

視認性向上のための書式

書式	--
{ ~ }	<u>改行可能エリア</u>
:	<u>区切り記号</u>
;	<u>一行コメント</u> （改行するまで有効）
/* ~ */	<u>コメント範囲指定</u> （改行しても有効）

改行可能エリア

MMLトラックの定義、ドライバマクロの定義、コンパイラマクロの定義を行う際、「{ ~ }」（波カッコ）で括られた範囲は、改行しても一つの処理単位とみなします。

次の三つの MML は全て同じ演奏を指示しています。

[例 1]

```
ABC v13 o4 q6 l8 ccddeeff erdrc4.r
```

[例 2]

```
ABC { v13 o4 q6 l8
      ccddeeff
      erdrc4.r
}
```

[例 3]

```
$macro1 {
      v13 o4 q6 l8
      ccddeeff erdrc4.r
}
ABC $macro1
```

■ 記法の問題

現在は「}」の後、改行せずに MML を記述しても有効となってしまいますので、下記の例はすべて同等の演奏結果となります。下記の例 3 や例 4 のような記述は推奨しません。現バージョンではコンパイルできますが、今後、エラーとなる恐れがあります。

[例 1] 正しい記法

```
A {
  cdef
  gab>c
}
```

[例 2] 正しい記法

```
A {cdef
  gab>c}
```

[例 3] 誤った記法

```
A {cdef
  }gab>c
```

[例 4] 誤った記法

```
A {cdef}gab>c
```

区切り記号

「:」(コロン) の内部処理はタブや半角空白と同等で、それ自体に意味を持ちません。ドライバマクロやコンパイラマクロの終了、及び PSG 音色定義などの区切り文字として使用できます。

タブや半角空白も区切り文字として使用できますが、視認性を上げたい場合は「:」(コロン) を使用してください。

[例]

```
!KICK @0 l#1 (o4d<gd<gd<gd<g)e#16 ; #24
$Snare @1 o3
$Tom @2 o6
A [16 !KICK r8 $Snare:b8 !KICK !KICK r4 $Tom:b-8]
```

; キックをドライバマクロで、スネア / タムをコンパイラマクロで演奏。

; 「\$Snareb8」とは書けません(マクロ名と認識してしまうため)。
; 「\$Snare b8」と書いても構いませんが、
; 以下のように、発音単位が把握しにくくなります。

```
A [16 !KICK r8 $Snare b8 !KICK !KICK r4 $Tom b-8]
```

一行コメント

「;」(セミコロン) 以降の行端までを無視し、コメントとして扱います。

[例]

```
A l4 cgeg dgfg ; セミコロンより後ろはコメント
; これもコメント
```

コメント範囲指定

「/* ~ */」で括った範囲がすべてコメントとなり、コンパイル時に無視されます。複数行のコメントを入れるときや MML の編集集中に便利な機能です。

ネスト(入れ子)の回数に制限はありませんので、「/*a/*b/*c ~ c*/b*/a*/」のような記述もできます。この場合、入れ子の中がすべてコメントになります。

なお、「;」(セミコロン) との優先度は同等となっており、先に記述したほうが有効です。「;」によって「/*」をコメントアウトした際、「*/」だけが残っている状態になるとエラーが発生しますのでご注意ください。

また、「/*」だけを記述すると、その位置からテキスト終端までがすべてコメントになります。

[例]

```
A { o4 l4
  cdef edc.r8
/*
  efga gfe.r8
  crcr crcr
```

```

*/
l8ccddeeff erdrc4r4
}
; efga gfe.r8 と crcrcrcr がコメントとして無視され、
; cdef edc.r8 l8ccddeeff erdrc4r4 が演奏される

```

[例]

```

A { o4 l4
  cdef edc.r8
  efga /* gfe.r8
  crcr crcr */
  l8ccddeeff erdrc4r4
}
; この場合は gfer8. と crcr crcr がコメント扱いになり、
; cdef edc.r8 efga l8ccddeeff erdrc4r4 が演奏される

```

[例]

```

A { o4 l4
/*  cdef edc.r8
  efga /* gfe.r8 */
  crcr crcr
*/ l8ccddeeff erdrc4.r
}
; この場合は 2 段階のネストになるので、
; 最後の l8ccddeeff erdrc4.r のみ演奏される

```

[例]

```

;A /* cdef
A gab>c */
; 1行目の「;」によって「/*」がコメントアウトされているので、
; 「*/」だけが検出されてしまい、エラーが発生する

```