comportement à l'aide de la variable toplevel-score-handler placée en tête.
- Un bloc \bookpart. Un ouvrage peut se découper en plusieurs parties à l'aide de blocs \bookpart, aussi bien pour alléger le travail de l'algorithme de calcul des sauts de page, que si les réglages du bloc \paper diffèrent d'une partie à l'autre.
- Une expression musicale telle que

  ```
  { c'4 d' e'2 }
  ```

  Ce bout de code sera placé dans un \score et intégré à l'ouvrage en même temps que tous les autres \scores ou expressions musicales. En d'autres termes, un fichier qui ne contiendrait que cette simple expression musicale sera traduit en

  ```
  \book {
    \score {
      \new Staff {
        \new Voice {
          { c'4 d' e'2 }
        }
      }
    }
  \layout { }
  \header { }
  ```

```
}
```

Vous pouvez modifier ce comportement à l'aide de la variable `toplevel-music-handler` placée en tête.

- Du texte sous forme de *markup* comme les paroles d'un couplet

```
\markup {
   2.  Le première ligne du deuxième couplet.
}
```

De tels *markups* seront imprimés là où ils apparaissent, avant, après ou entre les expressions musicales.

- Une variable, ou identificateur, telle que

```
toto = { c4 d e d }
```

Vous pourrez la réutiliser plus loin dans votre fichier en saisissant simplement `\toto`. Le nom des indentificateurs ne doit être formés que de caractères alphabétiques – sans chiffre ni caractère souligné ou tiret.

Voici trois éléments que vous pouvez placer à un niveau supérieur :

```
\layout {
  % pas en pleine largeur
  ragged-right = ##t
}

\header {
   title = "Do-re-mi"
}

{ c'4 d' e2 }
```

Vous pouvez placer, n'importe où dans votre fichier, les instruction suivantes :

- `\version`
- `\include`
- `\sourcefilename`
- `\sourcefileline`
- Une ligne de commentaire, introduite par le signe `%`.
- Un bloc de commentaire, délimité par `%{ .. %}`.

Vous pouvez insérer des espaces dans votre fichier source afin de lui apporter une meilleure lisibilité. Les espaces superflus sont normalement ignorés. Notez cependant qu'il est des cas où l'espace est requis pour éviter tout risque d'erreur :

- Autour d'une accolade, qu'elle soit ouvrant ou fermante ;
- Après chaque commande ou variable, autrement dit tout élément qui commence par un \ ;
- Après tout élément qui sera interprété comme une expression Scheme, autrement dit tout élément qui commence par un `#` ;
- Pour séparer les éléments d'une expression Scheme ;
- En mode parole – `lyricmode` – pour séparer les termes des commandes `\override` et `\set`. Précisons à ce sujet qu'en plus d'ajouter une espace avant et après l'intégralité de la commande, vous devrez encadrer d'espace le point et le signe égal qu'elle peut contenir, comme dans `\override Score . LyricText #'font-size = #5`.

Voir aussi

Manuel d'initiation : Section "Organisation des fichiers LilyPond" dans *Manuel d'initiation*.

3.2 Titres et entêtes

La plupart de la musique qui est éditée comporte un titre et le nom de son compositeur ; certains ouvrages divulguent beaucoup plus d'information.

3.2.1 Création de titres

Des éléments de titrage peuvent s'attacher à un bloc `\score`, une partie – créée par un bloc `\bookpart` – ou bien un ouvrage entier créé par un bloc `\book`.

Les différents éléments du titrage sont contenus dans un bloc `\header`. Un ouvrage complet peut comporter :

`dedication`
: Le dédicataire de l'œuvre, centré en haut de la première page.

`title` Le titre de l'œuvre, centré en dessous de la dédicace.

`subtitle` Le sous-titre, centré sous le titre.

`subsubtitle`
: Un niveau supplémentaire de sous-titre, centré en dessous du sous-titre.

`poet` Le poète, parolier ou librétiste, aligné à gauche en dessous du deuxième sous-titre.

`instrument`
: L'instrument, centré en dessous du deuxième sous-titre. Il sera rappelé sur les pages suivant la première, centré sur la ligne d'entête.

`composer` Le compositeur, aligné à droite en dessous du deuxième sous-titre.

`meter` Le libellé du tempo, aligné à gauche sous le poète.

`arranger` L'arrangeur, aligné à droite sous le compositeur.

`piece` La pièce ou le mouvement, aligné à gauche sous le tempo.

`opus` L'opus ou le numéro au catalogue, aligné à droite sous l'arrangeur.

`breakbefore`
: Ce commutateur (activé par ##t ; ##f pour le désactiver) permet de forcer le saut de page avant d'afficher les titres.

`copyright`
: La notice de droits d'auteur, centrée en pied de la première page. Pour imprimer le symbole *copyright*, consultez la rubrique Section 3.3.3 [Codage du texte], page 399.

`tagline` Mention spéciale imprimée en pied de la dernière page.

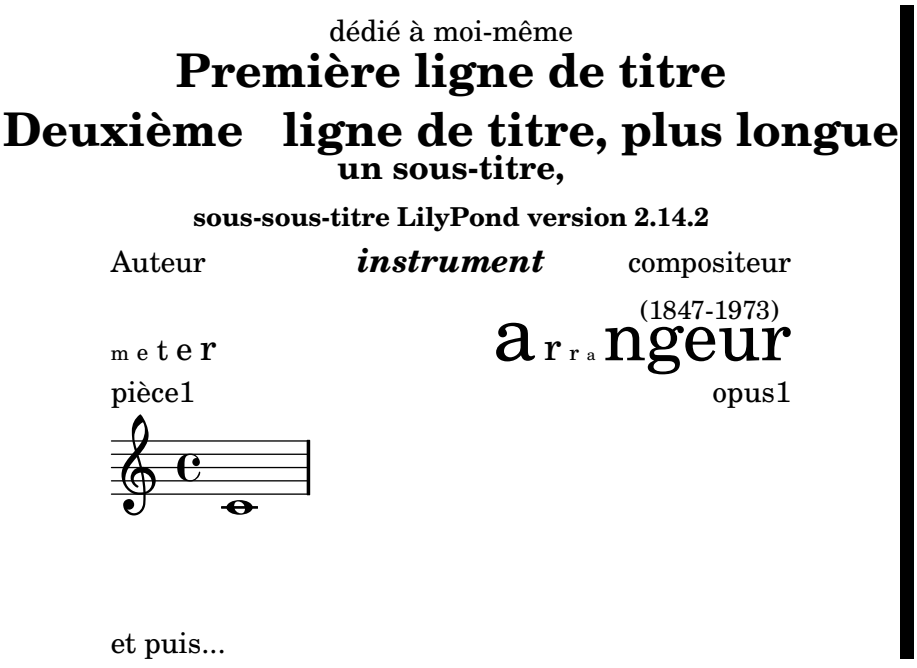
Voici ce que cela peut donner lorsque tout ces champs sont utilisés. Les différentes commandes mentionnées à la rubrique Section 1.8.2 [Mise en forme du texte], page 202 sont opérationnelles pour tous ces éléments.

```
\paper {
  line-width = 9.0\cm
  paper-height = 10.0\cm
}

\book {
  \header {
    dedication = "dédié à moi-même"
    title = \markup \center-column { "Première ligne de titre" "Deuxième
  ligne de titre, plus longue" }
```
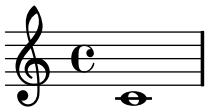
```
    subtitle = "un sous-titre,"
    subsubtitle = #(string-append "sous-sous-titre LilyPond version "
(lilypond-version))
    poet = "Auteur"
    composer =  \markup \center-column { "compositeur" \small "(1847-1973)" }
    texttranslator = "Traduit par"
    meter = \markup { \teeny "m" \tiny "e" \normalsize "t" \large "e" \huge
"r" }
    arranger = \markup { \fontsize #8.5 "a" \fontsize #2.5 "r" \fontsize
#-2.5 "r" \fontsize #-5.3 "a" \fontsize #7.5 "ngeur" }
    instrument = \markup \bold \italic "instrument"
    piece = "Pièce"
  }

  \score {
    { c'1 }
    \header {
      piece = "pièce1"
      opus = "opus1"
    }
  }
  \markup {
      et puis...
  }
  \score {
    { c'1 }
    \header {
      piece = "pièce2"
      opus = "opus2"
    }
  }
}
```



2 ***instrument***

pièce2 opus2



Music engraving by LilyPond 2.14.2—www.lilypond.org

Comme nous venons de la voir, vous pouvez multiplier les blocs `\header`. Lorsqu'un même champ apparaît dans plusieurs blocs, LilyPond utilisera le dernier mentionné :

```
\header {
  composer = "Compositeur"
}
\header {
  piece = "Morceau"
}
\score {
  \new Staff { c'4 }
  \header {
    piece = "Nouveau morceau"  % remplace le précédent
  }
}
```

Lorsque le bloc `\header` est défini à l'intérieur du bloc `\score`, seul les champs `piece` et `opus` seront imprimés. Attention cependant à toujours placer le bloc `\header` après l'expression musicale.

```
\score {
  { c'4 }
  \header {
    title = "title"  % not printed
    piece = "piece"
    opus = "opus"
  }
}
```

piece opus



Vous pouvez modifer ce comportement et imprimer tous les champs d'un bloc `\header` défini dans un bloc `\score` en ajoutant

```
\paper{
  print-all-headers = ##t
}
```

Les pieds de page sont vides, hormis pour la première page qui portera la champ `copyright` du `\header`, et pour la dernière page où apparaîtra le `tagline`. Celui-ci est par défaut « Music engraving by LilyPond (*version*) ».[1]

Un champ de titrage peut être désactivé :

```
\header {
  tagline = ##f
  composer = ##f
}
```

3.2.2 Titrages personnalisés

Vous pouvez personnaliser, dans le bloc `\paper`, la définition des variables listées ci-dessous. Leur mise en forme par défaut est inscrite dans le fichier d'initialisation '`../ly/titling-init.ly`'.

`bookTitleMarkup`
: Ce titre est apposé au début du document final. Il comprend normalement le compositeur et le titre de l'œuvre.

`scoreTitleMarkup`
: Ce titre est attaché à un bloc `\score`. Il comprend normalement le nom du mouvement – le champ `piece`.

`oddHeaderMarkup`
: L'entête des pages impaires.

`evenHeaderMarkup`
: L'entête des pages paires. Lorsqu'il n'est pas défini, LilyPond utilise celui des pages impaires.

  Les entêtes comportent par défaut le nom de l'instrument, au centre, et le numéro de page, sur le bord extérieur de la page.

`oddFooterMarkup`
: Le pied de page impaire.

`evenFooterMarkup`
: Le pied de page paire. Lorsqu'il n'est pas défini, LilyPond utilise celui des pages impaires.

  Par défaut, le pied de la première page comporte la notice de droits d'auteur, et celui de la dernière page la « mention spéciale » – le `tagline`.

La définition suivante permt d'obtenir, sur la même ligne, le titre aligné à gauche et le compositeur aligné à droite.

```
\paper {
  bookTitleMarkup = \markup {
   \fill-line {
     \fromproperty #'header:title
     \fromproperty #'header:composer
   }
  }
}
```

[1] Afin que l'on sache qui est capable d'une telle qualité, nous vous saurons gré de ne pas suprimer cette mention spéciale, autant que faire se peut.

Les entêtes et pieds de page sont créés respectivement par les fonctions `make-header` and `make-footer`, que vous pouvez définir au sein du bloc `\paper`. Leur implémentation est contenue dans les fichiers d'initialisation '`ly/paper-defaults-init.ly`' et '`ly/titling-init.ly`'.

Voici comment centrer le numéro en pied de chaque page :

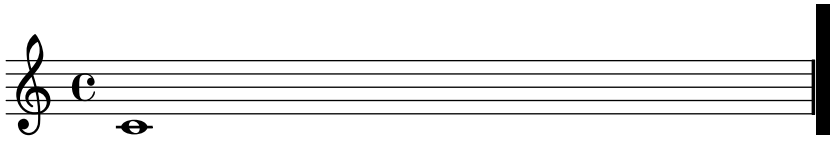
```
\paper {
  print-page-number = ##t
  print-first-page-number = ##t
  oddHeaderMarkup = \markup \fill-line { " " }
  evenHeaderMarkup = \markup \fill-line { " " }
  oddFooterMarkup = \markup {
    \fill-line {
      \bold \fontsize #3
      \on-the-fly #print-page-number-check-first
      \fromproperty #'page:page-number-string
    }
  }
  evenFooterMarkup = \markup {
    \fill-line {
      \bold \fontsize #3
      \on-the-fly #print-page-number-check-first
      \fromproperty #'page:page-number-string
    }
  }
}
```

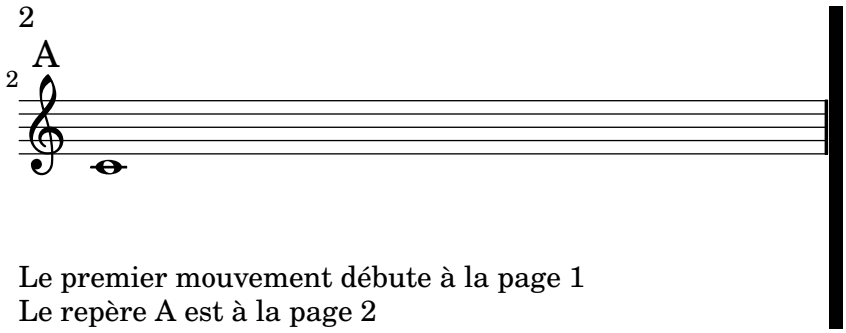
3.2.3 Référencement des numéros de page

LilyPond vous permet, à l'aide de la commande `\label`, d'insérer des points de référence dans un ouvrage, aussi bien en dehors qu'au fil de la musique. Ce point de référence pourra être ensuite repris à l'intérieur d'un *markup* ; vous pourrez même y ajouter le numéro de page grâce à la commande de *markup* `\page-ref`.

```
\header { tagline = ##f }
\book {
  \label #'firstScore
  \score {
    {
      c'1
      \pageBreak \mark A \label #'markA
      c'1
    }
  }

  \markup { Le premier mouvement débute à la page \page-ref #'firstScore "0" "?" }
  \markup { Le repère A est à la page \page-ref #'markA "0" "?" }
}
```





Le premier mouvement débute à la page 1
Le repère A est à la page 2

L'instruction `\page-ref` prend trois arguments :

1. le point de référence, sous la forme d'un symbole scheme, comme par exemple `#'firstScore`,
2. un « emporte-pièce » afin d'estimer la longueur totale du *markup*,
3. un texte de remplacement au cas où la référence ne serait pas retrouvée.

La présence de l'emporte-pièce est rendue nécessaire par le fait que les *markups* sont générés avant que les sauts de page ne soient positionnés. Bien que le numéro de page en question ne soit pas encore déterminé, LilyPond doit connaître les dimensions de ce *markup*. Vous pouvez, lorsque l'ouvrage contiendra plus de dix pages, stipuler un emporte-pièce sur deux caractères – soit `"00"`.

Commandes prédéfinies

`\label`, `\page-ref`.

3.2.4 Table des matières

La commande `\markuplines \table-of-contents` vous permettra de générer une table des matières. Les éléments qui la composeront sont créés par la commande `\tocItem`, insérée indépendamment ou au sein d'une expression musicale.

```
\markuplines \table-of-contents
\pageBreak

\tocItem \markup "Premier mouvement"
\score {
  {
    c'4  % ...
    \tocItem \markup "Passage spécifique du premier mouvement"
    d'4  % ...
  }
}

\tocItem \markup "Second mouvement"
\score {
  {
    e'4 % ...
  }
}
```

Les *markups* dévolus à la mise en forme de la table des matières se définissent dans le bloc `\paper`. Il s'agit par défaut de `tocTitleMarkup` pour le titre de la table, et de `tocItemMarkup` pour ses éléments – composés de leur libellé et numéro de page. Vous pouvez bien entendu personnaliser ces variables :

```
\paper {
  %% Translate the toc title into French:
  tocTitleMarkup = \markup \huge \column {
```

```
    \fill-line { \null "Table des matières" \null }
    \hspace #1
  }
  %% use larger font size
  tocItemMarkup = \markup \large \fill-line {
    \fromproperty #'toc:text \fromproperty #'toc:page
  }
}
```

Notez bien la manière de référencer le libellé et le numéro de page dans la définition de `tocItemMarkup`.

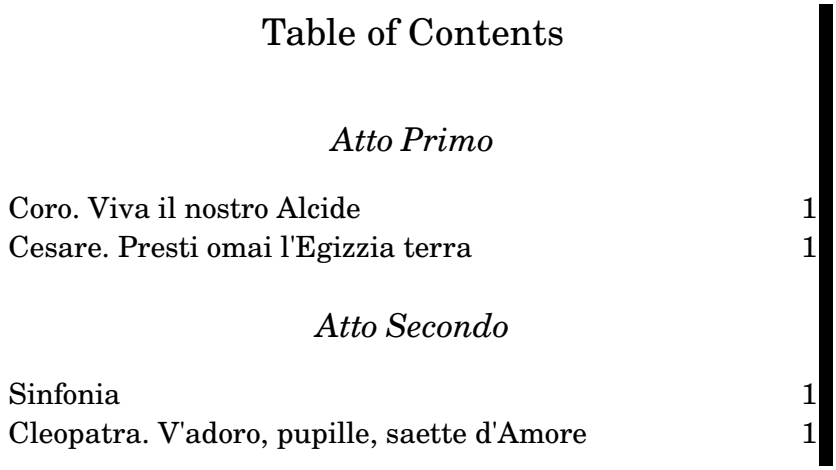
N'hésitez pas à définir vous-même d'autres commandes et *markups* afin de construire une table plus élaborée :

- commencez par définir une nouvelle variable de type `markup` au sein du bloc `\paper`,
- puis définissez une fonction musicale chargée d'insérer un élément de la table à partir de cette variable.

Dans l'exemple qui suit, nous avons créé un nouveau style d'élément dans le but de mentionner les actes dans la table des matières d'un opéra :

```
\paper {
  tocActMarkup = \markup \large \column {
    \hspace #1
    \fill-line { \null \italic \fromproperty #'toc:text \null }
    \hspace #1
  }
}

tocAct =
#(define-music-function (parser location text) (markup?)
   (add-toc-item! 'tocActMarkup text))
```



L'élément et son numéro de page peuvent se rejoindre par une ligne pointillée :

```
\header { tagline = ##f }
\paper {
  tocItemMarkup = \tocItemWithDotsMarkup
}

\book {
```

```
  \markuplines \table-of-contents
  \tocItem \markup { Allegro }
  \tocItem \markup { Largo }
  \markup \null
}
```

Table of Contents

Allegro 1
Largo 1

Voir aussi

Fichiers d'initialisation : '`../ly/toc-init.ly`'.

Commandes prédéfinies

`\table-of-contents`, `\tocItem`.

3.3 Travail sur des fichiers texte

3.3.1 Insertion de fichiers LilyPond

Lorsqu'un projet prend de l'importance en volume, il est judicieux de le scinder en plusieurs fichiers, auxquels vous fereez référence avec un simple

```
\include "autrefichier.ly"
```

Une ligne `\include "autrefichier.ly"` dans un fichier revient à recopier intégralement le contenu de '`autrefichier.ly`' à l'endroit même ou est placée l'instruction `\include`. Vous pouvez par exemple écrire un fichier individuel par instrument, puis les regrouper pour former le fichier « conducteur ». Les différentes variables définies dans les fichiers séparés seront normalement reprises et utilisables dans le fichier formant le conducteur. Les sections balisées dans les fichiers individuels peuvent être réutilisées en différents endroit de la partition , comme expliqué à la rubrique Section 3.3.2 [Différentes éditions à partir d'une même source], page 394.

Lorsque le fichier auquel il est fait référence se trouve dans le même répertoire, donner seulement son nom en argument à la commande `\include` suffit. S'il se trouve ailleurs, vous devrez indiquer le chemin d'accès, absolu ou relatif, en respectant toutefois la syntaxe UNIX – autrement dit, le séparateur de répertoire est une oblique normale `/` et non l'oblique inverse `\` de DOS/Windows. Par exemple, si le fichier '`truc.ly`' se trouve dans le répertoire supérieur au répertoire de travail, la ligne devra être

```
\include "../stuff.ly"
```

ou bien, si les fichiers correspondant aux parties d'orchestre se trouvent dans le sous-répertoire '`parties`' relativement au répertoire courant, vous devrez mentionner

```
\include "parties/VI.ly"
\include "parties/VII.ly"
... etc.
```

Les fichiers à inclure peuvent eux-même contenir des instructions `\include`. Ces instructions `\include` de second niveau ne pourront, par défaut, être interprétées qu'une fois intégrées dans le fichier principal ; leur argument doit donc comporter le chemin relativement au fichier principal et non par rapport au fichier dan lesquel cette inclusion est mentionnée. Vous pouvez toutefois influer sur ce comportement à l'aide de l'option `-drelative-includes` en ligne de commande ou en ajoutant une clause `#(ly:set-option 'relative-includes #t)` en tête du fichier principal.

Une fois `relative-include` activé, le chemin à suivre pour chacune des commandes `\include` sera pris relativement au fichier qui la contient.

Vous pouvez inclure des fichiers dont vous spécifierez le chemin d'accès sur la ligne de commande au moment de lancer la compilation. L'appel à ces fichiers ne mentionnerra alors que leur nom. Par exemple, si vous voulez compiler avec cette méthode le fichier '`principal.ly`' qui inclut des fichiers situés dans le sous-répertoire '`parties`', placez vous dans le répertoire contenant '`principal.ly`', puis tapez

```
lilypond --include=parties principal.ly
```

tout en ayant bien dans '`principal.ly`'

```
\include "VI.ly"
\include "VII.ly"
... etc
```

Lorsqu'un fichier est voué à être inclus dans nombre de partitions, vous pouvez le placer dans le répertoire de LilyPond '`../ly`'. Attention : ce répertoire varie selon votre installation, comme indiqué au chapitre Section "Autres sources de documentation" dans *Manuel d'initiation*. Ce fichier sera inclus dès lors que vous fournirez uniquement son nom en argument à la fonction `\include`. C'est par exemple le cas du fichier de définition particulier '`gregorian.ly`'.

Au moment où vous lancez LilyPond, un certain nombre de fichiers se retrouvent inclus par défaut ; il suffit d'activer le mode verbeux en faisant `lilypond --verbose` pour s'en rendre compte. Vous verrez ainsi défiler, en plus de nombreuses informations, le nom d'un certain nombre de fichier et de chemins d'accès. Les fichiers les plus important sont mentionnés au chapitre Section "Autres sources de documentation" dans *Manuel d'initiation*. Si vous venez à les modifier, rappelez-vous qu'ils seront écrasés à l'installation d'une nouvelle version de LilyPond.

Vous trouverez quelques exemples simples d'utilisation de la commande `\include` au chapitre Section "Conducteurs et parties" dans *Manuel d'initiation*.

Voir aussi

Manuel d'initiation : Section "Autres sources de documentation" dans *Manuel d'initiation*, Section "Conducteurs et parties" dans *Manuel d'initiation*.

Problèmes connus et avertissements

Lorsque vous incluez un fichier qui porte le même nom que l'un des fichiers d'initialisation de LilyPond, le fichier de la distribution de LilyPond aura préséance.
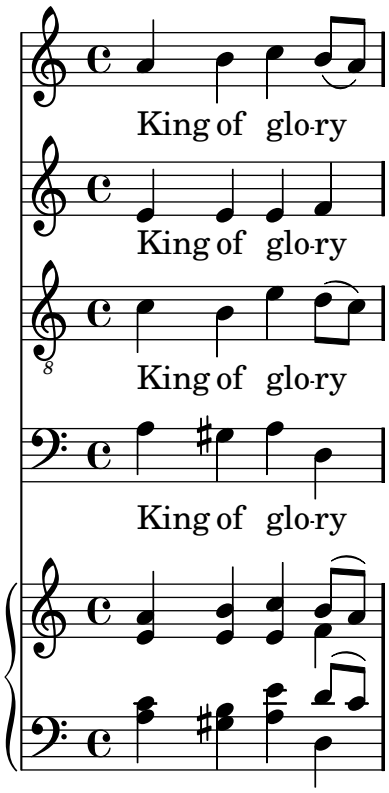
3.3.2 Différentes éditions à partir d'une même source

Différents mécanismes permettent de générer plusieurs versions d'une partition à partir d'une même source. Les variables – ou idendificateurs – sont sûrement le moyen le plus simple de combiner de différente manière des passages relativement longs, alors que les balises permettront de sélectionner de courts fragments selon leur utilisation. Quelle que soit la méthode utilisée, séparer la notation de la structure de la partition vous donnera plus de liberté dans l'agencement de l'ouvrage final, puisque vous ne reviendrez pas sur la musique qui le compose.

Utilisation de variables

Un fragment musical identifié par une variable est réutilisable à divers endroits de la partition, comme nous l'avons vu à la rubrique Section "Organisation du code source avec des variables" dans *Manuel d'initiation*. Par exemple, une partition pour chœur *a cappella* comporte souvent une réduction pour piano reprenant toutes les voix ; il s'agit de la même musique, et vous ne devrez donc la saisir qu'une seule fois. D'autre part, la musique issue de deux variables peut se combiner sur une seule portée, comme nous l'avons vu à la rubrique [Regroupement automatique de parties], page 148. Prenons l'exemple suivant :

```
sopranoMusic = \relative c'' { a4 b c b8( a) }
altoMusic = \relative g' { e4 e e f }
tenorMusic = \relative c' { c4 b e d8( c) }
bassMusic = \relative c' { a4 gis a d, }
allLyrics = \lyricmode {King of glo -- ry }
<<
  \new Staff = "Soprano" \sopranoMusic
  \new Lyrics \allLyrics
  \new Staff = "Alto" \altoMusic
  \new Lyrics \allLyrics
  \new Staff = "Tenor" {
    \clef "treble_8"
    \tenorMusic
  }
  \new Lyrics \allLyrics
  \new Staff = "Bass" {
    \clef "bass"
    \bassMusic
  }
  \new Lyrics \allLyrics
  \new PianoStaff <<
    \new Staff = "RH" {
      \set Staff.printPartCombineTexts = ##f
      \partcombine
      \sopranoMusic
      \altoMusic
    }
    \new Staff = "LH" {
      \set Staff.printPartCombineTexts = ##f
      \clef "bass"
      \partcombine
      \tenorMusic
      \bassMusic
    }
  >>
>>
```



Générer une partition chorale ou la réduction pour piano ne requiert que de modifier la structure des éléments, sans aucunement toucher à la musique.

Dans le cas d'une partition relativement longue, vous pouvez isoler la définition des différentes variables dans des fichiers séparés que vous rappellerez ensuite, comme indiqué à la rubrique Section 3.3.1 [Insertion de fichiers LilyPond], page 393.

Utilisation de balises

La commande `\tag #'partieA` affecte à une expression musicale le nom *partieA*. Les expressions ainsi balisées pourront être filtrées par la suite, à l'aide de `\keepWithTag #'nom` ou `\removeWithTag #'nom`. Ces filtres fonctionnent de la manière suivante :

Filtre	**Résultat**
Musique balisée précédée de `\keepWithTag #'nom`	Musique non balisée et musique balisée par *nom* seront incluses ; la musique balisée autrement est exclue.
Musique balisée précédée de `\removeWithTag #'nom`	Musique non balisée et fragments appelés autrement que *nom* seront inclus ; la musique balisée par *nom* est exclue.
Musiqeu balisée non précédée de `\keepWithTag` ou `\removeWithTag`	Musique balisée et non balisée seront incluses.

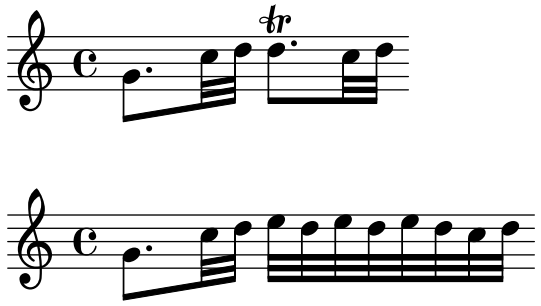
Les arguments des commandes `\tag`, `\keepWithTag` et `\removeWithTag` doivent être un symbole (tel que `#'conducteur` ou `#'partie`), suivi d'une expression musicale.

Dans l'exemple qui suit, nous obtenons deux versions du même extrait, l'une pour le conducteur, l'autre pour l'instrumentiste, et qui comportera les ornements développés.

```
music = \relative g' {
  g8. c32 d
  \tag #'trills { d8.\trill }
```

```
  \tag #'expand { \repeat unfold 3 { e32 d } }
  c32 d
 }

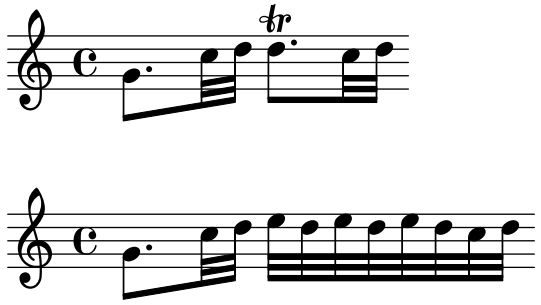
\score {
  \keepWithTag #'trills \music
}
\score {
  \keepWithTag #'expand \music
}
```



Il est parfois plus aisé d'exclure des fragments :

```
music = \relative g' {
  g8. c32 d
  \tag #'trills { d8.\trill }
  \tag #'expand {\repeat unfold 3 { e32 d } }
  c32 d
 }

\score {
  \removeWithTag #'expand
  \music
}
\score {
  \removeWithTag #'trills
  \music
}
```



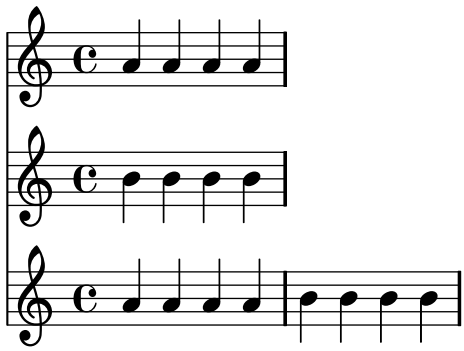
Ce principe de filtrage peut s'appliquer aux articulations, textes, etc. Il suffit de positionner `-\tag #ma-balise` avant l'articulation ou le texte, comme ici :

```
c1-\tag #'doigt ^4
c1-\tag #'gaffe ^"Attention !"
```

Ceci définira une note avec une indication conditionnelle de doigté ou un texte.

Vous pouvez baliser différemment la même expression musicale en saisissant plusieurs `\tag` :

```
music = \relative c'' {
  \tag #'a \tag #'both { a4 a a a }
  \tag #'b \tag #'both { b4 b b b }
}
<<
\keepWithTag #'a \music
\keepWithTag #'b \music
\keepWithTag #'both \music
>>
```



L'application concomitante de plusieurs filtres `\removeWithTag` à la même expression musicale permet d'exclure plusieurs balisages :

```
music = \relative c'' {
\tag #'A { a4 a a a }
\tag #'B { b4 b b b }
\tag #'C { c4 c c c }
\tag #'D { d4 d d d }
}
{
\removeWithTag #'B
\removeWithTag #'C
\music
}
```



L'application de plus d'un filtre `\keepWithTag` à la même expression musicale aboutit à l'exclusion de **tous** les balisages. En effet, si le premier filtre exclut tous les autres balisages, l'application du second exclura les effets du premier.

Voir aussi

Manuel d'initiation : Section "Organisation du code source avec des variables" dans *Manuel d'initiation*.

Manuel de notation : [Regroupement automatique de parties], page 148, Section 3.3.1 [Insertion de fichiers LilyPond], page 393.

Globalisation des réglages

Vous pouvez regrouper dans un fichier indépendant vos réglages personnels que vous incluerez au besoin :

```
lilypond -dinclude-settings=MES_REGLAGES.ly MA_PARTITION.ly
```

Vous pouvez ainsi stocker dans un fichier séparé vos réglages en matière de format de papier, de fontes utilisées ou vos définitions particulières. Selon le fichier de réglages que vous mentionnerez, vous obtiendrez facilement différentes éditions à partir d'une même source quelle qu'elle soit.

Ceete technique peut s'utiliser en combinaison avec des feuilles de styles, comme indiqué au chapitre Section "Feuilles de style" dans *Manuel d'initiation*.

Voir aussi

Manuel d'initiation : Section "Organisation du code source avec des variables" dans *Manuel d'initiation*, Section "Feuilles de style" dans *Manuel d'initiation*.

Manuel de notation : Section 3.3.1 [Insertion de fichiers LilyPond], page 393.

3.3.3 Codage du texte

LilyPond utilise le jeu de caractères défini par le consortium Unicode et la norme ISO/CEI 10646. Chaque caractère est identifié par un nom unique et associé à un point de code, ce qui permet dans l'absolu de couvrir tous les langages. Unicode permet de coder tous les caractères utilisés par toutes les langues écrites du monde. LilyPond utilise le codage UTF-8 (UTF pour *Unicode Transformation Format*) qui permet de représenter les caractères latins sur un octet et les autres sur une longueur allant jusqu'à quatre octets.
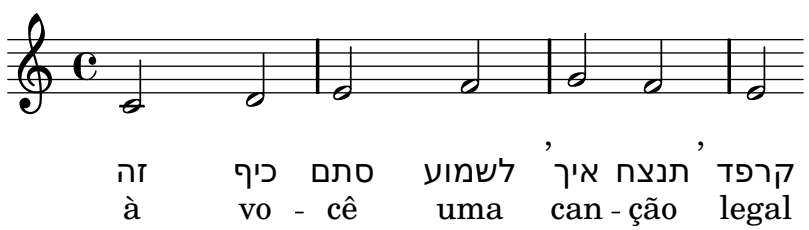
L'apparence réelle des caractères est déterminée par les glyphes ou graphèmes tels que définis dans les différentes polices disponibles. Une police, ou une fonte, définit la mise en correspondance d'un sous-ensemble de points de code unicode en glyphes. LilyPond recourt à la bibliothèque Pango pour assurer le rendu des textes multilingues.

LilyPond n'effectue aucune conversion d'encodage que ce soit. Ceci implique donc que tout texte, qu'il s'agisse d'un titre, de paroles ou même d'instruction musicale, comportant des caractères non ASCII soit codé en UTF-8. Le plus sûr moyen de saisir du texte de la sorte consiste à utiliser un éditeur supportant l'unicode et à enregistrer vos fichier en UTF-8. C'est le cas pour la plupart des éditeurs actuels, que ce soit vim, Emacs, jEdit et GEdit. Tous les systèmes Windows postérieurs à NT utilisent Unicode en natif ; même Notepad est capable d'éditer et sauvegarder un fichie en UTF-8 – sans parler de l'excellente alternative qu'est BabelPad.

La compilation d'un fichier LilyPond comportant des caractères non ASCII qui n'aurait pas été enregistré dans l'encodage UTF-8 vous renverra l'erreur

```
FT_Get_Glyph_Name () erreur : invalid argument
```

Voici un exemple utilisant du texte en cyrilique, en hébreux et en portugais.



Lorsque vous avez besoin d'un caractère dont vous connaissez le point de code mais que votre éditeur ne permet pas de saisir directement, vous pouvez utitliser les instructions `\char ##xhhhh` ou `\char #dddd` au sein d'un bloc `\markup` – `hhhh` et `dddd` correspondant respectivement à la

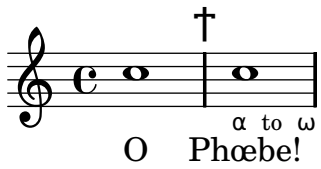
valeur hexadécimale ou décimale. Même s'il est inutile de saisir les zéros superflus, il est de bon ton de stipuler les quatre caractères formant la représentation hexadécimale. Évitez cependant l'encodage UTF-8 d'un point de code après un `\char` ; les encodages UTF-8 comprennent un bit supplémentaire indiquant le nombre d'octets. Une table de correspondance entre les codes Unicode et le nom des caractères ainsi que leur code hexadécimal est disponible sur le site du consortium Unicode, http://www.unicode.org/.

Par exemple, `\char ##x03BE` et `\char #958` correspondent tous deux au caractère unicode U+03BE, dénommé « Greek Small Letter Xi ».

Quel que soit le point de code spécifié de cette manière, il ne vous sera alors pas nécessaire d'enregistrer votre fichier en UTF-8. Vous devrez toutefois disposer d'une fonte contenant ce caractère qui soit accessible à LilyPond.

L'exemple suivant illustre la manière d'insérer un caractère sous sa forme hexadécimale, à la fois dans un repère, dans une articulation, dans des paroles et dans du texte indépendant.

```
\score {
  \relative c'' {
    c1 \mark \markup { \char ##x03EE }
    c1_\markup { \tiny { \char ##x03B1 " to " \char ##x03C9 } }
  }
  \addlyrics { O \markup { \concat { Ph \char ##x0153 be! } } }
}
\markup { "Copyright 2008--2011" \char ##x00A9 }
```



Copyright 2008--2011 ©

Le signe *copyright* dans le champ de titrage consacré s'inscrit de la manière suivante :

```
\header {
  copyright = \markup { \char ##x00A9 "2008" }
}
```

3.3.4 Affichage de notation au format LilyPond

La fonction musicale `\displayLilyMusic`, disponible uniquement pour un traitement en ligne de commande, permet d'afficher en notation LilyPond une expression musicale. Par exemple,

```
{
  \displayLilyMusic \transpose c a, { c4 e g a bes }
}
```

affichera

```
{ a,4 cis e fis g }
```

LilyPond affichera le résultat sous forme de message en console, au milieu de toutes les informations de compilation. Afin d'isoler ces messages et enregistrer le résultat de la fonction `\display{STUFF}`, pensez à rediriger la sortie vers un fichier.

```
lilypond fichier.ly >display.txt
```

3.4 Contrôle des sorties

3.4.1 Extraction de fragments musicaux

LilyPond vous permet d'extraire des fragments d'une partition à l'instar du choriste amateur qui alimente son album avec des coupures de journaux.

Vous devrez pour cela définir les mesures à découper. La définition suivante, incluse dans votre fichier source,

```
\layout {
  clip-regions
  = #(list
      (cons
       (make-rhythmic-location 5 1 2)
       (make-rhythmic-location 7 3 4)))
}
```

vous permettra d'extraire un fragment compris entre le milieu de la cinquième mesure et quelque part dans la septième. Le triplet `5 1 2` signifie : après la durée d'une blanche dans la mesure 5 ; le `7 3 4` signifie quant à lui que l'on s'arrête à la mesure 7, après la durée de trois noires.

Rien ne vous empêche d'extraire plusieurs fragments, dès lors que vous définissez dans la liste d'autres paires d'emplacements ryhtmiques.

Cette fonctionnalité n'est toutefois effective que lorsque vous lancez LilyPond avec l'option `-dclip-systems`. Les « coupures » seront générées sous la forme de fichiers EPS, convertis en PDF ou PNG selon le format que vous aurez stipulé.

Pour de plus amples informations quant au format des résultats, consultez le chapitre Section "Lancement de lilypond" dans *Utilisation des programmes*.

3.4.2 Ignorer des passages de la partition

Dans un travail de transcription ou de recopie de la musique, ce qui vous intéresse plus particulièrement se situe à la fin, là même où vous en êtes dans la notation. Dans le but de gagner du temps dans le processus de correction, vous pouvez « escamoter » le reste et ne générer que les dernières mesures en insérant

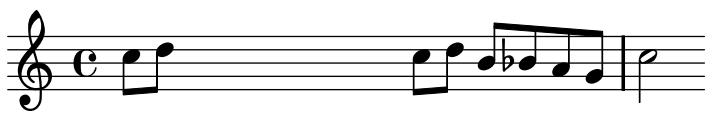
```
showLastLength = R1*5
\score { ... }
```

dans votre fichier source. Ceci aura pour effet de ne générer que les cinq dernières mesures – si tant est que le morceau soit à 4/4 – de tous les `\score` de votre fichier. Dans le cas d'un œuvre conséquente, cette pratique s'avère fort utile puisqu'elle évite de tout générer. Vous pourriez aussi être amené à retravailler le début d'une œuvre, pour y ajouter une partie par exemple, auquel cas c'est la propriété `showFirstLength` que vous utiliserez.

Vous pouvez contrôler très finement les parties à escamoter, grâce u commutateur `Score.skipTypesetting` : lorsqu'il est activé, aucune gravure n'est réalisée.

Ce commutateur agit aussi sur la sortie MIDI. Notez bien que tous les événements seront escamotés, y compris les changements de tempo ou d'instrument – vous voilà prévenu !

```
c8 d
\set Score.skipTypesetting = ##t
e8 e e e e e e e
\set Score.skipTypesetting = ##f
c8 d b bes a g c2
```



Dans le cadre de musique polyphonique, `Score.skipTypesetting` s'applique à toutes les voix et portées. Vous gagnerez donc encore plus de temps.

3.4.3 Formats de sortie alternatifs
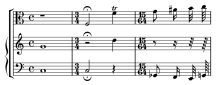
En matière de partition imprimable, LilyPond génère par défaut des documents au format PstScript (PS) et Portable Document Format (PDF). Vous pouvez aussi obtenir des documents au format Scalable Vector Graphics (SVG), Encapsulated PostScript (EPS) ou Portable Network Graphics (PNG) dès lors que vous aurez lancé LilyPond en ligne de commande avec l'option *ad hoc* – voir Section "Options en ligne de commande pour lilypond" dans *Utilisation des programmes* à ce sujet.

3.4.4 Changement des fontes musicales
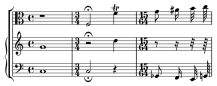
Gonville est ube alternative à la fonte Feta que LilyPond utilise par défaut. Vous pouvez la télécharger à partir de

`http://www.chiark.greenend.org.uk/~sgtatham/gonville/`

Voici quelques mesures utilisant la police Gonville :



Et ces même mesures avec la police de LilyPond, Feta :



Instructions d'installation pour MacOS

Téléchargez puis décompressez l'archive zip. Recopiez le répertoire `lilyfonts` dans '*SHARE_DIR*`/lilypond/current`' – voir Section "Autres sources de documentation" dans *Manuel d'initiation* à ce sujet. Renommez le répertoire `fonts` qui s'y trouve en `fonts_orig`, puis les répertoire `lilyfonts` en `fonts`. Il vous suffira, pour retrouver la fonte Feta, de renommer `fonts_orig` en `fonts`.

Voir aussi

Manuel d'initiation : Section "Autres sources de documentation" dans *Manuel d'initiation*.

Problèmes connus et avertissements

Gonville ne permet pas de générer de la notation ancienne. Consultez le site de l'auteur pour de plus amples informations ainsi qu'à propos des conditions d'utilisation.

3.5 Sortie MIDI

MIDI (Musical Instrument Digital Interface) constitue un standard en matière de connexion et de contrôle des instruments électroniques. Un fichier MIDI contient une série de notes réparties dans différentes pistes. Il ne s'agit en rien d'un fichier sonore ; il vous faudra un logiciel capable de traduire ces séries de notes en sons.

Vous pouvez convertir vos partition en fichier MIDI de manière à entendre ce que vous avez saisi. Ceci vous permet de contrôler aisément ce que vous avez saisi : octaves et altérations erronées heuteront votre oreille avertie grâce au MIDI.

Dans une sortie MIDI, LilyPond alloue un canal à chaque portée, plus un pour les réglages globaux. Par voie de conséquence, un fichier MIDI ne peut comporter au maximmum que quinze portées (quatorze en l'absence de percussions) ; les portées supplémentaires seront tout simplement silencieuses.

3.5.1 Création de fichiers MIDI

LilyPond génèrera un fichier MIDI dès que vous ajouterez un bloc `\midi` à la structure de votre partition, comme ici :

```
\score {
  ...musique...
  \midi { }
}
```

Lorsque le bloc `\score` contient un bloc `\midi` mais pas de bloc `\layout`, LilyPond ne produira qu'une sortie MIDI. Si donc vous avez besoin aussi d'un support visuel, vous devrez aussi mentionner un bloc `\layout`.

```
\score {
  ...musique...
  \midi { }
  \layout { }
}
```

Hauteurs, durées, liaisons de prolongation, nuances et changements de tempo seront interprétés et traduits en événements MIDI. Les indications de nuances, crescendos et decrescendos sont traduits en niveau de volume ; les indications sous la forme d'une fraction déterminée du volume disponible, et crescendos et decrescendos sous la forme d'une progression linéaire entre les deux extrêmes. Le rendu des indications de nuance peut être désactivé pour le MIDI – voir Section 3.5.2 [Le bloc MIDI], page 405.

Le tempo initial ainsi que ses changements sont normalement indiqués au fil de la notation à l'aide de la commande `\tempo` ; ils seront retranscrits dans le fichier MIDI. La commande `\tempo` donne lieu à l'impression d'une indication métronomique que vous pouvez toutefois rendre invisible, comme indiqué à la rubrique [Indication métronomique], page 59. Une autre manière de spécifier le tempo initial pour un fichier MIDI est indiquée plus avant – voir Section 3.5.2 [Le bloc MIDI], page 405.

En raison de certaines limitations de Windows, les fichiers MIDI doivent y porter l'extension `.mid`. D'autres systèmes utilisent l'extension `.midi`. Si besoin est, placez la ligne suivante au début de votre fichier source, avant l'ouverture de tout bloc `\book`, `\bookpart` ou `\score` :

```
#(ly:set-option 'midi-extension "midi")
```

Cette ligne déterminera `.midi` comme extension par défaut pour les fichiers MIDI.

Vous pouvez aussi le faire en ligne de commande :

```
lilypond ... -dmidi-extension=midi fichierLily.ly
```

Noms d'instrument

L'instrument MIDI affecté à un canal particulier est déterminé par la propiété `Staff.midiInstrument`. Vous pouvez choisr l'un des instruments répertoriés à l'annexe Section A.4 [Instruments MIDI], page 489.

```
\new Staff {
  \set Staff.midiInstrument = #"glockenspiel"
  ...notes...
}

\new Staff \with {midiInstrument = #"cello"} {
  ...notes...
}
```

Lorsque l'instrument choisi ne correspond pas exactement à l'une des dénominations consacrées, LilyPond le replacera par un piano de concert (`"acoustic grand"`).
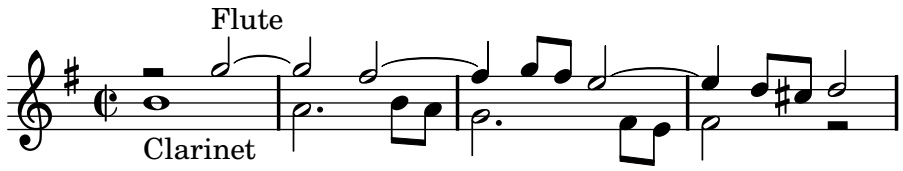
Morceaux choisis

Affectation d'un canal MIDI par voix

Lorsque LilyPond génère un fichier MIDI, chaque portée sera par défaut affectée à un canal, quel que soit le nombre de voix qu'elle contient. Ceci permet d'éviter de se retrouver à court de canaux, sachant qu'il n'y en a que seize de disponibles.

Le fait de déplacer le `Staff_performer` dans le contexte `Voice` permet d'affecter à chaque voix d'une même portée un canal MIDI spécifique. Dans l'exemple suivant, la même portée donnera lieu à deux canaux MIDI différents, chacun étant affecté de son propre `midiInstrument`.

```
\score {
  \new Staff <<
    \new Voice \relative c''' {
      \set midiInstrument = #"flute"
      \voiceOne
      \key g \major
      \time 2/2
      r2 g-"Flute" ~
      g fis ~
      fis4 g8 fis e2 ~
      e4 d8 cis d2
    }
    \new Voice \relative c'' {
      \set midiInstrument = #"clarinet"
      \voiceTwo
      b1-"Clarinet"
      a2. b8 a
      g2. fis8 e
      fis2 r
    }
  >>
  \layout { }
  \midi {
    \context {
      \Staff
      \remove "Staff_performer"
    }
    \context {
      \Voice
      \consists "Staff_performer"
    }
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 2)
    }
  }
}
```



Problèmes connus et avertissements

Un changement de volume ne peut intervenir que sur le démarrage d'une note. C'est la raison pour laquelle la succession d'un crescendo et d'un diminuendo ne peut se produire sur une même note.

Certains lecteurs MIDI ne rendent pas correctement les changements de tempo. MS Windows Media Player et timidity le font sans problème.

3.5.2 Le bloc MIDI

Dès lors que vous désirez obtenir une sortie MIDI, vous devrez inscrire un bloc `\midi` au sein du bloc `\score`. Son fonctionnement est comparable à ce lui du bloc `\layout`, voire plus simple. Si le bloc `\midi` est la plupart du temps laissé vide, il peut contenir des aménagements pour certains contextes, la définition de contextes particuliers ou du code permettant de déterminer la valeur de certianes propriétés. L'exemple suivant détermine le tempo initial du fichier MIDI tout en se passant de son indication sur la partition imprimée.

```
\score {
  ...musique...
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 4)
    }
  }
}
```

Ici, le tempo est fixé à 72 noires à la minute. Spécifier un tempo de la sorte ne permet pas de doner une valeur pour une note pointée. Vous devrez, en pareil cas, subdiviser la note pointée en durée plus courte. Par exemple, indiquer 90 à la noire pointée est équivalent à spécifier 270 croches à la minute :

```
tempoWholesPerMinute = #(ly:make-moment 270 8)
```

La syntaxe permettant de définir un contexte pour le `\midi` est en tout point identique à celle que vous utilisez dans le bloc `\layout`, à ceci près que le « graveur » est remplacé par un « interprète ». Les différents contextes disponibles en matière de MIDI sont répertoriés dans le fichier d'initialisation '`../ly/performer-init.ly`' – pour plus de détail, voir Section "Autres sources de documentation" dans *Manuel d'initiation*. Si vous voulez vous passer des nuances dans votre fichier MIDI, il vous suffit d'insérer les lignes suivantes dans votre bloc `\midi{ }`.

```
\midi {
  ...
  \context {
    \Voice
    \remove "Dynamic_performer"
  }
}
```

LilyPond ne génèrera de sortie MIDI que si vous incluez un bloc `\midi` dans la structure de la partition, initialisée par la commande `\score`.

```
\score {
  { ...notes... }
  \midi { }
}
```

3.5.3 Contenu de la sortie MIDI

Éléments pris en compte dans le MIDI

Un fichier MIDI généré par LilyPond comprendra les éléments de notation suivants :

- les hauteurs ;
- les microtonalités – voir [Altérations], page 5. Leur rendu nécessite cependant un lecteur qui prenne en charge la modulation ;
- les accords nommés ;
- le rythme en tant que durée de note, y compris les nolets ;
- les tremolos, exceptés ceux utilisant la syntaxe « `:`[*nombre*] » ;
- les liaisons de prolongation ;
- les indications de nuance
- les crescendos et decrescendos s'étalant sur plusieurs notes ;
- les changements de tempo indiqués par un `\tempo` ;
- les paroles.

Éléments non pris en compte dans le MIDI
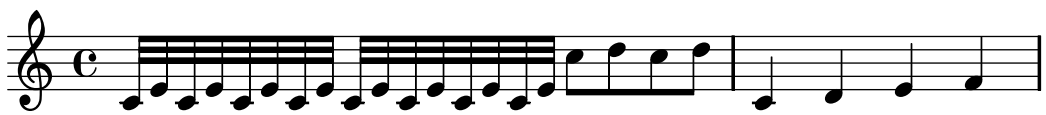
LilyPond ne peut générer d'événement MIDI pour les éléments suivant :
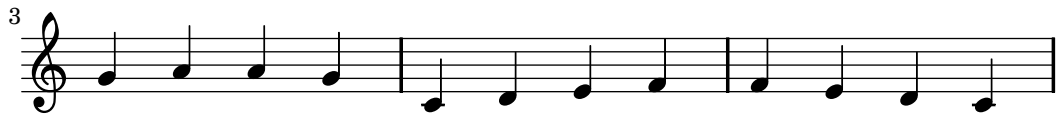
- le rythme indiqué sous forme d'annotation (p.ex. *swing*) ;
- les changements de tempo indiqués sous forme d'annotation (sans `\tempo`) ;
- les staccatos et autres articulations ou ornementations ;
- les liaisons d'articulation et de prhasé ;
- les crescendos ou decrescendos sur une sele note ;
- les tremolos indiqués par la syntaxe « `:`[*nombre*] » ;
- la basse chiffrée
- les accords en microtonalité.

3.5.4 Répétitions et MIDI

Au prix de quelques réglages, les reprises de toutes sortes peuvent être rendues dans le fichier MIDI. Il suffit pour cela de recourir à la fonction `\unfoldRepeats`, qui développe toutes les reprises. En d'autre termes, `\unfoldRepeats` transforme toutes les reprises en reprises de type `unfold`.

```
\unfoldRepeats {
  \repeat tremolo 8 {c'32 e' }
  \repeat percent 2 { c''8 d'' }
  \repeat volta 2 {c'4 d' e' f'}
  \alternative {
    { g' a' a' g' }
    {f' e' d' c' }
  }
}
\bar "|."
```





Lorsque l'on veut utiliser `\unfoldRepeats` seulement pour le rendu MIDI, il faut établir deux blocs `\score` : un pour le MIDI, avec des reprises explicites, et l'autre pour la partition, avec des reprises notées sous forme de barres de reprise, de trémolo ou de symboles de pourcentage. Par exemple

```
\score {
  ..musique..
  \layout { ..  }
}
\score {
  \unfoldRepeats ..musique..
  \midi { .. }
}
```

3.5.5 Gestion des nuances en MIDI

Les nuances MIDI sont générées par le `Dynamic_performer`, affecté par défaut au contexte `Voice`. Vous pouvez contrôler à la fois le volume général, celui des indications de nuance ainsi que celui des différents instruments.

Indications de nuance

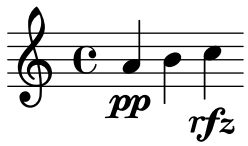
Les indications de nuances sont transcrites en fraction du volume MIDI. Nous allons, par défaut, de 0,25 pour un *ppppp* à 0,95 pour un *fffff*. Les correspondances entre nuance et fraction de volume sont répertoriées dans le fichier '`../scm/midi.scm`' – consultez la rubrique Section "Autres sources de documentation" dans *Manuel d'initiation* si vous ne savez comment le localiser. Vous pouvez modifier ou étendre ce jeu grâce à une fonction qui prendra en argument une indication de nuance et renverra la fraction désirée, puis en affectant cette fonction à `Score.dynamicAbsoluteVolumeFunction`.

Prenons un exemple. Votre partition comporte un *rinforzando* que vous avez indiqué par `\rfz`. Cette indication de nuance n'étant pas répertoriée dans le jeu par défaut, elle ne produira aucun effet en MIDI. Il en sera d'ailleurs de même pour toute indication créée de toute pièce à l'aide de l'instruction `make-dynamic-script`. Voici comment procéder pour ajuster le volume MIDI de ce *rinforzando* que le compositeur a indiqué. La fonction Scheme définit une fraction de 0,9 en cas de *rfz*, et demande d'utiliser la fonction par défaut dans les autre cas.

```
#(define (myDynamics dynamic)
    (if (equal? dynamic "rfz")
      0.9
      (default-dynamic-absolute-volume dynamic)))

\score {
  \new Staff {
    \set Staff.midiInstrument = #"cello"
    \set Score.dynamicAbsoluteVolumeFunction = #myDynamics
    \new Voice {
      \relative c'' {
        a4\pp b c-\rfz
      }
    }
  }
  \layout {}
  \midi {}
```

```
}
```



Si vous étiez amené à devoir modifier l'intégralité du jeu des correspondances, nous vous conseillons d'utiliser la procédure `default-dynamic-absolute-volume` contenue dans le fichier '`../scm/midi.scm`' ansi que la table d'association comme base. Le dernier exemple de cette partie illustre la manière de procéder.

Amplitude du volume en MIDI
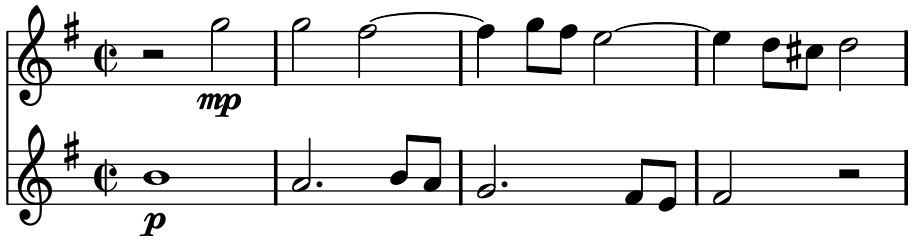
Les valeurs extrêmes du volume MIDI des nuances se contrôlent à l'aide des propriétés `midiMinimumVolume` et `midiMaximumVolume` qui agissent au niveau `Score`. Ces propriétés sont effectives dès lors qu'une nuance est indiquée ; une nuance de départ est donc requise pour que le volume soit ajusté dès le début de la partition. Vous pouvez alors modifier la fraction correspondant à chaque nuance à l'aide de la formule

```
midiMinimumVolume + (midiMaximumVolume - midiMinimumVolume) * fraction
```

Voici comment ajuster les nuances tout en limitant l'amplitude du volume entre 0,2 et 0,5 :

```
\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Staff.midiInstrument = #"flute"
      \new Voice \relative c''' {
        r2 g\mp g fis~
        fis4 g8 fis e2~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout {}
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 2)
      midiMinimumVolume = #0.2
      midiMaximumVolume = #0.5
    }
  }
```
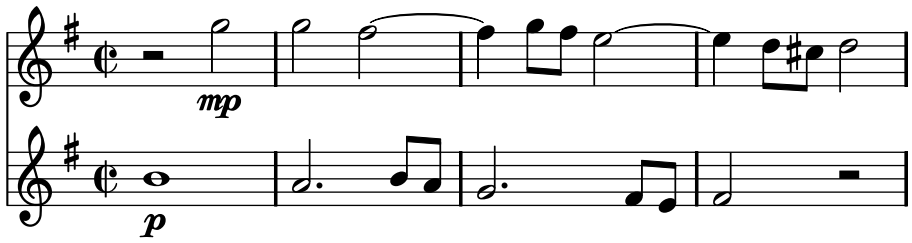
```
}
```



Égalisation de plusieurs instruments (i)

La définition de l'amplitude du volume MIDI au niveau d'un contexte `Staff` permet de gérer les volumes relatifs entre les différents instruments. Ceci constitue en quelque sorte un égaliseur, ce qui permet d'améliorer notablement la qualité de la sortie MIDI.

La clarinette de l'exemple suivant jouera relativement moins fort que la flûte. Rappelez-vous que pour que cela fonctionne correctement, la première note de chacun des instruments doit être affublée d'une indication de nuance.

```
\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Staff.midiInstrument = #"flute"
      \set Staff.midiMinimumVolume = #0.7
      \set Staff.midiMaximumVolume = #0.9
      \new Voice \relative c''' {
        r2 g\mp g fis~
        fis4 g8 fis e2~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \set Staff.midiMinimumVolume = #0.3
      \set Staff.midiMaximumVolume = #0.6
      \new Voice \relative c'' {
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout {}
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 2)
    }
  }
}
```



Égalisation de plusieurs instruments (ii)

Lorsque les propriétés volume minimum et maximum n'ont pas été définies, LilyPond appliquera par défaut une légère égalisation pour quelques instruments. Les instrument concernés ainsi que le niveau d'égalisation sont répertoriés dans une table *instrument-equalizer-alist* du fichier '`../scm/midi.scm`'.

Vous pouvez remplacer l'égaliseur basique en définissant une nouvelle procédure Scheme `instrumentEqualizer` au sein du contexte `Score`. Cette procédure prend en unique argument le nom d'un instrument MIDI et renverra une paire de fractions correspondant au minimum et maximum de volume alloué à cet instrument. Cette substitution fonctionne selon le même principe que celui que nous avons vu en dédut de chapitre avec `dynamicAbsoluteVolumeFunction`. L'égaliseur par défaut, *default-instrument-equalizer*, est défini dans le fichier '`../scm/midi.scm`' ; son analyse vous aidera à construire votre propre procédure.

Nous allons, dans l'exemple suivant, régler le volume relatif de la flûte et de la clarinette – au même niveau que dans le précédent.
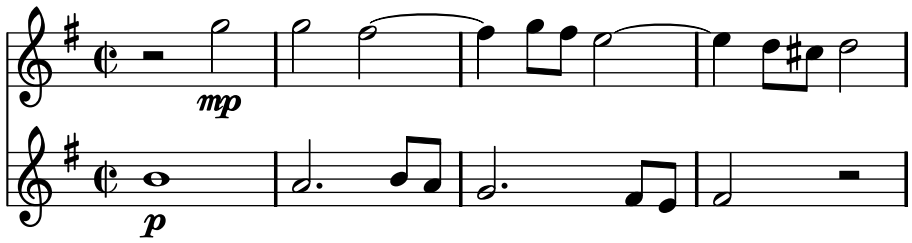
```
#(define my-instrument-equalizer-alist '())

#(set! my-instrument-equalizer-alist
  (append
    '(
      ("flute" . (0.7 . 0.9))
      ("clarinet" . (0.3 . 0.6)))
    my-instrument-equalizer-alist))

#(define (my-instrument-equalizer s)
  (let ((entry (assoc s my-instrument-equalizer-alist)))
    (if entry
      (cdr entry))))

\score {
  <<
    \new Staff {
      \key g \major
      \time 2/2
      \set Score.instrumentEqualizer = #my-instrument-equalizer
      \set Staff.midiInstrument = #"flute"
      \new Voice \relative c''' {
        r2 g\mp g fis~
        fis4 g8 fis e2~
        e4 d8 cis d2
      }
    }
    \new Staff {
      \key g \major
      \set Staff.midiInstrument = #"clarinet"
      \new Voice \relative c'' {
```

```
        b1\p a2. b8 a
        g2. fis8 e
        fis2 r
      }
    }
  >>
  \layout { }
  \midi {
    \context {
      \Score
      tempoWholesPerMinute = #(ly:make-moment 72 2)
    }
  }
}
```



3.5.6 MIDI et percussions

La notation pour percussions recourt généralement à un contexte particulier – le `DrumStaff` – qui permet d'affecter directement les instruments concernés au canal 10 qui leur est réservé.

Certains instruments, tels le xylophone, le marimba, le vibraphone ou les timbales, se traitent cependant comme des instruments « classiques » puisqu'ils sont capables d'émettre des hauteurs différentes ; leurs notation relève donc d'un contexte `Staff` standard, et non d'un `DrumStaff` pour pouvoir être rendus correctement en MIDI.

D'autres percussions, bien que n'émettant qu'un seul son et inclus dans le standard *general MIDI*, comme le tom mélodique ou le tambour taiko, ne sont pas attachés au canal 10. Ces instruments doivent donc être saisi dans un contexte `Staff`, en utilisant la hauteur appropiée.

De nombreux instruments de la famille des percussions, les castagnettes par exemple, n'existent pas dans le standard *general MIDI*. L'alternative, bien que peu satisfaisante, consiste à leur attribuer le son le plus proche dans la banque standard.

Problèmes connus et avertissements

Le standard *general MIDI* ne dispose pas du *rim shot* ; LilyPond lui subtitue un *sidestick*.

4 Gestion de l'espace

L'agencement général d'une partition dépend de trois facteurs interdépendants : la mise en page, les sauts de ligne et l'espacement. Les choix faits en matière d'espacement détermineront la densité de chacun des systèmes, ce qui influera sur le positionnement des sauts de lignes et, par voie de conséquence, sur le nombre de pages de la partition.

En pratique, cette procédure comporte quatre étapes. Dans un premier temps, des distances élastiques *(springs)* sont déterminées sur la base des durées. Sont alors calculées différentes possibilités de saut de ligne, chacune se voyant attribuer un « coefficient de laideur », puis est estimée la hauteur de chaque système. LilyPond opte enfin pour la combinaison entre sauts de page et de ligne qui offre la meilleure occupation de l'espace, tant horizontalement que verticalement.

Les réglages qui influencent la mise en forme se placent dans deux blocs différents. Le bloc `\paper {...}` étudié à la rubrique Section 4.1 [Mise en forme de la page], page 412 contient les réglages applicables à l'intégralité du document – à toutes les partitions d'un ouvrage – tels que format du papier, impression ou non des numéros de page, etc. Quant au bloc `\layout {...}`, qui fait l'objet de la rubrique Section 4.2 [Mise en forme de la partition], page 422, il détermine la mise en forme de la musique : le nombre de systèmes utilisés, l'espacement des regroupements de portées, etc.

Note : Vous verrez au fil de ce chapitre apparaître certains termes dont la traduction vous semblera assurément erronée. Il n'en est cependant rien : certains termes techniques ont une histoire particulière selon leur langue d'origine. Ainsi, le vocable anglais *Ragged* signifie en lambeau, en loques ; dans l'univers typographique, un maître français voit un alignement à gauche – il dira « au fer à gauche » – alors que son homologue anglophone constate un *ragged-right* – donc du vide à droite.

4.1 Mise en forme de la page

Nous allons examiner ici les options qui contrôlent la mise en forme des pages attachées au bloc `\paper`.

4.1.1 Le bloc \paper

Un bloc `\paper` peut apparaître au sein d'un bloc `\book`. Les réglages effectués à l'aide du bloc `\paper` s'appliqueront à l'intégralité d'un ouvrage, qui peut comporter plusieurs partitions (des blocs *score*) ; il ne doit donc en aucun cas se trouver dans un bloc `\score`. Peuvent apparaître dans un bloc `\paper` :

- la fonction scheme `set-paper-size`,
- des variables propres au bloc `\paper` qui viendront adapter la mise en page, et
- la définition des différents *markups* qui personnaliseront la mise en forme des entêtes, pied de page et titrages.

La fonction `set-paper-size` fait l'objet de la rubrique qui suit – Section 4.1.2 [Format du papier et adaptation automatique], page 413. Les variables du bloc `\paper` chargées de la mise en page sont abordées plus loin dans ce chapitre. Quant aux définitions relatives aux *markups* des entête, pied de page et titrages, elles sont étudiées à la rubrique Section 3.2.2 [Titrages personnalisés], page 389.

La plupart des variables gérant le papier ne sont fonctionnelles que lorsque mentionnées dans un bloc `\paper`. Certaines, qui peuvent toutefois apparaître dans un bloc `\layout`, sont référencées à la rubrique Section 4.2.1 [Le bloc \layout], page 422.

Sauf mention contraire, toutes les variables du bloc `\paper` qui correspondent à des dimensions sont exprimées en millimètre – vous pouvez bien entendu spécifier un autre système de mesure. Voici comment, par exemple, définir la marge haute (`top-margin`) à dix millimètres :

```
\paper {
  top-margin = 10
}
```

Si vous préférez lui affecter une valeur de 0,5 pouce, vous devrez mentionner le suffixe d'unité `\in` :

```
\paper {
  top-margin = 0.5\in
}
```

LilyPond accepte les suffixes d'unité `\mm`, `\cm`, `\in` et `\pt`. Ces unités sont des conversions de millimètres, répertoriées dans le fichier '`ly/paper-defaults-init.ly`'. Pour plus de lisibilité, et bien que ce ne soit pas techniquement requis, nous vous conseillons d'ajouter `\mm` à votre code lorsque vous travaillez en millimètres.

Vous pouvez aussi définir les valeurs du bloc `\paper` à l'aide de fonctions Scheme. Voici l'équivalent de l'exemple précédent :

```
\paper {
  #(define top-margin (* 0.5 in))
}
```

Voir aussi

Manuel de notation : Section 4.1.2 [Format du papier et adaptation automatique], page 413, Section 3.2.2 [Titrages personnalisés], page 389. Section 4.2.1 [Le bloc \layout], page 422.

Fichiers d'initialisation : '`ly/paper-defaults-init.ly`'.

4.1.2 Format du papier et adaptation automatique

Format du papier

Le format du papier est déterminé par deux fonctions différentes : `set-default-paper-size` et `set-paper-size`. `set-default-paper-size` se place en tête de fichier, alors que `set-paper-size` se place à l'intérieur d'un bloc `\paper` :

```
#(set-default-paper-size "a4")
\paper {
  #(set-paper-size "a4")
}
```

La seule restriction à l'utilisation isolée de la fonction `set-default-paper-size` est qu'elle doit intervenir avant le premier bloc `\paper`. À l'intérieur d'un bloc `\paper`, il vaut mieux utiliser la fonction `set-paper-size`, avant toute autre variables. Les raisons à ceci sont abordées dans la rubrique [Adaptation automatique au format], page 414.

`set-default-paper-size` fixe le format pour toutes les pages, alors que `set-paper-size` détermine le format des feuilles rattachées à un bloc `\paper` particulier. Ainsi, lorsque le bloc `\paper` se trouve en tête de fichier, le format du papier s'appliquera à toutes les pages ; si `\paper` apparaît dans un bloc `\book`, la taille ne s'appliquera qu'au *book* en question.

Les formats `a4`, `letter`, `legal` et `11x17` (ou tabloïd) sont couramment utilisés pour les éditions musicales. Bien d'autres formats sont disponibles et sont répertoriés à la rubrique `paper-alist` du fichier d'initialisation '`scm/paper.scm`'.

Note : Par défaut, le papier est au format A4 (codé `a4`).

Rien ne vous empêche d'ajouter un format de papier inhabituel à la rubrique `paper-alist` du fichier '`scm/paper.scm`'. Sachez cependant que celui-ci est écrasé à chaque mise à jour de LilyPond.

Le fait d'ajouter l'argument `'landscape` à l'instruction `set-default-paper-size` permet d'obtenir une présentation à l'italienne – ou paysage si vous préférez – et donc des lignes plus longues.

```
#(set-default-paper-size "a6" 'landscape)
```

Voir aussi

Notation Reference: [Adaptation automatique au format], page 414.

Fichiers d'initialisation : '`scm/paper.scm`'.

Adaptation automatique au format

Toute modification du format de papier à l'aide des fonctions scheme `set-default-paper-size` ou `set-paper-size`, que nous avons vues à la rubrique [Format du papier], page 413, se traduira automatiquement par l'ajustement d'un certain nombre de variables attachées au bloc `\paper` afin qu'elles soient en concordance avec le format spécifié. Vous pouvez annuler l'ajustement automatique d'une variable particulière en redéfinissant sa valeur après avoir spécifié le format de papier utilisé. Notez bien que le simple fait d'affecter une valeur à `paper-height` ou `paper-width` ne déclenchera pas l'étalonnage automatique, bien que spécifier une largeur de papier *(*`paper-width`*)* peut influencer d'autres valeurs – mais c'est une autre histoire dont nous parlerons plus tard et qui n'a rien à voir avec la mise à l'echelle.

L'adaptation automatique affecte les dimensionnements verticaux `top-margin` et `bottom-margin` – voir Section 4.1.3 [Variables d'espacement vertical fixe], page 414 –, ainsi que les dimmensionnements horizontaux `left-margin`, `right-margin`, `inner-margin`, `outer-margin`, `binding-offset`, `indent` et `short-indent` – voir Section 4.1.5 [Variables d'espacement horizontal], page 417.

Les valeurs par défaut de ces dimensionnements sont contenues dans le fichier '`ly/paper-defaults-init.ly`' et utilisent les variables internes `top-margin-default`, `bottom-margin-default` etc. correspondant au format par défaut – papier A4 – pour lequel `paper-height` est à `297\mm` et `paper-width` à `210\mm`.

Voir aussi

Manuel de notation : Section 4.1.3 [Variables d'espacement vertical fixe], page 414, Section 4.1.5 [Variables d'espacement horizontal], page 417.

Fichiers d'initialisation : '`ly/paper-defaults-init.ly`', '`scm/paper.scm`'.

4.1.3 Variables d'espacement vertical fixe

Note : Certains dimmensionnements attachés au bloc `\paper` sont automatiquement ajustés selon le format du papier, ce qui peut conduire à un résultat inattendu – voir [Adaptation automatique au format], page 414.

Les valeurs par défaut (avant étalonnage) sont définies dans le fichier '`ly/paper-defaults-init.ly`'.

`paper-height`
: La hauteur de la feuille. Il s'agit par défaut de la dimension du papier utilisé. Notez bien que cette variable n'affectera pas l'ajustement automatique d'un certain nombre de dimmensionnement vertical.

`top-margin`
: La marge entre le bord supérieur de la feuille et la surface imprimable. Elle est fixée par défaut à `5\mm` et s'ajustera selon le format de papier.

`bottom-margin`
: La marge entre la surface imprimable et le bord inférieur de la feuille. Elle est fixée par défaut à `6\mm` et s'ajustera selon le format de papier.

`ragged-bottom`
: L'activation de cette variable permet de ne pas répartir verticalement les systèmes sur les pages hormis la dernière. La valeur par défaut est `##f`. Lorsque la partition ne comporte que deux ou trois systèmes par page, comme pour un conducteur d'orchestre, nous vous conseillons d'activer cette variable.

`ragged-last-bottom`
: La désactivation de cette variable permet de répartir verticalement les systèmes de la dernière page d'une partition. La valeur par défaut est `##t`. Nous vous conseillons, lorsque des pièces couvrent deux pages ou plus, d'activer cette variable.
Notez bien que la variable `ragged-last-bottom` affecte aussi la dernière page de chacune des parties – créées à l'aide d'un bloc `\bookpart` – d'un même ouvrage.

Voir aussi

Manuel de notation : [Adaptation automatique au format], page 414.

Fichiers d'initialisation : '`ly/paper-defaults-init.ly`'.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Problèmes connus et avertissements

Les titrages (contenus dans le bloc `\header{}`) sont considérés comme des systèmes à part entière ; ils seront donc affectés par `ragged-bottom` et `ragged-last-bottom`, qui éventuellement ajouteront de l'espace avant le premier système de la partition.

4.1.4 Variables d'espacement vertical fluctuant

Il est souvent judicieux d'apporter un peu de flexibilité à l'espacement entre différents éléments (marges, titres, systèmes ou mouvements), en dilatation ou compression selon le cas. Un certain nombre de variables de type `\paper` répertoriées ci-dessous vous permettront d'affiner ces dimensionnements.

Gardez à l'esprit que les variables du bloc `\paper` dont nous parlons ici n'influencent en rien l'espacement des portées d'un même système. L'espacement au sein des systèmes est géré par des propriétés attachées à des objets graphiques (*grobs*) qui, elles, se définissent au niveau du bloc `\score` – voir à ce sujet Section 4.4.1 [Espacement vertical au sein d'un système], page 432.

Structure des variables d'espacement vertical fluctuant

Chacune de ces variables attachées au bloc `\paper` est constituée d'une liste associative *(alist)* à quatre *clés* :

- `basic-distance` *(distance de base)* – la grandeur d'espace par défaut, exprimée en hauteur de portée, séparant les *points de référence* de deux éléments, qui évite tout risque de collision en l'absence de dilatation ou compression. Le point de référence d'un titre ou d'un *markup* est son sommet, celui d'un système est le centre vertical du `StaffSymbol` le plus proche – même lorsqu'une ligne de « non-portée » viendrait à s'intercaler. Une `basic-distance` inférieure à `padding` ou `minimum-distance` sera sans effet, dans la mesure où l'espacement résultant ne saurait être inférieur à `padding` ou `minimum-distance`.

- `minimum-distance` *(distance-minimale)* – l'espacement minimal, exprimé en hauteur de portée, entre les points de référence des deux éléments alors qu'il y a déjà un effet de compression. Une `minimum-distance` inférieure à la valeur du `padding` sera sans effet, dans la mesure où l'espacement résultant ne saurait être inférieur au `padding`.
- `padding` *(décalage)* – la grandeur minimale de « blanc » qui sépare deux éléments, exprimée en hauteur de portée. On peut le voir comme la hauteur minimale d'un rectangle vide qui devrait s'étendre sur toute la largeur des deux éléments.
- `stretchability` *(dilatation)* – le coefficient d'étirement de cet espace. Un coefficient nul permet de figer l'espacement, à moins qu'il n'en résulte des collisions. Un coefficient positif déterminera la propension d'un espacement à s'étirer, tout en tenant compte du coefficient affecté aux autres espacements. Par exemple, lorsque le coefficient de dilatation d'une dimension est double de celui d'une autre, elle pourra s'étirer deux fois plus que cette dernière. Il ne saurait être négatif. La valeur `+inf.0` provoque une `programming_error` (erreur de programmation) et est ignorée ; vous pouvez toutefois utiliser `1.0e7` pour obtenir une valeur proche de l'infini. Lorsque cette *clé* n'est pas définie, sa valeur est par défaut égale à `space`. Notez bien que l'utilisateur ne peut définir une propension à la compression ; elle est en fait égale à (`basic-distance` − `minimum-distance`).

Lorsque l'impression n'est pas en pleine page – elle est donc *ragged bottom* pour les anglophones – l'élément `space` n'est pas étiré. Les hauteurs sur une telle page correspondront donc au maximum de

- `basic-distance`, plus
- `minimum-distance` et
- `padding`, augmenté de ce qu'il faut pour éviter les chevauchements.

Les manières de modifier des listes associatives font l'objet d'un Section "chapitre spécifique" dans *Manuel de notation*. L'exemple suivant indique deux façons de modifier une liste associative. La première déclaration intervient sur une seule clé, alors que la deuxième redéfinit complètement la variable.

```
\paper {
  system-system-spacing #'basic-distance = #8

  score-system-spacing =
    #'((padding . 1)
       (basic-distance . 12)
       (minimum-distance . 6)
       (stretchability . 12))
}
```

Liste des variables d'espacement fluctuant

Le nom des dimensionnements à hauteur variable sont de la forme *`avant-après`*`-spacing`, où *`avant`* et *`après`* représentent les éléments qui doivent être espacés. La distance s'établit entre les points de référence des deux éléments concernés (voir la rubrique précédente pou plus de précision). Notez bien que, dans les règles de nommage des variables qui suivent, le terme `markup` fait référence aussi bien à un *markup de titrage* (`bookTitleMarkup` ou `scoreTitleMarkup`) qu'à un *markup de haut niveau* (voir Section 3.1.5 [Structure de fichier], page 384). Toutes les distances sont exprimées en espace de portée.

Leurs valeurs par défaut sont inscrites dans le fichier '`ly/paper-defaults-init.ly`'.

`markup-system-spacing`
: détermine l'espacement entre un titre ou un *markup* de premier niveau, et le système qui le suit.

`score-markup-spacing`
: détermine l'espacement entre le dernier système et le titre ou *markup* de haut niveau qui le suit.

`score-system-spacing`
: détermine l'espacement entre le dernier système d'une partition et le premier système de la partition suivante, en l'absence de titrage ou *markup* qui les sépare.

`system-system-spacing`
: détermine l'espacement entre deux systèmes d'un même mouvement.

`markup-markup-spacing`
: détermine l'espacement entre deux titres ou *markups* de premier niveau.

`last-bottom-spacing`
: détermine la distance entre le dernier système ou le dernier *markup* de haut niveau, et le bas de la surface imprimable – autrement dit le haut de la marge basse.

`top-system-spacing`
: détermine l'espace entre le haut de la surface imprimable (le bas de la marge haute) et le milieu du premier système. Cette variable n'est effective qu'en l'absence de titre ou *markup* de premier niveau en haut de page.

`top-markup-spacing`
: détermine l'espace entre le haut de la surface imprimable (le bas de la marge haute) et le premier titre ou *markup* de premier niveau. Cette variable n'est effective qu'en l'absence de système en haut de page.

Voir aussi

Manuel de notation : Section 4.4.1 [Espacement vertical au sein d'un système], page 432.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Snippets: Section "Espacements" dans *Morceaux choisis*.

4.1.5 Variables d'espacement horizontal

Note : Certains dimmensionnements attachés au bloc `\paper` sont automatiquement ajustés selon le format du papier, ce qui peut conduire à un résultat inattendu – voir [Adaptation automatique au format], page 414.

Variables de marge et de largeur

Les valeurs par défaut (avant étalonnage) sont définies dans le fichier '`ly/paper-defaults-init.ly`'.

`paper-width`
: La largeur de la page. Elle correspond par défaut à la largeur du format de papier utilisé. Si `paper-width` n'a aucun effet en matière d'ajustement automatique, cette variable influe sur la variable `line-width`. Lorsque vous définissez à la fois les valeurs de `paper-width` et `line-width`, les valeurs de `left-margin` et `right-margin` seront recalculées. Voir aussi `check-consistency`.

`line-width`
: la longueur d'un système musical sans indentation et justifiée sur toute la largeur de la page. La valeur par défaut est égale à `paper-width`, auquel sont retranchés `left-margin` et `right-margin`. Lorsque vous définissez `line-width` sans modifier les valeurs de `left-margin` et `right-margin`, les marges seront alors recalculées

de telle sorte que les systèmes soient centrés. Voir aussi `check-consistency`. La variable `line-width` peut se définir aussi dans un bloc `\layout`.

`left-margin`
: la marge entre le bord gauche de la feuille et le début de chaque système. La valeur par défaut est de `10\mm` ; elle sera ajustée selon le format du papier. Lorsque vous définissez `line-width` et `right-margin`, sans modifier la valeur de `left-margin`, cette dernière sera alors égale à `(paper-width − line-width − right-margin)`. Lorsque seule `line-width` est définie, les deux marges correspondent à `((paper-width − line-width) / 2)`, ce qui a pour effet de centrer les systèmes sur la page. Voir aussi `check-consistency`.

`right-margin`
: La marge entre le bord droit de la page et la fin des systèmes en pleine largeur (non *ragged*). La valeur par défaut est de `10\mm` et s'ajustera selon le format du papier. Lorsque vous définissez `line-width` et `left-margin`, sans modifier la valeur de `right-margin`, cette dernière sera alors égale à `(paper-width − line-width − left-margin)`. Lorsque seule `line-width` est définie, les deux marges correspondent à `((paper-width − line-width) / 2)`, ce qui a pour effet de centrer les systèmes sur la page. Voir aussi `check-consistency`.

`check-consistency`
: Lorsqu'elle est activée, cette variable vérifie que `left-margin`, `line-width` et `right-margin` sont en cohérence, et que l'addition de ces trois éléments ne dépassera pas la largeur du papier (`paper-width`). La valeur par défaut est `##t`. Dans le cas d'une incohérence, un message d'avertissement est émis et les trois variables – marges et longueur de ligne – rétablies à leur valeur par défaut (ajustées selon le fromat du papier).

`ragged-right`
: Lorsque cette variable est activée, les systèmes ne s'étendront pas sur la longueur de la ligne, mais s'arrêteront à leur longueur normale. La valeur par défaut est `##f` mais, si la partition ne comporte qu'un seul système, elle passe à `##t`. Cette variable peut aussi se gérer au sein d'un bloc `\layout`.

`ragged-last`
: Lorsqu'elle est activée, cette variable permet de ne pas étendre le dernier système de façon à occuper toute la longueur de la ligne. La valeur par défaut est `##f`. Cette variable peut aussi se gérer au sein d'un bloc `\layout`.

Voir aussi

Notation Reference: [Adaptation automatique au format], page 414.

Fichiers d'initialisation : '`ly/paper-defaults-init.ly`'.

Variables spécifiques pour l'impression recto-verso

Les valeurs par défaut (avant étalonnage) sont définies dans le fichier '`ly/paper-defaults-init.ly`'.

`two-sided`
: Cette variable permet de gérer efficacement les impressions recto-verso. Lorsqu'elle est activée, les réglages affectés à `inner-margin`, `outer-margin` ainsi que `binding-offset` détermineront les différentes marges selon qu'il s'agit d'une page paire ou impaire. Cette variable s'applique en lieu et place de `left-margin` et `right-margin`. La valeur par défaut est `##f`.

`inner-margin`
: La marge que toutes les pages d'une partie ou de tout un ouvrage devront avoir du côté intérieur. Bien entendu, cette variable n'est effective que lorsque vous comptez générer un fichier imprimable en recto-verso – propriété `two-sided` définie à vrai. La valeur par défaut est de `10\mm` et s'ajustera selon le format du papier.

`outer-margin`
: la marge que toutes les pages d'une partie ou de tout un ouvrage devront avoir du côté extérieur – opposé à la reliure. Bien entendu, cette variable n'est effective que lorsque vous comptez générer un fichier imprimable en recto-verso – propriété `two-sided` définie à vrai. La valeur par défaut est de `20\mm` et s'ajustera selon le format du papier.

`binding-offset`
: La gouttière, ou marge de reliure, permet d'augmenter en conséquence la valeur de la marge intérieure `inner-margin` de telle sorte que rien ne soit masqué par la reliure. Bien entendu, cette variable n'est effective que lorsque vous comptez générer un fichier imprimable en recto-verso – propriété `two-sided` définie à vrai. La valeur par défaut est de `0` et s'ajustera selon le format du papier.

Voir aussi

Manuel de notation : [Adaptation automatique au format], page 414.

Fichiers d'initialisation : '`ly/paper-defaults-init.ly`'.

Variables d'indentation et de décalage

Les valeurs par défaut (avant étalonnage) sont définies dans le fichier '`ly/paper-defaults-init.ly`'.

`horizontal-shift`
: Tous les systèmes, ainsi que les titres et séparateurs de systèmes, seront poussés d'autant vers la droite. la valeur par défaut est de `0.0`.

`indent`
: Le niveau d'indentation du premier système d'une partition. La valeur par défaut est de `15\mm` et s'ajustera selon le format du papier. Cette variable peut aussi se gérer au sein d'un bloc `\layout`.

`short-indent`
: Le niveau d'indentation de tous les systèmes hormis le premier. La valeur par défaut est de `0`, et s'ajustera selon le format du papier dès lors que vous lui affecterez une valeur. Cette variable peut aussi se gérer au sein d'un bloc `\layout`.

Voir aussi

Manuel de notation : [Adaptation automatique au format], page 414.

Fichiers d'initialisation : '`ly/paper-defaults-init.ly`'.

Snippets: Section "Spacing" dans *Morceaux choisis*.

4.1.6 Autres variables du bloc `\paper`

Variables de gestion des sauts de ligne

`max-systems-per-page`
: Le nombre maximal de systèmes qu'une page pourra comporter. Cette variable n'est prise en compte, à ce jour, que par l'option `ly:optimal-breaking`, et n'est pas définie.

`min-systems-per-page`
: Le nombre minimal de systèmes qu'une page pourra comporter. Attention cependant aux risques de débordement s'il est trop important. Cette variable n'est prise en compte, à ce jour, que par l'option `ly:optimal-breaking`, et n'est pas définie.

`systems-per-page`
: Le nombre de systèmes que devrait comporter chaque page. Cette variable n'est à ce jour prise en charge que par l'algorithme `ly:optimal-breaking` et n'est pas définie par défaut.

`system-count`
: Le nombre de systèmes requis par la partition. Cette variable n'est pas définie par défaut. Cette variable peut se gérer au sein d'un bloc `\layout`.

Voir aussi

Manuel de notation : Section 4.3.1 [Sauts de ligne], page 424.

Variables de gestion des sauts de page

Les valeurs par défaut sont définies dans le fichier '`ly/paper-defaults-init.ly`'.

`blank-after-score-page-force`
: Pénalité pour apparition d'une page blanche entre deux partitions. Sa valeur est par défaut inférieure à celle de `blank-page-force` ; nous préférons q'une page blanche s'insère après la fin de la partition plutôt qu'au milieu.

`blank-last-page-force`
: Pénalité pour fin de partition intervenant sur une page impaire. La valeur par défaut est de `0`.

`blank-page-force`
: Pénalité pour apparition d'une page blanche en cours de partition. L'algorithme `ly:optimal-breaking` n'en tiendra pas compte puisqu'il ne conçoit pas la présence d'une page blanche au milieu d'une partition. La valeur par défaut est de `5`.

`page-breaking`
: L'algorithme de calcul des sauts de page à utiliser. Vous avez le choix entre `ly:minimal-breaking`, `ly:page-turn-breaking` et `ly:optimal-breaking`.

`page-breaking-system-system-spacing`
: Cette variable permet de « tromper » l'algorithme de gestion des sauts de page quant à la valeur de `system-system-spacing`. Ainsi, lorsque `page-breaking-system-system-spacing #'padding` a une valeur nettement supérieure à `system-system-spacing #'padding`, l'algorithme en question aura tendance à disposer moins de systèmes sur une même page. Cette variable est par défaut non définie.

`page-count`
: Le nombre de pages que devra comporter la partition. Cette variable est par défaut non définie.

Voir aussi

Manuel de notation : Section 4.3.2 [Sauts de page], page 426, Section 4.3.3 [Optimisation des sauts de page], page 427, Section 4.3.4 [Optimisation des tournes], page 427, Section 4.3.5 [Minimisation des sauts de page], page 428.

Fichiers d'initialisation : '`ly/paper-defaults-init.ly`'.

Variables de gestion des numéros de page

Les valeurs par défaut sont définies dans le fichier '`ly/paper-defaults-init.ly`'.

`auto-first-page-number`
: L'algorithme qui gère les sauts de page prend en compte le fait que le premier numéro de page soit pair ou impair. Lorsque cette fonctionnalité est activée, l'algorithme des sauts de page décidera de lui-même si le premier numéro sera pair ou impair, ce qui se traduira par un éventuel incrément de un. La valeur par défaut est `##f`.

`first-page-number`
: Le numéro de la première page. La valeur par défaut est de `#1`.

`print-first-page-number`
: Cette variable permet d'imprimer le numéro de page y compris sur la première. La valeur par défaut est `##f`.

`print-page-number`
: La désactivation de cette variable permet d'obtenir des pages non numérotées. La valeur par défaut est `##t`.

Voir aussi

Fichiers d'initialisation : '`ly/paper-defaults-init.ly`'.

Problèmes connus et avertissements

Les pages au numéro impair sont toujours à droite. Pour que la musique commence en page 1, le dos de la page de garde doit être vide de telle sorte que la page une se retrouve à droite.

Variables supplémentaires

`page-spacing-weight`
: Cette variable définit l'importance relative des espacements entre la page (verticalité) et la ligne (horizontalité). Une valeur élevée privilégiera l'espacement au niveau de la page. La valeur par défaut est de `#10`.

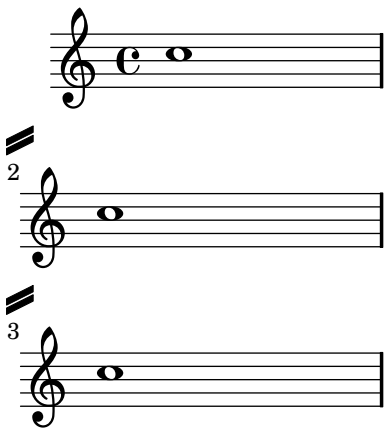
`print-all-headers`
: Lorsque cette variable est activée, l'intégralité des champs d'entête sera imprimée pour chaque bloc `\score`, plutôt que les seuls champs `piece` et `opus`. La valeur par défaut est `##f`.

`system-separator-markup`
: Il s'agit en l'occurrence d'insérer un objet de type *markup* entre chaque système, comme on le voit dans nombre de partitions orchestrales. Cette variable n'est pas définie par défaut. La commande `\slashSeparator` – définie dans le fichier '`ly/titling-init.ly`' – fournit un *markup* relativement courant :

```
#(set-default-paper-size "a8")

\book {
  \paper {
    system-separator-markup = \slashSeparator
  }
  \header {
    tagline = ##f
  }
  \score {
    \relative c'' { c1 \break c1 \break c1 }
```

```
  }
}
```



Voir aussi

Fichiers d'initialisation : '`ly/titling-init.ly`'.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Problèmes connus et avertissements

L'entête par défaut, formé d'une seule ligne, est constitué du numéro de page et du champ `instrument` contenu dans le bloc `\header`.

4.2 Mise en forme de la partition

Nous allons voir ici les options du bloc `\layout`. Elles sont plus particulièrement destinées à gérer la mise en forme de la partition.

4.2.1 Le bloc `\layout`

Alors que le bloc `\paper` définit le formatage des pages pour l'intégralité du document, le bloc `\layout` s'occupe de la mise en forme spécifique à la partition. La mise en forme de la musique peut concerner toutes les partitions d'un même ouvrage, auquel cas un bloc `\layout` indépendant se placera en tête de fichier. Dans le cas où la mise en forme concerne une partition en particulier, un bloc `\layout` se placera au sein du bloc `\score` en question. Sont susceptibles d'apparître dans un bloc `\layout` :

- la fonction scheme `layout-set-staff-size`,
- dans des blocs `\contexte`, les modifications apportées aux différents contextes, et
- les variables normalement attachées au bloc `\paper` qui affacteront la mise en forme de la partition.

La fonction `layout-set-staff-size` fait l'objet de la rubrique suivante, Section 4.2.2 [Définition de la taille de portée], page 423. La modification des contextes est abordée dans d'autres chapitres – voir ⟨undefined⟩ [Modification des propriétés d'un contexte], page ⟨undefined⟩ et Section 5.1.5 [Modification des réglages par défaut d'un contexte], page 466. Les variables du bloc `\paper` que l'on peut retrouver dans un bloc `\layout` sont :

- `line-width`, `ragged-right` et `ragged-last` (voir [Variables de marge et de largeur], page 417)
- `indent` et `short-indent` (see [Variables d'indentation et de décalage], page 419)
- `system-count` (see [Variables de gestion des sauts de ligne], page 419)

Voici un exemple de bloc `\layout` :

```
\layout {
  indent = 2\cm
  \context {
    \StaffGroup
    \override StaffGrouper #'staff-staff-spacing #space = #8
  }
  \context {
    \Voice
    \override TextScript #'padding = #1
    \override Glissando #'thickness = #3
  }
}
```

Voir aussi

Manuel de notation : Section 5.1.5 [Modification des réglages par défaut d'un contexte], page 466.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.2.2 Définition de la taille de portée

La **taille de portée** *(staff size)* est fixée par défaut à 20 points. Il existe deux manières de la modifier :

La taille des portées peut se définir globalement pour toutes les partitions d'un même fichier, ou plus précisément d'un bloc `\book`, à l'aide de `set-global-staff-size`.

```
#(set-global-staff-size 14)
```

Ceci définit donc la hauteur des portées à 14 points par défaut ; toutes les fontes seront ajustées en conséquence.

Vous pouvez aussi spécifier une taille à une partition en particulier en procédant comme ci-dessous :

```
\score{
  ...
  \layout{
    #(layout-set-staff-size 15)
  }
}
```

La fonte Feta fournit les symboles musicaux dans huit tailles différentes. Chaque fonte correspond à une hauteur particulière de portée ; les petites tailles comportent des symboles plus épais pour être cohérent avec l'épaisseur relativement plus importante des lignes de la portée. Le tableau suivant répertorie les différentes tailles de police.

nom de la fonte	**hauteur de portée (pt)**	**hauteur de portée (mm)**	**utilisation**
feta11	11,22	3,9	format de poche
feta13	12,60	4,4	
feta14	14,14	5,0	
feta16	15,87	5,6	
feta18	17,82	6,3	carnet de chant

feta20	20	7,0	partition standard
feta23	22,45	7,9	
feta26	25,2	8,9	

Ces fontes sont disponibles dans toutes les tailles. La propriété de contexte `fontSize` ainsi que la propriété de mise en forme `staff-space` (voir Section "StaffSymbol" dans *Référence des propriétés internes*) permettent d'ajuster individuellement la taille de chaque portée. La taille de chacune des portées est relative à la taille globale.

Voir aussi

Manuel de notation : [Indication de la taille de fonte musicale], page 182.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Problèmes connus et avertissements

`layout-set-staff-size` ne modifie en rien l'espacement entre les lignes d'une portée.

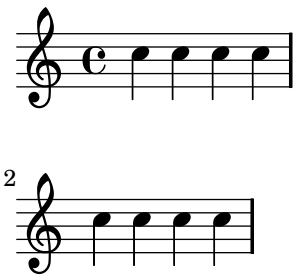
4.3 Sauts

4.3.1 Sauts de ligne

Les sauts de ligne sont normalement gérés de façon automatique. Ils interviennent de telle sorte qu'une ligne ne soit ni trop resserrée, ni trop aérée, et que des lignes consécutives aient à peu près la même densité.
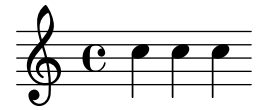
Vous pouvez cependant insérer l'instruction `\break` à l'endroit où vous le jugez utile pour « forcer » le passage à la ligne suivante :
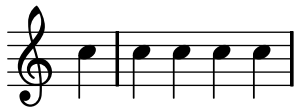
```
c4 c c c | \break
c4 c c c |
```



Par défaut, un saut de ligne ne saurait intervenir au beau milieu d'une mesure ; LilyPond vous le signalera par un message si tel était le cas. Si d'aventure vous voulez forcer un saut de ligne en l'absence de barre de mesure, vous devrez auparavant insérer une barre invisible – à l'aide de `\bar ""`.
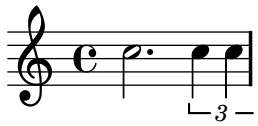
```
c4 c c
\bar "" \break
c |
c4 c c c |
```
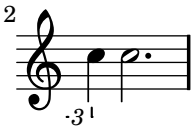




LilyPond ignorera un `\break` placé sur une barre à la fin d'une mesure dès lors que la précédente avait une note en suspend – c'est typiquement le cas lorsqu'un nolet est à cheval sur deux mesures. L'instruction `\break` sera alors opérationnelle si vous avez auparavant désactivé le `Forbid_line_break_engraver` du contexte `Voice` concerné. Notez bien que les sauts de ligne forcés doivent être saisis au sein d'une expression polyphonique :
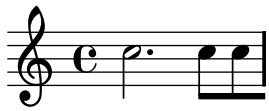
```
\new Voice \with {
  \remove Forbid_line_break_engraver
} \relative c'' {
  <<
    { c2. \times 2/3 { c4 c c } c2. | }
    { s1 | \break s1 | }
  >>
}
```
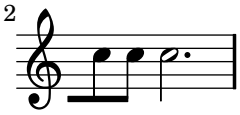




Selon le même principe, un saut de ligne ne peut intervenir alors qu'une ligature s'étend sur deux mesures consécutives. Il faut en ce cas là introduire la dérogation `\override Beam #'breakable = ##t`.

```
\override Beam #'breakable = ##t
c2. c8[ c | \break
c8 c] c2. |
```





L'instruction opposée, `\noBreak`, interdira toute tentative de saut de ligne à la fin de la mesure où elle est explicitée.

LilyPond dispose de deux variables de base pour influencer l'espacement au niveau des lignes. Toutes deux se définissent dans un bloc `\layout`, `indent` réglant l'indentation de la première ligne, et `line-width` la longueur des lignes.

L'activation du commutateur `ragged-right` au sein du bloc `\layout` aura pour effet de terminer les systèmes là où il le feraient normalement plutôt que de les étirer sur toute la longueur de la ligne. Ceci est particulièrement utile pour de petits fragments ou pour vérifier la densité induite par l'espacement naturel.

Le commutateur `ragged-last` est équivalent à `ragged-right`, à ceci près qu'il n'affecte que la dernière ligne de la pièce.

```
\layout {
  indent = 0\mm
  line-width = 150\mm
  ragged-last = ##t
}
```

L'utilisation conjointe de `\break` et de blancs dans une section `\repeat` vous permettra de positionner des sauts de ligne à intervalle régulier. Par exemple, les 28 mesures de ce qui suit, si 'lon est à 4/4, seront coupées toutes les quatre mesures, pas ailleurs :

```
<<
  \repeat unfold 7 {
    s1 \noBreak s1 \noBreak
    s1 \noBreak s1 \break
  }
  { the actual music... }
>>
```

Commandes prédéfinies

`\break`, `\noBreak`.

Voir aussi

Notation Reference: [Variables de gestion des sauts de ligne], page 419.

Référence des propriétés internes : Section "LineBreakEvent" dans *Référence des propriétés internes*.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.3.2 Sauts de page

La gestion automatique des sauts de page se contrôle à l'aide des commandes `\pageBreak` et `\noPageBreak`. Ces commandes fonctionnent de manière analogue à `\break` et `\noBreak` pour les sauts de ligne et se placent donc au moment d'une barre de mesure. Elles permettent de forcer, ou d'interdire, un saut de page à l'endroit indiqué. Comme on peut s'y attendre, `\pageBreak` force le saut de ligne.

Les commandes `\pageBreak` et `\noPageBreak` peuvent se trouver à des niveaux supérieurs, entre deux partitions ou *markups* de premier rang.

Tout comme `ragged-right` et `ragged-last` qui permettent de gérer la répartition horizontale, LilyPond dispose de commutateurs équivalents au niveau de la verticalité : `ragged-bottom` et `ragged-last-bottom`. Lorsqu'ils sont tous deux activés – affectés de `##t` – les systèmes ne seront pas répartis sur les pages y compris la dernière. Pour de plus amples détails, reportez-vous à Section 4.1.3 [Variables d'espacement vertical fixe], page 414.

Les sauts de page sont générés par la fonction `page-breaking`. LilyPond dispose de trois différents algorithmes en la matière : `ly:optimal-breaking`, `ly:page-turn-breaking` et `ly:minimal-breaking`. C'est `ly:optimal-breaking` qui est activé par défaut, mais rien ne vous empêche d'en changer, par l'intermédiaire du bloc `\paper` :

```
\paper {
  page-breaking = #ly:page-turn-breaking
}
```

Lorsqu'un ouvrage contient plusieurs partitions et un certain nombre de pages, la gestion des sauts de page finit par devenir très gourmande, tant au niveau du processeur que de la mémoire. Vous pouvez cependant alléger la charge en recourant à des blocs `\bookpart` afin de sectionner l'ouvrage que vous traitez ; les sauts de pages seront alors gérés individuellement au niveau de chacune des parties. Par ailleurs, cela vous autorisera une gestion différente selon les sections.

```
\bookpart {
  \header {
    subtitle = "Préface"
  }
  \paper {
     %% Pour une partie constituée principalement de texte
     %% ly:minimal-breaking est plus judicieux.
     page-breaking = #ly:minimal-breaking
  }
  \markup { ... }
  ...
}
\bookpart {
  %% Cette partie étant purement musicale,
  %% retour au style par défaut (optimal-breaking).
  \header {
    subtitle = "Premier mouvement"
  }
  \score { ... }
  ...
}
```

Commandes prédéfinies

`\pageBreak`, `\noPageBreak`.

Voir aussi

Manuel de notation : [Variables de gestion des sauts de page], page 420.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.3.3 Optimisation des sauts de page

LilyPond, pour déterminer où placer un saut de page, utilise par défaut la fonction `ly:optimal-breaking`. Celle-ci tend à trouver une rupture qui évite d'obtenir à la fois une page trop dense ou exagérément aérée. Contrairement à la fonction `ly:page-turn-breaking`, elle n'a aucune notion de ce qu'est une « tourne ».

Voir aussi

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.3.4 Optimisation des tournes

Aboutir à une configuration des sauts de page de telle sorte que les pages de droite se terminent toujours par un silence devient souvent une nécessité. En effet, l'exécutant pourra alors tourner la page sans risquer de manquer des notes. La fonction `ly:page-turn-breaking` tend à trouver une rupture qui évite d'obtenir à la fois une page trop dense ou exagérément aérée, tout en tenant compte du fait qu'une tourne ne saurait intervenir qu'à certains endroits.

L'utilisation de cette fonction se fait en deux étapes. Il vous faut tout d'abord l'activer au sein du bloc `\paper` comme indiqué à la rubrique Section 4.3.2 [Sauts de page], page 426. Vous devrez, dans un deuxième temps, informer la fonction des endroits où les sauts de page sont permis.

Cette deuxième étape se réalise de deux manières différentes. Vous pouvez spécifier manuellement chaque tourne potentielle en insérant un `\allowPageTurn` à l'endroit approprié de votre fichier source.

Toutefois, cette option peut vite se révéler fastidieuse selon l'ampleur de l'œuvre. Vous pouvez alors recourir au `Page_turn_engraver` que vous mentionnerez dans un contexte de voix ou de portée. Ce graveur de tournes recherchera dans le contexte en question les passages sans note. Notez bien qu'il ne recherche pas des silences, mais l'absence de notes ; autrement dit, il ne restera pas inactif dans la cadre d'une portée polyphonique dont l'une des parties contiendrait des silences. Lorsqu'il rencontre un fragment suffisamment long ne contenant aucune note, il insère un `\allowPageTurn` à la barre terminant ce fragment, à moins qu'il ne rencontre en chemin une « barre spéciale » – telle une double barre – auquel cas il y déposera le `\allowPageTurn`.

Le `Page_turn_engraver` examine la propriété de contexte `minimumPageTurnLength` pour déterminer quelle doit être la longueur d'un fragment sans note avant une tourne. La valeur par défaut de `minimumPageTurnLength` est `#(ly:make-moment 1 1)`, soit une ronde, et s'ajuste de la manière suivante :

```
\new Staff \with { \consists "Page_turn_engraver" }
{
  a4 b c d |
  R1 | % une tourne peut se placer ici
  a4 b c d |
  \set Staff.minimumPageTurnLength = #(ly:make-moment 5 2)
  R1 | % il ne peut pas y avoir de tourne ici
  a4 b r2 |
  R1*2 | % une tourne peut se placer ici
  a1
}
```

Le `Page_turn_engraver` tient compte des reprises. C'est pourquoi il ne permettra une tourne que dans le cas où il y aura suffisamment de temps au début et à la fin de la reprise pour que l'exécutant ait le temps de revenir à la page précédente. Le `Page_turn_engraver` est même capable d'interdire un tourne dans le cas d'une reprise de courte durée, ajustable au travers de la propriété de contexte `minimumRepeatLengthForPageTurn`.

Les commandes de tourne – `\pageTurn`, `\noPageTurn` et `\allowPageTurn` – peuvent s'utiliser à des niveaux supérieurs, entre des blocs `\score` ou des *markups* de haut niveau.

Commandes prédéfinies

`\pageTurn`, `\noPageTurn`, `\allowPageTurn`.

Voir aussi

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Problèmes connus et avertissements

Une partition ne devrait contenir qu'une seule instance du `Page_turn_engraver`, au risque de se contredire les uns les autres.

4.3.5 Minimisation des sauts de page

La fonction `ly:minimal-breaking` est celle qui réalise le moins de calculs pour positionner les sauts de page. Elle mettra le plus de systèmes possible sur une page avant de passer à la suivante. On peut donc la préférer lorsque la partition s'étend sur beaucoup de pages ou lorsque les autres fonctions de gestion des sauts de page ralentissent nettement le traitement, sont trop gourmandes en mémoire ou qu'il y a beaucoup de texte. Il suffit de la mentionner au sein du bloc `\paper` :

```
\paper {
  page-breaking = #ly:minimal-breaking
}
```

Voir aussi

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.3.6 Sauts explicites

Il arrive parfois que LilyPond rejette des `\break` ou des `\pageBreak` explicites. Vous pouvez alors prendre le contrôle avec ces deux instructions dérogatoires :

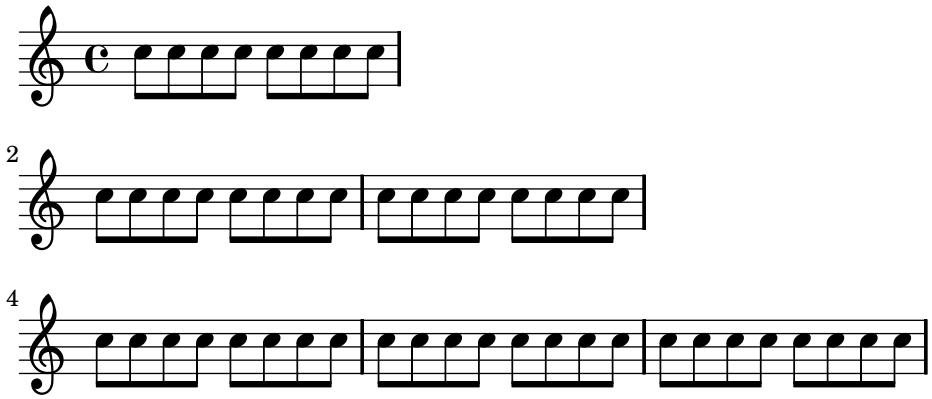
```
\override NonMusicalPaperColumn #'line-break-permission = ##f
\override NonMusicalPaperColumn #'page-break-permission = ##f
```

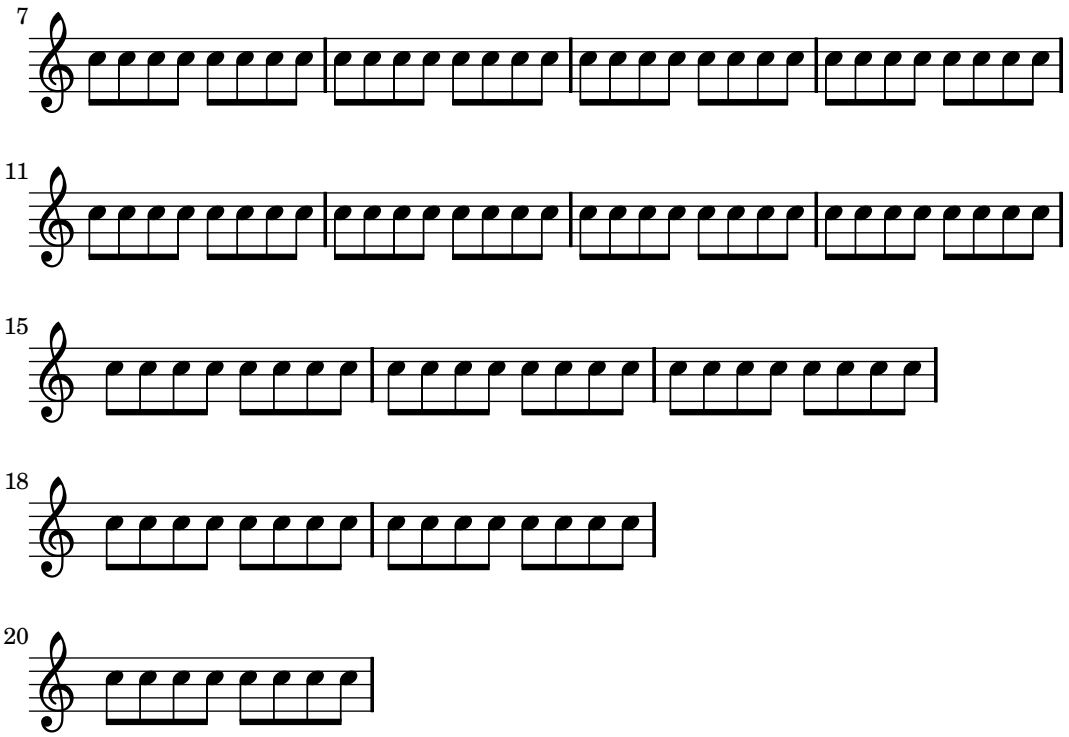
Lorsque vous désactivez `line-break-permission`, LilyPond ne passera à la ligne suivante qu'en présence d'un `\break` explicite, et nulle part ailleurs. De la même façon, la désactivation de `page-break-permission` aura pour conséquence que LilyPond ne changera de page que lorsqu'il rencontrera un `\pageBreak`, et nulle part ailleurs.

```
\paper {
  indent = #0
  ragged-right = ##t
  ragged-bottom = ##t
}

musique = \relative c'' { c8 c c c }

\score {
  \new Staff {
    \repeat unfold 2 { \musique } \break
    \repeat unfold 4 { \musique } \break
    \repeat unfold 6 { \musique } \break
    \repeat unfold 8 { \musique } \pageBreak
    \repeat unfold 8 { \musique } \break
    \repeat unfold 6 { \musique } \break
    \repeat unfold 4 { \musique } \break
    \repeat unfold 2 { \musique }
  }
  \layout {
    \context {
      \Score
      \override NonMusicalPaperColumn #'line-break-permission = ##f
      \override NonMusicalPaperColumn #'page-break-permission = ##f
    }
  }
}
```





Voir aussi

Morceaux choisis : Section “Espacements” dans *Morceaux choisis*.

4.3.7 Recours à une voix supplémentaire pour gérer les sauts

La plupart du temps, les informations concernant les sauts de ligne ou de page se retrouvent directement au milieu des notes.

```
musique = \relative c'' { c4 c c c }
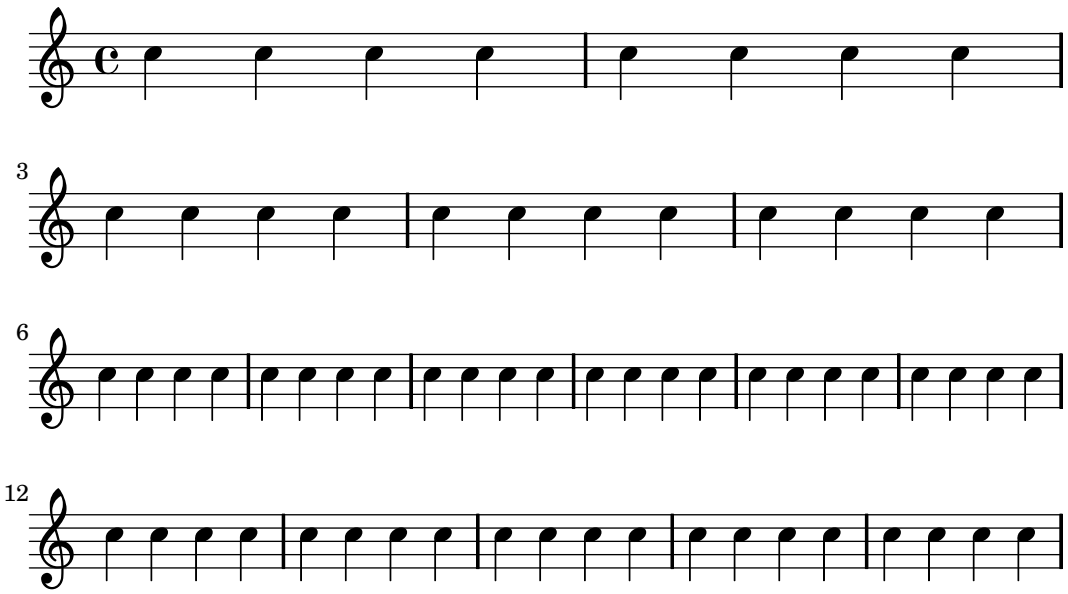
\score {
  \new Staff {
    \repeat unfold 2 { \musique } \break
    \repeat unfold 3 { \musique }
  }
}
```

Bien que cela constitue un moyen aisé de saisir les commandes `\break` et `\pageBreak`, les données musicales se retrouvent mélangées à des informations qui concernent plutôt l’agencement de la musique sur le papier. Vous pouvez tout à fait séparer ce qui est purement musical et les informations concernant les sauts de ligne ou de page en créant une voix supplémentaire dédiée. Cette voix spécifique ne contiendra que des blancs – des silences invisibles `\skip` –, des `\break`, des `\pageBreak` et autres informations concernant les ruptures.

```
music = \relative c'' { c4 c c c }

\score {
  \new Staff <<
    \new Voice {
      s1 * 2 \break
      s1 * 3 \break
      s1 * 6 \break
      s1 * 5 \break
    }
    \new Voice {
```

```
      \repeat unfold 2 { \music }
      \repeat unfold 3 { \music }
      \repeat unfold 6 { \music }
      \repeat unfold 5 { \music }
    }
  >>
}
```



Cette manière de procéder est tout à fait indiquée lorsque vous ajustez les `line-break-system-details` et autres propriétés fort intéressantes de `NonMusicalPaperColumnGrob`, comme vous pouvez le voir au chapitre Section 4.4 [Espacement vertical], page 432.

```
music = \relative c'' { c4 c c c }
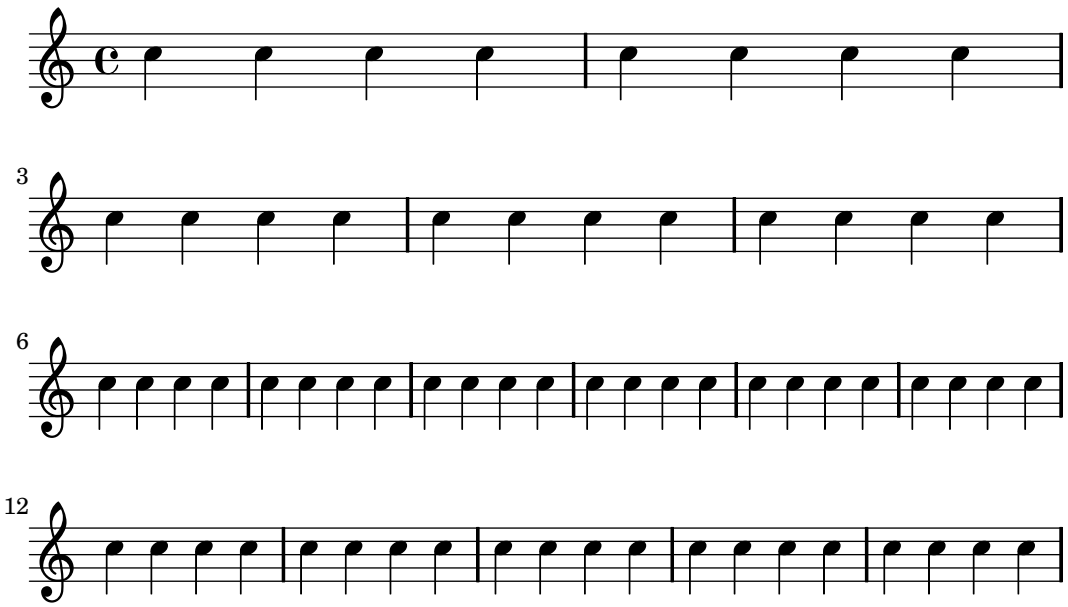
\score {
  \new Staff <<
    \new Voice {
      \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 0))
      s1 * 2 \break

      \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 35))
      s1 * 3 \break

      \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 70))
      s1 * 6 \break

      \overrideProperty "Score.NonMusicalPaperColumn"
        #'line-break-system-details #'((Y-offset . 105))
      s1 * 5 \break
    }
    \new Voice {
      \repeat unfold 2 { \music }
```

```
      \repeat unfold 3 { \music }
      \repeat unfold 6 { \music }
      \repeat unfold 5 { \music }
    }
  >>
}
```



Voir aussi

Manuel de notation : Section 4.4 [Espacement vertical], page 432.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.4 Espacement vertical

L'espacement vertical dépend de trois éléments : la surface disponible – par exemple format de papier et marges –, l'espace qui doit séparer les systèmes, et l'espace qui sépare les portées d'un même système.

4.4.1 Espacement vertical au sein d'un système

LilyPond dispose de trois différents mécanismes permettant de contrôler l'espacement au sein d'un système selon trois catégories :

- *portées isolées*,
- *portées regroupées* (portées d'un même groupe, telles celles d'un `ChoirStaff`, etc.), et
- *lignes de non-portée* (`Lyrics`, `ChordNames`, etc.).

La hauteur de chaque système se détermine en deux phases. Les portées sont tout d'abord espacées selon la surface disponible. Puis les lignes autres que des portées, comme les paroles ou les accords, sont réparties entre les portées.

Les paragraphes qui suivent traitent exclusivement de la manière de gérer l'espacement entre les lignes d'un système – portée musicale ou non. Pour ce qui a trait aux espacements entre les systèmes, mouvements, annotations et marge, ils sont contrôlés par des variables attachées au bloc `\paper` et font l'objet du chapitre Section 4.1.4 [Variables d'espacement vertical fluctuant], page 415.

Propriétés d'espacement au sein d'un système

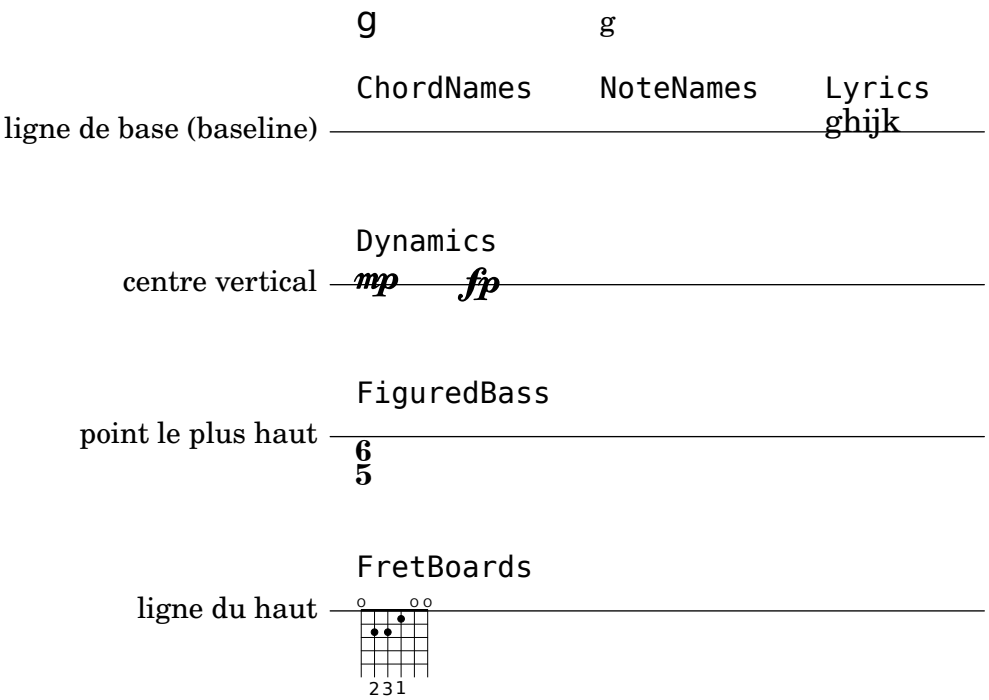
L'espacement entre les portées est géré par deux jeux de propriétés d'objet graphique (*grob*). Le premier, associé à l'objet graphique `VerticalAxisGroup`, est créé pour toute ligne de portée ou de non-portée. Le second, associé à l'objet graphique `StaffGrouper`, doit être explicitement créé pour un regroupement de portées particulier. Les propriétés qui leur sont attachées sont abordées en fin de section.

Le nom de ces propriétés, sauf `staff-affinity`, suit le schéma *`item1`*`-`*`item2`*`-spacing` – *`item1`* et *`item2`* étant les éléments à espacer. Notez bien que *`item2`* n'est pas forcément placé au-dessous : c'est le cas pour la propriété `nonstaff-relatedstaff-spacing` qui spécifie la distance entre une ligne de non-portée alors que sa `staff-affinity` a été déterminée à `#UP`.

Toutes ces distances sont mesurées entre les points de référence respectifs de éléments considérés. Le *point de référence* d'une portée est le centre vertical du `StaffSymbol` – la ligne médiane si `line-count` est impair, l'interligne médian si `line-count` est pair. Quant aux lignes rattachées à des portées – lignes de non-portée – le tableau suivant présente le *point de référence* pour chacun d'entre eux :

Ligne de non-portée	**Point de référence**
`ChordNames`	ligne de base
`NoteNames`	ligne de base
`Lyrics`	ligne de base
`Dynamics`	centre vertical
`FiguredBass`	point le plus haut
`FretBoards`	ligne supérieure

En voici une représentation graphique :



Hormis `staff-affinity` – propriété attachée au *grob* `VerticalAxisGroup` –, chacune de ces propriétés est enregistrée sous la forme d'une liste associative dont la structure est identique à celle des variables du bloc `\paper` que nous avons examinées au chapitre Section 4.1.4 [Variables

d'espacement vertical fluctuant], page 415. Les particularités en matière de modification d'une liste associative font l'objet d'un Section "chapitre particulier" dans *Manuel de notation*. Les propriétés des objets graphiques se règlent avec un `\override` mentionné dans un bloc `\score` ou `\layout`, pas dans le bloc `\paper`.

L'exemple suivant illustre deux façons de modifier une liste associative. La première déclaration n'agit que sur une seule clé, alors que la seconde redéfinit la propriété dans son intégralité.

```
\new Staff \with {
  \override VerticalAxisGroup #'staff-staff-spacing #'basic-distance = #10
} { ... }

\new Staff \with {
  \override VerticalAxisGroup #'staff-staff-spacing =
    #'(('basic-distance  . 10)
       (minimum-distance . 9)
       (padding . 1)
       (stretchability . 10))
} { ... }
```

La modification d'un espacement au niveau global se mentionne au sein du bloc `\layout` :

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup #'staff-staff-spacing #'basic-distance = #10
  }
}
```

Les réglages concernant les propriétés d'espacement vertical des objets graphiques sont répertoriées aux chapitres Section "VerticalAxisGroup" dans *Référence des propriétés internes* et Section "StaffGrouper" dans *Référence des propriétés internes*. Les propriétés relatives aux lignes de non-portée sont répertoriées selon la définition de leur contexte dans la Section "Contexts" dans *Référence des propriétés internes*.

Propriétés de l'objet `VerticalAxisGroup`

Les propriétés de l'objet `VerticalAxisGroup` s'ajustent à l'aide d'un `\override` au niveau d'un contexte `Staff` (ou son équivalent).

`staff-staff-spacing`

Il s'agit de la distance entre la portée en cours et la portée qui suit au sein du même regroupement, qu'il y ait ou non une ligne de non-portée (`Lyrics` ou autre) entre les deux. Cette propriété ne s'applique pas à la dernière portée d'un système.

En tout état de cause, la fonction Scheme `staff-staff-spacing` d'un `VerticalAxisGroup` affectera les propriétés du `StaffGrouper` si la portée est incluse dans un regroupement ; elle s'appliquera au `default-staff-staff-spacing` en l'absence de regroupement. Les portées peuvent donc s'aligner différemment selon qu'elles sont ou non regroupées. Pour obtenir le même espacement sans tenir compte dees éventuels regroupements, cette fonction peut faire place à une complète redéfinition des espacements fluctuants à l'aide de règles dérogatoires comme vu précédemment.

`default-staff-staff-spacing`

Il s'agit de la distance qui s'appliquera par défaut aux portées isolées, à loins que `staff-staff-spacing` n'ait été redéfini explicitement par un `\override`.

`staff-affinity`
: Il s'agit de la direction – `UP`, `DOWN` ou `CENTER` – que prendra une ligne de non-portée pour aller s'accoler aux portées adjacentes . Si vous lui attribuez `CENTER`, cette ligne de non-portée ira se placer à équidistance entre les portées qui l'encadrent, tout en tenant compte des éventuels risques de collision et des autres contraintes d'espacement. Des lignes de non-portée adjacentes devraient avoir une `staff-affinity` allant de haut en bas – autrement dit, pas de `UP` après un `DOWN`. Une ligne de non-portée en dessous d'un système devrait avoir sa `staff-affinity` définie à `UP`. De la même manière, lorsque cette ligne surplombe un système, sa `staff-affinity` devrait être définie à `DOWN`. Prenez garde à la valeur que vous affectez à `staff-affinity` : si vous lui affectez la valeur `#f` à une ligne de non-portée, cette ligne sera considérée comme étant une portée ; à l'inverse, utiliser la propriété `staff-affinity` pour une portée lui fera perdre cette qualité.

`nonstaff-relatedstaff-spacing`
: Il s'agit de la distance entre la ligne de non-portée en cours et la portée la plus proche selon la `staff-affinity`, à la double condition qu'il n'y a pas déjà une autre ligne de non-portée et que la valeur de `staff-affinity` soit `UP` ou `DOWN`. Lorsque la valeur de `staff-affinity` est égale à `CENTER`, la valeur de `nonstaff-relatedstaff-spacing` servira à centrer la ligne de non-portée entre les deux portées adjacentes même si une autre non-portée est présente (quelque soit le côté).

`nonstaff-nonstaff-spacing`
: Il s'agit de la distance entre deux lignes de non-portée selon l'orientation définie par `staff-affinity` et qu'elles ont la même orientation. Bien entendu, ceci ne peut concerner que les valeurs `UP` et `DOWN` de `staff-affinity`.

`nonstaff-unrelatedstaff-spacing`
: Il s'agit de la distance entre une ligne de non-portée et la portée à l'opposé de l'orientation adoptée, à la double condition qu'il n'y a pas déjà une autre ligne de non-portée et que la valeur de `staff-affinity` soit `UP` ou `DOWN`. Cette propriété trouve toute sa légitimité pour décaler une ligne de `Lyrics` de la portée à laquelle elle ne correspond pas.

Propriétés de l'objet `StaffGrouper`

Les propriétés de l'objet `StaffGrouper` s'ajustent à l'aide d'un `\override` au niveau d'un contexte `StaffGroup` (ou son équivalent).

`staff-staff-spacing`
: Il s'agit de la distance entre deux portées consécutives d'un même système. La propriété `staff-staff-spacing` de l'objet `VerticalAxisGroup` d'un portée en particulier peut se redéfinir à l'aide de règles dérogatoires.

`staffgroup-staff-spacing`
: Il s'agit de la distance entre la dernière portée d'un regroupement et la portée suivante, au sein d'un même système, y compris lorsqu'une ou plusieurs lignes de non-portée (tel `Lyrics`) s'insèrent entre les deux. Cette propriété ne concerne pas la dernière portée d'un système. Dans le cas où la propriété `staff-staff-spacing` d'une portée du regroupement a été ajustée au niveau de son propre `VerticalAxisGroup`, cette dernière aura préséance.

Voir aussi

Fichiers d'initialisation : '`ly/engraver-init.ly`', '`scm/define-grobs.scm`'.

Référence des propriétés internes : Section "Contexts" dans *Référence des propriétés internes*, Section "VerticalAxisGroup" dans *Référence des propriétés internes*, Section "StaffGrouper" dans *Référence des propriétés internes*.

Espacement de portées isolées

Les `Staff`, `DrumStaff`, `TabStaff` entre autres sont des contextes de « portée » pouvant contenir plusieurs voix, mais pas une portée.

L'espacement de ces *portées isolées* est géré par les propriétés suivantes :

- Propriétés du `VerticalAxisGroup` :
  - `default-staff-staff-spacing`
  - `staff-staff-spacing`

Ces propriétés d'objet graphique sont expliquées une à une au chapitre [Propriétés d'espacement au sein d'un système], page 433.

Certaines propriétés supplémentaires s'appliqueront dès lors que ces portées sont regroupées – voir [Espacement de portées regroupées], page 437.

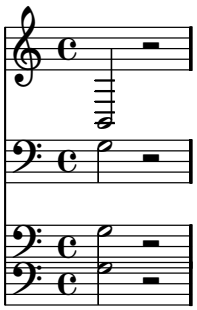
L'exemple suivant illustre la manière de gérer l'espacement de portées isolées à l'aide de la propriété `default-staff-staff-spacing`. Les mêmes règles appliquées de manière dérogatoire au `staff-staff-spacing` produiront les mêmes effets, y comris au sein de regroupements.

```
\layout {
  \context {
    \Staff
    \override VerticalAxisGroup #'default-staff-staff-spacing =
      #'((basic-distance . 8)
         (minimum-distance . 7)
         (padding . 1))
  }
}

<<
  % The very low note here needs more room than 'basic-distance
  % can provide, so the distance between this staff and the next
  % is determined by 'padding.
  \new Staff { b,2 r | }

  % Here, 'basic-distance provides enough room, and there is no
  % need to compress the space (towards 'minimum-distance) to make
  % room for anything else on the page, so the distance between
  % this staff and the next is determined by 'basic-distance.
  \new Staff { \clef bass g2 r | }

  % By setting 'padding to a negative value, staves can be made to
  % collide.  The lowest acceptable value for 'basic-distance is 0.
  \new Staff \with {
    \override VerticalAxisGroup #'default-staff-staff-spacing =
      #'((basic-distance . 3.5)
         (padding . -10))
  } { \clef bass g2 r | }
  \new Staff { \clef bass g2 r | }
>>
```



Voir aussi

Fichiers d'initialisation : '`scm/define-grobs.scm`'.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Référence des propriétés internes : Section "VerticalAxisGroup" dans *Référence des propriétés internes*.

Espacement de portées regroupées

Dans les partitions orchestrales ou de grande ampleur, il arrive souvent que des portées soient regroupées. L'espacement est alors plus important entre deux regroupements qu'entre les portées d'un même groupe.

Les *regroupement de portées* tels le `StaffGroup` ou le `ChoirStaff` sont des contextes qui peuvent contenir simultanément une ou plusieurs portées.

L'espacement entre les portées d'un même regroupement est géré par les propriétés suivantes :

- Propriétés du `VerticalAxisGroup` :
  - `staff-staff-spacing`
- Propriétés du `StaffGrouper` :
  - `staff-staff-spacing`
  - `staffgroup-staff-spacing`

Ces propriétés d'objet graphique sont expliquées une à une au chapitre [Propriétés d'espacement au sein d'un système], page 433.

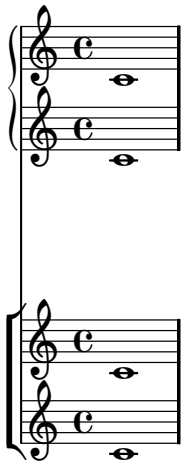
L'exemple suivant illustre la manière de gérer l'espacement de portées regroupes, à l'aide des propriétés de l'objet graphique `StaffGrouper` :

```
\layout {
  \context {
    \Score
    \override StaffGrouper #'staff-staff-spacing #'padding = #0
    \override StaffGrouper #'staff-staff-spacing #'basic-distance = #1
  }
}

<<
  \new PianoStaff \with {
    \override StaffGrouper #'staffgroup-staff-spacing #'basic-distance = #20
  } <<
    \new Staff { c'1 }
    \new Staff { c'1 }
  >>

  \new StaffGroup <<
    \new Staff { c'1 }
    \new Staff { c'1 }
```

```
  >>
>>
```



Voir aussi

Fichiers d'initialisation : '`scm/define-grobs.scm`'.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Référence des propriétés internes: Section "VerticalAxisGroup" dans *Référence des propriétés internes*, Section "StaffGrouper" dans *Référence des propriétés internes*.

Espacement des lignes rattachées à des portées

Les *lignes de non-portée*, comme les `Lyrics` ou les `ChordNames` sont des contextes dont les objets de rendu sont gravés à l'instar des portées – une ligne horizontale dans un système. En fait, les lignes de non-portée sont des contextes qui vont créer un objet de rendu `VerticalAxisGroup`.

L'espacement des lignes de non-portée est géré par les propriétés suivantes :

- Propriétés du `VerticalAxisGroup` :
  - `staff-affinity`
  - `nonstaff-relatedstaff-spacing`
  - `nonstaff-nonstaff-spacing`
  - `nonstaff-unrelatedstaff-spacing`

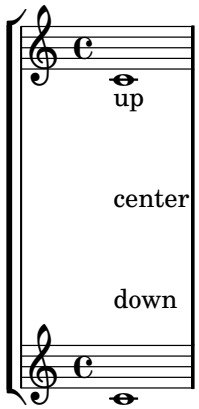
Ces propriétés d'objet graphique sont expliquées une à une au chapitre [Propriétés d'espacement au sein d'un système], page 433.

L'exemple suivant utilise la propriété `nonstaff-nonstaff-spacing` pour gérer l'espacement entre des lignes consécutives de non-portée. Vous noterez que la valeur élevée attribuée à la clé `stretchability` permet aux paroles de s'étirer plus que de raison.

```
\layout {
  \context {
    \Lyrics
    \override VerticalAxisGroup
      #'nonstaff-nonstaff-spacing #'stretchability = #1000
  }
}

\new StaffGroup
<<
```

```
  \new Staff \with {
    \override VerticalAxisGroup #'staff-staff-spacing = #'((basic-distance . 30))
  } { c'1 }
  \new Lyrics \with {
    \override VerticalAxisGroup #'staff-affinity = #UP
  } \lyricmode { up }
  \new Lyrics \with {
    \override VerticalAxisGroup #'staff-affinity = #CENTER
  } \lyricmode { center }
  \new Lyrics \with {
    \override VerticalAxisGroup #'staff-affinity = #DOWN
  } \lyricmode { down }
  \new Staff { c'1 }
>>
```



Voir aussi

Fichiers d'initialisation : '`ly/engraver-init.ly`', '`scm/define-grobs.scm`'.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Contexts" dans *Référence des propriétés internes*, Section "VerticalAxisGroup" dans *Référence des propriétés internes*.

4.4.2 Positionnement explicite des portées et systèmes

Pour bien comprendre comment fonctionnent les réglages de `VerticalAxisGroup` et de `\paper` abordés dans les deux rubriques précédentes, rien ne vaut une collection d'exemples illustrant les différentes mises au point du décalage vertical appliqué aux portées et systèmes distribués sur une page.

Une autre approche de l'espacement vertical est le recours à `NonMusicalPaperColumn #'line-break-system-details`. Alors que `VerticalAxisGroup` et `\paper` gèrent un décalage vertical, `NonMusicalPaperColumn #'line-break-system-details` spécifiera le positionnement vertical absolu sur la page.

`NonMusicalPaperColumn #'line-break-system-details` prend en charge une liste associative de trois mises au point :

- `X-offset`
- `Y-offset`
- `alignment-distances`

Les dérogations en matière d'objet graphique, y compris celles concernant les `NonMusicalPaperColumn` ci-dessus, peuvent se placer à trois différents endroits de votre fichier source :

- directement au beau milieu des notes
- au sein d'un bloc `\context`
- dans un bloc `\with`

Le réglage de `NonMusicalPaperColumn` s'effectue à l'aide d'une simple commande `\override` au sein d'un bloc `\context` ou `\with`. Dans le cas ou il est stipulé au fil des notes, c'est la commande spécifique `\overrideProperty` qui doit intervenir. Voici quelques exemples de réglages de `NonMusicalPaperColumn` à l'aide de la commande `\overrideProperty` :

```
\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20))

\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((Y-offset . 40))
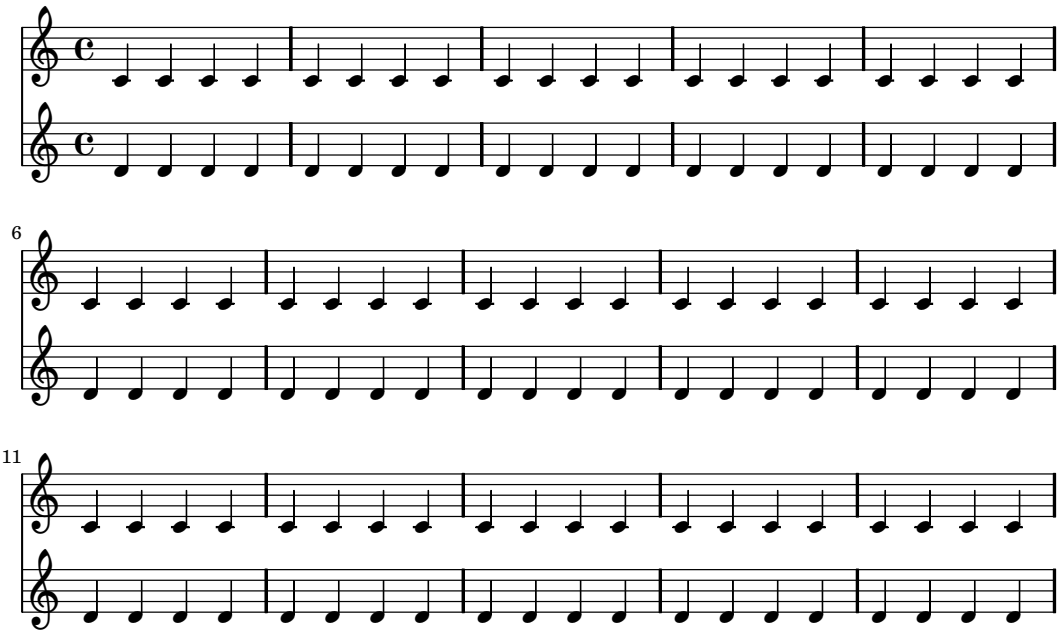
\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20)
                                 (Y-offset . 40))

\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((alignment-distances . (15)))

\overrideProperty NonMusicalPaperColumn
  #'line-break-system-details #'((X-offset . 20)
                                 (Y-offset . 40)
                                 (alignment-distances . (15)))
```

Nous allons maintenant voir ces différents réglages en action. Commençons par examiner un exemple dépourvu de toute mise au point.
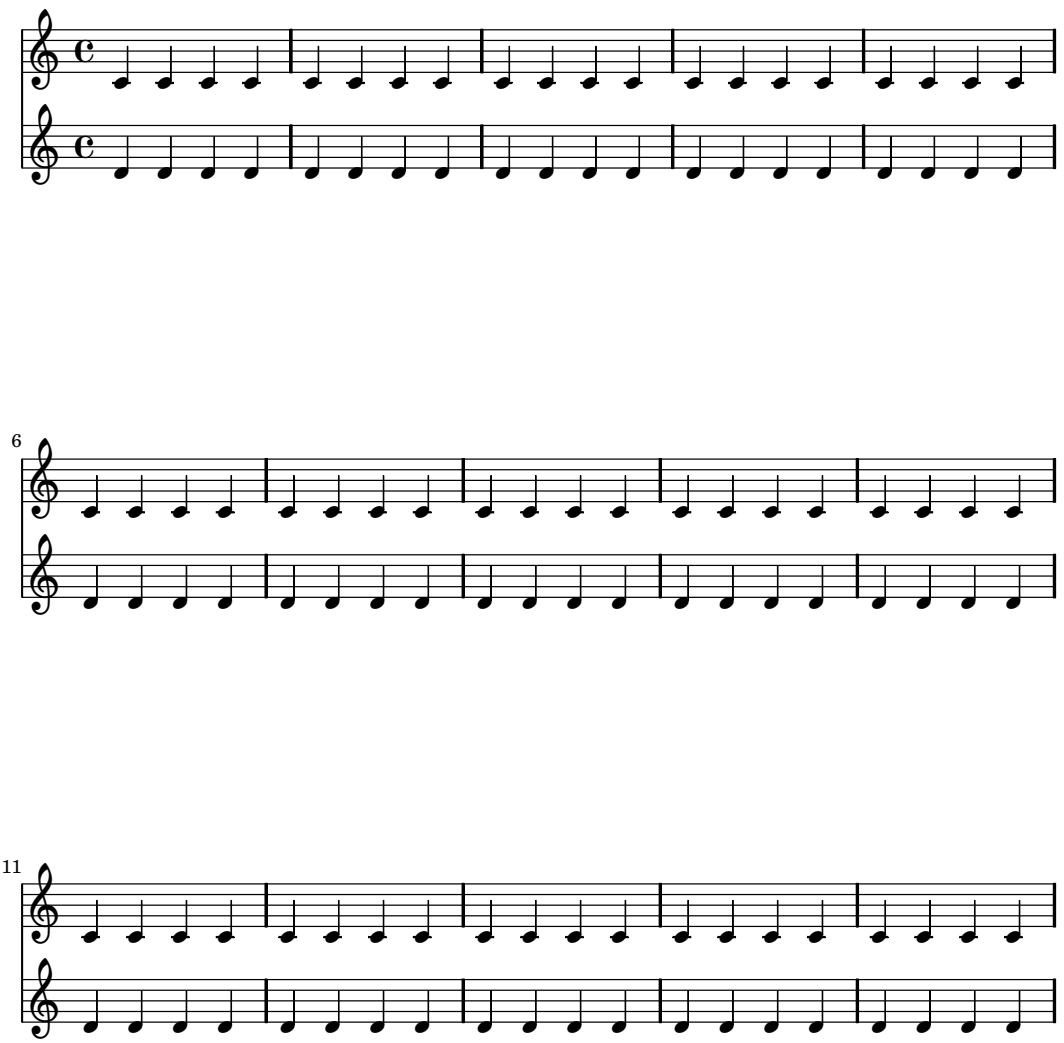
```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          s1*5 \break
          s1*5 \break
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```



Cette partition isole les informations de saut de ligne ou de page dans une voix spécifique. La mise en forme est ainsi séparée des événements musicaux ; ceci nous permettra d'y voir plus clair au fur et à mesure que nous avancerons. Pour plus de précisions, relisez Section 4.3.7 [Recours à une voix supplémentaire pour gérer les sauts], page 430.

Les `\breaks` explicites répartissent la musique en lignes de cinq mesures chacune. L'espacement vertical est celui que LilyPond attribue par défaut. Nous pouvons, afin de fixer explicitement le point d'attache vertical de chacun des systèmes, définir un doublet `Y-offset` en tant qu'attribut du `line-break-system-details` de l'objet `NonMusicalPaperColumn` :

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 0))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 40))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 80))
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```
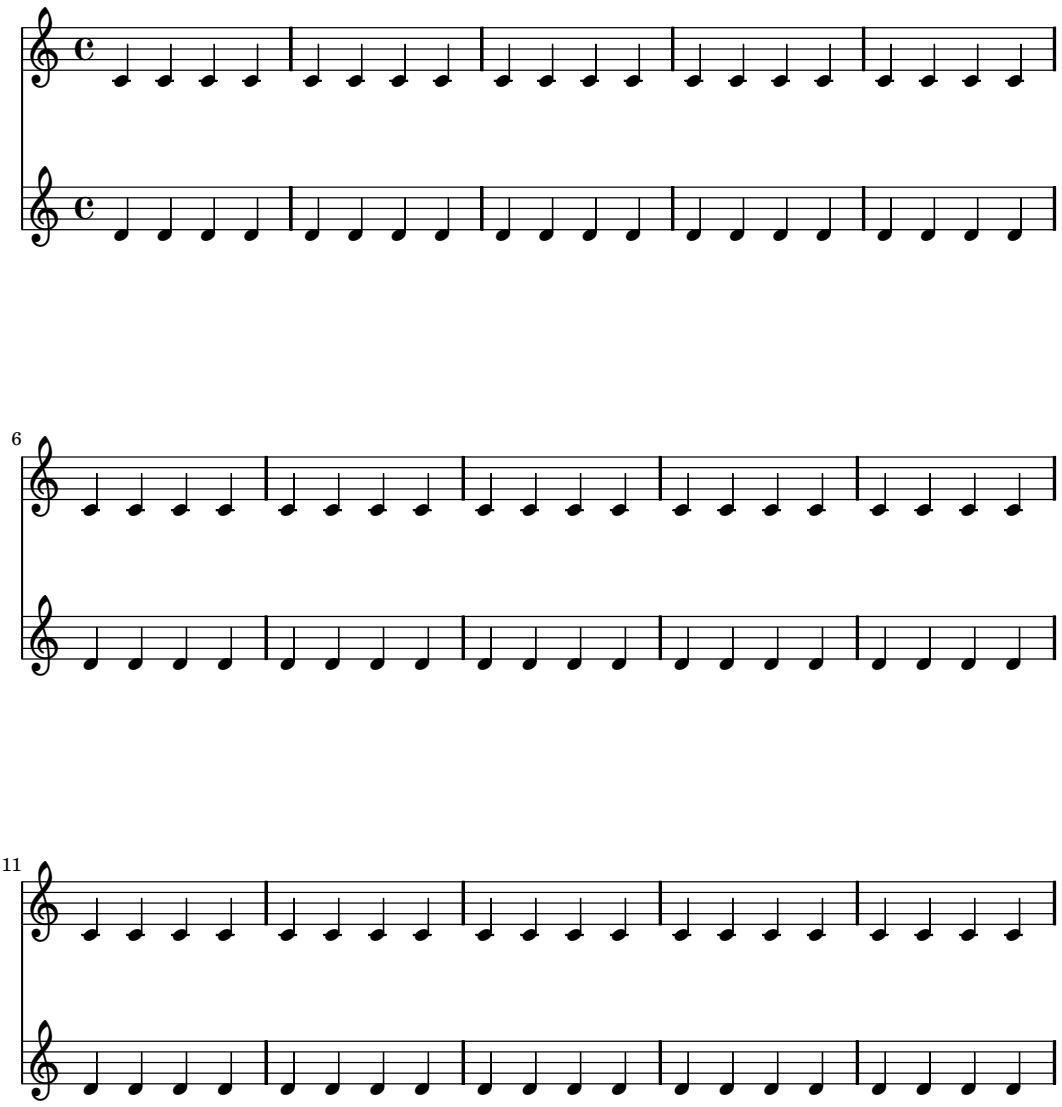


Vous aurez remarqué que nous n'avons déterminé qu'une seule valeur, même si la liste associative de `line-break-system-details` peut en comporter un certain nombre. Vous aurez aussi noté que la propriété `Y-offset` détermine ici le point de départ de chacun des systèmes de la page.

Maintenant que chaque système est explicitement positionné, nous pouvons jouer sur la distance séparant les portées de chacun des systèmes, grâce à la sous-propriété `alignment-distances` de `line-break-system-details`.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 20)
                                           (alignment-distances . (15)))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 60)
                                           (alignment-distances . (15)))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 100)
```

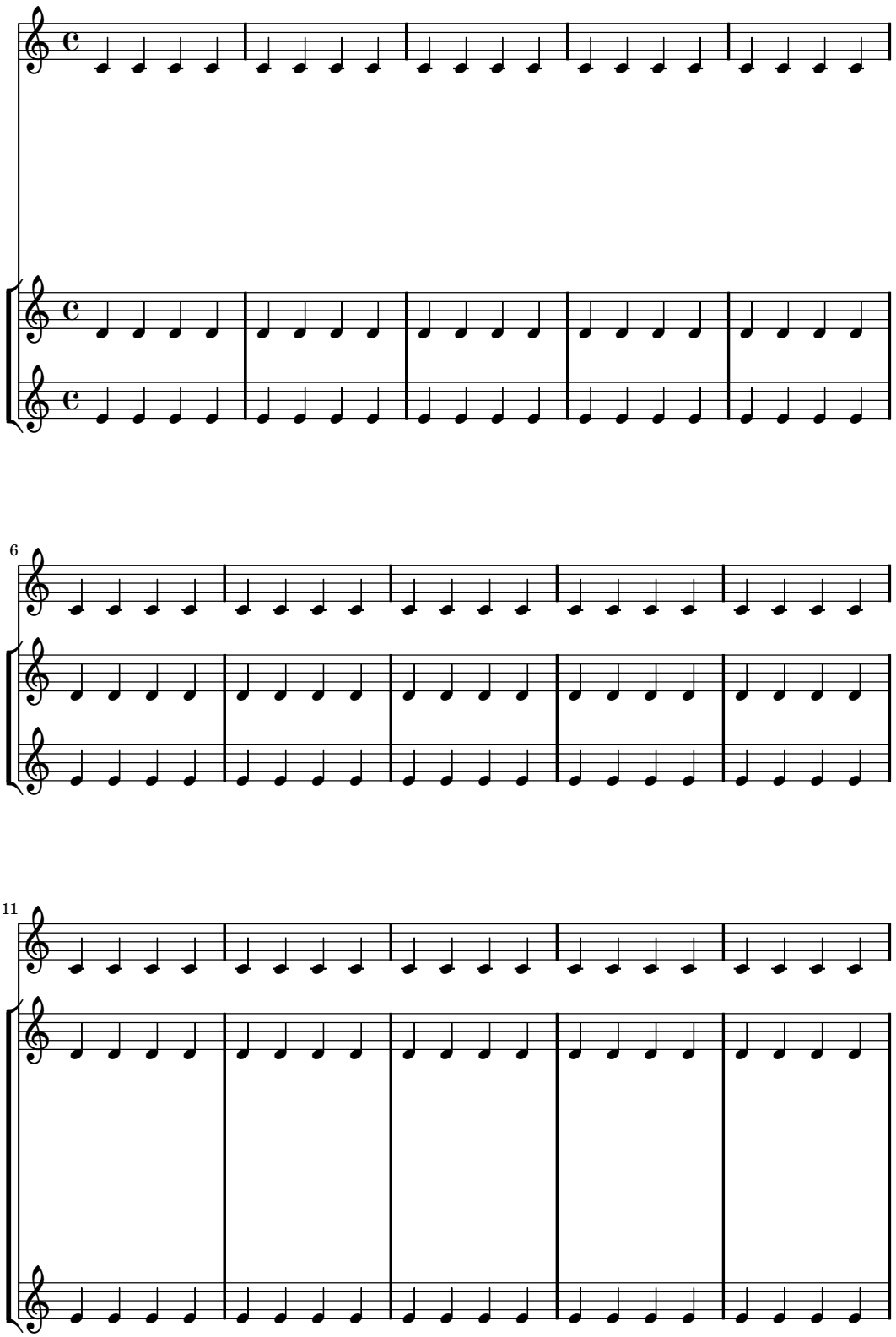
```
                                        (alignment-distances . (15)))
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new Staff {
        \repeat unfold 15 { d'4 d' d' d' }
      }
    >>
  }
}
```



Nous avons maintenant assigné deux valeurs différentes à l'attribut `line-break-system-details` de l'objet `NonMusicalPaperColumn`. `line-break-system-details` pourrait prendre

bien d'autres paramètres d'espacement, y compris un doublet `X-offset`, mais nous n'avons utilisé que `Y-offset` et `alignment-distances` pour contrôler le positionnement de chaque système et de chaque portée. Vous noterez enfin que `alignment-distances` traite le positionnement des portées, non d'un regroupement de portées.

```
\header { tagline = ##f }
\paper { left-margin = 0\mm }
\book {
  \score {
    <<
      \new Staff <<
        \new Voice {
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 0)
                                           (alignment-distances . (30 10)))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 60)
                                           (alignment-distances . (10 10)))
          s1*5 \break
          \overrideProperty #"Score.NonMusicalPaperColumn"
            #'line-break-system-details #'((Y-offset . 100)
                                           (alignment-distances . (10 30)))
          s1*5 \break
        }
        \new Voice { \repeat unfold 15 { c'4 c' c' c' } }
      >>
      \new StaffGroup <<
        \new Staff { \repeat unfold 15 { d'4 d' d' d' } }
        \new Staff { \repeat unfold 15 { e'4 e' e' e' } }
      >>
    >>
  }
}
```



Quelques points à prendre en considération :

- Lorsque vous utilisez `alignment-distances`, les paroles et autres lignes de non-portée ne comptent pas pour une portée.
- Les nombres fournis à `X-offset`, `Y-offset` et `alignment-distances` sont considérés comme des multiples de la distance entre des portées adjacentes. Des valeurs positives remontent les portées et paroles, des valeurs négatives les descendent.
- Dans la mesure où `NonMusicalPaperColumn #'line-break-system-details` permet de positionner systèmes et portées n'importe où sur une page, vous pourriez être en contra-

diction avec les dimensionnements de la feuille ou bien aboutir à des surimpressions. Soyez donc raisonnables quant aux différentes valeurs que vous affectez à ces réglages.

Voir aussi

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

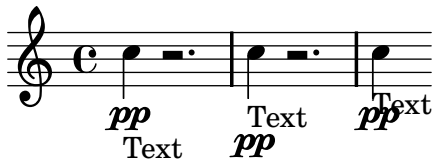
4.4.3 Résolution des collisions verticales

Vous savez de manière intuitive qu'un certain nombre d'objets en matière de notation musicale appartiennent à la portée, et que d'autres se placent en dehors de la portée. Entre autres objets externes, nous avons les marques repères, les textes et les nuances ; nous les appellerons « objets extérieurs à la portée ». La règle qu'applique LilyPond pour positionner verticalement ces objets extérieurs consiste à les placer au plus près de la portée tout en prenant garde d'éviter qu'il y ait chevauchement.

LilyPond utilise la propriété `outside-staff-priority` afin de déterminer si un objet est ou non un objet extérieur à la portée : lorsque la valeur de `outside-staff-priority` est numérique, il s'agit d'un objet extérieur à la portée. De plus, la propriété `outside-staff-priority` indique à LilyPond l'ordre dans lequel ces objets doivent être disposés.

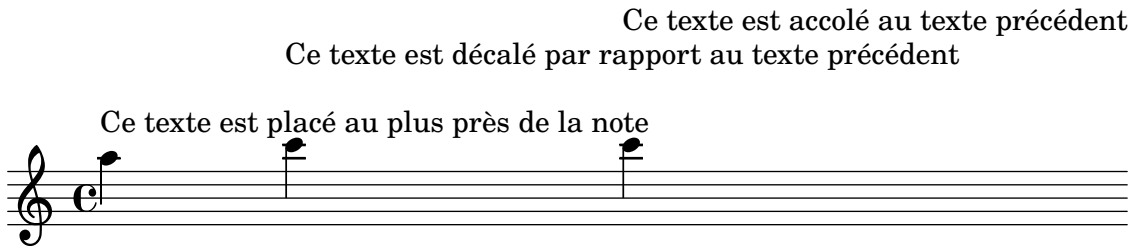
Tout d'abord, LilyPond dispose tous les objets qui ne sont pas externes. Les objets extérieurs à la portée sont alors triés selon l'ordre croissant de leur `outside-staff-priority`. Enfin, LilyPond prend chacun des ces objets et les positionne de telle sorte qu'il n'entrent pas en collision avec ceux qui on déjà été placés. Autrement dit, lorsque deux objets devraient se placer au même endroit, celui dont la `outside-staff-priority` est la plus faible sera disposé au plus près de la portée.

```
c4_"Text"\pp
r2.
\once \override TextScript #'outside-staff-priority = #1
c4_"Text"\pp % this time the text will be closer to the staff
r2.
% by setting outside-staff-priority to a non-number,
% we disable the automatic collision avoidance
\once \override TextScript #'outside-staff-priority = ##f
\once \override DynamicLineSpanner #'outside-staff-priority = ##f
c4_"Text"\pp % now they will collide
```



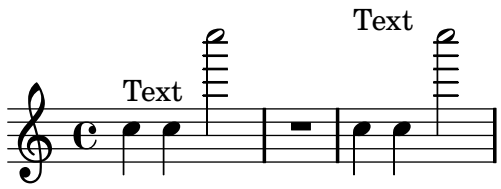
Le décalage vertical entre un objet extérieur à la portée et celui qui le précède se contrôle par la propriété `outside-staff-padding`.

```
\once \override TextScript #'outside-staff-padding = #0
a'^"Ce texte est placé au plus près de la note"
\once \override TextScript #'outside-staff-padding = #3
c^"Ce texte est décalé par rapport au texte précédent"
c^"Ce texte est accolé au texte précédent"
```



Par défaut, les objets extérieurs à la portée sont positionnés en évitant les collisions horizontales avec des objets précédemment positionnés. Ceci peut cependant générer des situations où des objets se trouvent horizontalement trop proches. L'espacement vertical entre les portées peut dans une certaine mesure permettre de mieux intercaler les objets extérieurs à la portée. La propriété `outside-staff-horizontal-padding` repoussera verticalement un objet pour éviter ce genre de situation.

```
% the markup is too close to the following note
c4^"Text"
c4
c''2
% setting outside-staff-horizontal-padding fixes this
R1
\once \override TextScript #'outside-staff-horizontal-padding = #1
c,,4^"Text"
c4
c''2
```



Voir aussi

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.
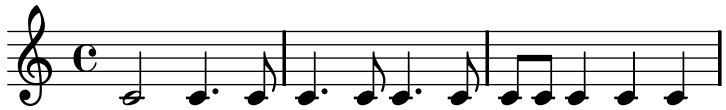
4.5 Espacement horizontal

4.5.1 Généralités sur l'espacement horizontal

Le moteur d'espacement traduit les différences de durée en distances étirables (*springs* pour ressorts) de différente longueur. Des durées importantes prennent ainsi plus de place que des durées moins longues. Les durées les plus courtes se verront attribuer un espace fixe, contrôlé par la propriété `shortest-duration-space` de l'objet Section "SpacingSpanner" dans *Référence des propriétés internes*. Au plus la durée s'allonge, au plus elle prendra d'espace : le doublement d'une durée attribuera à la note un espace fixé d'après la propriété `spacing-increment`.

L'exemple suivant comporte des blanches, des noires et un certain nombre de croches. La croche est suivie d'un espace de la largeur d'une tête de note ; pour la noire , cette espace est de deux têtes ; il est de trois pour la blanche.

```
c2 c4. c8 c4. c8 c4. c8 c8
c8 c4 c4 c4
```



`spacing-increment` est normalement défini à 1,2 espace de portée – ce qui correspond à peu près à la largeur d'une tête de note – et `shortest-duration-space` à 2,0. La note la plus courte s'étendra donc sur l'équivalent de 2,4 espaces de portée (2 fois le `spacing-increment`). Le point de départ de cet espace se situe à l'extrémité gauche du symbole ; la note la plus courte est donc suivie en général d'un espace égal à la largeur d'une tête de note.
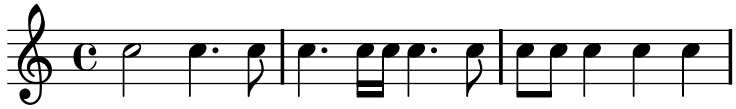
Si l'on suit à la lettre ce qui précède, ajouter une simple triple croche à une partition qui comporte déjà des croches et des doubles augmentera considérablement son volume : la durée la plus courte n'est plus la double mais la triple croche, ce qui aura pour conséquence d'ajouter une largeur de tête à chacune des notes. Pour s'affranchir de cet effet quelque peu pervers, la durée la plus courte prise en considération au niveau de l'espacement n'est pas la note la plus brève de la partition, mais celle qui apparaît le plus souvent.

La courte durée la plus fréquente est déterminée à partir de la note la plus courte de chaque mesure. C'est elle qui servira de base pour l'espacement, à cette nuance près que la plus courte durée ne saurait être strictement supérieure à la croche. Cette « durée de référence » est d'ailleurs affichée lorsque vous lancez `lilypond` avec l'option `--verbose`.
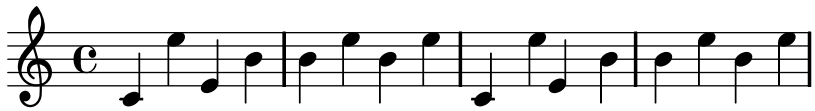
Ces durées peuvent être adaptées. Vous pouvez définir la durée de base pour les espacements grâce à la propriété `common-shortest-duration` de l'objet Section "SpacingSpanner" dans *Référence des propriétés internes*. La durée maximale de cet étalon, normalement la croche, est gérée par la propriété `base-shortest-duration`.

Les notes plus courtes que la note témoin sont suivies d'un espace proportionnel à la durée témoin. Si donc nous ajoutions quelques doubles croches à l'exemple précédent, elles seraient suivies d'une demie largeur de tête :

```
c2 c4. c8 c4. c16[ c] c4. c8 c8 c8 c4 c4 c4
```



Dans notre *Essai sur la gravure musicale automatisée*, nous avons vu comment la direction des hampes peut influencer l'espacement – voir Section "Espacement" dans *Essai*. Ceci est contrôlé par la propriété `stem-spacing-correction` de l'objet Section "NoteSpacing" dans *Référence des propriétés internes*, créé pour chaque contexte Section "Voice" dans *Référence des propriétés internes*. L'objet Section "StaffSpacing" dans *Référence des propriétés internes*, généré au niveau d'un contexte `Staff`, possède une même propriété qui contrôlera l'espacement hampe-barre de mesure. L'exemple suivant montre ces adaptations, tout d'abord selon les réglages par défaut, puis avec des corrections forcées.



L'espacement spécifique à la notation proportionnelle fait l'objet d'une Section "rubrique dédiée" dans *Manuel de notation*.

Voir aussi

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

Référence des propriétés internes : Section "SpacingSpanner" dans *Référence des propriétés internes*, Section "NoteSpacing" dans *Référence des propriétés internes*, Section "StaffSpacing" dans *Référence des propriétés internes*, Section "NonMusicalPaperColumn" dans *Référence des propriétés internes*.

Essai sur la gravure musicale automatisée : Section "Espacement" dans *Essai*.

Problèmes connus et avertissements

Il n'existe pas de mécanisme simple et efficace qui permette de forcer manuellement l'espacement. La solution ci-dessous permet cependant « d'aérer » artificiellement une partition ; il vous suffit d'ajuster la valeur du décalage *(padding)* autant que de besoin.

```
\override Score.NonMusicalPaperColumn #'padding = #10
```
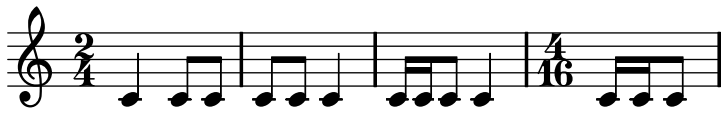
Il n'y a aucun moyen de diminuer l'espacement.

4.5.2 Changement d'espacement en cours de partition

Il arrive, au cours d'un même mouvement, qu'une nouvelle partie modifie substantiellement la notion de valeur brève et valeur longue. La commande `newSpacingSection` permet alors de réinitialiser les paramètres d'espacement.

Dans l'exemple qui suit, le changement de métrique marque le début d'une nouvelle partie ; remarquez comme les doubles-croches sont alors plus espacées :

```
\time 2/4
c4 c8 c
c8 c c4 c16[ c c8] c4
\newSpacingSection
\time 4/16
c16[ c c8]
```



La commande `\newSpacingSection` crée un nouvel objet `SpacingSpanner`, ce qui permet d'appliquer de nouveaux `\overrides` à partir de ce point.

Voir aussi
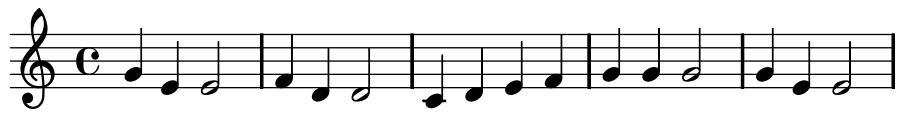
Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

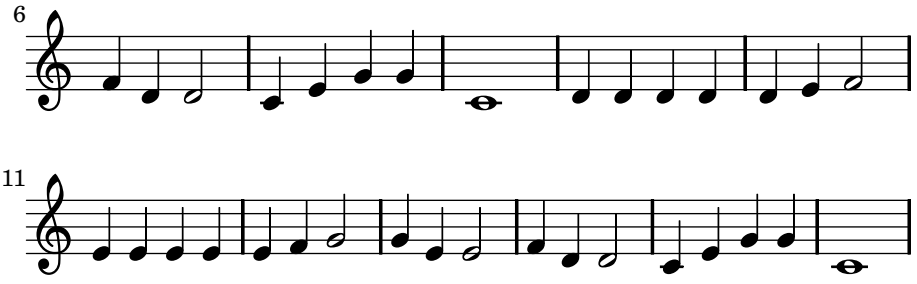
Référence des propriétés internes : Section "SpacingSpanner" dans *Référence des propriétés internes*.

4.5.3 Modification de l'espacement horizontal
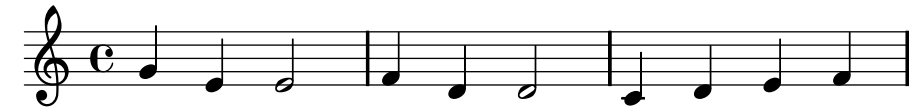
Vous pouvez influencer l'espacement horizontal à l'aide de la propriété `base-shortest-duration`. Comparons les deux partitions qui suivent, toutes deux montrant la même musique. La première partition applique les réglages par défaut, alors que la seconde bénéficie d'ajustement de la propriété `base-shortest-duration`. Au plus la valeur de `ly:make-moment` est grande, au plus la musique sera resserrée. En effet, `ly:make-moment` construit une durée : `1 4` est plus long que `1 16`.
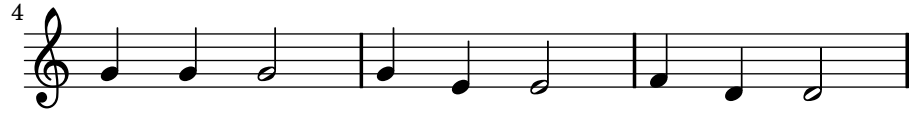
```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
}
```

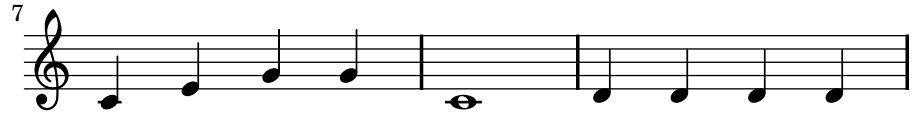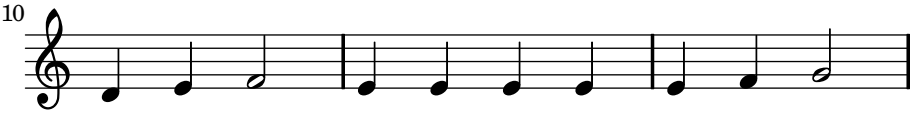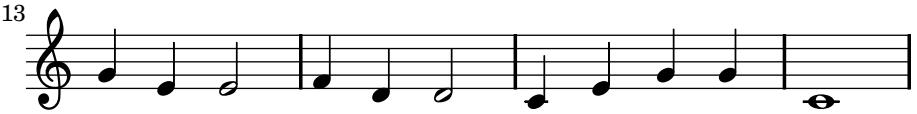





```
\score {
  \relative c'' {
    g4 e e2 | f4 d d2 | c4 d e f | g4 g g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
    d4 d d d | d4 e f2 | e4 e e e | e4 f g2 |
    g4 e e2 | f4 d d2 | c4 e g g | c,1 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner
        #'base-shortest-duration = #(ly:make-moment 1 16)
    }
  }
}
```







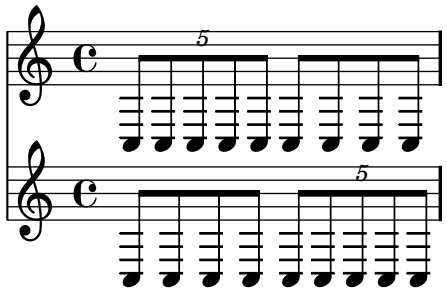




Morceaux choisis

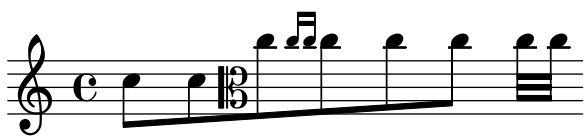
L'espacement au sein d'un nolet dépend par défaut d'un certain nombre de facteurs qui ne sont pas liés à la durée (altération, changement de clef, etc.). `Score.SpacingSpanner #'uniform-stretching` permet d'ignorer ces symboles et, par voie de conséquence, forcer l'espacement sur la simple durée. Notez bien que cette propriété s'appliquera à toute la partition, puisque mentionnée au sein d'un bloc `\layout`.

```
\score {
  <<
    \new Staff {
      \times 4/5 {
        c8 c8 c8 c8 c8
      }
      c8 c8 c8 c8
    }
    \new Staff {
      c8 c8 c8 c8
      \times 4/5 {
        c8 c8 c8 c8 c8
      }
    }
  >>
  \layout {
    \context {
      \Score
      \override SpacingSpanner #'uniform-stretching = ##t
    }
  }
}
```



L'activation du commutateur `strict-note-spacing` permet d'espacer les notes sans tenir compte des clefs, barres de mesure ou notes d'ornement qui pourraient apparaître :

```
\override Score.SpacingSpanner #'strict-note-spacing = ##t
\new Staff { c8[ c \clef alto c \grace { c16[ c] } c8 c c]  c32[ c32] }
```



Voir aussi

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.5.4 Longueur de ligne

Deux réglages de bases ont une influence considérable sur l'espacement : `line-width` et `indent`. Tous deux se placent dans le bloc `\layout`. Ils contrôleront la longueur des lignes et l'indentation de la première.

L'activation du commutateur `ragged-right` au sein du bloc `\layout` permet de terminer les systèmes naturellement plutôt que de les voir s'étirer sur toute la largeur de la page. Cette

option est particulièrement utile lorsque vous traitez de courts fragment, ou bien pour vérifier ce que donnerait l'espacement naturel. Bien qu'il soit désactivé par défaut, il sera activé si la partition ne comporte qu'un seul système.

Le fonctionnement de l'option `ragged-last` est en tout point identique à celui de `ragged-right`, à ceci près qu'il ne concerne que la dernière ligne de la partition. Il n'y a pas de restriction quant à cette ligne. Il en va de même que pour le formatage d'un paragraphe de texte, la dernière ligne s'arrête au dernier caractère.

```
\layout {
  indent = #0
  line-width = #150
  ragged-last = ##t
}
```

Voir aussi

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.5.5 Notation proportionnelle

LilyPond prend en charge la notation proportionnelle. Il s'agit dans ce cas de représenter la notation selon un espacement strictement relatif aux durées. Ce type d'espacement pourrait se comparer à l'utilisation de papier millimétré pour positionner les notes au fil de la portée. Certaines œuvres de la fin du XXe siècle et à l'aube du XXIe utilisent cette proportionnalité dans le but de clarifier des structures rythmiques complexes, d'aider au positionnement d'indications temporelles ou autres éléments graphiques directement dans la partition.
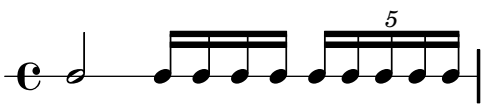
LilyPond met à votre disposition cinq réglages différents, qui peuvent s'utiliser conjointement ou individuellement, aux fins de mettre au point cette notation proportionnelle.

- `proportionalNotationDuration`
- `uniform-stretching`
- `strict-note-spacing`
- `\remove Separating_line_group_engraver`
- `\override PaperColumn #'used = ##t`

Nous allons examiner, dans les différents exemples qui suivent, les effets de ces réglages et comment ils interagissent.

Commençons par cette mesure toute simple qui utilise l'espacement classique et justifiée à gauche.

```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
}
```



Vous constatez que la blanche qui entame la mesure prend moins de la moitié de l'espace. De même, les doubles croches et le quintolet de doubles (donc des vingtièmes de ronde) qui terminent cette mesure n'en occupent pas la moitié de l'espace horizontal.

En matière de gravure traditionnelle, cet espacement correspond tout à fait à nos attentes, puisque nous pouvons rogner l'espace sur la blanche et ainsi en gagner en largeur sur toute la mesure qui fait une ronde.

Par contre, si nous avons besoin d'insérer une indication temporelle ou un autre graphisme en surplomb ou en dessous de notre partition, nous aurons besoin de la notation proportionnelle. Celle-ci s'active an définissant la propriété `proportionalNotationDuration`.

```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
 \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 20)
    }
  }
}
```



La blanche du début et les notes plus rapides de la deuxième moitié de la mesure occupent maintenant exactement le même espace horizontal. Nous pourrions donc y insérer, au-dessus ou au-dessous, une indication temporelle ou autre graphisme.

`proportionalNotationDuration` est une propriété attachée au contexte `Score`. Rappelez-vous que vous pouvez régler les propriétés d'un contexte à trois différents endroits de votre fichier : dans un bloc `\with`, dans un bloc `\context` ou au beau milieu de la musique à l'aide de la commande `\set`. Vous pouvez donc définir `proportionalNotationDuration` selon l'une de ces trois façons, à l'instar de n'importe quel définition de contexte.

La propriété `proportionalNotationDuration` prend en unique argument la durée de référence qui servira de base pour espacer toute la musique. La fonction scheme `make-moment` intégrée à LilyPond prend deux arguments : un numérateur et un dénominateur qui représentent une fraction de ronde. L'appel de `#(ly:make-moment 1 20)` produit donc une durée de référence égale à un vingtième de ronde. Vous pourriez tout aussi bien utiliser `#(ly:make-moment 1 16)`, `#(ly:make-moment 1 8)` ou `#(ly:make-moment 3 97)`.

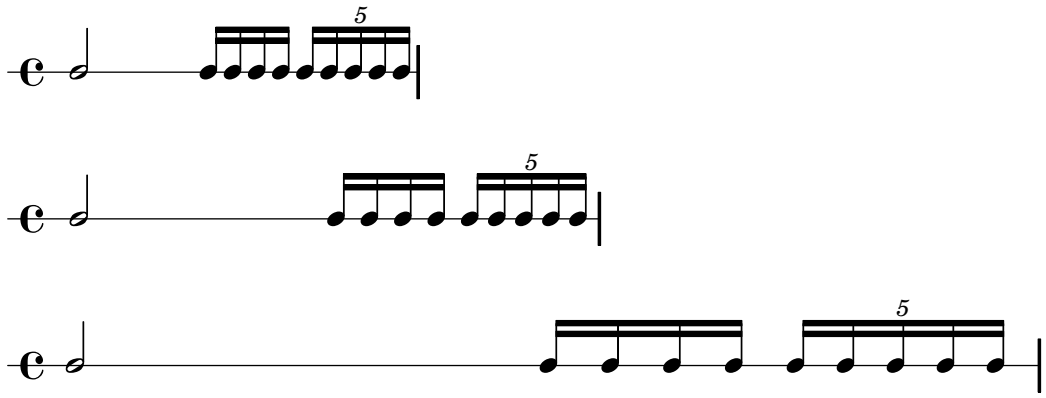
Se pose alors le problème de fournir la juste durée de référence à `proportionalNotationDuration`. Il faut en l'occurrence procéder par tâtonnement, en commençant par une valeur proche de la note la plus rapide (la durée la plus courte) du morceau. Au plus la durée de référence est petite, au plus la musique sera étalée ; à l'inverse, une durée de référence élevée produira une musique resserrée.

```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 8)
    }
  }
}

\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 16)
    }
  }
}

\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 32)
    }
  }
```

```
}
```

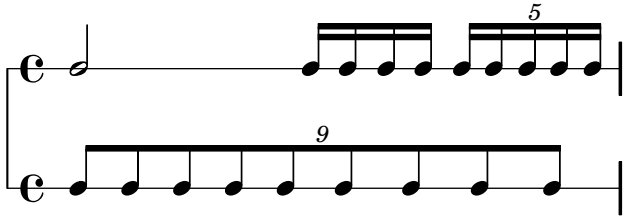


Vous ne manquerez pas de noter qu'une durée de référence trop grande, comme la croche pour la première ligne, a pour conséquence de resserrer la musique, ce qui peut aboutir à des chevauchements de têtes. Vous remarquez aussi que , par principe, la notation proportionnelle occupe beaucoup plus d'espace horizontal que l'espacement traditionnel. La notation proportionnelle met en évidence le rythme au détriment de l'espacement horizontal.

Examinons à présent le moyen d'optimiser l'espacement de nolets en tuilage.

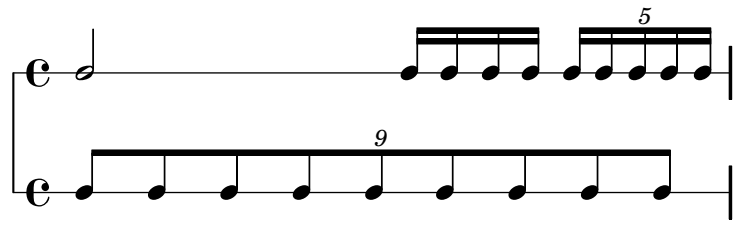
Reprenons notre exemple de départ, avec son espacement traditionnel, et ajoutons lui une portée incluant un autre type de nolet.

```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
    \new RhythmicStaff {
      \times 8/9 {
        c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
      }
    }
  >>
}
```



L'espacement est loin d'être idéal, pour la simple raison que l'espacement régulier des notes de la portée inférieure ne s'étire pas uniformément. Il est vrai que de telles constructions complexes en nolets sont assez rares en gravure traditionnelle, ce qui explique que les règles qu'elle applique peuvent amener à ce résultat. Le recours à `proportionalNotationDuration` permet d'arranger les choses.

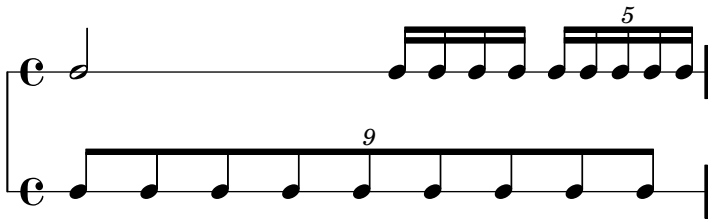
```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
    \new RhythmicStaff {
      \times 8/9 {
        c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
      }
    }
  >>
  \layout {
    \context {
      \Score
      proportionalNotationDuration = #(ly:make-moment 1 20)
    }
  }
}
```



Cependant, si l'on observe de près, il est évident que les notes de la deuxième moitié du ennaolet ont tendance à s'espacer légèrement plus que celles de la première moitié. Afin d'uniformiser cet étalement, nous allons activer le `uniform-stretching`, propriété attachée au `SpacingSpanner`.

```
\score {
  <<
    \new RhythmicStaff {
      c'2
      c'16 c'16 c'16 c'16
      \times 4/5 {
        c'16 c'16 c'16 c'16 c'16
      }
    }
    \new RhythmicStaff {
      \times 8/9 {
        c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8 c'8
      }
    }
  >>
  \layout {
    \context {
      \Score
```

```
      proportionalNotationDuration = #(ly:make-moment 1 20)
      \override SpacingSpanner #'uniform-stretching = ##t
    }
  }
}
```



L'espacement sur les deux portées est maintenant correct, les relations rythmiques sont clairement perceptibles, et nous pourrions y insérer une indication temporelle ou autre graphisme selon notre envie.

Notez bien que la prise en charge de la notation proportionnelle par LilyPond demande que, dans chaque partition, soit activée la propriété `uniform-stretching` du `SpacingSpanner`. Dans le cas contraire, utiliser `proportionalNotationDuration` aura pour conséquence, entre autres, un espacement erroné lorsque vous y aurez inséré des silences invisibles *skip*.

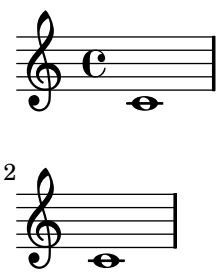
Le `SpacingSpanner` est en fait un objet graphique abstrait attaché au contexte `Score`. Tout comme pour la propriété `proportionalNotationDuration`, les réglages du `SpacingSpanner` peuvent se faire à trois différents endroits de votre fichier : dans un bloc `\with`, dans un bloc `\context` ou au beau milieu de la musique à l'aide de la commande `\set`.

Gardez bien à l'esprit qu'il n'y a qu'un seul `SpacingSpanner` par `Score`. Il s'ensuit que `uniform-stretching` est soit activé, soit désactivé, et dans tous les cas pour l'intégralité de la partition. Vous pourriez toutefois avoir besoin de modifier ce comportement en cours de partition, et recourir alors à l'instruction `\newSpacingSection` – pour de plus amples détails, voir la rubrique Section 4.5.2 [Changement d'espacement en cours de partition], page 449.

Intéressons-nous maintenant au `Separating_line_group_engraver`, qui est désactivé pour la plupart des partitions en notation proportionnelle. Voici ce qui apparaît dans une partition traditionnelle : il y a toujours un « espace préservé » juste avant la première note de chaque portée.

```
\paper {
  indent = #0
}

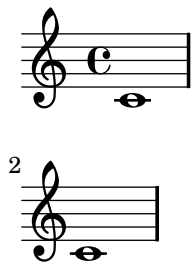
\new Staff {
  c'1
  \break
  c'1
}
```



Cet espace, géré par le `Separating_line_group_engraver`, est aussi présent lorsqu'intervient un changement de métrique, d'armure ou de clef. Désactiver le `Separating_line_group_engraver` revient à réduire cet espace à zéro.

```
\paper {
  indent = #0
}

\new Staff \with {
  \remove Separating_line_group_engraver
} {
  c'1
  \break
  c'1
}
```



Les éléments non musicaux tels que métrique, armure et clef, posent problème lorsqu'on travaille en notation proportionnelle. Bien qu'aucune notion de durée ne leur soit attachée, ces éléments « consomment » de l'espace. Différentes approches permettent cependant de gérer ce problème.
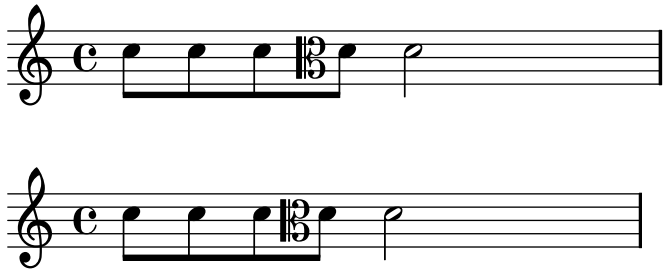
Éviter les problèmes d'espacement avec l'armure est chose aisée : il suffit qu'il n'y en ait pas ! C'est bien souvent le cas en musique contemporaine, où l'on trouve le plus d'ouvrages en notation proportionnelle. Il en va de même pour la métrique, et tout particulièrement lorsque la partition comporte un quadrillage temporel ou autres graphismes. L'absence de métrique reste cependant exceptionnelle et la plupart des partitions en notation proportionnelle laissent apparaître quelques métriques. Il est par contre pratiquement impossible de se passer de clef et d'altération.

L'une des options permettant de s'affranchir de l'espacement dû aux éléments non musicaux consiste en l'activation de la propriété `strict-note-spacing` attachée au `SpacingSpanner`. Observons les deux portées suivantes :

```
\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1 16)
  c''8
  c''8
  c''8
  \clef alto
  d'8
  d'2
}

\new Staff {
  \set Score.proportionalNotationDuration = #(ly:make-moment 1 16)
  \override Score.SpacingSpanner #'strict-note-spacing = ##t
  c''8
```

```
  c''8
  c''8
  \clef alto
  d'8
  d'2
}
```



Toutes deux affichent un espacement proportionnel. Cependant, la première ligne laisse apparaître un espacement plus lâche en raison de la présence d'un changement de clef. En ce qui concerne la deuxième ligne, l'espacement est strictement observé dès lors que la propriété `strict-note-spacing` a préalablement été activée. Comme vous pouvez le constater, l'activation de `strict-note-spacing` a pour conséquence que l'algorithme d'espacement ignore tout bonnement la largeur des métriques, armures, clefs et altérations.

En plus ceux que nous venons de voir, vous trouverez d'autres réglages en usage dans la notation proportionnelle comme, entre autres,

- `\override SpacingSpanner #'strict-grace-spacing = ##t`
- `tupletFullLength = ##t`
- `\override Beam #'breakable = ##t`
- `\override Glissando #'breakable = ##t`
- `\override TextSpanner #'breakable = ##t`
- `\remove Forbid_line_break_engraver (dans un contexte de voix`

Ces différents réglages permettent un espacement strict des notes d'ornement, d'étendre les indications de nolet afin d'indiquer de façon évidente leurs bornes et d'autoriser le tronçonnement des extenseurs à l'occasion d'un saut de ligne ou de page. Nous vous renvoyons aux différentes rubriques associées du manuel pour chacun de ces réglages.

Voir aussi

Manuel de notation : Section 4.5.2 [Changement d'espacement en cours de partition], page 449.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.6 Réduction du nombre de pages de la partition

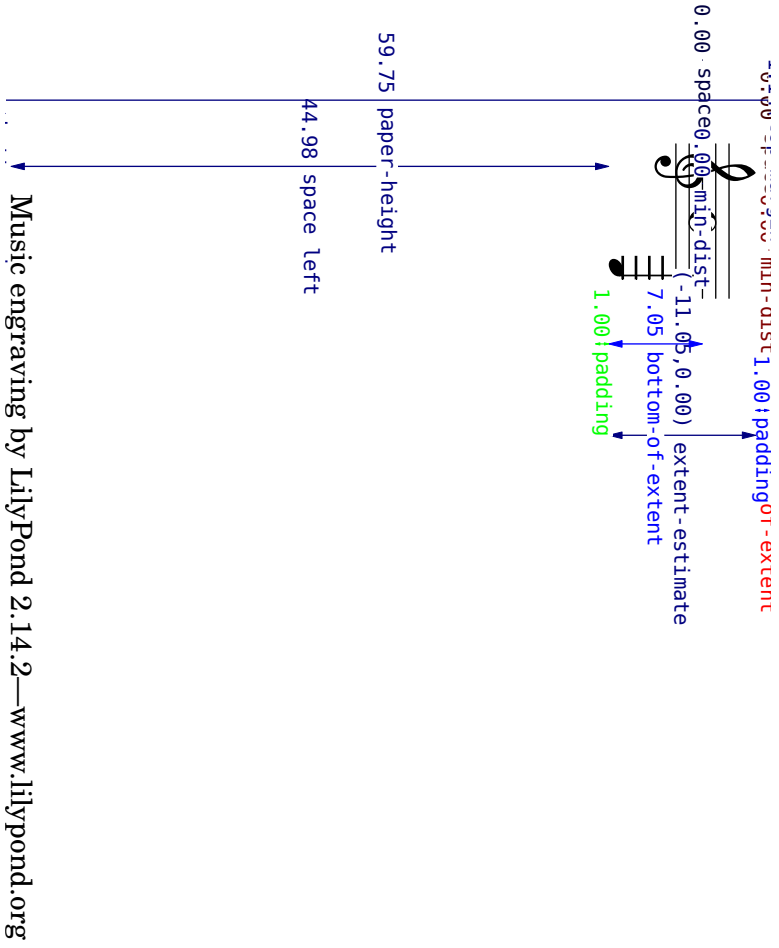
Vous pourriez un jour être confronté au problème suiven : l'une des pages de votre partition ne comporte que deux portées alors que – ce qui est d'autant plus frustrant – l'espace libre sur les autres pages aurait permis une distribution différente.

L'instruction `annotate-spacing` se révèle être un outil indispensable pour pour l'analyse des problèmes de mise en forme. Cette commande met en surimpression la valeur des différentes variables d'espacement et de mise en forme, comme nous allons le voir dans la rubrique Section 4.6.1 [Mise en évidence de l'espacement], page 460.

4.6.1 Mise en évidence de l'espacement

Le meilleur moyen d'appréhender les différentes variables de dimensionnement vertical sur lesquelles vous pouvez jouer au niveau de la mise en page, consiste à activer, au sein du bloc `\paper`, la fonction `annotate-spacing` :

```
#(set-default-paper-size "a6" 'landscape)
\book {
  \score { { c4 } }
  \paper { annotate-spacing = ##t }
}
```



Toutes les dimensions sont exprimées en espace de portée *(staff-space)* quelle que soit l'unité mentionnée dans les blocs `\paper` ou `\layout`. Dans cet exemple, la hauteur de la feuille (`paper-height`) est de 59,75 espaces de portée (`staff-spaces`) et la taille de portée (`staff-size`) de 20 points – sa valeur par défaut. Notez que :

1 point = (25,4/72,27) mm

1 staff-space = (`staff-size`)/4 pts

= (staff-size)/4 *
(25,4/72,27) mm

Dans le cas qui nous occupe, un `staff-space` égale environ 1,757 millimètres. Les 59,75 `staff-spaces` de `paper-height` correspondent donc à 105 millimètres, soit la hauteur d'une feuille au format A6 à l'italienne. Les paires `(a,b)` sont des intervalles, *a* en étant l'extrémité inférieure et *b* l'extrémité supérieure.

Voir aussi

Manuel de notation : Section 4.2.2 [Définition de la taille de portée], page 423.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

4.6.2 Modification de l'espacement

Les informations fournies par `annotate-spacing` en matière de dimensionnement vertical sont incomparables. Pour plus de détails sur les manières de modifier marges et autres variables connexes, consultez la rubrique Section 4.1 [Mise en forme de la page], page 412.

En dehors des marges, vous disposez de quelques moyens supplémentaires pour gagner de l'espace :

- Rapprocher les systèmes le plus possible les uns des autres, de telle sorte qu'il en tienne un maximum sur une même page, tout en les espaçant suffisamment pour éviter le blanc en bas de page.
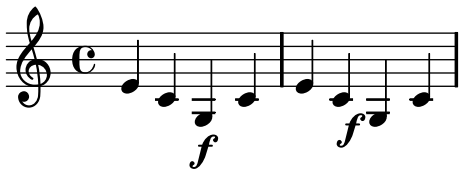
```
\paper {
  system-system-spacing = #'((basic-distance . 0.1) (padding . 0))
  ragged-last-bottom = ##f
  ragged-bottom = ##f
}
```

- Forcer le nombre de systèmes par page. Ceci peut se révéler judicieux à deux titres. D'une part, le fait de définir un nombre de systèmes – même s'il est égal à la valeur par défaut – peut aboutir à plus de systèmes par page dans la mesure où l'une des étapes d'estimation des hauteurs est tout simplement sautée. D'autre part, réduire le nombre de systèmes par page permet d'en disposer plus sur les suivantes. Par exemple, avec un nombre par défaut de 11 systèmes par pages, l'instruction suivante le force à 10.

```
\paper {
  system-count = #10
}
```
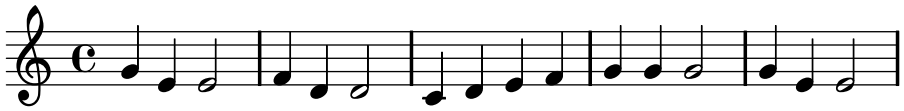
- Éviter ou réduire les objets qui augmentent la hauteur des systèmes. Par exemple, un crochet de reprise ou d'alternative consomme de l'espace. Le fait de les reporter sur plusieurs systèmes regroupés diminue d'autant l'espace disponible que si seul le premier ne comportait l'indication. Autre exemple, les indications de nuance qui se « détachent » d'un système peuvent être rapprochés de la portée :

```
e4 c g\f c
e4 c g-\tweak #'X-offset #-2.7 -\tweak #'Y-offset #2.5 \f c
```
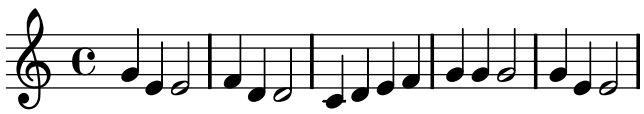


- Modifier l'espacement horizontal à l'aide du `SpacingSpanner`, comme indiqué à la rubrique Section 4.5.3 [Modification de l'espacement horizontal], page 449. Voici ce que donne l'espacement par défaut :

```
\score {
  \relative c'' {
    g4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
}
```



Par contre, le fait de modifier la valeur de la propriété `common-shortest-duration` en passant de `1/4` à `1/2` – bien que la noire soit la durée la plus courante, nous prenons une valeur plus longue – donnera un effet « resserré » à la musique :

```
\score {
  \relative c'' {
    g4 e e2 |
    f4 d d2 |
    c4 d e f |
    g4 g g2 |
    g4 e e2 |
  }
  \layout {
    \context {
      \Score
      \override SpacingSpanner
        #'common-shortest-duration = #(ly:make-moment 1 2)
    }
  }
}
```



La propriété `common-shortest-duration` ne peut être modifiée dynamiquement. Elle se place toujours dans un bloc `\context` et s'applique à l'intégralité de la partition.

Voir aussi

Manuel de notation : Section 4.1 [Mise en forme de la page], page 412, Section 4.5.3 [Modification de l'espacement horizontal], page 449.

Morceaux choisis : Section "Espacements" dans *Morceaux choisis*.

5 Modification des réglages prédéfinis

LilyPond est conçu pour engendrer, par défaut, des partitions de la plus haute qualité. Cependant, on peut parfois avoir à modifier cette mise en forme par défaut. Celle-ci est réglée par tout un ensemble de « leviers et manettes » plus connus sous le terme de « propriétés », dont ce chapitre ne cherche pas à faire l'inventaire exhaustif – vous en trouverez un Section "aperçu" dans *Manuel d'initiation* dans le manuel d'initiation. Le propos est plutôt ici de mettre en évidence les différents groupes auxquels s'apparentent ces contrôles, et d'expliquer comment trouver le bon levier pour obtenir tel ou tel effet en particulier.

Les moyens de contrôle des différents réglages sont décrits dans un document séparé, Section "Top" dans *Référence des propriétés internes*. Ce guide répertorie toutes les variables, fonctions et autres options que LilyPond met à votre disposition. Il est consultable en ligne, au format HTML, mais est également inclus dans la documentation fournie avec le logiciel.

En sous-main, LilyPond se sert du langage Scheme (un dérivé du LISP) comme infrastructure. Modifier les choix de mise en page revient à pénétrer dans les entrailles du programme, et de ce fait requiert l'emploi du Scheme. Les fragments de Scheme, dans un fichier '`.ly`', sont introduits par le caractère *hash*, (`#`, improprement surnommé « dièse »).[1]

Il est quatre domaines dans lesquels on peut modifier les réglages par défaut :

- La notation automatique, ce qui revient à modifier la manière dont les éléments de notation sont automatiquement créés – par exemple, les règles de ligature.
- L'apparence, qui se rapporte aux objets pris individuellement – ainsi de la direction des hampes, du placement des indications textuelles.
- Les contextes, qui recouvrent la manière dont les évènements musicaux sont représentés sous forme de notation – par exemple, le fait d'attribuer un chiffre de mesure distinct à chaque portée.
- La mise en page, autrement dit les marges, l'espacement, la taille du papier ; tous ces facteurs font l'objet des chapitres Chapitre 3 [Généralités en matière d'entrée et sortie], page 380 et Chapitre 4 [Gestion de l'espace], page 412.

5.1 Contextes d'interprétation

Cette section traite des contextes.

Voir aussi

Manuel d'initiation : Section "Contextes et graveurs" dans *Manuel d'initiation*.

Fichiers d'initialisation : '`ly/engraver-init.ly`', '`ly/performer-init.ly`'.

Morceaux choisis : Section "Contexts and engravers" dans *Morceaux choisis*.

Référence des propriétés internes : Section "Contexts" dans *Référence des propriétés internes*, Section "Engravers and Performers" dans *Référence des propriétés internes*.

5.1.1 Tout savoir sur les contextes

Score – le père de tous les contextes

Contextes de haut niveau – les systèmes

Contextes de niveau intermédiaire – les portées

[1] Le Section "Tutoriel Scheme" dans *Extension de LilyPond* fournit quelques notions de base pour saisir des nombres, des listes, des chaînes de caractères ou des symboles, en Scheme.

Contextes de bas niveau – les voix

5.1.2 Création d'un contexte

5.1.3 Conservation d'un contexte

5.1.4 Modification des greffons de contexte

Les contextes, tels que `Score` ou `Staff`, ne contiennent pas que des propriétés ; ils mettent également en œuvre certains sous-programmes (*plug-ins* pour employer le terme consacré) nommés « graveurs » (*engravers* pour reprendre le terme anglais). Ces sous-programmes sont chargés de créer les différents éléments de notation : on trouve ainsi dans le contexte `Voice` un graveur `Note_heads_engraver`, chargé des têtes de notes et, dans le contexte `Staff`, un graveur `Key_signature_engraver`, chargé de l'armure.

Vous trouverez une description exhaustive de chaque graveur dans Référence des propriétés internes ↦ Translation ↦ Engravers. Chaque contexte mentionné dans Référence des propriétés internes ↦ Translation ↦ Context. répertorie les graveurs mis en œuvre.
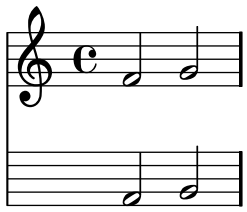
On peut faire, au moyen de ces graveurs, sa propre « cuisine », en modifiant les contextes à volonté.

Lorsque un contexte est créé, par la commande `\new` ou `\context`, on peut y adjoindre un bloc `\with` (en anglais 'avec'), dans lequel il est possible d'ajouter (commande `\consists`) ou d'enlever (commande `\remove`) des graveurs :

```
\new contexte \with {
  \consists ...
  \consists ...
  \remove ...
  \remove ...
  etc.
}
{
  ..musique..
}
```

Ici les points de suspension ... devront être remplacés par les noms des graveurs désirés. Dans l'exemple suivant, on enlève du contexte `Staff`, la métrique (graveur `Time_signature_engraver`) et la clé (graveur `Clef_engraver`).

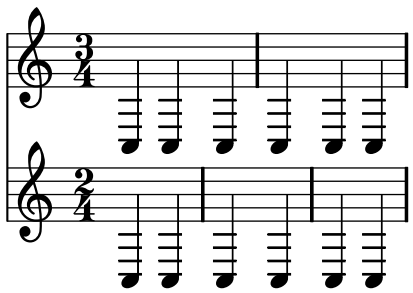
```
<<
  \new Staff {
    f2 g
  }
  \new Staff \with {
     \remove "Time_signature_engraver"
     \remove "Clef_engraver"
  } {
    f2 g2
  }
>>
```



La clé et le chiffre de mesure ont disparu de la deuxième portée. C'est une méthode quelque peu radicale, puisqu'elle affectera toute la portée jusqu'à la fin de la partition. L'espacement s'en trouve également affecté, ce qui peut être ou non l'effet recherché. Une méthode plus sophistiquée aurait été de rendre ces objets transparents (voir Section "Visibilité et couleur des objets" dans *Manuel d'initiation*).

Dans l'exemple suivant, voici une mise en pratique plus utile. En temps normal, les barres de mesure et la métrique sont synchronisées verticalement dans toute la partition. Les graveurs qui en sont responsables se nomment `Timing_translator` et `Default_bar_line_engraver`. En les enlevant du contexte `Score` pour les attribuer au contexte `Staff`, chaque portée peut désormais avoir sa propre métrique.

```
\score {
  <<
    \new Staff \with {
      \consists "Timing_translator"
      \consists "Default_bar_line_engraver"
    } {
        \time 3/4
        c4 c c c c c
    }
  \new Staff \with {
    \consists "Timing_translator"
    \consists "Default_bar_line_engraver"
  } {
      \time 2/4
      c4 c c c c c
  }
>>
\layout {
  \context {
    \Score
    \remove "Timing_translator"
    \remove "Default_bar_line_engraver"
    }
  }
}
```



En règle générale, l'ordre dans lequel les graveurs sont mentionnés importe peu. Il se peut toutefois qu'un graveur écrive une propriété qui sera interprétée par un autre, ou qu'un graveur

crée un objet graphique qui sera traité par un autre ; l'ordre d'apparition de ces graveurs prendra lors tout son importance.

Pour information, les ordonnancements suivants sont importants : le `Bar_engraver` devrait toujours être le premier ; le `New_fingering_engraver` doit toujours précéder le `Script_column_engraver`. Nous vous conseillons, pour les autres, de vérifier les évenntuelles dépendances.

5.1.5 Modification des réglages par défaut d'un contexte

Les réglages montrés dans les sous-sections Section 5.3.2 [La commande de fixation (set)], page 472, Section 5.1.4 [Modification des greffons de contexte], page 464 peuvent également être saisis indépendamment de la musique dans le bloc `\layout`, au moyen de la commande `\context` :

```
\layout {
  ...
  \context {
    \Staff

    \set fontSize = #-2
    \override Stem #'thickness = #4.0
    \remove "Time_signature_engraver"
  }
}
```

Le raccourci `\Staff` invoque les définitions inhérentes au contexte `Staff`, de façon à ce qu'elles puissent être modifiées.

Les lignes suivantes affecteront toutes les portées (tous les contextes `Staff`) dans la partition.

```
\set fontSize = #-2
\override Stem #'thickness = #4.0
\remove "Time_signature_engraver"
```

Les autres contextes peuvent être modifiés de même manière.

La commande `\set`, dans le bloc `\layout`, est facultative ; aussi les lignes suivantes produiront-elles le même effet.

```
\context {
  ...
  fontSize = #-2
}
```

Problèmes connus et avertissements

Il est impossible de stocker des modifications de contexte dans un identificateur.

Le raccourci `\Staff \RemoveEmptyStaves` détruit tous les réglages du contexte `\Staff`. Pour modifier des propriétés de portées gouvernées par `\Staff \RemoveEmptyStaves`, il faut le faire impérativement **après** avoir invoqué `\Staff \RemoveEmptyStaves` :

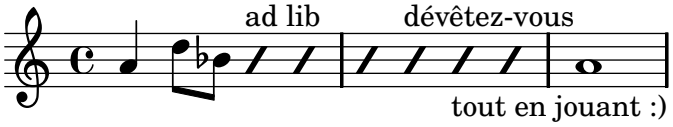
```
\layout {
  \context {
    \Staff \RemoveEmptyStaves

    \override Stem #'thickness = #4.0
  }
}
```

5.1.6 Définition de nouveaux contextes

Les contextes tels que `Staff` ou `Voice` sont faits de briques de constructions empilées. En combinant divers graveurs, il est possible de créer de nouveaux types de contextes.

Dans l'exemple suivant, on construit, de zéro, un nouveau contexte très semblable à `Voice`, mais qui n'imprime que des têtes de notes en forme de barres obliques au centre de la portée. Un tel contexte, par exemple, peut servir à indiquer un passage improvisé dans un morceau de jazz.



On a rassemblé les réglages dans un bloc `\context`, lui-même dans le bloc `\layout` :

```
\layout {
  \context {
    ...
  }
}
```

En lieu et place des points (. . .), voici les éléments à saisir :

- Tout d'abord, il convient de donner un nom `\name` à notre nouveau contexte :

  ```
  \name ImproVoice
  ```

- Comme il est très semblable à `Voice`, nous souhaitons que toutes les commandes associées au `Voice` déjà existant, restent valables. D'où nécessité de la commande `\alias`, qui va l'associer au contexte `Voice` :

  ```
  \alias Voice
  ```

- Ce contexte doit pouvoir imprimer des notes, et des indications textuelles ; on ajoute donc les graveurs appropriés.

  ```
  \consists Note_heads_engraver
  \consists Text_engraver
  ```

- Cependant, on veut que les notes s'affichent toutes au centre de la portée :

  ```
  \consists Pitch_squash_engraver
  squashedPosition = #0
  ```

  Le graveur Section "Pitch_squash_engraver" dans *Référence des propriétés internes* intercepte les notes créées par Section "Note_heads_engraver" dans *Référence des propriétés internes*, et les « écrase » pour qu'elles aient toutes la même position verticale, définie par `squashedPosition` : ici il s'agit de la valeur `0`, c'est-à-dire la ligne du milieu.

- On veut que les notes aient la forme d'une barre oblique, sans aucune hampe :

  ```
  \override NoteHead #'style = #'slash
  \override Stem #'transparent = ##t
  ```

- Afin que tous ces graveurs puissent travailler de concert, on leur adjoint un sous-programme spécial, introduit par la commande `\type` : il s'agit de `Engraver_group`,

  ```
  \type "Engraver_group"
  ```

Récapitulons – on se retrouve avec le bloc suivant :

```
\context {
  \name ImproVoice
  \type "Engraver_group"
  \consists "Note_heads_engraver"
```

```
  \consists "Text_engraver"
  \consists Pitch_squash_engraver
  squashedPosition = #0
  \override NoteHead #'style = #'slash
  \override Stem #'transparent = ##t
  \alias Voice
}
```

Ce n'est pas tout. En effet, on veut intégrer le nouveau contexte `ImproVoice` dans la hiérarchie des contextes. Tout comme le contexte `Voice`, sa place est au sein du contexte `Staff`. Nous allons donc modifier le contexte `Staff`, au moyen de la commande `\accepts` :

```
\context {
  \Staff
  \accepts ImproVoice
}
```

Le contraire de `\accepts` est `\denies`, qui est parfois utile lorsque l'on recycle des définitions de contextes déjà existantes.

Enfin, tout cela doit prendre place dans le bloc `\layout`, comme suit :
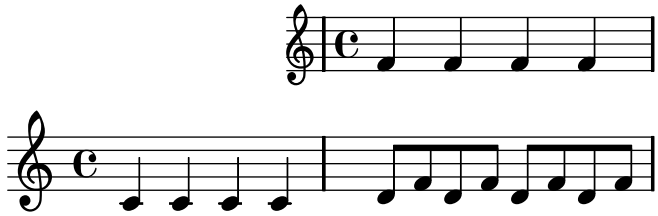
```
\layout {
  \context {
    \name ImproVoice
    ...
  }
  \context {
    \Staff
    \accepts "ImproVoice"
  }
}
```

On peut alors saisir la musique, comme dans l'exemple plus haut :

```
\relative c'' {
  a4 d8 bes8
  \new ImproVoice {
    c4^"ad lib" c
    c4 c^"dévêtez-vous"
    c c_"tout en jouant :)"
  }
  a1
}
```

5.1.7 Alignement des contextes

Il est possible d'aligner verticalement chaque nouveau contexte, en-dessous ou au-dessus, par exemple dans le cas de musique vocale (Section "Ensemble vocal" dans *Manuel d'initiation*) ou d'*ossias*.
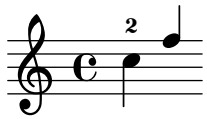


5.2 En quoi consiste la référence des propriétés internes

5.2.1 Navigation dans les références du programme

Comment, par exemple, déplacer le doigté dans le fragment suivant ?

```
c-2
\stemUp
f
```



Sur la page de la documentation relative aux doigtés, c'est-à-dire [Doigtés], page 183, se trouve l'indication suivante :

Voir aussi

Référence des propriétés internes : Section "Fingering" dans *Référence des propriétés internes*.

Ladite référence est disponible au format HTML, ce qui rend la navigation bien plus aisée. Il est possible soit de la lire en ligne, soit de la télécharger dans ce format. La démarche présentée ici sera plus difficile à comprendre dans un document au format PDF.

Suivons le lien Section "Fingering" dans *Référence des propriétés internes*. En haut de la nouvelle page, on peut lire

Fingering objects are created by: Section "Fingering_engraver" dans *Référence des propriétés internes* and Section "New_fingering_engraver" dans *Référence des propriétés internes*.

En d'autres termes, *Les indications de doigtés* (`Fingering` en anglais) *sont créées par les graveurs Section "Fingering_engraver" dans Référence des propriétés internes et Section "New_fingering_engraver" dans Référence des propriétés internes.*

En suivant derechef les liens propres à la référence du programme, on suit en fait le cheminement qui aboutit à la création de la partition :

- Section "Fingering" dans *Référence des propriétés internes*: Section "Fingering" dans *Référence des propriétés internes* objects are created by: Section "Fingering_engraver" dans *Référence des propriétés internes*
- Section "Fingering_engraver" dans *Référence des propriétés internes*: Music types accepted: Section "fingering-event" dans *Référence des propriétés internes*
- Section "fingering-event" dans *Référence des propriétés internes*: Music event type `fingering-event` is in Music expressions named Section "FingeringEvent" dans *Référence des propriétés internes*

Ce cheminement se produit, bien sûr, en sens inverse : nous sommes ici partis du résultat, et avons abouti aux évènements (en anglais *Events*) engendrés par le fichier d'entrée. L'inverse est également possible : on peut partir d'un évènement, et suivre le cheminement de LilyPond qui aboutit à la création d'un ou plusieurs objets graphiques.

La référence des propriétés internes peut également se parcourir comme un document normal. On y trouve des chapitres tels que `Music definitions` Section "Translation" dans *Référence des propriétés internes*, ou encore Section "Backend" dans *Référence des propriétés internes*. Chaque chapitre recense toutes les définitions employées, et les propriétés sujettes à ajustements.

5.2.2 Interfaces de rendu

Tous les éléments de notation sont considérés comme des objets graphiques (en anglais *Graphical Object*, d'où le diminutif *Grob*). Chaque objet est doté d'un certain nombre de propriétés (l'épaisseur du trait, l'orientation, etc.), et lié à d'autres objets. Le fonctionnement de ces objets est décrit en détail dans Section "grob-interface" dans *Référence des propriétés internes*.

Prenons l'exemple des doigtés (en anglais *Fingering*). La page `Fingering` de la Référence des propriétés internes établit une liste de définitions propres à ce type d'objet :

`padding` (dimension, in staff space):

`0.5`

Ce qui signifie que les doigtés doivent être maintenus à une certaine distance (*padding*) des notes : 0,5 unités *staff-space* (espace de portée).

Chaque objet peut avoir plusieurs attributs, en tant qu'élément typographique ou musical. Ainsi, un doigté (objet *Fingering*) possède les attributs suivants :

- Sa taille ne dépend pas de l'espacement horizontal, contrairement aux liaisons ou ligatures.
- C'est du texte – un texte vraiment court, certes.
- Ce texte est imprimé au moyen d'une fonte, contrairement aux liaisons ou ligatures.
- Sur l'axe horizontal, le centre de ce symbole doit être aligné avec le centre de la note.
- Sur l'axe vertical, le symbole doit être proche de la note et de la portée.
- Sur l'axe vertical encore, il doit également s'ordonner avec les éventuels autres symboles, ponctuations, ou éléments textuels.

Faire appliquer ces différents attributs est le rôle des *interfaces*, que l'on trouve en bas de la page Section "Fingering" dans *Référence des propriétés internes*.

This object supports the following interfaces: Section "item-interface" dans *Référence des propriétés internes*, Section "self-alignment-interface" dans *Référence des propriétés internes*, Section "side-position-interface" dans *Référence des propriétés internes*, Section "text-interface" dans *Référence des propriétés internes*, Section "text-script-interface" dans *Référence des propriétés internes*, Section "font-interface" dans *Référence des propriétés internes*, Section "finger-interface" dans *Référence des propriétés internes*, and Section "grob-interface" dans *Référence des propriétés internes*.

En français,

Cet objet admet les interfaces suivantes :

Suit la liste des interfaces en question, présentées comme autant de liens, qui conduisent sur les pages dédiées à chacune d'entre elles. Chaque interface est dotée d'un certain nombre de propriétés, dont certaines peuvent être modifiées, et d'autres non (les *Internal properties*, ou propriétés internes).

Pour aller encore plus loin, plutôt que de simplement parler de l'objet `Fingering`, ce qui ne nous avance pas à grand chose, on peut aller explorer son âme même, dans les fichiers source de LilyPond (voir Section "Autres sources de documentation" dans *Manuel d'initiation*), en l'occurence le fichier '`scm/define-grobs.scm`' :

```
(Fingering
  . ((padding . 0.5)
     (avoid-slur . around)
     (slur-padding . 0.2)
     (staff-padding . 0.5)
     (self-alignment-X . 0)
     (self-alignment-Y . 0)
     (script-priority . 100)
```
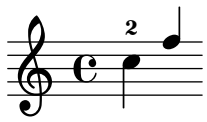
```
(stencil . ,ly:text-interface::print)
(direction . ,ly:script-interface::calc-direction)
(font-encoding . fetaText)
(font-size . -5)  ; don't overlap when next to heads.
(meta . ((class . Item)
(interfaces . (finger-interface
               font-interface
               text-script-interface
               text-interface
               side-position-interface
               self-alignment-interface
               item-interface))))))
```

. . .où l'on découvre que l'objet `Fingering` n'est rien de plus qu'un amas de variables et de réglages. La page de la Référence du programme est en fait directement engendrée par cette définition.

5.2.3 Détermination de la propriété d'un objet graphique (grob)

Nous voulions changer la position du chiffre **2** dans le fragment suivant :

```
c-2
\stemUp
f
```



Dans la mesure où le **2** est placé, verticalement, à proximité de la note qui lui correspond, nous allons devoir trouver l'interface en charge de ce placement, qui se trouve être `side-position-interface`. Sur la page de cette interface, on peut lire :

`side-position-interface`

Position a victim object (this one) next to other objects (the support). The property `direction` signifies where to put the victim object relative to the support (left or right, up or down?)

Ce qui signifie

`side-position-interface`

Placer l'objet affecté à proximité d'autres objets. La propriété `direction` indique où placer l'objet (à droite ou à gauche, en haut ou en bas).

En-dessous de cette description se trouve décrite la variable `padding` :

`padding` (dimension, in staff space)
Add this much extra space between objects that are next to each other.

Ce qui signifie
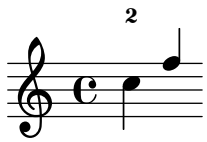
Ajouter tel espace supplémentaire entre des objets proches les uns des autres.

En augmentant la valeur de `padding`, on peut donc éloigner le doigté de la note. La commande suivante insère trois unités d'espace vide entre la note et le doigté :

```
\once \override Voice.Fingering #'padding = #3
```

En ajoutant cette commande avant la création du doigté (de l'objet `Fingering`), donc avant `c2`, on obtient le résultat suivant :

```
\once \override Voice.Fingering #'padding = #3
c-2
\stemUp
f
```



Dans le cas présent, le réglage intervient dans le contexte `Voice`, ce qui pouvait également se déduire de la Référence du programme, où la page du graveur Section “Fingering_engraver” dans *Référence des propriétés internes* indique :

Fingering_engraver is part of contexts: . . . Section “Voice” dans *Référence des propriétés internes*

Ce qui signifie

Le graveur Fingering_engraver fait partie des contextes : . . . Section “Voice” dans *Référence des propriétés internes*

5.2.4 Conventions de nommage

5.3 Modification de propriétés

5.3.1 Vue d’ensemble de la modification des propriétés

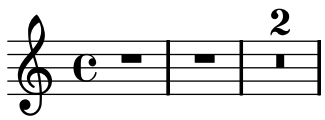
5.3.2 La commande `\set`

Chaque contexte peut avoir plusieurs *propriétés*, c’est-à-dire des variables qu’il inclut. Ces dernières peuvent être modifiées « à la volée », c’est-à-dire pendant que la compilation s’accomplit. C’est là le rôle de la commande `\set`.

```
\set contexte.propriété = #valeur
```

Ainsi :

```
R1*2
\set Score.skipBars = ##t
R1*2
```
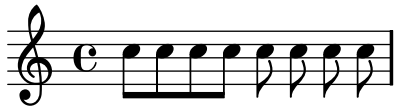


Cette commande permet de condenser les mesures vides de notes, en des silences multi-mesures. Il s’agit d’un objet Scheme, auquel on attribue la valeur booléenne « vrai », c’est-à-dire la lettre `#t` pour « True » en anglais.

Ce changement étant appliqué « à la volée », il n’affecte que le second groupe de notes.
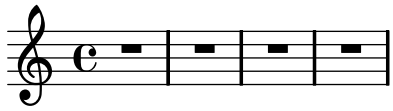
Si l’argument *contexte* n’est pas spécifié, alors la propriété cherchera à s’appliquer dans le contexte le plus restreint où elle est employée : le plus souvent `ChordNames`, `Voice`, ou `Lyrics`. Dans l’exemple suivant,

```
c8 c c c
\set autoBeaming = ##f
c8 c c c
```



aucun argument *contexte* n'a été donné à la commande `\set`. De ce fait, les ligatures automatiques sont désactivées dans le contexte actuel, c'est-à-dire Section "Voice" dans *Référence des propriétés internes*. Notez que le contexte le plus restreint n'est pas toujours le bon, et peut ne pas contenir la propriété qui vous intéresse : ainsi, la propriété `skipBars`, évoquée plus haut, ne relève pas du contexte `Voice`, et le code suivant ne fonctionnera pas.

```
R1*2
\set skipBars = ##t
R1*2
```



Les contextes s'organisent de façon hiérarchique : aussi, lorsqu'un contexte de niveau supérieur est spécifié (par exemple `Staff`), la propriété sera modifée dans tous les contextes inférieurs (tous les contextes `Voice`, par exemple) qu'il contient.

La commande `\unset` permet d'annuler la définition d'une propriété :

`\unset` *`contexte.propriété`*

si et seulement si cette propriété à été définie dans ce *contexte* précis ; ainsi,

```
\set Staff.autoBeaming = ##f
```

même s'il s'applique à tous les contextes `Voice` contenus dans le contexte `Staff`, ne peut être annulé au niveau `Voice`. Le code suivant sera sans effet.

```
\unset Voice.autoBeaming
```

En d'autres termes, la commande `\unset` doit impérativement être accompagnée du même contexte que la commande `\set` d'origine. Pour annuler l'effet, dans notre exemple, de `Staff.autoBeaming = ##f`, il faut donc entrer :
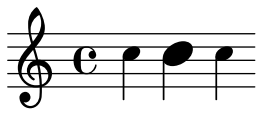
```
\unset Staff.autoBeaming
```

Si l'on se trouve dans le contexte le plus restreint, il n'est pas obligatoire, là encore, de spécifier le *contexte*. Ainsi, les deux lignes suivantes sont équivalentes.

```
\set Voice.autoBeaming = ##t
\set autoBeaming = ##t
```

Pour modifier une propriété de façon à ce qu'elle ne s'applique qu'une seule fois, il convient d'employer la commande `\once` :

```
c4
\once \set fontSize = #4.7
c4
c4
```



Ici le changement de taille est annulé aussitôt après la note concernée.

La référence du programme contient une description exhaustive de toutes les propriétés contexte par contexte : voir Translation ↦ Tunable context properties.

5.3.3 La commande `\override`

La commande `\override` permet de modifier la mise en page en détail. Examinons son utilisation concrètementet dans les détails. La syntaxe de cette commande ressemble généralement à :

`\override` *`contexte.objet #'propriété = #valeur`*
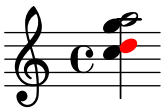
La propriété *propriété* de l'objet *objet*, appartenant au contexte *contexte*, se voit ainsi attribuer la valeur *valeur*.

5.3.4 La commande `\tweak`

Dans certains cas, on peut passer par un raccourci pour arranger les objets graphiques. Lorsqu'un objet est directement engendré par un élément distinct du fichier source, on peut utiliser la commande `\tweak`.

Dans l'accord suivant, les notes sont modifiées une par une :

```
<
  c
  \tweak #'color #red d
  g
  \tweak #'duration-log #1  a
>4-\tweak #'padding #10 -.
```



Comme on peut le voir, les propriétés sont ici modifiées directement en même temps que les objets sont saisis. Il n'est plus besoin de spécifier ni le nom de l'objet (*grob*), ni le contexte dans lequel cela doit s'appliquer. Ce procédé ne marche que pour des objets directement liés aux évènements (Section "Event" dans *Référence des propriétés internes*) du fichier source. Par exemple :

- Les têtes de notes au sein d'un accord, qui sont directement engendrées par les hauteurs indiquées
- Les signes d'articulation, engendrés par les indications de ponctuation.

En revanche, les hampes ou les altérations sont engendrées par les têtes de notes, et non par des évènements dans le fichier source. De même pour les clés, qui ne sont pas directement engendrées par le fichier source, mais plutôt par le changement d'une propriété interne.

En fait, très peu d'objets passent *directement* du code source à la partition. Une note toute simple, par exemple `c4`, fait l'objet d'un traitement et n'est donc pas directement rendue ; c'est pourquoi le code suivant ne sera d'aucun effet :

```
\tweak #'color #red c4
```

Voir pour plus de détails Section "Affichage d'expressions musicales" dans *Extension de LilyPond*.

5.3.5 `\set` ou `\override`

Si les propriétés peuvent être modifiées de deux façons, par les commandes `\set` et `\override`, c'est qu'il y a deux types de propriétés.

Les contextes peuvent avoir des propriétés, dont les noms commencent par une minuscule puis comprennent une ou plusieurs majuscules (de style `totoTutu`). Elles ont surtout trait à la notation des éléments musicaux : par exemple, `localKeySignature` permet de choisir s'il faut ou non imprimer une altération, ou `measurePosition` permet de choisir quand il faut imprimer une barre de mesure. Ces propriétés de contextes sont appelées à changer au long de l'interprétation de la partition : `measurePosition` en est un exemple évident. Ces propriétés doivent être modifiées avec la commande `\set`.

Il existe un type particulier de propriété : les descriptions d'éléments. Ces propriétés, dont les noms commencent par une majuscule, puis comprennent une ou plusieurs majuscules (de style `TotoTata`), contiennent les réglages « par défaut » pour les objets graphiques. Ces réglages sont sous forme de liste Scheme ; on peut les consulter dans le fichier '`scm/define-grobs.scm`'.

Les descriptions d'éléments doivent être modifiées avec la commande `\override`.

`\override` est en fait un raccourci :

`\override` *`contexte.objet`* `#'`*`propriété`* `= #`*`valeur`*

est plus ou moins l'équivalent de

`\set` *`contexte.objet`* `#'`*`propriété`* `= #(cons (cons '`*`propriété valeur`*`) <valeur précédente de co`*
`texte``)`

La valeur de `context` (la liste Scheme, ou *alist*) sert à initialiser les propriétés des objets un par un. Les objets eux-même ont leurs propriétés, dont les noms, dans la tradition du langage Scheme, comprennent un trait d'union (`toto-titi`). Ces propriétés internes changent constamment au cours de la mise en page : en fait, la gravure d'une page n'est autre que le calcul de toutes ces propriétés, au moyen de fonctions de rappel.

La propriété `fontSize` est une exception : c'est un raccourci, qui équivaudrait à saisir `\override ... #'font-size` pour tous les objets textuels. Dans la mesure où il s'agit d'une manipulation très courante, une propriété spéciale a été créée. Elle doit être modifiée avec la commande `\set`.

5.3.6 Modification de listes associatives

5.4 Propriétés et contextes utiles

5.4.1 Modes de saisie

5.4.2 Direction et positionnement

5.4.3 Ordre des contextes de mise en forme

5.4.4 Distances et unités de mesure

5.4.5 Propriétés des symboles de la portée

5.4.6 Extenseurs et prolongateurs

Utilisation de `spanner-interface`

Utilisation de `line-spanner-interface`

5.4.7 Visibilité des objets

Suppression des stencils

Transparence des objets

Blanchiment des objets

Utilisation de break-visibility

Considérations spécifiques

5.4.8 Styles de ligne

5.4.9 Rotation des objets

Rotation des objets de mise en forme

Rotation des étiquettes

5.5 Retouches avancées

5.5.1 Alignement des objets

Détermination directe de `X-offset` et `Y-offset`

Utilisation de `side-position-interface`

Utilisation de `self-alignment-interface`

Utilisation des procédures `centered-on-parent`

Utilisation de `break-aligned-interface`

5.5.2 Regroupement vertical d'objets graphiques

Les objets `VerticalAlignment` et `VerticalAxisGroup` travaillent de concert. Comme leurs noms anglais l'indiquent, `VerticalAxisGroup` regroupe différents objets tels que les portées (`Staff`), les paroles (`Lyrics`) et ainsi de suite ; puis `VerticalAlignment` synchronise verticalement ces différents groupes. En général, il n'y a qu'un seul `VerticalAlignment` pour l'ensemble de la partition, mais chaque contexte `Staff`, `Lyrics`, etc. possède son propre `VerticalAxisGroup`.

5.5.3 Modification des stencils

5.5.4 Modification de l'allure des éléments

Modification des liaisons

5.6 Utilisation de fonctions musicales

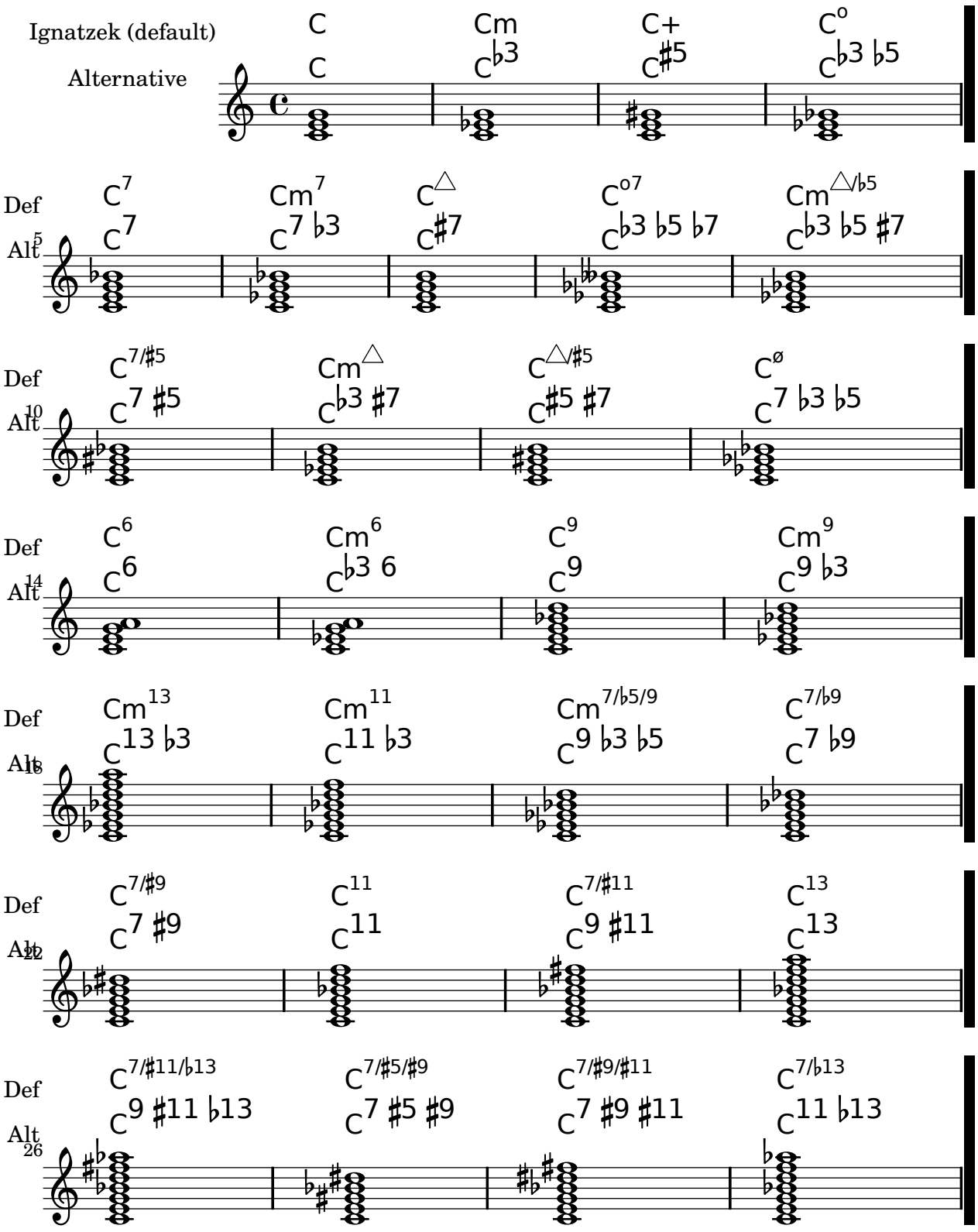
5.6.1 Syntaxe d'une fonction de substitution

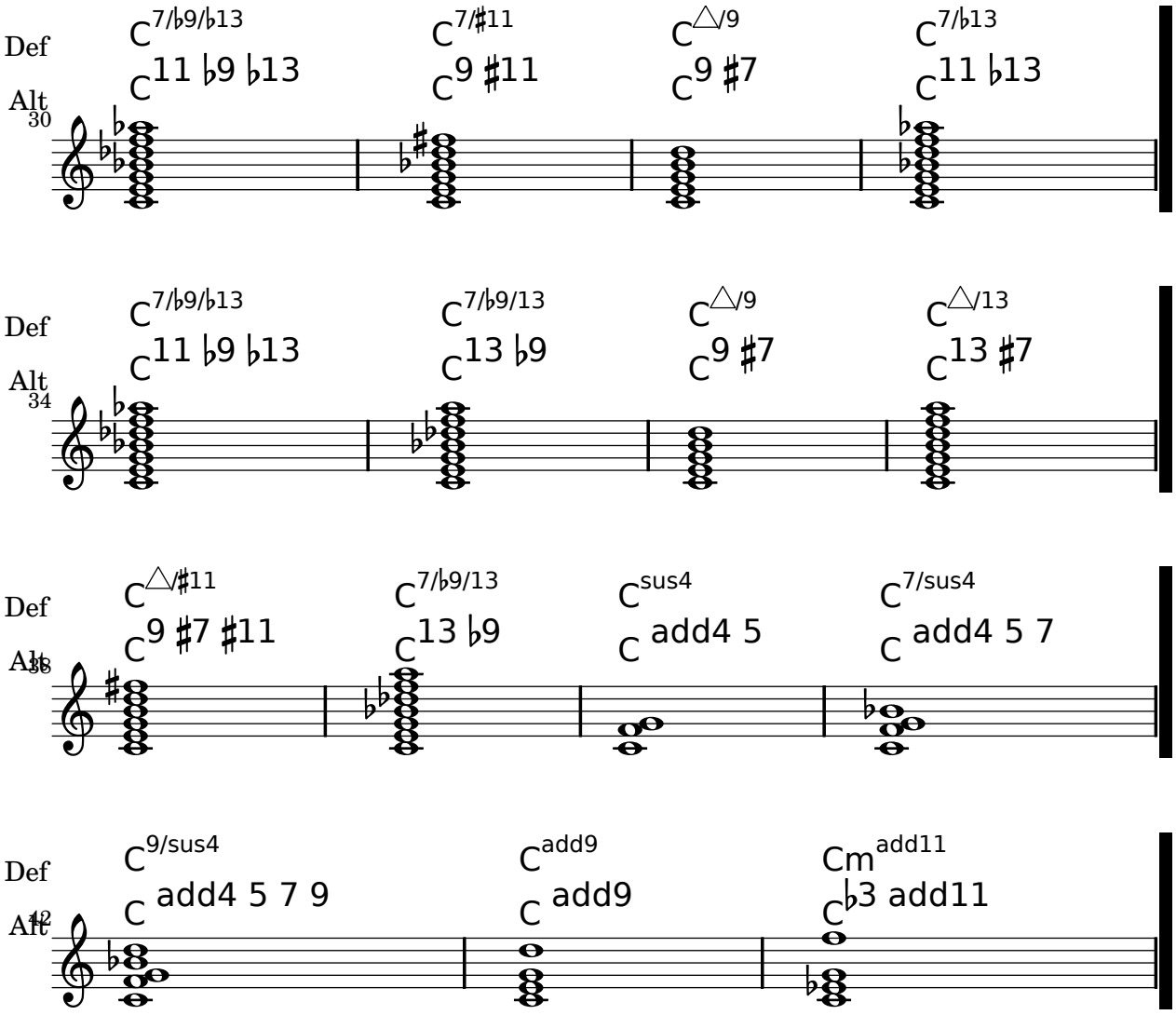
5.6.2 Exemples de fonction de substitution

Annexe A Tables du manuel de notation

A.1 Table des noms d'accord

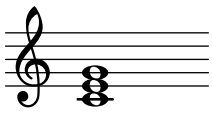
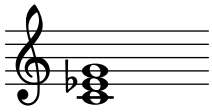
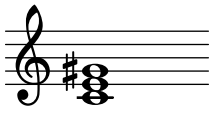
La liste suivante répertorie les noms des accords selon les deux types les plus répandus, ainsi que ce qu'ils représentent.
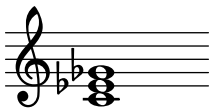
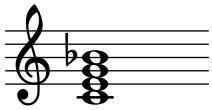
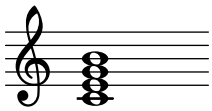
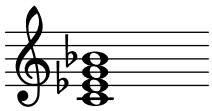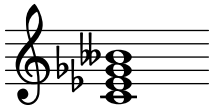
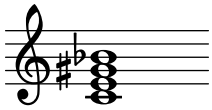
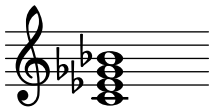
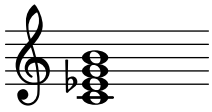
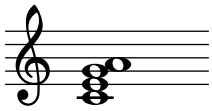
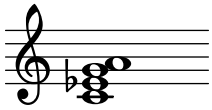


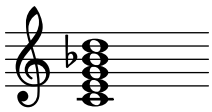
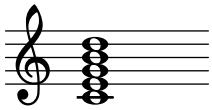
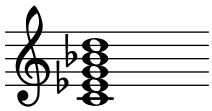
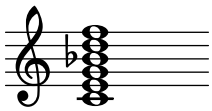
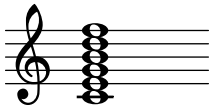
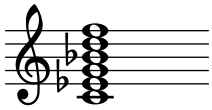
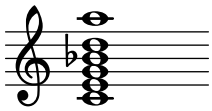
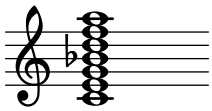
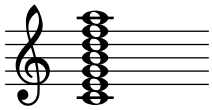
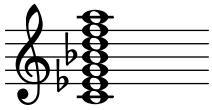


A.2 Modificateurs d'accord usuels

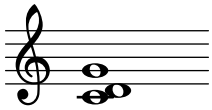
Le tableau suivant indique les différents modificateurs utilisables en mode accords (`\chordmode`) qui permettent d'obtenir les structures habituelles d'un certain nombre d'accords.

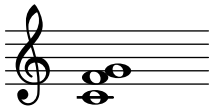
Type d'accord	**Intervalles**	**Modificateur(s)**	**Exemple**
Majeur	Tierce mejeure et quinte juste	`5` ou rien	
Mineur	Tierce minaure et quinte juste	`m` ou `m5`	
Augmenté	Tierce majeure et quinte augmentée	`aug`	

Diminué	Tierce mineure et quinte diminuée	`dim`	
Septième de dominante	Triton majeur et septième mineure	`7`	
Septième majeure	Triton majeur et septième majeure	`maj7` or `maj`	
Septième mineure	Triton mineur et septième mineure	`m7`	
Septième diminuée	Triton diminué et septième diminuée	`dim7`	
Septième augmentée	Triton augmenté et septième mineure	`aug7`	
Septième semi-diminuée	Triton diminué et septième mineure	`m7.5-`	
Accord mineur avec septième majeure	Triton mineur et septième majeure	`maj7.3-`	
Sixte majeure	Triton majeur et sixte	`6`	
Sixte mineure	Triton mineur et sixte	`m6`	

Neuvième de dominante	Septième de dominante et neuvième majeure	`9`	
Neuvième majeure	Septième majeure et neuvième majeure	`maj9`	
Neuvième mineure	Septième mineure et neuvième majeure	`m9`	
Onzième de dominante	Neuvième de dominante et onzième juste	`11`	
Onzième majeure	Neuvième majeure et onzième juste	`maj11`	
Onzième mineure	Neuvième mineure et onzième juste	`m11`	
Trézième de dominante	Neuvième de dominante et trézième majeure	`13`	
Trézième de dominante	Onzième de dominante et trézième majeure	`13.11`	
Trézième majeure	Onzième majeure et trézième majeure	`maj13.11`	
Trézième mineure	Onzièmem mineure et trézième majeure	`m13.11`	

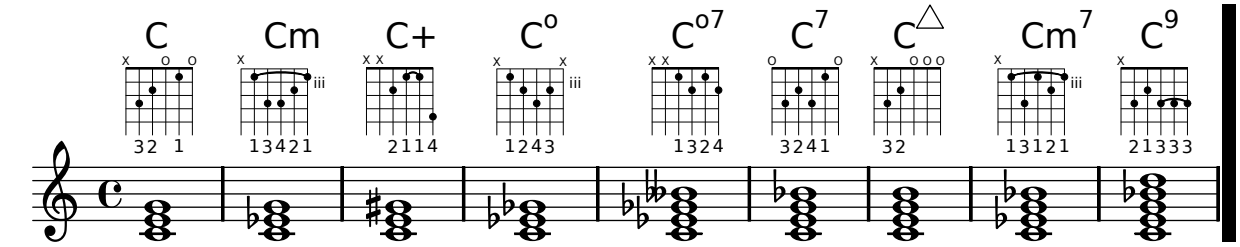
Seconde suspendue	Seconde majeure et quinte juste	`sus2`
Quarte suspendue	Quarte juste et quqinte juste	`sus4`

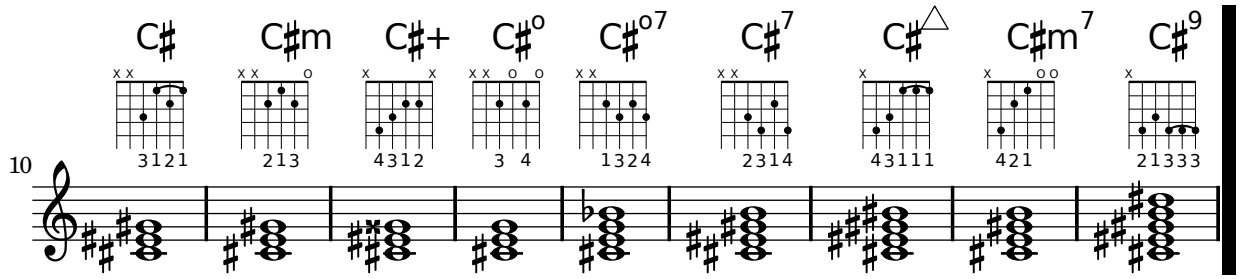


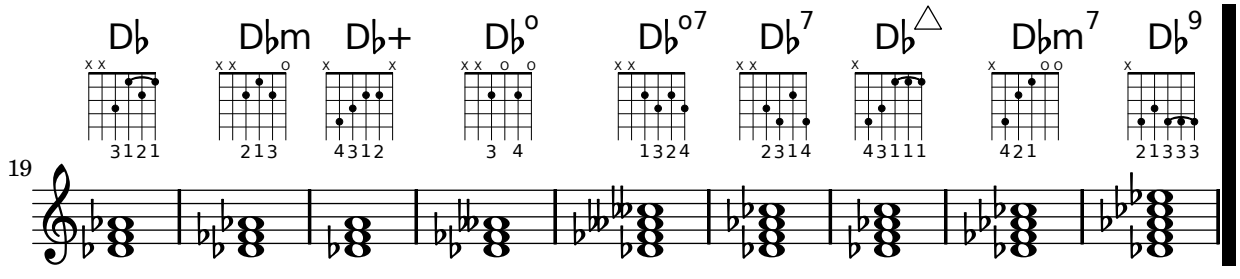


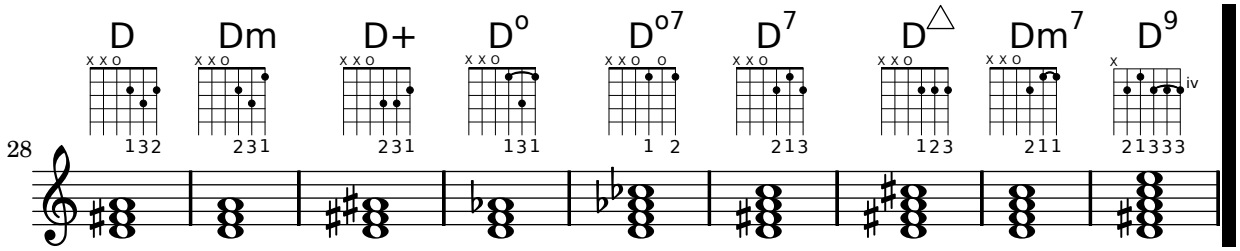
A.3 Diagrammes d'accord prédéfinis

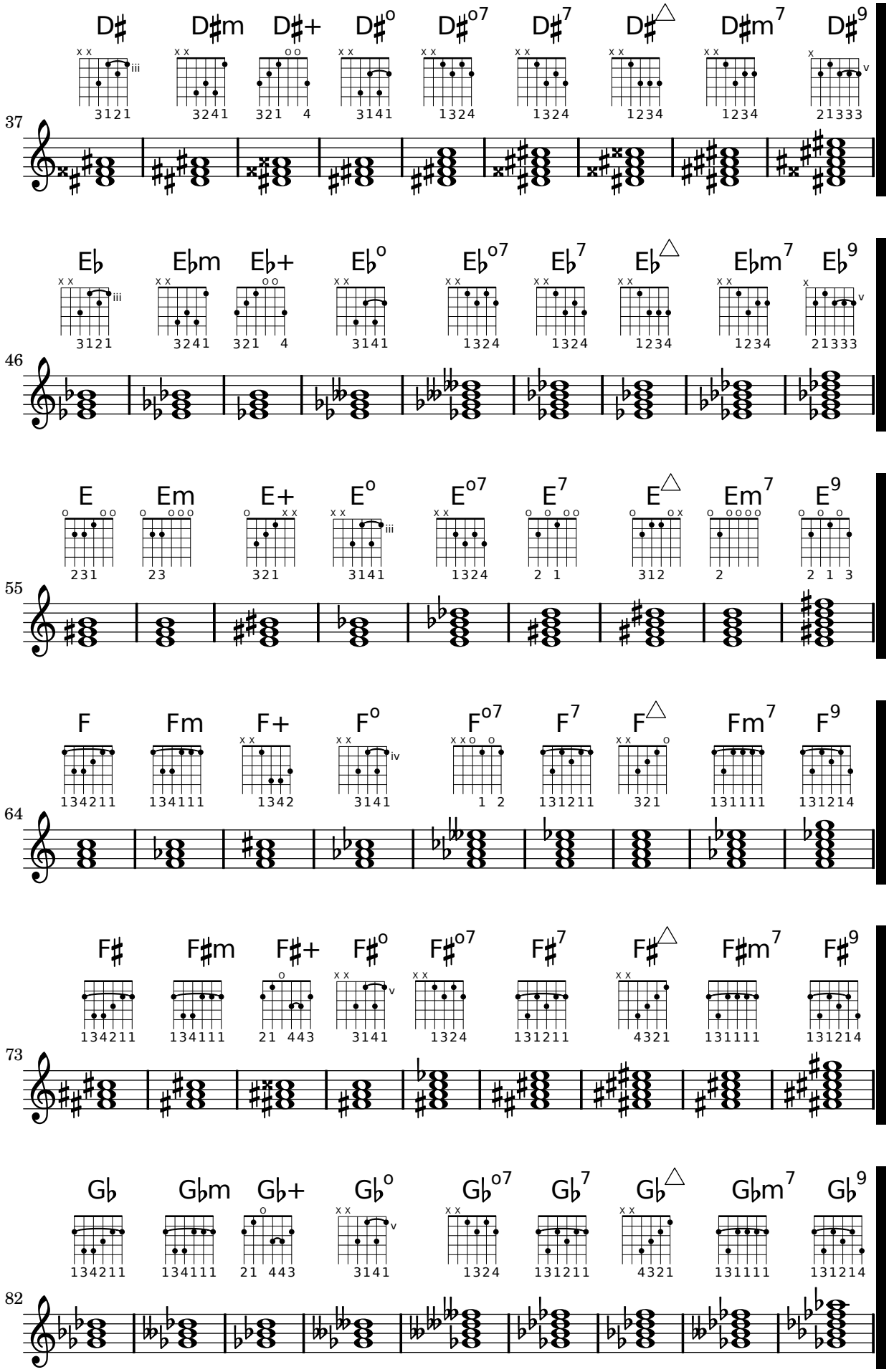
La liste suivante répertorie les tablatures prédéfinies pour la guitare.

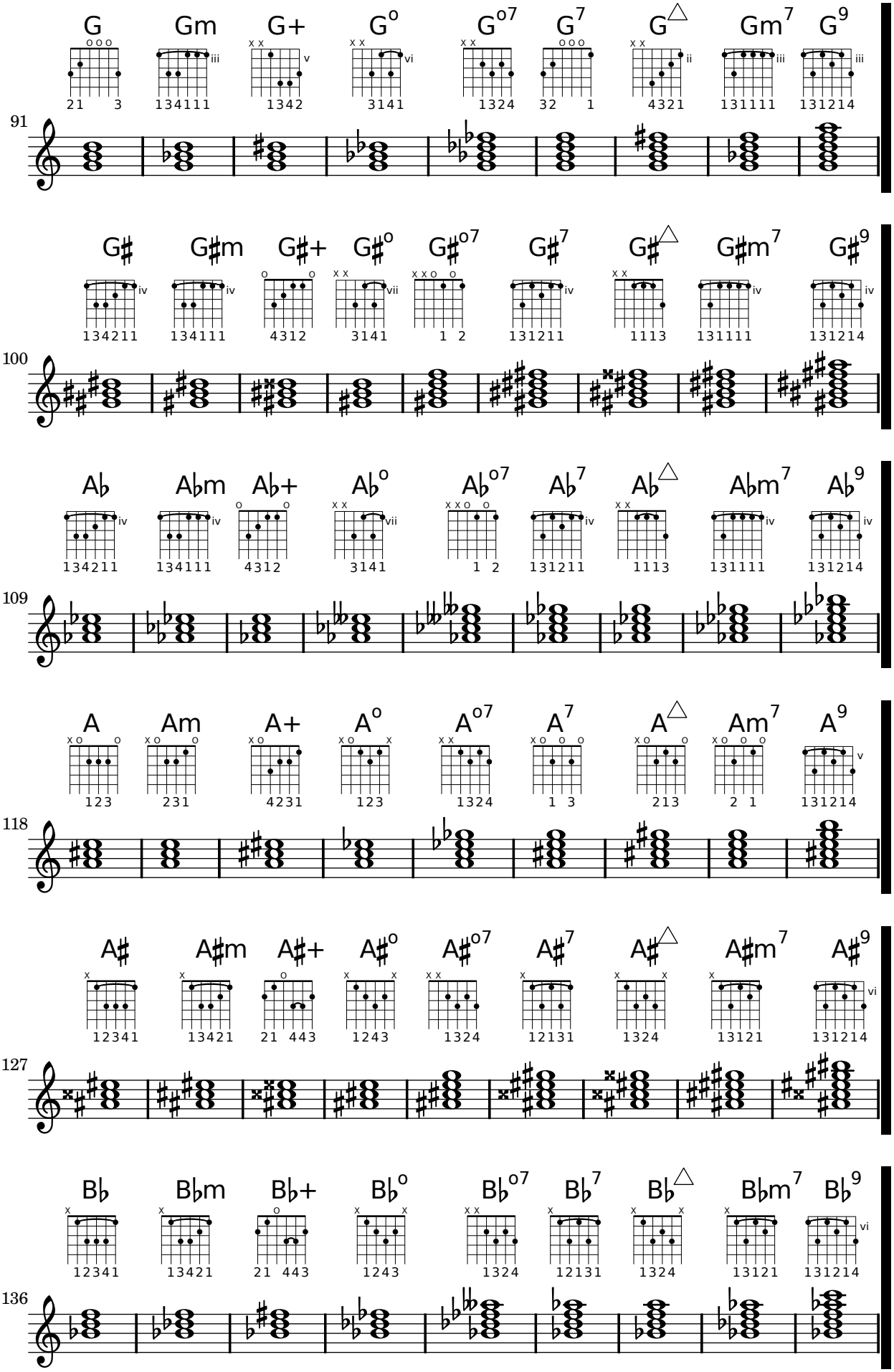


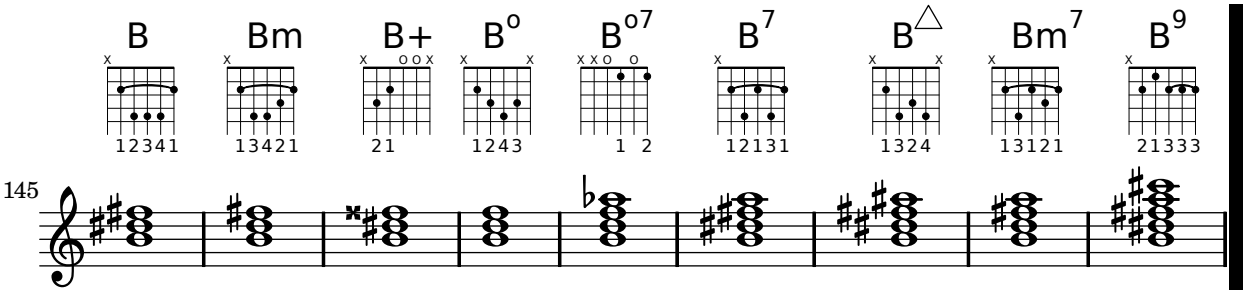




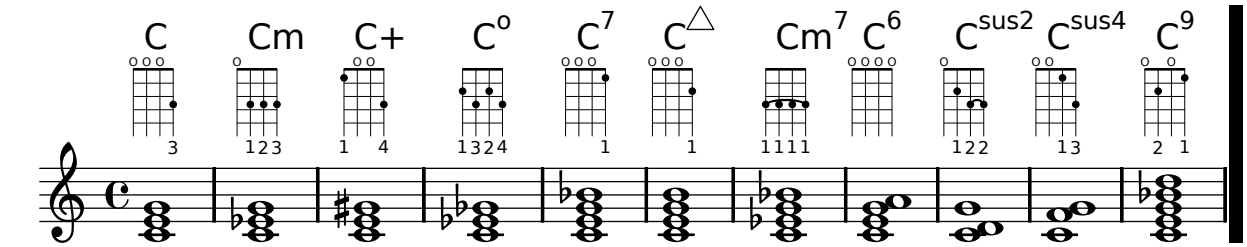


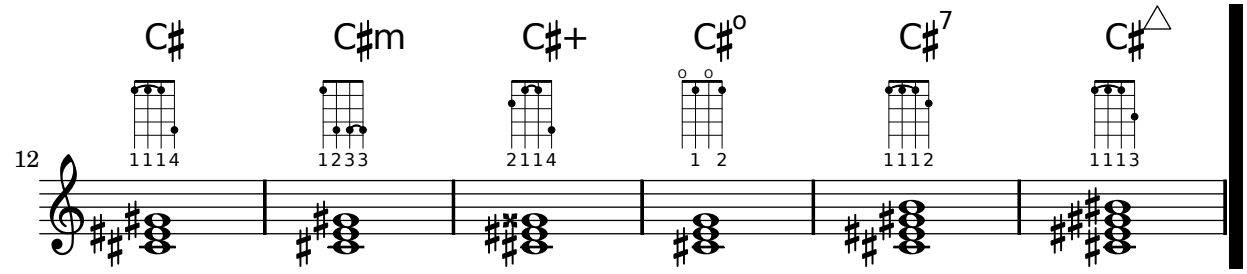

D♯ D♯m D♯+ D♯o D♯o7 D♯7 D♯△ D♯m7 D♯9
3121 3241 321 4 3141 1324 1324 1234 1234 21333
37
E♭ E♭m E♭+ E♭o E♭o7 E♭7 E♭△ E♭m7 E♭9
3121 3241 321 4 3141 1324 1324 1234 1234 21333
46
E Em E+ Eo Eo7 E7 E△ Em7 E9
231 23 321 3141 1324 2 1 312 2 2 1 3
55
F Fm F+ Fo Fo7 F7 F△ Fm7 F9
134211 134111 1342 3141 1 2 131211 321 131111 131214
64
F♯ F♯m F♯+ F♯o F♯o7 F♯7 F♯△ F♯m7 F♯9
134211 134111 21 443 3141 1324 131211 4321 131111 131214
73
G♭ G♭m G♭+ G♭o G♭o7 G♭7 G♭△ G♭m7 G♭9
134211 134111 21 443 3141 1324 131211 4321 131111 131214
82

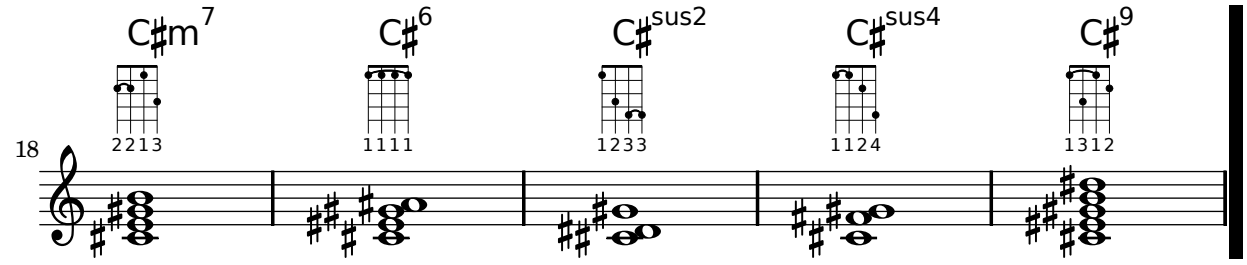

G Gm G+ G° G°7 G7 G△ Gm7 G9
21 3 134111 1342 3141 1324 32 1 4321 131111 131214
91
G♯ G♯m G♯+ G♯° G♯°7 G♯7 G♯△ G♯m7 G♯9
134211 134111 4312 3141 1 2 131211 1113 131111 131214
100
A♭ A♭m A♭+ A♭° A♭°7 A♭7 A♭△ A♭m7 A♭9
134211 134111 4312 3141 1 2 131211 1113 131111 131214
109
A Am A+ A° A°7 A7 A△ Am7 A9
123 231 4231 123 1324 1 3 213 2 1 131214
118
A♯ A♯m A♯+ A♯° A♯°7 A♯7 A♯△ A♯m7 A♯9
12341 13421 21 443 1243 1324 12131 1324 13121 131214
127
B♭ B♭m B♭+ B♭° B♭°7 B♭7 B♭△ B♭m7 B♭9
12341 13421 21 443 1243 1324 12131 1324 13121 131214
136

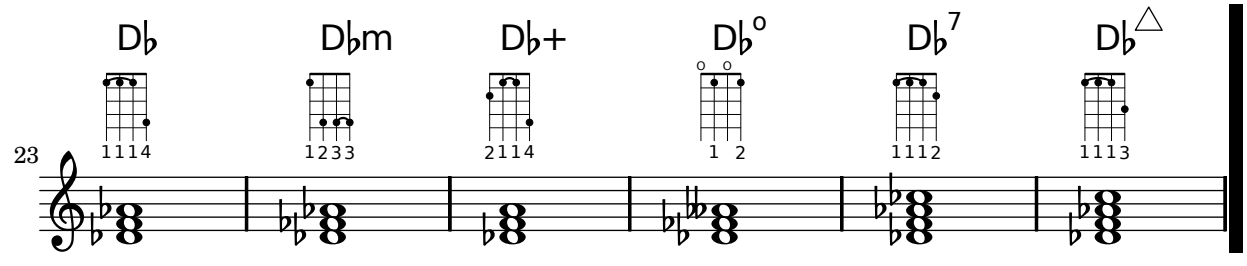


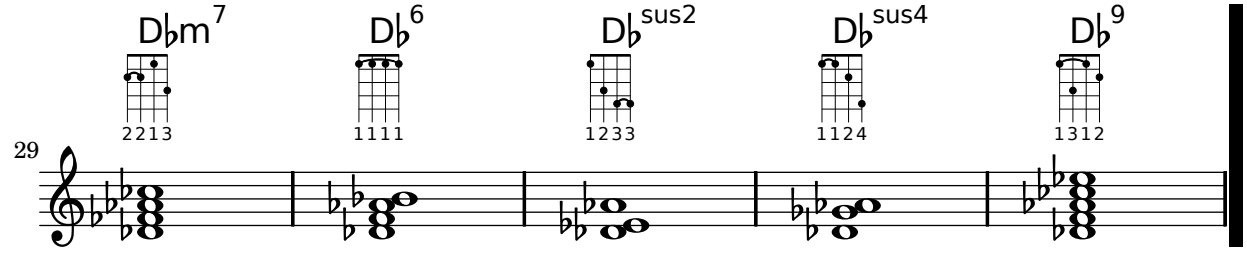
La liste suivante répertorie les tablatures prédéfinies pour le ukulele.
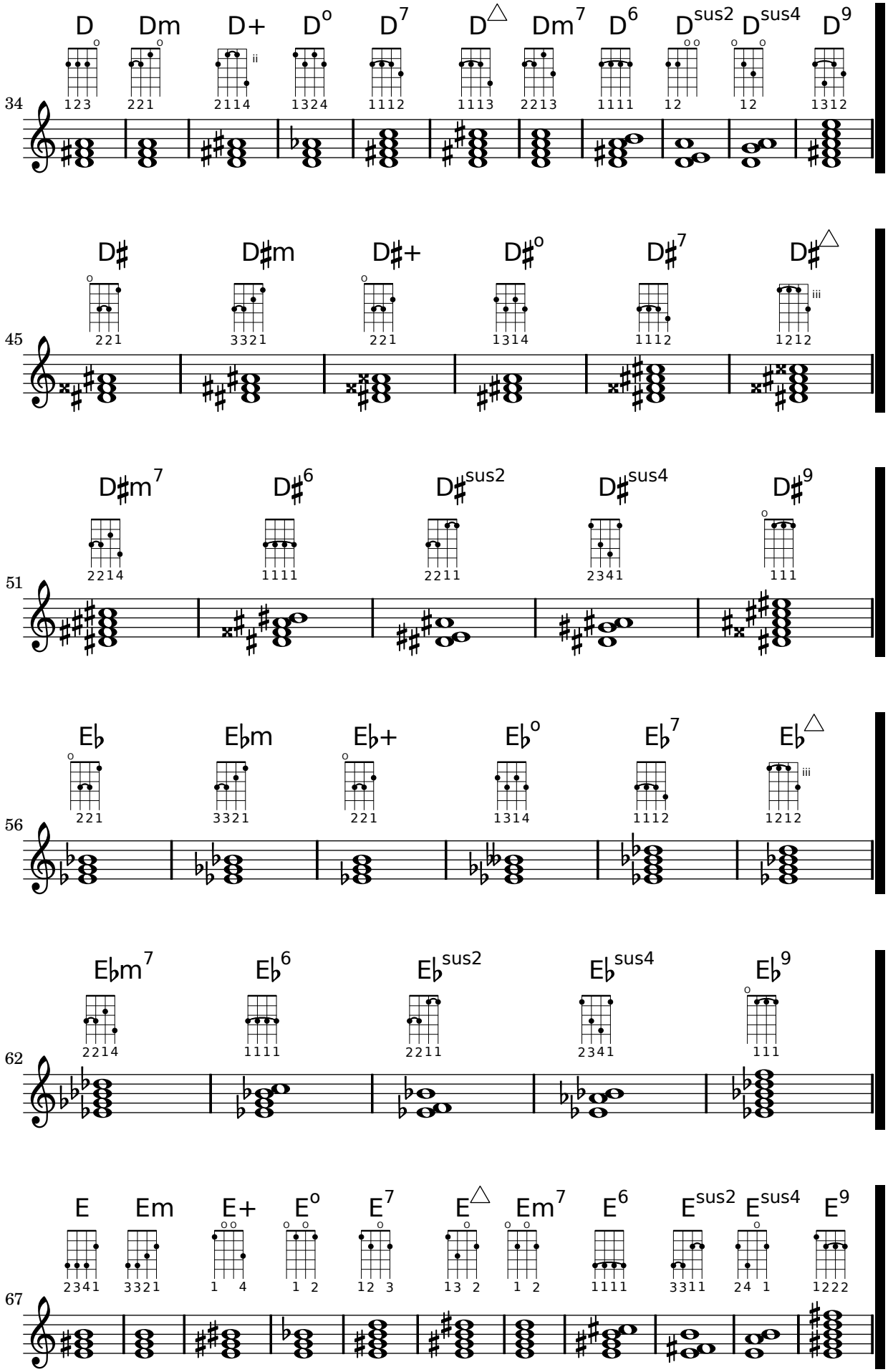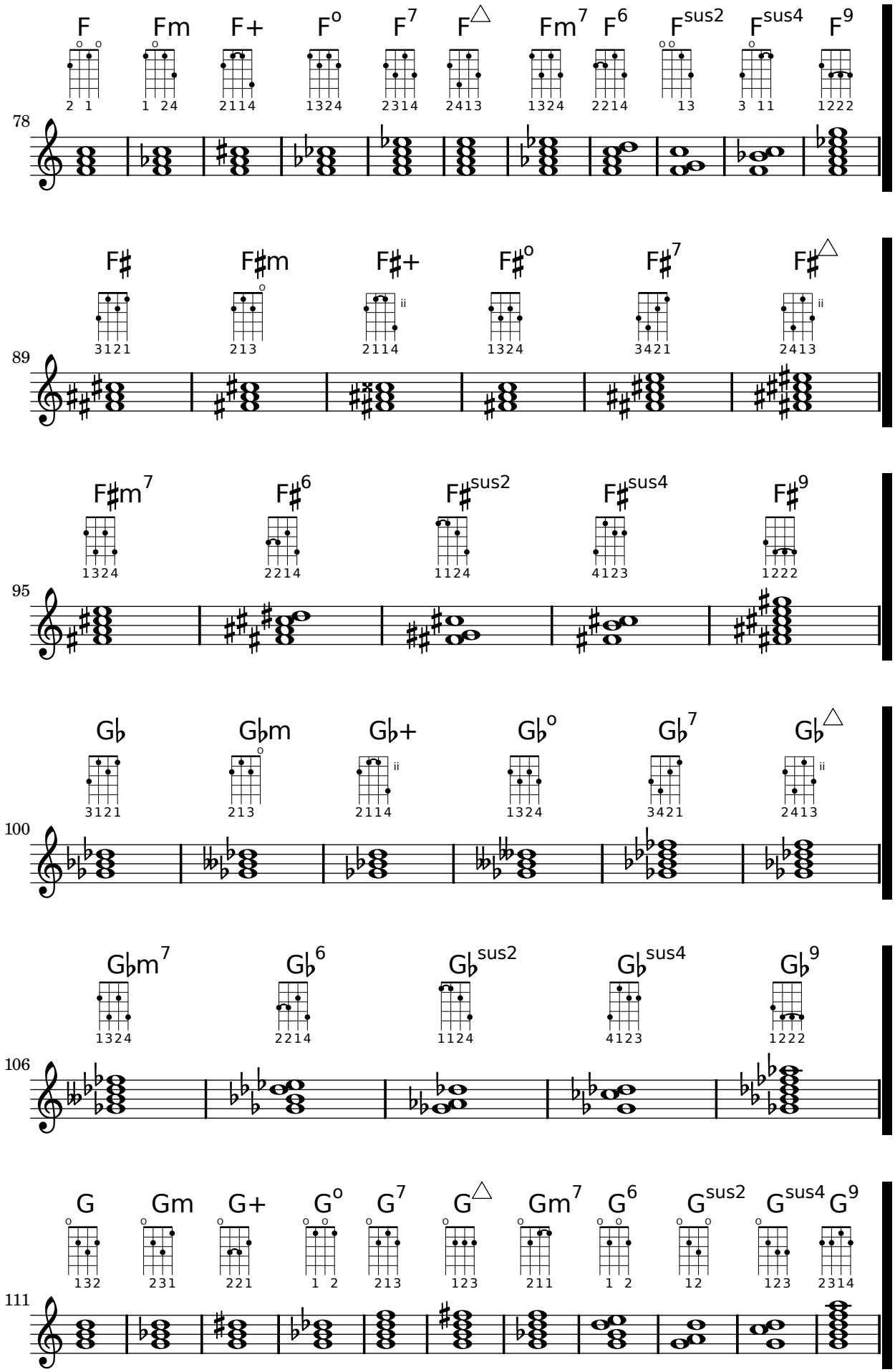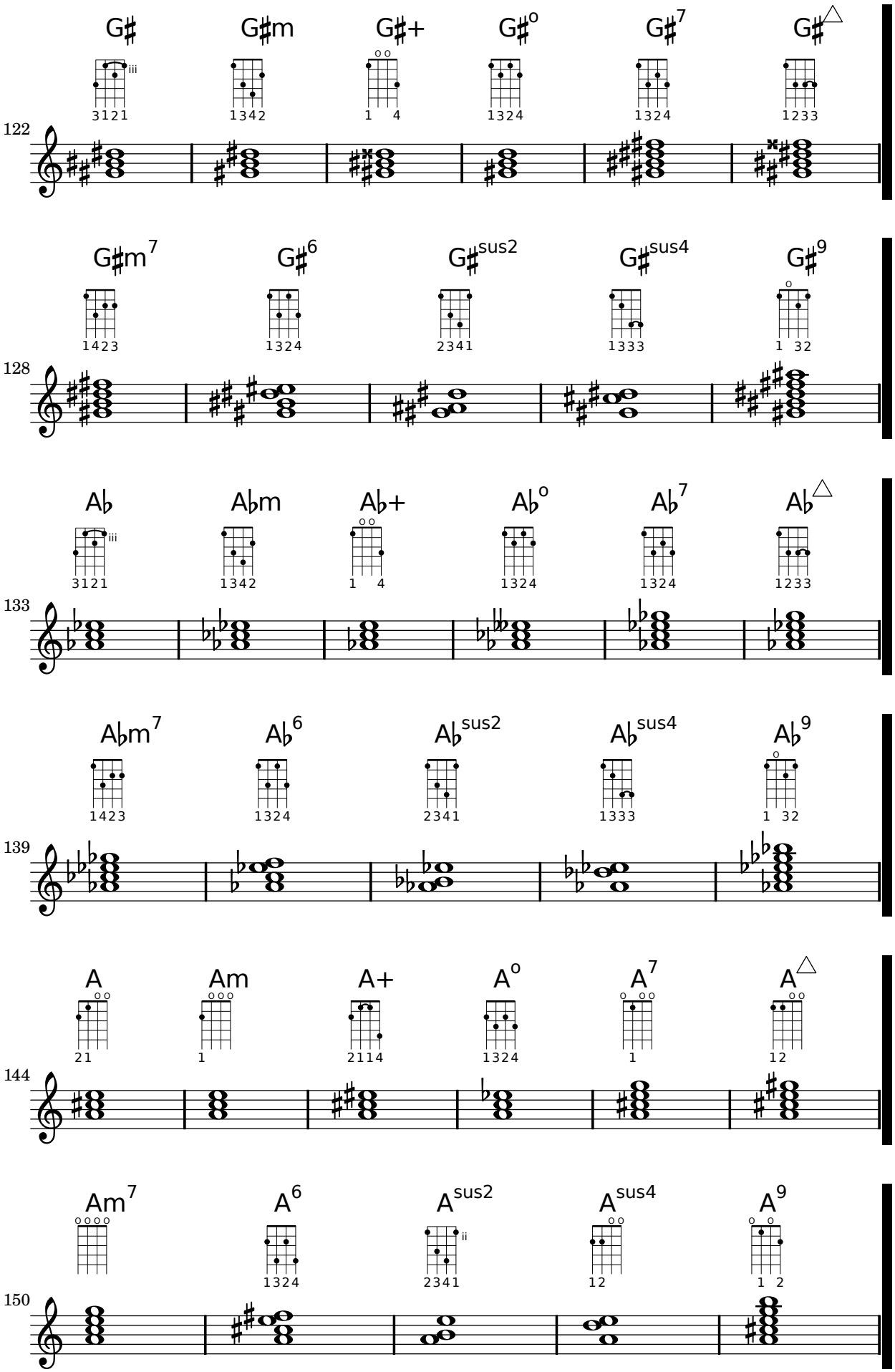






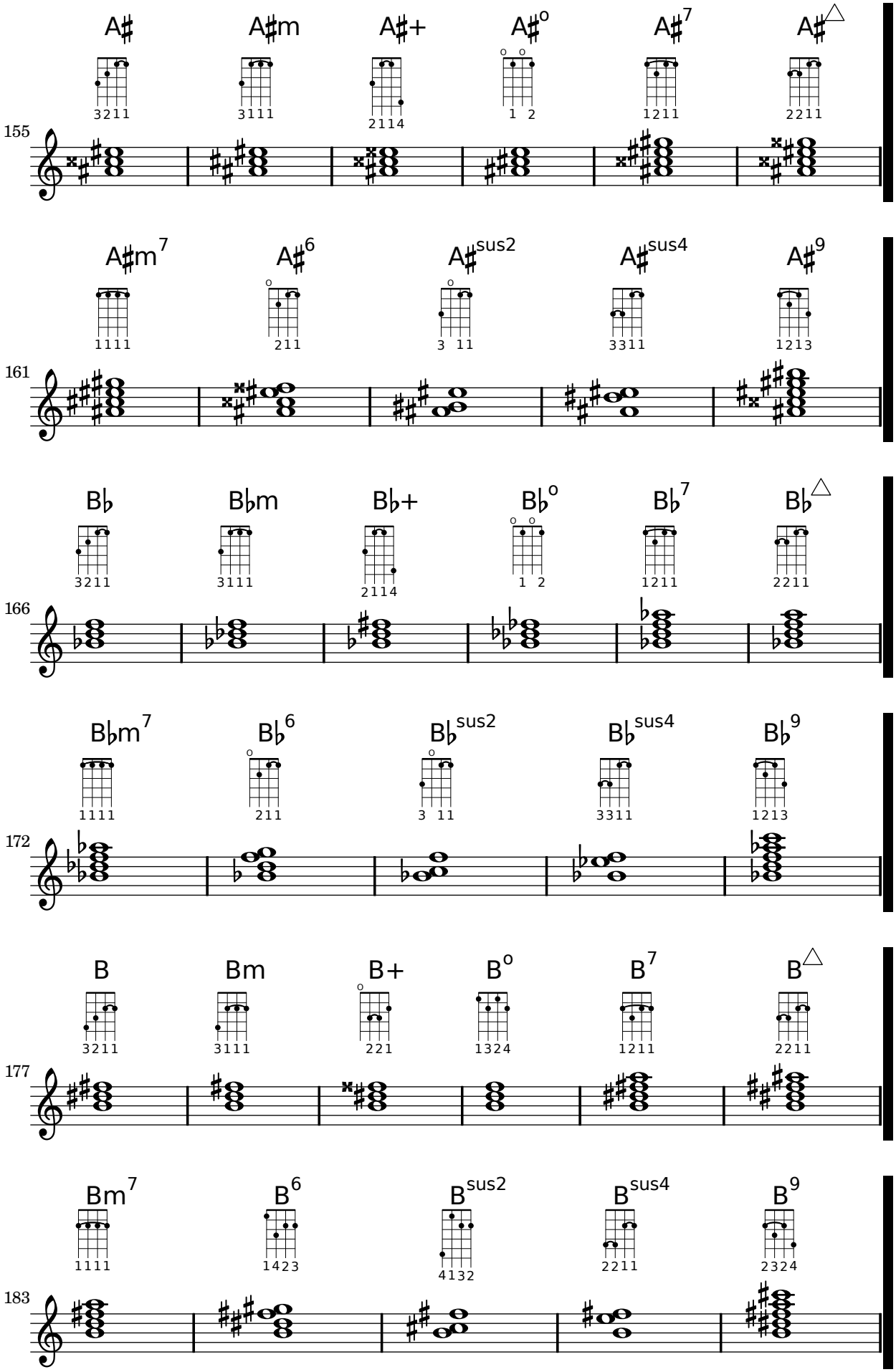





D Dm D+ D° D7 D△ Dm7 D6 Dsus2 Dsus4 D9
1 2 3 2 2 1 2 1 1 4 1 3 2 4 1 1 1 2 1 1 1 3 2 2 1 3 1 1 1 1 1 2 1 2 1 3 1 2
34
D♯ D♯m D♯+ D♯° D♯7 D♯△
2 2 1 3 3 2 1 2 2 1 1 3 1 4 1 1 1 2 1 2 1 2
45
D♯m7 D♯6 D♯sus2 D♯sus4 D♯9
2 2 1 4 1 1 1 1 2 2 1 1 2 3 4 1 1 1 1
51
E♭ E♭m E♭+ E♭° E♭7 E♭△
2 2 1 3 3 2 1 2 2 1 1 3 1 4 1 1 1 2 1 2 1 2
56
E♭m7 E♭6 E♭sus2 E♭sus4 E♭9
2 2 1 4 1 1 1 1 2 2 1 1 2 3 4 1 1 1 1
62
E Em E+ E° E7 E△ Em7 E6 Esus2 Esus4 E9
2 3 4 1 3 3 2 1 1 4 1 2 1 2 3 1 3 2 1 2 1 1 1 1 3 3 1 1 2 4 1 1 2 2 2
67


F Fm F+ F° F7 F△ Fm7 F6 Fsus2 Fsus4 F9
2 1 1 24 2114 1324 2314 2413 1324 2214 13 3 11 1222
78
F♯ F♯m F♯+ F♯° F♯7 F♯△
3121 213 2114 1324 3421 2413
89
F♯m7 F♯6 F♯sus2 F♯sus4 F♯9
1324 2214 1124 4123 1222
95
G♭ G♭m G♭+ G♭° G♭7 G♭△
3121 213 2114 1324 3421 2413
100
G♭m7 G♭6 G♭sus2 G♭sus4 G♭9
1324 2214 1124 4123 1222
106
G Gm G+ G° G7 G△ Gm7 G6 Gsus2 Gsus4 G9
132 231 221 1 2 213 123 211 1 2 12 123 2314
111


G♯ G♯m G♯+ G♯° G♯7 G♯△
3121 1342 1 4 1324 1324 1233
122
G♯m7 G♯6 G♯sus2 G♯sus4 G♯9
1423 1324 2341 1333 1 32
128
A♭ A♭m A♭+ A♭° A♭7 A♭△
3121 1342 1 4 1324 1324 1233
133
A♭m7 A♭6 A♭sus2 A♭sus4 A♭9
1423 1324 2341 1333 1 32
139
A Am A+ A° A7 A△
21 1 2114 1324 1 12
144
Am7 A6 Asus2 Asus4 A9
1324 2341 12 1 2
150


A♯ A♯m A♯+ A♯° A♯7 A♯△
3211 3111 2114 1 2 1211 2211
155
A♯m7 A♯6 A♯sus2 A♯sus4 A♯9
1111 211 3 11 3311 1213
161
B♭ B♭m B♭+ B♭° B♭7 B♭△
3211 3111 2114 1 2 1211 2211
166
B♭m7 B♭6 B♭sus2 B♭sus4 B♭9
1111 211 3 11 3311 1213
172
B Bm B+ B° B7 B△
3211 3111 221 1324 1211 2211
177
Bm7 B6 Bsus2 Bsus4 B9
1111 1423 4132 2211 2324
183

A.4 Instruments MIDI

La liste suivante répertorie les différentes dénominations que vous pouvez affecter à la propriété `midiInstrument`. L'ordre dans lequel ils sont rangés, par colonne, correspond aux 128 programmes du standard *General MIDI*.

```
acoustic grand
bright acoustic
electric grand
honky-tonk
electric piano 1
electric piano 2
harpsichord
clav
celesta
glockenspiel
music box
vibraphone
marimba
xylophone
tubular bells
dulcimer
drawbar organ
percussive organ
rock organ
church organ
reed organ
accordion
harmonica
concertina
acoustic guitar (nylon)
acoustic guitar (steel)
electric guitar (jazz)
electric guitar (clean)
electric guitar (muted)
overdriven guitar
distorted guitar
guitar harmonics
acoustic bass
electric bass (finger)
electric bass (pick)
fretless bass
slap bass 1
slap bass 2
synth bass 1
synth bass 2
violin
viola
cello
contrabass
tremolo strings
pizzicato strings
orchestral harp
timpani
string ensemble 1
string ensemble 2
synthstrings 1
synthstrings 2
choir aahs
voice oohs
synth voice
orchestra hit
trumpet
trombone
tuba
muted trumpet
french horn
brass section
synthbrass 1
synthbrass 2
soprano sax
alto sax
tenor sax
baritone sax
oboe
english horn
bassoon
clarinet
piccolo
flute
recorder
pan flute
blown bottle
shakuhachi
whistle
ocarina
lead 1 (square)
lead 2 (sawtooth)
lead 3 (calliope)
lead 4 (chiff)
lead 5 (charang)
lead 6 (voice)
lead 7 (fifths)
lead 8 (bass+lead)
pad 1 (new age)
pad 2 (warm)
pad 3 (polysynth)
pad 4 (choir)
pad 5 (bowed)
pad 6 (metallic)
pad 7 (halo)
pad 8 (sweep)
fx 1 (rain)
fx 2 (soundtrack)
fx 3 (crystal)
fx 4 (atmosphere)
fx 5 (brightness)
fx 6 (goblins)
fx 7 (echoes)
fx 8 (sci-fi)
sitar
banjo
shamisen
koto
kalimba
bagpipe
fiddle
shanai
tinkle bell
agogo
steel drums
woodblock
taiko drum
melodic tom
synth drum
reverse cymbal
guitar fret noise
breath noise
seashore
bird tweet
telephone ring
helicopter
applause
gunshot
```

A.5 Liste des couleurs

Couleurs de base

La syntaxe appropriée à la gestion des couleurs est traitée au chapitre [Coloration d'objets], page 186.

```
black       white          red         green
blue        cyan           magenta     yellow
grey        darkred        darkgreen   darkblue
darkcyan    darkmagenta    darkyellow
```

Noms de couleur X

Les noms de couleur X peuvent s'employer de différentes manières.

Un nom de couleur peut s'écrire sous la forme d'un mot composé et sans espace auquel chaque élément prend un initiale capitalisée (p.ex. `LightSlateBlue`). Il peut aussi s'exprimer sous la forme d'une suite de mots, auquel cas les initiales ne sont pas capitalisées (p.ex. `light slate blue`).

Le *gris* accepte aussi bien l'orthographe « grey » que « gray » (p.ex. `DarkSlateGray`).

Certains noms peuvent prendre un suffixe numérique, comme `LightSalmon4`.

Noms de couleur sans suffixe numérique

```
snow              GhostWhite       WhiteSmoke       gainsboro        FloralWhite
OldLace           linen            AntiqueWhite     PapayaWhip       BlanchedAlmond
bisque            PeachPuff        NavajoWhite      moccasin         cornsilk
ivory             LemonChiffon     seashell         honeydew         MintCream
azure             AliceBlue        lavender         LavenderBlush    MistyRose
white             black            DarkSlateGrey    DimGrey          SlateGrey
LightSlateGrey    grey             LightGrey        MidnightBlue     navy
NavyBlue          CornflowerBlue   DarkSlateBlue    SlateBlue        MediumSlateBlue
LightSlateBlue    MediumBlue       RoyalBlue        blue             DodgerBlue
DeepSkyBlue       SkyBlue          LightSkyBlue     SteelBlue        LightSteelBlue
LightBlue         PowderBlue       PaleTurquoise    DarkTurquoise    MediumTurquoise
turquoise         cyan             LightCyan        CadetBlue        MediumAquamarine
aquamarine        DarkGreen        DarkOliveGreen   DarkSeaGreen     SeaGreen
MediumSeaGreen    LightSeaGreen    PaleGreen        SpringGreen      LawnGreen
green             chartreuse       MediumSpringGreen       GreenYellow      LimeGreen
YellowGreen       ForestGreen      OliveDrab        DarkKhaki        khaki
PaleGoldenrod     LightGoldenrodYellow     LightYellow      yellow   gold
LightGoldenrod    goldenrod        DarkGoldenrod    RosyBrown        IndianRed
SaddleBrown       sienna           peru             burlywood        beige
wheat             SandyBrown       tan              chocolate        firebrick
brown             DarkSalmon       salmon           LightSalmon      orange
DarkOrange        coral            LightCoral       tomato           OrangeRed
red               HotPink          DeepPink         pink             LightPink
PaleVioletRed     maroon           MediumVioletRed  VioletRed        magenta
violet            plum             orchid           MediumOrchid     DarkOrchid
DarkViolet        BlueViolet       purple           MediumPurple     thistle
DarkGrey          DarkBlue         DarkCyan         DarkMagenta      DarkRed
LightGreen
```

Noms de couleur avec suffixe numérique

Les couleurs suivantes acceptent un suffixe numérique N compris entre 1 et 4.

```
snowN             seashellN        AntiqueWhiteN    bisqueN          PeachPuffN
```

NavajoWhiteN	LemonChiffonN	cornsilkN	ivoryN	honeydewN
LavenderBlushN	MistyRoseN	azureN	SlateBlueN	RoyalBlueN
blueN	DodgerBlueN	SteelBlueN	DeepSkyBlueN	SkyBlueN
LightSkyBlueN	LightSteelBlueN	LightBlueN	LightCyanN	PaleTurquoiseN
CadetBlueN	turquoiseN	cyanN	aquamarineN	DarkSeaGreenN
SeaGreenN	PaleGreenN	SpringGreenN	greenN	chartreuseN
OliveDrabN	DarkOliveGreenN	khakiN	LightGoldenrodN	LightYellowN
yellowN	goldN	goldenrodN	DarkGoldenrodN	RosyBrownN
IndianRedN	siennaN	burlywoodN	wheatN	tanN
chocolateN	firebrickN	brownN	salmonN	LightSalmonN
orangeN	DarkOrangeN	coralN	tomatoN	OrangeRedN
redN	DeepPinkN	HotPinkN	pinkN	LightPinkN
PaleVioletRedN	maroonN	VioletRedN	magentaN	orchidN
plumN	MediumOrchidN	DarkOrchidN	purpleN	MediumPurpleN
thistleN				

Échelle de gris

Une échelle de gris s'obtient en utilisant

`greyN`

où N est compris entre 0 et 100.

A.6 La fonte Feta

Voici, regroupés par type, les différents symboles de la fonte Emmentaler qui peuvent s'inclure directement dans un objet *markup*. Pour ce faire, il suffit d'employer le nom du glyphe, comme dans `g^\markup {\musicglyph #"scripts.segno" }` ou `\markup {\musicglyph #"five"}`. Pour de plus amples informations, reportez-vous au chapitre Section 1.8.2 [Mise en forme du texte], page 202.

Glyphes de clef

`clefs.C`		`clefs.C_change`	
`clefs.F`		`clefs.F_change`	
`clefs.G`		`clefs.G_change`	
`clefs.percussion`		`clefs.percussion_change`	
`clefs.tab`		`clefs.tab_change`	

Glyphes de métrique

`timesig.C44`		`timesig.C22`	

Glyphes de chiffre

plus	+	comma	,
hyphen	-	period	.
zero	0	one	1
two	2	three	3
four	4	five	5
six	6	seven	7
eight	8	nine	9

Glyphes d'altération

accidentals.sharp	♯	accidentals .sharp.arrowup	♯
accidentals .sharp.arrowdown	♯	accidentals .sharp.arrowboth	♯
accidentals.sharp .slashslash.stem	♯	accidentals.sharp .slashslashslash.stemstem	♯
accidentals.sharp .slashslashslash.stem	♯	accidentals.sharp .slashslash.stemstemstem	♯
accidentals.natural	♮	accidentals .natural.arrowup	♮
accidentals .natural.arrowdown	♮	accidentals .natural.arrowboth	♮
accidentals.flat	♭	accidentals.flat.arrowup	♭

accidentals .flat.arrowdown		accidentals .flat.arrowboth	
accidentals.flat.slash		accidentals.flat .slashslash	
accidentals .mirroredflat.flat		accidentals.mirroredflat	
accidentals .mirroredflat.backslash		accidentals.flatflat	
accidentals .flatflat.slash		accidentals.doublesharp	
accidentals.rightparen		accidentals.leftparen	

Glyphes de tête de note par défaut

noteheads.uM2		noteheads.dM2	
noteheads.sM1		noteheads.s0	
noteheads.s1		noteheads.s2	

Glyphes de tête de note spéciale

noteheads.sM1double		noteheads.s0diamond	
noteheads.s1diamond		noteheads.s2diamond	
noteheads.s0triangle		noteheads.d1triangle	
noteheads.u1triangle		noteheads.u2triangle	
noteheads.d2triangle		noteheads.s0slash	

noteheads.s1slash		noteheads.s2slash	
noteheads.s0cross		noteheads.s1cross	
noteheads.s2cross		noteheads.s2xcircle	
noteheads.s0harmonic		noteheads.s2harmonic	

Glyphes de tête de note à forme variable

noteheads.s0do		noteheads.d1do	
noteheads.u1do		noteheads.d2do	
noteheads.u2do		noteheads.s0doThin	
noteheads.d1doThin		noteheads.u1doThin	
noteheads.d2doThin		noteheads.u2doThin	
noteheads.s0re		noteheads.u1re	
noteheads.d1re		noteheads.u2re	
noteheads.d2re		noteheads.s0reThin	
noteheads.u1reThin		noteheads.d1reThin	
noteheads.u2reThin		noteheads.d2reThin	
noteheads.s0mi		noteheads.s1mi	

noteheads.s2mi		noteheads.s0miMirror	
noteheads.s1miMirror		noteheads.s2miMirror	
noteheads.s0miThin		noteheads.s1miThin	
noteheads.s2miThin		noteheads.u0fa	
noteheads.d0fa		noteheads.u1fa	
noteheads.d1fa		noteheads.u2fa	
noteheads.d2fa		noteheads.u0faThin	
noteheads.d0faThin		noteheads.u1faThin	
noteheads.d1faThin		noteheads.u2faThin	
noteheads.d2faThin		noteheads.s0sol	
noteheads.s1sol		noteheads.s2sol	
noteheads.s0la		noteheads.s1la	
noteheads.s2la		noteheads.s0laThin	
noteheads.s1laThin		noteheads.s2laThin	
noteheads.s0ti		noteheads.u1ti	

noteheads.d1ti		noteheads.u2ti	
noteheads.d2ti		noteheads.s0tiThin	
noteheads.u1tiThin		noteheads.d1tiThin	
noteheads.u2tiThin		noteheads.d2tiThin	
noteheads.u0doFunk		noteheads.d0doFunk	
noteheads.u1doFunk		noteheads.d1doFunk	
noteheads.u2doFunk		noteheads.d2doFunk	
noteheads.u0reFunk		noteheads.d0reFunk	
noteheads.u1reFunk		noteheads.d1reFunk	
noteheads.u2reFunk		noteheads.d2reFunk	
noteheads.u0miFunk		noteheads.d0miFunk	
noteheads.u1miFunk		noteheads.d1miFunk	
noteheads.s2miFunk		noteheads.u0faFunk	
noteheads.d0faFunk		noteheads.u1faFunk	
noteheads.d1faFunk		noteheads.u2faFunk	

noteheads.d2faFunk		noteheads.s0solFunk	
noteheads.s1solFunk		noteheads.s2solFunk	
noteheads.s0laFunk		noteheads.s1laFunk	
noteheads.s2laFunk		noteheads.u0tiFunk	
noteheads.d0tiFunk		noteheads.u1tiFunk	
noteheads.d1tiFunk		noteheads.u2tiFunk	
noteheads.d2tiFunk		noteheads.s0doWalker	
noteheads.u1doWalker		noteheads.d1doWalker	
noteheads.u2doWalker		noteheads.d2doWalker	
noteheads.s0reWalker		noteheads.u1reWalker	
noteheads.d1reWalker		noteheads.u2reWalker	
noteheads.d2reWalker		noteheads.s0miWalker	
noteheads.s1miWalker		noteheads.s2miWalker	
noteheads.s0faWalker		noteheads.u1faWalker	
noteheads.d1faWalker		noteheads.u2faWalker	

noteheads.d2faWalker		noteheads.s0laWalker	
noteheads.s1laWalker		noteheads.s2laWalker	
noteheads.s0tiWalker		noteheads.u1tiWalker	
noteheads.d1tiWalker		noteheads.u2tiWalker	
noteheads.d2tiWalker			

Glyphes de silence

rests.0		rests.1	
rests.0o		rests.1o	
rests.M3		rests.M2	
rests.M1		rests.2	
rests.2classical		rests.3	
rests.4		rests.5	
rests.6		rests.7	

Glyphes de crochet de croche

flags.u3		flags.u4	
flags.u5		flags.u6	
flags.u7		flags.d3	
flags.d4		flags.d5	
flags.d6		flags.d7	
flags.ugrace		flags.dgrace	

Glyphes de point

dots.dot	

Glyphes de nuance

space		f	
m		p	
r		s	
z			

Glyphes de script

scripts.ufermata		scripts.dfermata	
scripts.ushortfermata		scripts.dshortfermata	
scripts.ulongfermata		scripts.dlongfermata	
scripts.uverylongfermata		scripts.dverylongfermata	
scripts.thumb		scripts.sforzato	
scripts.espr		scripts.staccato	
scripts.ustaccatissimo		scripts.dstaccatissimo	
scripts.tenuto		scripts.uportato	
scripts.dportato		scripts.umarcato	
scripts.dmarcato		scripts.open	
scripts.halfopen		scripts.stopped	
scripts.upbow		scripts.downbow	
scripts.reverseturn		scripts.turn	
scripts.trill		scripts.upedalheel	

scripts.dpedalheel		scripts.upedaltoe	
scripts.dpedaltoe		scripts.flageolet	
scripts.segno		scripts.varsegno	
scripts.coda		scripts.varcoda	
scripts.rcomma		scripts.lcomma	
scripts.rvarcomma		scripts.lvarcomma	
scripts.arpeggio		scripts.trill_element	
scripts.arpeggio .arrow.M1		scripts.arpeggio.arrow.1	
scripts.trilelement		scripts.prall	
scripts.mordent		scripts.prallprall	
scripts.prallmordent		scripts.upprall	
scripts.upmordent		scripts.pralldown	
scripts.downprall		scripts.downmordent	
scripts.prallup		scripts.lineprall	
scripts.caesura.curved		scripts.caesura.straight	

scripts.snappizzicato		scripts.ictus	
scripts.uaccentus		scripts.daccentus	
scripts.usemicirculus		scripts.dsemicirculus	
scripts.circulus		scripts.augmentum	
scripts .usignumcongruentiae		scripts .dsignumcongruentiae	

Glyphes de flèche

arrowheads.open.01		arrowheads.open.0M1	
arrowheads.open.11		arrowheads.open.1M1	
arrowheads.close.01		arrowheads.close.0M1	
arrowheads.close.11		arrowheads.close.1M1	

Glyphes d'extrémité d'accolade

brackettips.up		brackettips.down	

Glyphes de pédale

pedal.*		pedal.M	
pedal..		pedal.P	
pedal.d		pedal.e	
pedal.Ped			

Glyphes d'accordéon

accordion.discant		accordion.dot	
accordion.freebass		accordion.stdbass	
accordion.bayanbass		accordion.oldEE	
accordion.push		accordion.pull	

Glyphes de style vaticana

noteheads .svaticana.punctum		noteheads.svaticana .punctum.cavum	
noteheads.svaticana .linea.punctum		noteheads.svaticana .linea.punctum.cavum	
noteheads.svaticana .inclinatum		noteheads.svaticana.lpes	
noteheads .svaticana.vlpes		noteheads.svaticana.upes	
noteheads .svaticana.vupes		noteheads .svaticana.plica	
noteheads .svaticana.vplica		noteheads .svaticana.epiphonus	
noteheads.svaticana .vepiphonus		noteheads.svaticana .reverse.plica	
noteheads.svaticana .reverse.vplica		noteheads.svaticana .inner.cephalicus	
noteheads.svaticana .cephalicus		noteheads .svaticana.quilisma	

clefs.vaticana.do

clefs.vaticana.do_change

clefs.vaticana.fa 

clefs.vaticana.fa_change 

custodes.vaticana.u0

custodes.vaticana.u1

custodes.vaticana.u2

custodes.vaticana.d0

custodes.vaticana.d1

custodes.vaticana.d2

accidentals.vaticanaM1

accidentals.vaticana0

dots.dotvaticana .

Glyphes de style medicaea

noteheads.smedicaea .inclinatum

noteheads .smedicaea.punctum

noteheads .smedicaea.rvirga

noteheads .smedicaea.virga

clefs.medicaea.do

clefs.medicaea.do_change

clefs.medicaea.fa

clefs.medicaea.fa_change

custodes.medicaea.u0

custodes.medicaea.u1

custodes.medicaea.u2

custodes.medicaea.d0

custodes.medicaea.d1

custodes.medicaea.d2

accidentals.medicaeaM1

Glyphes de style Hufnagel

noteheads .shufnagel.punctum		noteheads .shufnagel.virga	
noteheads.shufnagel.lpes		clefs.hufnagel.do	
clefs.hufnagel.do_change		clefs.hufnagel.fa	
clefs.hufnagel.fa_change		clefs.hufnagel.do.fa	
clefs.hufnagel .do.fa_change		custodes.hufnagel.u0	
custodes.hufnagel.u1		custodes.hufnagel.u2	
custodes.hufnagel.d0		custodes.hufnagel.d1	
custodes.hufnagel.d2		accidentals.hufnagelM1	

Glyphes de style mensural

rests.M3mensural		rests.M2mensural	
rests.M1mensural		rests.0mensural	
rests.1mensural		rests.2mensural	
rests.3mensural		rests.4mensural	
noteheads.slmensural		noteheads.sM3mensural	
noteheads.sM3ligmensural		noteheads.sM2mensural	

noteheads.sM1mensural 

noteheads .sM3blackmensural 

noteheads .sM3blackligmensural 

noteheads .sM2blackmensural 

noteheads .sM1blackmensural 

noteheads .sM3semimensural 

noteheads .sM3semiligmensural 

noteheads .sM2semimensural 

noteheads .sM1semimensural 

noteheads.s0mensural

noteheads.s1mensural

noteheads.s2mensural

noteheads .s0blackmensural

clefs.mensural.c 

clefs.mensural.c_change 

clefs.mensural.f 

clefs.mensural.f_change 

clefs.mensural.g 

clefs.mensural.g_change 

custodes.mensural.u0 

custodes.mensural.u1 

custodes.mensural.u2 

custodes.mensural.d0 

custodes.mensural.d1 

custodes.mensural.d2 

accidentals.mensural1

accidentals.mensuralM1

flags.mensuralu03 

flags.mensuralu13

flags.mensuralu23

flags.mensurald03		flags.mensurald13	
flags.mensurald23		flags.mensuralu04	
flags.mensuralu14		flags.mensuralu24	
flags.mensurald04		flags.mensurald14	
flags.mensurald24		flags.mensuralu05	
flags.mensuralu15		flags.mensuralu25	
flags.mensurald05		flags.mensurald15	
flags.mensurald25		flags.mensuralu06	
flags.mensuralu16		flags.mensuralu26	
flags.mensurald06		flags.mensurald16	
flags.mensurald26		timesig.mensural44	
timesig.mensural22		timesig.mensural32	
timesig.mensural64		timesig.mensural94	
timesig.mensural34		timesig.mensural68	

timesig.mensural98		timesig.mensural48	
timesig.mensural68alt		timesig.mensural24	

Glyphes de style néomensural

rests.M3neomensural		rests.M2neomensural	
rests.M1neomensural		rests.0neomensural	
rests.1neomensural		rests.2neomensural	
rests.3neomensural		rests.4neomensural	
noteheads.slneomensural		noteheads.sM3neomensural	
noteheads.sM2neomensural		noteheads.sM1neomensural	
noteheads.s0neomensural		noteheads.s1neomensural	
noteheads.s2neomensural		clefs.neomensural.c	
clefs.neomensural .c_change		timesig.neomensural44	
timesig.neomensural22		timesig.neomensural32	
timesig.neomensural64		timesig.neomensural94	
timesig.neomensural34		timesig.neomensural68	
timesig.neomensural98		timesig.neomensural48	
timesig.neomensural68alt		timesig.neomensural24	

Glyphes de style Petrucci

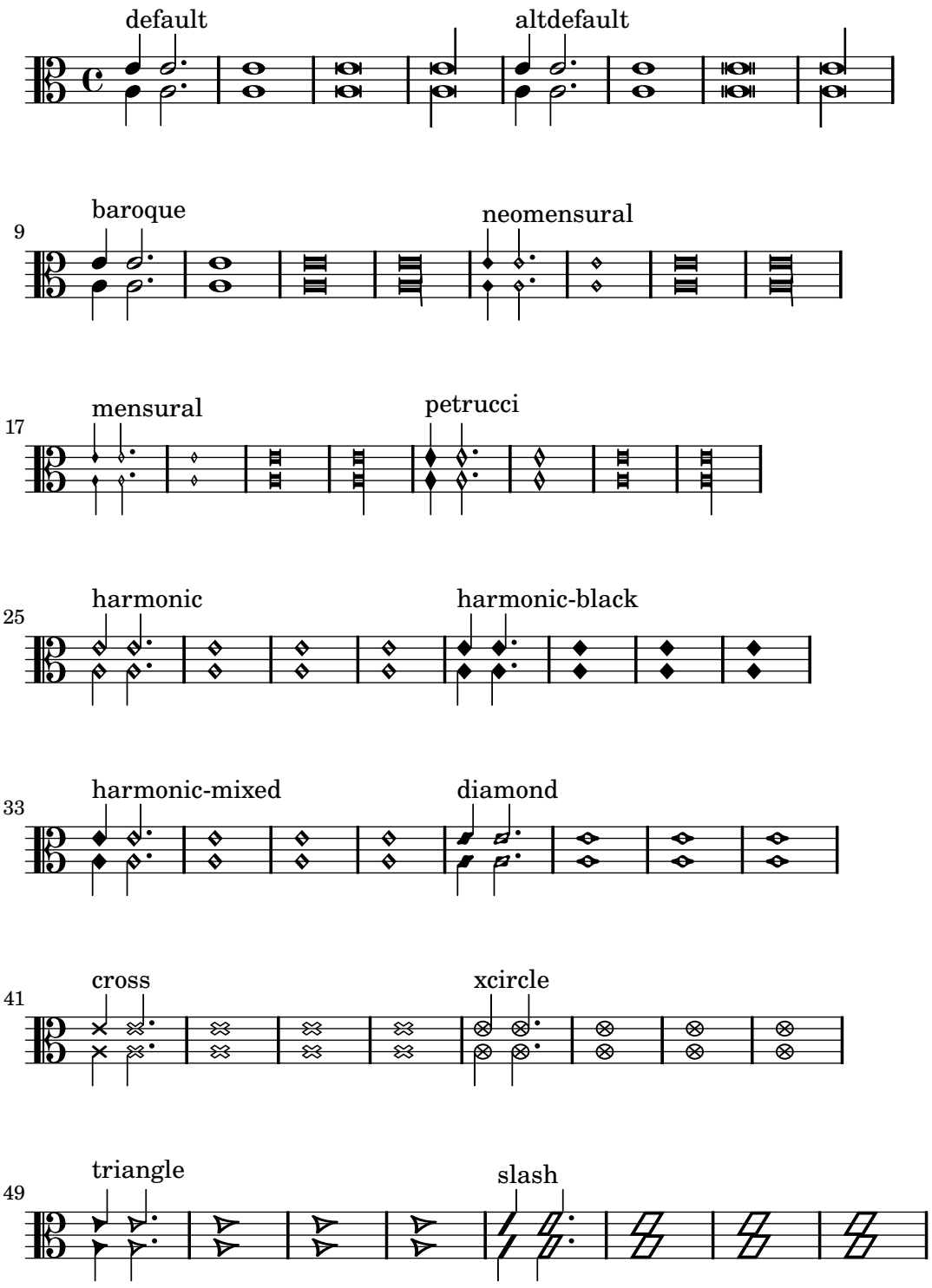
noteheads.s0petrucci		noteheads.s1petrucci	
noteheads.s2petrucci		noteheads .s0blackpetrucci	
noteheads .s1blackpetrucci		noteheads .s2blackpetrucci	
clefs.petrucci.c1		clefs.petrucci.c1_change	
clefs.petrucci.c2		clefs.petrucci.c2_change	
clefs.petrucci.c3		clefs.petrucci.c3_change	
clefs.petrucci.c4		clefs.petrucci.c4_change	
clefs.petrucci.c5		clefs.petrucci.c5_change	
clefs.petrucci.f		clefs.petrucci.f_change	
clefs.petrucci.g		clefs.petrucci.g_change	

Glyphes de style Solesmes

noteheads.ssolesmes .incl.parvum		noteheads .ssolesmes.auct.asc	
noteheads .ssolesmes.auct.desc		noteheads.ssolesmes .incl.auctum	
noteheads .ssolesmes.stropha		noteheads.ssolesmes .stropha.aucta	
noteheads .ssolesmes.oriscus			

A.7 Styles de tête de note

Voici les différents styles de tête de note disponibles.



A.8 Text markup commands

Les commandes suivantes peuvent être utilisées dans un bloc `\markup { }`.

The following commands can all be used inside `\markup { }`.

A.8.1 Font

`\abs-fontsize` *size* (number) *arg* (markup)

Use *size* as the absolute font size to display *arg*. Adjusts `baseline-skip` and `word-space` accordingly.

```
\markup {
  default text font size
  \hspace #2
  \abs-fontsize #16 { text font size 16 }
  \hspace #2
  \abs-fontsize #12 { text font size 12 }
}
```



`\bold` *arg* (markup)

Switch to bold font-series.

```
\markup {
  default
  \hspace #2
  \bold
  bold
}
```



`\box` *arg* (markup)

Draw a box round *arg*. Looks at `thickness`, `box-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \override #'(box-padding . 0.5)
  \box
  \line { V. S. }
}
```



Used properties:

- `box-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

`\caps` *arg* (markup)

Copy of the `\smallCaps` command.

```
\markup {
  default
  \hspace #2
  \caps {
    Text in small caps
  }
}
```

default TEXT IN SMALL CAPS

`\dynamic` *arg* (markup)
Use the dynamic font. This font only contains **s**, **f**, **m**, **z**, **p**, and **r**. When producing phrases, like 'più **f**', the normal words (like 'più') should be done in a different font. The recommended font for this is bold and italic.

```
\markup {
  \dynamic {
    sfzp
  }
}
```

sfzp

`\finger` *arg* (markup)
Set *arg* as small numbers.

```
\markup {
  \finger {
    1 2 3 4 5
  }
}
```

1 2 3 4 5

`\fontCaps` *arg* (markup)
Set `font-shape` to `caps`

Note: `\fontCaps` requires the installation and selection of fonts which support the `caps` font shape.

`\fontsize` *increment* (number) *arg* (markup)
Add *increment* to the font-size. Adjusts `baseline-skip` accordingly.

```
\markup {
  default
  \hspace #2
  \fontsize #-1.5
  smaller
}
```

default smaller

Used properties:

- `baseline-skip` (2)
- `word-space` (1)
- `font-size` (0)

`\huge` *arg* (markup)
Set font size to +2.

```
\markup {
  default
  \hspace #2
  \huge
  huge
```

```
}
```

default **huge**

`\italic` *arg* (markup)
Use italic `font-shape` for *arg*.

```
\markup {
  default
  \hspace #2
  \italic
  italic
}
```

default *italic*

`\large` *arg* (markup)
Set font size to +1.

```
\markup {
  default
  \hspace #2
  \large
  large
}
```

default large

`\larger` *arg* (markup)
Increase the font size relative to the current setting.

```
\markup {
  default
  \hspace #2
  \larger
  larger
}
```

default larger

`\magnify` *sz* (number) *arg* (markup)
Set the font magnification for its argument. In the following example, the middle A is 10% larger:

```
A \magnify #1.1 { A } A
```

Note: Magnification only works if a font name is explicitly selected. Use `\fontsize` otherwise.

```
\markup {
  default
  \hspace #2
  \magnify #1.5 {
    50% larger
  }
}
```

default 50% larger

`\medium` *arg* (markup)
Switch to medium font-series (in contrast to bold).

```
\markup {
  \bold {
    some bold text
    \hspace #2
    \medium {
      medium font series
    }
    \hspace #2
    bold again
  }
}
```

some bold text medium font series **bold again**

`\normal-size-sub` *arg* (markup)
Set *arg* in subscript with a normal font size.

```
\markup {
  default
  \normal-size-sub {
    subscript in standard size
  }
}
```

default $_{\text{subscript in standard size}}$

Used properties:

- `baseline-skip`

`\normal-size-super` *arg* (markup)
Set *arg* in superscript with a normal font size.

```
\markup {
  default
  \normal-size-super {
    superscript in standard size
  }
}
```

default $^{\text{superscript in standard size}}$

Used properties:

- `baseline-skip`

`\normal-text` *arg* (markup)
Set all font related properties (except the size) to get the default normal text font, no matter what font was used earlier.

```
\markup {
  \huge \bold \sans \caps {
    Some text with font overrides
```

```
    \hspace #2
    \normal-text {
      Default text, same font-size
    }
    \hspace #2
    More text as before
  }
}
```

SOME TEXT WITH FONT OVERRIDES Default text, same font-size **MORE**

`\normalsize` *arg* (markup)
Set font size to default.

```
\markup {
  \teeny {
    this is very small
    \hspace #2
    \normalsize {
      normal size
    }
    \hspace #2
    teeny again
  }
}
```

this is very small normal size teeny again

`\number` *arg* (markup)
Set font family to `number`, which yields the font used for time signatures and fingerings. This font contains numbers and some punctuation; it has no letters.

```
\markup {
  \number {
    0 1 2 3 4 5 6 7 8 9 . ,
  }
}
```

0123456789.,

`\roman` *arg* (markup)
Set font family to `roman`.

```
\markup {
  \sans \bold {
    sans serif, bold
    \hspace #2
    \roman {
      text in roman font family
    }
    \hspace #2
    return to sans
  }
}
```

sans serif, bold **text in roman font family** **return to sans**

`\sans` *arg* (markup)
Switch to the sans serif font family.

```
\markup {
  default
  \hspace #2
  \sans {
    sans serif
  }
}
```

default sans serif

`\simple` *str* (string)
A simple text string; `\markup { foo }` is equivalent with `\markup { \simple #"foo" }`.

Note: for creating standard text markup or defining new markup commands, the use of `\simple` is unnecessary.

```
\markup {
  \simple #"simple"
  \simple #"text"
  \simple #"strings"
}
```

simple text strings

`\small` *arg* (markup)
Set font size to -1.

```
\markup {
  default
  \hspace #2
  \small
  small
}
```

default small

`\smallCaps` *arg* (markup)
Emit *arg* as small caps.

Note: `\smallCaps` does not support accented characters.

```
\markup {
  default
  \hspace #2
  \smallCaps {
    Text in small caps
  }
}
```

default Text in small caps

`\smaller` *arg* (markup)
Decrease the font size relative to the current setting.

```
\markup {
  \fontsize #3.5 {
    some large text
    \hspace #2
    \smaller {
      a bit smaller
    }
    \hspace #2
    more large text
  }
}
```

some large text a bit smaller more large text

`\sub` *arg* (markup)
Set *arg* in subscript.

```
\markup {
  \concat {
    H
    \sub {
      2
    }
    O
  }
}
```

H_2O

Used properties:

- `baseline-skip`
- `font-size` (0)

`\super` *arg* (markup)
Set *arg* in superscript.

```
\markup {
  E =
  \concat {
    mc
    \super
    2
  }
}
```

$E = mc^2$

Used properties:

- `baseline-skip`
- `font-size` (0)

`\teeny` *arg* (markup)
Set font size to -3.

```
\markup {
  default
  \hspace #2
  \teeny
  teeny
}
```

default teeny

`\text` *arg* (markup)
Use a text font instead of music symbol or music alphabet font.

```
\markup {
  \number {
    1, 2,
    \text {
      three, four,
    }
    5
  }
}
```

1, 2, three, four, **5**

`\tiny` *arg* (markup)
Set font size to -2.

```
\markup {
  default
  \hspace #2
  \tiny
  tiny
}
```

default tiny

`\typewriter` *arg* (markup)
Use `font-family` typewriter for *arg*.

```
\markup {
  default
  \hspace #2
  \typewriter
  typewriter
}
```

default `typewriter`

`\underline` *arg* (markup)
Underline *arg*. Looks at `thickness` to determine line thickness, and `offset` to determine line y-offset.

```
\markup \fill-line {
  \underline "underlined"
  \override #'(offset . 5)
  \override #'(thickness . 1)
```

```
  \underline "underlined"
  \override #'(offset . 1)
  \override #'(thickness . 5)
  \underline "underlined"
}
```

underlined underlined underlined

Used properties:

- `offset` (2)
- `thickness` (1)

`\upright` *arg* (markup)
Set `font-shape` to `upright`. This is the opposite of `italic`.

```
\markup {
  \italic {
    italic text
    \hspace #2
    \upright {
      upright text
    }
    \hspace #2
    italic again
  }
}
```

italic text upright text *italic again*

A.8.2 Align

`\center-align` *arg* (markup)
Align `arg` to its X center.

```
\markup {
  \column {
    one
    \center-align
    two
    three
  }
}
```

one
two
three

`\center-column` *args* (markup list)
Put `args` in a centered column.

```
\markup {
  \center-column {
    one
    two
    three
```

```
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\column` *args* (markup list)
: Stack the markups in *args* vertically. The property `baseline-skip` determines the space between markups in *args*.

```
\markup {
  \column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\combine` *arg1* (markup) *arg2* (markup)
: Print two markups on top of each other.

Note: `\combine` cannot take a list of markups enclosed in curly braces as an argument; the follow example will not compile:

```
\combine { a list }
```

```
\markup {
  \fontsize #5
  \override #'(thickness . 2)
  \combine
    \draw-line #'(0 . 4)
    \arrow-head #Y #DOWN ##f
}
```



`\concat` *args* (markup list)
: Concatenate *args* in a horizontal line, without spaces in between. Strings and simple markups are concatenated on the input level, allowing ligatures. For example, `\concat { "f" \simple #"i" }` is equivalent to `"fi"`.

```
\markup {
  \concat {
    one
    two
```

```
    three
  }
}
```

onetwothree

`\dir-column` *args* (markup list)
: Make a column of *args*, going up or down, depending on the setting of the `direction` layout property.

```
\markup {
  \override #`(direction . ,UP) {
    \dir-column {
      going up
    }
  }
  \hspace #1
  \dir-column {
    going down
  }
  \hspace #1
  \override #'(direction . 1) {
    \dir-column {
      going up
    }
  }
}
```

up going
going down
up going

Used properties:

- `baseline-skip`
- `direction`

`\fill-line` *args* (markup list)
: Put *markups* in a horizontal line of width *line-width*. The markups are spaced or flushed to fill the entire line. If there are no arguments, return an empty stencil.

```
\markup {
  \column {
    \fill-line {
      Words evenly spaced across the page
    }
    \null
    \fill-line {
      \line { Text markups }
      \line {
        \italic { evenly spaced }
      }
      \line { across the page }
    }
  }
```

```
}
```

Words evenly spaced across the page

Text markups *evenly spaced* across the page

Used properties:

- `line-width` (#f)
- `word-space` (0.6)
- `text-direction` (1)

`\fill-with-pattern` *space* (number) *dir* (direction) *pattern* (markup) *left* (markup) *right* (markup)

Put *left* and *right* in a horizontal line of width `line-width` with a line of markups *pattern* in between. Patterns are spaced apart by *space*. Patterns are aligned to the *dir* markup.

```
\markup \column {
  "right-aligned :"
  \fill-with-pattern #1 #RIGHT . first right
  \fill-with-pattern #1 #RIGHT . second right
  \null
  "center-aligned :"
  \fill-with-pattern #1.5 #CENTER - left right
  \null
  "left-aligned :"
  \override #'(line-width . 50) \fill-with-pattern #2 #LEFT : left first
  \override #'(line-width . 50) \fill-with-pattern #2 #LEFT : left second
}
```

right-aligned :
first . . . . . right
second . . . . . right

center-aligned :
left - right

left-aligned :
left : : : : : : : : : : : : : : : first
left : : : : : : : : : : : : : : : second

Used properties:

- `line-width`
- `word-space`

`\general-align` *axis* (integer) *dir* (number) *arg* (markup)

Align *arg* in *axis* direction to the *dir* side.

```
\markup {
  \column {
    one
    \general-align #X #LEFT
    two
```

```
    three
    \null
    one
    \general-align #X #CENTER
    two
    three
    \null
    \line {
      one
      \general-align #Y #UP
      two
      three
    }
    \null
    \line {
      one
      \general-align #Y #3.2
      two
      three
    }
  }
}
```

one
two
three

one
two
three

one two three

one three
two

`\halign` *dir* (number) *arg* (markup)
: Set horizontal alignment. If *dir* is `-1`, then it is left-aligned, while `+1` is right. Values in between interpolate alignment accordingly.

```
\markup {
  \column {
    one
    \halign #LEFT
    two
    three
    \null
    one
    \halign #CENTER
    two
    three
    \null
```

```
    one
    \halign #RIGHT
    two
    three
    \null
    one
    \halign #-5
    two
    three
  }
}
```

one
two
three
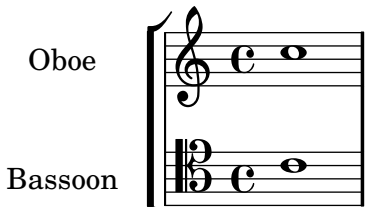
one
two
three

one
two
three

one
two
three

`\hcenter-in` *length* (number) *arg* (markup)
Center *arg* horizontally within a box of extending *length*/2 to the left and right.

```
\new StaffGroup <<
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Oboe
    }
    c''1
  }
  \new Staff {
    \set Staff.instrumentName = \markup {
      \hcenter-in #12
      Bassoon
    }
    \clef tenor
    c'1
  }
>>
```



`\hspace` *amount* (number)

Create an invisible object taking up horizontal space *amount*.

```
\markup {
  one
  \hspace #2
  two
  \hspace #8
  three
}
```

one two three

Used properties:

- `word-space`

`\justify-field` *symbol* (symbol)

Justify the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur adipisicing
    elit, sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua.  Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \justify-field #'header:myText
    }
  }
}

\markup {
  \null
}
```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\justify` *args* (markup list)

Like `\wordwrap`, but with lines stretched to justify the margins. Use `\override #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

```
\markup {
  \justify {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\justify-string` *arg* (string)

Justify a string. Paragraphs may be separated with double newlines

```
\markup {
  \override #'(line-width . 40)
  \justify-string #"Lorem ipsum dolor sit amet, consectetur
      adipisicing elit, sed do eiusmod tempor incididunt ut labore
      et dolore magna aliqua.


      Ut enim ad minim veniam, quis nostrud exercitation ullamco
      laboris nisi ut aliquip ex ea commodo consequat.


      Excepteur sint occaecat cupidatat non proident, sunt in culpa
      qui officia deserunt mollit anim id est laborum"
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.
Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.
Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

`\left-align` *arg* (markup)

Align *arg* on its left edge.

```
\markup {
  \column {
    one
    \left-align
    two
    three
  }
}
```

one
two
three

`\left-column` *args* (markup list)

Put `args` in a left-aligned column.

```
\markup {
  \left-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\line` *args* (markup list)

Put *args* in a horizontal line. The property `word-space` determines the space between markups in *args*.

```
\markup {
  \line {
    one two three
  }
}
```

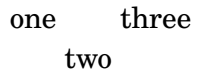
one two three

Used properties:

- `text-direction` (1)
- `word-space`

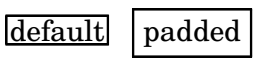
`\lower` *amount* (number) *arg* (markup)
: Lower *arg* by the distance *amount*. A negative *amount* indicates raising; see also `\raise`.

```
\markup {
  one
  \lower #3
  two
  three
}
```



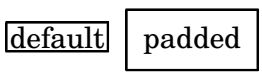
`\pad-around` *amount* (number) *arg* (markup)
: Add padding *amount* all around *arg*.

```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-around #0.5 {
      padded
    }
  }
}
```



`\pad-markup` *amount* (number) *arg* (markup)
: Add space around a markup object.

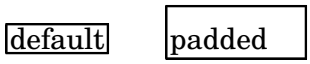
```
\markup {
  \box {
    default
  }
  \hspace #2
  \box {
    \pad-markup #1 {
      padded
    }
  }
}
```



`\pad-to-box` *x-ext* (pair of numbers) *y-ext* (pair of numbers) *arg* (markup)
: Make *arg* take at least *x-ext*, *y-ext* space.

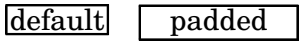
```
\markup {
  \box {
    default
```

```
  }
  \hspace #4
  \box {
    \pad-to-box #'(0 . 10) #'(0 . 3) {
      padded
    }
  }
}
```



`\pad-x` *amount* (number) *arg* (markup)
: Add padding *amount* around *arg* in the X direction.

```
\markup {
  \box {
    default
  }
  \hspace #4
  \box {
    \pad-x #2 {
      padded
    }
  }
}
```



`\put-adjacent` *axis* (integer) *dir* (direction) *arg1* (markup) *arg2* (markup)
: Put *arg2* next to *arg1*, without moving *arg1*.

`\raise` *amount* (number) *arg* (markup)
: Raise *arg* by the distance *amount*. A negative *amount* indicates lowering, see also `\lower`.

  The argument to `\raise` is the vertical displacement amount, measured in (global) staff spaces. `\raise` and `\super` raise objects in relation to their surrounding markups.

  If the text object itself is positioned above or below the staff, then `\raise` cannot be used to move it, since the mechanism that positions it next to the staff cancels any shift made with `\raise`. For vertical positioning, use the `padding` and/or `extra-offset` properties.

```
\markup {
  C
  \small
  \bold
  \raise #1.0
  9/7+
}
```

C **9/7+**

`\right-align` *arg* (markup)
: Align *arg* on its right edge.

```
\markup {
  \column {
    one
    \right-align
    two
    three
  }
}
```

one
two
three

`\right-column` *args* (markup list)
: Put `args` in a right-aligned column.

```
\markup {
  \right-column {
    one
    two
    three
  }
}
```

one
two
three

Used properties:

- `baseline-skip`

`\rotate` *ang* (number) *arg* (markup)
: Rotate object with *ang* degrees around its center.

```
\markup {
  default
  \hspace #2
  \rotate #45
  \line {
    rotated 45˚
  }
}
```

default rotated 45°

`\translate` *offset* (pair of numbers) *arg* (markup)
: Translate *arg* relative to its surroundings. *offset* is a pair of numbers representing the displacement in the X and Y axis.

```
\markup {
  *
  \translate #'(2 . 3)
  \line { translated two spaces right, three up }
```

```
}
```



`\translate-scaled` *offset* (pair of numbers) *arg* (markup)
Translate *arg* by *offset*, scaling the offset by the `font-size`.

```
\markup {
  \fontsize #5 {
    * \translate #'(2 . 3) translate
    \hspace #2
    * \translate-scaled #'(2 . 3) translate-scaled
  }
}
```

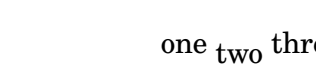


Used properties:

- `font-size` (0)

`\vcenter` *arg* (markup)
Align `arg` to its Y center.

```
\markup {
  one
  \vcenter
  two
  three
}
```



`\vspace` *amount* (number)
Create an invisible object taking up vertical space of *amount* multiplied by 3.

```
\markup {
    \center-column {
    one
    \vspace #2
    two
    \vspace #5
    three
  }
}
```

one

two

three

`\wordwrap-field` *symbol* (symbol)
Wordwrap the data which has been assigned to *symbol*.

```
\header {
  title = "My title"
  myText = "Lorem ipsum dolor sit amet, consectetur adipisicing
    elit, sed do eiusmod tempor incididunt ut labore et dolore magna
    aliqua.  Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat."
}

\paper {
  bookTitleMarkup = \markup {
    \column {
      \fill-line { \fromproperty #'header:title }
      \null
      \wordwrap-field #'header:myText
    }
  }
}

\markup {
  \null
}
```

My title

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

`\wordwrap` *args* (markup list)
Simple wordwrap. Use `\override #'(line-width . X)` to set the line width, where X is the number of staff spaces.

```
\markup {
  \wordwrap {
    Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed
    do eiusmod tempor incididunt ut labore et dolore magna aliqua.
    Ut enim ad minim veniam, quis nostrud exercitation ullamco
    laboris nisi ut aliquip ex ea commodo consequat.
```

```
  }
}
```

Lorem ipsum dolor sit amet, consectetur adipisicing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-string` *arg* (string)

Wordwrap a string. Paragraphs may be separated with double newlines.

```
\markup {
  \override #'(line-width . 40)
  \wordwrap-string #"Lorem ipsum dolor sit amet, consectetur
      adipisicing elit, sed do eiusmod tempor incididunt ut labore
      et dolore magna aliqua.


      Ut enim ad minim veniam, quis nostrud exercitation ullamco
      laboris nisi ut aliquip ex ea commodo consequat.


      Excepteur sint occaecat cupidatat non proident, sunt in culpa
      qui officia deserunt mollit anim id est laborum"
}
```

Lorem ipsum dolor sit amet,
consectetur adipisicing elit, sed do
eiusmod tempor incididunt ut labore
et dolore magna aliqua.
Ut enim ad minim veniam, quis
nostrud exercitation ullamco laboris
nisi ut aliquip ex ea commodo
consequat.
Excepteur sint occaecat cupidatat non
proident, sunt in culpa qui officia
deserunt mollit anim id est laborum

Used properties:

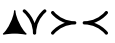
- `text-direction` (1)
- `word-space`
- `line-width`
- `baseline-skip`

A.8.3 Graphic

`\arrow-head` *axis* (integer) *dir* (direction) *filled* (boolean)

Produce an arrow head in specified direction and axis. Use the filled head if *filled* is specified.

```
\markup {
  \fontsize #5 {
    \general-align #Y #DOWN {
      \arrow-head #Y #UP ##t
      \arrow-head #Y #DOWN ##f
      \hspace #2
      \arrow-head #X #RIGHT ##f
      \arrow-head #X #LEFT ##f
    }
  }
}
```



`\beam` *width* (number) *slope* (number) *thickness* (number)

Create a beam with the specified parameters.

```
\markup {
  \beam #5 #1 #2
}
```



`\bracket` *arg* (markup)

Draw vertical brackets around *arg*.

```
\markup {
  \bracket {
    \note #"2." #UP
  }
}
```



`\circle` *arg* (markup)

Draw a circle around *arg*. Use `thickness`, `circle-padding` and `font-size` properties to determine line thickness and padding around the markup.

```
\markup {
  \circle {
    Hi
  }
}
```



Used properties:

- `circle-padding` (0.2)
- `font-size` (0)
- `thickness` (1)

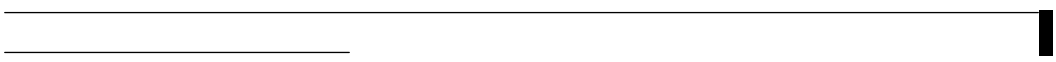
`\draw-circle` *radius* (number) *thickness* (number) *filled* (boolean)
A circle of radius *radius* and thickness *thickness*, optionally filled.

```
\markup {
  \draw-circle #2 #0.5 ##f
  \hspace #2
  \draw-circle #2 #0 ##t
}
```



`\draw-hline`
Draws a line across a page, where the property `span-factor` controls what fraction of the page is taken up.

```
\markup {
  \column {
    \draw-hline
    \override #'(span-factor . 1/3)
    \draw-hline
  }
}
```



Used properties:

- `span-factor` (1)
- `line-width`
- `draw-line-markup`

`\draw-line` *dest* (pair of numbers)
A simple line.

```
\markup {
  \draw-line #'(4 . 4)
  \override #'(thickness . 5)
  \draw-line #'(-3 . 0)
}
```



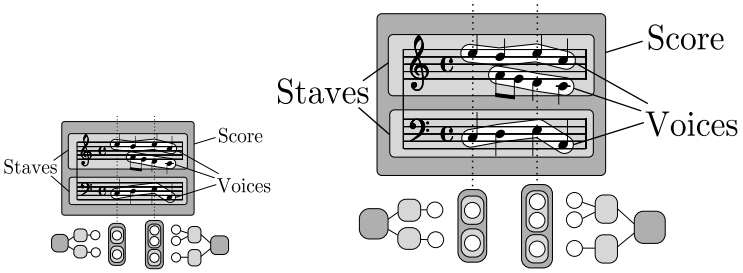
Used properties:

- `thickness` (1)

`\epsfile` *axis* (number) *size* (number) *file-name* (string)
Inline an EPS image. The image is scaled along *axis* to *size*.

```
\markup {
  \general-align #Y #DOWN {
    \epsfile #X #20 #"context-example.eps"
```

```
    \epsfile #Y #20 #"context-example.eps"
  }
}
```



`\filled-box` *xext* (pair of numbers) *yext* (pair of numbers) *blot* (number)

Draw a box with rounded corners of dimensions *xext* and *yext*. For example,

```
\filled-box #'(-.3 . 1.8) #'(-.3 . 1.8) #0
```

creates a box extending horizontally from -0.3 to 1.8 and vertically from -0.3 up to 1.8, with corners formed from a circle of diameter 0 (i.e., sharp corners).

```
\markup {
  \filled-box #'(0 . 4) #'(0 . 4) #0
  \filled-box #'(0 . 2) #'(-4 . 2) #0.4
  \filled-box #'(1 . 8) #'(0 . 7) #0.2
  \with-color #white
  \filled-box #'(-4.5 . -2.5) #'(3.5 . 5.5) #0.7
}
```



`\hbracket` *arg* (markup)

Draw horizontal brackets around *arg*.

```
\markup {
  \hbracket {
    \line {
      one two three
    }
  }
}
```

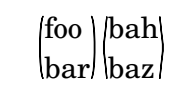
one two three

`\parenthesize` *arg* (markup)

Draw parentheses around *arg*. This is useful for parenthesizing a column containing several lines of text.

```
\markup {
  \line {
    \parenthesize {
      \column {
        foo
```

```
        bar
      }
    }
    \override #'(angularity . 2) {
      \parenthesize {
        \column {
          bah
          baz
        }
      }
    }
  }
}
```



Used properties:

- `width` (0.25)
- `thickness` (1)
- `size` (1)
- `padding`
- `angularity` (0)

`\path` *thickness* (number) *commands* (list)

Draws a path with line thickness *thickness* according to the directions given in *commands*. *commands* is a list of lists where the `car` of each sublist is a drawing command and the `cdr` comprises the associated arguments for each command.

Line-cap styles and line-join styles may be customized by overriding the `line-cap-style` and `line-join-style` properties, respectively. Available line-cap styles are `'butt`, `'round`, and `'square`. Available line-join styles are `'miter`, `'round`, and `'bevel`.

The property `filled` specifies whether or not the path is filled with color.

There are seven commands available to use in the list `commands`: `moveto`, `rmoveto`, `lineto`, `rlineto`, `curveto`, `rcurveto`, and `closepath`. Note that the commands that begin with *r* are the relative variants of the other three commands.

The commands `moveto`, `rmoveto`, `lineto`, and `rlineto` take 2 arguments; they are the X and Y coordinates for the destination point.

The commands `curveto` and `rcurveto` create cubic Bézier curves, and take 6 arguments; the first two are the X and Y coordinates for the first control point, the second two are the X and Y coordinates for the second control point, and the last two are the X and Y coordinates for the destination point.

The `closepath` command takes zero arguments and closes the current subpath in the active path.

Note that a sequence of commands *must* begin with a `moveto` or `rmoveto` to work with the SVG output.

```
samplePath =
  #'((moveto 0 0)
     (lineto -1 1)
     (lineto 1 1)
```

```
      (lineto 1 -1)
      (curveto -5 -5 -5 5 -1 0)
      (closepath))

\markup {
  \path #0.25 #samplePath
}
```



Used properties:

- `filled` (#f)
- `line-join-style` (round)
- `line-cap-style` (round)
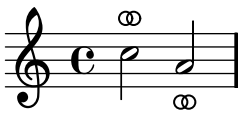
`\postscript` *str* (string)

This inserts *str* directly into the output as a PostScript command string.

```
ringsps = #"
  0.15 setlinewidth
  0.9 0.6 moveto
  0.4 0.6 0.5 0 361 arc
  stroke
  1.0 0.6 0.5 0 361 arc
  stroke
  "

rings = \markup {
  \with-dimensions #'(-0.2 . 1.6) #'(0 . 1.2)
  \postscript #ringsps
}
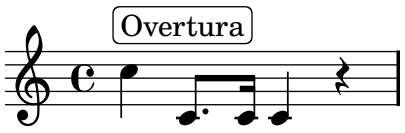
\relative c'' {
  c2^\rings
  a2_\rings
}
```



`\rounded-box` *arg* (markup)

Draw a box with rounded corners around *arg*. Looks at `thickness`, `box-padding` and `font-size` properties to determine line thickness and padding around the markup; the `corner-radius` property makes it possible to define another shape for the corners (default is 1).

```
c4^\markup {
  \rounded-box {
    Overtura
  }
}
c,8. c16 c4 r
```



Used properties:

- `box-padding` (0.5)
- `font-size` (0)
- `corner-radius` (1)
- `thickness` (1)

`\scale` *factor-pair* (pair of numbers) *arg* (markup)
Scale *arg*. *factor-pair* is a pair of numbers representing the scaling-factor in the X and Y axes. Negative values may be used to produce mirror images.

```
\markup {
  \line {
    \scale #'(2 . 1)
    stretched
    \scale #'(1 . -1)
    mirrored
  }
}
```

stretched **mirrored**

`\triangle` *filled* (boolean)
A triangle, either filled or empty.

```
\markup {
  \triangle ##t
  \hspace #2
  \triangle ##f
}
```

▲ △

Used properties:

- `baseline-skip` (2)
- `font-size` (0)
- `thickness` (0.1)

`\with-url` *url* (string) *arg* (markup)
Add a link to URL *url* around *arg*. This only works in the PDF backend.

```
\markup {
  \with-url #"http://lilypond.org/web/" {
    LilyPond ... \italic {
      music notation for everyone
    }
  }
}
```

LilyPond ... *music notation for everyone*

A.8.4 Music

`\customTabClef` *num-strings* (integer) *staff-space* (number)
Draw a tab clef sans-serif style.

`\doubleflat`
Draw a double flat symbol.

```
\markup {
  \doubleflat
}
```



`\doublesharp`
Draw a double sharp symbol.

```
\markup {
  \doublesharp
}
```



`\flat`
Draw a flat symbol.

```
\markup {
  \flat
}
```



`\musicglyph` *glyph-name* (string)
glyph-name is converted to a musical symbol; for example, `\musicglyph #"accidentals.natural"` selects the natural sign from the music font. See Section "The Feta font" dans *Manuel de notation* for a complete listing of the possible glyphs.

```
\markup {
  \musicglyph #"f"
  \musicglyph #"rests.2"
  \musicglyph #"clefs.G_change"
}
```



`\natural`
Draw a natural symbol.

```
\markup {
  \natural
}
```



`\note-by-number` *log* (number) *dot-count* (number) *dir* (number)
Construct a note symbol, with stem. By using fractional values for *dir*, longer or shorter stems can be obtained.

```
\markup {
  \note-by-number #3 #0 #DOWN
  \hspace #2
  \note-by-number #1 #2 #0.8
}
```



Used properties:

- `style` ('())
- `font-size` (0)

`\note` *duration* (string) *dir* (number)

This produces a note with a stem pointing in *dir* direction, with the *duration* for the note head type and augmentation dots. For example, `\note #"4." #-0.75` creates a dotted quarter note, with a shortened down stem.

```
\markup {
  \override #'(style . cross) {
    \note #"4.." #UP
  }
  \hspace #2
  \note #"breve" #0
}
```



Used properties:

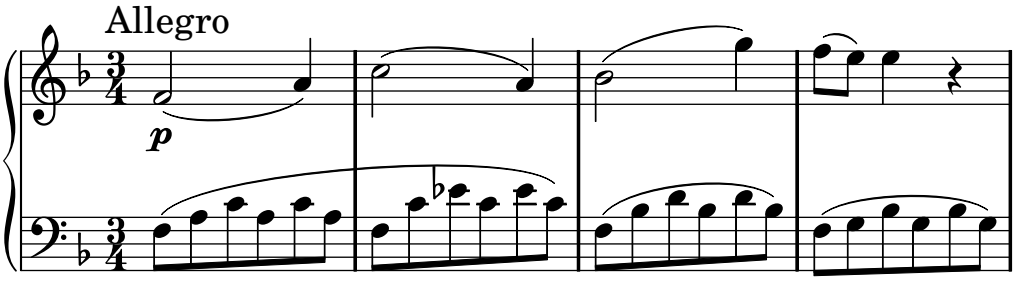
- `style` ('())
- `font-size` (0)

`\score` *score* (score)

Inline an image of music.

```
\markup {
  \score {
    \new PianoStaff <<
      \new Staff \relative c' {
        \key f \major
        \time 3/4
        \mark \markup { Allegro }
        f2\p( a4)
        c2( a4)
        bes2( g'4)
        f8( e) e4 r
      }
      \new Staff \relative c {
        \clef bass
        \key f \major
        \time 3/4
        f8( a c a c a
        f c' es c es c)
```

```
          f,( bes d bes d bes)
          f( g bes g bes g)
        }
      >>
      \layout {
        indent = 0.0\cm
        \context {
          \Score
          \override RehearsalMark #'break-align-symbols =
            #'(time-signature key-signature)
          \override RehearsalMark #'self-alignment-X = #LEFT
        }
        \context {
          \Staff
          \override TimeSignature #'break-align-anchor-alignment = #LEFT
        }
      }
    }
  }
```



Used properties:

- `baseline-skip`

`\semiflat`

Draw a semiflat symbol.

```
\markup {
  \semiflat
}
```



`\semisharp`

Draw a semisharp symbol.

```
\markup {
  \semisharp
}
```



`\sesquiflat`

Draw a 3/2 flat symbol.

```
\markup {
  \sesquiflat
```

```
}
```



`\sesquisharp`
: Draw a 3/2 sharp symbol.

```
\markup {
  \sesquisharp
}
```



`\sharp`
: Draw a sharp symbol.

```
\markup {
  \sharp
}
```



`\tied-lyric` *str* (string)
: Like simple-markup, but use tie characters for '~' tilde symbols.

```
\markup {
  \tied-lyric #"Lasciate~i monti"
}
```

Lasciate‿i monti

A.8.5 Instrument Specific Markup

`\fret-diagram` *definition-string* (string)
: Make a (guitar) fret diagram. For example, say

`\markup \fret-diagram #"s:0.75;6-x;5-x;4-o;3-2;2-3;1-2;"`

for fret spacing 3/4 of staff space, D chord diagram

Syntax rules for *definition-string*:

- Diagram items are separated by semicolons.
- Possible items:
  - `s:`*number* – Set the fret spacing of the diagram (in staff spaces). Default: 1.
  - `t:`*number* – Set the line thickness (in staff spaces). Default: 0.05.
  - `h:`*number* – Set the height of the diagram in frets. Default: 4.
  - `w:`*number* – Set the width of the diagram in strings. Default: 6.
  - `f:`*number* – Set fingering label type (0 = none, 1 = in circle on string, 2 = below string). Default: 0.
  - `d:`*number* – Set radius of dot, in terms of fret spacing. Default: 0.25.
  - `p:`*number* – Set the position of the dot in the fret space. 0.5 is centered; 1 is on lower fret bar, 0 is on upper fret bar. Default: 0.6.
  - `c:`*string1-string2-fret* – Include a barre mark from *string1* to *string2* on *fret*.

- *string*`-`*fret* – Place a dot on *string* at *fret*. If *fret* is '`o`', *string* is identified as open. If *fret* is '`x`', *string* is identified as muted.
- *string*`-`*fret*`-`*fingering* – Place a dot on *string* at *fret*, and label with *fingering* as defined by the `f:` code.

– Note: There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-terse` *definition-string* (string)

Make a fret diagram markup using terse string-based syntax.

Here is an example

```
\markup \fret-diagram-terse #"x;x;o;2;3;2;"
```

for a D chord diagram.

Syntax rules for *definition-string*:

- Strings are terminated by semicolons; the number of semicolons is the number of strings in the diagram.
- Mute strings are indicated by '`x`'.
- Open strings are indicated by '`o`'.
- A number indicates a fret indication at that fret.
- If there are multiple fret indicators desired on a string, they should be separated by spaces.
- Fingerings are given by following the fret number with a `-`, followed by the finger indicator, e.g. '`3-2`' for playing the third fret with the second finger.
- Where a barre indicator is desired, follow the fret (or fingering) symbol with `-(` to start a barre and `-)` to end the barre.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\fret-diagram-verbose` *marking-list* (pair)

Make a fret diagram containing the symbols indicated in *marking-list*.

For example,

```
\markup \fret-diagram-verbose
  #'((mute 6) (mute 5) (open 4)
     (place-fret 3 2) (place-fret 2 3) (place-fret 1 2))
```

produces a standard D chord diagram without fingering indications.

Possible elements in *marking-list*:

`(mute` *string-number*`)`

Place a small 'x' at the top of string *string-number*.

`(open` *string-number*`)`

Place a small 'o' at the top of string *string-number*.

`(barre` *start-string end-string fret-number*`)`
: Place a barre indicator (much like a tie) from string *start-string* to string *end-string* at fret *fret-number*.

`(capo` *fret-number*`)`
: Place a capo indicator (a large solid bar) across the entire fretboard at fret location *fret-number*. Also, set fret *fret-number* to be the lowest fret on the fret diagram.

`(place-fret` *string-number fret-number* `[`*finger-value* `[`*color-modifier*`]])`
: Place a fret playing indication on string *string-number* at fret *fret-number* with an optional fingering label *finger-value*, and an optional color modifier *color-modifier*. By default, the fret playing indicator is a solid dot. This can be globally changed by setting the value of the variable *dot-color*. Setting *color-modifier* to `inverted` inverts the dot color for a specific fingering. If the *finger* part of the `place-fret` element is present, *finger-value* will be displayed according to the setting of the variable *finger-code*. There is no limit to the number of fret indications per string.

Used properties:

- `thickness` (0.5)
- `fret-diagram-details`
- `size` (1.0)
- `align-dir` (-0.4)

`\harp-pedal` *definition-string* (string)
: Make a harp pedal diagram.

  Possible elements in *definition-string*:

`^`	pedal is up
`-`	pedal is neutral
`v`	pedal is down
`|`	vertical divider line
`o`	the following pedal should be circled (indicating a change)

  The function also checks if the string has the typical form of three pedals, then the divider and then the remaining four pedals. If not it prints out a warning. However, in any case, it will also print each symbol in the order as given. This means you can place the divider (even multiple dividers) anywhere you want, but you'll have to live with the warnings.

  The appearance of the diagram can be tweaked inter alia using the size property of the TextScript grob (`\override Voice.TextScript #'size = #0.3`) for the overall, the thickness property (`\override Voice.TextScript #'thickness = #3`) for the line thickness of the horizontal line and the divider. The remaining configuration (box sizes, offsets and spaces) is done by the harp-pedal-details list of properties (`\override Voice.TextScript #'harp-pedal-details #'box-width = #1`). It contains the following settings: `box-offset` (vertical shift of the box center for up/down pedals), `box-width`, `box-height`, `space-before-divider` (the spacing between two boxes before the divider) and `space-after-divider` (box spacing after the divider).

```
\markup \harp-pedal #"^-v|--ov^"
```



Used properties:

- `thickness` (0.5)
- `harp-pedal-details` ('())
- `size` (1.2)

`\woodwind-diagram` *instrument* (symbol) *user-draw-commands* (list)

Make a woodwind-instrument diagram. For example, say

```
\markup \woodwind-diagram #'oboe #'((lh . (d ees)) (cc . (five3qT1q)) (rh . (gis))
```

for an oboe with the left-hand d key, left-hand ees key, and right-hand gis key depressed while the five-hole of the central column effectuates a trill between 1/4 and 3/4 closed.

The following instruments are supported:

- piccolo
- flute
- oboe
- clarinet
- bass-clarinet
- saxophone
- bassoon
- contrabassoon

To see all of the callable keys for a given instrument, include the function `(print-keys 'instrument)` in your .ly file, where instrument is the instrument whose keys you want to print.

Certain keys allow for special configurations. The entire gamut of configurations possible is as follows:

- 1q (1/4 covered)
- 1h (1/2 covered)
- 3q (3/4 covered)
- R (ring depressed)
- F (fully covered; the default if no state put)

Additionally, these configurations can be used in trills. So, for example, `three3qTR` effectuates a trill between 3/4 full and ring depressed on the three hole. As another example, `threeRT` effectuates a trill between R and open, whereas `threeTR` effectuates a trill between open and shut. To see all of the possibilities for all of the keys of a given instrument, invoke `(print-keys-verbose 'instrument)`.

Lastly, substituting an empty list for the pressed-key alist will result in a diagram with all of the keys drawn but none filled, for example:

```
\markup \woodwind-diagram #'oboe #'()
```

Used properties:

- `graphical` (#t)
- `thickness` (0.1)
- `size` (1)

A.8.6 Other

`\backslashed-digit` *num* (integer)
A feta number, with backslash. This is for use in the context of figured bass notation.

```
\markup {
  \backslashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \backslashed-digit #7
}
```



Used properties:

- `thickness` (1.6)
- `font-size` (0)

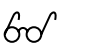
`\char` *num* (integer)
Produce a single character. Characters encoded in hexadecimal format require the prefix `#x`.

```
\markup {
  \char #65 \char ##x00a9
}
```



`\eyeglasses`
Prints out eyeglasses, indicating strongly to look at the conductor.

```
\markup { \eyeglasses }
```

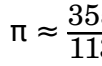


`\footnote` *mkup* (markup) *note* (markup)
Have footnote *note* act as an annotation to the markup *mkup*.

`\fraction` *arg1* (markup) *arg2* (markup)
Make a fraction of two markups.

```
\markup {

  \fraction 355 113
}
```



Used properties:

- `font-size` (0)

`\fromproperty` *symbol* (symbol)
Read the *symbol* from property settings, and produce a stencil from the markup contained within. If *symbol* is not defined, it returns an empty markup.

```
\header {
  myTitle = "myTitle"
  title = \markup {
    from
    \italic
    \fromproperty #'header:myTitle
  }
}
\markup {
  \null
}
```



`\left-brace` *size* (number)

A feta brace in point size *size*.

```
\markup {
  \left-brace #35
  \hspace #2
  \left-brace #45
}
```



`\lookup` *glyph-name* (string)

Lookup a glyph by name.

```
\markup {
  \override #'(font-encoding . fetaBraces) {
    \lookup #"brace200"
    \hspace #2
    \rotate #180
    \lookup #"brace180"
  }
}
```



`\markalphabet` *num* (integer)

Make a markup letter for *num*. The letters start with A to Z and continue with double letters.

```
\markup {
  \markalphabet #8
  \hspace #2
  \markalphabet #26
}
```

H AA

`\markletter` *num* (integer)
: Make a markup letter for *num*. The letters start with A to Z (skipping letter I), and continue with double letters.

```
\markup {
  \markletter #8
  \hspace #2
  \markletter #26
}
```

J AB

`\null`
: An empty markup with extents of a single point.

```
\markup {
  \null
}
```

`\on-the-fly` *procedure* (symbol) *arg* (markup)
: Apply the *procedure* markup command to *arg*. *procedure* should take a single argument.

`\override` *new-prop* (pair) *arg* (markup)
: Add the argument *new-prop* to the property list. Properties may be any property supported by Section “font-interface” dans *Référence des propriétés internes*, Section “text-interface” dans *Référence des propriétés internes* and Section “instrument-specific-markup-interface” dans *Référence des propriétés internes*.

```
\markup {
  \line {
    \column {
      default
      baseline-skip
    }
    \hspace #2
    \override #'(baseline-skip . 4) {
      \column {
        increased
        baseline-skip
      }
    }
  }
}
```

default baseline-skip increased baseline-skip

`\page-link` *page-number* (number) *arg* (markup)
Add a link to the page *page-number* around *arg*. This only works in the PDF backend.

```
\markup {
  \page-link #2  { \italic { This links to page 2... } }
}
```

This links to page 2...

`\page-ref` *label* (symbol) *gauge* (markup) *default* (markup)
Reference to a page number. *label* is the label set on the referenced page (using the `\label` command), *gauge* a markup used to estimate the maximum width of the page number, and *default* the value to display when *label* is not found.

`\pattern` *count* (integer) *axis* (integer) *space* (number) *pattern* (markup)
Prints *count* times a *pattern* markup. Patterns are spaced apart by *space*. Patterns are distributed on *axis*.

```
\markup \column {
  "Horizontally repeated :"
  \pattern #7 #X #2 \flat
  \null
  "Vertically repeated :"
  \pattern #3 #Y #0.5 \flat
}
```

Horizontally repeated :
♭ ♭ ♭ ♭ ♭ ♭ ♭

Vertically repeated :
♭
♭
♭

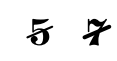
`\right-brace` *size* (number)
A feta brace in point size *size*, rotated 180 degrees.

```
\markup {
  \right-brace #45
  \hspace #2
  \right-brace #35
}
```



`\slashed-digit` *num* (integer)
A feta number, with slash. This is for use in the context of figured bass notation.

```
\markup {
  \slashed-digit #5
  \hspace #2
  \override #'(thickness . 3)
  \slashed-digit #7
}
```



Used properties:

- `thickness` (1.6)
- `font-size` (0)

`\stencil` *stil* (stencil)
: Use a stencil as markup.

```
\markup {
  \stencil #(make-circle-stencil 2 0 #t)
}
```



`\strut`
: Create a box of the same height as the space in the current font.

`\transparent` *arg* (markup)
: Make *arg* transparent.

```
\markup {
  \transparent {
    invisible text
  }
}
```

`\verbatim-file` *name* (string)
: Read the contents of file *name*, and include it verbatim.

```
\markup {
  \verbatim-file #"simple.ly"
}
```

```
%% A simple piece in LilyPond, a scale.
\relative c' {
  c d e f g a b c
}
%% Optional helper for automatic updating by convert-ly.  May be omit
\version "2.14.0"
```

`\whiteout` *arg* (markup)
: Provide a white background for *arg*.

```
\markup {
  \combine
    \filled-box #'(-1 . 10) #'(-3 . 4) #1
    \whiteout whiteout
}
```



`\with-color` *color* (color) *arg* (markup)
Draw *arg* in color specified by *color*.

```
\markup {
  \with-color #red
  red
  \hspace #2
  \with-color #green
  green
  \hspace #2
  \with-color #blue
  blue
}
```

red green blue

`\with-dimensions` *x* (pair of numbers) *y* (pair of numbers) *arg* (markup)
Set the dimensions of *arg* to *x* and *y*.

`\with-link` *label* (symbol) *arg* (markup)
Add a link to the page holding label *label* around *arg*. This only works in the PDF backend.

```
\markup {
  \with-link #"label" { \italic { This links to the page containing the label... }
}
```

This links to the page containing the label...

A.9 Text markup list commands

Les commandes suivantes peuvent être utilisées dans un bloc `\markuplines { }`.

`\column-lines` *args* (markup list)
Like `\column`, but return a list of lines instead of a single markup. `baseline-skip` determines the space between each markup in *args*.

Used properties:

- `baseline-skip`

`\justified-lines` *args* (markup list)
Like `\justify`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width; *X* is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`

- `line-width` (#f)
- `baseline-skip`

`\override-lines` *new-prop* (pair) *args* (markup list)
Like `\override`, for markup lists.

`\table-of-contents`
`\wordwrap-internal` *justify* (boolean) *args* (markup list)
Internal markup list command used to define `\justify` and `\wordwrap`.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)

`\wordwrap-lines` *args* (markup list)
Like `\wordwrap`, but return a list of lines instead of a single markup. Use `\override-lines #'(line-width . X)` to set the line width, where X is the number of staff spaces.

Used properties:

- `text-direction` (1)
- `word-space`
- `line-width` (#f)
- `baseline-skip`

`\wordwrap-string-internal` *justify* (boolean) *arg* (string)
Internal markup list command used to define `\justify-string` and `\wordwrap-string`.

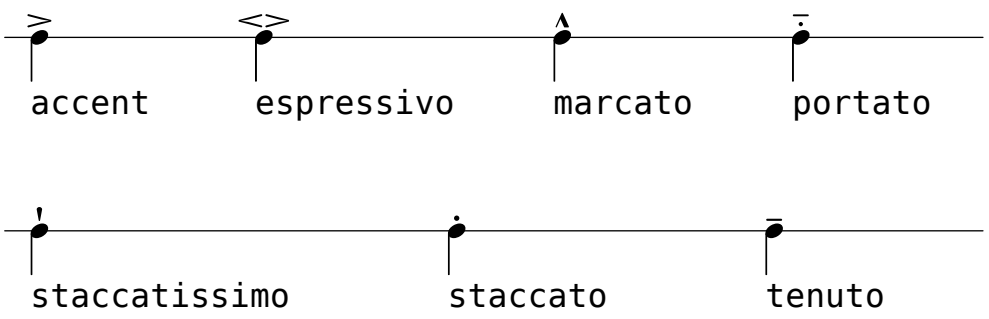
Used properties:

- `text-direction` (1)
- `word-space`
- `line-width`

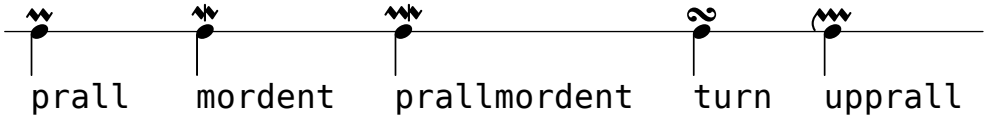
A.10 Liste des signes d'articulation

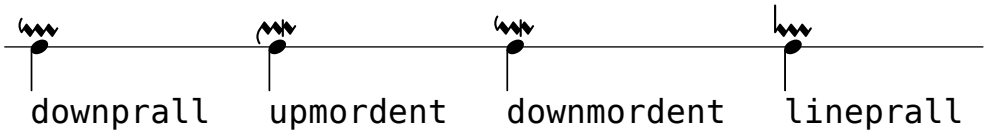
Voici la liste exhaustive des symboles prédéfinis et que vous pouvez attacher à une note (p.ex. `c\accent`) :

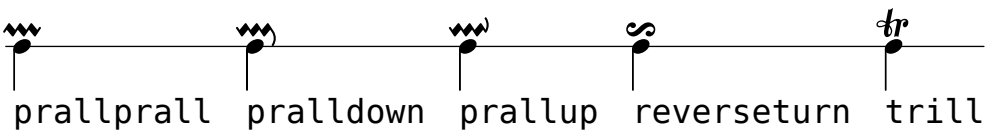
Scripts d'articulation



Scripts d'ornement

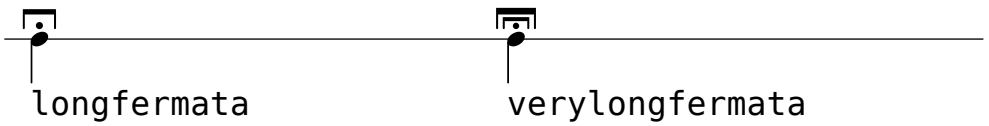






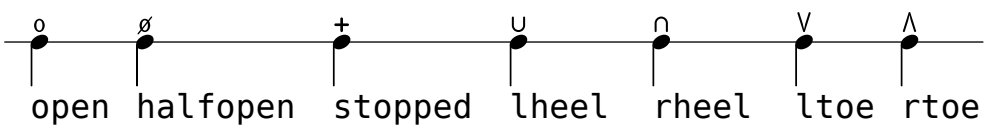
Scripts de point d'orgue et point d'arrêt



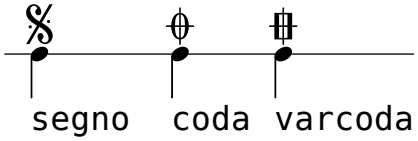


Scripts spécifiques à certains instruments

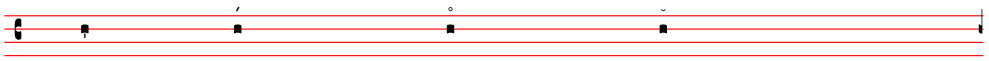


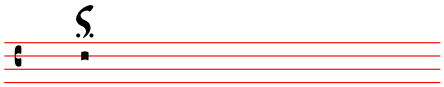


Scripts de reprise et de répétition

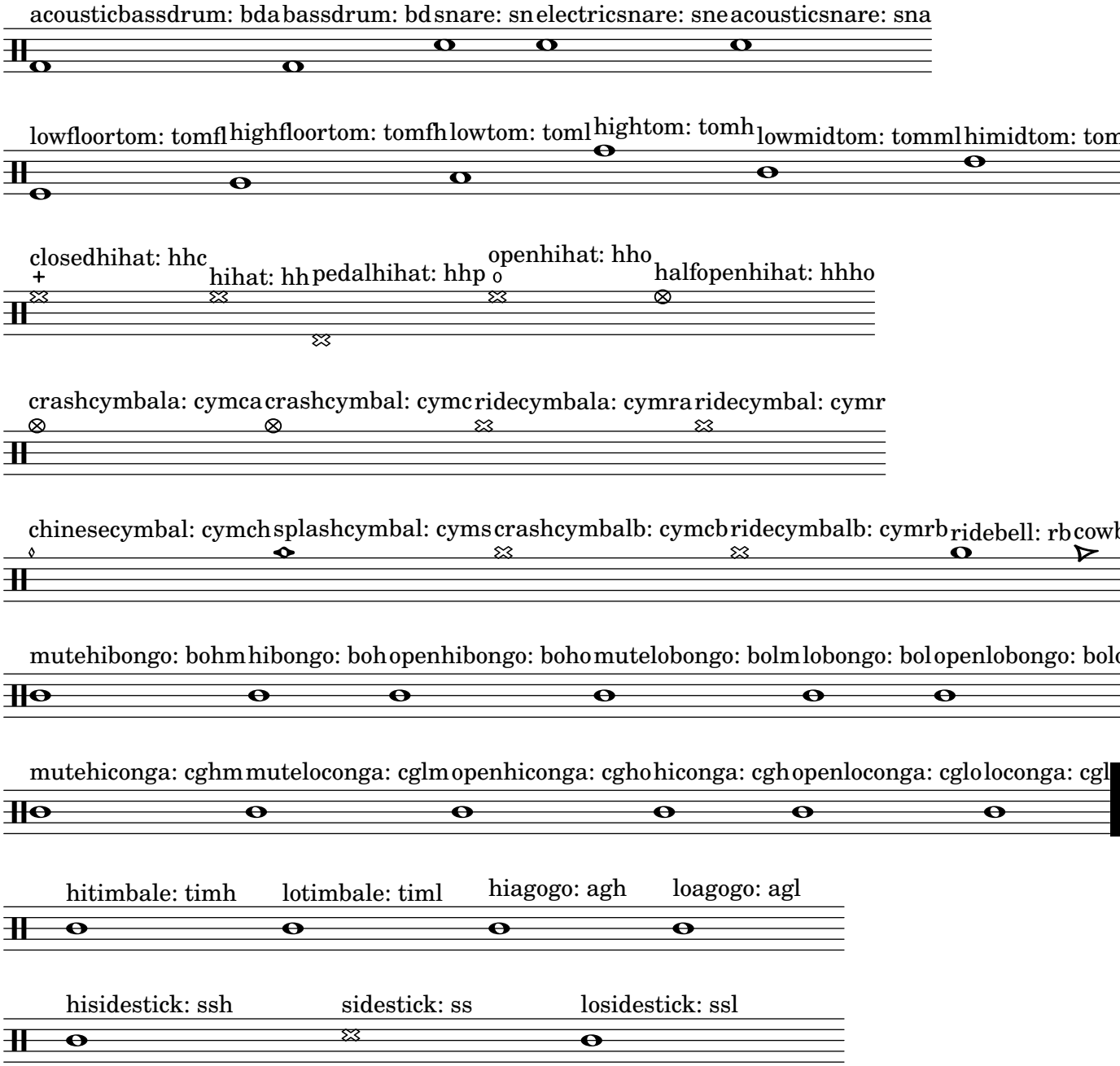


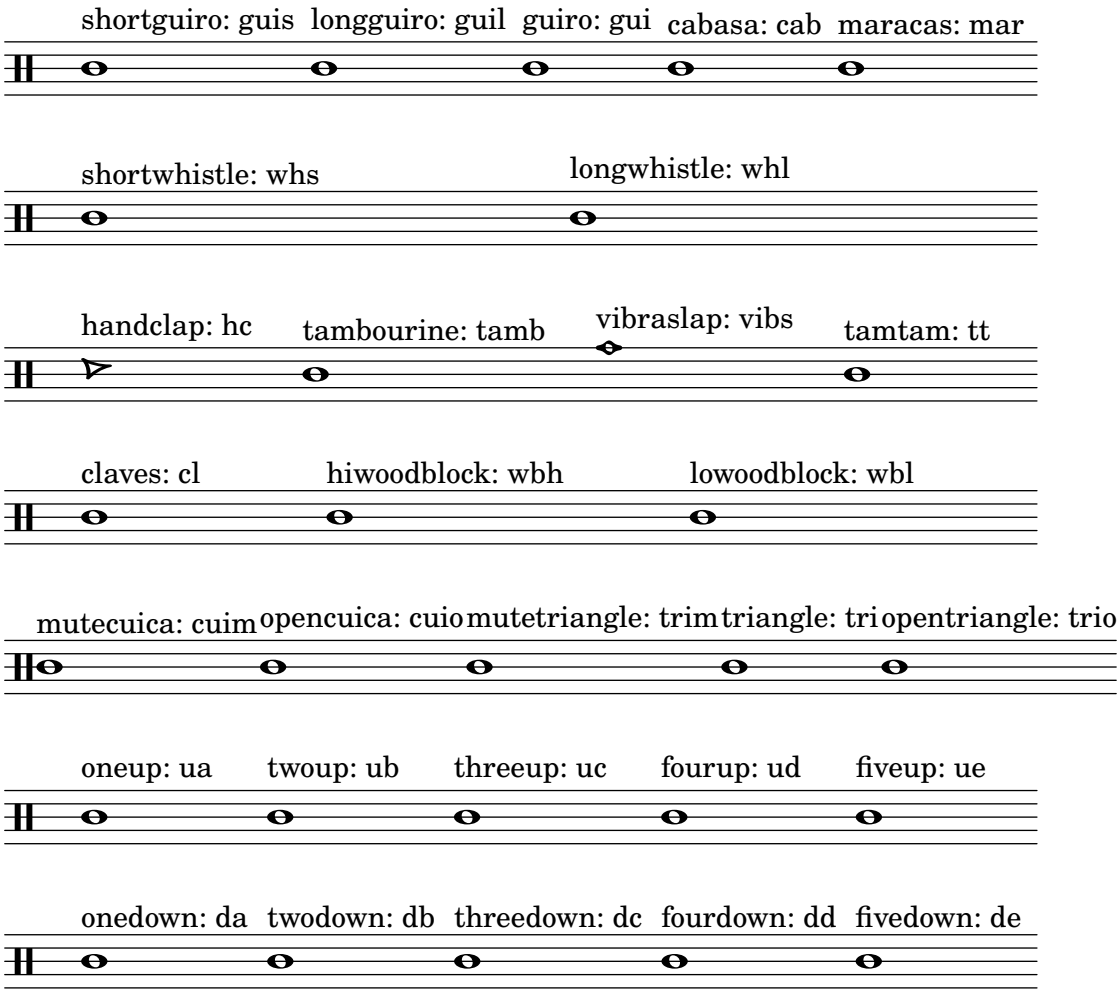
Scripts pour musique ancienne





A.11 Notes utilisées en percussion





A.12 Glossaire technique

callback

glyph

grob

interface

lexer

output-def

parser

parser variable

prob

simple-closure

smob

stencil

A.13 Liste des propriétés de contexte

`aDueText` (markup)
: Text to print at a unisono passage.

`alignAboveContext` (string)
: Where to insert newly created context in vertical alignment.

`alignBassFigureAccidentals` (boolean)
: If true, then the accidentals are aligned in bass figure context.

`alignBelowContext` (string)
: Where to insert newly created context in vertical alignment.

`associatedVoice` (string)
: Name of the `Voice` that has the melody for this `Lyrics` line.

`autoAccidentals` (list)
: List of different ways to typeset an accidental.

  For determining when to print an accidental, several different rules are tried. The rule that gives the highest number of accidentals is used.

  Each entry in the list is either a symbol or a procedure.

  symbol
  : The symbol is the name of the context in which the following rules are to be applied. For example, if *context* is Section "Score" dans *Référence des propriétés internes* then all staves share accidentals, and if *context* is Section "Staff" dans *Référence des propriétés internes* then all voices in the same staff share accidentals, but staves do not.

  procedure
  : The procedure represents an accidental rule to be applied to the previously specified context.

    The procedure takes the following arguments:

    `context`
    : The current context to which the rule should be applied.

    `pitch`
    : The pitch of the note to be evaluated.

    `barnum`
    : The current bar number.

    `measurepos`
    : The current measure position.

    The procedure returns a pair of booleans. The first states whether an extra natural should be added. The second states whether an accidental should be printed. `(#t . #f)` does not make sense.

`autoBeamCheck` (procedure)
: A procedure taking three arguments, *context*, *dir* [start/stop (-1 or 1)], and *test* [shortest note in the beam]. A non-`#f` return value starts or stops the auto beam.

`autoBeaming` (boolean)
: If set to true then beams are generated automatically.

`autoCautionaries` (list)
: List similar to `autoAccidentals`, but it controls cautionary accidentals rather than normal ones. Both lists are tried, and the one giving the most accidentals wins. In case of draw, a normal accidental is typeset.

`automaticBars` (boolean)
: If set to false then bar lines will not be printed automatically; they must be explicitly created with a `\bar` command. Unlike the `\cadenzaOn` keyword, measures are still

counted. Bar line generation will resume according to that count if this property is unset.

`barAlways` (boolean)
: If set to true a bar line is drawn after each note.

`barCheckSynchronize` (boolean)
: If true then reset `measurePosition` when finding a bar check.

`barNumberVisibility` (procedure)
: A Procedure that takes an integer and returns whether the corresponding bar number should be printed.

`baseMoment` (moment)
: Smallest unit of time that will stand on its own as a subdivided section.

`bassFigureFormatFunction` (procedure)
: A procedure that is called to produce the formatting for a `BassFigure` grob. It takes a list of `BassFigureEvent`s, a context, and the grob to format.

`bassStaffProperties` (list)
: An alist of property settings to apply for the down staff of `PianoStaff`. Used by `\autochange`.

`beamExceptions` (list)
: An alist of exceptions to autobeam rules that normally end on beats.

`beatStructure` (list)
: List of `baseMoment`s that are combined to make beats.

`chordChanges` (boolean)
: Only show changes in chords scheme?

`chordNameExceptions` (list)
: An alist of chord exceptions. Contains `(`*`chord . markup`*`)` entries.

`chordNameExceptionsFull` (list)
: An alist of full chord exceptions. Contains `(`*`chord . markup`*`)` entries.

`chordNameExceptionsPartial` (list)
: An alist of partial chord exceptions. Contains `(`*`chord . (prefix-markup suffix-markup`*`))` entries.

`chordNameFunction` (procedure)
: The function that converts lists of pitches to chord names.

`chordNameLowercaseMinor` (boolean)
: Downcase roots of minor chords?

`chordNameSeparator` (markup)
: The markup object used to separate parts of a chord name.

`chordNoteNamer` (procedure)
: A function that converts from a pitch object to a text markup. Used for single pitches.

`chordPrefixSpacer` (number)
: The space added between the root symbol and the prefix of a chord name.

`chordRootNamer` (procedure)
: A function that converts from a pitch object to a text markup. Used for chords.

`clefGlyph` (string)
: Name of the symbol within the music font.

`clefOctavation` (integer)
: Add this much extra octavation. Values of 7 and -7 are common.

`clefPosition` (number)
: Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

`completionBusy` (boolean)
: Whether a completion-note head is playing.

`connectArpeggios` (boolean)
: If set, connect arpeggios across piano staff.

`countPercentRepeats` (boolean)
: If set, produce counters for percent repeats.

`createKeyOnClefChange` (boolean)
: Print a key signature whenever the clef is changed.

`createSpacing` (boolean)
: Create `StaffSpacing` objects? Should be set for staves.

`crescendoSpanner` (symbol)
: The type of spanner to be used for crescendi. Available values are '`hairpin`' and '`text`'. If unset, a hairpin crescendo is used.

`crescendoText` (markup)
: The text to print at start of non-hairpin crescendo, i.e., '`cresc.`'.

`cueClefGlyph` (string)
: Name of the symbol within the music font.

`cueClefOctavation` (integer)
: Add this much extra octavation. Values of 7 and -7 are common.

`cueClefPosition` (number)
: Where should the center of the clef symbol go, measured in half staff spaces from the center of the staff.

`currentBarNumber` (integer)
: Contains the current barnumber. This property is incremented at every bar line.

`decrescendoSpanner` (symbol)
: The type of spanner to be used for decrescendi. Available values are '`hairpin`' and '`text`'. If unset, a hairpin decrescendo is used.

`decrescendoText` (markup)
: The text to print at start of non-hairpin decrescendo, i.e., '`dim.`'.

`defaultBarType` (string)
: Set the default type of bar line. See `whichBar` for information on available bar types.

  This variable is read by Section "Timing_translator" dans *Référence des propriétés internes* at Section "Score" dans *Référence des propriétés internes* level.

`defaultStrings` (list)
: A list of strings to use in calculating frets for tablatures and fretboards if no strings are provided in the notes for the current moment.

`doubleRepeatType` (string)
: Set the default bar line for double repeats.

`doubleSlurs` (boolean)
: If set, two slurs are created for every slurred note, one above and one below the chord.

`drumPitchTable` (hash table)
: A table mapping percussion instruments (symbols) to pitches.

`drumStyleTable` (hash table)
: A hash table which maps drums to layout settings. Predefined values: '`drums-style`', '`timbales-style`', '`congas-style`', '`bongos-style`', and '`percussion-style`'.

  The layout style is a hash table, containing the drum-pitches (e.g., the symbol '`hihat`') as keys, and a list `(`*`notehead-style script vertical-position`*`)` as values.

`explicitClefVisibility` (vector)
: '`break-visibility`' function for clef changes.

`explicitCueClefVisibility` (vector)
: '`break-visibility`' function for cue clef changes.

`explicitKeySignatureVisibility` (vector)
: '`break-visibility`' function for explicit key changes. '`\override`' of the `break-visibility` property will set the visibility for normal (i.e., at the start of the line) key signatures.

`extendersOverRests` (boolean)
: Whether to continue extenders as they cross a rest.

`extraNatural` (boolean)
: Whether to typeset an extra natural sign before accidentals changing from a non-natural to another non-natural.

`figuredBassAlterationDirection` (direction)
: Where to put alterations relative to the main figure.

`figuredBassCenterContinuations` (boolean)
: Whether to vertically center pairs of extender lines. This does not work with three or more lines.

`figuredBassFormatter` (procedure)
: A routine generating a markup for a bass figure.

`figuredBassPlusDirection` (direction)
: Where to put plus signs relative to the main figure.

`fingeringOrientations` (list)
: A list of symbols, containing '`left`', '`right`', '`up`' and/or '`down`'. This list determines where fingerings are put relative to the chord being fingered.

`firstClef` (boolean)
: If true, create a new clef when starting a staff.

`followVoice` (boolean)
: If set, note heads are tracked across staff switches by a thin line.

`fontSize` (number)
: The relative size of all grobs in a context.

`forbidBreak` (boolean)
: If set to `##t`, prevent a line break at this point.

`forceClef` (boolean)
: Show clef symbol, even if it has not changed. Only active for the first clef after the property is set, not for the full staff.

`fretLabels` (list)
: A list of strings or Scheme-formatted markups containing, in the correct order, the labels to be used for lettered frets in tablature.

`glissandoMap` (list)
: A map in the form of '((source1 . target1) (source2 . target2) (sourcen . targetn)) showing the glissandi to be drawn for note columns. The value '() will default to '((0 . 0) (1 . 1) (n . n)), where n is the minimal number of note-heads in the two note columns between which the glissandi occur.

`gridInterval` (moment)
: Interval for which to generate `GridPoint`s.

`handleNegativeFrets` (symbol)
: How the automatic fret calculator should handle calculated negative frets. Values include `'ignore`, to leave them out of the diagram completely, `'include`, to include them as calculated, and `'recalculate`, to ignore the specified string and find a string where they will fit with a positive fret number.

`harmonicAccidentals` (boolean)
: If set, harmonic notes in chords get accidentals.

`harmonicDots` (boolean)
: If set, harmonic notes in dotted chords get dots.

`highStringOne` (boolean)
: Whether the first string is the string with highest pitch on the instrument. This used by the automatic string selector for tablature notation.

`ignoreBarChecks` (boolean)
: Ignore bar checks.

`ignoreFiguredBassRest` (boolean)
: Don't swallow rest events.

`ignoreMelismata` (boolean)
: Ignore melismata for this Section "Lyrics" dans *Référence des propriétés internes* line.

`implicitBassFigures` (list)
: A list of bass figures that are not printed as numbers, but only as extender lines.

`implicitTimeSignatureVisibility` (vector)
: break visibility for the default time signature.

`includeGraceNotes` (boolean)
: Do not ignore grace notes for Section "Lyrics" dans *Référence des propriétés internes*.

`instrumentCueName` (markup)
: The name to print if another instrument is to be taken.

`instrumentEqualizer` (procedure)
: A function taking a string (instrument name), and returning a `(min . max)` pair of numbers for the loudness range of the instrument.

`instrumentName` (markup)
: The name to print left of a staff. The `instrumentName` property labels the staff in the first system, and the `shortInstrumentName` property labels following lines.

`instrumentTransposition` (pitch)
: Define the transposition of the instrument. Its value is the pitch that sounds like middle C. This is used to transpose the MIDI output, and `\quote`s.

`internalBarNumber` (integer)
: Contains the current barnumber. This property is used for internal timekeeping, among others by the `Accidental_engraver`.

`keepAliveInterfaces` (list)
: A list of symbols, signifying grob interfaces that are worth keeping a staff with `remove-empty` set around for.

`keyAlterationOrder` (list)
: An alist that defines in what order alterations should be printed. The format is `(`*`step . alter`*`)`, where *step* is a number from 0 to 6 and *alter* from -2 (sharp) to 2 (flat).

`keySignature` (list)
: The current key signature. This is an alist containing `(`*`step . alter`*`)` or `((`*`octave . step`*`) . `*`alter`*`)`, where *step* is a number in the range 0 to 6 and *alter* a fraction, denoting alteration. For alterations, use symbols, e.g. `keySignature = #`((6 . ,FLAT))`.

`lyricMelismaAlignment` (direction)
: Alignment to use for a melisma syllable.

`majorSevenSymbol` (markup)
: How should the major 7th be formatted in a chord name?

`markFormatter` (procedure)
: A procedure taking as arguments the context and the rehearsal mark. It should return the formatted mark as a markup object.

`maximumFretStretch` (number)
: Don't allocate frets further than this from specified frets.

`measureLength` (moment)
: Length of one measure in the current time signature.

`measurePosition` (moment)
: How much of the current measure have we had. This can be set manually to create incomplete measures.

`melismaBusyProperties` (list)
: A list of properties (symbols) to determine whether a melisma is playing. Setting this property will influence how lyrics are aligned to notes. For example, if set to `#'(melismaBusy beamMelismaBusy)`, only manual melismata and manual beams are considered. Possible values include `melismaBusy`, `slurMelismaBusy`, `tieMelismaBusy`, and `beamMelismaBusy`.

`metronomeMarkFormatter` (procedure)
: How to produce a metronome markup. Called with two arguments: a `TempoChangeEvent` and context.

`middleCClefPosition` (number)
: The position of the middle C, as determined only by the clef. This can be calculated by looking at `clefPosition` and `clefGlyph`.

`middleCCuePosition` (number)
: The position of the middle C, as determined only by the clef of the cue notes. This can be calculated by looking at `cueClefPosition` and `cueClefGlyph`.

`middleCOffset` (number)
: The offset of middle C from the position given by `middleCClefPosition` This is used for ottava brackets.

`middleCPosition` (number)
: The place of the middle C, measured in half staff-spaces. Usually determined by looking at `middleCClefPosition` and `middleCOffset`.

`midiChannelMapping` (symbol)
: How to map MIDI channels: per `instrument` (default), `staff` or `voice`.

`midiInstrument` (string)
: Name of the MIDI instrument to use.

`midiMaximumVolume` (number)
: Analogous to `midiMinimumVolume`.

`midiMergeUnisons` (boolean)
: If true, output only one MIDI note-on event when notes with the same pitch, in the same MIDI-file track, overlap.

`midiMinimumVolume` (number)
: Set the minimum loudness for MIDI. Ranges from 0 to 1.

`minimumFret` (number)
: The tablature auto string-selecting mechanism selects the highest string with a fret at least `minimumFret`.

`minimumPageTurnLength` (moment)
: Minimum length of a rest for a page turn to be allowed.

`minimumRepeatLengthForPageTurn` (moment)
: Minimum length of a repeated section for a page turn to be allowed within that section.

`noChordSymbol` (markup)
: Markup to be displayed for rests in a ChordNames context.

`noteToFretFunction` (procedure)
: Convert list of notes and list of defined strings to full list of strings and fret numbers. Parameters: The context, a list of note events, a list of tabstring events, and the fretboard grob if a fretboard is desired.

`ottavation` (markup)
: If set, the text for an ottava spanner. Changing this creates a new text spanner.

`output` (music output)
: The output produced by a score-level translator during music interpretation.

`partCombineTextsOnNote` (boolean)
: Print part-combine texts only on the next note rather than immediately on rests or skips.

`pedalSostenutoStrings` (list)
: See `pedalSustainStrings`.

`pedalSostenutoStyle` (symbol)
: See `pedalSustainStyle`.

`pedalSustainStrings` (list)
: A list of strings to print for sustain-pedal. Format is `(up updown down)`, where each of the three is the string to print when this is done with the pedal.

`pedalSustainStyle` (symbol)
: A symbol that indicates how to print sustain pedals: `text`, `bracket` or `mixed` (both).

`pedalUnaCordaStrings` (list)
: See `pedalSustainStrings`.

`pedalUnaCordaStyle` (symbol)
: See `pedalSustainStyle`.

`predefinedDiagramTable` (hash table)
: The hash table of predefined fret diagrams to use in FretBoards.

`printKeyCancellation` (boolean)
: Print restoration alterations before a key signature change.

`printOctaveNames` (boolean)
: Print octave marks for the `NoteNames` context.

`printPartCombineTexts` (boolean)
: Set ‘Solo’ and ‘A due’ texts in the part combiner?

`proportionalNotationDuration` (moment)
: Global override for shortest-playing duration. This is used for switching on proportional notation.

`rehearsalMark` (integer)
: The last rehearsal mark printed.

`repeatCommands` (list)
: This property is a list of commands of the form `(list 'volta x)`, where x is a string or `#f`. `'end-repeat` is also accepted as a command.

`repeatCountVisibility` (procedure)
: A procedure taking as arguments an integer and context, returning whether the corresponding percent repeat number should be printed when `countPercentRepeats` is set.

`restCompletionBusy` (boolean)
: Signal whether a completion-rest is active.

`restNumberThreshold` (number)
: If a multimeasure rest has more measures than this, a number is printed.

`searchForVoice` (boolean)
: Signal whether a search should be made of all contexts in the context hierarchy for a voice to provide rhythms for the lyrics.

`shapeNoteStyles` (vector)
: Vector of symbols, listing style for each note head relative to the tonic (qv.) of the scale.

`shortInstrumentName` (markup)
: See `instrumentName`.

`shortVocalName` (markup)
: Name of a vocal line, short version.

`skipBars` (boolean)
: If set to true, then skip the empty bars that are produced by multimeasure notes and rests. These bars will not appear on the printed output. If not set (the default), multimeasure notes and rests expand into their full length, printing the appropriate number of empty bars so that synchronization with other voices is preserved.

```
{
  r1 r1*3 R1*3
  \set Score.skipBars= ##t
  r1*3 R1*3
}
```

`skipTypesetting` (boolean)
: If true, no typesetting is done, speeding up the interpretation phase. Useful for debugging large scores.

`soloIIText` (markup)
: The text for the start of a solo for voice 'two' when part-combining.

`soloText` (markup)
: The text for the start of a solo when part-combining.

`squashedPosition` (integer)
: Vertical position of squashing for Section "Pitch_squash_engraver" dans *Référence des propriétés internes*.

`staffLineLayoutFunction` (procedure)
: Layout of staff lines, `traditional`, or `semitone`.

`stanza` (markup)
: Stanza 'number' to print before the start of a verse. Use in `Lyrics` context.

`stemLeftBeamCount` (integer)
: Specify the number of beams to draw on the left side of the next note. Overrides automatic beaming. The value is only used once, and then it is erased.

`stemRightBeamCount` (integer)
: See `stemLeftBeamCount`.

`stringNumberOrientations` (list)
: See `fingeringOrientations`.

`stringOneTopmost` (boolean)
: Whether the first string is printed on the top line of the tablature.

`stringTunings` (list)
: The tablature strings tuning. It is a list of the pitches of each string (starting with the lowest numbered one).

`strokeFingerOrientations` (list)
: See `fingeringOrientations`.

`subdivideBeams` (boolean)
: If set, multiple beams will be subdivided at `baseMoment` positions by only drawing one beam over the beat.

`suggestAccidentals` (boolean)
: If set, accidentals are typeset as cautionary suggestions over the note.

`systemStartDelimiter` (symbol)
: Which grob to make for the start of the system/staff? Set to `SystemStartBrace`, `SystemStartBracket` or `SystemStartBar`.

`systemStartDelimiterHierarchy` (pair)
: A nested list, indicating the nesting of a start delimiters.

`tablatureFormat` (procedure)
: A function formatting a tablature note head. Called with three arguments: context, string number and, fret number. It returns the text as a markup.

`tabStaffLineLayoutFunction` (procedure)
: A function determining the staff position of a tablature note head. Called with two arguments: the context and the string.

`tempoHideNote` (boolean)
: Hide the note = count in tempo marks.

`tempoWholesPerMinute` (moment)
: The tempo in whole notes per minute.

`tieWaitForNote` (boolean)
: If true, tied notes do not have to follow each other directly. This can be used for writing out arpeggios.

`timeSignatureFraction` (pair of numbers)
: A pair of numbers, signifying the time signature. For example, `#'(4 . 4)` is a 4/4 time signature.

`timeSignatureSettings` (list)
: A nested alist of settings for time signatures. Contains elements for various time signatures. The element for each time signature contains entries for `baseMoment`, `beatStructure`, and `beamExceptions`.

`timing` (boolean)
: Keep administration of measure length, position, bar number, etc.? Switch off for cadenzas.

`tonic` (pitch)
: The tonic of the current scale.

`topLevelAlignment` (boolean)
: If true, the *Vertical_align_engraver* will create a *VerticalAlignment*; otherwise, it will create a *StaffGrouper*

`trebleStaffProperties` (list)
: An alist of property settings to apply for the up staff of `PianoStaff`. Used by `\autochange`.

`tremoloFlags` (integer)
: The number of tremolo flags to add if no number is specified.

`tupletFullLength` (boolean)
: If set, the tuplet is printed up to the start of the next note.

`tupletFullLengthNote` (boolean)
: If set, end at the next note, otherwise end on the matter (time signatures, etc.) before the note.

`tupletSpannerDuration` (moment)
: Normally, a tuplet bracket is as wide as the `\times` expression that gave rise to it. By setting this property, you can make brackets last shorter.

```
{
  \set tupletSpannerDuration = #(ly:make-moment 1 4)
```

```
  \times 2/3 { c8 c c c c c }
}
```

`useBassFigureExtenders` (boolean)
: Whether to use extender lines for repeated bass figures.

`vocalName` (markup)
: Name of a vocal line.

`voltaSpannerDuration` (moment)
: This specifies the maximum duration to use for the brackets printed for `\alternative`. This can be used to shrink the length of brackets in the situation where one alternative is very large.

`whichBar` (string)
: This property is read to determine what type of bar line to create.

  Example:

  ```
  \set Staff.whichBar = "|:"
  ```

  This will create a start-repeat bar in this staff only. Valid values are described in Section “bar-line-interface” dans *Référence des propriétés internes*.

A.14 Propriétés de mise en forme

`add-stem-support` (boolean)
: If set, the `Stem` object is included in this script’s support.

`after-line-breaking` (boolean)
: Dummy property, used to trigger callback for `after-line-breaking`.

`align-dir` (direction)
: Which side to align? `-1`: left side, `0`: around center of width, `1`: right side.

`allow-loose-spacing` (boolean)
: If set, column can be detached from main spacing.

`allow-span-bar` (boolean)
: If false, no inter-staff bar line will be created below this bar line.

`alteration` (number)
: Alteration numbers for accidental.

`alteration-alist` (list)
: List of `(`*`pitch . accidental`*`)` pairs for key signature.

`annotation` (string)
: Annotate a grob for debug purposes.

`annotation-balloon` (boolean)
: Print the balloon around an annotation.

`annotation-line` (boolean)
: Print the line from an annotation to the grob that it annotates.

`arpeggio-direction` (direction)
: If set, put an arrow on the arpeggio squiggly line.

`arrow-length` (number)
: Arrow length.

`arrow-width` (number)
: Arrow width.

`auto-knee-gap` (dimension, in staff space)
: If a gap is found between note heads where a horizontal beam fits that is larger than this number, make a kneed beam.

`average-spacing-wishes` (boolean)
: If set, the spacing wishes are averaged over staves.

`avoid-note-head` (boolean)
: If set, the stem of a chord does not pass through all note heads, but starts at the last note head.

`avoid-slur` (symbol)
: Method of handling slur collisions. Choices are `inside`, `outside`, `around`, and `ignore`. `inside` adjusts the slur if needed to keep the grob inside the slur. `outside` moves the grob vertically to the outside of the slur. `around` moves the grob vertically to the outside of the slur only if there is a collision. `ignore` does not move either. In grobs whose notational significance depends on vertical position (such as accidentals, clefs, etc.), `outside` and `around` behave like `ignore`.

`axes` (list)
: List of axis numbers. In the case of alignment grobs, this should contain only one number.

`base-shortest-duration` (moment)
: Spacing is based on the shortest notes in a piece. Normally, pieces are spaced as if notes at least as short as this are present.

`baseline-skip` (dimension, in staff space)
: Distance between base lines of multiple lines of text.

`beam-thickness` (dimension, in staff space)
: Beam thickness, measured in `staff-space` units.

`beam-width` (dimension, in staff space)
: Width of the tremolo sign.

`beamed-stem-shorten` (list)
: How much to shorten beamed stems, when their direction is forced. It is a list, since the value is different depending on the number of flags and beams.

`beaming` (pair)
: Pair of number lists. Each number list specifies which beams to make. `0` is the central beam, `1` is the next beam toward the note, etc. This information is used to determine how to connect the beaming patterns from stem to stem inside a beam.

`beamlet-default-length` (pair)
: A pair of numbers. The first number specifies the default length of a beamlet that sticks out of the left hand side of this stem; the second number specifies the default length of the beamlet to the right. The actual length of a beamlet is determined by taking either the default length or the length specified by `beamlet-max-length-proportion`, whichever is smaller.

`beamlet-max-length-proportion` (pair)
: The maximum length of a beamlet, as a proportion of the distance between two adjacent stems.

`before-line-breaking` (boolean)
: Dummy property, used to trigger a callback function.

`between-cols` (pair)
: Where to attach a loose column to.

`bound-details` (list)
: An alist of properties for determining attachments of spanners to edges.

`bound-padding` (number)
: The amount of padding to insert around spanner bounds.

`bracket-flare` (pair of numbers)
: A pair of numbers specifying how much edges of brackets should slant outward. Value `0.0` means straight edges.

`bracket-visibility` (boolean or symbol)
: This controls the visibility of the tuplet bracket. Setting it to false prevents printing of the bracket. Setting the property to `if-no-beam` makes it print only if there is no beam associated with this tuplet bracket.

`break-align-anchor` (number)
: Grobs aligned to this break-align grob will have their X-offsets shifted by this number. In bar lines, for example, this is used to position grobs relative to the (visual) center of the bar line.

`break-align-anchor-alignment` (number)
: Read by `ly:break-aligned-interface::calc-extent-aligned-anchor` for aligning an anchor to a grob's extent.

`break-align-orders` (vector)
: Defines the order in which prefatory matter (clefs, key signatures) appears. The format is a vector of length 3, where each element is one order for end-of-line, middle of line, and start-of-line, respectively. An order is a list of symbols.

  For example, clefs are put after key signatures by setting

```
\override Score.BreakAlignment #'break-align-orders =
  #(make-vector 3 '(span-bar
                    breathing-sign
                    staff-bar
                    key
                    clef
                    time-signature))
```

`break-align-symbol` (symbol)
: This key is used for aligning and spacing breakable items.

`break-align-symbols` (list)
: A list of symbols that determine which break-aligned grobs to align this to. If the grob selected by the first symbol in the list is invisible due to break-visibility, we will align to the next grob (and so on). Choices are `left-edge`, `ambitus`, `breathing-sign`, `clef`, `staff-bar`, `key-cancellation`, `key-signature`, `time-signature`, and `custos`.

`break-overshoot` (pair of numbers)
: How much does a broken spanner stick out of its bounds?

`break-visibility` (vector)
: A vector of 3 booleans, `#(`*`end-of-line unbroken begin-of-line`*`)`. `#t` means visible, `#f` means killed.

`breakable` (boolean)
: Allow breaks here.

`c0-position` (integer)
: An integer indicating the position of middle C.

`circled-tip` (boolean)
: Put a circle at start/end of hairpins (al/del niente).

`clip-edges` (boolean)
: Allow outward pointing beamlets at the edges of beams?

`collapse-height` (dimension, in staff space)
: Minimum height of system start delimiter. If equal or smaller, the bracket/brace/line is removed.

`collision-interfaces` (list)
: A list of interfaces for which automatic beam-collision resolution is run.

`collision-voice-only` (boolean)
: Does automatic beam collsion apply only to the voice in which the beam was created?

`color` (color)
: The color of this grob.

`common-shortest-duration` (moment)
: The most common shortest note length. This is used in spacing. Enlarging this sets the score tighter.

`concaveness` (number)
: A beam is concave if its inner stems are closer to the beam than the two outside stems. This number is a measure of the closeness of the inner stems. It is used for damping the slope of the beam.

`connect-to-neighbor` (pair)
: Pair of booleans, indicating whether this grob looks as a continued break.

`control-points` (list)
: List of offsets (number pairs) that form control points for the tie, slur, or bracket shape. For Béziers, this should list the control points of a third-order Bézier curve.

`damping` (number)
: Amount of beam slope damping.

`dash-definition` (pair)
: List of `dash-elements` defining the dash structure. Each `dash-element` has a starting t value, an ending t-value, a `dash-fraction`, and a `dash-period`.

`dash-fraction` (number)
: Size of the dashes, relative to `dash-period`. Should be between `0.0` (no line) and `1.0` (continuous line).

`dash-period` (number)
: The length of one dash together with whitespace. If negative, no line is drawn at all.

`default-direction` (direction)
: Direction determined by note head positions.

`default-staff-staff-spacing` (list)
: The settings to use for `staff-staff-spacing` when it is unset, for ungrouped staves and for grouped staves that do not have the relevant `StaffGrouper` property set (`staff-staff-spacing` or `staffgroup-staff-spacing`).

`details` (list)
: Alist of parameters for detailed grob behavior. More information on the allowed parameters for a grob can be found by looking at the top of the Internals Reference page for each interface having a `details` property.

`digit-names` (vector)
: Names for string finger digits.

`direction` (direction)
: If `side-axis` is `0` (or `#X`), then this property determines whether the object is placed `#LEFT`, `#CENTER` or `#RIGHT` with respect to the other object. Otherwise, it determines whether the object is placed `#UP`, `#CENTER` or `#DOWN`. Numerical values may also be used: `#UP=1`, `#DOWN=-1`, `#LEFT=-1`, `#RIGHT=1`, `#CENTER=0`.

`dot-count` (integer)
: The number of dots.

`dot-negative-kern` (number)
: The space to remove between a dot and a slash in percent repeat glyphs. Larger values bring the two elements closer together.

`dot-placement-list` (list)
: List consisting of `(`*`description string-number fret-number finger-number`*`)` entries used to define fret diagrams.

`duration-log` (integer)
: The 2-log of the note head duration, i.e., `0` = whole note, `1` = half note, etc.

`eccentricity` (number)
: How asymmetrical to make a slur. Positive means move the center to the right.

`edge-height` (pair)
: A pair of numbers specifying the heights of the vertical edges: `(`*`left-height . right-height`*`)`.

`edge-text` (pair)
: A pair specifying the texts to be set at the edges: `(`*`left-text . right-text`*`)`.

`expand-limit` (integer)
: Maximum number of measures expanded in church rests.

`extra-dy` (number)
: Slope glissandi this much extra.

`extra-offset` (pair of numbers)
: A pair representing an offset. This offset is added just before outputting the symbol, so the typesetting engine is completely oblivious to it. The values are measured in `staff-space` units of the staff's `StaffSymbol`.

`extra-spacing-height` (pair of numbers)
: In the horizontal spacing problem, we increase the height of each item by this amount (by adding the 'car' to the bottom of the item and adding the 'cdr' to the top of the item). In order to make a grob infinitely high (to prevent the horizontal spacing problem from placing any other grobs above or below this grob), set this to `(-inf.0 . +inf.0)`.

`extra-spacing-width` (pair of numbers)
: In the horizontal spacing problem, we pad each item by this amount (by adding the 'car' on the left side of the item and adding the 'cdr' on the right side of the item). In order to make a grob take up no horizontal space at all, set this to `(+inf.0 . -inf.0)`.

`extra-X-extent` (pair of numbers)
: A grob is enlarged in X dimension by this much.

`extra-Y-extent` (pair of numbers)
: A grob is enlarged in Y dimension by this much.

`flag` (stencil)
: A function returning the full flag stencil for the `Stem`, which is passed to the function as the only argument. The default ly:stem::calc-stencil function uses the `flag-style` property to determine the correct glyph for the flag. By providing your own function, you can create arbitrary flags.

`flag-count` (number)
: The number of tremolo beams.

`flag-style` (symbol)
: A symbol determining what style of flag glyph is typeset on a `Stem`. Valid options include `'()` for standard flags, `'mensural` and `'no-flag`, which switches off the flag.

`font-encoding` (symbol)
: The font encoding is the broadest category for selecting a font. Currently, only lilypond's system fonts (Emmentaler) are using this property. Available values are `fetaMusic` (Emmentaler), `fetaBraces`, `fetaText` (Emmentaler).

`font-family` (symbol)
: The font family is the broadest category for selecting text fonts. Options include: `sans`, `roman`.

`font-name` (string)
: Specifies a file name (without extension) of the font to load. This setting overrides selection using `font-family`, `font-series` and `font-shape`.

`font-series` (symbol)
: Select the series of a font. Choices include `medium`, `bold`, `bold-narrow`, etc.

`font-shape` (symbol)
: Select the shape of a font. Choices include `upright`, `italic`, `caps`.

`font-size` (number)
: The font size, compared to the 'normal' size. `0` is style-sheet's normal size, `-1` is smaller, `+1` is bigger. Each step of 1 is approximately 12% larger; 6 steps are exactly a factor 2 larger. Fractional values are allowed.

`footnote-text` (markup)
: A footnote for the grob.

`force-hshift` (number)
: This specifies a manual shift for notes in collisions. The unit is the note head width of the first voice note. This is used by Section "note-collision-interface" dans *Référence des propriétés internes*.

`fraction` (pair of numbers)
: Numerator and denominator of a time signature object.

`french-beaming` (boolean)
: Use French beaming style for this stem. The stem stops at the innermost beams.

`fret-diagram-details` (list)
: An alist of detailed grob properties for fret diagrams. Each alist entry consists of a `(`*`property . value`*`)` pair. The properties which can be included in `fret-diagram-details` include the following:

  - `barre-type` – Type of barre indication used. Choices include `curved`, `straight`, and `none`. Default `curved`.

- `capo-thickness` – Thickness of capo indicator, in multiples of fret-space. Default value 0.5.
- `dot-color` – Color of dots. Options include `black` and `white`. Default `black`.
- `dot-label-font-mag` – Magnification for font used to label fret dots. Default value 1.
- `dot-position` – Location of dot in fret space. Default 0.6 for dots without labels, 0.95-`dot-radius` for dots with labels.
- `dot-radius` – Radius of dots, in terms of fret spaces. Default value 0.425 for labeled dots, 0.25 for unlabeled dots.
- `finger-code` – Code for the type of fingering indication used. Options include `none`, `in-dot`, and `below-string`. Default `none` for markup fret diagrams, `below-string` for `FretBoards` fret diagrams.
- `fret-count` – The number of frets. Default 4.
- `fret-label-custom-format` – The format string to be used label the lowest fret number, when `number-type` equals to `custom`. Default `"~a"`.
- `fret-label-font-mag` – The magnification of the font used to label the lowest fret number. Default 0.5.
- `fret-label-vertical-offset` – The offset of the fret label from the center of the fret in direction parallel to strings. Default 0.
- `label-dir` – Side to which the fret label is attached. `-1`, `#LEFT`, or `#DOWN` for left or down; `1`, `#RIGHT`, or `#UP` for right or up. Default `#RIGHT`.
- `mute-string` – Character string to be used to indicate muted string. Default `"x"`.
- `number-type` – Type of numbers to use in fret label. Choices include `roman-lower`, `roman-upper`, `arabic` and `custom`. In the later case, the format string is supplied by the `fret-label-custom-format` property. Default `roman-lower`.
- `open-string` – Character string to be used to indicate open string. Default `"o"`.
- `orientation` – Orientation of fret-diagram. Options include `normal`, `landscape`, and `opposing-landscape`. Default `normal`.
- `string-count` – The number of strings. Default 6.
- `string-label-font-mag` – The magnification of the font used to label fingerings at the string, rather than in the dot. Default value 0.6 for `normal` orientation, 0.5 for `landscape` and `opposing-landscape`.
- `string-thickness-factor` – Factor for changing thickness of each string in the fret diagram. Thickness of string k is given by `thickness` * (1+`string-thickness-factor`) ^ (k-1). Default 0.
- `top-fret-thickness` – The thickness of the top fret line, as a multiple of the standard thickness. Default value 3.
- `xo-font-magnification` – Magnification used for mute and open string indicators. Default value 0.5.
- `xo-padding` – Padding for open and mute indicators from top fret. Default value 0.25.

`full-length-padding` (number)
: How much padding to use at the right side of a full-length tuplet bracket.

`full-length-to-extent` (boolean)
: Run to the extent of the column for a full-length tuplet bracket.

`full-measure-extra-space` (number)
: Extra space that is allocated at the beginning of a measure with only one note. This property is read from the NonMusicalPaperColumn that begins the measure.

`full-size-change` (boolean)
: Don't make a change clef smaller.

`gap` (dimension, in staff space)
: Size of a gap in a variable symbol.

`gap-count` (integer)
: Number of gapped beams for tremolo.

`glyph` (string)
: A string determining what 'style' of glyph is typeset. Valid choices depend on the function that is reading this property.

`glyph-name` (string)
: The glyph name within the font.

`glyph-name-alist` (list)
: An alist of key-string pairs.

`graphical` (boolean)
: Display in graphical (vs. text) form.

`grow-direction` (direction)
: Crescendo or decrescendo?

`hair-thickness` (number)
: Thickness of the thin line in a bar line.

`harp-pedal-details` (list)
: An alist of detailed grob properties for harp pedal diagrams. Each alist entry consists of a `(property . value)` pair. The properties which can be included in harp-pedal-details include the following:

  - `box-offset` – Vertical shift of the center of flat/sharp pedal boxes above/below the horizontal line. Default value 0.8.
  - `box-width` – Width of each pedal box. Default value 0.4.
  - `box-height` – Height of each pedal box. Default value 1.0.
  - `space-before-divider` – Space between boxes before the first divider (so that the diagram can be made symmetric). Default value 0.8.
  - `space-after-divider` – Space between boxes after the first divider. Default value 0.8.
  - `circle-thickness` – Thickness (in unit of the line-thickness) of the ellipse around circled pedals. Default value 0.5.
  - `circle-x-padding` – Padding in X direction of the ellipse around circled pedals. Default value 0.15.
  - `circle-y-padding` – Padding in Y direction of the ellipse around circled pedals. Default value 0.2.

`head-direction` (direction)
: Are the note heads left or right in a semitie?

`height` (dimension, in staff space)
: Height of an object in `staff-space` units.

`height-limit` (dimension, in staff space)
: Maximum slur height: The longer the slur, the closer it is to this height.

`hide-tied-accidental-after-break` (boolean)
: If set, an accidental that appears on a tied note after a line break will not be displayed.

`horizontal-shift` (integer)
: An integer that identifies ranking of `NoteColumn`s for horizontal shifting. This is used by Section "note-collision-interface" dans *Référence des propriétés internes*.

`horizontal-skylines` (pair of skylines)
: Two skylines, one to the left and one to the right of this grob.

`ignore-collision` (boolean)
: If set, don't do note collision resolution on this `NoteColumn`.

`implicit` (boolean)
: Is this an implicit bass figure?

`inspect-index` (integer)
: If debugging is set, set beam and slur configuration to this index, and print the respective scores.

`inspect-quants` (pair of numbers)
: If debugging is set, set beam and slur quants to this position, and print the respective scores.

`keep-inside-line` (boolean)
: If set, this column cannot have objects sticking into the margin.

`kern` (dimension, in staff space)
: Amount of extra white space to add. For bar lines, this is the amount of space after a thick line.

`knee` (boolean)
: Is this beam kneed?

`knee-spacing-correction` (number)
: Factor for the optical correction amount for kneed beams. Set between `0` for no correction and `1` for full correction.

`labels` (list)
: List of labels (symbols) placed on a column.

`layer` (integer)
: An integer which determines the order of printing objects. Objects with the lowest value of layer are drawn first, then objects with progressively higher values are drawn, so objects with higher values overwrite objects with lower values. By default most objects are assigned a layer value of 1.

`ledger-line-thickness` (pair of numbers)
: The thickness of ledger lines. It is the sum of 2 numbers: The first is the factor for line thickness, and the second for staff space. Both contributions are added.

`left-bound-info` (list)
: An alist of properties for determining attachments of spanners to edges.

`left-padding` (dimension, in staff space)
: The amount of space that is put left to an object (e.g., a lyric extender).

`length` (dimension, in staff space)
: User override for the stem length of unbeamed stems.

`length-fraction` (number)
: Multiplier for lengths. Used for determining ledger lines and stem lengths.

`line-break-penalty` (number)
: Penalty for a line break at this column. This affects the choices of the line breaker; it avoids a line break at a column with a positive penalty and prefers a line break at a column with a negative penalty.

`line-break-permission` (symbol)
: Instructs the line breaker on whether to put a line break at this column. Can be `force` or `allow`.

`line-break-system-details` (list)
: An alist of properties to use if this column is the start of a system.

`line-count` (integer)
: The number of staff lines.

`line-positions` (list)
: Vertical positions of staff lines.

`line-thickness` (number)
: The thickness of the tie or slur contour.

`long-text` (markup)
: Text markup. See Section “Formatting text” dans *Manuel de notation*.

`max-beam-connect` (integer)
: Maximum number of beams to connect to beams from this stem. Further beams are typeset as beamlets.

`max-stretch` (number)
: The maximum amount that this `VerticalAxisGroup` can be vertically stretched (for example, in order to better fill a page).

`measure-count` (integer)
: The number of measures for a multi-measure rest.

`measure-length` (moment)
: Length of a measure. Used in some spacing situations.

`merge-differently-dotted` (boolean)
: Merge note heads in collisions, even if they have a different number of dots. This is normal notation for some types of polyphonic music.

  `merge-differently-dotted` only applies to opposing stem directions (i.e., voice 1 & 2).

`merge-differently-headed` (boolean)
: Merge note heads in collisions, even if they have different note heads. The smaller of the two heads is rendered invisible. This is used in polyphonic guitar notation. The value of this setting is used by Section “note-collision-interface” dans *Référence des propriétés internes*.

  `merge-differently-headed` only applies to opposing stem directions (i.e., voice 1 & 2).

`minimum-distance` (dimension, in staff space)
: Minimum distance between rest and notes or beam.

`minimum-length` (dimension, in staff space)
: Try to make a spanner at least this long, normally in the horizontal direction. This requires an appropriate callback for the `springs-and-rods` property. If added to a `Tie`, this sets the minimum distance between noteheads.

`minimum-length-fraction` (number)
: Minimum length of ledger line as fraction of note head size.

`minimum-space` (dimension, in staff space)
: Minimum distance that the victim should move (after padding).

`minimum-X-extent` (pair of numbers)
: Minimum size of an object in X dimension, measured in `staff-space` units.

`minimum-Y-extent` (pair of numbers)
: Minimum size of an object in Y dimension, measured in `staff-space` units.

`neutral-direction` (direction)
: Which direction to take in the center of the staff.

`neutral-position` (number)
: Position (in half staff spaces) where to flip the direction of custos stem.

`next` (graphical (layout) object)
: Object that is next relation (e.g., the lyric syllable following an extender).

`no-alignment` (boolean)
: If set, don't place this grob in a `VerticalAlignment`; rather, place it using its own `Y-offset` callback.

`no-ledgers` (boolean)
: If set, don't draw ledger lines on this object.

`no-stem-extend` (boolean)
: If set, notes with ledger lines do not get stems extending to the middle staff line.

`non-break-align-symbols` (list)
: A list of symbols that determine which NON-break-aligned interfaces to align this to.

`non-default` (boolean)
: Set for manually specified clefs.

`non-musical` (boolean)
: True if the grob belongs to a `NonMusicalPaperColumn`.

`nonstaff-nonstaff-spacing` (list)
: The spacing alist controlling the distance between the current non-staff line and the next non-staff line in the direction of `staff-affinity`, if both are on the same side of the related staff, and `staff-affinity` is either `UP` or `DOWN`. See `staff-staff-spacing` for a description of the alist structure.

`nonstaff-relatedstaff-spacing` (list)
: The spacing alist controlling the distance between the current non-staff line and the nearest staff in the direction of `staff-affinity`, if there are no non-staff lines between the two, and `staff-affinity` is either `UP` or `DOWN`. If `staff-affinity` is `CENTER`, then `nonstaff-relatedstaff-spacing` is used for the nearest staves on *both* sides, even if other non-staff lines appear between the current one and either of the staves. See `staff-staff-spacing` for a description of the alist structure.

`nonstaff-unrelatedstaff-spacing` (list)
: The spacing alist controlling the distance between the current non-staff line and the nearest staff in the opposite direction from `staff-affinity`, if there are no other non-staff lines between the two, and `staff-affinity` is either `UP` or `DOWN`. See `staff-staff-spacing` for a description of the alist structure.

`normalized-endpoints` (pair)
: Represents left and right placement over the total spanner, where the width of the spanner is normalized between 0 and 1.

`note-names` (vector)
: Vector of strings containing names for easy-notation note heads.

`outside-staff-horizontal-padding` (number)
: By default, an outside-staff-object can be placed so that is it very close to another grob horizontally. If this property is set, the outside-staff-object is raised so that it is not so close to its neighbor.

`outside-staff-padding` (number)
: The padding to place between this grob and the staff when spacing according to `outside-staff-priority`.

`outside-staff-priority` (number)
: If set, the grob is positioned outside the staff in such a way as to avoid all collisions. In case of a potential collision, the grob with the smaller `outside-staff-priority` is closer to the staff.

`packed-spacing` (boolean)
: If set, the notes are spaced as tightly as possible.

`padding` (dimension, in staff space)
: Add this much extra space between objects that are next to each other.

`padding-pairs` (list)
: An alist mapping `(`*`name . name`*`)` to distances.

`page-break-penalty` (number)
: Penalty for page break at this column. This affects the choices of the page breaker; it avoids a page break at a column with a positive penalty and prefers a page break at a column with a negative penalty.

`page-break-permission` (symbol)
: Instructs the page breaker on whether to put a page break at this column. Can be `force` or `allow`.

`page-turn-penalty` (number)
: Penalty for a page turn at this column. This affects the choices of the page breaker; it avoids a page turn at a column with a positive penalty and prefers a page turn at a column with a negative penalty.

`page-turn-permission` (symbol)
: Instructs the page breaker on whether to put a page turn at this column. Can be `force` or `allow`.

`parenthesized` (boolean)
: Parenthesize this grob.

`positions` (pair of numbers)
: Pair of staff coordinates `(`*`left . right`*`)`, where both *left* and *right* are in `staff-space` units of the current staff. For slurs, this value selects which slur candidate to use; if extreme positions are requested, the closest one is taken.

`prefer-dotted-right` (boolean)
: For note collisions, prefer to shift dotted up-note to the right, rather than shifting just the dot.

`ratio` (number)
: Parameter for slur shape. The higher this number, the quicker the slur attains its `height-limit`.

`remove-empty` (boolean)
: If set, remove group if it contains no interesting items.

`remove-first` (boolean)
: Remove the first staff of an orchestral score?

`restore-first` (boolean)
: Print a natural before the accidental.

`rhythmic-location` (rhythmic location)
: Where (bar number, measure position) in the score.

`right-bound-info` (list)
: An alist of properties for determining attachments of spanners to edges.

`right-padding` (dimension, in staff space)
: Space to insert on the right side of an object (e.g., between note and its accidentals).

`rotation` (list)
: Number of degrees to rotate this object, and what point to rotate around. For example, `#'(45 0 0)` rotates by 45 degrees around the center of this object.

`same-direction-correction` (number)
: Optical correction amount for stems that are placed in tight configurations. This amount is used for stems with the same direction to compensate for note head to stem distance.

`script-priority` (number)
: A sorting key that determines in what order a script is within a stack of scripts.

`self-alignment-X` (number)
: Specify alignment of an object. The value `-1` means left aligned, `0` centered, and `1` right-aligned in X direction. Other numerical values may also be specified.

`self-alignment-Y` (number)
: Like `self-alignment-X` but for the Y axis.

`shorten-pair` (pair of numbers)
: The lengths to shorten a text-spanner on both sides, for example a pedal bracket. Positive values shorten the text-spanner, while negative values lengthen it.

`shortest-duration-space` (dimension, in staff space)
: Start with this much space for the shortest duration. This is expressed in `spacing-increment` as unit. See also Section “spacing-spanner-interface” dans *Référence des propriétés internes*.

`shortest-playing-duration` (moment)
: The duration of the shortest note playing here.

`shortest-starter-duration` (moment)
: The duration of the shortest note that starts here.

`side-axis` (number)
: If the value is `#X` (or equivalently `0`), the object is placed horizontally next to the other object. If the value is `#Y` or `1`, it is placed vertically.

`side-relative-direction` (direction)
: Multiply direction of `direction-source` with this to get the direction of this object.

`size` (number)
: Size of object, relative to standard size.

`skyline-horizontal-padding` (number)
: For determining the vertical distance between two staves, it is possible to have a configuration which would result in a tight interleaving of grobs from the top staff and the bottom staff. The larger this parameter is, the farther apart the staves are placed in such a configuration.

`skyline-vertical-padding` (number)
: The amount by which the left and right skylines of a column are padded vertically, beyond the `Y-extent`s and `extra-spacing-height`s of the constituent grobs in the column. Increase this to prevent interleaving of grobs from adjacent columns.

`slash-negative-kern` (number)
: The space to remove between slashes in percent repeat glyphs. Larger values bring the two elements closer together.

`slope` (number)
: The slope of this object.

`slur-padding` (number)
: Extra distance between slur and script.

`space-alist` (list)
: A table that specifies distances between prefatory items, like clef and time-signature. The format is an alist of spacing tuples: `(break-align-symbol type . distance)`, where *type* can be the symbols `minimum-space` or `extra-space`.

`space-to-barline` (boolean)
: If set, the distance between a note and the following non-musical column will be measured to the bar line instead of to the beginning of the non-musical column. If there is a clef change followed by a bar line, for example, this means that we will try to space the non-musical column as though the clef is not there.

`spacing-increment` (number)
: Add this much space for a doubled duration. Typically, the width of a note head. See also Section “spacing-spanner-interface” dans *Référence des propriétés internes*.

`spacing-pair` (pair)
: A pair of alignment symbols which set an object's spacing relative to its left and right `BreakAlignment`s.

  For example, a `MultiMeasureRest` will ignore prefatory items at its bounds (i.e., clefs, key signatures and time signatures) using the following override:

  ```
  \override MultiMeasureRest #'spacing-pair = #'(staff-bar . staff-bar)
  ```

`spanner-id` (string)
: An identifier to distinguish concurrent spanners.

`springs-and-rods` (boolean)
: Dummy variable for triggering spacing routines.

`stacking-dir` (direction)
: Stack objects in which direction?

`staff-affinity` (direction)
: The direction of the staff to use for spacing the current non-staff line. Choices are `UP`, `DOWN`, and `CENTER`. If `CENTER`, the non-staff line will be placed equidistant between the two nearest staves on either side, unless collisions or other spacing constraints prevent this. Setting `staff-affinity` for a staff causes it to be treated as a non-staff line. Setting `staff-affinity` to `#f` causes a non-staff line to be treated as a staff.

`staff-padding` (dimension, in staff space)
: Maintain this much space between reference points and the staff. Its effect is to align objects of differing sizes (like the dynamics **p** and **f**) on their baselines.

`staff-position` (number)
: Vertical position, measured in half staff spaces, counted from the middle line.

`staff-space` (dimension, in staff space)
: Amount of space between staff lines, expressed in global `staff-space`.

`staff-staff-spacing` (list)
: When applied to a staff-group's `StaffGrouper` grob, this spacing alist controls the distance between consecutive staves within the staff-group. When applied to a staff's `VerticalAxisGroup` grob, it controls the distance between the staff and the nearest staff below it in the same system, replacing any settings inherited from the `StaffGrouper` grob of the containing staff-group, if there is one. This property remains in effect even when non-staff lines appear between staves. The alist can contain the following keys:

  - `basic-distance` – the vertical distance, measured in staff-spaces, between the reference points of the two items when no collisions would result, and no stretching or compressing is in effect.
  - `minimum-distance` – the smallest allowable vertical distance, measured in staff-spaces, between the reference points of the two items, when compressing is in effect.
  - `padding` – the minimum required amount of unobstructed vertical whitespace between the bounding boxes (or skylines) of the two items, measured in staff-spaces.
  - `stretchability` – a unitless measure of the dimension's relative propensity to stretch. If zero, the distance will not stretch (unless collisions would result).

`staffgroup-staff-spacing` (list)
: The spacing alist controlling the distance between the last staff of the current staff-group and the staff just below it in the same system, even if one or more non-staff lines exist between the two staves. If the `staff-staff-spacing` property of the staff's `VerticalAxisGroup` grob is set, that is used instead. See `staff-staff-spacing` for a description of the alist structure.

`stem-attachment` (pair of numbers)
: An `(x . y)` pair where the stem attaches to the notehead.

`stem-end-position` (number)
: Where does the stem end (the end is opposite to the support-head)?

`stem-spacing-correction` (number)
: Optical correction amount for stems that are placed in tight configurations. For opposite directions, this amount is the correction for two normal sized stems that overlap completely.

`stemlet-length` (number)
: How long should be a stem over a rest?

`stencil` (stencil)
: The symbol to print.

`stencils` (list)
: Multiple stencils, used as intermediate value.

`strict-grace-spacing` (boolean)
: If set, main notes are spaced normally, then grace notes are put left of the musical columns for the main notes.

`strict-note-spacing` (boolean)
: If set, unbroken columns with non-musical material (clefs, bar lines, etc.) are not spaced separately, but put before musical columns.

`stroke-style` (string)
: Set to `"grace"` to turn stroke through flag on.

`style` (symbol)
: This setting determines in what style a grob is typeset. Valid choices depend on the `stencil` callback reading this property.

`text` (markup)
: Text markup. See Section “Formatting text” dans *Manuel de notation*.

`text-direction` (direction)
: This controls the ordering of the words. The default `RIGHT` is for roman text. Arabic or Hebrew should use `LEFT`.

`thick-thickness` (number)
: Bar line thickness, measured in `line-thickness`.

`thickness` (number)
: Line thickness, generally measured in `line-thickness`.

`thin-kern` (number)
: The space after a hair-line in a bar line.

`tie-configuration` (list)
: List of `(`*`position . dir`*`)` pairs, indicating the desired tie configuration, where *position* is the offset from the center of the staff in staff space and *dir* indicates the direction of the tie (`1`=>up, `-1`=>down, `0`=>center). A non-pair entry in the list causes the corresponding tie to be formatted automatically.

`to-barline` (boolean)
: If true, the spanner will stop at the bar line just before it would otherwise stop.

`toward-stem-shift` (number)
: Amount by which scripts are shifted toward the stem if their direction coincides with the stem direction. `0.0` means keep the default position (centered on the note head), `1.0` means centered on the stem. Interpolated values are possible.

`transparent` (boolean)
: This makes the grob invisible.

`uniform-stretching` (boolean)
: If set, items stretch proportionally to their durations. This looks better in complex polyphonic patterns.

`used` (boolean)
: If set, this spacing column is kept in the spacing problem.

`vertical-skylines` (pair of skylines)
: Two skylines, one above and one below this grob.

`when` (moment)
: Global time step associated with this column happen?

`whiteout` (boolean)
: If true, the grob is printed over a white background to white-out underlying material, if the grob is visible. Usually #f by default.

`width` (dimension, in staff space)
: The width of a grob measured in staff space.

`word-space` (dimension, in staff space)
: Space to insert between words in texts.

`X-extent` (pair of numbers)
: Hard coded extent in X direction.

`X-offset` (number)
: The horizontal amount that this object is moved relative to its X-parent.

`Y-extent` (pair of numbers)
: Hard coded extent in Y direction.

`Y-offset` (number)
: The vertical amount that this object is moved relative to its Y-parent.

`zigzag-length` (dimension, in staff space)
: The length of the lines of a zigzag, relative to `zigzag-width`. A value of `1` gives 60-degree zigzags.

`zigzag-width` (dimension, in staff space)
: The width of one zigzag squiggle. This number is adjusted slightly so that the glissando line can be constructed from a whole number of squiggles.

A.15 Fonctions mucicales

`acciaccatura` - *music* (music)
: Create an acciaccatura from the following music expression

`addChordShape` - *key-symbol* (symbol) *tuning* (pair) *shape-definition* (string or pair)
: Add chord shape *shape-definition* to the *chord-shape-table* hash with the key *(cons key-symbol tuning)*.

`addInstrumentDefinition` - *name* (string) *lst* (list)
: Create instrument *name* with properties *list*.

`addQuote` - *name* (string) *music* (music)
: Define *music* as a quotable music expression named *name*

`afterGrace` - *main* (music) *grace* (music)
: Create *grace* note(s) after a *main* music expression.

`allowPageTurn`
: Allow a page turn. May be used at toplevel (ie between scores or markups), or inside a score.

`applyContext` - *proc* (procedure)
: Modify context properties with Scheme procedure *proc*.

`applyMusic` - *func* (procedure) *music* (music)
: Apply procedure *func* to *music*.

`applyOutput` - *ctx* (symbol) *proc* (procedure)
: Apply function `proc` to every layout object in context `ctx`

`appoggiatura` - *music* (music)
: Create an appoggiatura from *music*

`assertBeamQuant` - *l* (pair) *r* (pair)
: Testing function: check whether the beam quants *l* and *r* are correct

`assertBeamSlope` - *comp* (procedure)
: Testing function: check whether the slope of the beam is the same as `comp`

`autochange` - *music* (music)
: Make voices that switch between staves automatically

`balloonGrobText` - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup)
: Attach *text* to *grob-name* at offset *offset* (use like `\once`)

`balloonText` - *offset* (pair of numbers) *text* (markup)
: Attach *text* at *offset* (use like `\tweak`)

`bar` - *type* (string)
: Insert a bar line of type *type*

`barNumberCheck` - *n* (integer)
: Print a warning if the current bar number is not *n*.

`bendAfter` - *delta* (real number)
: Create a fall or doit of pitch interval *delta*.

`bookOutputName` - *newfilename* (string)
: Direct output for the current book block to *newfilename*.

`bookOutputSuffix` - *newsuffix* (string)
: Set the output filename suffix for the current book block to *newsuffix*.

`breathe` Insert a breath mark.

`chordGlissando` - *mus1* (music) *mus2* (music)
: Make a glissando between the notes of triads `mus1` and `mus2`.

`clef` - *type* (string)
: Set the current clef to *type*.

`compoundMeter` - *args* (pair)
: Create compound time signatures. The argument is a Scheme list of lists. Each list describes one fraction, with the last entry being the denominator, while the first entries describe the summands in the enumerator. If the time signature consists of just one fraction, the list can be given directly, i.e. not as a list containing a single list. For example, a time signature of (3+1)/8 + 2/4 would be created as `\compoundMeter #'((3 1 8) (2 4))`, and a time signature of (3+2)/8 as `\compoundMeter #'((3 2 8))` or shorter `\compoundMeter #'(3 2 8)`.

`contextStringTuning` - *tuning* (symbol) *chord* (music)
: (undocumented; fixme)

`cueClef` - *type* (string)
: Set the current cue clef to *type*.

`cueClefUnset`
: Unset the current cue clef.

`cueDuring` - *what* (string) *dir* (direction) *main-music* (music)
: Insert contents of quote *what* corresponding to *main-music*, in a CueVoice oriented by *dir*.

`cueDuringWithClef` - *what* (string) *dir* (direction) *clef* (string) *main-music* (music)
: Insert contents of quote *what* corresponding to *main-music*, in a CueVoice oriented by *dir*.

`deadNote` - *note* (music)
: Print *note* with a cross-shaped note head.

`defaultNoteHeads`
: Revert to the default note head style.

`displayLilyMusic` - *music* (music)
: Display the LilyPond input representation of *music* to the console.

`displayMusic` - *music* (music)
: Display the internal representation of *music* to the console.

`endSpanners` - *music* (music)
: Terminate the next spanner prematurely after exactly one note without the need of a specific end spanner.

`featherDurations` - *factor* (moment) *argument* (music)
: Adjust durations of music in *argument* by rational *factor*.

`footnote` - *offset* (pair of numbers) *text* (markup) *footnote* (markup)
: Attach *text* at *offset* with *text* referring to *footnote* (use like `\tweak`)

`footnoteGrob` - *grob-name* (symbol) *offset* (pair of numbers) *text* (markup) *footnote* (markup)
: Attach *text* to *grob-name* at offset *offset*, with *text* referring to *footnote* (use like `\once`)

`grace` - *music* (music)
: Insert *music* as grace notes.

`harmonicByFret` - *fret* (number) *music* (music)
: (undocumented; fixme)

`harmonicByRatio` - *ratio* (number) *music* (music)
: (undocumented; fixme)

`harmonicNote` - *note* (music)
: Print *note* with a diamond-shaped note head.

`harmonicsOn`
: Set the default note head style to a diamond-shaped style.

`instrumentSwitch` - *name* (string)
: Switch instrument to *name*, which must be predefined with `\addInstrumentDefinition`.

`inversion` - *around* (music) *to* (music) *music* (music)
: Invert *music* about *around* and transpose from *around* to *to*.

`keepWithTag` - *tag* (symbol) *music* (music)
Include only elements of *music* that are tagged with *tag*.

`killCues` - *music* (music)
Remove cue notes from *music*.

`label` - *label* (symbol)
Create *label* as a bookmarking label.

`language` - *language* (string)
Set note names for language *language*.

`languageRestore`
Restore a previously-saved pitchnames alist.

`languageSaveAndChange` - *language* (string)
Store the previous pitchnames alist, and set a new one.

`makeClusters` - *arg* (music)
Display chords in *arg* as clusters.

`makeDefaultStringTunings` - *default-tuning-alist* (list)
(undocumented; fixme)

`makeStringTuning` - *tuning* (symbol) *chord* (music)
(undocumented; fixme)

`modalInversion` - *around* (music) *to* (music) *scale* (music) *music* (music)
Invert *music* about *around* using *scale* and transpose from *around* to *to*.

`modalTranspose` - *from* (music) *to* (music) *scale* (music) *music* (music)
Transpose *music* from pitch *from* to pitch *to* using *scale*.

`musicMap` - *proc* (procedure) *mus* (music)
Apply *proc* to *mus* and all of the music it contains.

`noPageBreak`
Forbid a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

`noPageTurn`
Forbid a page turn. May be used at toplevel (i.e., between scores or markups), or inside a score.

`octaveCheck` - *pitch-note* (music)
Octave check.

`ottava` - *octave* (integer)
Set the octavation.

`overrideProperty` - *name* (string) *property* (symbol) *value* (any type)
Set *property* to *value* in all grobs named *name*. The *name* argument is a string of the form `"Context.GrobName"` or `"GrobName"`.

`overrideTimeSignatureSettings` - *time-signature* (pair) *base-moment* (pair) *beat-structure* (list) *beam-exceptions* (list)
Override `timeSignatureSettings` for time signatures of *time-signature* to have settings of *base-moment*, *beat-structure*, and *beam-exceptions*.

`pageBreak`
Force a page break. May be used at toplevel (i.e., between scores or markups), or inside a score.

`pageTurn` Force a page turn between two scores or top-level markups.

`palmMute` - *note* (music)
Print *note* with a triangle-shaped note head.

`palmMuteOn`
Set the default note head style to a triangle-shaped style.

`parallelMusic` - *voice-ids* (list) *music* (music)
Define parallel music sequences, separated by '`|`' (bar check signs), and assign them to the identifiers provided in *voice-ids*.

voice-ids: a list of music identifiers (symbols containing only letters)

music: a music sequence, containing BarChecks as limiting expressions.

Example:

```
\parallelMusic #'(A B C) {
  c c | d d | e e |
  d d | e e | f f |
}
<==>
  A = { c c | d d | }
  B = { d d | e e | }
  C = { e e | f f | }
```

`parenthesize` - *arg* (music)
Tag *arg* to be parenthesized.

`partcombine` - *part1* (music) *part2* (music)
Take the music in *part1* and *part2* and typeset so that they share a staff.

`partcombineForce` - *type* (symbol-or-boolean) *once* (boolean)
Override the part-combiner.

`phrasingSlurDashPattern` - *dash-fraction* (number) *dash-period* (number)
Set up a custom style of dash pattern for *dash-fraction* ratio of line to space repeated at *dash-period* interval.

`pitchedTrill` - *main-note* (music) *secondary-note* (music)
Print a trill with *main-note* as the main note of the trill and print *secondary-note* as a stemless note head in parentheses.

`pointAndClickOff`
Suppress generating extra code in final-format (e.g. pdf) files to point back to the lilypond source statement.

`pointAndClickOn`
Enable generation of code in final-format (e.g. pdf) files to reference the originating lilypond source statement; this is helpful when developing a score but generates bigger final-format files.

`quoteDuring` - *what* (string) *main-music* (music)
Indicate a section of music to be quoted. *what* indicates the name of the quoted voice, as specified in an `\addQuote` command. *main-music* is used to indicate the length of music to be quoted; usually contains spacers or multi-measure rests.

`removeWithTag` - *tag* (symbol) *music* (music)
Remove elements of *music* that are tagged with *tag*.

`resetRelativeOctave` - *reference-note* (music)
Set the octave inside a \relative section.

`retrograde` - *music* (music)
: Return *music* in reverse order.

`revertTimeSignatureSettings` - *time-signature* (pair)
: Revert `timeSignatureSettings` for time signatures of *time-signature*.

`rightHandFinger` - *finger* (number or string)
: Apply *finger* as a fingering indication.

`scaleDurations` - *fraction* (pair of numbers) *music* (music)
: Multiply the duration of events in *music* by *fraction*.

`shiftDurations` - *dur* (integer) *dots* (integer) *arg* (music)
: Scale *arg* up by a factor of *2^dur*(2-(1/2)^dots)*.

`slurDashPattern` - *dash-fraction* (number) *dash-period* (number)
: (undocumented; fixme)

`spacingTweaks` - *parameters* (list)
: Set the system stretch, by reading the 'system-stretch property of the 'parameters' assoc list.

`storePredefinedDiagram` - *fretboard-table* (hash table) *chord* (music) *tuning* (pair) *diagram-definition* (string or pair)
: Add predefined fret diagram defined by *diagram-definition* for the chord pitches *chord* and the stringTuning *tuning*.

`styledNoteHeads` - *style* (symbol) *heads* (list or symbol) *music* (music)
: Set *heads* in *music* to *style*.

`tabChordRepetition`
: Include the string information in a chord repetition.

`tag` - *tag* (symbol) *arg* (music)
: Add *tag* to the `tags` property of *arg*.

`tieDashPattern` - *dash-fraction* (number) *dash-period* (number)
: (undocumented; fixme)

`tocItem` - *text* (markup)
: Add a line to the table of content, using the `tocItemMarkup` paper variable markup

`transposedCueDuring` - *what* (string) *dir* (direction) *pitch-note* (music) *main-music* (music)
: Insert notes from the part *what* into a voice called `cue`, using the transposition defined by *pitch-note*. This happens simultaneously with *main-music*, which is usually a rest. The argument *dir* determines whether the cue notes should be notated as a first or second voice.

`transposition` - *pitch-note* (music)
: Set instrument transposition

`tweak` - *sym* (symbol) *val* (any type) *arg* (music)
: Add `sym . val` to the `tweaks` property of *arg*.

`unfoldRepeats` - *music* (music)
: Force any `\repeat volta`, `\repeat tremolo` or `\repeat percent` commands in *music* to be interpreted as `\repeat unfold`.

`withMusicProperty` - *sym* (symbol) *val* (any type) *music* (music)
: Set *sym* to *val* in *music*.

`xNote` - *note* (music)
: Print *note* with a cross-shaped note head.

`xNotesOn` Set the default note head style to a cross-shaped style.

A.16 Types de prédicats prédéfinis

R5RS primary predicates

Type predicate	Description
`boolean?`	boolean
`char?`	character
`number?`	number
`pair?`	pair
`port?`	port
`procedure?`	procedure
`string?`	string
`symbol?`	symbol
`vector?`	vector

R5RS secondary predicates

Type predicate	Description
`char-alphabetic?`	alphabetic character
`char-lower-case?`	lower-case character
`char-numeric?`	numeric character
`char-upper-case?`	upper-case character
`char-whitespace?`	whitespace character
`complex?`	complex number
`eof-object?`	end-of-file object
`even?`	even number
`exact?`	exact number
`inexact?`	inexact number
`input-port?`	input port
`integer?`	integer
`list?`	list *(use* `cheap-list?` *for faster processing)*
`negative?`	negative number
`null?`	null
`odd?`	odd number
`output-port?`	output port
`positive?`	positive number
`rational?`	rational number
`real?`	real number
`zero?`	zero

Guile predicates

Type predicate	Description
`hash-table?`	hash table

LilyPond scheme predicates

Type predicate	Description
`boolean-or-symbol?`	boolean or symbol
`cheap-list?`	list *(use this instead of* `list?` *for faster processing)*
`color?`	color

`grob-list?`	list of grobs
`list-or-symbol?`	list or symbol
`markup?`	markup
`markup-command-list?`	markup command list
`markup-list?`	markup list
`moment-pair?`	pair of moment objects
`number-or-grob?`	number or grob
`number-or-pair?`	number or pair
`number-or-string?`	number or string
`number-pair?`	pair of numbers
`rhythmic-location?`	rhythmic location
`scheme?`	any type
`string-or-pair?`	string or pair
`string-or-symbol?`	string or symbol

LilyPond exported predicates

Type predicate	**Description**
`ly:box?`	box
`ly:context?`	context
`ly:dimension?`	dimension, in staff space
`ly:dir?`	direction
`ly:dispatcher?`	dispatcher
`ly:duration?`	duration
`ly:font-metric?`	font metric
`ly:grob?`	graphical (layout) object
`ly:grob-array?`	array of grobs
`ly:input-location?`	input location
`ly:item?`	item
`ly:iterator?`	iterator
`ly:lily-lexer?`	lily-lexer
`ly:lily-parser?`	lily-parser
`ly:listener?`	listener
`ly:moment?`	moment
`ly:music?`	music
`ly:music-function?`	music function
`ly:music-list?`	list of music objects
`ly:music-output?`	music output
`ly:otf-font?`	OpenType font
`ly:output-def?`	output definition
`ly:page-marker?`	page marker
`ly:pango-font?`	pango font
`ly:paper-book?`	paper book
`ly:paper-system?`	paper-system Prob
`ly:pitch?`	pitch
`ly:prob?`	property object
`ly:score?`	score
`ly:simple-closure?`	simple closure
`ly:skyline?`	skyline
`ly:skyline-pair?`	pair of skylines
`ly:source-file?`	source file

`ly:spanner?`	spanner
`ly:stencil?`	stencil
`ly:stream-event?`	stream event
`ly:translator?`	translator
`ly:translator-group?`	translator group

A.17 Fonctions Scheme

`ly:add-context-mod` *contextmods modification* [Fonction]
Adds the given context *modification* to the list *contextmods* of context modifications.

`ly:add-file-name-alist` *alist* [Fonction]
Add mappings for error messages from *alist*.

`ly:add-interface` *iface desc props* [Fonction]
Add a new grob interface. *iface* is the interface name, *desc* is the interface description, and *props* is the list of user-settable properties for the interface.

`ly:add-listener` *list disp cl* [Fonction]
Add the listener *list* to the dispatcher *disp*. Whenever *disp* hears an event of class *cl*, it is forwarded to *list*.

`ly:add-option` *sym val description* [Fonction]
Add a program option *sym*. *val* is the default value and *description* is a string description.

`ly:all-grob-interfaces` [Fonction]
Return the hash table with all grob interface descriptions.

`ly:all-options` [Fonction]
Get all option settings in an alist.

`ly:all-stencil-expressions` [Fonction]
Return all symbols recognized as stencil expressions.

`ly:assoc-get` *key alist default-value strict-checking* [Fonction]
Return value if *key* in *alist*, else *default-value* (or `#f` if not specified). If *strict-checking* is set to `#t` and *key* is not in *alist*, a programming_error is output.

`ly:axis-group-interface::add-element` *grob grob-element* [Fonction]
Set *grob* the parent of *grob-element* on all axes of *grob*.

`ly:beam-score-count` [Fonction]
count number of beam scores.

`ly:book-add-bookpart!` *book-smob book-part* [Fonction]
Add *book-part* to *book-smob* book part list.

`ly:book-add-score!` *book-smob score* [Fonction]
Add *score* to *book-smob* score list.

`ly:book-book-parts` *book* [Fonction]
Return book parts in *book*.

`ly:book-header` *book* [Fonction]
Return header in *book*.

`ly:book-paper` *book* [Fonction]
Return paper in *book*.

`ly:book-process` *book-smob default-paper default-layout output* [Fonction]
Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).

`ly:book-process-to-systems` *book-smob default-paper default-layout output* [Fonction]
Print book. *output* is passed to the backend unchanged. For example, it may be a string (for file based outputs) or a socket (for network based output).

`ly:book-scores` *book* [Fonction]
Return scores in *book*.

`ly:box?` *x* [Fonction]
Is *x* a `Box` object?

`ly:bp` *num* [Fonction]
num bigpoints (1/72th inch).

`ly:bracket` *a iv t p* [Fonction]
Make a bracket in direction *a*. The extent of the bracket is given by *iv*. The wings protrude by an amount of *p*, which may be negative. The thickness is given by *t*.

`ly:broadcast` *disp ev* [Fonction]
Send the stream event *ev* to the dispatcher *disp*.

`ly:camel-case->lisp-identifier` *name-sym* [Fonction]
Convert `FooBar_Bla` to `foo-bar-bla` style symbol.

`ly:chain-assoc-get` *key achain default-value strict-checking* [Fonction]
Return value for *key* from a list of alists *achain*. If no entry is found, return *default-value* or `#f` if *default-value* is not specified. With *strict-checking* set to `#t`, a programming_error is output in such cases.

`ly:cm` *num* [Fonction]
num cm.

`ly:command-line-code` [Fonction]
The Scheme code specified on command-line with '`-e`'.

`ly:command-line-options` [Fonction]
The Scheme options specified on command-line with '`-d`'.

`ly:command-line-verbose?` [Fonction]
Was `be_verbose_global` set?

`ly:connect-dispatchers` *to from* [Fonction]
Make the dispatcher *to* listen to events from *from*.

`ly:context?` *x* [Fonction]
Is *x* a `Context` object?

`ly:context-current-moment` *context* [Fonction]
Return the current moment of *context*.

`ly:context-event-source` *context* [Fonction]
Return `event-source` of context *context*.

`ly:context-events-below` *context* [Fonction]
Return a `stream-distributor` that distributes all events from *context* and all its subcontexts.

`ly:context-find` *context name* [Fonction]
Find a parent of *context* that has name or alias *name*. Return `#f` if not found.

`ly:context-grob-definition` *context name* [Fonction]
Return the definition of *name* (a symbol) within *context* as an alist.

`ly:context-id` *context* [Fonction]
Return the ID string of *context*, i.e., for `\context Voice = "one" ...` return the string `one`.

`ly:context-name` *context* [Fonction]
Return the name of *context*, i.e., for `\context Voice = "one" ...` return the symbol `Voice`.

`ly:context-now` *context* [Fonction]
Return `now-moment` of context *context*.

`ly:context-parent` *context* [Fonction]
Return the parent of *context*, `#f` if none.

`ly:context-property` *context sym def* [Fonction]
Return the value for property *sym* in *context*. If *def* is given, and property value is `'()`, return *def*.

`ly:context-property-where-defined` *context name* [Fonction]
Return the context above *context* where *name* is defined.

`ly:context-pushpop-property` *context grob eltprop val* [Fonction]
Do a single `\override` or `\revert` operation in *context*. The grob definition *grob* is extended with *eltprop* (if *val* is specified) or reverted (if unspecified).

`ly:context-set-property!` *context name val* [Fonction]
Set value of property *name* in context *context* to *val*.

`ly:context-unset-property` *context name* [Fonction]
Unset value of property *name* in context *context*.

`ly:default-scale` [Fonction]
Get the global default scale.

`ly:dimension?` *d* [Fonction]
Return *d* as a number. Used to distinguish length variables from normal numbers.

`ly:dir?` *s* [Fonction]
Is *s* a direction? Valid directions are `-1`, `0`, or `1`, where `-1` represents left or down, `1` represents right or up, and `0` represents a neutral direction.

`ly:dispatcher?` *x* [Fonction]
Is *x* a `Dispatcher` object?

`ly:duration?` *x* [Fonction]
Is *x* a `Duration` object?

`ly:duration<?` *p1 p2* [Fonction]
Is *p1* shorter than *p2*?

`ly:duration->string` *dur* [Fonction]
Convert *dur* to a string.

`ly:duration-dot-count` *dur* [Fonction]
Extract the dot count from *dur*.

`ly:duration-factor` *dur* [Fonction]
Extract the compression factor from *dur*. Return it as a pair.

`ly:duration-length` *dur* [Fonction]
The length of the duration as a `moment`.

`ly:duration-log` *dur* [Fonction]
Extract the duration log from *dur*.

`ly:effective-prefix` [Fonction]
Return effective prefix.

`ly:encode-string-for-pdf` *str* [Fonction]
Encode the given string to either Latin1 (which is a subset of the PDFDocEncoding) or if that's not possible to full UTF-16BE with Byte-Order-Mark (BOM).

`ly:engraver-announce-end-grob` *engraver grob cause* [Fonction]
Announce the end of a grob (i.e., the end of a spanner) originating from given *engraver* instance, with *grob* being a grob. *cause* should either be another grob or a music event.

`ly:engraver-make-grob` *engraver grob-name cause* [Fonction]
Create a grob originating from given *engraver* instance, with given *grob-name*, a symbol. *cause* should either be another grob or a music event.

`ly:error` *str rest* [Fonction]
A Scheme callable function to issue the error *str*. The error is formatted with `format` and *rest*.

`ly:eval-simple-closure` *delayed closure scm-start scm-end* [Fonction]
Evaluate a simple *closure* with the given *delayed* argument. If *scm-start* and *scm-end* are defined, evaluate it purely with those start and end points.

`ly:event-deep-copy` *m* [Fonction]
Copy *m* and all sub expressions of *m*.

`ly:event-property` *sev sym* [Fonction]
Get the property *sym* of stream event *mus*. If *sym* is undefined, return `'()`.

`ly:event-set-property!` *ev sym val* [Fonction]
Set property *sym* in event *ev* to *val*.

`ly:expand-environment` *str* [Fonction]
Expand `$VAR` and `${VAR}` in *str*.

`ly:export` *arg* [Fonction]
Export a Scheme object to the parser so it is treated as an identifier.

`ly:find-file` *name* [Fonction]
Return the absolute file name of *name*, or `#f` if not found.

`ly:font-config-add-directory` *dir* [Fonction]
Add directory *dir* to FontConfig.

`ly:font-config-add-font` *font* [Fonction]
Add font *font* to FontConfig.

`ly:font-config-display-fonts` [Fonction]
Dump a list of all fonts visible to FontConfig.

`ly:font-config-get-font-file` *name* [Fonction]
Get the file for font *name*.

`ly:font-design-size` *font* [Fonction]
Given the font metric *font*, return the design size, relative to the current output-scale.

`ly:font-file-name` *font* [Fonction]
Given the font metric *font*, return the corresponding file name.

`ly:font-get-glyph` *font name* [Fonction]
Return a stencil from *font* for the glyph named *name*. If the glyph is not available, return an empty stencil.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-glyph-name-to-charcode` *font name* [Fonction]
Return the character code for glyph *name* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-glyph-name-to-index` *font name* [Fonction]
Return the index for *name* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-index-to-charcode` *font index* [Fonction]
Return the character code for *index* in *font*.

Note that this command can only be used to access glyphs from fonts loaded with `ly:system-font-load`; currently, this means either the Emmentaler or Emmentaler-Brace fonts, corresponding to the font encodings `fetaMusic` and `fetaBraces`, respectively.

`ly:font-magnification` *font* [Fonction]
Given the font metric *font*, return the magnification, relative to the current output-scale.

`ly:font-metric?` *x* [Fonction]
Is *x* a `Font_metric` object?

`ly:font-name` *font* [Fonction]
Given the font metric *font*, return the corresponding name.

`ly:font-sub-fonts` *font* [Fonction]
Given the font metric *font* of an OpenType font, return the names of the subfonts within *font*.

`ly:format` *str rest* [Fonction]
LilyPond specific format, supporting `~a` and `~[0-9]f`. Basic support for `~s` is also provided.

`ly:format-output` *context* [Fonction]
Given a global context in its final state, process it and return the `Music_output` object in its final state.

`ly:get-all-function-documentation` [Fonction]
Get a hash table with all LilyPond Scheme extension functions.

`ly:get-all-translators` [Fonction]
Return a list of all translator objects that may be instantiated.

`ly:get-context-mods` *contextmod* [Fonction]
Returns the list of context modifications stored in *contextmod*.

`ly:get-listened-event-classes` [Fonction]
Return a list of all event classes that some translator listens to.

`ly:get-option` *var* [Fonction]
Get a global option setting.

`ly:gettext` *original* [Fonction]
A Scheme wrapper function for `gettext`.

`ly:grob?` *x* [Fonction]
Is *x* a `Grob` object?

`ly:grob-alist-chain` *grob global* [Fonction]
Get an alist chain for grob *grob*, with *global* as the global default. If unspecified, `font-defaults` from the layout block is taken.

`ly:grob-array?` *x* [Fonction]
Is *x* a `Grob_array` object?

`ly:grob-array->list` *grob-arr* [Fonction]
Return the elements of *grob-arr* as a Scheme list.

`ly:grob-array-length` *grob-arr* [Fonction]
Return the length of *grob-arr*.

`ly:grob-array-ref` *grob-arr index* [Fonction]
Retrieve the *index*th element of *grob-arr*.

`ly:grob-basic-properties` *grob* [Fonction]
Get the immutable properties of *grob*.

`ly:grob-chain-callback` *grob proc sym* [Fonction]
Find the callback that is stored as property *sym* of grob *grob* and chain *proc* to the head of this, meaning that it is called using *grob* and the previous callback's result.

`ly:grob-common-refpoint` *grob other axis* [Fonction]
Find the common refpoint of *grob* and *other* for *axis*.

`ly:grob-common-refpoint-of-array` *grob others axis* [Fonction]
Find the common refpoint of *grob* and *others* (a grob-array) for *axis*.

`ly:grob-default-font` *grob* [Fonction]
Return the default font for grob *grob*.

`ly:grob-extent` *grob refp axis* [Fonction]
Get the extent in *axis* direction of *grob* relative to the grob *refp*.

`ly:grob-interfaces` *grob* [Fonction]
Return the interfaces list of grob *grob*.

`ly:grob-layout` *grob* [Fonction]
Get `\layout` definition from grob *grob*.

`ly:grob-object` *grob sym* [Fonction]
Return the value of a pointer in grob *grob* of property *sym*. It returns `'()` (end-of-list) if *sym* is undefined in *grob*.

`ly:grob-original` *grob* [Fonction]
Return the unbroken original grob of *grob*.

`ly:grob-parent` *grob axis* [Fonction]
Get the parent of *grob*. *axis* is 0 for the X-axis, 1 for the Y-axis.

`ly:grob-pq<?` *a b* [Fonction]
Compare two grob priority queue entries. This is an internal function.

`ly:grob-properties` *grob* [Fonction]
Get the mutable properties of *grob*.

`ly:grob-property` *grob sym val* [Fonction]
Return the value for property *sym* of *grob*. If no value is found, return *val* or `'()` if *val* is not specified.

`ly:grob-property-data` *grob sym* [Fonction]
Return the value for property *sym* of *grob*, but do not process callbacks.

`ly:grob-relative-coordinate` *grob refp axis* [Fonction]
Get the coordinate in *axis* direction of *grob* relative to the grob *refp*.

`ly:grob-robust-relative-extent` *grob refp axis* [Fonction]
Get the extent in *axis* direction of *grob* relative to the grob *refp*, or `(0,0)` if empty.

`ly:grob-script-priority-less` *a b* [Fonction]
Compare two grobs by script priority. For internal use.

`ly:grob-set-nested-property!` *grob symlist val* [Fonction]
Set nested property *symlist* in grob *grob* to value *val*.

`ly:grob-set-object!` *grob sym val* [Fonction]
Set *sym* in grob *grob* to value *val*.

`ly:grob-set-parent!` *grob axis parent-grob* [Fonction]
Set *parent-grob* the parent of grob *grob* in axis *axis*.

`ly:grob-set-property!` *grob sym val* [Fonction]
Set *sym* in grob *grob* to value *val*.

`ly:grob-staff-position` *sg* [Fonction]
Return the Y-position of *sg* relative to the staff.

`ly:grob-suicide!` *grob* [Fonction]
Kill *grob*.

`ly:grob-system` *grob* [Fonction]
Return the system grob of *grob*.

`ly:grob-translate-axis!` *grob d a* [Fonction]
Translate *grob* on axis *a* over distance *d*.

`ly:gulp-file` *name size* [Fonction]
Read *size* characters from the file *name*, and return its contents in a string. If *size* is undefined, the entire file is read. The file is looked up using the search path.

`ly:hash-table-keys` *tab* [Fonction]
Return a list of keys in *tab*.

`ly:inch` *num* [Fonction]
num inches.

`ly:input-both-locations` *sip* [Fonction]
Return input location in *sip* as `(file-name first-line first-column last-line last-column)`.

`ly:input-file-line-char-column` *sip* [Fonction]
Return input location in *sip* as `(file-name line char column)`.

`ly:input-location?` *x* [Fonction]
Is *x* an `input-location`?

`ly:input-message` *sip msg rest* [Fonction]
Print *msg* as a GNU compliant error message, pointing to the location in *sip*. *msg* is interpreted similar to `format`'s argument, using *rest*.

`ly:interpret-music-expression` *mus ctx* [Fonction]
Interpret the music expression *mus* in the global context *ctx*. The context is returned in its final state.

`ly:interpret-stencil-expression` *expr func arg1 offset* [Fonction]
Parse *expr*, feed bits to *func* with first arg *arg1* having offset *offset*.

`ly:intlog2` *d* [Fonction]
The 2-logarithm of $1/d$.

`ly:is-listened-event-class` *sym* [Fonction]
Is *sym* a listened event class?

`ly:item?` *g* [Fonction]
Is *g* an `Item` object?

`ly:item-break-dir` *it* [Fonction]
The break status direction of item *it*. `-1` means end of line, `0` unbroken, and `1` beginning of line.

`ly:iterator?` *x* [Fonction]
Is *x* a `Music_iterator` object?

`ly:lexer-keywords` *lexer* [Fonction]
Return a list of `(KEY . CODE)` pairs, signifying the LilyPond reserved words list.

`ly:lily-lexer?` *x* [Fonction]
Is *x* a `Lily_lexer` object?

`ly:lily-parser?` *x* [Fonction]
Is *x* a `Lily_parser` object?

`ly:listener?` *x* [Fonction]
Is *x* a `Listener` object?

`ly:make-book` *paper header scores* [Fonction]
Make a `\book` of *paper* and *header* (which may be `#f` as well) containing `\scores`.

`ly:make-book-part` *scores* [Fonction]
Make a `\bookpart` containing `\scores`.

`ly:make-dispatcher` [Fonction]
Return a newly created dispatcher.

`ly:make-duration` *length dotcount num den* [Fonction]
length is the negative logarithm (base 2) of the duration: 1 is a half note, 2 is a quarter note, 3 is an eighth note, etc. The number of dots after the note is given by the optional argument *dotcount*.

The duration factor is optionally given by *num* and *den*.

A duration is a musical duration, i.e., a length of time described by a power of two (whole, half, quarter, etc.) and a number of augmentation dots.

`ly:make-global-context` *output-def* [Fonction]
Set up a global interpretation context, using the output block *output-def*. The context is returned.

`ly:make-global-translator` *global* [Fonction]
Create a translator group and connect it to the global context *global*. The translator group is returned.

`ly:make-listener` *callback* [Fonction]
Create a listener. Any time the listener hears an object, it will call *callback* with that object.

callback should take exactly one argument.

`ly:make-moment` *n d gn gd* [Fonction]
Create the rational number with main timing n/d, and optional grace timing gn/gd.

A *moment* is a point in musical time. It consists of a pair of rationals (m, g), where m is the timing for the main notes, and g the timing for grace notes. In absence of grace notes, g is zero.

`ly:make-music` *props* [Fonction]
Make a C++ `Music` object and initialize it with *props*.

This function is for internal use and is only called by `make-music`, which is the preferred interface for creating music objects.

`ly:make-music-function` *signature func* [Fonction]
Make a function to process music, to be used for the parser. *func* is the function, and *signature* describes its arguments. *signature* is a list containing either `ly:music?` predicates or other type predicates.

`ly:make-output-def` [Fonction]
Make an output definition.

`ly:make-page-label-marker` *label* [Fonction]
Return page marker with label *label*.

`ly:make-page-permission-marker` *symbol permission* [Fonction]
Return page marker with page breaking and turning permissions.

`ly:make-pango-description-string` *chain size* [Fonction]
Make a `PangoFontDescription` string for the property alist *chain* at size *size*.

`ly:make-paper-outputter` *port format* [Fonction]
Create an outputter that evaluates within `output-`*format*, writing to *port*.

`ly:make-pitch` *octave note alter* [Fonction]
octave is specified by an integer, zero for the octave containing middle C. *note* is a number indexing the global default scale, with 0 corresponding to pitch C and 6 usually corresponding to pitch B. *alter* is a rational number of 200-cent whole tones for alteration.

`ly:make-prob` *type init rest* [Fonction]
Create a `Prob` object.

`ly:make-scale` *steps* [Fonction]
Create a scale. The argument is a vector of rational numbers, each of which represents the number of 200 cent tones of a pitch above the tonic.

`ly:make-score` *music* [Fonction]
Return score with *music* encapsulated in it.

`ly:make-simple-closure` *expr* [Fonction]
Make a simple closure. *expr* should be form of `(func a1 a2 ...)`, and will be invoked as `(func delayed-arg a1 a2 ...)`.

`ly:make-stencil` *expr xext yext* [Fonction]
Stencils are device independent output expressions. They carry two pieces of information:

1. A specification of how to print this object. This specification is processed by the output backends, for example '`scm/output-ps.scm`'.
2. The vertical and horizontal extents of the object, given as pairs. If an extent is unspecified (or if you use `(1000 . -1000)` as its value), it is taken to be empty.

`ly:make-stream-event` *cl proplist* [Fonction]
Create a stream event of class *cl* with the given mutable property list.

`ly:message` *str rest* [Fonction]
A Scheme callable function to issue the message *str*. The message is formatted with `format` and *rest*.

`ly:minimal-breaking` *pb* [Fonction]
Break (pages and lines) the `Paper_book` object *pb* without looking for optimal spacing: stack as many lines on a page before moving to the next one.

`ly:mm` *num* [Fonction]
num mm.

`ly:module->alist` *mod* [Fonction]
Dump the contents of module *mod* as an alist.

`ly:module-copy` *dest src* [Fonction]
Copy all bindings from module *src* into *dest*.

`ly:modules-lookup` *modules sym def* [Fonction]
Look up *sym* in the list *modules*, returning the first occurence. If not found, return *def* or `#f` if *def* isn't specified.

`ly:moment?` *x* [Fonction]
Is *x* a `Moment` object?

`ly:moment<?` *a b* [Fonction]
Compare two moments.

`ly:moment-add` *a b* [Fonction]
Add two moments.

`ly:moment-div` *a b* [Fonction]
Divide two moments.

`ly:moment-grace-denominator` *mom* [Fonction]
Extract denominator from grace timing.

`ly:moment-grace-numerator` *mom* [Fonction]
Extract numerator from grace timing.

`ly:moment-main-denominator` *mom* [Fonction]
Extract denominator from main timing.

`ly:moment-main-numerator` *mom* [Fonction]
Extract numerator from main timing.

`ly:moment-mod` *a b* [Fonction]
Modulo of two moments.

`ly:moment-mul` *a b* [Fonction]
Multiply two moments.

`ly:moment-sub` *a b* [Fonction]
Subtract two moments.

`ly:music?` *obj* [Fonction]
Is *obj* a music object?

`ly:music-compress` *m factor* [Fonction]
Compress music object *m* by moment *factor*.

`ly:music-deep-copy` *m* [Fonction]
Copy *m* and all sub expressions of *m*.

`ly:music-duration-compress` *mus fact* [Fonction]
Compress *mus* by factor *fact*, which is a `Moment`.

`ly:music-duration-length` *mus* [Fonction]
Extract the duration field from *mus* and return the length.

`ly:music-function?` *x* [Fonction]
Is *x* a `music-function`?

`ly:music-function-extract` *x* [Fonction]
Return the Scheme function inside *x*.

`ly:music-length` *mus* [Fonction]
Get the length of music expression *mus* and return it as a `Moment` object.

`ly:music-list?` *lst* [Fonction]
Is *lst* a list of music objects?

`ly:music-mutable-properties` *mus* [Fonction]
Return an alist containing the mutable properties of *mus*. The immutable properties are not available, since they are constant and initialized by the `make-music` function.

`ly:music-output?` *x* [Fonction]
Is *x* a `Music_output` object?

`ly:music-property` *mus sym val* [Fonction]
Return the value for property *sym* of music expression *mus*. If no value is found, return *val* or `'()` if *val* is not specified.

`ly:music-set-property!` *mus sym val* [Fonction]
Set property *sym* in music expression *mus* to *val*.

`ly:music-transpose` *m p* [Fonction]
Transpose *m* such that central C is mapped to *p*. Return *m*.

`ly:note-column-accidentals` *note-column* [Fonction]
Return the `AccidentalPlacement` grob from *note-column* if any, or `SCM_EOL` otherwise.

`ly:note-column-dot-column` *note-column* [Fonction]
Return the `DotColumn` grob from *note-column* if any, or `SCM_EOL` otherwise.

`ly:note-head::stem-attachment` *font-metric glyph-name* [Fonction]
Get attachment in *font-metric* for attaching a stem to notehead *glyph-name*.

`ly:number->string` *s* [Fonction]
Convert *s* to a string without generating many decimals.

`ly:optimal-breaking` *pb* [Fonction]
Optimally break (pages and lines) the `Paper_book` object *pb* to minimize badness in bother vertical and horizontal spacing.

`ly:option-usage` [Fonction]
Print `ly:set-option` usage.

`ly:otf->cff` *otf-file-name* [Fonction]
Convert the contents of an OTF file to a CFF file, returning it as a string.

`ly:otf-font?` *font* [Fonction]
Is *font* an OpenType font?

`ly:otf-font-glyph-info` *font glyph* [Fonction]
Given the font metric *font* of an OpenType font, return the information about named glyph *glyph* (a string).

`ly:otf-font-table-data` *font tag* [Fonction]
Extract a table *tag* from *font*. Return empty string for non-existent *tag*.

`ly:otf-glyph-count` *font* [Fonction]
Return the number of glyphs in *font*.

`ly:otf-glyph-list` *font* [Fonction]
Return a list of glyph names for *font*.

`ly:output-def?` *def* [Fonction]
Is *def* an output definition?

`ly:output-def-clone` *def* [Fonction]
Clone output definition *def*.

`ly:output-def-lookup` *def sym val* [Fonction]
Return the value of *sym* in output definition *def* (e.g., `\paper`). If no value is found, return *val* or `'()` if *val* is undefined.

`ly:output-def-parent` *def* [Fonction]
Return the parent output definition of *def*.

`ly:output-def-scope` *def* [Fonction]
Return the variable scope inside *def*.

`ly:output-def-set-variable!` *def sym val* [Fonction]
Set an output definition *def* variable *sym* to *val*.

`ly:output-description` *output-def* [Fonction]
Return the description of translators in *output-def*.

`ly:output-formats` [Fonction]
Formats passed to '`--format`' as a list of strings, used for the output.

`ly:outputter-close` *outputter* [Fonction]
Close port of *outputter*.

`ly:outputter-dump-stencil` *outputter stencil* [Fonction]
Dump stencil *expr* onto *outputter*.

`ly:outputter-dump-string` *outputter str* [Fonction]
Dump *str* onto *outputter*.

`ly:outputter-module` *outputter* [Fonction]
Return output module of *outputter*.

`ly:outputter-output-scheme` *outputter expr* [Fonction]
Eval *expr* in module of *outputter*.

`ly:outputter-port` *outputter* [Fonction]
Return output port for *outputter*.

`ly:page-marker?` *x* [Fonction]
Is *x* a `Page_marker` object?

`ly:page-turn-breaking` *pb* [Fonction]
Optimally break (pages and lines) the `Paper_book` object *pb* such that page turns only happen in specified places, returning its pages.

`ly:pango-font?` *f* [Fonction]
Is *f* a pango font?

`ly:pango-font-physical-fonts` *f* [Fonction]
Return alist of `(ps-name file-name font-index)` lists for Pango font *f*.

`ly:paper-book?` *x* [Fonction]
Is *x* a `Paper_book` object?

`ly:paper-book-header` *pb* [Fonction]
Return the header definition (`\header`) in `Paper_book` object *pb*.

`ly:paper-book-pages` *pb* [Fonction]
Return pages in `Paper_book` object *pb*.

`ly:paper-book-paper` *pb* [Fonction]
Return the paper output definition (`\paper`) in `Paper_book` object *pb*.

`ly:paper-book-performances` *pb* [Fonction]
Return performances in `Paper_book` object *pb*.

`ly:paper-book-scopes` *pb* [Fonction]
Return scopes in `Paper_book` object *pb*.

`ly:paper-book-systems` *pb* [Fonction]
Return systems in `Paper_book` object *pb*.

`ly:paper-fonts` *def* [Fonction]
Return a list containing the fonts from output definition *def* (e.g., `\paper`).

`ly:paper-get-font` *def chain* [Fonction]
Find a font metric in output definition *def* satisfying the font-qualifiers in alist chain *chain*, and return it. (An alist chain is a list of alists, containing grob properties.)

`ly:paper-get-number` *def sym* [Fonction]
Return the value of variable *sym* in output definition *def* as a double.

`ly:paper-outputscale` *def* [Fonction]
Return the output-scale for output definition *def*.

`ly:paper-score-paper-systems` *paper-score* [Fonction]
Return vector of `paper_system` objects from *paper-score*.

`ly:paper-system?` *obj* [Fonction]
Is *obj* a C++ `Prob` object of type `paper-system`?

`ly:paper-system-minimum-distance` *sys1 sys2* [Fonction]
Measure the minimum distance between these two paper-systems, using their stored skylines if possible and falling back to their extents otherwise.

`ly:parse-file` *name* [Fonction]
Parse a single `.ly` file. Upon failure, throw `ly-file-failed` key.

`ly:parser-clear-error` *parser* [Fonction]
Clear the error flag for the parser.

`ly:parser-clone` *parser-smob* [Fonction]
Return a clone of *parser-smob*.

`ly:parser-define!` *parser-smob symbol val* [Fonction]
Bind *symbol* to *val* in *parser-smob*'s module.

`ly:parser-error` *parser msg input* [Fonction]
Display an error message and make the parser fail.

`ly:parser-has-error?` *parser* [Fonction]
Does *parser* have an error flag?

`ly:parser-include-string` *parser-smob ly-code* [Fonction]
Include the string *ly-code* into the input stream for *parser-smob*.

`ly:parser-lexer` *parser-smob* [Fonction]
Return the lexer for *parser-smob*.

`ly:parser-lookup` *parser-smob symbol* [Fonction]
Look up *symbol* in *parser-smob*'s module. Return `'()` if not defined.

`ly:parser-output-name` *parser* [Fonction]
Return the base name of the output file.

`ly:parser-parse-string` *parser-smob ly-code* [Fonction]
Parse the string *ly-code* with *parser-smob*. Upon failure, throw `ly-file-failed` key.

`ly:parser-set-note-names` *parser names* [Fonction]
Replace current note names in *parser*. *names* is an alist of symbols. This only has effect if the current mode is notes.

`ly:parser-set-repetition-function` *parser fun* [Fonction]
Replace the current repetition function in *parser*. *fun* is the new repetition function.

`ly:parser-set-repetition-symbol` *parser sym* [Fonction]
Replace the current repetition symbol in *parser*. *sym* is the new repetition symbol.

`ly:performance-write` *performance filename* [Fonction]
Write *performance* to *filename*.

`ly:pfb->pfa` *pfb-file-name* [Fonction]
Convert the contents of a Type 1 font in PFB format to PFA format.

`ly:pitch?` *x* [Fonction]
Is *x* a `Pitch` object?

`ly:pitch<?` *p1 p2* [Fonction]
Is *p1* lexicographically smaller than *p2*?

`ly:pitch-alteration` *pp* [Fonction]
Extract the alteration from pitch *pp*.

`ly:pitch-diff` *pitch root* [Fonction]
Return pitch *delta* such that *pitch* transposed by *delta* equals *root*.

`ly:pitch-negate` *p* [Fonction]
Negate *p*.

`ly:pitch-notename` *pp* [Fonction]
Extract the note name from pitch *pp*.

`ly:pitch-octave` *pp* [Fonction]
Extract the octave from pitch *pp*.

`ly:pitch-quartertones` *pp* [Fonction]
Calculate the number of quarter tones of *pp* from middle C.

`ly:pitch-semitones` *pp* [Fonction]
Calculate the number of semitones of *pp* from middle C.

`ly:pitch-steps` *p* [Fonction]
Number of steps counted from middle C of the pitch *p*.

`ly:pitch-transpose` *p delta* [Fonction]
Transpose *p* by the amount *delta*, where *delta* is relative to middle C.

`ly:pointer-group-interface::add-grob` *grob sym grob-element* [Fonction]
Add *grob-element* to *grob*'s *sym* grob array.

`ly:position-on-line?` *sg spos* [Fonction]
Return whether *spos* is on a line of the staff associated with the grob *sg* (even on an extender line).

`ly:prob?` *x* [Fonction]
Is *x* a `Prob` object?

`ly:prob-immutable-properties` *prob* [Fonction]
Retrieve an alist of immutable properties.

`ly:prob-mutable-properties` *prob* [Fonction]
Retrieve an alist of mutable properties.

`ly:prob-property` *prob sym val* [Fonction]
Return the value for property *sym* of Prob object *prob*. If no value is found, return *val* or `'()` if *val* is not specified.

`ly:prob-property?` *obj sym* [Fonction]
Is boolean prop *sym* of *sym* set?

`ly:prob-set-property!` *obj sym value* [Fonction]
Set property *sym* of *obj* to *value*.

`ly:prob-type?` *obj type* [Fonction]
Is *obj* the specified prob-type?

`ly:programming-error` *str rest* [Fonction]
A Scheme callable function to issue the internal warning *str*. The message is formatted with `format` and *rest*.

`ly:progress` *str rest* [Fonction]
A Scheme callable function to print progress *str*. The message is formatted with `format` and *rest*.

`ly:property-lookup-stats` *sym* [Fonction]
Return hash table with a property access corresponding to *sym*. Choices are `prob`, `grob`, and `context`.

`ly:protects` [Fonction]
Return hash of protected objects.

`ly:pt` *num* [Fonction]
num printer points.

`ly:register-stencil-expression` *symbol* [Fonction]
Add *symbol* as head of a stencil expression.

`ly:relative-group-extent` *elements common axis* [Fonction]
Determine the extent of *elements* relative to *common* in the *axis* direction.

`ly:reset-all-fonts` [Fonction]
Forget all about previously loaded fonts.

`ly:round-filled-box` *xext yext blot* [Fonction]
Make a `Stencil` object that prints a black box of dimensions *xext*, *yext* and roundness *blot*.

`ly:round-filled-polygon` *points blot* [Fonction]
Make a `Stencil` object that prints a black polygon with corners at the points defined by *points* (list of coordinate pairs) and roundness *blot*.

`ly:run-translator` *mus output-def* [Fonction]
Process *mus* according to *output-def*. An interpretation context is set up, and *mus* is interpreted with it. The context is returned in its final state.

Optionally, this routine takes an object-key to to uniquely identify the score block containing it.

`ly:score?` *x* [Fonction]
Is *x* a `Score` object?

`ly:score-add-output-def!` *score def* [Fonction]
Add an output definition *def* to *score*.

`ly:score-embedded-format` *score layout* [Fonction]
Run *score* through *layout* (an output definition) scaled to correct output-scale already, returning a list of layout-lines.

`ly:score-error?` *score* [Fonction]
Was there an error in the score?

`ly:score-header` *score* [Fonction]
Return score header.

`ly:score-music` *score* [Fonction]
Return score music.

`ly:score-output-defs` *score* [Fonction]
All output definitions in a score.

`ly:score-set-header!` *score module* [Fonction]
Set the score header.

`ly:set-default-scale` *scale* [Fonction]
Set the global default scale. This determines the tuning of pitches with no accidentals or key signatures. The first pitch is C. Alterations are calculated relative to this scale. The number of pitches in this scale determines the number of scale steps that make up an octave. Usually the 7-note major scale.

`ly:set-grob-creation-callback` *cb* [Fonction]
Specify a procedure that will be called every time a new grob is created. The callback will receive as arguments the grob that was created, the name of the C++ source file that caused the grob to be created, and the corresponding line number in the C++ source file.

`ly:set-grob-modification-callback` *cb* [Fonction]
Specify a procedure that will be called every time LilyPond modifies a grob property. The callback will receive as arguments the grob that is being modified, the name of the C++ file in which the modification was requested, the line number in the C++ file in which the modification was requested, the name of the function in which the modification was requested, the property to be changed, and the new value for the property.

`ly:set-middle-C!` *context* [Fonction]
Set the `middleCPosition` variable in *context* based on the variables `middleCClefPosition` and middleCOffset.

`ly:set-option` *var val* [Fonction]
Set a program option.

`ly:set-property-cache-callback` *cb* [Fonction]
Specify a procedure that will be called whenever lilypond calculates a callback function and caches the result. The callback will receive as arguments the grob whose property it is, the name of the property, the name of the callback that calculated the property, and the new (cached) value of the property.

`ly:simple-closure?` *clos* [Fonction]
Is *clos* a simple closure?

`ly:skyline?` *x* [Fonction]
Is *x* a `Skyline` object?

`ly:skyline-pair?` *x* [Fonction]
Is *x* a `Skyline_pair` object?

`ly:slur-score-count` [Fonction]
count number of slur scores.

`ly:smob-protects` [Fonction]
Return LilyPond's internal smob protection list.

`ly:solve-spring-rod-problem` *springs rods length ragged* [Fonction]
Solve a spring and rod problem for *count* objects, that are connected by *count*-1 *springs*, and an arbitrary number of *rods*. *count* is implicitly given by *springs* and *rods*. The *springs* argument has the format `(ideal, inverse_hook)` and *rods* is of the form `(idx1, idx2, distance)`.

length is a number, *ragged* a boolean.

The function returns a list containing the force (positive for stretching, negative for compressing and `#f` for non-satisfied constraints) followed by *spring-count*+1 positions of the objects.

`ly:source-file?` *x* [Fonction]
Is *x* a `Source_file` object?

`ly:spanner?` *g* [Fonction]
Is *g* a spanner object?

`ly:spanner-bound` *spanner dir* [Fonction]
Get one of the bounds of *spanner*. *dir* is `-1` for left, and `1` for right.

`ly:spanner-broken-into` *spanner* [Fonction]
Return broken-into list for *spanner*.

`ly:spanner-set-bound!` *spanner dir item* [Fonction]
Set grob *item* as bound in direction *dir* for *spanner*.

`ly:spawn` *command rest* [Fonction]
Simple interface to g_spawn_sync *str*. The error is formatted with `format` and *rest*.

`ly:staff-symbol-line-thickness` *grob* [Fonction]
Returns the `line-thickness` of the staff associated with *grob*.

`ly:staff-symbol-staff-space` *grob* [Fonction]
Returns the `staff-space` of the staff associated with *grob*.

`ly:start-environment` [Fonction]
Return the environment (a list of strings) that was in effect at program start.

`ly:stderr-redirect` *file-name mode* [Fonction]
Redirect stderr to *file-name*, opened with *mode*.

`ly:stencil?` *x* [Fonction]
Is *x* a `Stencil` object?

`ly:stencil-add` *args* [Fonction]
Combine stencils. Takes any number of arguments.

`ly:stencil-aligned-to` *stil axis dir* [Fonction]
Align *stil* using its own extents. *dir* is a number. `-1` and `1` are left and right, respectively. Other values are interpolated (so `0` means the center).

`ly:stencil-combine-at-edge` *first axis direction second padding minimum* [Fonction]
Construct a stencil by putting *second* next to *first*. *axis* can be 0 (x-axis) or 1 (y-axis). *direction* can be -1 (left or down) or 1 (right or up). The stencils are juxtaposed with *padding* as extra space. If this puts the reference points closer than *minimum*, they are moved by the latter amount. *first* and *second* may also be `'()` or `#f`.

`ly:stencil-empty?` *stil* [Fonction]
Return whether *stil* is empty.

`ly:stencil-expr` *stil* [Fonction]
Return the expression of *stil*.

`ly:stencil-extent` *stil axis* [Fonction]
Return a pair of numbers signifying the extent of *stil* in *axis* direction (`0` or `1` for x and y axis, respectively).

`ly:stencil-fonts` *s* [Fonction]
Analyze *s*, and return a list of fonts used in *s*.

`ly:stencil-in-color` *stc r g b* [Fonction]
Put *stc* in a different color.

`ly:stencil-rotate` *stil angle x y* [Fonction]
Return a stencil *stil* rotated *angle* degrees around the relative offset (*x*, *y*). E.g. an offset of (-1, 1) will rotate the stencil around the left upper corner.

`ly:stencil-rotate-absolute` *stil angle x y* [Fonction]
Return a stencil *stil* rotated *angle* degrees around point (*x*, *y*), given in absolute coordinates.

`ly:stencil-scale` *stil x y* [Fonction]
Scale *stil* using the horizontal and vertical scaling factors *x* and *y*.

`ly:stencil-translate` *stil offset* [Fonction]
Return a *stil*, but translated by *offset* (a pair of numbers).

`ly:stencil-translate-axis` *stil amount axis* [Fonction]
Return a copy of *stil* but translated by *amount* in *axis* direction.

`ly:stream-event?` *obj* [Fonction]
Is *obj* a `Stream_event` object?

`ly:string-percent-encode` *str* [Fonction]
Encode all characters in string *str* with hexadecimal percent escape sequences, with the following exceptions: characters `-`, `.`, `/`, and `_`; and characters in ranges `0-9`, `A-Z`, and `a-z`.

`ly:string-substitute` *a b s* [Fonction]
Replace string *a* by string *b* in string *s*.

`ly:success` *str rest* [Fonction]
A Scheme callable function to issue a success message *str*. The message is formatted with `format` and *rest*.

`ly:system-font-load` *name* [Fonction]
Load the OpenType system font '*name*`.otf`'. Fonts loaded with this command must contain three additional SFNT font tables called `LILC`, `LILF`, and `LILY`, needed for typesetting musical elements. Currently, only the Emmentaler and the Emmentaler-Brace fonts fulfill these requirements.

Note that only `ly:font-get-glyph` and derived code (like `\lookup`) can access glyphs from the system fonts; text strings are handled exclusively via the Pango interface.

`ly:text-interface::interpret-markup` [Fonction]
Convert a text markup into a stencil. Takes three arguments, *layout*, *props*, and *markup*.

layout is a `\layout` block; it may be obtained from a grob with `ly:grob-layout`. *props* is an alist chain, i.e. a list of alists. This is typically obtained with `(ly:grob-alist-chain grob (ly:output-def-lookup layout 'text-font-defaults))`. *markup* is the markup text to be processed.

`ly:translator?` *x* [Fonction]
Is *x* a `Translator` object?

`ly:translator-context` *trans* [Fonction]
Return the context of the translator object *trans*.

`ly:translator-description` *me* [Fonction]
Return an alist of properties of translator *me*.

`ly:translator-group?` *x* [Fonction]
Is *x* a `Translator_group` object?

`ly:translator-name` *trans* [Fonction]
Return the type name of the translator object *trans*. The name is a symbol.

`ly:transpose-key-alist` *l pit* [Fonction]
Make a new key alist of *l* transposed by pitch *pit*.

`ly:truncate-list!` *lst i* [Fonction]
Take at most the first *i* of list *lst*.

`ly:ttf->pfa` *ttf-file-name idx* [Fonction]
Convert the contents of a TrueType font file to PostScript Type 42 font, returning it as a string. The optional *idx* argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of *idx* is 0.

`ly:ttf-ps-name` *ttf-file-name idx* [Fonction]
Extract the PostScript name from a TrueType font. The optional *idx* argument is useful for TrueType collections (TTC) only; it specifies the font index within the TTC. The default value of *idx* is 0.

`ly:unit` [Fonction]
Return the unit used for lengths as a string.
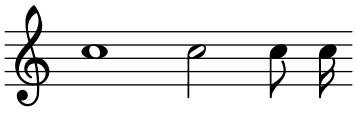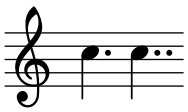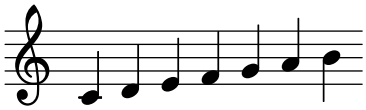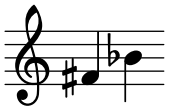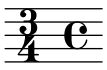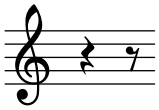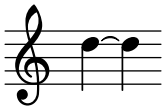
`ly:usage` [Fonction]
Print usage message.
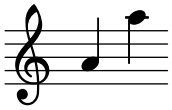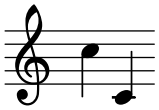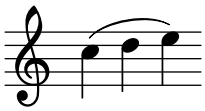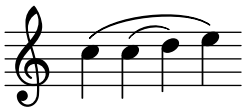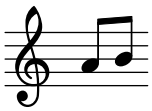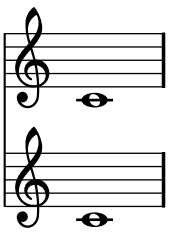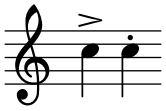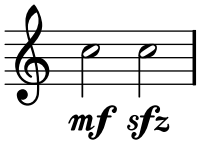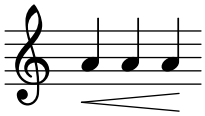
`ly:version` [Fonction]
Return the current lilypond version as a list, e.g., `(1 3 127 uu1)`.
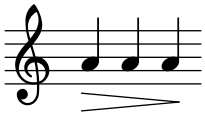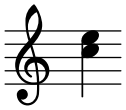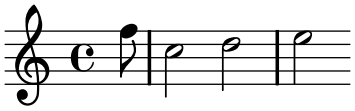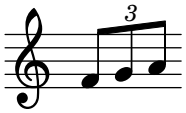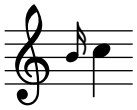
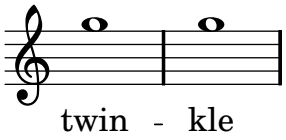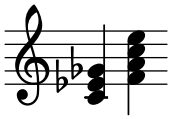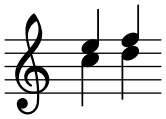
`ly:warning` *str rest* [Fonction]
A Scheme callable function to issue the warning *str*. The message is formatted with `format` and *rest*.

`ly:wide-char->utf-8` *wc* [Fonction]
Encode the Unicode codepoint *wc*, an integer, as UTF-8.

Annexe B Aide-mémoire

Syntaxe	Description	Exemple
`1 2 8 16`	valeurs rythmiques	
`c4. c4..`	notes pointées	
`c d e f g a b`	gamme	
`fis bes`	altérations	
`\clef treble \clef bass`	clés	
`\time 3/4 \time 4/4`	chiffre de mesure, métrique	
`r4 r8`	silences	
`d ~ d`	liaison de tenue	
`\key es \major`	armure	

note`'`	monter d'une octave	
note`,`	baisser d'une octave	
`c( d e)`	liaisons	
`c\( c( d) e\)`	liaisons de phrasé	
`a8[ b]`	ligatures	
`<< \new Staff ... >>`	ajouter des portées	
`c-> c-.`	indications d'articulation	
`c2\mf c\sfz`	nuances	
`a\< a a\!`	crescendo	

`a\> a a\!`	decrescendo	
`< >`	accords	
`\partial 8`	levées, anacrouses	
`\times 2/3 {f g a}`	triolets	
`\grace`	appogiatures	
`\lyricmode { twinkle }`	ajouter des paroles	twinkle
`\new Lyrics`	imprimer les paroles	**twinkle**
`twin -- kle`	diviser un mot en plusieurs syllabes	
`\chordmode { c:dim f:maj7 }`	accords chiffrés	
`\context ChordNames`	imprimer les chiffrages d'accords	C° F△
`<<{e f} \\ {c d}>>`	polyphonie	

`s4 s8 s16`	silences invisibles

Annexe C Grammaire de LilyPond

Cette annexe présente la grammaire du langage de LilyPond, telle qu'elle ressort de l'analyseur syntaxique.

Grammar

```
lilypond: /* empty */
        | lilypond toplevel_expression
        | lilypond assignment
        | lilypond error
        | lilypond "\invalid"

toplevel_expression: lilypond_header
                   | book_block
                   | bookpart_block
                   | score_block
                   | composite_music
                   | full_markup
                   | full_markup_list
                   | output_def

embedded_scm: SCM_TOKEN
            | SCM_IDENTIFIER

lilypond_header_body: /* empty */
                    | lilypond_header_body assignment

lilypond_header: "\header" '{' lilypond_header_body '}'

assignment_id: STRING
             | LYRICS_STRING

assignment: assignment_id '=' identifier_init
          | assignment_id property_path '=' identifier_init
          | embedded_scm

identifier_init: score_block
               | book_block
               | bookpart_block
               | output_def
               | context_def_spec_block
               | music
               | post_event
               | number_expression
               | string
               | embedded_scm
               | full_markup
               | full_markup_list
               | DIGIT
               | context_modification

context_def_spec_block: "\context" '{' context_def_spec_body '}'
```

```
39 context_def_spec_body: /* empty */
40                      | CONTEXT_DEF_IDENTIFIER
41                      | context_def_spec_body
                            "\grobdescriptions"
                            embedded_scm
42                      | context_def_spec_body context_mod
43                      | context_def_spec_body context_modification

44 book_block: "\book" '{' book_body '}'

45 book_body: /* empty */
46          | BOOK_IDENTIFIER
47          | book_body paper_block
48          | book_body bookpart_block
49          | book_body score_block
50          | book_body composite_music
51          | book_body full_markup
52          | book_body full_markup_list
53          | book_body lilypond_header
54          | book_body error

55 bookpart_block: "\bookpart" '{' bookpart_body '}'

56 bookpart_body: /* empty */
57              | BOOK_IDENTIFIER
58              | bookpart_body paper_block
59              | bookpart_body score_block
60              | bookpart_body composite_music
61              | bookpart_body full_markup
62              | bookpart_body full_markup_list
63              | bookpart_body lilypond_header
64              | bookpart_body error

65 score_block: "\score" '{' score_body '}'

66 score_body: music
67           | SCORE_IDENTIFIER
68           | score_body lilypond_header
69           | score_body output_def
70           | score_body error

71 paper_block: output_def

72 output_def: output_def_body '}'

73 output_def_head: "\paper"
74                | "\midi"
75                | "\layout"

76 output_def_head_with_mode_switch: output_def_head
```

```
 77 output_def_body: output_def_head_with_mode_switch '{'
 78                | output_def_head_with_mode_switch
                       '{'
                       OUTPUT_DEF_IDENTIFIER
 79                | output_def_body assignment
 80                | output_def_body context_def_spec_block
 81                | output_def_body error

 82 tempo_event: "\tempo" steno_duration '=' tempo_range
 83            | "\tempo" scalar steno_duration '=' tempo_range
 84            | "\tempo" scalar

 85 music_list: /* empty */
 86           | music_list music
 87           | music_list embedded_scm
 88           | music_list error

 89 music: simple_music
 90      | composite_music

 91 alternative_music: /* empty */
 92                  | "\alternative" '{' music_list '}'

 93 repeated_music: "\repeat"
                     simple_string
                     unsigned_number
                     music
                     alternative_music

 94 sequential_music: "\sequential" '{' music_list '}'
 95                 | '{' music_list '}'

 96 simultaneous_music: "\simultaneous" '{' music_list '}'
 97                   | "<<" music_list ">>"

 98 simple_music: event_chord
 99             | MUSIC_IDENTIFIER
100             | music_property_def
101             | context_change

103 context_modification: "\with" $ '{' context_mod_list '}'
104                     | "\with" CONTEXT_MOD_IDENTIFIER
105                     | CONTEXT_MOD_IDENTIFIER

106 optional_context_mod: /* empty */
107                     | context_modification

108 context_mod_list: /* empty */
109                 | context_mod_list context_mod
110                 | context_mod_list CONTEXT_MOD_IDENTIFIER

111 composite_music: prefix_composite_music
```

```
112                | grouped_music_list

113 grouped_music_list: simultaneous_music
114                   | sequential_music

115 function_scm_argument: embedded_scm
116                      | simple_string

117 function_arglist_music_last: EXPECT_MUSIC function_arglist music

118 function_arglist_nonmusic_last: EXPECT_MARKUP
                                      function_arglist
                                      full_markup
119                                 | EXPECT_MARKUP
                                      function_arglist
                                      simple_string
120                                 | EXPECT_SCM
                                      function_arglist
                                      function_scm_argument

121 function_arglist_nonmusic: EXPECT_NO_MORE_ARGS
122                            | EXPECT_MARKUP
                                 function_arglist_nonmusic
                                 full_markup
123                            | EXPECT_MARKUP
                                 function_arglist_nonmusic
                                 simple_string
124                            | EXPECT_SCM
                                 function_arglist_nonmusic
                                 function_scm_argument

125 function_arglist: EXPECT_NO_MORE_ARGS
126                 | function_arglist_music_last
127                 | function_arglist_nonmusic_last

128 generic_prefix_music_scm: MUSIC_FUNCTION function_arglist

129 optional_id: /* empty */
130            | '=' simple_string

131 prefix_composite_music: generic_prefix_music_scm
132                       | "\context"
                            simple_string
                            optional_id
                            optional_context_mod
                            music
133                       | "\new"
                            simple_string
                            optional_id
                            optional_context_mod
                            music
134                       | "\times" fraction music
```

```
135                       | repeated_music
136                       | "\transpose"
                              pitch_also_in_chords
                              pitch_also_in_chords
                              music
137                       | mode_changing_head grouped_music_list
138                       | mode_changing_head_with_context
                              optional_context_mod
                              grouped_music_list
139                       | relative_music
140                       | re_rhythmed_music

141 mode_changing_head: "\notemode"
142                   | "\drummode"
143                   | "\figuremode"
144                   | "\chordmode"
145                   | "\lyricmode"

146 mode_changing_head_with_context: "\drums"
147                                 | "\figures"
148                                 | "\chords"
149                                 | "\lyrics"

150 relative_music: "\relative" absolute_pitch music
151               | "\relative" composite_music

153 new_lyrics: "\addlyrics" $ grouped_music_list

155 new_lyrics: "\addlyrics" $ MUSIC_IDENTIFIER

157 new_lyrics: new_lyrics "\addlyrics" $ grouped_music_list

159 new_lyrics: new_lyrics "\addlyrics" $ MUSIC_IDENTIFIER

160 re_rhythmed_music: grouped_music_list new_lyrics
161                 | MUSIC_IDENTIFIER new_lyrics

163 re_rhythmed_music: "\lyricsto" simple_string $ music

164 context_change: "\change" STRING '=' STRING

165 property_path_revved: embedded_scm
166                     | property_path_revved embedded_scm

167 property_path: property_path_revved

168 property_operation: STRING '=' scalar
169                   | "\unset" simple_string
170                   | "\override" simple_string property_path '=' scalar
171                   | "\revert" simple_string embedded_scm

172 context_def_mod: "\consists"
```

```
173                | "\remove"
174                | "\accepts"
175                | "\defaultchild"
176                | "\denies"
177                | "\alias"
178                | "\type"
179                | "\description"
180                | "\name"

181 context_mod: property_operation
182             | context_def_mod STRING
183             | context_def_mod embedded_scm

184 context_prop_spec: simple_string
185                   | simple_string '.' simple_string

186 simple_music_property_def: "\override"
                                context_prop_spec
                                property_path
                                '='
                                scalar
187                          | "\revert" context_prop_spec embedded_scm
188                          | "\set" context_prop_spec '=' scalar
189                          | "\unset" context_prop_spec

190 music_property_def: simple_music_property_def
191                    | "\once" simple_music_property_def

192 string: STRING
193       | STRING_IDENTIFIER
194       | string '+' string

195 simple_string: STRING
196               | LYRICS_STRING
197               | STRING_IDENTIFIER

198 scalar: string
199       | LYRICS_STRING
200       | bare_number
201       | embedded_scm
202       | full_markup
203       | DIGIT

204 event_chord: simple_chord_elements post_events
205             | CHORD_REPETITION optional_notemode_duration post_events
206             | MULTI_MEASURE_REST optional_notemode_duration post_events
207             | command_element
208             | note_chord_element

209 note_chord_element: chord_body optional_notemode_duration post_events

210 chord_body: "<" chord_body_elements ">"
```

```
211 chord_body_elements: /* empty */
212                    | chord_body_elements chord_body_element

213 chord_body_element: pitch
                          exclamations
                          questions
                          octave_check
                          post_events
214                   | DRUM_PITCH post_events
215                   | music_function_chord_body

216 music_function_identifier_musicless_prefix: MUSIC_FUNCTION

217 music_function_chord_body: music_function_identifier_musicless_prefix
                                 EXPECT_MUSIC
                                 function_arglist_nonmusic
                                 chord_body_element
218                            | music_function_identifier_musicless_prefix
                                 function_arglist_nonmusic

219 music_function_event: music_function_identifier_musicless_prefix
                            EXPECT_MUSIC
                            function_arglist_nonmusic
                            post_event
220                       | music_function_identifier_musicless_prefix
                            function_arglist_nonmusic

221 command_element: command_event
222                | "\skip" duration_length
223                | "\["
224                | "\]"
225                | "\"
226                | '|'
227                | "\partial" duration_length
228                | "\time" fraction
229                | "\mark" scalar

230 command_event: "\~"
231              | "\mark" "\default"
232              | tempo_event
233              | "\key" "\default"
234              | "\key" NOTENAME_PITCH SCM_IDENTIFIER

235 post_events: /* empty */
236            | post_events post_event

237 post_event: direction_less_event
238           | script_dir music_function_event
239           | "--"
240           | "__"
241           | script_dir direction_reqd_event
```

```
242           | script_dir direction_less_event
243           | string_number_event

244 string_number_event: E_UNSIGNED

245 direction_less_char: '['
246                    | ']'
247                    | '~'
248                    | '('
249                    | ')'
250                    | "\!"
251                    | "\("
252                    | "\)"
253                    | "\>"
254                    | "\<"

255 direction_less_event: direction_less_char
256                     | EVENT_IDENTIFIER
257                     | tremolo_type

258 direction_reqd_event: gen_text_def
259                     | script_abbreviation

260 octave_check: /* empty */
261             | '='
262             | '=' sub_quotes
263             | '=' sup_quotes

264 sup_quotes: '\''
265           | sup_quotes '\''

266 sub_quotes: ','
267           | sub_quotes ','

268 steno_pitch: NOTENAME_PITCH
269            | NOTENAME_PITCH sup_quotes
270            | NOTENAME_PITCH sub_quotes

271 steno_tonic_pitch: TONICNAME_PITCH
272                  | TONICNAME_PITCH sup_quotes
273                  | TONICNAME_PITCH sub_quotes

274 pitch: steno_pitch

275 pitch_also_in_chords: pitch
276                     | steno_tonic_pitch

277 gen_text_def: full_markup
278             | string
279             | DIGIT

280 script_abbreviation: '^'
```

```
281                     | '+'
282                     | '-'
283                     | '|'
284                     | ">"
285                     | '.'
286                     | '_'

287 script_dir: '_'
288           | '^'
289           | '-'

290 absolute_pitch: steno_pitch

291 duration_length: multiplied_duration

292 optional_notemode_duration: /* empty */
293                           | multiplied_duration

294 steno_duration: bare_unsigned dots
295               | DURATION_IDENTIFIER dots

296 multiplied_duration: steno_duration
297                     | multiplied_duration '*' bare_unsigned
298                     | multiplied_duration '*' FRACTION

299 fraction: FRACTION
300         | UNSIGNED '/' UNSIGNED

301 dots: /* empty */
302     | dots '.'

303 tremolo_type: ':'
304             | ':' bare_unsigned

305 bass_number: DIGIT
306            | UNSIGNED
307            | STRING
308            | full_markup

309 figured_bass_alteration: '-'
310                        | '+'
311                        | '!'

312 bass_figure: "_"
313            | bass_number
314            | bass_figure ']'
315            | bass_figure figured_bass_alteration
316            | bass_figure figured_bass_modification

317 figured_bass_modification: "\+"
318                           | "\!"
319                           | '/'
```

```
320                         | "\"

321 br_bass_figure: bass_figure
322               | '[' bass_figure

323 figure_list: /* empty */
324            | figure_list br_bass_figure

325 figure_spec: FIGURE_OPEN figure_list FIGURE_CLOSE

326 optional_rest: /* empty */
327              | "\rest"

328 simple_element: pitch
                      exclamations
                      questions
                      octave_check
                      optional_notemode_duration
                      optional_rest
329               | DRUM_PITCH optional_notemode_duration
330               | RESTNAME optional_notemode_duration
331               | lyric_element optional_notemode_duration

332 simple_chord_elements: simple_element
333                       | new_chord
334                       | figure_spec optional_notemode_duration

335 lyric_element: lyric_markup
336              | LYRICS_STRING

337 new_chord: steno_tonic_pitch optional_notemode_duration
338          | steno_tonic_pitch
                optional_notemode_duration
                chord_separator
                chord_items

339 chord_items: /* empty */
340            | chord_items chord_item

341 chord_separator: ":"
342                | "^"
343                | "/" steno_tonic_pitch
344                | "/+" steno_tonic_pitch

345 chord_item: chord_separator
346           | step_numbers
347           | CHORD_MODIFIER

348 step_numbers: step_number
349             | step_numbers '.' step_number

350 step_number: bare_unsigned
```

```
351             | bare_unsigned '+'
352             | bare_unsigned "-"

353 tempo_range: bare_unsigned
354             | bare_unsigned '~' bare_unsigned

355 number_expression: number_expression '+' number_term
356                 | number_expression '-' number_term
357                 | number_term

358 number_term: number_factor
359             | number_factor '*' number_factor
360             | number_factor '/' number_factor

361 number_factor: '-' number_factor
362               | bare_number

363 bare_number: UNSIGNED
364             | REAL
365             | NUMBER_IDENTIFIER
366             | REAL NUMBER_IDENTIFIER
367             | UNSIGNED NUMBER_IDENTIFIER

368 bare_unsigned: UNSIGNED
369               | DIGIT

370 unsigned_number: bare_unsigned
371                 | NUMBER_IDENTIFIER

372 exclamations: /* empty */
373              | exclamations '!'

374 questions: /* empty */
375          | questions '?'

376 lyric_markup: LYRIC_MARKUP_IDENTIFIER

378 lyric_markup: LYRIC_MARKUP $ markup_top

379 full_markup_list: MARKUPLINES_IDENTIFIER

381 full_markup_list: "\markuplines" $ markup_list

382 full_markup: MARKUP_IDENTIFIER

384 full_markup: "\markup" $ markup_top

385 markup_top: markup_list
386           | markup_head_1_list simple_markup
387           | simple_markup

388 markup_list: MARKUPLINES_IDENTIFIER
```

```
  389             | markup_composed_list
  390             | markup_braced_list
  391             | markup_command_list

  392 markup_composed_list: markup_head_1_list markup_braced_list

  393 markup_braced_list: '{' markup_braced_list_body '}'

  394 markup_braced_list_body: /* empty */
  395                        | markup_braced_list_body markup
  396                        | markup_braced_list_body markup_list

  397 markup_command_list: MARKUP_LIST_FUNCTION markup_command_list_arguments

  398 markup_command_basic_arguments: EXPECT_MARKUP_LIST
                                        markup_command_list_arguments
                                        markup_list
  399                               | EXPECT_SCM
                                        markup_command_list_arguments
                                        embedded_scm
  400                               | EXPECT_NO_MORE_ARGS

  401 markup_command_list_arguments: markup_command_basic_arguments
  402                              | EXPECT_MARKUP
                                     markup_command_list_arguments
                                     markup

  403 markup_head_1_item: MARKUP_FUNCTION
                            EXPECT_MARKUP
                            markup_command_list_arguments

  404 markup_head_1_list: markup_head_1_item
  405                   | markup_head_1_list markup_head_1_item

  406 simple_markup: STRING
  407              | MARKUP_IDENTIFIER
  408              | LYRIC_MARKUP_IDENTIFIER
  409              | STRING_IDENTIFIER

  411 simple_markup: "\score" $0 '{' score_body '}'
  412              | MARKUP_FUNCTION markup_command_basic_arguments

  413 markup: markup_head_1_list simple_markup
  414       | simple_markup


Terminals, with rules where they appear



"-" (319) 352
"--" (340) 239
```

"/" (320) 343
"/+" (316) 344
":" (318) 341
"<" (321) 210
"<<" (323) 97
">" (322) 210 284
">>" (324) 97
"\!" (329) 250 318
"\" (325) 225 320
"\(" (331) 251
"\)" (328) 252
"\+" (334) 317
"\<" (333) 254
"\>" (326) 253
"\[" (330) 223
"\]" (332) 224
"\accepts" (261) 174
"\addlyrics" (259) 153 155 157 159
"\alias" (262) 177
"\alternative" (263) 92
"\book" (264) 44
"\bookpart" (265) 55
"\C[haracter]" (327)
"\change" (266) 164
"\chordmode" (267) 144
"\chords" (268) 148
"\consists" (269) 172
"\context" (270) 38 132
"\default" (271) 231 233
"\defaultchild" (272) 175
"\denies" (273) 176
"\description" (274) 179
"\drummode" (275) 142
"\drums" (276) 146
"\figuremode" (277) 143
"\figures" (278) 147
"\grobdescriptions" (279) 41
"\header" (280) 18
"\invalid" (281) 5
"\key" (282) 233 234
"\layout" (283) 75
"\lyricmode" (284) 145
"\lyrics" (285) 149
"\lyricsto" (286) 163
"\mark" (287) 229 231
"\markup" (288) 384
"\markuplines" (289) 381
"\midi" (290) 74
"\name" (291) 180
"\new" (315) 133
"\notemode" (292) 141
"\octave" (293)

```
"\once" (294) 191
"\override" (295) 170 186
"\paper" (296) 73
"\partial" (297) 227
"\relative" (298) 150 151
"\remove" (299) 173
"\repeat" (300) 93
"\rest" (301) 327
"\revert" (302) 171 187
"\score" (303) 65 411
"\sequential" (304) 94
"\set" (305) 188
"\simultaneous" (306) 96
"\skip" (307) 222
"\tempo" (308) 82 83 84
"\time" (314) 228
"\times" (309) 134
"\transpose" (310) 136
"\type" (311) 178
"\unset" (312) 169 189
"\with" (313) 103 104
"\~" (335) 230
"^" (317) 342
"_" (339) 312
"__" (336) 240
$end (0) 0
'!' (33) 311 373
'(' (40) 248
')' (41) 249
'*' (42) 297 298 359
'+' (43) 194 281 310 351 355
',' (44) 266 267
'-' (45) 282 289 309 356 361
'.' (46) 185 285 302 349
'/' (47) 300 319 360
':' (58) 303 304
'=' (61) 21 22 82 83 130 164 168 170 186 188 261 262 263
'?' (63) 375
'[' (91) 245 322
'\'' (39) 264 265
']' (93) 246 314
'^' (94) 280 288
'_' (95) 286 287
'{' (123) 18 38 44 55 65 77 78 92 94 95 96 103 393 411
'|' (124) 226 283
'}' (125) 18 38 44 55 65 72 92 94 95 96 103 393 411
'~' (126) 247 354
BOOK_IDENTIFIER (352) 46 57
CHORD_MODIFIER (354) 347
CHORD_REPETITION (355) 205
CHORDMODIFIER_PITCH (353)
CHORDMODIFIERS (341)
```

```
CONTEXT_DEF_IDENTIFIER (356) 40
CONTEXT_MOD_IDENTIFIER (357) 104 105 110
DIGIT (344) 36 203 279 305 369
DRUM_PITCH (358) 214 329
DURATION_IDENTIFIER (359) 295
E_UNSIGNED (345) 244
error (256) 4 54 64 70 81 88
EVENT_IDENTIFIER (360) 256
EXPECT_MARKUP (347) 118 119 122 123 402 403
EXPECT_MARKUP_LIST (350) 398
EXPECT_MUSIC (348) 117 217 219
EXPECT_NO_MORE_ARGS (351) 121 125 400
EXPECT_SCM (349) 120 124 399
FIGURE_CLOSE (337) 325
FIGURE_OPEN (338) 325
FRACTION (361) 298 299
LYRIC_MARKUP (342) 378
LYRIC_MARKUP_IDENTIFIER (363) 376 408
LYRICS_STRING (362) 20 196 199 336
MARKUP_FUNCTION (364) 403 412
MARKUP_IDENTIFIER (366) 382 407
MARKUP_LIST_FUNCTION (365) 397
MARKUPLINES_IDENTIFIER (367) 379 388
MULTI_MEASURE_REST (343) 206
MUSIC_FUNCTION (368) 128 216
MUSIC_IDENTIFIER (369) 99 155 159 161
NOTENAME_PITCH (370) 234 268 269 270
NUMBER_IDENTIFIER (371) 365 366 367 371
OUTPUT_DEF_IDENTIFIER (372) 78
PREC_BOT (260)
PREC_TOP (258)
REAL (373) 364 366
RESTNAME (374) 330
SCM_IDENTIFIER (375) 15 234
SCM_TOKEN (376) 14
SCORE_IDENTIFIER (377) 67
STRING (378) 19 164 168 182 192 195 307 406
STRING_IDENTIFIER (379) 193 197 409
TONICNAME_PITCH (380) 271 272 273
UNARY_MINUS (381)
UNSIGNED (346) 300 306 363 367 368

Nonterminals, with rules where they appear


$@1 (179)
    on left: 102, on right: 103
$@10 (291)
    on left: 410, on right: 411
$@2 (196)
    on left: 152, on right: 153
$@3 (197)
```

```
    on left: 154, on right: 155
$@4 (198)
    on left: 156, on right: 157
$@5 (199)
    on left: 158, on right: 159
$@6 (201)
    on left: 162, on right: 163
$@7 (275)
    on left: 377, on right: 378
$@8 (277)
    on left: 380, on right: 381
$@9 (279)
    on left: 383, on right: 384
absolute_pitch (240)
    on left: 290, on right: 150
alternative_music (173)
    on left: 91 92, on right: 93
assignment (155)
    on left: 21 22 23, on right: 3 17 79
assignment_id (154)
    on left: 19 20, on right: 21 22
bare_number (269)
    on left: 363 364 365 366 367, on right: 200 362
bass_number (248)
    on left: 305 306 307 308, on right: 313
book_block (159)
    on left: 44, on right: 7 25
bookpart_block (161)
    on left: 55, on right: 8 26 48
br_bass_figure (252)
    on left: 321 322, on right: 324
chord_body (216)
    on left: 210, on right: 209
chord_body_element (218)
    on left: 213 214 215, on right: 212 217
chord_body_elements (217)
    on left: 211 212, on right: 210 212
chord_item (262)
    on left: 345 346 347, on right: 340
chord_items (260)
    on left: 339 340, on right: 338 340
chord_separator (261)
    on left: 341 342 343 344, on right: 338 345
command_event (223)
    on left: 230 231 232 233 234, on right: 221
composite_music (182)
    on left: 111 112, on right: 10 50 60 90 151
context_change (202)
    on left: 164, on right: 101
context_def_spec_block (157)
    on left: 38, on right: 28 80
context_mod (207)
```

```
    on left: 181 182 183, on right: 42 109
context_mod_list (181)
    on left: 108 109 110, on right: 103 109 110
context_modification (178)
    on left: 103 104 105, on right: 37 43 107
context_prop_spec (208)
    on left: 184 185, on right: 186 187 188 189
direction_less_event (228)
    on left: 255 256 257, on right: 237 242
direction_reqd_event (229)
    on left: 258 259, on right: 241
dots (246)
    on left: 301 302, on right: 294 295 302
duration_length (241)
    on left: 291, on right: 222 227
event_chord (214)
    on left: 204 205 206 207 208, on right: 98
exclamations (272)
    on left: 372 373, on right: 213 328 373
figure_list (253)
    on left: 323 324, on right: 324 325
figure_spec (254)
    on left: 325, on right: 334
figured_bass_alteration (249)
    on left: 309 310 311, on right: 315
figured_bass_modification (251)
    on left: 317 318 319 320, on right: 316
fraction (245)
    on left: 299 300, on right: 134 228
full_markup_list (276)
    on left: 379 381, on right: 12 35 52 62
function_arglist_music_last (185)
    on left: 117, on right: 126
function_arglist_nonmusic_last (186)
    on left: 118 119 120, on right: 127
function_scm_argument (184)
    on left: 115 116, on right: 120 124
gen_text_def (237)
    on left: 277 278 279, on right: 258
generic_prefix_music_scm (189)
    on left: 128, on right: 131
lilypond (149)
    on left: 1 2 3 4 5, on right: 0 2 3 4 5
lilypond_header (153)
    on left: 18, on right: 6 53 63 68
lilypond_header_body (152)
    on left: 16 17, on right: 17 18
lyric_element (258)
    on left: 335 336, on right: 331
lyric_markup (274)
    on left: 376 378, on right: 335
markup (292)
```

```
    on left: 413 414, on right: 395 402
markup_braced_list (283)
    on left: 393, on right: 390 392
markup_braced_list_body (284)
    on left: 394 395 396, on right: 393 395 396
markup_command_list (285)
    on left: 397, on right: 391
markup_composed_list (282)
    on left: 392, on right: 389
markup_head_1_item (288)
    on left: 403, on right: 404 405
markup_head_1_list (289)
    on left: 404 405, on right: 386 392 405 413
markup_list (281)
    on left: 388 389 390 391, on right: 381 385 396 398
markup_top (280)
    on left: 385 386 387, on right: 378 384
mode_changing_head (192)
    on left: 141 142 143 144 145, on right: 137
multiplied_duration (244)
    on left: 296 297 298, on right: 291 293 297 298
music_function_chord_body (220)
    on left: 217 218, on right: 215
music_function_event (221)
    on left: 219 220, on right: 238
music_list (171)
    on left: 85 86 87 88, on right: 86 87 88 92 94 95 96 97
music_property_def (210)
    on left: 190 191, on right: 100
new_chord (259)
    on left: 337 338, on right: 333
new_lyrics (195)
    on left: 153 155 157 159, on right: 157 159 160 161
note_chord_element (215)
    on left: 209, on right: 208
number_expression (266)
    on left: 355 356 357, on right: 31 355 356
number_factor (268)
    on left: 361 362, on right: 358 359 360 361
number_term (267)
    on left: 358 359 360, on right: 355 356 357
octave_check (230)
    on left: 260 261 262 263, on right: 213 328
optional_context_mod (180)
    on left: 106 107, on right: 132 133 138
optional_id (190)
    on left: 129 130, on right: 132 133
optional_rest (255)
    on left: 326 327, on right: 328
output_def (166)
    on left: 72, on right: 13 27 69 71
output_def_body (169)
```

```
    on left: 77 78 79 80 81, on right: 72 79 80 81
output_def_head (167)
    on left: 73 74 75, on right: 76
output_def_head_with_mode_switch (168)
    on left: 76, on right: 77 78
paper_block (165)
    on left: 71, on right: 47 58
pitch (235)
    on left: 274, on right: 213 275 328
pitch_also_in_chords (236)
    on left: 275 276, on right: 136
post_events (224)
    on left: 235 236, on right: 204 205 206 209 213 214 236
property_operation (205)
    on left: 168 169 170 171, on right: 181
property_path (204)
    on left: 167, on right: 22 170 186
property_path_revved (203)
    on left: 165 166, on right: 166 167
questions (273)
    on left: 374 375, on right: 213 328 375
re_rhythmed_music (200)
    on left: 160 161 163, on right: 140
relative_music (194)
    on left: 150 151, on right: 139
repeated_music (174)
    on left: 93, on right: 135
score_block (163)
    on left: 65, on right: 9 24 49 59
score_body (164)
    on left: 66 67 68 69 70, on right: 65 68 69 70 411
script_dir (239)
    on left: 287 288 289, on right: 238 241 242
sequential_music (175)
    on left: 94 95, on right: 114
simple_chord_elements (257)
    on left: 332 333 334, on right: 204
simple_element (256)
    on left: 328 329 330 331, on right: 332
simple_music (177)
    on left: 98 99 100 101, on right: 89
simultaneous_music (176)
    on left: 96 97, on right: 113
steno_duration (243)
    on left: 294 295, on right: 82 83 296
steno_pitch (233)
    on left: 268 269 270, on right: 274 290
step_number (264)
    on left: 350 351 352, on right: 348 349
step_numbers (263)
    on left: 348 349, on right: 346 349
string (211)
```

```
    on left: 192 193 194, on right: 32 194 198 278
string_number_event (226)
    on left: 244, on right: 243
sub_quotes (232)
    on left: 266 267, on right: 262 267 270 273
sup_quotes (231)
    on left: 264 265, on right: 263 265 269 272
tempo_event (170)
    on left: 82 83 84, on right: 232
tempo_range (265)
    on left: 353 354, on right: 82 83
toplevel_expression (150)
    on left: 6 7 8 9 10 11 12 13, on right: 2
tremolo_type (247)
    on left: 303 304, on right: 257
unsigned_number (271)
    on left: 370 371, on right: 93
```

Annexe D GNU Free Documentation License

Version 1.3, 3 November 2008

Copyright © 2000, 2001, 2002, 2007, 2008 Free Software Foundation, Inc.
http://fsf.org/

Everyone is permitted to copy and distribute verbatim copies
of this license document, but changing it is not allowed.

0. PREAMBLE

   The purpose of this License is to make a manual, textbook, or other functional and useful document *free* in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

   This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

   We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

1. APPLICABILITY AND DEFINITIONS

   This License applies to any manual or other work, in any medium, that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. Such a notice grants a world-wide, royalty-free license, unlimited in duration, to use that work under the conditions stated herein. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you". You accept the license if you copy, modify or distribute the work in a way requiring permission under copyright law.

   A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

   A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (Thus, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

   The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License. If a section does not fit the above definition of Secondary then it is not allowed to be designated as Invariant. The Document may contain zero Invariant Sections. If the Document does not identify any Invariant Sections then there are none.

   The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License. A Front-Cover Text may be at most 5 words, and a Back-Cover Text may be at most 25 words.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, that is suitable for revising the document straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup, or absence of markup, has been arranged to thwart or discourage subsequent modification by readers is not Transparent. An image format is not Transparent if used for any substantial amount of text. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML, PostScript or PDF designed for human modification. Examples of transparent image formats include PNG, XCF and JPG. Opaque formats include proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML, PostScript or PDF produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

The "publisher" means any person or entity that distributes copies of the Document to the public.

A section "Entitled XYZ" means a named subunit of the Document whose title either is precisely XYZ or contains XYZ in parentheses following text that translates XYZ in another language. (Here XYZ stands for a specific section name mentioned below, such as "Acknowledgements", "Dedications", "Endorsements", or "History".) To "Preserve the Title" of such a section when you modify the Document means that it remains a section "Entitled XYZ" according to this definition.

The Document may include Warranty Disclaimers next to the notice which states that this License applies to the Document. These Warranty Disclaimers are considered to be included by reference in this License, but only as regards disclaiming warranties: any other implication that these Warranty Disclaimers may have is void and has no effect on the meaning of this License.

2. VERBATIM COPYING

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

3. COPYING IN QUANTITY

If you publish printed copies (or copies in media that commonly have printed covers) of the Document, numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both

covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a computer-network location from which the general network-using public has access to download using public-standard network protocols a complete Transparent copy of the Document, free of added material. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

4. MODIFICATIONS

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

A. Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.

B. List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has fewer than five), unless they release you from this requirement.

C. State on the Title page the name of the publisher of the Modified Version, as the publisher.

D. Preserve all the copyright notices of the Document.

E. Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.

F. Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.

G. Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.

H. Include an unaltered copy of this License.

I. Preserve the section Entitled "History", Preserve its Title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section Entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its

Title Page, then add an item describing the Modified Version as stated in the previous sentence.

J. Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.

K. For any section Entitled "Acknowledgements" or "Dedications", Preserve the Title of the section, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.

L. Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.

M. Delete any section Entitled "Endorsements". Such a section may not be included in the Modified Version.

N. Do not retitle any existing section to be Entitled "Endorsements" or to conflict in title with any Invariant Section.

O. Preserve any Warranty Disclaimers.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section Entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties—for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

5. COMBINING DOCUMENTS

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice, and that you preserve all their Warranty Disclaimers.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections Entitled "History" in the various original documents, forming one section Entitled "History"; likewise combine any sections Entitled "Acknowledgements", and any sections Entitled "Dedications". You must delete all sections Entitled "Endorsements."

6. COLLECTIONS OF DOCUMENTS

   You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

   You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

7. AGGREGATION WITH INDEPENDENT WORKS

   A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, is called an "aggregate" if the copyright resulting from the compilation is not used to limit the legal rights of the compilation's users beyond what the individual works permit. When the Document is included in an aggregate, this License does not apply to the other works in the aggregate which are not themselves derivative works of the Document.

   If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one half of the entire aggregate, the Document's Cover Texts may be placed on covers that bracket the Document within the aggregate, or the electronic equivalent of covers if the Document is in electronic form. Otherwise they must appear on printed covers that bracket the whole aggregate.

8. TRANSLATION

   Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License, and all the license notices in the Document, and any Warranty Disclaimers, provided that you also include the original English version of this License and the original versions of those notices and disclaimers. In case of a disagreement between the translation and the original version of this License or a notice or disclaimer, the original version will prevail.

   If a section in the Document is Entitled "Acknowledgements", "Dedications", or "History", the requirement (section 4) to Preserve its Title (section 1) will typically require changing the actual title.

9. TERMINATION

   You may not copy, modify, sublicense, or distribute the Document except as expressly provided under this License. Any attempt otherwise to copy, modify, sublicense, or distribute it is void, and will automatically terminate your rights under this License.

   However, if you cease all violation of this License, then your license from a particular copyright holder is reinstated (a) provisionally, unless and until the copyright holder explicitly and finally terminates your license, and (b) permanently, if the copyright holder fails to notify you of the violation by some reasonable means prior to 60 days after the cessation.

   Moreover, your license from a particular copyright holder is reinstated permanently if the copyright holder notifies you of the violation by some reasonable means, this is the first time you have received notice of violation of this License (for any work) from that copyright holder, and you cure the violation prior to 30 days after your receipt of the notice.

Termination of your rights under this section does not terminate the licenses of parties who have received copies or rights from you under this License. If your rights have been terminated and not permanently reinstated, receipt of a copy of some or all of the same material does not give you any rights to use it.

10. FUTURE REVISIONS OF THIS LICENSE

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation. If the Document specifies that a proxy can decide which future versions of this License can be used, that proxy's public statement of acceptance of a version permanently authorizes you to choose that version for the Document.

11. RELICENSING

"Massive Multiauthor Collaboration Site" (or "MMC Site") means any World Wide Web server that publishes copyrightable works and also provides prominent facilities for anybody to edit those works. A public wiki that anybody can edit is an example of such a server. A "Massive Multiauthor Collaboration" (or "MMC") contained in the site means any set of copyrightable works thus published on the MMC site.

"CC-BY-SA" means the Creative Commons Attribution-Share Alike 3.0 license published by Creative Commons Corporation, a not-for-profit corporation with a principal place of business in San Francisco, California, as well as future copyleft versions of that license published by that same organization.

"Incorporate" means to publish or republish a Document, in whole or in part, as part of another Document.

An MMC is "eligible for relicensing" if it is licensed under this License, and if all works that were first published under this License somewhere other than this MMC, and subsequently incorporated in whole or in part into the MMC, (1) had no cover texts or invariant sections, and (2) were thus incorporated prior to November 1, 2008.

The operator of an MMC Site may republish an MMC contained in the site under CC-BY-SA on the same site at any time before August 1, 2009, provided the MMC is eligible for relicensing.

ADDENDUM: How to use this License for your documents

To use this License in a document you have written, include a copy of the License in the document and put the following copyright and license notices just after the title page:

```
Copyright (C)  year  your name.
Permission is granted to copy, distribute and/or modify this document
under the terms of the GNU Free Documentation License, Version 1.3
or any later version published by the Free Software Foundation;
with no Invariant Sections, no Front-Cover Texts, and no Back-Cover
Texts.  A copy of the license is included in the section entitled ``GNU
Free Documentation License''.
```

If you have Invariant Sections, Front-Cover Texts and Back-Cover Texts, replace the "with. . . Texts." line with this:

```
with the Invariant Sections being list their titles, with
the Front-Cover Texts being list, and with the Back-Cover Texts
being list.
```

If you have Invariant Sections without Cover Texts, or some other combination of the three, merge those two alternatives to suit the situation.

If your document contains nontrivial examples of program code, we recommend releasing these examples in parallel under your choice of free software license, such as the GNU General Public License, to permit their use in free software.

Annexe E Index des commandes LilyPond

Cet index recense toutes les commandes et mots réservés de LilyPond, sous forme de lien vers les endroits du manuel où ils sont abordés. Chaque lien est constitué de deux parties : la première pointe directement là où la commande ou le terme apparaît, la seconde au début de la section où la commande ou le terme est abordé.

!

! 5

'

' 1

,

, ... 1

-

- 102

.

. 37

/

/ 345
/+ 346

:

: 136

<

< 137
<...> 137

=

= 9

>

> 137

?

? 5

[

[..... 80

]

] 80

_

_ 226

\

\! 104
\(..... 114
\) 114
\< 104
\> 104
\\ 142
\abs-fontsize 511
\accent 102
\accepts 468
\addChordShape 316
\addInstrumentDefinition 173
\addlyrics 221, 223
\addQuote 174
\aeolian 16
\afterGrace 96
\aikenHeads 33
\aikenHeadsMinor 33
\allowPageTurn 428
\alternative 124
\arpeggio 119
\arpeggioArrowDown 119
\arpeggioArrowUp 119
\arpeggioBracket 119
\arpeggioNormal 119
\arpeggioParenthesis 119
\arpeggioParenthesisDashed 119
\arrow-head 211, 534
\ascendens 369
\auctum 369
\augmentum 369
\autoBeamOff 70
\autoBeamOn 70
\autochange 281
\backslashed-digit 547
\balloonGrobText 189
\balloonLengthOff 189
\balloonLengthOn 189
\balloonText 189
\bar 84, 88
\barNumberCheck 93
\beam 534
\bendAfter 117
\bold 204, 511
\book 381, 384
\bookpart 382, 384, 426
\box 210, 511
\bracket 110, 210, 534
\break 426
\breathe 116
\breve 37, 47

\cadenzaOff ... 63
\cadenzaOn ... 63
\caesura ... 362
\caps ... 511
\cavum ... 369
\center-align ... 206, 519
\center-column ... 208, 519
\change ... 280
\char ... 547
\chordmode ... 3, 12, 313
\circle ... 210, 534
\clef ... 13
\coda ... 102
\column ... 208, 520
\column-lines ... 552
\combine ... 211, 520
\compressFullBarRests ... 51, 52
\concat ... 520
\cr ... 104
\cresc ... 106
\crescHairpin ... 106
\crescTextCresc ... 106
\cueDuring ... 178
\customTabClef ... 540
\decr ... 104
\decresc ... 106
\defaultTimeSignature ... 55
\deminutum ... 369
\denies ... 468
\descendens ... 369
\dim ... 106
\dimHairpin ... 106
\dimTextDecr ... 106
\dimTextDecresc ... 106
\dimTextDim ... 106
\dir-column ... 521
\displayLilyMusic ... 400
\divisioMaior ... 362
\divisioMaxima ... 362
\divisioMinima ... 362
\dorian ... 16
\dotsDown ... 38
\dotsNeutral ... 38
\dotsUp ... 38
\doubleflat ... 540
\doublesharp ... 540
\downbow ... 102
\downbow ... 292
\downmordent ... 102
\downprall ... 102
\draw-circle ... 211, 535
\draw-hline ... 535
\draw-line ... 211, 535
\drummode ... 154
\dynamic ... 110, 512
\dynamicDown ... 106
\dynamicNeutral ... 106
\dynamicUp ... 106
\easyHeadsOff ... 31
\easyHeadsOn ... 31
\epsfile ... 212, 535
\espressivo ... 102
\espressivo ... 105
\expandFullBarRests ... 51, 52
\eyeglasses ... 547
\f ... 104
\featherDurations ... 83
\fermata ... 102
\fermataMarkup ... 52
\fermataMarkup ... 102
\ff ... 104
\fff ... 104
\ffff ... 104
\fffff ... 104
\fill-line ... 208, 521
\fill-with-pattern ... 522
\filled-box ... 211, 536
\finalis ... 362
\finger ... 183, 512
\flageolet ... 102
\flat ... 540
\flexa ... 369
\fontCaps ... 512
\fontsize ... 204, 512
\footnote ... 547
\fp ... 104
\fraction ... 547
\frenchChords ... 349
\fret-diagram ... 305, 543
\fret-diagram-terse ... 307, 544
\fret-diagram-verbose ... 308, 544
\fromproperty ... 547
\funkHeads ... 33
\funkHeadsMinor ... 33
\general-align ... 208, 522
\germanChords ... 349
\glissando ... 118
\grace ... 95
\halfopen ... 102
\halign ... 207, 523
\harmonic ... 293
\harmonicsOff ... 293
\harmonicsOn ... 293
\harp-pedal ... 545
\hbracket ... 210, 536
\hcenter-in ... 524
\header ... 384
\hideKeySignature ... 334
\hideNotes ... 186
\hideSplitTiedTabNotes ... 299
\hideStaffSwitch ... 283
\hspace ... 525
\huge ... 182, 206, 512
\improvisationOff ... 36, 68
\improvisationOn ... 36, 68
\inclinatum ... 369
\include ... 393
\instrumentSwitch ... 173
\ionian ... 16
\italianChords ... 349
\italic ... 204, 513
\justified-lines ... 215, 552
\justify ... 209, 526
\justify-field ... 525
\justify-string ... 526
\keepWithTag ... 396
\key ... 16, 33
\killCues ... 179
\label ... 391
\laissezVibrer ... 45

\large ... 182, 206, 513
\larger ... 204, 206, 513
\layout ... 384, 422
\left-align ... 206, 527
\left-brace ... 548
\left-column ... 527
\lheel ... 102
\line ... 527
\linea ... 369
\lineprall ... 102
\locrian ... 16
\longa ... 37, 47
\longfermata ... 102
\lookup ... 548
\lower ... 207, 528
\ltoe ... 102
\lydian ... 16
\lyricmode ... 220, 221
\lyricsto ... 221, 223
\magnify ... 204, 513
\major ... 16
\makeClusters ... 140
\marcato ... 102
\mark ... 93, 197
\markalphabet ... 548
\markletter ... 549
\markup ... 197
\markup ... 201
\markup ... 202
\markuplines ... 201
\markuplines ... 215
\maxima ... 37, 47
\medium ... 514
\melisma ... 227
\melismaEnd ... 227
\mergeDifferentlyDottedOff ... 144
\mergeDifferentlyDottedOn ... 144
\mergeDifferentlyHeadedOff ... 144
\mergeDifferentlyHeadedOn ... 144
\mf ... 104
\midi ... 384
\minor ... 16
\mixolydian ... 16
\mordent ... 102
\mp ... 104
\musicglyph ... 94, 540
\natural ... 540
\noBeam ... 81
\noBreak ... 426
\noPageBreak ... 427
\noPageTurn ... 428
\normal-size-sub ... 514
\normal-size-super ... 514
\normal-text ... 514
\normalsize ... 182, 206, 515
\note ... 541
\note-by-number ... 540
\null ... 207, 549
\number ... 515
\numericTimeSignature ... 55
\octaveCheck ... 9
\on-the-fly ... 549
\oneVoice ... 141
\open ... 102
\open ... 292
\oriscus ... 369
\ottava ... 18
\override ... 549
\override en mode \lyricmode ... 220
\override-lines ... 553
\overrideTimeSignatureSettings ... 55
\p ... 104
\pad-around ... 210, 528
\pad-markup ... 210, 528
\pad-to-box ... 210, 528
\pad-x ... 210, 529
\page-link ... 550
\page-ref ... 391, 550
\pageBreak ... 427
\pageTurn ... 428
\paper ... 384, 413
\parallelMusic ... 151
\parenthesize ... 188
\parenthesize ... 536
\partcombine ... 148
\partial ... 62, 124, 125
\path ... 537
\pattern ... 550
\pes ... 369
\phrasingSlurDashed ... 114
\phrasingSlurDashPattern ... 115
\phrasingSlurDotted ... 114
\phrasingSlurDown ... 114
\phrasingSlurHalfDashed ... 115
\phrasingSlurHalfSolid ... 115
\phrasingSlurNeutral ... 114
\phrasingSlurSolid ... 114
\phrasingSlurUp ... 114
\phrygian ... 16
\pitchedTrill ... 123
\portato ... 102
\postscript ... 212, 538
\pp ... 104
\ppp ... 104
\pppp ... 104
\ppppp ... 104
\prall ... 102
\pralldown ... 102
\prallmordent ... 102
\prallprall ... 102
\prallup ... 102
\predefinedFretboardsOff ... 321
\predefinedFretboardsOn ... 321
\put-adjacent ... 529
\quilisma ... 369
\quoteDuring ... 174, 178
\raise ... 207, 529
\relative ... 2, 12, 282
\RemoveEmptyStaves ... 168, 169
\removeWithTag ... 396
\repeat ... 124
\repeat percent ... 133
\repeat tremolo ... 135
\repeat unfold ... 131
\repeatTie ... 45
\repeatTie ... 127
\repeatTie ... 243
\rest ... 47
\reverseturn ... 102
\revertTimeSignatureSettings ... 56

\rfz ... 104
\rheel ... 102
\right-align ... 206, 529
\right-brace ... 550
\right-column ... 530
\rightHandFinger ... 323
\roman ... 515
\rotate ... 530
\rounded-box ... 210, 538
\rtoe ... 102
\sacredHarpHeads ... 33
\sacredHarpHeadsMinor ... 33
\sans ... 516
\scale ... 539
\scaleDurations ... 43, 64
\score ... 380, 384, 541
\segno ... 102
\semiflat ... 542
\semiGermanChords ... 349
\semisharp ... 542
\sesquiflat ... 542
\sesquisharp ... 543
\set ... 73, 472
\sf ... 104
\sff ... 104
\sfz ... 104
\sharp ... 543
\shiftOff ... 144
\shiftOn ... 144
\shiftOnn ... 144
\shiftOnnn ... 144
\shortfermata ... 102
\showKeySignature ... 334
\showStaffSwitch ... 283
\signumcongruentiae ... 102
\simple ... 516
\skip ... 49
\slashed-digit ... 550
\slurDashed ... 112
\slurDashPattern ... 112
\slurDotted ... 112
\slurDown ... 112
\slurHalfDashed ... 112
\slurHalfSolid ... 112
\slurNeutral ... 112
\slurSolid ... 112
\slurUp ... 113
\small ... 182, 206, 516
\smallCaps ... 516
\smaller ... 204, 206, 516
\snappizzicato ... 102
\sostenutoOff ... 285
\sostenutoOn ... 285
\southernHarmonyHeads ... 33
\southernHarmonyHeadsMinor ... 33
\sp ... 104
\spp ... 104
\staccatissimo ... 102
\staccato ... 102
\startGroup ... 192
\startStaff ... 163, 164
\startTrillSpan ... 122
\stemDown ... 188
\stemNeutral ... 188
\stemUp ... 188
\stencil ... 551
\stopGroup ... 192
\stopped ... 102
\stopStaff ... 163, 164, 168
\stopTrillSpan ... 122
\storePredefinedDiagram ... 316
\stropha ... 369
\strut ... 551
\sub ... 205, 517
\super ... 205, 517
\sustainOff ... 285
\sustainOn ... 285
\tabFullNotation ... 297
\table-of-contents ... 393, 553
\tag ... 396
\taor ... 334
\teeny ... 182, 206, 517
\tempo ... 59
\tenuto ... 102
\text ... 518
\textLengthOff ... 52
\textLengthOff ... 195
\textLengthOn ... 52
\textLengthOn ... 195
\textSpannerDown ... 195
\textSpannerNeutral ... 195
\textSpannerUp ... 195
\thumb ... 102
\thumb ... 183
\tied-lyric ... 543
\tieDashed ... 45
\tieDotted ... 45
\tieDown ... 45
\tieNeutral ... 45
\tieSolid ... 45
\tieUp ... 45
\time ... 54, 73
\times ... 39, 64
\tiny ... 182, 206, 518
\tocItem ... 393
\translate ... 208, 530
\translate-scaled ... 208, 531
\transparent ... 551
\transpose ... 2, 3, 10, 12
\transposedCueDuring ... 180
\transposition ... 19, 174
\treCorde ... 285
\triangle ... 211, 539
\trill ... 102
\trill ... 122
\tupletDown ... 39
\tupletNeutral ... 39
\tupletUp ... 39
\turn ... 102
\tweak ... 474
\typewriter ... 518
\unaCorda ... 285
\underline ... 204, 518
\unfoldRepeats ... 406
\unHideNotes ... 186
\unset ... 473
\upbow ... 102
\upbow ... 292
\upmordent ... 102
\upprall ... 102

\upright 519
\varcoda 102
\vcenter 531
\verbatim-file 551
\verylongfermata 102
\virga 369
\virgula 362
\voiceFourStyle 143
\voiceNeutralStyle 143
\voiceOne 141
\voiceOne ... \voiceFour 141
\voiceOneStyle 143
\voiceThreeStyle 143
\voiceTwoStyle 143
\vspace 531
\walkerHeads 33
\walkerHeadsMinor 33
\whiteout 551
\with 464
\with-color 186, 552
\with-dimensions 552
\with-link 552
\with-url 539
\woodwind-diagram 546
\wordwrap 209, 532
\wordwrap-field 532
\wordwrap-internal 553
\wordwrap-lines 215, 553
\wordwrap-string 533
\wordwrap-string-internal 553

|

| 92

~

~ 44

A

add ChordShape 316
addInstrumentDefinition 173
addQuote 174
aeolian 16
afterGrace 96
aikenHeads 33
aikenHeadsMinor 33
annotate-spacing 460
arpeggio 119
arpeggioArrowDown 119
arpeggioArrowUp 119
arpeggioBracket 119
arpeggioNormal 119
arpeggioParenthesis 119
arpeggioParenthesisDashed 119
arranger 386
arrow-head 211
aug 345
auto-first-page-number 421
autoBeaming 73
autoBeamOff 70
autoBeamOn 70
autochange 281

B

Balloon_engraver 189
balloonGrobText 189
balloonLengthOff 189
balloonLengthOn 189
balloonText 189
banjo-c-tuning 326
banjo-modal-tuning 326
banjo-open-d-tuning 326
banjo-open-dm-tuning 326
bar 84, 88
barCheckSynchronize 92
BarNumber 89
barNumberCheck 93
barNumberVisibility 89
bartype 88
base-shortest-duration 448
baseMoment 73
beamExceptions 73
beatStructure 73
bendAfter 117
binding-offset 419
blank-after-score-page-force 420
blank-last-page-force 420
blank-page-force 420
bold 204
bookTitleMarkup 389
bottom-margin 415
box 210
bracket 110, 210, 285
breakable 71
breakbefore 386
breathe 116
breve 37, 47

C

cadenzaOff 63
cadenzaOn 63
center-align 206
center-column 208
change 280
check-consistency 418
chordmode 3, 12, 313
chordNameExceptions 347
ChordNames 313
chordNameSeparator 348
chordNoteNamer 349
chordPrefixSpacer 349
chordRootNamer 349
circle 210
clef 13
color 186
column 208
combine 211
common-shortest-duration 448
Completion_heads_engraver 67
composer 386
compressFullBarRests 51, 52
controlpitch 9
copyright 386
cr 104
cresc 106
crescHairpin 106

crescTextCresc 106
cross 30
cross-staff 283
cueDuring 178
currentBarNumber 88, 100

D

decr 104
decresc 106
dedication 386
default 20, 22
defaultBarType 88
defaultTimeSignature 55
dim 106, 345
dimHairpin 106
dimTextDecr 106
dimTextDecresc 106
dimTextDim 106
dodecaphonic 25
dorian 16
dotsDown 38
dotsNeutral 38
dotsUp 38
draw-circle 211
draw-line 211
drummode 154
DrumStaff 154
dynamic 110
dynamicDown 106
DynamicLineSpanner 106
dynamicNeutral 106
dynamicUp 106

E

easyHeadsOff 31
easyHeadsOn 31
epsfile 212
espressivo 105
evenFooterMarkup 389
evenHeaderMarkup 389
expandFullBarRests 51, 52

F

f 104
featherDurations 83
fermataMarkup 52
ff 104
fff 104
ffff 104
fffff 104
fill-line 208
filled-box 211
finger 183
first-page-number 421
flag-style 283
followVoice 283
font-interface 183, 216
font-size 182, 183
fontsize 204
fontSize 182
forget 26
four-string-banjo 326
fp 104
fret-diagram 305
fret-diagram-interface 310
fret-diagram-terse 307
fret-diagram-verbose 308
FretBoards 312
funkHeads 33
funkHeadsMinor 33

G

general-align 208
glissando 118
grace 95
GregorianTranscriptionStaff 154
Grid_line_span_engraver 190
Grid_point_engraver 190
gridInterval 190
grow-direction 83

H

halign 207
hbracket 210
hideKeySignature 334
hideNotes 186
hideStaffSwitch 283
horizontal-shift 419
Horizontal_bracket_engraver 192
huge 182, 206

I

improvisationOff 36, 68
improvisationOn 36, 68
indent 172, 419, 451
inner-margin 419
instrument 386
instrumentSwitch 173
ionian 16
italic 204

J

justified-lines 215
justify 209

K

key 16, 33
killCues 179

L

laissezVibrer 45
large 182, 206
larger 204, 206
last-bottom-spacing 417
layout file 423
left-align 206
left-margin 418
length 283
line-width 417, 451

locrian ... 16
longa ... 37, 47
lower ... 207
ly:minimal-breaking ... 428
ly:optimal-breaking ... 427
ly:page-turn-breaking ... 427
lydian ... 16

M

m ... 345
magnify ... 204
magstep ... 182
maj ... 345
major ... 16
majorSevenSymbol ... 348
make-dynamic-script ... 110
make-pango-font-tree ... 218
makeClusters ... 140
mark ... 93, 197
markup ... 197
markup ... 201
markup ... 202
markup-markup-spacing ... 417
markup-system-spacing ... 416
markuplines ... 201
markuplines ... 215
max-systems-per-page ... 419
maxima ... 37, 47
measureLength ... 73, 100
measurePosition ... 62, 100
MensuralStaff ... 154
mergeDifferentlyDottedOff ... 144
mergeDifferentlyDottedOn ... 144
mergeDifferentlyHeadedOff ... 144
mergeDifferentlyHeadedOn ... 144
meter ... 386
mf ... 104
min-systems-per-page ... 420
minimumFret ... 298
minimumPageTurnLength ... 428
minimumRepeatLengthForPageTurn ... 428
minor ... 16
mixed ... 285
mixolydian ... 16
modern ... 22
modern-cautionary ... 23
modern-voice ... 23
modern-voice-cautionary ... 23
mp ... 104
MultiMeasureRestText ... 52
musicglyph ... 94

N

neo-modern ... 24
neo-modern-cautionary ... 24
neo-modern-voice ... 24
neo-modern-voice-cautionary ... 25
no-reset ... 25
noBeam ... 81
normalsize ... 182, 206
Note_heads_engraver ... 67
null ... 207
numericTimeSignature ... 55

O

octaveCheck ... 9
oddFooterMarkup ... 389
oddHeaderMarkup ... 389
oneVoice ... 141
opus ... 386
ottava ... 18
outer-margin ... 419
outside-staff-horizontal-padding ... 446
outside-staff-padding ... 446
outside-staff-priority ... 446

P

p ... 104
pad-around ... 210
pad-markup ... 210
pad-to-box ... 210
pad-x ... 210
page-breaking ... 420
page-breaking-system-system-spacing ... 420
page-count ... 420
page-spacing-weight ... 421
paper-height ... 414
paper-width ... 417
parallelMusic ... 151
parenthesize ... 188
partcombine ... 148
partial ... 62
pedalSustainStyle ... 285
percent ... 133
phrasingSlurDashed ... 114
phrasingSlurDashPattern ... 115
phrasingSlurDotted ... 114
phrasingSlurDown ... 114
phrasingSlurHalfDashed ... 115
phrasingSlurHalfSolid ... 115
phrasingSlurNeutral ... 114
phrasingSlurSolid ... 114
phrasingSlurUp ... 114
phrygian ... 16
piano accidentals ... 23
piano-cautionary ... 24
PianoStaff ... 279, 281
piece ... 386
pipeSymbol ... 92
Pitch_squash_engraver ... 68
pitchedTrill ... 123
poet ... 386
postscript ... 212
pp ... 104
ppp ... 104
pppp ... 104
ppppp ... 104
predefinedFretboardsOff ... 321
predefinedFretboardsOn ... 321
print-all-headers ... 388, 421
print-first-page-number ... 421
print-page-number ... 421

Q

quotedCueEventTypes 177
quotedEventTypes 177
quoteDuring 174, 178

R

r 47
R 50
ragged-bottom 415
ragged-last 418, 451
ragged-last-bottom 415
ragged-right 418, 451
raise 207
relative 2, 12, 282
repeatCommands 129
repeatTie 45
rest 47
rfz 104
rgb-color 187
RhythmicStaff 154
right-align 206
right-margin 418
rightHandFinger 323
rounded-box 210

S

s 49
sacredHarpHeads 33
sacredHarpHeadsMinor 33
scaleDurations 43, 64
score-markup-spacing 417
score-system-spacing 417
scoreTitleMarkup 389
set 73
set-accidental-style 20
set-octavation 18
sf 104
sff 104
sfz 104
shiftOff 144
shiftOn 144
shiftOnn 144
shiftOnnn 144
short-indent 172, 419
show-available-fonts 218
showFirstLength 401
showKeySignature 334
showLastLength 401
showStaffSwitch 283
skip 49
skipTypesetting 401
slurDashed 112
slurDashPattern 112
slurDotted 112
slurDown 112
slurHalfDashed 112
slurHalfSolid 112
slurNeutral 112
slurSolid 112
slurUp 113
small 182, 206
smaller 204, 206
sostenutoOff 285
sostenutoOn 285
southernHarmonyHeads 33
southernHarmonyHeadsMinor 33
sp 104
spacing 448
spp 104
Staff.midiInstrument 403
Staff_symbol_engraver 168
start-repeat 129
startGroup 192
startStaff 163, 164
startTrillSpan 122
Stem 283
stem-spacing-correction 448
stemDown 188
stemLeftBeamCount 81
stemNeutral 188
stemRightBeamCount 81
stemUp 188
stopGroup 192
stopStaff 163, 164, 168
stopTrillSpan 122
storePredefinedDiagram 316
stringTunings 312
StringTunings 302
sub 205
subsubtitle 386
subtitle 386
super 205
sus 345
sustainOff 285
sustainOn 285
system-count 420
system-separator-markup 421
system-system-spacing 417
systems-per-page 420

T

TabStaff 154, 296
TabVoice 296
tagline 386
taor 334
teaching 25
teeny 182, 206
tempo 59
text 285
textLengthOff 195
textLengthOn 52
textLengthOn 195
textLenthOff 52
textSpannerDown 195
textSpannerNeutral 195
textSpannerUp 195
thumb 183
tieDashed 45
tieDotted 45
tieDown 45
tieNeutral 45
tieSolid 45
tieUp 45
time 54, 73
times 39, 64

timeSignatureFraction 64
tiny 182, 206
title 386
top-margin 415
top-markup-spacing 417
top-system-spacing 417
translate 208
translate-scaled 208
transpose 2, 3, 10, 12
transposedCueDuring 180
transposition 19, 174
treCorde 285
tremolo 135
tremoloFlags 136
triangle 211
trill 122
tupletDown 39
tupletNeutral 39
TupletNumber 40
tupletNumberFormatFunction 39
tupletSpannerDuration 39
tupletUp 39
two-sided 418

U

unaCorda 285
underline 204
unfold 131
unHideNotes 186

V

VaticanaStaff 154
voice 20, 22
Voice 141
voiceOne 141

W

walkerHeads 33
walkerHeadsMinor 33
whichBar 88
with-color 186
wordwrap 209
wordwrap-lines 215

X

x11-color 186, 187

Annexe F Index de LilyPond

En plus des commandes et mots réservés de LilyPond, cet index recense les termes musicaux qui s'y rapportent. Tout comme dans l'index des commandes, chaque lien est constitué de deux parties : la première pointe directement là où la commande ou le terme apparaît, la seconde au début de la section où la commande ou le terme est abordé.

!

! 5

'

' 1

,

, ... 1

-

- 102

.

. 37

/

/ 345
/+ 346

:

: 136

<

< 137
<...> 137

=

= 9

>

> 137

?

? 5

[

[..... 80

]

] 80

_

_ 226

\

\! 104
\(..... 114
\) 114
\< 104
\> 104
\\ 142
\abs-fontsize 511
\accent 102
\accepts 468
\addChordShape 316
\addInstrumentDefinition 173
\addlyrics 221, 223
\addQuote 174
\aeolian 16
\afterGrace 96
\aikenHeads 33
\aikenHeadsMinor 33
\allowPageTurn 428
\alternative 124
\arpeggio 119
\arpeggioArrowDown 119
\arpeggioArrowUp 119
\arpeggioBracket 119
\arpeggioNormal 119
\arpeggioParenthesis 119
\arpeggioParenthesisDashed 119
\arrow-head 211, 534
\ascendens 369
\auctum 369
\augmentum 369
\autoBeamOff 70
\autoBeamOff et \partcombine 72
\autoBeamOn 70
\autochange 281
\backslashed-digit 547
\balloonGrobText 189
\balloonLengthOff 189
\balloonLengthOn 189
\balloonText 189
\bar 84, 88
\barNumberCheck 93
\beam 534
\bendAfter 117
\bold 204, 511
\book 381, 384
\bookpart 382, 384, 426
\box 210, 511
\bracket 110, 210, 534
\break 426
\breathe 116

\breve ... 37, 47
\cadenzaOff ... 63
\cadenzaOn ... 63
\caesura ... 362
\caps ... 511
\cavum ... 369
\center-align ... 206, 519
\center-column ... 208, 519
\change ... 280
\char ... 547
\chordmode ... 3, 12, 313
\circle ... 210, 534
\clef ... 13
\coda ... 102
\column ... 208, 520
\column-lines ... 552
\combine ... 211, 520
\compressFullBarRests ... 51, 52
\concat ... 520
\cr ... 104
\cresc ... 106
\crescHairpin ... 106
\crescTextCresc ... 106
\cueDuring ... 178
\customTabClef ... 540
\decr ... 104
\decresc ... 106
\defaultTimeSignature ... 55
\deminutum ... 369
\denies ... 468
\descendens ... 369
\dim ... 106
\dimHairpin ... 106
\dimTextDecr ... 106
\dimTextDecresc ... 106
\dimTextDim ... 106
\dir-column ... 521
\displayLilyMusic ... 400
\divisioMaior ... 362
\divisioMaxima ... 362
\divisioMinima ... 362
\dorian ... 16
\dotsDown ... 38
\dotsNeutral ... 38
\dotsUp ... 38
\doubleflat ... 540
\doublesharp ... 540
\downbow ... 102
\downbow ... 292
\downmordent ... 102
\downprall ... 102
\draw-circle ... 211, 535
\draw-hline ... 535
\draw-line ... 211, 535
\drummode ... 154
\dynamic ... 110, 512
\dynamicDown ... 106
\dynamicNeutral ... 106
\dynamicUp ... 106
\easyHeadsOff ... 31
\easyHeadsOn ... 31
\epsfile ... 212, 535
\espressivo ... 102
\espressivo ... 105
\expandFullBarRests ... 51, 52
\eyeglasses ... 547
\f ... 104
\featherDurations ... 83
\fermata ... 102
\fermataMarkup ... 52
\fermataMarkup ... 102
\ff ... 104
\fff ... 104
\ffff ... 104
\fffff ... 104
\fill-line ... 208, 521
\fill-with-pattern ... 522
\filled-box ... 211, 536
\finalis ... 362
\finger ... 183, 512
\flageolet ... 102
\flat ... 540
\flexa ... 369
\fontCaps ... 512
\fontsize ... 204, 512
\footnote ... 547
\fp ... 104
\fraction ... 547
\frenchChords ... 349
\fret-diagram ... 305, 543
\fret-diagram-terse ... 307, 544
\fret-diagram-verbose ... 308, 544
\fromproperty ... 547
\funkHeads ... 33
\funkHeadsMinor ... 33
\general-align ... 208, 522
\germanChords ... 349
\glissando ... 118
\grace ... 95
\halfopen ... 102
\halign ... 207, 523
\harmonic ... 293
\harmonicsOff ... 293
\harmonicsOn ... 293
\harp-pedal ... 545
\hbracket ... 210, 536
\hcenter-in ... 524
\header ... 384
\hideKeySignature ... 334
\hideNotes ... 186
\hideSplitTiedTabNotes ... 299
\hideStaffSwitch ... 283
\hspace ... 525
\huge ... 182, 206, 512
\improvisationOff ... 36, 68
\improvisationOn ... 36, 68
\inclinatum ... 369
\include ... 393
\instrumentSwitch ... 173
\ionian ... 16
\italianChords ... 349
\italic ... 204, 513
\justified-lines ... 215, 552
\justify ... 209, 526
\justify-field ... 525
\justify-string ... 526
\keepWithTag ... 396
\key ... 16, 33
\killCues ... 179
\label ... 391

\laissezVibrer ... 45
\large ... 182, 206, 513
\larger ... 204, 206, 513
\layout ... 384, 422
\left-align ... 206, 527
\left-brace ... 548
\left-column ... 527
\lheel ... 102
\line ... 527
\linea ... 369
\lineprall ... 102
\locrian ... 16
\longa ... 37, 47
\longfermata ... 102
\lookup ... 548
\lower ... 207, 528
\ltoe ... 102
\lydian ... 16
\lyricmode ... 220, 221
\lyricsto ... 221, 223
\magnify ... 204, 513
\major ... 16
\makeClusters ... 140
\marcato ... 102
\mark ... 93, 197
\markalphabet ... 548
\markletter ... 549
\markup ... 197
\markup ... 201
\markup ... 202
\markuplines ... 201
\markuplines ... 215
\maxima ... 37, 47
\medium ... 514
\melisma ... 227
\melismaEnd ... 227
\mergeDifferentlyDottedOff ... 144
\mergeDifferentlyDottedOn ... 144
\mergeDifferentlyHeadedOff ... 144
\mergeDifferentlyHeadedOn ... 144
\mf ... 104
\midi ... 384
\minor ... 16
\mixolydian ... 16
\mordent ... 102
\mp ... 104
\musicglyph ... 94, 540
\natural ... 540
\noBeam ... 81
\noBreak ... 426
\noPageBreak ... 427
\noPageTurn ... 428
\normal-size-sub ... 514
\normal-size-super ... 514
\normal-text ... 514
\normalsize ... 182, 206, 515
\note ... 541
\note-by-number ... 540
\null ... 207, 549
\number ... 515
\numericTimeSignature ... 55
\octaveCheck ... 9
\on-the-fly ... 549
\once ... 473
\oneVoice ... 141
\open ... 102
\open ... 292
\oriscus ... 369
\ottava ... 18
\override ... 549
\override en mode \lyricmode ... 220
\override-lines ... 553
\overrideTimeSignatureSettings ... 55
\p ... 104
\pad-around ... 210, 528
\pad-markup ... 210, 528
\pad-to-box ... 210, 528
\pad-x ... 210, 529
\page-link ... 550
\page-ref ... 391, 550
\pageBreak ... 427
\pageTurn ... 428
\paper ... 384
\paper ... 389
\paper ... 413
\parallelMusic ... 151
\parenthesize ... 188
\parenthesize ... 536
\partcombine ... 148
\partcombine et \autoBeamOff ... 72
\partial ... 62, 124, 125
\path ... 537
\pattern ... 550
\pes ... 369
\phrasingSlurDashed ... 114
\phrasingSlurDashPattern ... 115
\phrasingSlurDotted ... 114
\phrasingSlurDown ... 114
\phrasingSlurHalfDashed ... 115
\phrasingSlurHalfSolid ... 115
\phrasingSlurNeutral ... 114
\phrasingSlurSolid ... 114
\phrasingSlurUp ... 114
\phrygian ... 16
\pitchedTrill ... 123
\portato ... 102
\postscript ... 212, 538
\pp ... 104
\ppp ... 104
\pppp ... 104
\ppppp ... 104
\prall ... 102
\pralldown ... 102
\prallmordent ... 102
\prallprall ... 102
\prallup ... 102
\predefinedFretboardsOff ... 321
\predefinedFretboardsOn ... 321
\put-adjacent ... 529
\quilisma ... 369
\quoteDuring ... 174, 178
\raise ... 207, 529
\relative ... 2, 12, 282
\RemoveEmptyStaves ... 168, 169
\removeWithTag ... 396
\repeat ... 124
\repeat percent ... 133
\repeat tremolo ... 135
\repeat unfold ... 131
\repeatTie ... 45

\repeatTie ... 127
\repeatTie ... 243
\rest ... 47
\reverseturn ... 102
\revertTimeSignatureSettings ... 56
\rfz ... 104
\rheel ... 102
\right-align ... 206, 529
\right-brace ... 550
\right-column ... 530
\rightHandFinger ... 323
\roman ... 515
\rotate ... 530
\rounded-box ... 210, 538
\rtoe ... 102
\sacredHarpHeads ... 33
\sacredHarpHeadsMinor ... 33
\sans ... 516
\scale ... 539
\scaleDurations ... 43, 64
\score ... 380, 384, 541
\segno ... 102
\semiflat ... 542
\semiGermanChords ... 349
\semisharp ... 542
\sesquiflat ... 542
\sesquisharp ... 543
\set ... 73, 472
\sf ... 104
\sff ... 104
\sfz ... 104
\sharp ... 543
\shiftOff ... 144
\shiftOn ... 144
\shiftOnn ... 144
\shiftOnnn ... 144
\shortfermata ... 102
\showKeySignature ... 334
\showStaffSwitch ... 283
\signumcongruentiae ... 102
\simple ... 516
\skip ... 49
\slashed-digit ... 550
\slurDashed ... 112
\slurDashPattern ... 112
\slurDotted ... 112
\slurDown ... 112
\slurHalfDashed ... 112
\slurHalfSolid ... 112
\slurNeutral ... 112
\slurSolid ... 112
\slurUp ... 113
\small ... 182, 206, 516
\smallCaps ... 516
\smaller ... 204, 206, 516
\snappizzicato ... 102
\sostenutoOff ... 285
\sostenutoOn ... 285
\southernHarmonyHeads ... 33
\southernHarmonyHeadsMinor ... 33
\sp ... 104
\spp ... 104
\staccatissimo ... 102
\staccato ... 102
\startGroup ... 192
\startStaff ... 163, 164
\startTrillSpan ... 122
\stemDown ... 188
\stemNeutral ... 188
\stemUp ... 188
\stencil ... 551
\stopGroup ... 192
\stopped ... 102
\stopStaff ... 163, 164, 168
\stopTrillSpan ... 122
\storePredefinedDiagram ... 316
\stropha ... 369
\strut ... 551
\sub ... 205, 517
\super ... 205, 517
\sustainOff ... 285
\sustainOn ... 285
\tabFullNotation ... 297
\table-of-contents ... 393, 553
\tag ... 396
\taor ... 334
\teeny ... 182, 206, 517
\tempo ... 59
\tenuto ... 102
\text ... 518
\textLengthOff ... 52
\textLengthOff ... 195
\textLengthOn ... 52
\textLengthOn ... 195
\textSpannerDown ... 195
\textSpannerNeutral ... 195
\textSpannerUp ... 195
\thumb ... 102
\thumb ... 183
\tied-lyric ... 543
\tieDashed ... 45
\tieDotted ... 45
\tieDown ... 45
\tieNeutral ... 45
\tieSolid ... 45
\tieUp ... 45
\time ... 54, 73
\times ... 39, 64
\tiny ... 182, 206, 518
\tocItem ... 393
\translate ... 208, 530
\translate-scaled ... 208, 531
\transparent ... 551
\transpose ... 2, 3, 10, 12
\transposedCueDuring ... 180
\transposition ... 19, 174
\treCorde ... 285
\triangle ... 211, 539
\trill ... 102
\trill ... 122
\tupletDown ... 39
\tupletNeutral ... 39
\tupletUp ... 39
\turn ... 102
\tweak ... 474
\typewriter ... 518
\unaCorda ... 285
\underline ... 204, 518
\unfoldRepeats ... 406
\unHideNotes ... 186

\unset ... 473
\upbow ... 102
\upbow ... 292
\upmordent ... 102
\upprall ... 102
\upright ... 519
\varcoda ... 102
\vcenter ... 531
\verbatim-file ... 551
\verylongfermata ... 102
\virga ... 369
\virgula ... 362
\voiceFourStyle ... 143
\voiceNeutralStyle ... 143
\voiceOne ... 141
\voiceOne ... \voiceFour ... 141
\voiceOneStyle ... 143
\voiceThreeStyle ... 143
\voiceTwoStyle ... 143
\vspace ... 531
\walkerHeads ... 33
\walkerHeadsMinor ... 33
\whiteout ... 551
\with ... 464
\with-color ... 186, 552
\with-dimensions ... 552
\with-link ... 552
\with-url ... 539
\woodwind-diagram ... 546
\wordwrap ... 209, 532
\wordwrap-field ... 532
\wordwrap-internal ... 553
\wordwrap-lines ... 215, 553
\wordwrap-string ... 533
\wordwrap-string-internal ... 553

Displaying music expressions

Displaying music expressions ... 474

Espacement

Espacement ... 448

Markup construction in Scheme

Markup construction in Scheme ... 111, 260

References bibliographiques

Références bibliographiques ... 347

Tutoriel Scheme

Tutoriel Scheme ... 463

|

| ... 92

~

~ ... 44

1

15ma ... 18

8

8va ... 18
8ve ... 18

A

a due ... 148
à l'italienne, papier ... 414
absolues, hauteurs ... 1
absolues, octaves ... 1
accent ... 102, 553
acciaccatura ... 583
acciaccature ... 95
acciaccature, multi-note ... 99
accidentel, quart de ton ... 7
accidentelles automatiques ... 20
accidentelles, altérations ... 20
accolade verticale ... 155
accolades, taille ... 216
accord arpégé ... 119
accord et mode relatif ... 138
accord nommés et diagrammes de fret ... 313
accord, carrures pour cordes frettées ... 316
accord, diagrammes ... 304, 312
accord, diagrammes automatiques ... 320
accordages de banjo ... 326
accordages prédéfinis et cordes frettées ... 302
accordéon, symboles de registre ... 286
accordéon, tirettes ... 286
accords ... 137, 346
accords chiffrés, exceptions ... 348
accords et altérations ... 26
accords et couleur ... 187
accords et liaisons de tenue ... 44
accords et octave relative ... 4
accords incomplets ... 344
accords jazz, chiffrage ... 349
accords nommés en MIDI ... 406
accords, chiffrage jazz ... 343
accords, doigtés ... 184
accords, répartition sur plusieurs portées avec \autochange ... 283
add ChordShape ... 316
addChordShape ... 583
adding a white background to text ... 551
addInstrumentDefinition ... 173
addInstrumentDefinition ... 583
addQuote ... 174
addQuote ... 583
aeolian ... 16
afterGrace ... 96
afterGrace ... 583
agrégats ... 346
Aiken, tête de note ... 33
aikenHeads ... 33
aikenHeadsMinor ... 33
ajout de texte ... 194
al niente ... 107
alignAboveContext ... 468
alignBelowContext ... 468

aligné, ragged ... 412
alignement des markups ... 206
alignement du numéro de mesure ... 91
alignement du texte ... 206
alignement du texte, commandes ... 210
alignement et cadence ... 100
alignement horizontal du texte ... 206
alignement vertical du texte ... 207
`allowPageTurn` ... 583
altération ... 5
altération de précaution ... 5
altération de précaution style *modern voice* ... 23
altération de précaution, style *modern* ... 23
altération entre parenthèses ... 5
altération et liaison de tenue ... 6
altération, style ... 20
altération, style *dodecaphonic* ... 25
altération, style *modern* ... 22
altération, style *modern cautionary* ... 23
altération, style *no reset* ... 25
altération, style par défaut ... 20
altération, trilles avec hauteur explicite ... 123
altérations ... 355
altérations et accords ... 26
altérations et notes simultanées ... 26
altérations multivoix ... 23, 24
altérations style *piano* ... 23
altérations style *piano cautionary* ... 24
altérations, style *default* ... 22
altérations, style *forget* ... 26
altérations, style *modern* ... 22
altérations, style *modern-cautionary* ... 22
altérations, style *neo-modern* ... 24
altérations, style *teaching* ... 25
altérations, style *voice* ... 22
alternative et liaison de prolongation ... 127
alternative et paroles ... 242
alternative et texte ... 130
alternative, bascule sur une mélodie ... 250
alternative, reprise ... 124
Amazing Grace : exemple pour cornemuse ... 334
ambitus ... 27
amplitude ... 27
anacrouse ... 62
anacrouse et reprise ... 125
analyse musicologique ... 192
`annotate-spacing` ... 460
`applyContext` ... 584
`applyMusic` ... 584
`applyOutput` ... 584
`appoggiatura` ... 584
appoggiature ... 95
arabe, noms des notes en ... 374
armure ... 5, 16
armures arabes ... 375
arpège ... 119
arpège, style parenthèse ... 121
arpège, symboles spéciaux ... 119
`arpeggio` ... 119
`arpeggioArrowDown` ... 119
`arpeggioArrowUp` ... 119
`arpeggioBracket` ... 119
`arpeggioNormal` ... 119
`arpeggioParenthesis` ... 119
`arpeggioParenthesisDashed` ... 119
`arranger` ... 386
`arrow-head` ... 211
articulation, liaison ... 112
articulation-event ... 177
articulations ... 102, 360
artificiels, harmoniques ... 293
aspect d'un symbole de demi-bémol ... 374
`assertBeamQuant` ... 584
`assertBeamSlope` ... 584
`associatedVoice` ... 221
associatedVoice ... 223, 250
`aug` ... 345
`auto-first-page-number` ... 421
autobeam ... 73
`autoBeaming` ... 73
`autoBeamOff` ... 70
`autoBeamOn` ... 70
`autochange` ... 281
`autochange` ... 584
autochange et mode relatif ... 282
Automatique, changement de portée ... 281
automatique, diagramme d'accord ... 320
automatique, diagramme de fret ... 320
automatiques, altérations accidentelles ... 20

B

backslashed digits ... 547
bagpipe ... 334
balise ... 396
`Balloon_engraver` ... 189
`balloonGrobText` ... 189
`balloonGrobText` ... 584
`balloonLengthOff` ... 189
`balloonLengthOn` ... 189
`balloonText` ... 189
`balloonText` ... 584
banjo, accordages ... 326
banjo, tablature ... 302
banjo, tablature pour ... 294
banjo, tablatures ... 326
`banjo-c-tuning` ... 326
`banjo-modal-tuning` ... 326
`banjo-open-d-tuning` ... 326
`banjo-open-dm-tuning` ... 326
Banter ... 349
`bar` ... 84, 88
`bar` ... 584
`barCheckSynchronize` ... 92
`BarNumber` ... 89
`barNumberCheck` ... 93
`barNumberCheck` ... 584
`barNumberVisibility` ... 89
baroque, ornementation ... 102, 553
barre de mesure invisible ... 84
barre de mesure manuelle ... 84
barre de mesure par défaut, modification du style ... 88
barre finale ... 84
barré, indication de ... 305
barres de mesure ... 84
barres de mesure, suppression ... 63
barres de mesure, symboles au dessus de ... 197
barres de reprise ... 84

Bartók pizzicato 294
bartype 88
base-shortest-duration 448
baseMoment 73
basse chiffrée 350
basse continue 350
battements par minute 59
batterie 326, 327
batterie, portée 154
beamExceptions 73
beatStructure 73
bécarre 5
bémol 5
bendAfter 117
bendAfter 584
binding-offset 419
bisbigliando 290
blanc 385
blank-after-score-page-force 420
blank-last-page-force 420
blank-page-force 420
blocs de texte 194
bold 204
bookOutputName 584
bookOutputSuffix 584
bookTitleMarkup 389
bottom-margin 415
bouché 102, 553
box 210
bracket 110, 210, 285
breakable 71
breakbefore 386
breathe 116
breathe 584
breve 37, 47
broderies 95
bulle 189

C

cadence 63, 100
cadence et alignement 100
cadence et ligature 64
cadence et saut de ligne 64
cadence et saut de page 64
cadenza 100
cadenzaOff 63
cadenzaOn 63
calques 142
cantique 266
caractères réservés, impression 203
caractères spéciaux en mode markup 203
caratères, taille 204
cases 298
center-align 206
center-column 208
centering a column of text 519
centrage des nuances entre deux portées de piano 279
centrage du texte sur la page 208
césure 116
change 280
changement d'instrument 173
changement de fonte 204
changement de nom d'instrument 173
Changement de portée automatique 281
changement de portée et collision 280
changement de portée forcé 280
changements de portée 283
changements de portée manuels 280
changing direction of text columns 521
check-consistency 418
chevrons 137
chiffrage d'accords, exceptions 348
chiffrage de mesure 54
chiffrages d'accords 346
chiffrages des accords 345
chiffre indicateur de mesure 54
chiffres de mesure multiples 465
chœur, citation d'instrument 260
chœur, partition de 254
chœur, système 155
chorale, clef de tenor 14
chordGlissando 584
chordmode 3, 12, 313
chordNameExceptions 347
ChordNames 313
chordNameSeparator 348
chordNoteNamer 349
chordPrefixSpacer 349
chordRootNamer 349
Christian Harmony, tête de note 33
chutes 117
circle 210
circling text 534
citation 174
citation, fin 179
clavier, portées pour instrument à 279
claviers, centrage des nuances 279
clé d'ut 13
clé de fa 13
clé de sol 13
clef 5
clef 13
clef 584
clef et transposition 14
clefs 357
clés 357
clés, portées pour instrument à 279
cluster 140
coda 94, 102, 553
coda sur une barre de mesure 197
collision de notes 144
collision et changement de portée 280
collision et numéro de mesure 92
collisions 144
colonnes de texte 208
color 186
coloration d'objets 186
coloration de notes 186
colorier des voix 143
coloring text 552
colorisation d'objets 186
colorisation de notes 186
column 208
combine 211
comma 378
commandes d'alignement du texte 210
commentaire textuel 202
common-shortest-duration 448

`Completion_heads_engraver` ... 67
`composer` ... 386
composite, métrique ... 64
`compoundMeter` ... 584
`compressFullBarRests` ... 51, 52
concatenating text ... 520
condenser les silences ... 54
contexte, définition en MIDI ... 405
`contextStringTuning` ... 584
contrôle des hauteurs ... 9
controlling general text alignment ... 522
`controlpitch` ... 9
`copyright` ... 386
copyright ... 389, 400
corde à vide, indication ... 292
cordes d'orchestre ... 291
cordes frettées, carrures d'accord ... 316
cordes frettées, doigtés main droite ... 323
cordes frettées, harmoniques ... 325
cordes frettées, indication de la position et du barré ... 325
cordes frettées, notes étouffées ... 325
cordes frettés et accordages prédéfinis ... 302
cordes numérotées ... 295
cordes, écriture pour ... 291
cornemuse ... 334
cornemuse : exemple ... 334
couches ... 142
couleur rgb ... 187
couleur rvb ... 187
couleur, notes d'un accord ... 187
couleurs ... 186
couleurs x11 ... 186
Couleurs, liste ... 490
coup d'archet ... 102
coups de gratte, indication ... 68
courbes ... 111
`cr` ... 104
creating empty text objects ... 549
creating horizontal spaces in text ... 525
creating text fractions ... 547
creating vertical spaces in text ... 531, 551
`cresc` ... 106
crescendo ... 104
`crescHairpin` ... 106
`crescTextCresc` ... 106
crochet de nolet, positionnement ... 39
crochet de regroupement de notes ... 192
crochet vertical ... 155
crochets ... 192, 359
crochets de phrasé ... 192
croix, tête de note ... 30
`cross` ... 30
`cross-staff` ... 283
cross-staff tremolo ... 137
`cueClef` ... 585
`cueClefUnset` ... 585
`cueDuring` ... 178
`cueDuring` ... 585
`cueDuringWithClef` ... 585
`currentBarNumber` ... 88, 100
custodes ... 361
custos ... 361

D

D.S al Fine ... 94
`deadNote` ... 585
début de portée ... 163
décélage de voix ... 144
décoration du texte ... 210
`decr` ... 104
`decresc` ... 106
decrescendo ... 104
`dedication` ... 386
`default` ... 20, 22
default, style d'altérations ... 22
`defaultBarType` ... 88
`defaultNoteHeads` ... 585
`defaultTimeSignature` ... 55
défaut, durée ... 37
délimitation, systèmes imbriqués ... 159
demi-bémol ... 8, 374
demi-bémols ... 5
demi-dièse ... 8
demi-dièses ... 5
dérogations en mode paroles ... 220
diagrammes d'accord pour instrument fretté ... 304
diagrammes de fret ... 304
diagrammes de fret et transposition ... 314
diagrammes de fret personnalisés ... 310
diagrammes de fret personnalisés, ajout ... 315
diagrammes personnalisés de fret ... 304, 310
dièse ... 5
`dim` ... 106, 345
`dimHairpin` ... 106
diminuendo ... 104
`dimTextDecr` ... 106
`dimTextDecresc` ... 106
`dimTextDim` ... 106
`displayLilyMusic` ... 585
`displayMusic` ... 585
disponibilité des fontes ... 218
distance entre deux portées de piano ... 283
distance entre les portées ... 432
divisio ... 362
division de notes ... 67
division et paroles ... 245
divisiones ... 362
`dodecaphonic` ... 25
dodecaphonic, style d'altérations ... 25
doigté ... 183
doigtés des accords ... 184
doigtés et numéros de corde ... 295
doigtés et silences multimesure ... 54
doigtés main droite et cordes frettées ... 323
doigtés, ajout à des diagrammes de fret ... 322
`dorian` ... 16
dorien ... 16
`dotsDown` ... 38
`dotsNeutral` ... 38
`dotsUp` ... 38
double barre ... 84
double bémol ... 5
double dièse ... 5
double pause ... 47
double point ... 37
doublement pointée, note ... 37
`draw-circle` ... 211

`draw-line` ... 211
drawing a line across a page ... 535
drawing beams within text ... 534
drawing boxes with rounded corners ... 536
drawing boxes with rounded corners around text ... 538
drawing circles within text ... 535
drawing lines within text ... 535
drawing paths ... 537
drawing solid boxes within text ... 536
drawing triangles within text ... 539
`drummode` ... 154
`DrumStaff` ... 154
durée d'un silence ... 47
durée par défaut ... 37
durées ... 37
durées, échelonnement ... 43
`dynamic` ... 110
dynamic-event ... 177
`dynamicDown` ... 106
`DynamicLineSpanner` ... 106
`dynamicNeutral` ... 106
dynamics, vertical positioning ... 106
`dynamicUp` ... 106

E

easy play, têtes de note ... 31
`easyHeadsOff` ... 31
`easyHeadsOn` ... 31
échelonnement de musique ... 43
échelonnement des durées ... 43
écrire la musique en parallèle ... 151
embedding graphic objects ... 210
encadrement du texte ... 210
encapsulated postscript ... 402
enclosing text in a box with rounded corners ... 538
enclosing text within a box ... 511
`endSpanners` ... 585
entête ... 389
éolien ... 16
EPS, format de sortie ... 402
`epsfile` ... 212
espace ... 385
espacement au sein d'un système ... 432
espacement autour du texte ... 210
espacement des paroles ... 237
espacement entre les portées ... 432
espacement horizontal ... 447
espacement vertical ... 432
espacement, affichage des valeurs ... 460
espaces dans les paroles ... 226
espaces, dans les paroles ... 220
espressivo ... 102
`espressivo` ... 105
espressivo ... 553
étiquette ... 202
Étiquette de texte ... 194
étiquette et silence multimesure ... 52
étiquette textuelle ... 202
`evenFooterMarkup` ... 389
`evenHeaderMarkup` ... 389
exceptions, chiffrage d'accords ... 348
exemple de musique arabe ... 377
`expandFullBarRests` ... 51, 52
expansion de reprise ... 131
exposant ... 205
expressions markup ... 202
extenseur ... 195, 230
extenseur, mise en forme ... 195

F

`f` ... 104
fantômes, notes ... 188
`featherDurations` ... 83, 585
`fermataMarkup` ... 52
Feta font ... 491
`ff` ... 104
`fff` ... 104
`ffff` ... 104
`fffff` ... 104
`fill-line` ... 208
`filled-box` ... 211
fin alternative ... 124
fin de citation ... 179
fin de portée ... 163
fin de réplique ... 179
finalis ... 362
`finger` ... 183
`first-page-number` ... 421
`flag-style` ... 283
flageolet ... 102, 553
flat, double ... 5
fois, première ... 124
`followVoice` ... 283
Font, Feta ... 491
`font-interface` ... 183, 216
`font-size` ... 182, 183
fonte, changement ... 204
fonte, définition de la taille ... 423
fontes disponibles ... 218
fontes et markup ... 216
fontes, choix par défaut ... 218
fontes, familles ... 205
fontes, généralités ... 216
`fontsize` ... 204
`fontSize` ... 182
footer ... 389
`footnote` ... 585
`footnoteGrob` ... 585
`forget` ... 26
forget, style d'altérations ... 26
format de repère ... 94
formatage du numéro de mesure ... 90
Forme Semai ... 376
`four-string-banjo` ... 326
`fp` ... 104
Frenched scores ... 168
fret et transposition ... 314
fret, ajout de diagrammes personnalisés ... 315
fret, ajout de doigtés aux diagrammes ... 322
fret, diagrammes ... 304, 312
fret, diagrammes automatiques ... 320
fret, diagrammes avec noms d'accord ... 313
fret, diagrammes personnalisés ... 304, 310
`fret-diagram` ... 305
fret-diagram et markup ... 305

`fret-diagram-interface` ... 310
`fret-diagram-terse` ... 307
fret-diagram-terse markup ... 307
`fret-diagram-verbose` ... 308
fret-diagram-verbose markup ... 308
`FretBoards` ... 312
frets ... 298
Funk, tête de note ... 33
`funkHeads` ... 33
`funkHeadsMinor` ... 33
fusion de notes ... 144

G

`general-align` ... 208
`glissando` ... 118
glissando, indication en tablature ... 300
glyphes musicales ... 94
gouttière ... 418
`grace` ... 95
`grace` ... 585
graphiques, intégration ... 212
graphiques, tracé d'objets ... 210
graphisme dans la notation ... 211
graphisme, tracé ... 210
gras ... 204
`GregorianTranscriptionStaff` ... 154
grégorien, ligatures de neumes carrés ... 364
`Grid_line_span_engraver` ... 190
`Grid_point_engraver` ... 190
`gridInterval` ... 190
groupement de mesures ... 77
groupement de pulsations ... 77
groupements de note manuels ... 80
`grow-direction` ... 83
grupetto ... 102, 553
guidon ... 361
guillemets dans les paroles ... 226
guillemets en mode markup ... 203
guillemets, dans les paroles ... 220
guitare basse, tablature ... 302
guitare, coups de gratte ... 68
guitare, grille d'accords ... 68
guitare, tablature pour ... 294
guitare, tête de note ... 30

H

Hal Leonard ... 31
`halign` ... 207
hampe barrée ... 97
hampe, enjambement portées ... 283
hampes ... 188
hampes invisibles ... 188
Harmonica Sacra, tête de note ... 33
`harmonicByFret` ... 585
`harmonicByRatio` ... 585
`harmonicNote` ... 585
`harmonicsOn` ... 585
harmonique, tête de note ... 30
harmoniques artificiels ... 293
harmoniques et cordes frettées ... 325
harmoniques naturels ... 293
harmoniques, indication en tablature ... 300
harpe sacrée, tête de note ... 33
harpe, diagrammes de pédales ... 290
harpe, pédales ... 290
harpes ... 290
hauteurs ... 1
hauteurs en MIDI ... 406
`hbracket` ... 210
header ... 389
`hideKeySignature` ... 334
`hideNotes` ... 186
`hideStaffSwitch` ... 283
horizontal, espacement ... 447
`horizontal-shift` ... 419
`Horizontal_bracket_engraver` ... 192
horizontally centering text ... 519
hufnagel ... 354
`huge` ... 182, 206
hymne ... 266

I

identificateurs ... 385
images, intégration ... 212
imbrication de reprises ... 129
imbrication de systèmes ... 159
importing stencils into text ... 551
impression de caractères réservés ... 203
impression de caractères spéciaux ... 203
improvisation ... 36
`improvisationOff` ... 36, 68
`improvisationOn` ... 36, 68
include-settings ... 399
inclusion de fichiers ... 393
`indent` ... 172, 419, 451
indépendant, texte ... 201
indication d'octave relative ... 2
indication de corde à vide ... 292
indication de la position et du barré pour cordes frettées ... 325
indication de nuance, personnalisation ... 109
indication de tempo et nolet ... 43
indication du barré ... 305
indication textuelle ... 197
indications d'archet ... 292
indice ... 205
info-bulle ... 189
inlining an Encapsulated PostScript image ... 535
`inner-margin` ... 419
inserting music into text ... 541
inserting PostScript directly into text ... 538
inserting URL links into text ... 539
`instrument` ... 386
instrument, changement ... 173
instrument, changement de nom ... 173
instrument, citation ... 260
instrument, nom d' ... 403
instrument, noms ... 171
instrument, noms abrégés ... 171
instruments à vent ... 330
instruments transpositeurs ... 10
`instrumentSwitch` ... 173
`instrumentSwitch` ... 585
intégration de graphiques ... 212
inter-portée, ligatures ... 280

inter-portée, notes ... 280
interfaces de rendu ... 470
intervalles de comma ... 378
intervalles medium ... 373
`inversion` ... 585
invisible, barre de mesure ... 84
invisibles, notes ... 186
`ionian` ... 16
ionien ... 16
`italic` ... 204
italique ... 204

J

jazz, chiffrages d'accords ... 349
justifié, texte ... 209
`justified-lines` ... 215
`justify` ... 209
justifying lines of text ... 552
justifying text ... 526

K

`keepWithTag` ... 586
`key` ... 16, 33
`killCues` ... 179
`killCues` ... 586
kirchenpausen ... 52

L

`label` ... 586
laissez vibrer ... 45
`laissezVibrer` ... 45
landscape ... 414
`language` ... 586
`languageRestore` ... 586
`languageSaveAndChange` ... 586
langues, noms de note ... 7
`large` ... 182, 206
`larger` ... 204, 206
`last-bottom-spacing` ... 417
`layout file` ... 423
left aligning text ... 527
`left-align` ... 206
`left-margin` ... 418
legato ... 111
`length` ... 283
levée ... 62
levée dans une reprise ... 125
liaison d'articulation ... 111
liaison de prolongation ... 44
liaison de prolongation et alternative ... 127
liaison de prolongation et reprise avec alternative ... 45
liaison de prolongation, apparence ... 45
liaison de prolongation, pointillés ... 45
liaison de prolongation, tirets ... 45
liaison de prolongationt et reprise ... 127
liaison de tenue et accords ... 44
liaison de tenue et altération ... 6
liaison de tenue et répétition ... 45
liaison et reprise ... 129
liaison, au-dessous des notes ... 112
liaison, au-dessus des notes ... 112
liaison, densité des tirets ... 112, 115
liaison, laissez vibrer ... 45
liaison, orientation manuelle ... 112
liaison, style de trait ... 112, 114
liaison, tirets ... 112, 114
liaison, tirets et trait continu ... 112, 115
liaison, trait plein ... 112, 114
liaison, trait pointillé ... 112, 114
liaisons d'articulation multiples ... 112
liaisons d'articulation simultanées ... 112
liaisons de phrasé ... 114
liaisons de phrasé multiples ... 114
liaisons de phrasé simultanées ... 114
liaisons de prolongation, positionnement ... 45
liaisons, dans les paroles ... 226
ligature en fin de partition ... 78
ligature en fin de voix polyphonique ... 78
ligature en soufflet ... 83
ligature et cadence ... 64
ligature et mélisme ... 71
ligature et métrique ... 55
ligature et musique non mesurée ... 64
ligature et musique polymétrique ... 65
ligature et saut de ligne ... 71
ligature inter-portée ... 280
ligature manuelle et notes d'ornement ... 80
ligature manuelle, orientation ... 80
ligature, mesure à 4/4 ... 76
Ligatures ... 362
ligatures automatiques, génération de ... 73
ligatures automatiques, personnalisation ... 73
ligatures in text ... 520
ligatures manuelles ... 70, 80
Ligatures mensurales ... 363
Ligatures mensurales blanches ... 363
ligatures, définition de règles ... 70
ligatures, personnalisation ... 70
ligatures, subdivision ... 77
ligne supplémentaire ... 162
ligne, passer à la suivante ... 424
lignes ... 117
lignes de portée, épaisseur ... 162
lignes de portée, nombre de ... 162
`line-width` ... 417, 451
Liste des couleurs ... 490
liturgie et musique ... 266
`locrian` ... 16
locrien ... 16
`longa` ... 37, 47
longueur de note ... 37
losange, tête de note ... 30, 293
`lower` ... 207
lowering text ... 528
`ly:minimal-breaking` ... 428
`ly:optimal-breaking` ... 427
`ly:page-turn-breaking` ... 427
`lydian` ... 16
lydien ... 16

M

`m` ... 345
`magnify` ... 204

magnifying text ... 513
`magstep` ... 182
main droite, doigtés pour cordes frettées ... 323
`maj` ... 345
majeur ... 16
`major` ... 16
`majorSevenSymbol` ... 348
makam ... 379
makamlar ... 378, 379
`make-dynamic-script` ... 110
`make-pango-font-tree` ... 218
`makeClusters` ... 140
`makeClusters` ... 586
`makeDefaultStringTunings` ... 586
`makeStringTuning` ... 586
mandoline, tablature ... 302
manuelle, barre de mesure ... 84
`Manuels` ... 1
maqam ... 373, 378
maqams ... 373
marcato ... 102, 553
marges, texte qui dépasse ... 195
`mark` ... 93, 197
`markup` ... 197
`markup` ... 201
`markup` ... 202
markup et fontes ... 216
markup et fret-diagram ... 305
markup multi-ligne ... 208
markup multi-page ... 215
markup, centrage sur la page ... 208
markup, commandes d'alignement du texte ... 210
markup, encadrement du texte ... 210
markup, expressions ... 202
markup, inclusion de musique ... 212
markup, inclusion de partition ... 214
markup, ornementation du texte ... 210
markup, rembourrage du texte ... 210
markup, syntaxe ... 202
markup, texte au kilomètre ... 209
markup, texte justifié ... 209
markup, texte multi-page ... 215
`markup-markup-spacing` ... 417
`markup-system-spacing` ... 416
`markuplines` ... 201
`markuplines` ... 215
markups, alignement ... 206
Masquage de portée ... 168
masquage des hampes ... 188
masquées, notes ... 186
`max-systems-per-page` ... 419
`maxima` ... 37, 47
`measureLength` ... 73, 100
`measurePosition` ... 62, 100
Medicaea, Editio ... 354
mélisme ... 227, 230
mélisme et ligature ... 71
mélodie alternative ... 250
mélodie d'une portée à une autre ... 283
mélodie, affichage du rythme seul ... 68
mensural ... 354
`MensuralStaff` ... 154
MensuralStaffContext ... 370, 371
MensuralVoiceContext ... 370, 371
`mergeDifferentlyDottedOff` ... 144
`mergeDifferentlyDottedOn` ... 144
`mergeDifferentlyHeadedOff` ... 144
`mergeDifferentlyHeadedOn` ... 144
merging text ... 520
mesure à 3/4 et ligature ... 76
mesure de silence ... 47
mesure entière de silence ... 50
mesure incomplète ... 62
mesure, changement de longueur ... 62
mesure, numéro de ... 88, 100
mesure, répétition de ... 133
mesure, subdivision ... 77
mesures à compter ... 50
mesures, vérification des limites ... 92
`meter` ... 386
métrique ... 54, 359
métrique composite ... 64
métrique double ... 64
métrique et ligature ... 55
métrique par défaut ... 55
métrique polymétrique ... 64
métrique, retour au propriétés par défaut ... 56
métrique, styles ... 55
métrique, visibilité ... 54
métriques arabes ... 376
métriques composites ... 58
métriques décomposées ... 58
métronome, indication ... 59
`mf` ... 104
microtonalité en MIDI ... 406
microtonalités ... 8
MIDI et définition de contexte ... 405
MIDI et transposition ... 19
MIDI, accords nommés ... 406
MIDI, hauteurs ... 406
MIDI, microtonalité ... 406
MIDI, quart de ton ... 406
MIDI, Rythme ... 406
`min-systems-per-page` ... 420
mineur ... 16
`minimumFret` ... 298
`minimumPageTurnLength` ... 428
`minimumRepeatLengthForPageTurn` ... 428
`minor` ... 16
mirroring markup ... 539
mise en page ... 389
`mixed` ... 285
`mixolydian` ... 16
mixolydien ... 16
`modalInversion` ... 586
`modalTranspose` ... 586
mode accords ... 344
mode markup et caractères spéciaux ... 203
mode markup et guillemets ... 203
mode relatif et accord ... 138
mode relatif et autochange ... 282
modèle de musique arabe ... 377
`modern` ... 22
modern accidental style ... 23
modern cautionary, style d'altération ... 23
modern voice, styme d'altération de précaution ... 23
modern, style d'altération ... 22, 23
modern, style d'altération de précaution ... 23
`modern-cautionary` ... 23
modern-cautionary, style d'altération ... 22

`modern-voice` ... 23
`modern-voice-cautionary` ... 23
modes ... 16
modes anciens ... 16
modification du style par défaut des barres de mesure ... 88
modifier des propriétés ... 472
mordant ... 102, 553
motet ... 254
mouvements, plusieurs ... 381
`mp` ... 104
multi-ligne, markup ... 208
multi-ligne, texte ... 208
multi-page, markup ... 215
`MultiMeasureRestText` ... 52
multiples voix ... 144
multiples, liaisons d'articulation ... 112
multiples, liaisons de phrasé ... 114
multivoix, altérations ... 23, 24
musicale, citation ... 260
`musicglyph` ... 94
`musicMap` ... 586
musique ancienne, masquage de portée ... 169
musique ancienne, transcription ... 158
Musique arabe ... 373
musique arabe, exemple ... 377
musique dans un markup ... 212
musique échelonnée ... 43
musique en parallèle ... 151
musique entremêlée ... 151
musique non mesurée ... 63, 100
musique non mesurée et ligature ... 64
musique non mesurée et saut de ligne ... 64
musique non mesurée et saut de page ... 64
musique ottomane ... 378
musique répétitive ... 131
musique turque ... 378

N

naturel ... 5
naturels, harmoniques ... 293
`neo-modern` ... 24
neo-modern, style d'altérations ... 24
`neo-modern-cautionary` ... 24
neo-modern-cautionary, style d'altérations ... 24
`neo-modern-voice` ... 24
neo-modern-voice accidental style ... 24
neo-modern-voice, style d'altération ... 24
`neo-modern-voice-cautionary` ... 25
neumes carrés et ligatures ... 364
niente, al ... 107
no reset, style d'altérations ... 25
`no-reset` ... 25
`noBeam` ... 81
nolet, modification du chiffre ... 40
nolets ... 39
nolets, formatage ... 39
nom d'instrument ... 171
nom de personnage ... 248
nom du chanteur ... 248
nombre de portées ... 164
noms d'instrument, autres contextes ... 173
noms d'instrument, centrés ... 171
noms de note ... 1
noms de note, hollandais ... 5
noms de note, langues ... 7
noms de note, par défaut ... 5
noms des notes en arabe ... 374
non mesurée, musique ... 100
non musicaux, symboles ... 211
non-ASCII characters ... 399
`noPageBreak` ... 586
`noPageTurn` ... 586
`normalsize` ... 182, 206
notation dans un markup ... 212
notation et graphisme ... 211
notation facile ... 31
notation, explication ... 189
note d'ornement, synchronisation ... 99
note fondamentale ... 344
note invisible ... 186
note masquée ... 186
note-event ... 177
`Note_heads_engraver` ... 67
notes ajoutées ... 344
notes colorisées dans un accord ... 187
notes d'ornement ... 95
notes d'ornement en fin de note ... 96
notes d'ornement et crochet de nolet ... 43
notes d'ornement et retouches ... 97
notes d'ornement, mise en forme ... 97
notes en couleur ... 186
notes étouffées et cordes frettées ... 325
notes fantômes ... 188
notes inter-portée ... 280
notes penchées ... 36
notes pointées ... 37
notes profilées ... 33
notes simultanées et altérations ... 26
notes within text by log and dot-count ... 540
notes within text by string ... 541
notes, division ... 67
notes, longueur ... 37
notes, noms selon la langue ... 7
notes, petites ... 181
notes, taille ... 182
notes, taille standard ... 183
nouvelle portée ... 154
nuances ... 104
nuances éditoriales ... 110
nuances entre les portées d'un système pianistique ... 279
nuances entre parenthèses ... 110
nuances personnalisées ... 109
nuances successives sur une note tenue ... 105
nuances suggérées ... 110
`null` ... 207
`numericTimeSignature` ... 55
numéro de corde ... 295
numéro de couplet ... 247
numéro de mesure ... 100
numéro de mesure à intervalle régulier ... 89
numéro de mesure et collision ... 92
numéro de mesure et reprises ... 129
numéro de mesure, alignement ... 91
numéro de mesure, formatage ... 90
numéros de corde et doigtés ... 295
numéros de mesure ... 88

numérotation des mesures, suppression 63

O

objets en couleur 186
objets graphiques 470
objets graphiques, tracé 210
octave et clef 14
octave relative et accords 4
octave relative et transposition 3
octave relative, indication 2
octave, spécification 1
octave, vérification 9
`octaveCheck` 9
`octaveCheck` 586
octaves absolues 1
octaviation 18
`oddFooterMarkup` 389
`oddHeaderMarkup` 389
`oneVoice` 141
`opus` 386
oratorio 254
orchestre, cordes 291
orgue, marque de pédale 102
orgue, marque de pédale d' 553
orientation 414
ornementation baroque 102, 553
ornementation, symboles 102
ornements 95
ornements et paroles 249
ossia 164, 169, 468
`ottava` 18
`ottava` 586
`outer-margin` 419
`outside-staff-horizontal-padding` 446
`outside-staff-padding` 446
`outside-staff-priority` 446
ouvert 102, 553
`overrideProperty` 586
`overrideTimeSignatureSettings` 586
overriding properties within text markup 549

P

`p` 104
`pad-around` 210
`pad-markup` 210
`pad-to-box` 210
`pad-x` 210
padding 471
padding text 528
padding text horizontally 529
page breaks, forcing 386
page layout 452
page, format 413
`page-breaking` 420
`page-breaking-system-system-spacing` 420
`page-count` 420
`page-spacing-weight` 421
`pageBreak` 586
pages, saut 451
`pageTurn` 587
`palmMute` 587
`palmMuteOn` 587
Pango 216
`paper-height` 414
`paper-width` 417
papier, taille 413
`parallelMusic` 151
`parallelMusic` 587
parenthèses, altération entre 5
parenthèses, notes entre 188
`parenthesize` 188, 587
parlato 264
parlato, tête de note 30
paroles 220
paroles assignées à une voix 141
paroles divisées (reprises) 245
paroles et liaison de prolongation 243
paroles et ligatures 73
paroles et mélodies 223
paroles et ornements 249
paroles et répétition 239
paroles, alignement sur la mélodie 221
paroles, blanc 49
paroles, gestion de l'espacement 237
paroles, identificateurs 231
paroles, positionnement 232
paroles, reprise avec alternative 242
paroles, saut de notes 49
paroles, variables 231
`partcombine` 148
`partcombine` 587
`partcombineForce` 587
`partial` 62
partie vocale 254
parties, combiner des 148
partition incluse dans un markup 214
paths, drawing 537
paysage, papier 414
pédale sostenuto 285
pédale sustain 285
pédale sustain, style 285
pédale, indication combinée de 285
pédale, indication graphique de 285
pédale, indication textuelle de 285
pédale, styles d'indications de 285
pédales de harpe 290
pédales de piano 285
pédales, diagrammes pour harpe 290
`pedalSustainStyle` 285
`percent` 133
percussion, portée 154
percussions 326, 327
personnage, indication 258
petite note 95
petites notes 174, 181, 334
petites notes, formater des 178
Petrucci 354
phrasé, liaisons de 114
phrasé, pour des paroles 227
`phrasingSlurDashed` 114
`phrasingSlurDashPattern` 115
`phrasingSlurDashPattern` 587
`phrasingSlurDotted` 114
`phrasingSlurDown` 114
`phrasingSlurHalfDashed` 115
`phrasingSlurHalfSolid` 115
`phrasingSlurNeutral` 114

`phrasingSlurSolid` ... 114
`phrasingSlurUp` ... 114
`phrygian` ... 16
phrygien ... 16
`piano accidentals` ... 23
piano cautionary, style d'altérations ... 24
piano et altérations ... 23, 24
piano et pédales ... 285
piano, nuances entre les portées ... 279
piano, style d'altérations ... 23
piano, système ... 155
piano, système pour ... 279
`piano-cautionary` ... 24
`PianoStaff` ... 279, 281
`piece` ... 386
pied de page ... 389
`pipeSymbol` ... 92
`Pitch_squash_engraver` ... 68
`pitchedTrill` ... 123
`pitchedTrill` ... 587
pizzicato, Bartók ... 294
pizzicato, snap ... 294
placing horizontal brackets around text ... 536
placing parentheses around text ... 536
placing vertical brackets around text ... 534
plein, trait de liaison ... 112, 114
plusieurs lignes de texte ... 209
plusieurs mouvements ... 381
plusieurs pages de texte ... 215
`poet` ... 386
point ... 37
point d'arrêt ... 102, 116, 553
point d'orgue ... 94, 102, 553
point d'orgue et silence multimesure ... 52
point d'orgue sur une barre de mesure ... 197
`pointAndClickOff` ... 587
`pointAndClickOn` ... 587
pointe ... 553
pointée, note ... 37
pointillé, trait de liaison ... 112, 114
polices, choix par défaut ... 218
polymétrie ... 64
polymétrie et ligature ... 65
polymétrique, partition ... 465
polyphonie ... 144
polyphonie, portée simple ... 141
ponctuation ... 220
ponctuation et paroles ... 220
portato ... 102, 553
portée à la française ... 164
portée à quatre mesures ... 426
portée de batterie ... 154
portée de percussion ... 154
portée de tablature ... 154
portée multiple ... 155
portée rhytmique ... 154
portée simple ... 154
portée vide ... 168
portée, début ... 163
portée, définition de la taille ... 423
portée, fin ... 163
portée, initialisation ... 154
portée, instanciation ... 154
portée, lignes de ... 162
portée, nouvelle ... 154
portée, reprise ... 162
portée, suspension ... 162
portée, transcription de grégorien ... 154
portées pour instrument à clavier ... 279
portées pour instrument à clés ... 279
portées pour piano ... 279
portées rythmiques, masquage ... 169
portées, espacement ... 432
portées, groupe de ... 155
portées, nombre variable de ... 164
portées, regroupement ... 155
portées, regroupement et imbrication ... 159
positionnement des paroles ... 232
positionnement des silences multimesure ... 53
`postscript` ... 212
pouce ... 102, 553
pouce, doigté ... 183
pourcent, répétition ... 133
poussé, indication d'archet ... 292
pousser l'archet ... 553
`pp` ... 104
`ppp` ... 104
`pppp` ... 104
`ppppp` ... 104
précaution, altération de ... 5
`predefinedFretboardsOff` ... 321
`predefinedFretboardsOn` ... 321
première fois ... 124
présentation mensurale ... 158
`print-all-headers` ... 388, 421
`print-first-page-number` ... 421
`print-page-number` ... 421
prolongateur ... 230
prolongation de texte ... 195
propriétés ... 472
psalmodie ... 266
pulsation, regroupement ... 77
putting space around text ... 528

Q

quadrillage temporel ... 190
quart de ton ... 7
quart de ton en MIDI ... 406
quarts de ton ... 5
`quotedCueEventTypes` ... 177
`quotedEventTypes` ... 177
`quoteDuring` ... 174, 178
`quoteDuring` ... 587

R

`r` ... 47
`R` ... 50
ragged, aligné ... 412
`ragged-bottom` ... 415
`ragged-last` ... 418, 451
`ragged-last-bottom` ... 415
`ragged-right` ... 418, 451
`raise` ... 207
raising text ... 529
Référence des propriétés internes ... 463
referencing page labels in text ... 552
referencing page numbers in text ... 550

registres, symboles pour accordéon 286
réglage des ligatures automatiques 73
regroupement automatique de parties 148
relatif 2
relative 2, 12, 282
religieuse, musique 266
reliure 418
removeWithTag 587
renaissance, musique 158
rendu, interfaces de 470
repeatCommands 129
repeatTie 45
repère manuel 94
repère, format 94
repère, indication de 93
repère, personnalisation 94
repère, style 94
répétition courte 133
répétition de mesure 133
répétition et liaison de tenue 45
répétition, pourcent 133
répétitions 86, 124
réplique, fin 179
répliques 174
répliques, formatage 178
reprise avec alternative 124
reprise avec alternative et paroles 242
reprise avec levée 125
reprise courante 124
reprise de portée 162
reprise et anacrouse 125
reprise et liaison 129
reprise et liaison de prolongation 127
reprise et paroles 239
reprise manuelle 129
reprise, expansion 131
reprises 86, 124
reprises ambigües 129
reprises avec alternatives et liaisons de prolongation 45
reprises développées 406
reprises imbriquées 129
resetRelativeOctave 587
respiration, indication 116
rest 47
rest-event 177
retouche des notes d'ornement 97
retour au propriétés par défaut de la métrique 56
retrograde 588
revertTimeSignatureSettings 588
rfz 104
rgb, couleur 187
rgb-color 187
RhythmicStaff 154
right aligning text 529
right-align 206
right-margin 418
rightHandFinger 323, 588
rôle 248
rôle, indication 258
rotating text 530
rounded-box 210
rvb, couleur 187
Rythme en MIDI 406
rythmique d'une mélodie 68

S

s 49
sacredHarpHeads 33
sacredHarpHeadsMinor 33
saisir des accords 344
SATB 254
saut de ligne 424
saut de ligne et ligature 71
saut de page 451
sauts 117
sauts de durée 49
sauts de ligne 84
sauts de ligne et cadences 64
sauts de ligne et musique non mesurée 64
sauts de ligne réguliers 426
sauts de page et cadences 64
sauts de page et musique non mesurée 64
scalable vector graphics 402
scaleDurations 43, 64
scaleDurations 588
scaling markup 539
scaling text 531
score-markup-spacing 417
score-system-spacing 417
scoreTitleMarkup 389
Scottish highland bagpipe 334
script et silence multimesure 52
seconde fois 124
segno 85, 94, 102, 553
segno sur une barre de mesure 197
selection de la taille (notation) 182
sesqui-bémol 8
sesqui-dièse 8
set 73
set-accidental-style 20
set-octavation 18
setting extent of text objects 552
setting horizontal text alignment 523
setting subscript in standard font size 514
setting superscript in standard font size 514
sf 104
sff 104
sfz 104
shift note 144
shiftDurations 588
shiftOff 144
shiftOn 144
shiftOnn 144
shiftOnnn 144
short-indent 172, 419
show-available-fonts 218
showFirstLength 401
showKeySignature 334
showLastLength 401
showStaffSwitch 283
silence d'église 52
silence multimesure 47
silence multimesure et point d'orgue 52
silence multimesure, ajout de texte 52
silence multimesure, contraction 51
silence multimesure, étiquette 52

silence multimesure, expansion ... 51
silence multimesure, positionnement ... 53
silence multimesure, script ... 52
silence, spécification du positionnement vertical ... 48
silences ... 47
silences d'espacement ... 49
silences invisibles ... 49
silences multimesure et doigtés ... 54
silences multimesures ... 50
silences, collisions entre ... 54
silences, condenser les ... 54
silences, décalage automatique ... 144
silences, mesure entière ... 50
silences, musique ancienne ... 356
simple text strings ... 516
simple text strings with tie characters ... 543
simultanées, liaisons d'articulation ... 112
simultanées, liaisons de phrasé ... 114
`skip` ... 49
`skipTypesetting` ... 401
slashed digits ... 550
`slurDashed` ... 112
`slurDashPattern` ... 112
`slurDashPattern` ... 588
`slurDotted` ... 112
`slurDown` ... 112
`slurHalfDashed` ... 112
`slurHalfSolid` ... 112
`slurNeutral` ... 112
`slurSolid` ... 112
`slurUp` ... 113
`small` ... 182, 206
`smaller` ... 204, 206
snap pizzicato ... 294
solo ... 148
sos ... 285
sostenuto, pédale ... 285
`sostenutoOff` ... 285
`sostenutoOn` ... 285
soufflet ... 104
soufflet, ligature ... 83
souligné ... 204
sourdine ... 102, 553
`southernHarmonyHeads` ... 33
`southernHarmonyHeadsMinor` ... 33
`sp` ... 104
`spacing` ... 448
`spacingTweaks` ... 588
spécification d'un repère ... 94
`spp` ... 104
Sprechgesang ... 264
staccatissimo ... 102, 553
staccato ... 102, 553
stacking text in a column ... 520
`Staff.midiInstrument` ... 403
`Staff_symbol_engraver` ... 168
standard, taille des notes ... 183
`start-repeat` ... 129
`startGroup` ... 192
`startStaff` ... 163, 164
`startTrillSpan` ... 122
`Stem` ... 283
stem, direction ... 188
stem, down ... 188
stem, neutral ... 188
stem, up ... 188
`stem-spacing-correction` ... 448
`stemDown` ... 188
`stemLeftBeamCount` ... 81
`stemNeutral` ... 188
`stemRightBeamCount` ... 81
`stemUp` ... 188
`stopGroup` ... 192
`stopStaff` ... 163, 164, 168
`stopTrillSpan` ... 122
`storePredefinedDiagram` ... 316
`storePredefinedDiagram` ... 588
`stringTunings` ... 312
`StringTunings` ... 302
style d'accidentelle ... 20
style d'altération *modern* ... 23
style d'altération *neo-modern-cautionary* ... 24
style d'altération *neo-modern-voice* ... 24
style de repère ... 94
style de trait, liaison ... 112, 114
`styledNoteHeads` ... 588
styles de métrique ... 55
styles de voix ... 143
styles, tête de note ... 30
`sub` ... 205
subdivision de ligature ... 77
subscript text ... 517
substitution de doigt ... 183
`subsubtitle` ... 386
`subtitle` ... 386
`super` ... 205
superscript text ... 517
`sus` ... 345
suspension de portée ... 162
sustain, pédale ... 285
sustain, style de pédale ... 285
`sustainOff` ... 285
`sustainOn` ... 285
SVG, format de sortie ... 402
syllabes, durée automatique ... 223
symbole arabe d'un demi-bémol ... 374
symbole de portée ... 162
symboles d'ornementation ... 102
symboles non musicaux ... 211
synchronisation des notes d'ornement ... 99
syntaxe du mode markup ... 202
`system-count` ... 420
`system-separator-markup` ... 421
`system-system-spacing` ... 417
système ... 155
système choral ... 155
système pianistique ... 155
système, début de ... 155
système, grand ... 155
système, indicateur de séparation ... 160
systèmes imbriqués ... 159
`systems-per-page` ... 420

T

`tabChordRepetition` ... 588
tablature ... 154, 294
tablature et glissando ... 300
tablature et indication d'harmoniques ... 300

tablature pour banjo ... 294
tablature pour guitare ... 294
tablature, accordages prédéfinis ... 302
tablatures par défaut ... 296
tablatures personnalisées ... 302
tablatures pour banjo ... 326
tablatures, bases ... 296
`TabStaff` ... 154, 296
`TabVoice` ... 296
tag ... 396
`tag` ... 588
`tagline` ... 386
tagline ... 389
taille des notes ... 182
talon ... 553
`taor` ... 334
taqasim ... 376
`teaching` ... 25
teaching, style d'altérations ... 25
`teeny` ... 182, 206
`tempo` ... 59
temporel, quadrillage ... 190
temps, gestion du ... 100
tenor, clef ... 14
tenue et nuances successives ... 105
tenuto ... 102, 553
tessiture ... 27
tête de note allongée ... 36
tête de note en losange ... 293
tête de note et improvisation ... 36
tête de note, Aiken ... 33
tête de note, allure ... 33
tête de note, apprentissage ... 31
tête de note, Christian Harmony ... 33
tête de note, Funk ... 33
tête de note, Harmonica Sacra ... 33
tête de note, harpe sacrée ... 33
tête de note, Walker ... 33
têtes de note ... 182
têtes de note, musique ancienne ... 355
têtes de note, spéciales ... 30
têtes de note, styles ... 143
`text` ... 285
text columns, left-aligned ... 527
text columns, right-aligned ... 530
text, top-level ... 201
texte ajouté ... 202
texte en colonnes ... 208
texte en préambule ... 201
texte et alternative ... 130
texte et extenseur ... 195
texte et rembourrage ... 210
texte hors marges ... 195
texte indépendant ... 201
texte indiquant le nombre de mesures vides ... 52
texte isolé ... 201
texte justifié ... 209
texte multi-ligne ... 208
texte sur plusieurs lignes ... 209
texte, alignement ... 206
texte, alignement horizontal ... 206
texte, alignement vertical ... 207
Texte, autres langues ... 194
texte, centrage sur la page ... 208
texte, décoration ... 210
texte, encadrement ... 210
texte, maintien dans les marges ... 195
texte, mise en forme des extenseurs ... 195
texte, mise en forme des prolongations ... 195
texte, taille ... 204
`textLengthOff` ... 195
`textLengthOn` ... 52
`textLengthOn` ... 195
`textLenthOff` ... 52
`textSpannerDown` ... 195
`textSpannerNeutral` ... 195
`textSpannerUp` ... 195
`thumb` ... 183
`tieDashed` ... 45
`tieDashPattern` ... 588
`tieDotted` ... 45
`tieDown` ... 45
`tieNeutral` ... 45
`tieSolid` ... 45
`tieUp` ... 45
`time` ... 54, 73
`times` ... 39, 64
`timeSignatureFraction` ... 64
`tiny` ... 182, 206
tiré, indication d'archet ... 292
tirer l'archet ... 553
tiret, trait de liaison ... 112, 114
tirettes d'accordéon, symboles ... 286
`title` ... 386
titres ... 389
`tocItem` ... 588
`Top` ... 1
top-level text ... 201
`top-margin` ... 415
`top-markup-spacing` ... 417
`top-system-spacing` ... 417
tracé d'objets graphiques ... 210
trait d'union ... 230
transcription de musique ancienne ... 158
`translate` ... 208
`translate-scaled` ... 208
translating text ... 530, 531
transparentes, notes ... 186
`transpose` ... 2, 3, 10, 12
`transposedCueDuring` ... 180
`transposedCueDuring` ... 588
transpositeur, instrument ... 10
transposition ... 10
`transposition` ... 19, 174
`transposition` ... 588
transposition des hauteurs ... 10
transposition et clef ... 14
transposition et diagramme de fret ... 314
transposition et MIDI ... 19
transposition et mode octave relative ... 3
transposition, instrument ... 19
tre corde ... 285
`treCorde` ... 285
`tremolo` ... 135
tremolo, cross-staff ... 137
trémolo, indication de ... 136
trémolo, ligatures de ... 135
`tremoloFlags` ... 136
`triangle` ... 211
`trill` ... 122

trille ... 102, 553
trilles ... 122
trilles avec hauteur explicite ... 123
trilles avec hauteur explicite et altération ... 123
triolets ... 39
triolets, formatage ... 39
`tupletDown` ... 39
`tupletNeutral` ... 39
`TupletNumber` ... 40
`tupletNumberFormatFunction` ... 39
`tupletSpannerDuration` ... 39
`tupletUp` ... 39
turc, noms des notes ... 379
turque, musique ... 378
`tweak` ... 588
`two-sided` ... 418

U

U.C. ... 285
una corda ... 285
`unaCorda` ... 285
`underline` ... 204
underlining text ... 518
une pause par mesure ... 50
`unfold` ... 131
`unfoldRepeats` ... 588
`unHideNotes` ... 186
Unicode ... 399
UTF-8 ... 399

V

varcoda ... 102, 553
variables ... 385
variables, utilisation de ... 394
variante ... 164
variante rythmique ... 245
Vaticana, Editio ... 354
`VaticanaStaff` ... 154
VaticanaStaffContext ... 370
VaticanaVoiceContext ... 370
vents ... 330
vérification d'octave ... 9
vérification des limites de mesure ... 92
vertical positioning of dynamics ... 106
vertical spacing ... 452
vertical, espacement ... 432
vertically centering text ... 531
visibilité des hampes ... 188
vocalise ... 227
`voice` ... 20, 22
`Voice` ... 141
voice, style d'altérations ... 22
`voiceOne` ... 141
voix ... 141
voix entre deux portées ... 283
voix multiples ... 144
voix multiples et altérations ... 23, 24
voix, décalage ... 144
voix, division ... 256
voix, styles ... 143
volta ... 124

W

Walker shape, tête de note ... 33
`walkerHeads` ... 33
`walkerHeadsMinor` ... 33
`whichBar` ... 88
`with-color` ... 186
`withMusicProperty` ... 588
`wordwrap` ... 209
`wordwrap-lines` ... 215

X

x11, couleur ... 187
x11, couleurs ... 186
`x11-color` ... 186, 187
`xNote` ... 588
`xNotesOn` ... 588