

# music21 コードシンボル完全ガイド

## 目的

このドキュメントは、music21 の `harmony.ChordSymbol()` がパース可能なコード表記を一度で把握できるように、構文規則・正式略号・落とし穴・実例をすべてまとめたものです。GPT ベースのコード／MIDI 生成時に参照し、表記ゆれによる `ValueError` を根絶することを目指します。

## 1. フル正規表現 (簡易読み下し)

```
^                # 文字列全体がコード名のみ
(?P<root>[A-G])    # ① ルート
(?P<accidental>[#-]?) # ② 変化記号 (# = # , - = ♭)
(?P<quality>       # ③ コード品質 (省略可)
    maj|min|M|m|dim|o|
    aug|\+|sus2|sus4|
    mM|+M7|power
)?
(?P<extension>7|9|11|13)? # ④ 拡張度数 (省略可)
(?P<alter>([#b]\d+)*?) # ⑤ オルタレーション (#9, b5...) 複数可
(?P<add>(add\d+)?) # ⑥ add 系 (add9 など)
(?P<omit>(omit\d+)?) # ⑦ omit 系 (omit5 など)
(?P<resus>(sus2|sus4)?) # ⑧ 後置 sus (再指定用 / 省略可)
(?P<slash>        # ⑨ スラッシュ (分数)
    /[A-G][#-]?
)?
$
```

## ポイント

- 順序は固定：ルート → 変化記号 → 品質 → 拡張 → 修飾 → add/omit → 後置 sus → スラッシュ。
- 括弧・カンマ・空白は一切不可。
- ルート／ベースのフラットは **\*\*ハイフン\*\***、テンションのフラットは とする。
- 特殊ラベルとして、N.C. / NoChord は **Rest** 扱いで `note.Rest()` を挿入。

## 2. 品質 (quality) 一覧. 品質 (quality) 一覧

記号	意味	例	備考
(空)	メジャー三和音	C	デフォルト
maj/M	メジャー7 系	Cmaj7 CM9	maj+extension で 7th 内包
m/min	マイナー	Cm, Cm7	
mM	マイナー・メジャー7	CmM7	
dim/o	ディミニッシュ	Cdim, Co7	
aug/+	オーギュメント	C+, C7+	
sus2	サス2	Csus2	
sus4	サス4	Csus4, C7sus4	

power	パワーコード	Cpower	root+5th
-------	--------	--------	----------

### 3. 拡張度数と暗黙ルール

- 7 = ドミナント 7th。
- 9/11/13 を書くと 必ず **7th** を含む (例:C9 = C7+9)。
- メジャー系 9/11/13 は maj 品質を併記 → Cmaj9, Cmaj13。

### 4. テンション修飾 (alter)

- フォーマット : # / b + 度数数字。  
例:C7b9#11 G13#9#11
- 複数可、順序自由。推奨は度数昇順 (b9#11#13 ... )。

### 5. add / omit

- triad+ 単音追加 : Cadd9, Cmadd11。
- 省略 : C7omit3, Cmaj9omit5。

### 6. sus 再指定

- 7th/9th などの後にも sus4/sus2 を再度置ける : C9sus4。

### 7. 分数コード (インヴァージョン)

- 末尾に /Bass。ベースにも #/- を許可 : C/E, Cm7/B-。

### 8. 非和声音区間を示す方法

- **NoChord** : harmony.NoChord() もしくは "N.C." ラベル。
- 休符 : 和音ラベルを置かず note.Rest をインサート。

### 9. よくあるエラーと対策

誤表記	エラー原因	正表記
Am(add9)	括弧 NG	Amadd9 or Am9
C7(b9,#11)	括弧・カンマ NG	C7b9#11
E7-9	テンションに '-' 使用	E7b9
Gm7-5	同上	Gm7b5
C7#9-13	同上	C7b9#13
A7-9	同上	A7b9
Fmaj9	maj9 非対応	Fmaj7add9
Fmaj13	maj13 非対応	Fmaj7add13
B-maj9#11	複合テンション誤	B-maj7#11add9 or B-maj13#11
N.C. / NoChord	コードシンボル文字列では休符扱い不可	null or omit label + Rest

### 10. 参考例 (valid / invalid). 参考例 (valid / invalid)

✓ C, Cm, C7, Cmaj7, C9#11, C6add9, C7b9#13, C11, C13#11, C7sus4/E, Cpower

✗ C7(b9), Am(add9), Cmaj9(#11), Bbmaj7#11, Ebmaj7#11, C7b5(#9)

### 11. ベストプラクティス

1. ルート **accidentals** を必ず `` で統一。
2. 拡張 9/11/13 を書くなら 7th は省略しない (暗黙含有を理解)。
3. add と alter の混在時は **alter → add** の順に並べる : Cmaj7#11add13。
4. JSON 等に休符を書かず、生成側で Rest / NoChord を自動挿入。

5. コード自動生成 AI では、この正規表現を **事前バリデータ** として実装し、不合格なら修正ルーチンへフォールバック。

## 12. ポリコード (Polychord)

ポリコードは異なるルートをもつ2つの三和音を同時に鳴らす手法です。

- 表記方法：<コード A>/<コード B> ではなく、<コード A> over <コード B>または<コード A>+<コード B>と明示する。  
例：Cmaj7 over G7 または Cmaj7+G7。
- music21 では `harmony.ChordSymbol('Cmaj7')` と `harmony.ChordSymbol('G7')` のパートを重ねるか、テキストオブジェクトで表記。
- 自動生成時は単一文字列ではなく、**2つの ChordSymbol オブジェクト**を同時配置。XML/譜例注記としては Cmaj7/G7 を印字しても良い。

## 13. 特殊六の和音 (Sixth Chord Variations)

6th コードには以下のバリエーションがあります。

表記	解釈	例
C6	大6度追加 triad	C-E-G-A
Cm6	短6th (m triad +6)	C-E <sup>b</sup> -G-A
C6/9	6th +9th (暗黙7thなし)	C-E-G-A-D
Cmaj6	maj6 と同義 (C6 推奨)	C-E-G-A
C+6	オーギュメント +6	C-E-G <sup>#</sup> -A

注意：add6 表記は可能ですが、Cadd6 より C6 が一般的。

## 14. ジャズの Slash-Chord 変則

ジャズでは分数コードに追加修飾を組み合わせた場合があります。

- Cm7/G → Cマイナー7のベースをGに。
- G13alt/B- → ドミナント13thオルト (#9<sup>b</sup>13など) をB<sup>b</sup>ベースで：G13#9<sup>b</sup>13/B-。
- D7#9#5/F# → シャープ9, シャープ5付きD7をF#ベースで。
- 表記は `` の形式で一貫させる。

## 15. ボイシングと音域ガイド

ベースパートやアンサンブルでコードを演奏するとき、**構成音をどのオクターブに配置**するかが和声のクリアさや響きの豊かさを左右します。ここでは代表的なボイシング例と、音域を活かすコツをまとめます。

### 15.1 代表的ボイシング例

- **メジャー7th (三和音+長7)**
  - 根音下位配置：C3, E3, B3, G4
    - ◆ 低域にルート、上位にテンションを配置し、開放感を演出。
  - 近位クローズド：C4, E4, G4, B4
    - ◆ ジャズ・ピアノで多用。どのレンジでも和音がまとまりやすい。
- **マイナー9th (マイナー7+9)**
  - ドリアン感強調：D3, F3, C4, E4
    - ◆ ルート→3度→<sup>b</sup>7→9度の順。メロディを阻害しにくい。
  - クローズボイシング：D4, F4, A4, C5, E5
    - ◆ 上位に音を集中。ソロやリフのバックで効果的。

- **ドミナント 13th**

- オープンヴォイスイング：G2, B2, F3, E3, A3
  - ◆ ルート下、3度→7度→9度→13度。ベースと重ならない。
- ジャズギター向け：G3, F3, E3, A3, B3
  - ◆ 低いルート省略し、テンション主体に。

## 15.2 ベースとの干渉回避

- ベース担当パートが同じルートを担う場合、コードの低域は1オクターブ分ずらす。
- なるべく中～高域 (C4 以上) に3度・7度・テンションを集める。
- 原音の重複を避けることで、各パートの帯域がクリアに分離される。

## 15.3 楽器別アレンジ注意点

- **ピアノ**: 広域レンジを活かし、左手はルート／7th、中～右手でテンションを演奏。
- **ギター**: 押さえやすいポジションを優先。バレーコードが難しい場合はテンションを抜き、小型ボイスイングを選択。
- **管弦楽**: ストリングスではサステインを活かし、金管では強弱のコントラストを考慮。

## 16. インヴァージョン (転回形) の詳細ルール

### 16.1 セカンダリ・セブンスの転回形

- **V6/5 表記**: ドミナント 7th の第1転回形 (3度上をベース) と第2転回形 (5度上をベース) を区別。
  - G7/B = G7 の第1転回形 (B in bass)
  - G7/D = G7 の第2転回形 (D in bass)

### 16.2 6th, 9th の複数転回形

- 複数量数の転回例: C6/9/E は「C6add9」の E 構成。
  - 実装では ChordSymbol('C6') + ChordSymbol('Cadd9') を重ね、note.Rest 部分を E 付近に配置。

### 16.3 テンション付き転回形

- テンションを含む場合もベースを明示:
  - C7b9/D = C7b9 の第1転回形
  - Cmaj7#11/G = Cmaj7#11 の分数コード
- **music21** では ChordSymbol('C7b9') と Note オブジェクトで D を別パートに配置。

## 17. モーダル・インターチェンジと借用和音

### 17.1 サブドミナント借用

- メジャーキー IV をマイナー iv に借用:
  - F → Fm (Cキー)

### 17.2 ♭VII, ♭IIImaj7

- イオニアン (I) からエオリアン借用:
  - B- (♭VII)、E-maj7 (♭IIImaj7)

### 17.3 副代理進行

- 二次ドミナントや半減進行:
  - Dm7b5-G7/V-D7 など
- コードシンボル記述例: harmony.ChordSymbol('D#dim45') で拡張 G7/V を表現。

## 18. リハーモナイゼーション手法

### 18.1 テンションスワップ

- add13 と #11 の入れ替え:
  - Cmaj7add13 ↔ Cmaj7#11

### 18.2 クロマチックベースライン

- スムーズな下降:
  - Cmaj7 → Cmaj7/B- → B-maj7/A- → A-maj7

### 18.3 ポリリズムスラッシュコード

- 複合分数:
  - D7#9/E+G (D7#9 over E、かつ G トーンを強調)

## 19. 拡張／非調性コード

### 19.1 オルト (altered) コード

- 全テンション許容:
  - G7alt = G7b9#9#11b13
  - music21 では各テンションを手動追加後、ChordSymbol('G7alt') とタグ付け可能

### 19.2 複数テンション複合

- Cm13b5#9 = Cm7b5 に 13th と #9
- Cmaj13add#11 など自由に連結

### 19.3 カスタムコードネーム登録

- ChordSymbol.customGlyphMap['Hdim'] = 'C#dim' のように独自定義

## 20. 楽譜レンダリングと最適化

### 20.1 MuseScore/LilyPond 互換性

- power→5 自動変換
- B- 表記保持のため、エクスポート時に置換テーブル適用

### 20.2 コード折り返し／改行

- 7音以上のテンション列は を入れて二行表示

### 20.3 歌詞併記フォーマット

- : 各小節先頭にコード行を追加
- タブ → {'align': 'center'} プロパティ指定

## 番外編: pretty\_midi 特集

Python 実装で MIDI データを直接操作できる pretty\_midi ライブラリを、music21 と連携して使いこなすための詳細ガイドです。

### 番外編.1 概要とインストール

- 概要: pretty\_midi は MIDI データの読み書き、ノート編集、コントロールチェンジ、ピアノロール生成などを簡潔な API で提供します。
- インストール:  
pip install pretty\_midi
- 
- 依存: numpy、mido。

### 番外編.2 music21 から PrettyMIDI 変換

```
import io
import pretty_midi
from music21 import converter
```

```
# music21 オブジェクトを MIDI バイトに変換
m21_stream = converter.parse('example.musicxml')
midi_bytes = m21_stream.write('midi')
```

```
# BytesIO 経由で PrettyMIDI 読み込み
pm = pretty_midi.PrettyMIDI(io.BytesIO(midi_bytes))
```

- 注意: write('midi') はファイルパス文字列も受け取るが、BytesIO 経由でメモリ操作可能。

### 番外編.3 ノート・ベロシティ編集

```
for instrument in pm.instruments:
    for note in instrument.notes:
        # ベロシティを正規化
        note.velocity = int((note.velocity / 127) * 100) + 20
        # 持続時間の半分に短縮
        note.end = note.start + (note.end - note.start) * 0.5
```

- メモ: note.velocity は 0-127。動的レンジ調整に便利。

### 番外編.4 コントロールチェンジとプログラムチェンジ

```
# プログラムチェンジ: 0 = Acoustic Grand Piano
```

```
for inst in pm.instruments:
    inst.program = 0
```

```
# コントロールチェンジ追加
inst.control_changes.append(
    pretty_midi.ControlChange(number=7, value=100, time=0)
)
```

- **number:** MIDI CC 番号 (7=Volume, 10=Pan, 64=Hold Pedal など)。

### 番外編.5 テンポ・タイムシフト操作

```
# テンポ変更情報取得
```

```
tempo_times, tempo_vals = pm.get_tempo_changes()
```

```
# 全ノートを 0.5 秒後にシフト
```

```
for inst in pm.instruments:
```

```
    for note in inst.notes:
```

```
        note.start += 0.5
```

```
        note.end += 0.5
```

- get\_tempo\_changes() は (times, tempi) のタプルを返す。

### 番外編.6 ピアノロールと可視化

```
# piano_roll: (128, time_steps) の numpy array
```

```
fs = 100 # frames per second
```

```
roll = pm.get_piano_roll(fs)
```

```
print(roll.shape)
```

- 横軸フレーム、縦軸 MIDI ピッチ。機械学習の前処理に活用。

### 番外編.7 ファイル出力

```
pm.write('output.mid')
```

# さらに wave 化する場合は MIDI -> サウンドフォント経由で synth 必須

## 番外編.8 ヒントとトラブルシュート

- 時間精度: Float 時間は丸め誤差に注意。必要に応じて round()。
- チャンネル管理: pm.instruments の is\_drum フラグでドラム扱い。
- ポリフォニー解析: get\_piano\_roll() の閾値で同時発音数を解析可能。

© 2025 Harusan x o3 Harusan x o3 Harusan x o3 Harusan x o3 Harusan x o3

## music21におけるコードシンボル記述完全チェックリスト

music21でコード記号 (ChordSymbol) を使う際には、**決められた表記ルール**に従う必要があります。以下では、music21が正しく解釈・処理できるコードシンボルの形式を整理し、対応している/いないコード表記やテンション記号・オンコード (スラッシュ表記) の書式を確認します。最後に、これらの情報に基づいた**コード記述チェックリスト**をまとめます。

### 1. music21で解釈されるコードシンボルの基本形式

**基本フォーマット:** Root [Accidental] [Chord Quality/Kind] [Extension] [Alterations/Additions/Omissions] [/Bass]

- **ルート音 (Root):** コードのルート音は A ~ G の英字で指定し、大文字で書きます。必要に応じてシャープやフラットを付加します。シャープは「#」、フラットは「b」またはハイフン「-」で表記できます。例えば「F#」「E-」はそれぞれ F# (ヘ長調のシャープ) と E $\flat$  (変ホ) のルートを意味します。
  - 例: C (ハ長調)、F# (嬰ヘ長調)、B- または B $\flat$  (変ロ長調)。
- **コードの種類 (Quality/Kind):** ルートに続けて、メジャー・マイナー等のコード種別を指定します。コード種別は一般的な略号で表記でき、music21は**複数の表記ゆれを同義として認識**します。例えば「major」は省略形なし (デフォルト) または「maj」「M」で表せます。「minor」は小文字の「m」がデフォルトですが「min」やジャズ表記の「-」も認識されます (ただし「-」単体はフラットと紛らわしいため後述)。
  - **メジャーコード:** 何も付けないか、明示する場合は maj や M を使用 (例: C = C メジャー、Cmaj と同記述可能)。
  - **マイナーコード:** m (デフォルト) または min で指定 (例: Cm = C マイナー、Cmin も可)。ジャズ符号で「C-」と書くと **C** マイナーと解釈される場合があります (例: D-7 を D マイナー7 と判別)、ハイフンはフラット記号とも衝突するため注意が必要です (詳細は後述)。
  - **オーギュメント (増和音):** aug または記号の + で指定 (例: C $\sharp$ aug、C+)。music21は「+」と「aug」の両方を認識します。
  - **ディミニッシュ (減和音):** dim で指定 (例: Cdim)。完全五度を減五度に下げたディミニッシュトライアドを意味します。7th を付けると減七の和音になります (例: Cdim7)。なお、度数表記で「 $\circ$ 」(度記号) は ASCII ではないため、dim を使うのが無難です。
  - **ハーフディミニッシュ (半減七の和音):** 一般的な表記の **m7 $\flat$ 5** で指定します。music21は「m7 $\flat$ 5」を半減七の和音 (いわゆるマイナー7( $\flat$ 5)) として認識しま

す。例えばGm7b5はGø7 (ソの半減七) のことです。ø記号は入力では使わず、m7b5と表記してください。

- **その他のコード種別:** 上記以外にも **sus (サスペンデッド)** や**特殊コード**などを認識します。susについては後述しますが、例えばCsus4 (Cサスフォー) やCsus2 (Cサスツー) が指定可能です。またmusic21には特殊なコード種別も定義されており、例えば**パワーコード** (5度のみの和音) はCpowerと表記できます。
- **拡張音 (Extensions):** 7thや9thなど和音の拡張音を表す数字をコード種別の後に付けます。基本的なルールは**数字だけを書いた場合、その音までの奇数拡張音すべてを含む**というものです。
  - **7th:** 単に「7」と書くと**ドミナント 7th** (短7度の和音) を意味します。例えばC7はC-E-G-B<sup>b</sup>の構成 (C7=属七の和音) です。一方、長7度の場合は「maj7」(または大文字M7) と記します。Cmaj7はC-E-G-B (Cメジャー7) を意味します。
  - **9th, 11th, 13th:** 「9」「11」「13」と書くと7thコードにそれぞれ9度・11度・13度のテンションが**含まれる拡張和音**を表します (一般的なリードシートの約束と同様)。例えばC9はC7に9th (D音) を加えた和音 (C-E-G-B<sup>b</sup>-D) です。C13であれば原則として7度・9度・11度・13度まで含む13thコードを意味します (11thはしばし省略されることもありますが、music21では**基本的に含めて構成**します)。music21はmajor系やminor系の拡張も区別しており、Cmaj9ならCmaj7に9thを加えた和音、Cm11ならCm7に11thを加えた和音、というように解釈されます。
  - **6th:** 「6」は特殊で、**6の和音** (長6度を加えた和音) を意味します。例えばC6はC-E-G-A (長6の和音)、Cm6はC-E<sup>b</sup>-G-A (短6の和音) となります。これは「add6」と同義ですが、一般に「6」で表します。
  - **minor-major 系:** マイナーコードに長7度を加えた和音は「mM7」(またはmMaj7/minmaj7) のように表せます。music21でもこの表記を認識し、例えばCmM7 (Cマイナーに長7度) やCmM9 (長7度と9度を持つ和音) を正しく処理します。
- **テンション・追加音 (Additions/Alterations):** コード種別や拡張音に加えて、特定のテンションや変更音を明示的に追加・変更・省略する場合は以下のキーワードや記号を用います:
  - **add 系:** 7thを含まない和音に9th等を付加したい場合、「add 数」で指定します。例: Cadd9はCメジャーに9th (D音) だけを追加した和音です (C-E-G + D)。同様にCmadd9はCmに9thを追加、Cadd2やCadd4も認識され、それぞれ2度音や4度音を加えた和音となります。※**addと数字のみの差:** 単に「C9」とするとC7+9になりますが、「Cadd9」は7thを含まないためB<sup>b</sup>を含みません。このように“add”ありとなしでは含まれる構成音が異なるので、意図に応じて使い分けます。特に9/11/13という数字**単独**で書くと自動的に7th含みになりますので、「add」を入れることで「7thなしでその度数だけ追加」を明示できます (音楽理論上はadd2≡add9、add4≡add11ですが、music21では両方サポートしています)。
  - **sus 系:** **sus**は第三音 (3度) を長2度や完全4度に置き換えて\*\*懸垂 (サスペンション)\*\*することを表す記号です。music21ではsus4 (4度に置換) とsus2 (2度に置換) の両方を認識します。例えばCsus4はC-F-G (C和音で3度の



EをFに置換)、Csus2はC-D-Gとなります。7thと組み合わせることもでき、G7sus4はG7の和音から3度を抜き4度にしたもの(G-C-D-F)になります。**注意:** sus単独(数字なし)も入力上許容され、music21ではデフォルトで「sus4」と解釈されます(例: CsusはCsus4と同義)。

- **テンションの変化記号:** コード内の特定の度数を半音上げ下げする場合は、その度数に\*\*b(b)や#(#)を付けて表記します。典型的には**5th, 9th, 11th, 13th**に対して用い、「b5」「#5」「b9」「#9」「#11」「b13」等と記します。**music21**はこれら altered tension\*\*も解釈できます。例えばC7b5はC7の5thをフラットさせた和音(C-E-G<sup>b</sup>-B<sup>b</sup>)、C7#5はオーギュメントの別表記(C7+5)、C7b9は9thを半音下げた和音(C-E-G-B<sup>b</sup>-D<sup>b</sup>)です。複数のalterationも併記可能で、C7#5#9(5thと9thを両方変化)などと続けて書けます。major7系にもテンション変化は付けられ、Cmaj7#11のように書けば長7の和音にシャープ11thを含むコードになります。
- **省略(omit):** あるコード構成音を意図的に抜く(omitする)場合は、「omit度数」と表記します。**music21**ではomit3(3度抜き)やomit5(5度抜き)などを認識します。例えばC7omit3はC7から3度(E)を省いた和音(C-G-B<sup>b</sup>)で、結果的にC7susのような響きになります。省略指定はテンション追加と併用も可能ですが、コードの種類自体が曖昧になる場合(例: 3度を抜いたメジャー/マイナー不明の状態)は注意が必要です。
- **スラッシュ表記(オンコード/Bass指定):** コードシンボルの後ろに「/」を付けて音名を追記すると、その音を\*\*ベース音(最低音)\*\*として指定できます。いわゆるオンコード(分数コード)表記です。
  - スラッシュの前が和音本体(rootとコード種別など)、後がベース音を表します。ベース音もルートと同様、英字+必要なら#またはbで指定します。例: C/EはCコード(Cメジャー)の第3音Eをベースにした1転回形、D/F#はDコードのF#ベース(第3音ベース)です。ベース音がコード構成音でなくとも指定可能で、例えばF/GはFコード上にG音をベースに置いたコードを意味します(しばしばG11sus4とも解釈されるポリコード的な響き)。**music21**でもF/Gのような表記を受け付け、ルートFの和音にベースG音を含む構成で処理します(指定したベース音は和音の音リストにも加わります)。
  - マイナーやテンションを含むコードでもスラッシュ指定できます。例えばAm/C(Aマイナーの第3音Cをベース)、G7/B(G7の第3音Bをベース)、Ebmaj7/G(E<sup>b</sup>Maj7の第3音Gをベース)など。複雑な例ではB<sup>b</sup>maj7#11/D(B<sup>b</sup>Maj7(#11)の和音をD音ベースに)やCm7b5/E<sup>b</sup>(CハーフディミニッシュをE<sup>b</sup>ベースに)といった指定も可能です。
  - **表記上の注意:** スラッシュの前後にはスペースを入れず、ベース音は必ず「/」の後に書きます。また、「オンコード」はスラッシュ1回までです(C/G/Eのように二重のスラッシュは使えません)。複雑な分数和音は別途和音を分けて記述するか、必要に応じて**music21**で新たなコード種別を定義する方法もありますが(後述)、基本的には単一のスラッシュ指定に留めます。

以上が**music21**のコードシンボル文字列の基本構造です。**music21**はこの形式に則った文字列を解析し、**対応するピッチ集合**を持つChordSymbolオブジェクトを生成します。公式ドキュメントによれば、**music21**はMusicXMLのコード表記を基に多様な略記法をサポートしており、リードシート上で見かける様々なコード記号をかなり柔軟に解釈できるよう設計されています。

## 2. 最新版 music21 における対応コードと非対応コード

対応しているコード表記: music21 最新版 (2025 年現在) では、一般的なコード名称・構成はほぼ網羅的にサポートされています。以下は music21 の辞書に登録されている主なコード種別 (Chord Types) とその略号の例です:

- **メジャー系:** Major triad (長三和音) は略号なしでデフォルト認識 (例: C)。Major 7th は maj7 (または M7)、Major 9th/11th/13th は Maj9/Maj11/Maj13 など (「Maj」は大文字 M で表記することも多い)。
- **マイナー系:** Minor triad (短三和音) は m または min (例: Cm) で、Minor 7th は m7、Minor 9th/11th/13th は m9/m11/m13 です。
- **ドミナント系:** 7th (属七) は 7 (例: G7) で、9th/11th/13th は 9/11/13 と記します。特に断りがない限り、数字のみの 9,11,13 はドミナント (短7度) を含むコードと解釈されます。
- **拡張コード:** Major 系や Minor 系でも 7th を含む場合は上記のとおり maj7/m7 となり、それに 9th,11th,13th を加える場合は maj9/m11 等と表せます。minor-major (mM7 など) も対応。
- **オーギュメント:** Augmented triad (増三和音) は aug または +。Augmented 7th (増5度を含むドミナント7th) は例えば 7+ と略されます (この 7+ は「7#5」に相当)。Major7 に増5度を加えた和音は augMaj7 に相当し、music21 では +M7 のような形で出力される場合があります (例: E<sup>b</sup>augMaj7 が E-+M7 と表記されている)。入力時は C<sup>aug</sup>7 よりも C7#5 または C+7 といった明示的な表記の方が確実です。
- **ディミニッシュ:** Diminished triad (減三和音) は dim。Diminished 7th (減七の和音) は dim7 で表せます。
- **ハーフディミニッシュ:** 半減七の和音は上述の通り m7b5 で指定します。
- **サスティンド:** Suspended chords は sus4 または sus2。特に 7th 付きの場合は 7sus4 等、または分数表記で代用できます (例: G7sus4 = F/G)。music21 は sus の認識にも対応しています。
- **6 の和音:** 6 (長6度和音), m6 (短6度和音) をサポート。
- **パワーコード:** 5 度のみの和音は power と表記します (例: C<sup>power</sup>)。リードシートでよく使われる「C5」という表記は、music21 では出力時に C<sup>power</sup> とされます (入力でも C5 より C<sup>power</sup> を用いた方が確実です)。
- **特殊和音:** Neapolitan (ナポリの和音) や増六の和音 (イタリア・フランス・ドイツ和音) など分析用のコードも定義されています。例えば CN6 は C のナポリ和音、C<sup>It</sup>+6/C<sup>Fr</sup>+6/C<sup>Gr</sup>+6 がそれぞれイタリア・フランス・ドイツの増六和音を表す略号です。これらはクラシック分析用ですが、music21 の harmony モジュールで扱えます。

**非対応または注意が必要なコード表記:** ご提供のコードリストに怪しい表記がないか確認したところ、基本的には標準的な記法であれば music21 で解釈可能です。ただし、以下のような非標準・非対応の可能性のあるもの、または注意点がいくつか挙げられます:

- **「alt」表記:** ジャズでしばしば使われる「C7alt」(5度と9度のオルタードテンションを包括する記号) は、music21 の既定辞書にはありません。C7#5#9 など具体的な表記で代用してください。
- **Δ(三角記号):** 「Maj」をΔで表す俗称 (例: CΔ7) は認識されません。必ず maj や M7 と記してください。同様に半減七を示す ∅ 記号も使わず m7b5 で表現します。

- **add2/add4 vs sus:** add2/4 は認識されますが、音楽理論上は sus2 や add9 で代用できる場合があります。混乱を避けるため、慣習的に **9th を加える場合は add9** (または 9th 和音)、**2nd を加える場合も add9** を使うのが一般的です (もっとも music21 上は add2 でも通ります)。add4 も同様に、可能なら sus4 や add11 で表す方が誤解が少ないでしょう。
- **単独の数字 2 や 4:** C2 や C4 のようにただ数字を付けただけの表記は曖昧で、music21 には基本的に定義がありません。C2 と書いても music21 は「Chord Symbol Cannot Be Identified (認識不能)」となる可能性があります (C2 は理論的には Cadd9 または Csus2 を曖昧に示す記号ですが、正式ではありません)。必ず add や sus を明記してください。
- **マイナーのハイフン記法:** C-7 のような表記は、前述の通り文脈によって解釈が揺れる可能性があります。music21 では D-7 を与えると Db7 と解釈されるケースも報告されています (一方でジャズ的に D-7 = Dm7 と解釈可能) ため、安全策としてマイナーは **m** か **min** で表記することを推奨します。特にルート音にフラットが付く場合 (例: Dbm7) は Dbm7 や D- m7 のようにスペースを入れないと誤解釈の元です。結論: ハイフン (-) は極力フラット専用と考え、マイナーは m で記述するのが確実です。
- **カッコ付き表記:** 一部の出版譜ではテンションやオルタードを括弧で囲って書くことがあります (例: C7(b9) や C(ma7) 等)。music21 のパーサーは基本的に括弧なしの文字列を前提としているため、括弧は入れずに直接 C7b9 や Cmaj7 と書いてください。最近のバージョンでは括弧を含む RomanNumeral からの変換が改善されたとの情報もありますが、ChordSymbol 入力としては不要です。
- **no chord (休符) 記号:** 「N.C.」(no chord の略) や休符を示すようなコードシンボルは music21 には特に用意されていません。和音がない箇所を扱う場合は、ChordSymbol をそのタイミングに配置しないか、harmony.NoChord() クラスを検討します (ただし NoChord は出力時に明示的に "N.C." と表示されるわけではありません)。ユーザー提供リストに「N.C.」等が含まれている場合、music21 上は認識されないので注意してください。

以上を踏まえると、ユーザー提供のコードリスト中、一般的なコード (メジャー、マイナー、7th, 9th, sus, add, m7b5 など) はすべて music21 で対応しています。表記ゆれや特殊記号さえ避けていれば**非対応のコードは基本的に無い**と言えます。仮に「alt」「Δ」など前述の非対応例が含まれていればそれは修正すべきですし、今挙げた注意点に沿ってリストを見直せば誤記は見当たらないでしょう。

### 3. テンション記号やオンコード表記の書式・制約

上記 1,2 で触れた内容と重複する部分もありますが、特にテンション系記号と\*\*オンコード (スラッシュ表記)\*\*について整理します。

- **テンション記号の書式:** 基本ルールとして、コードシンボル中の数字 (2,4,6,7,9,11,13 など) はそのコードに含まれる拡張音・付加音を表します。注意すべきポイントは次の通りです:
  - 7を含む/含まない: **add** と付けるか否かで 7th の有無が変わります。例として add9 と 9 の違いは先述の通りですが、もう一度まとめると「9」と書けば 7 度あり、「add9」と書けば 7 度なしです。music21 もこの解釈に従います。したがって、**7th を入れたくないのに単に "9" と書かないことが肝心**です (その場合は必ず「add9」)。逆に 7th 入りで 9th を加えたい場合は「9」あるいは「7add9」ではなく\*\*「9」単独で十分\*\*です。

- 13の扱い: 「13」と書いた場合、理論上は7,9,11,13を含む和音ですが、実務上は11thを省略することもあります。music21では内部で「13thコード」を定義しており、基本的には11thも含めた形で処理します(不完全な構成でも自動判別します)。もし長13thコードで11thを#指定したい場合はmaj13#11のように併記できます。短13th(m13)も同様です。いずれにせよ、13と書いたら原則7thは入ると心得ます。7th無しで6のニュアンスを出したい場合は\*\*「6」や「add6」\*\*を使います(C6とCadd6は同義)。
- susとテンション: sus系とテンションを併用する場合の書式にも注意します。基本的には好きな順序で併記して構いません。例えば「C7に4度掛留し9thを加える」コードは、C7sus4add9と書いても良いし、一般にはC9sus4とも書かれます。この辺りmusic21はかなり柔軟で、C9sus4という入力もC7sus4add9と同等に解釈されます(内部的にはsus4が3度を置換する操作として扱われます)。実際、music21はRomanNumeral解析時に和声的掛留を特別扱いする機能もあり( $V_4^5$ を11thではなくsusとして扱う等)、ChordSymbolでもsusを適切に判断します。従って**susと他テンションの順序は厳密に規定されていませんが**、一般的可読性を考え7sus4add9のように書く方が無難でしょう。いずれにせよ「sus」は必ずsus4かsus2と明記し、単にsusだけにしない方が確実です。
- 複数テンションの順序: 複数のalteredテンション(変化音)やaddテンションを併記する場合、music21はその文字列全体を解析しますが、**書く順序に明確な制約はありません**。例えばBmaj7#11add9はBmaj7add9#11と書いても同じ結果になるはずです。一般的にはコード種別の後、数が小さいものから順に書くことが多いです(7th→9th→11th→13thの順、変化記号はその直前に付ける)。music21も人間同様に左から順に解釈していくため、極端に乱れた順序でなければ大丈夫ですが、**慣習的な順序で記述することを推奨**します。その方が誤解を招かず、将来的なメンテナンスや他ソフトへの移行時も安全です。
- **オンコード(スラッシュ)記法の書式:** こちらも前述していますが重要ポイントを再掲します:
  - スラッシュはルート側とベース側を区切るためだけに使います。ルートと音部分にスラッシュは含められません(例えば「A/B」を「A/B/C」のように二重にはできない)。
  - ベース音は必ず単音で、コードではありません。従って「G/C」「D7/G」のように書いた場合、CやGは単なる最低音であって、別の和音ではありません(ポリコードを示す表記とは解釈しない点に注意)。music21ではスラッシュ後の音をChordSymbolのbass属性として扱い、実際にChordSymbolが持つ構成音に加えます。ベース音も認識できない場合はエラーになりますので、必ずA~Gの範囲で指定してください。
  - **エイリアス:** スラッシュ表記は単純明快ですが、裏を返せば適切なコードネームで言い換え可能なケースも多いです。例えばF/Gは「G9sus4」コードとしても表現できます。music21での機能的な差はありませんが、もし自動解析結果が意図と違う表示になる場合は(例えば本当はF/Gと表示したいのにG9sus4と解析される等)、「コードシンボルの表記を固定する」設定も検討してください。ChordSymbolオブジェクトのfigureプロパティを自分で書き換えるか、無理に解析させず文字列そのものを使う方法もあります。ただ通常は入力した文字列どおりのコード名が保存されますので、意図した表記で入力することが大切です。

以上、テンション記号やオンコードの具体的な書式上の約束・制約を整理しました。要点をまとめると、\*\*「addの有無で意味が変わる」「susは数値付きで」「複数指定は順序に気をつけ読みやすく」「スラッシュは1回までで単音指定」\*\*といった点に留意すればよいでしょう。

#### 4. ユーザー提供コードリストの誤記チェック

ご提供いただいたコード一覧(例に挙がっている「Cadd9」「Bmaj7#11add9」「Gm7b5」など)について、上記のルールと照らし合わせて誤記・非標準な記法がないかを確認します。

- **Cadd9**: 「Cメジャーに9thを加えた和音」であり、表記・意味ともに正しいです。music21では問題なく**C-E-G-D**の和音として解釈されます。類似のCadd2も認識はされますが、前述のとおり通常はadd9で表すのが一般的です。
- **Bmaj7#11add9**: Bメジャー7thコードにシャープ11thと9thを加えた高度なコードですが、記法上まったく問題ありません。music21はまずBmaj7(長7度和音)を認識し、#11修飾とadd9を順次適用して**B-D#-F#-A#(基のBmaj7) に対しE#(#11)とC#(9th)を加えた構成**として処理します。したがって正しくBMaj9(#11)の響きが得られます。表記の順序も妥当です(maj7の後に#11, add9)。仮にBmaj7add9#11と入れ替えて書いても結果は同じですが、一般的にテンションは数値の昇順で書く方が読みやすいです。
- **Gm7b5**: Gの半減七(マイナー7<sup>b</sup>5)を示す標準的な記号です。music21では内部で**half-diminished seventh**として認識され、構成音G-B<sup>b</sup>-D<sup>b</sup>-Fを持つChordSymbolが生成されます。全く問題ありません(Gø7という記号は使えませんので、Gm7b5でOKです)。
- **(その他リスト中の一般的なコード)**: おそらくリストには上記の他にも様々なコードが含まれているでしょう。代表的なものについて言及します:
  - メジャー/マイナーのトライアド(例: D, Emなど) – **OK**。大文字のルート+mでマイナー、それ以外はメジャーになります。
  - 各種7thコード(例: A7, Dm7, FMaj7など) – **OK**。前述通り適切な略号で書かれていれば問題ありません。
  - 拡張コード(例: E9, A13, Bm11など) – **OK**。7thを含む和音として認識されます。メジャー系はMajを付け忘れないよう注意(F9はF7(9)でありFmaj9は長7の和音です)。
  - sus系(例: Asus4, D7sus4, Gsus2など) – **OK**。susの使い方に問題がなければ認識されます。Dsusと書けばDsus4になります。sus4とsus2を間違えないようにしましょう。
  - add系(例: Aadd9, Dm(add9), Gadd11など) – **OK**。addの後の数字が2桁でもしっかり読めます(11, 13も可)。括弧は不要なのでDmadd9で十分です。
  - オルタードテンション(例: C7#5, F7b9, Emaj7#5など) – **OK**。指定漏れがなければ期待通り解釈されます。それぞれC7(#5), F7(<sup>b</sup>9), E+Maj7の意味です。
  - スラッシュコード(例: C/E, Dm/F, B<sup>b</sup>maj7/Dなど) – **OK**。ベース音が正しく表記されていれば対応します。例えばB<sup>b</sup>maj7/Dは表記上はB-maj7/Dと同等で、B<sup>b</sup>のメジャー7をD音ベースで処理します。
- **誤記の可能性のあるもの**: リスト中にもし以下のようなものがあれば修正が必要で

す：

- 「H」や「Si」など英米以外の音名記法（独仏式）－ **修正**：A～Gの英名に統一する（B/H問題：Bはシ、Hもシですがmusic21はBを採用）。
- テンションの表記ミス－例：C7(9)と書くべきところをC(9)だけにしていたり、C7#のように数字抜けしていたりすると**認識不能**です。必ず「何を#するのか」まで書きます。
- ハイフン/シャープの半角全角違い－C#m7の#を全角で書いている等は認識されません。必ず半角の#やbを使います。コピー&ペーストで紛れ込んだUnicodeのb#にも注意してください。
- 大文字小文字の混乱－例えばcMaj7のようにルートを小文字にすると**認識されない可能性**があります（音名は大文字が基本）。コード種別は大小どちらでも認識されるケースがありますが、公式ドキュメントでは小文字主体の例が多いです。  
統一して小文字（maj, min, aug など）で書く方が良いでしょう。

以上より、ご提供リストのコードは概ね標準的であり、大半がmusic21でそのまま利用可能と思われます。誤記がないか**チェックすべきポイント**は次のセクションのチェックリストにもまとめますのでご参照ください。

## 5. コードシンボル運用上の注意点（表記ゆれ・エスケープ・大文字小文字など）

最後に、music21でコードシンボルを正しく扱うための注意事項を整理します。

- **表記ゆれへの対処**：音楽のコード表記には複数の言い方が存在しますが、**一つのプロジェクト内では統一**することを強く推奨します。music21はharmony.CHORD\_TYPES辞書に多数の同義略語を持っていますが、例えば「ある場所ではmin、別ではm」といった混在は可読性を下げます。music21では後から略号を変更する機能（changeAbbreviationFor）もありますが、基本的には**最初から統一表記**を心掛けてください。特に**minor**はmにするかminにするか、**major7**はmaj7かM7か、**半減七**はm7b5固定か、など決めておくとい良いでしょう。
- **エスケープと特殊文字**：コードシンボル文字列自体は特別なエスケープ処理を要しませんが、**環境によっては注意**が必要です。例として、Music21をPythonで使う際は文字列リテラルとして渡すので問題ありません。しかし、YAMLやINIなど設定ファイルにコードを直書きする場合、「#」や「/」がコメント開始やディレクトリ区切りと解釈される恐れがあります。実際、YAMLでは「F#7」をクォートせず書くと「F」が値、「7」はコメント扱いとなりエラーになります。**対策**：設定ファイル上ではコードシンボルを**必ずクォートで囲む**か、エスケープ（例えば「F#7」のようにバックスラッシュで#をエスケープ）してください。
- **大文字・小文字**：ルート音（A～G）は大文字が基本です。一方、コード種別やテンション部分は大文字小文字を厳密には区別しませんが、統一した方が無難です。例えば「Maj」「maj」「MAJ」はいずれも通る可能性がありますが、混在させる理由もないでしょう。一般的にコード種別略号は**小文字**で書くことが多いです（m, aug, dim, sus など）。MusicXMLなど他形式へエクスポートする際もその方が整合性が取れます。従って、特別な理由がなければCmaj7やGdim7のように**必要部分以外は小文字**で記述してください。
- **スペースや区切り**：コードシンボル中に**不要な空白は入れない**ようにします。例えば「C maj7」や「Ab 7」などとスペースを入れると認識されません

(ChordSymbol('C maj7') はエラーになります)。常にルートから続けて書きま  
す。また、テンション間をカンマやスペースで区切る表記 (例: C7 (9,11)) も使えま  
せん。一続きの文字列として書いてください。

- **出力時の表記:** music21でChordSymbolを扱いMusicXMLへ書き出すと、内部辞書に基づいた標準形で出力されます。これは若干書式が変わる場合があります (例: Cmaj7と入れてもMusicXMLではと出力される等)。ただし、基本的に**入力した表記は保存**されます。解析させて自動でコードシンボル生成した場合 (例えばchordify後にharmony.chordSymbolFigureFromChordを使った場合) は、music21既定の略号が使われます。もし出力表記をカスタムしたい場合、前述の略号変更関数 (changeAbbreviationFor) でデフォルト略号を変えることもできます。例えば「major」をmajに変更すれば以降ChordSymbol('C')の出力がCmajになります。こうした調整も可能ですが、まずは**既定の略号に従う**ことをお勧めします。

以上のポイントを踏まえれば、music21でコードシンボルを扱う際のトラブルをほぼ防げます。それでは最後に、これまでの内容を簡潔にまとめた**チェックリスト**を提示します。

### music21コード記述チェックリスト

1. **ルート音と事故記号:** コードのルートは**英大文字 A-G**で記述したか? シャープ「#」やフラット「b/-」は正しく付けたか (全角記号やH音名など使っていないか)。
2. **コード種別 (品質) の略号:** メジャー/マイナー/その他のコード種別を **music21の認識する略号**で書いたか? (Maj/maj、m/min、dim、aug/+、sus4/sus2、m7b5など)。表記をプロジェクト内で統一しているか。
3. **拡張番号:** 7th, 9th, 11th, 13thなど**適切な数字を付けたか**? また、その数字が意味する構成を理解しているか (単に「9」と書いて7thを含めて良いか、7th無しならadd9にすべき)。Major系はmajの有無に注意 (maj9と9は別物)。6の和音は6を使用。
4. **テンション追加 (add):** add○の用法は正しいか? 7th無しコードにテンションを足す場合は**必ずadd**付きで記述したか (add2/add9, add4/add11, add6など)。また、理論上存在しない「C2」「C4」のような略記を使っていないか。
5. **オルタード記号:** ♭5, #5, ♭9, #9, #11, ♭13等の**変化記号は数字の直前に付けて正しく**書いたか (「C7# 5」のようにスペースを入れたり「C7 5#」のように順序を間違えたりしていないか)。複数ある場合も読みやすい順序で並べたか (#5♭9は通常♭9, #5の順など)。「alt」のような包括表記は使わず明示したか。
6. **サスペンション:** sus4やsus2の指定は適切か? 3度を含まないことに留意したか (susコードに誤ってMaj7など付けていないか)。「sus」単独より**sus2/sus4を明記**したか。7thやaddテンションと併用時も表記ミスがないか (例: きちんと「7sus4add9」と書いているか)。
7. **オンコード表記:** スラッシュは**1回だけ**使用し、後ろには単一のベース音を指定したか。ベース音にもシャープやフラットを正しく付与したか (例: /E♭ は /E- または /Eb)。不要な空白を入れていないか (「C / E」ではなく C/E)。ポリコードを無理に「/」で表そうとしていないか (二重のスラッシュは禁止)。
8. **特殊コード:** 「N.C.」や特殊記号 (Δ, Ø, alt等) を使っていないか。必要ならば対応する既定表記 (dim, m7b5等) に置き換えたか。どうしても独自コードを使う場合は、music21のaddNewChordSymbol() 機能で追加定義する方法も検討 (ただし基本は既存の範囲で記述)。
9. **フォーマット/文字:** コードシンボル文字列中に**余分なスペースや改行がないか**再確認。全角英数字や特殊なUnicode記号を混ぜていないか。特に「#」「b」「/」は半角で

入力したか。

10. **一貫性と可読性:** コードの綴りは楽曲全体で統一されているか (表記ゆれチェック)。人間が読んでも理解できる妥当な順序か (慣習から大きく逸れていないか)。例えば EbmM7 のような複雑コードも正しく綴れているか (Eb マイナーに Maj7)。必要ならコメント等で和音の構成を補足できるよう準備。
11. **エスケープ (必要なら):** コードシンボルをプログラム以外のテキストに書く際、そのフォーマットのルール上問題がないか確認。特に **YAML/シェル** では # や / を適切にエスケープまたはクォートしたか。

以上のチェック項目を順番に確認すれば、music21 でコードシンボルが正しく扱える状態になります。実際に music21 の ChordSymbol クラスに文字列を渡してエラーが出ないかテストし、想定通りの構成音になるかを確かめるのも有効です。

これらのガイドラインとチェックリストを活用し、music21 で安全かつ正確にコード記号を取り扱ってください。音楽理論的な意味とプログラム上の仕様を両方踏まえることで、コード分析や自動伴奏生成などの作業をスムーズに進められるでしょう。