

音符表現について

MML ではテキストで、

< 音程 >< 音長 >

を音符の一単位として記述します。これを MMLトラックに列挙することで、さまざまな音楽表現を行います。

音程について

音階名と MML での音程指定名の対応は、以下の通りです。

ド	レ	ミ	ファ	ソ	ラ	シ	休符		
c	d	e	f	g	a	b	r	+	-

[例]

cdefgab

;「ド・レ・ミ・ファ・ソ・ラ・シ」を演奏します。

音程指定名の後に「+」をつけるとシャープ、「-」をつけるとフラットとなります。

[例]

c c+ d e- e

;「ド・ド ・レ・ミ ・ミ」を演奏します。

なお、「ダブルシャープ」「ダブルフラット」は記述できません。「c++」などと書くと「c+」(ド) と「+」(音長加算) に分けて解釈されます(「18 c++」なら「c+8+8」と等価となり、4 分音符のドになります)。ご注意ください。

調号コマンドを使用すると、シャープやフラットを省略できます。

関連

音長について , 調号コマンドについて

音長について

音長は、音楽的音長、もしくは絶対音長(カウント数)で表記します。必ず 10 進数で指定してください。

パラメータ	解説
1 ~ 192	音楽的音長
#<num>	絶対音長 (1 ~ 9999)
.	付点 (複付点も可)

^	音長加算（タイ）
+	〃
-	音長減算
\	〃
0	<u>Tone Doubler 用音程指定</u>

音楽的音長による指定

全音符を「1」、4分音符を「4」、8分音符を「8」...と表現する方法で、n分音符のnの値で指定します。1～192を指定可能です。音長の直後に「.」(ピリオド)をつけると付点音符として扱います。複数の付点も記述可能です。付点は解釈可能な範囲の間、機能します。

[例] c8 ; 8分音符
 [例] c8. ; 付点8分音符(8分音符 + 16分音符相当)
 [例] c4... ; 4分音符 + 8分音符 + 16分音符 + 32分音符相当

なお、「0」は特殊な扱いとなっており、Tone Doubler 用音程指定となります。

絶対音長（カウント数）による指定

数値の頭に「#」を置くと、ドライバの分解能単位での指定になります。全音符の分解能が192カウントの場合、4分音符は「#48」、8分音符は「#24」となります。付点は使用できません。指定可能範囲は1～9999です。これ以上の音長を表現する場合は、タイやレガートを使用してください。

[例] c#192 ; 全音符(分解能が192カウントの場合)
 [例] c#12 ; 16分音符(分解能が192カウントの場合 (192 ÷ 16=12))
 [例] c#12. ; エラー(付点は使えない)

ドライバ分解能の初期値は、4分音符 = 48 カウント(全音符 = 192 カウント)ですが、#COUNT ヘッダで変更可能です。

■ 音長の省略

音長は省略可能です。省略時の音長は 1 コマンドで設定してください。

■ 音長の加減算

「^」もしくは「+」で音長の加算（タイ）、「\」もしくは「-」で音長の減算が可能です。音楽的音長、絶対音長、音長省略を組み合わせ、柔軟な音長表現ができます。

音楽的音長の演算は他の数値とは異なり、楽典的な加減算となります。たとえば、「4+8」は12ではなく付点4分音符です。

なお、音程指定名は最初のひとつだけで構いません。たとえば4分音符と8分音符のドをタイで

繋げる場合は「c4^c8」ではなく「c4^8」となります。

```
[ 例 ] c16.^4      ; 付点 16 分音符 + 4 分音符相当
[ 例 ] c2^4^8      ; 複付点 2 分音符 (2 分音符 + 4 分音符 + 8 分音符) 相当
[ 例 ] l8 c4^^      ; 2 分音符 (4 分音符 + 8 分音符 + 8 分音符) 相当
[ 例 ] c2-16-4      ; 付点 8 分音符 (2 分音符 - 16 分音符 - 4 分音符) 相当
[ 例 ] c2-8^4.      ; 付点 2 分音符 (2 分音符 - 8 分音符 + 付点 4 分音符) 相当
[ 例 ] c2-#3        ; 2 分音符から 3 カウントぶん引いた音長
[ 例 ] c#12-32      ; 12 カウントから 32 分音符ぶん引いた音長
[ 例 ] l4 f#16       ; 付点 8 分音符 (4 分音符 - 16 分音符) 相当
[ 例 ] l4 c++2       ; 付点 2 分音符 (ド の 4 分音符 + 2 分音符) 相当
[ 例 ] l2 e--8       ; 付点 4 分音符 (ミ の 2 分音符 - 8 分音符) 相当
```

■ 音長表記の注意点

- ・#COUNT ヘッダで分解能を変更すると絶対音長が変化します。音楽的音長もカウント数の小数点以下が切り捨てられるため、演奏がずれる要因となります。ご注意ください。
- ・1 バイトのデータが持てる最大音長は 254 カウントです。音長加算の結果 254 を超えた場合、コンパイル時に 255+x1+x2... と 2 バイト以上に音長を畳み込んでデータを作成します (255 カウントなら 255+0、600 カウントなら 255+255+90、1000 カウントなら 255+255+255+235)。その際、q コマンドは最後の 1 バイトぶんの音長に対して機能するので、意図しない音の切れ方をすることがあります。この場合はスラーもしくはレガートをを用いて、明示的に音長を指定してください。

関連

#COUNT ヘッダ, q コマンド

スラーとレガートについて

書式	解説
&	スラー
(	レガートオン (スラー開始)
)	レガートオフ (スラー終了)

「&」はスラーです。音符コマンドの直後に付けたときのみ有効となりますので、ご注意ください。音符コマンド以外に付けたときは、その直前の音符コマンドでの指定が持続されます。

```
[ 例 ] c&v12&d      ; c&v12d 相当 (c のスラーが持続し、c と d が繋がる)
[ 例 ] cv12&d       ; cv12d 相当 (c にスラーがないので、c と d は繋がらない)
```

「(」と「)」はレガートのオン / オフです。複数の音符をスラーで繋げるときは、レガートを使うと柔軟に表現できます。

```
[ 例 ] (c4d16e8)f2 ; c4&d16&e8&f2 相当
[ 例 ] c4(d16e8)f2 ; c4d16&e8&f2 相当
[ 例 ] c4(d16)e8f2 ; c4d16&e8f2 相当
```

なお、同じ音程の音符を繋げる場合は「^」のタイを推奨します。