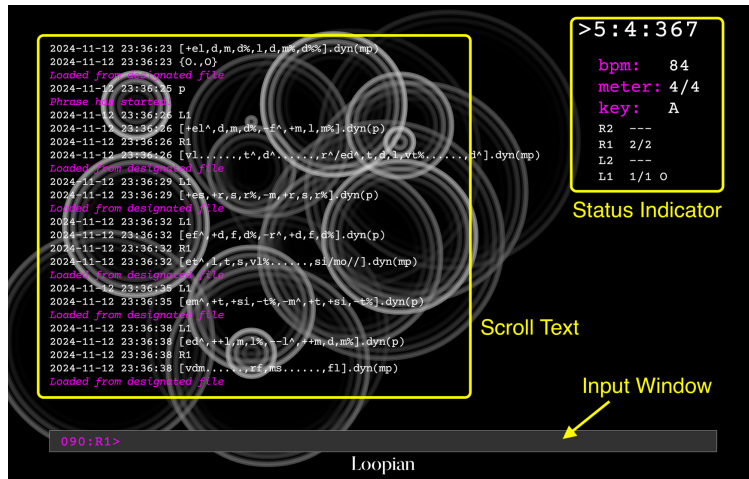


Loopian CheatSheet



1 起動と終了

- 起動: ターミナルにて、以下のように入力し return

```
./loopian
```

- 終了

```
>!quit  
>!q (省略形)
```

※Loopian の Window の閉じるボタンを押して終了した場合、ログファイルは記録されない

2 再生コントロール

- 再生開始

```
>play  
>p (省略形)
```

- 再生終了

```
>stop  
>fine
```

※ space key のあと return で、トグルで再生の開始、終了を繰り返す (表示はビリオド)

- 再生が途中で止まっている時、その小節の先頭から再生する

```
>resume
```

- 次の小節の先頭を再生した後、再生を止める

```
>fermata
```

- 今鳴っている音を消音する

```
>panic
```

3 入力Part切替

Input Windowのプロンプトに、コマンドの履歴番号と現在の入力パートが表示されている

```
001:R1>
```

- L1に切り替え

```
>L1  
>left1
```

同様に L2 に切り替えられる

- R1に切り替え

```
>R1  
>right1
```

同様に L2 に切り替えられる

4 Phrase入力

[] (角カッコ) 内に音符を入力する

記法

```
>[note] .fn1() .fn2(parameter)
```

note のところに、音符 (音の高さと長さの指定) が入る

fn() のように、Phraseに変化を与える関数を指定できる

fn(parameter) のように、関数にパラメータを追加できる

- パートの全Phrase削除

```
>[]
```

- スラッシュで小節区切り

```
>[.../.../...]
```

- Loopせず終了する

```
>[...//] (Phraseの最後に // をつける)
```

5 音程の指定方法

	ド	レ	ミ	ファ	ソ	ラ	シ
半音上	di	ri	(mi)	fi	si	li	(ti)
	d	r	m	f	s	l	t
半音下	(da)	ra	ma	(fa)	sa	la	ta

- 休符

```
>[x]
```

■ オクターブ指定

下に移動 >[d,s] (下のソが上より近い)

上に移動 >[d,++s]

二つ下のソに移動 >[d,-s]

二つ上のソに移動 >[d,++s]

■ example

同時発音 >[dms,rfil,msit] (三音同時に発音)

きらきら星のメロディ >[d,d,+s,s/l,l,s,x/f,f,m,m/r,r,d,x]

6 音価の指定方法

以下の音価を音程の前に記述する

	二分音符	四分音符	八分音符	十六分音符
	h	q	e	v
付点	h'	q'	e'	v'
三連符	3h	3q	3e	3v

基準音価とは、音価情報を書かないときに採用されている音価のこと

最後に指定された音価が基準音価になる。デフォルトは四分音符が基準音価となる。

(三連符を指定した場合、実際の長さは2/3となる)

「遠き山に日は落ちて」 >[hm,qs/h's/hm,qr/h'd/hr,qm/hs,qm/h'r]

Bach Invention No.1 >[ex,d,r,m,f,r,m,d/+qs,d,v,t,l,e,t,qd]

■ 最後にoを付けると小節の最後まで伸びる

小節の終わりまで伸ばす >[do/...]

次の小節も伸ばす >[do/o/]

■ ドットで基準音価の倍数だけ音を伸ばす

3倍伸ばす >[qd..]

■ アンダーバー＋音価でタイを表す

タイで伸ばす >[hd_e] (二分音符＋八分音符)

小節を跨ぐタイ >[hm,r/_e,...]

7 Composition入力

{ } (波カッコ) 内に和音などを入力する

記法 >{chord}

chord のところに、和音名が入る

■ パートの全Composition削除

>{ }

■ スラッシュで小節区切り

>{.../.../...}

■ Loopせず終了する

>{...//} (最後に // をつける)

8 和音の指定方法

和音は "root + type" という形で指定する

以下は root の指定方法である

root	ド	レ	ミ	ファ	ソ	ラ	シ
半音上	I#	II#	(III#)	IV#	V#	VI#	(VII#)
	I	II	III	IV	V	VI	VII
半音下	(Ib)	IIb	IIIb	(IVb)	Vb	VIb	VIIb

以下は type とその構成音の一覧である

type	1	2	3	4	5	6	7	8	9	10	11	12
_	●				●			●				
_m	●			●				●				
_7	●				●			●			●	
_m7	●			●				●			●	
_6	●				●			●		●		
_m6	●			●				●		●		
_M7,_maj7	●				●			●				●
_mM7	●			●				●				●
_add9	●		●		●			●				
_9	●		●		●			●			●	
_m9	●		●	●				●			●	
_M9,_maj9	●		●		●			●				●
_mM9	●		●	●				●				●
_+5,_aug	●				●				●			
_7+5,_aug7	●				●				●			●
_7-9	●	●			●			●			●	
_7+9	●			●	●			●			●	
_dim	●			●			●					
_dim7	●			●			●			●		

_													
_m7-5	●			●		●				●			
_sus4	●					●		●					
_7sus4	●					●		●				●	
_chr	●	●	●	●	●	●	●	●	●	●	●	●	●
_ion	●					●		●		●		●	
_dor	●		●	●		●		●		●		●	
_lyd	●		●		●		●	●		●		●	
_mix	●		●		●	●		●		●	●		
_aeo	●		●	●		●		●	●		●		

※ type の _ (アンダーバー) には root 指定が入る

- 和音なし、ダンパーペダル付き

```
>{0}
```

- 和音なし、ダンパーペダル無し

```
>{X}
```

- 長さの指定

1 小節単位

```
>{I/V}
```

一拍単位

```
>{I.,V./IV,III,II.}
```

(ビリオドで拍を伸ばす)

```
>{I.,,/V//}
```

(何も指定しないと和音なし)

- ビリオドで複数小節伸ばす

```
>{I/. /IV/.}
```

- example

循環和音

```
>{I/VIm/IV/V7}
```

Let It Be

```
>{I.,V./VIm.,IV./I.,V./IV,V,I.}
```

9 調、テンポ、拍子

- 調の指定

keyをFにする

```
>set.key(F)
```

keyをC3にする

```
>set.key(C3)
```

- テンポの指定

BPM (テンポ) を150にする

```
>set.bpm(150)
```

- 拍子の指定

拍子を 4/4 にする

```
>set.meter(4/4)
```

拍子を 6/8 にする

```
>set.meter(6/8)
```

- オクターブの指定

現在のパートの音程を 1 オクターブ上げる

```
>set.oct(+1)
```

現在のパートのオクターブを元に戻す

```
>set.oct(0)
```

10 Phraseの音符への指示

- ^で音量を上げる

```
>[d^,d^^,d^^^]
```

(^の数で音量が増える)

- %で音量を下げる

```
>[d%,d%%,d%%%]
```

(%の数で音量が増える)

- 'で音価を半分にする (スタカート)

```
>[qd']
```

```
>[<ed,r,m>']
```

(<>で複数音を指定)

- ~で音価を120%にする (レガート)

```
>[qd~]
```

```
>[<ed,r,m>~]
```

(<>で複数音を指定)

- 関数dyn() でPhrase全体の音量を指定する

```
>[d,r,m.].dyn(f)
```

(fはフォルテ)

```
>[d,r,m.].dyn(mp)
```

```
>[d,r,m.].dyn(pp)
```

- 関数dmp(off) で和音を指定していても、Phraseのダンパーペダルをオフにする

```
>[d,r,m.].dmp(off)
```

- 関数stacc() でPhrase全体をスタカート指定する

```
>[d,r,m.].stacc(80)
```

(音価を80%にする)

- 関数legato() でPhrase全体をレガート指定する

```
>[d,r,m.].legato(120)
```

(音価を120%にする)

11 動的パターン

Phrase の中で実際の音程を指定せず、和音やアルペジオを自動生成

Composition を使って和音を指定することを前提とした機能

Cluster (同時になる複数の音) は関数の形で記述 (引数でパラメータ指定)

```
>[C1s()]
```

```
>[C()]
```

(省略形)

```
>[hC(q,,),hC(e,,)]
```

 (音価指定)

- 第一引数は、和音連打の音価を表す（デフォルトは q）

```
>[Cls(e,,)]
```

音価は h, q, e, v を指定できる（二分音符、四分音符、八分音符、十六分音符）

- 第二引数は、同時に鳴る数を表す（範囲：2..5、デフォルトは4）

```
>[Cls(e,5,)]
```

- 第三引数は、最低音の音程の場所を表す（範囲：-6..7、デフォルトは0）

```
>[Cls(e,3,-2)]
```

Arpeggio（分散和音）も関数の形で記述

```
>[Arp()]
```

```
>[A()]
```

 (省略形)

```
>[hA(e,u,),hA(e,d,)]
```

 (音価指定)

- 第一引数は、分散した音の音価を表す（デフォルトは q）

```
>[Arp(e,,)]
```

音価は h, q, e, v を指定できる（二分音符、四分音符、八分音符、十六分音符）

- 第二引数は、同時に鳴る数を表す

```
>[Arp(e,u,)]
```

u:上昇 d:下降 ux:クロス上昇 dx:クロス下降（デフォルトはu）

- 第三引数は、最低音の音程の場所を表す（範囲：-6..7、デフォルトは0）

```
>[Arp(e,u,-2)]
```

filenameを省略すると、日時がファイル名になる

13 付録：和音進行の作り方

よく使う和音

よく使う	I	II ^m	III ^m	IV	V	VI ^m
少し複雑	IM7	II ^m 7	III ^m 7	IV ^M 7	V7	VI ^m 7
たまに使う	I7	II7	III7	IV7		VI7

和音進行の特徴

- ・ I で始まると長調っぽい開始、I で終わると長調っぽい終了
- ・ VI^m で始まると短調っぽい開始、VI^m で終わると短調っぽい終了
- ・ rootが3つ上の和音に行きたがる

I → IV

II → V

III → VI

V → I

VI → II

12 ファイル操作

- ・ アプリのあるフォルダに /log フォルダが自動的に作成される
- ・ !quit とコマンドで Loopian を終了した時、ログファイルが自動的にセーブされる
- ・ アプリのあるフォルダに /load フォルダを作成すると、Loopian のコマンドからファイルを読み出すことが可能になる

- /load 以下のサブフォルダ(sub_folder)がある場合、そのフォルダ名を指定

```
>set.path(sub_folder)
```

- !load でファイル(filename)をロードする

```
>!load,filename
```

 (拡張子は書かない)

```
>!l.filename
```

 (省略形)

```
>!l
```

 前回と同じならファイル名省略

- !save で特定行以降の入力をファイル(filename)をセーブする

```
>!save(20).filename
```