

# windowsにOvertoneをインストールする

2023/4/14

初版

## 概要

windowsにOvertoneをインストールする。

Overtoneは、SuperColliderへ接続するClojureプログラミング環境でClojureでSuperColliderの音源の音を鳴らすことができる。

## SuperColliderのインストール

まずは、音源であるSuperColliderを以下のURLからダウンロードしてインストールする。

[https://github.com/supercollider/supercollider/releases/download/Version-3.13.0/SuperCollider-3.13.0\\_Release-x64-VS-3188503.exe](https://github.com/supercollider/supercollider/releases/download/Version-3.13.0/SuperCollider-3.13.0_Release-x64-VS-3188503.exe)

起動は、メニューから以下を選択する：

SuperCollider-3.13.0

## OpenJDKのインストール

以下のurlからダウンロードしてインストールする：

<https://aka.ms/download-jdk/microsoft-jdk-17.0.6-windows-x64.msi>

## clojure/leiningenのインストール

以下を実行する：

```
scoop install leiningen
```

```
# バージョンの確認
```

```
lein --version
```

```
Leiningen 2.10.0 on Java 17.0.6 OpenJDK 64-Bit Server VM
```

```
clj
```

```
Clojure 1.11.1
```

```
user=>
```

```
# [Ctrl]+[C]で抜ける
```

# Overtoneの実行

#プロジェクトを作成する

```
lein new ot777
# ot777は任意
cd ot77
```

project.cljを編集して以下のようになる：

```
-----
(defproject ot777 "0.1.0-SNAPSHOT"
  :description "FIXME: write description"
  :url "http://example.com/FIXME"
  :license {:name "EPL-2.0 OR GPL-2.0-or-later WITH Classpath-exception-2.0"
            :url "https://www.eclipse.org/legal/epl-2.0/"}
  :dependencies [[org.clojure/clojure "1.11.1"] [overtone/overtone "0.10.6"]]
  :repl-options {:init-ns ot777.core})
-----
```

# 実行

```
lein repl
```

#出力

```
nREPL server started on port 49927 on host 127.0.0.1 - nrepl://127.0.0.1:49927
REPL-y 0.5.1, nREPL 1.0.0
Clojure 1.11.1
OpenJDK 64-Bit Server VM 17.0.6+10-LTS
  Docs: (doc function-name-here)
        (find-doc "part-of-name-here")
  Source: (source function-name-here)
  Javadoc: (javadoc java-object-or-class-here)
  Exit: Control+D or (exit) or (quit)
  Results: Stored in vars *1, *2, *3, an exception in *e
```

```
ot777.core=>
```

#以下のように入力する：

```
ot777.core=> (use 'overtone.core)
```

#出力

```
--> Loading Overtone...
```

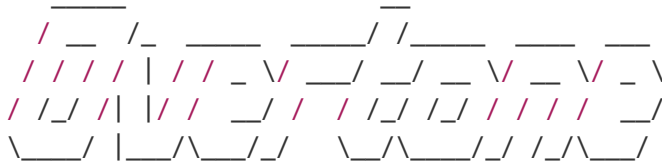
```
--> Please boot a server to start making noise:
```

- \* (boot-server) ; boot default server (honours config)
- \* (boot-internal-server) ; boot an internal server
- \* (boot-external-server) ; boot an external server
- \* (connect-external-server) ; connect to an existing external server

```
nil
```

#以下のように入力する：

```
ot777.core=> (boot-external-server)
--> Booting external SuperCollider server...
--> Connecting to external SuperCollider server: 127.0.0.1:30711
--> Connection established
```



Collaborative Programmable Music. v0.10.6

Hello User, just take a moment to pause and focus your creative powers...

:happy-hacking

```
ot777.core=>
```

#(デモ)以下を入力する：

```
ot777.core=> (demo (sin-osc))
#<synth-node[loading]: ot777.core/audition-synth 33>
ot777.core=> (demo 7 (lpf (mix (saw [50 (line 100 1600 5) 101 100.5])))
             #_=> (lin-lin (lf-tri (line 2 20 5)) -1 1 400 4000)))
#<synth-node[loading]: ot777.core/audition-synth 34>
ot777.core=>
```

#以下を入力する：

```
ot777.core=> (definst foo [] (saw 220))
#<instrument: foo>
ot777.core=> (foo)
#<synth-node[loading]: ot777.core/foo 39>
ot777.core=> (stop)
nil
ot777.core=>
```

#以下を入力する(複雑なもの)

```
-----

(definst kick [freq 120 dur 0.3 width 0.5]
  (let [freq-env (* freq (env-gen (perc 0 (* 0.99 dur))))
        env (env-gen (perc 0.01 dur) 1 1 0 1 FREE)
        sqr (* (env-gen (perc 0 0.01)) (pulse (* 2 freq) width))
        src (sin-osc freq-env)
        drum (+ sqr (* env src))]
    (compander drum drum 0.2 1 0.1 0.01 0.01)))

(definst c-hat [amp 0.8 t 0.04]
  (let [env (env-gen (perc 0.001 t) 1 1 0 1 FREE)]
```

```

      noise (white-noise)
      sqr (* (env-gen (perc 0.01 0.04)) (pulse 880 0.2))
      filt (bpf (+ sqr noise) 9000 0.5)]
    (* amp env filt)))

(def metro (metronome 128))

(defn player [beat]
  (at (metro beat) (kick))
  (at (metro (+ 0.5 beat)) (c-hat))
  (apply-by (metro (inc beat)) #'player (inc beat) []))

```

-----

#上のプログラムの実行：

```
ot777.core=> (player (metro))
```

#音を止める：

```
ot777.core=> (stop)
```

## 参考情報

Overtone関連：

<https://github.com/overtone/overtone>

[Overtoneで音楽プログラミング！](#)

[Clojureで音楽を奏でる](#)

[Clojure \(overtone + quil\)によるコーディングとコードを音楽にするライブコーディング環境の試作](#)

[Overtone: Clojureで音楽を書こう](#)

[Clojure で楽器を演奏しよう! Overtone をインストールしてみた](#)

FoxDot関連：

[Pythonで始めるアルゴレイヴ入門](#)

[FoxDot - Live Coding with Python v0.8](#)

[FoxDot（音楽用ライブコーディング環境）をインストール](#)

[Python & SuperColliderで奏でる音楽ライブコーディング環境「FoxDot」をセットアップ（インストール）する方法！](#)

[FoxDotの「超基礎」第一弾！「テンポ設定&サンプルを鳴らす&シンセを鳴らす」で遊び始めろ！](#)

[Pythonでライブコーディングできると聞いて、FoxDotを触ってみた](#)

SuperCollider関連：

[SuperCollider](#)

[Designing Sound in SuperCollider](#)

[オープンソースの音響プログラミング言語「SuperCollider」の紹介](#)  
[SuperCollider自主練 - 基本編](#)

以上

[Go to Toplevel](#)