

サウンド・デザイン

福岡女学院大学 2021年度 前期 木曜2限 第3週

松浦 知也 (teach@matsuuratomoya.com / teach.matsuuratomoya.com)

Puredata入門 基本操作

注意点

- このチュートリアルはMacとWindows両方をカバーしますが、ショートカットキーに関してはWindowsでは
 - Cmdキーと書かれていたらCtrl
 - Optionキーと書かれていたらalt
- と読み替えて進めてください。
- 初出のPd特有の用語はオレンジ色の文字で表記します

Pdとは？

- オープンソースのデータフロー型ビジュアルプログラミング環境
- リアルタイムでの音声信号処理を得意とする
- 幅広い環境で動作する（Raspberry Pi、Unity、Webブラウザ等）
- コミュニティが開発する様々な拡張（**externals**）オブジェクトによってグラフィックプログラミングや物理シミュレーションなど表現の幅を広げられる

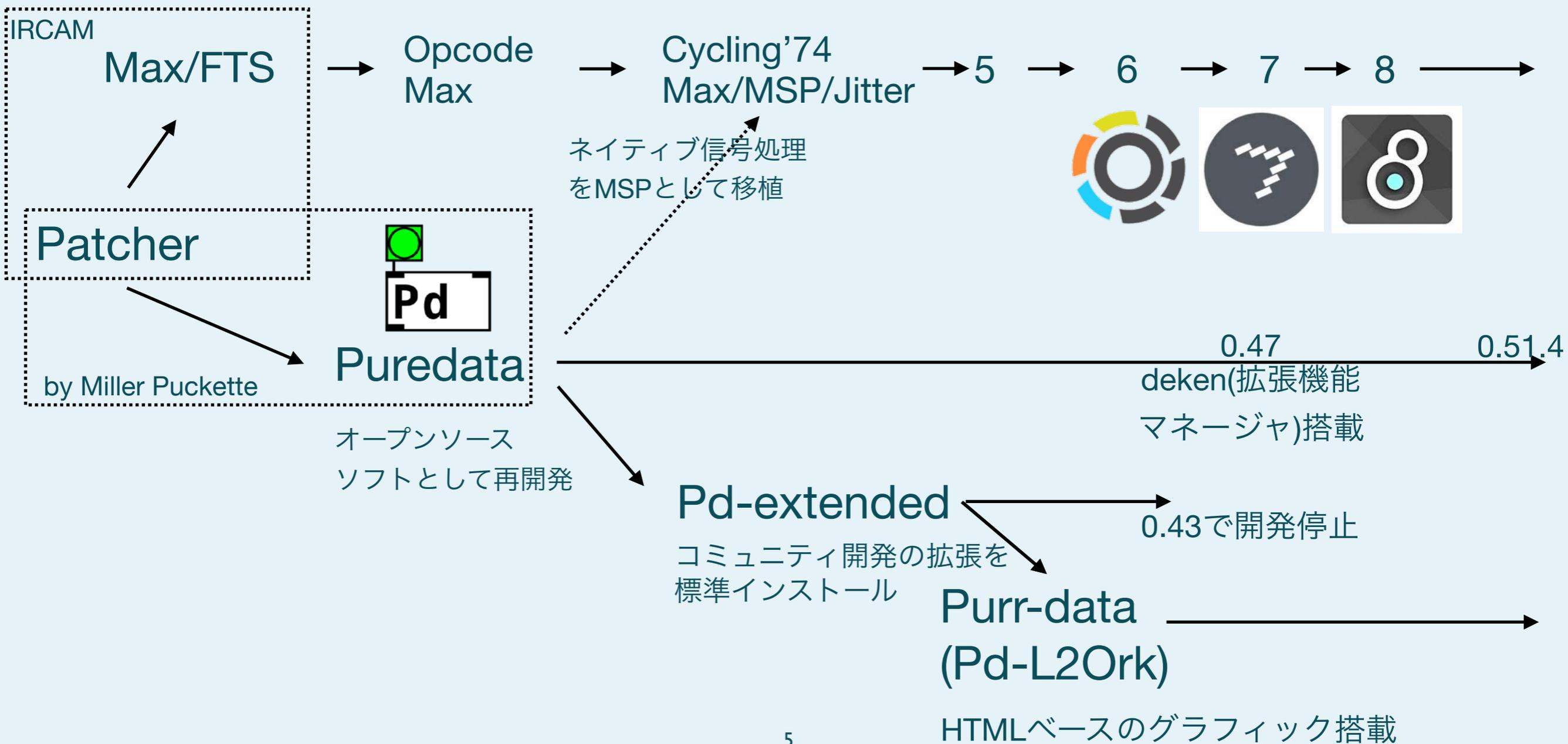
歴史

1990

2000

2010

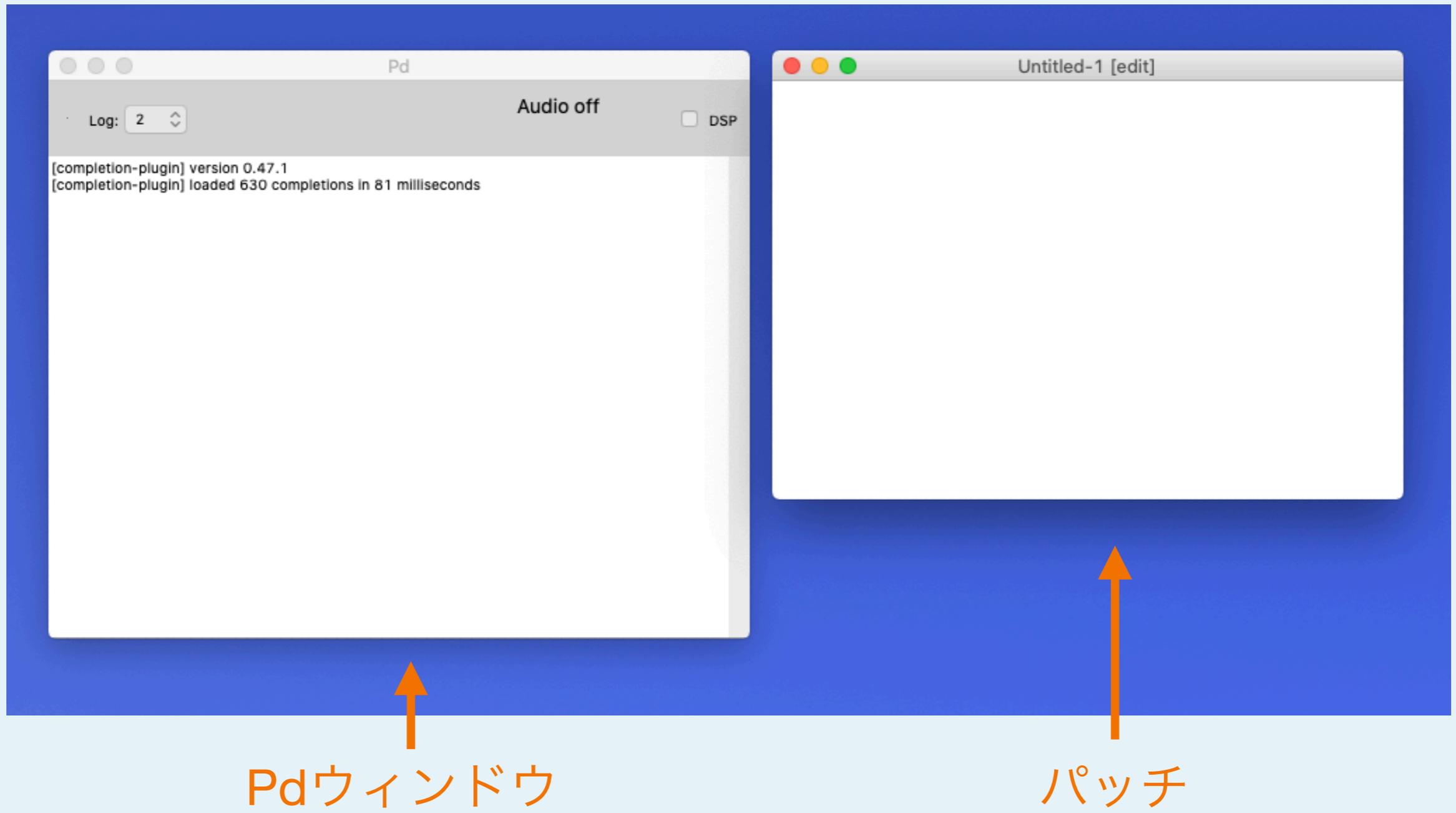
2020



新規作成

- Cmd+Nで新しいウィンドウが開く。この画面およびこれを保存した.pdファイルを**パッチ**と呼ぶ。
- オーディオテスト画面や各オブジェクトを右クリックして見られるヘルプ画面それ自体もパッチとして作られている。なので、ヘルプやexampleから直接コピーペーストもできる
- 起動した時に表示される、またはCmd+Rで出てくるウィンドウを**Pdウィンドウ**と呼ぶ。Pd実行エンジンのログとDSPエンジンon/offボタンがついている。

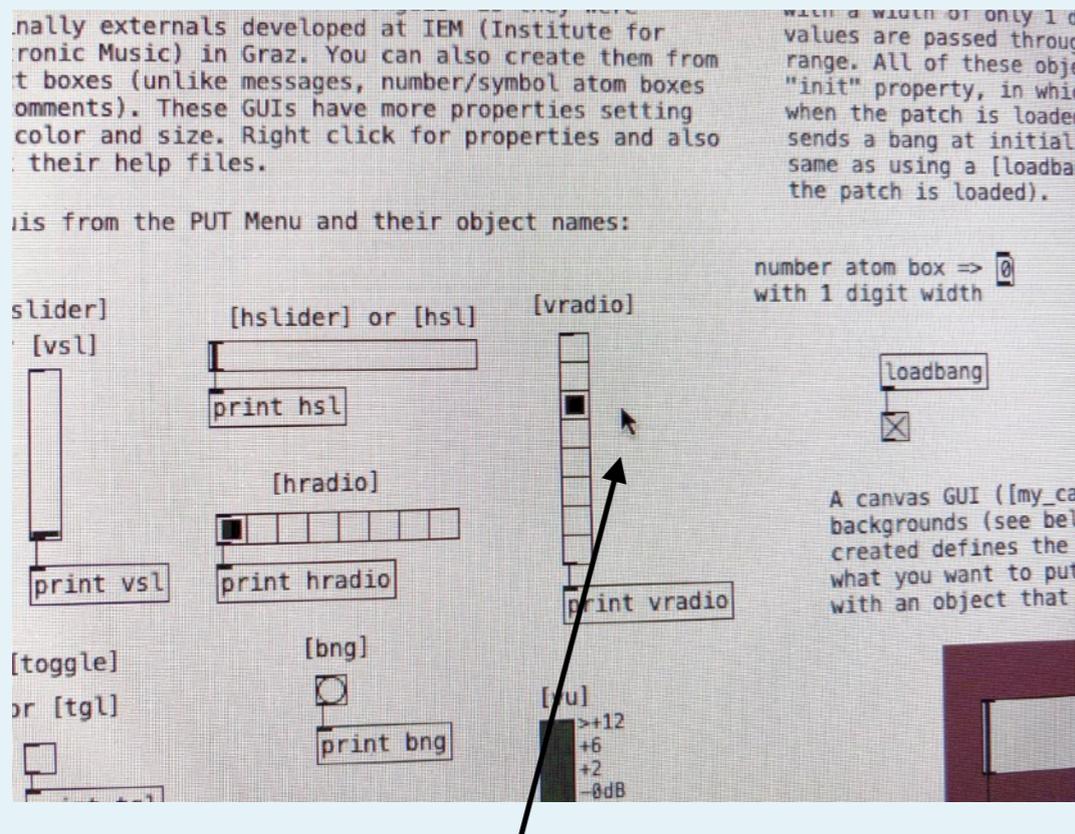
新規作成



編集モードの切り替え

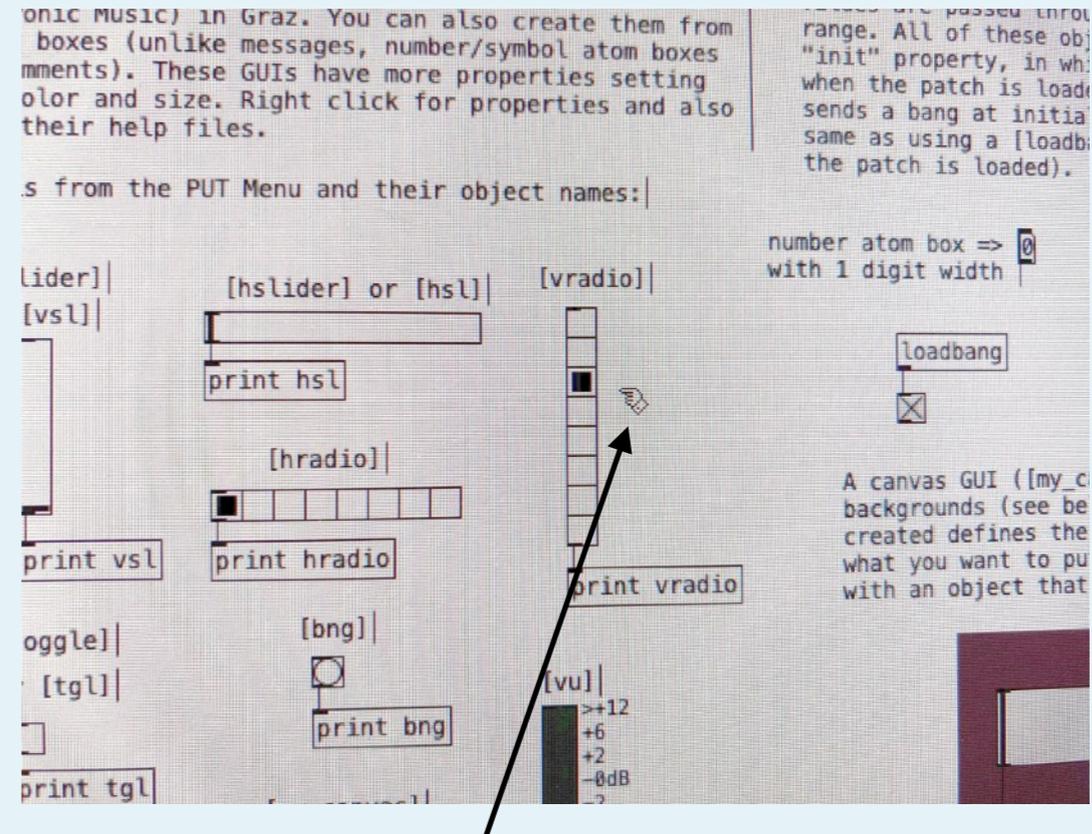
Cmd+e

実行モード



ボタンを実際に押せる、
数値を変更できる

編集モード

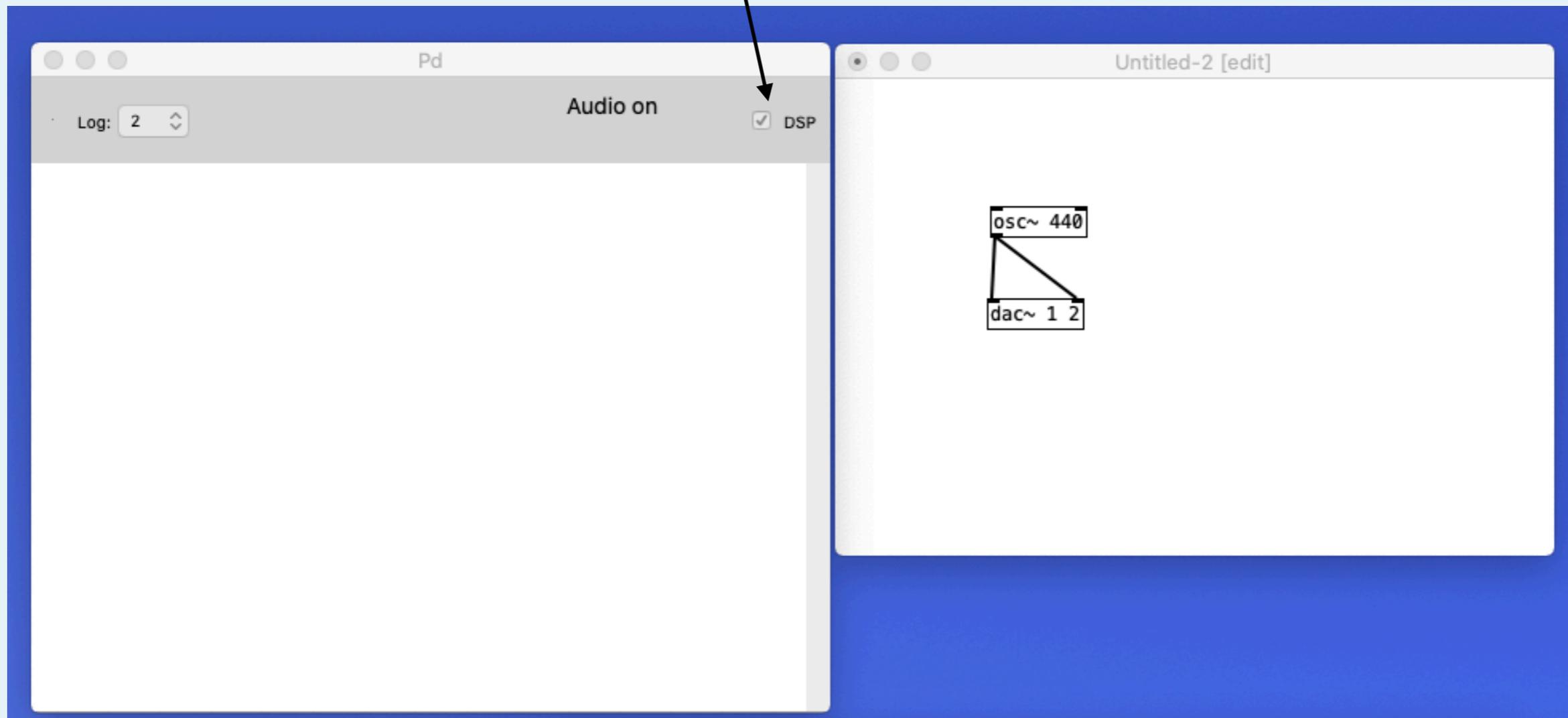


ボタンそのものの配置や移動

- Cmdキーを押しながらマウス操作で一時的に実行/編集を反転できる

とりあえず音を出そう

チェックを入れる

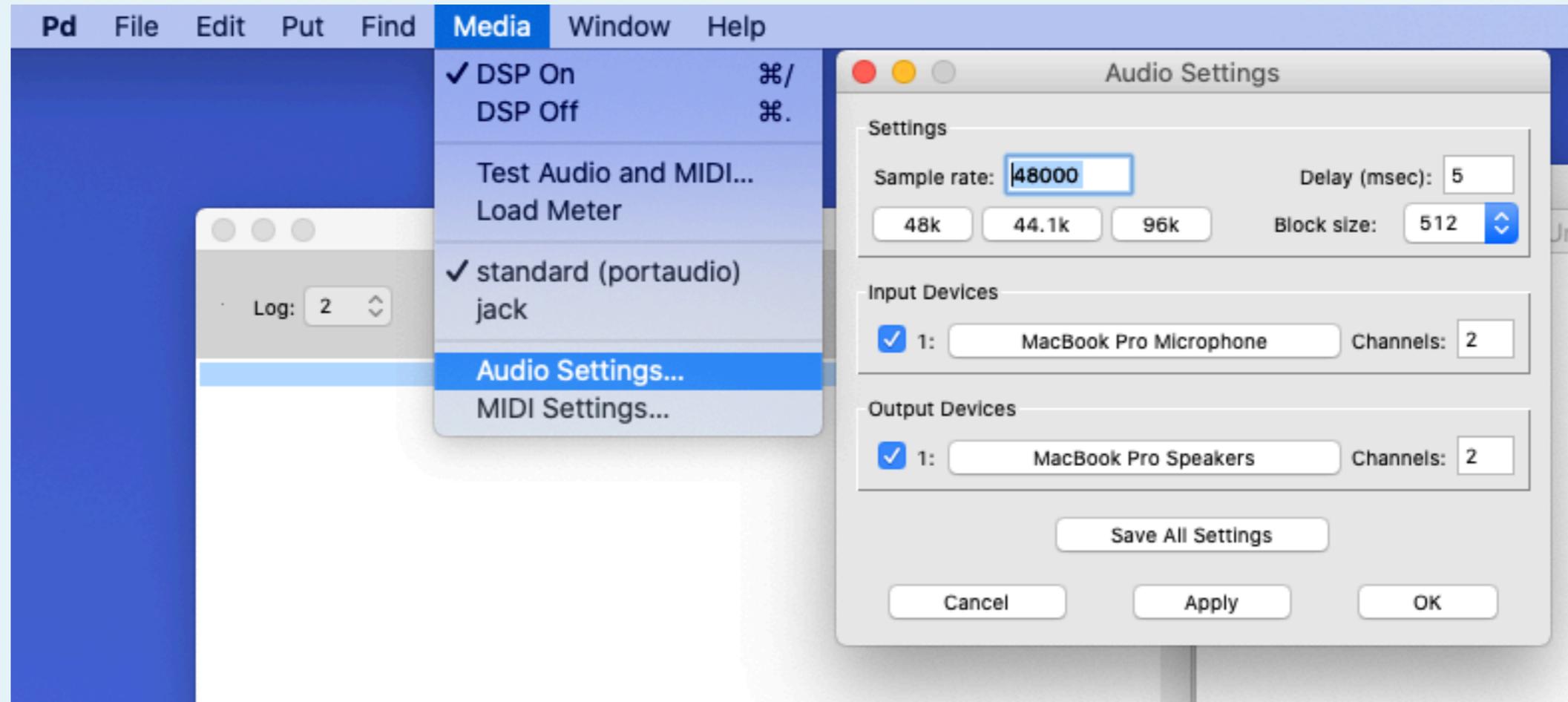


システムの音量に注意！！！！

とりあえず音を出そう

- Put->Objectを選択(Cmd+1)しオブジェクトを新規作成
 - “osc~ 440”と入力し確定(半角で、数字の前は半角スペース)
 - もう1つ作成し、“dac~ 1 2”と入力
- osc~の左下の黒い角へマウスを持っていくと、カーソルが○に
 - ドラッグしてdac~ の左上にドロップして線を繋ぐ
 - 同様に右上にもドラッグ&ドロップ

音が出ない時は

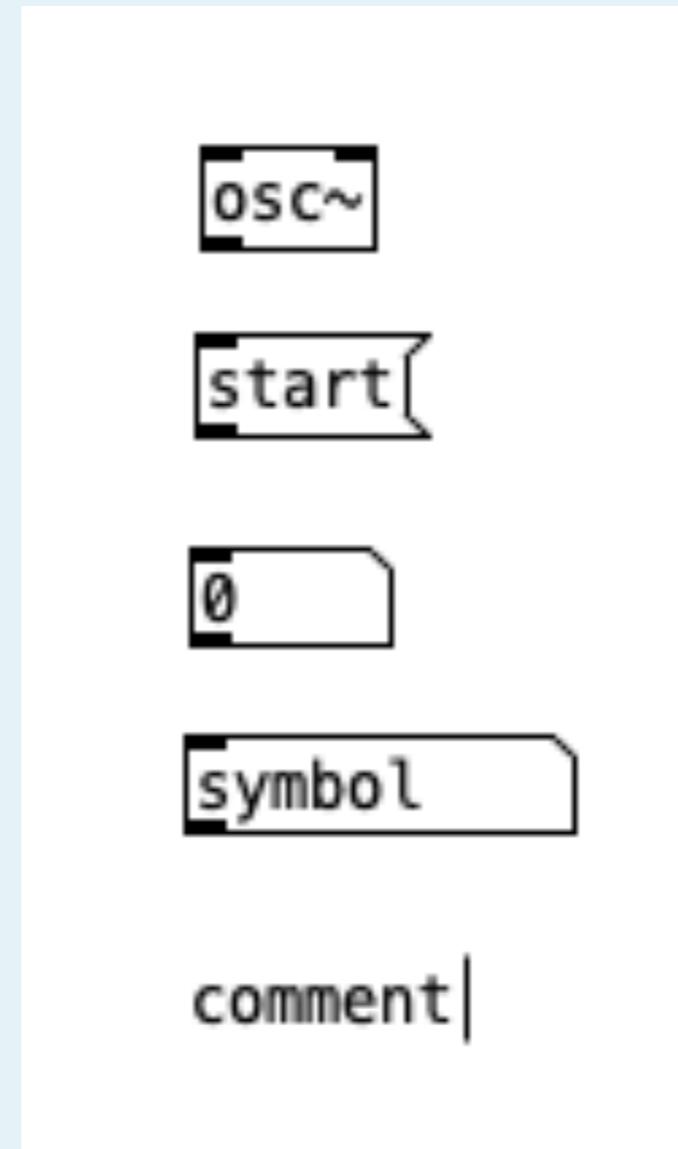


- DSP Onになっているか確認
- デバイスの名前を確認
- Block Sizeは128~512あたりがおすすめ
 - 小さいほど処理は重いがレイテンシ(入力に対する反応速度)が速くなる

基本的なオブジェクトの配置

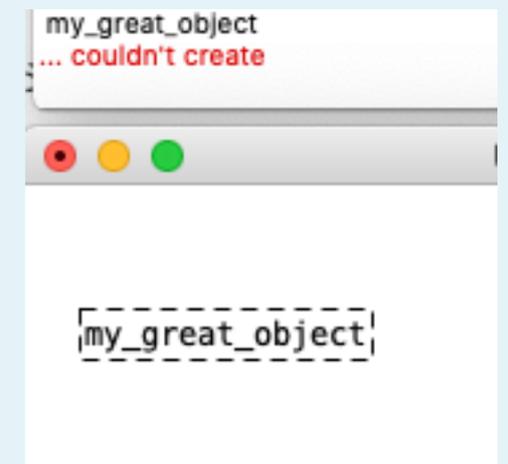
Cmd+1~5

- Cmd+1: オブジェクト [osc~]
- Cmd+2: メッセージ [start<
- Cmd+3: 数値
- Cmd+4: シンボル
- Cmd+5: コメント



オブジェクト

- Cmd+1でオブジェクトを作り、名前を入力するとその名前に応じた機能を持つボックスが作られる。
- オブジェクトは初めから用意されているもの(HelpのList of Objectsを参照)、自分でインストールしたexternalのどれかを入力する。またPdパッチのファイル名自体を指定して埋め込むこともできる。
(Abstraction)
- 入力した名前に対応するオブジェクトが見つからなかった場合、点線で表示されてエラーになる。
- オブジェクトによっては追加のパラメーター(Argument)をスペース区切りで入力できる。



ヘルプ

The image shows a Pure Data patch window titled "Untitled-2" and its help window titled "osc~-help.pd".

Untitled-2: Contains an `osc~ 440` object with a context menu open showing "Properties", "Open", and "Help" (highlighted). Below it is a `dac~ 1 2` object.

osc~-help.pd: Contains the following text and diagram:

osc~ - cosine wave oscillator

The `osc~` object outputs a cosine wave. If no argument is supplied, the input is taken to be an audio signal. With a floating-point argument, `osc~` takes floating-point messages to change frequency.

`; pd dsp 1; metro 1` <= click to start
`; metro 0` <= click to stop

incoming numbers change frequency: `0` (inlet) → `osc~ 1000` (creation argument sets initial frequency)

v— messages to right inlet set phase in cycles

amplitude controls:

- `0.1 100` on
- `0 100` off
- `line~` envelope generator

frequency control: `0` (inlet) → `sig~` (convert to audio signal oscillator) → `osc~` (invoked without argument to specify audio signal input)

graph the output: `tabwrite~ $0-array` → `graph $0-array`

audio output: `*~` → `dac~`

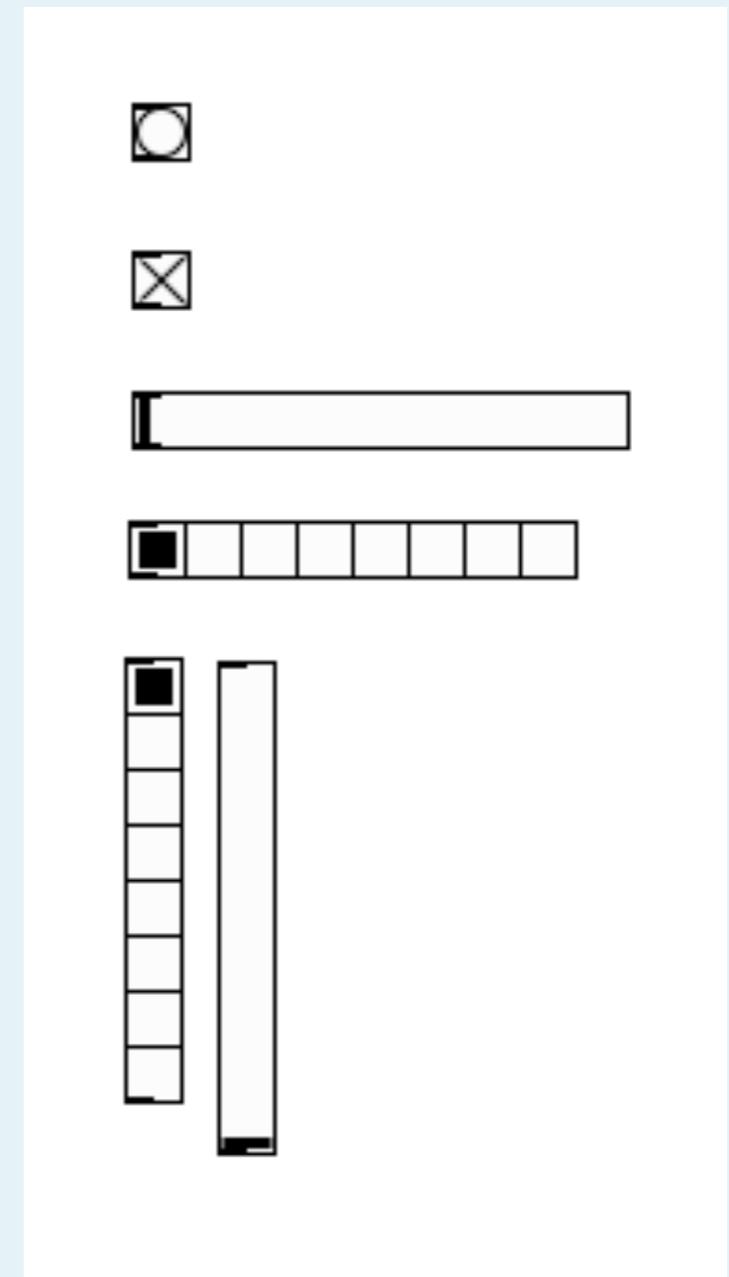
see also: `phasor~` `cos~` `tabread4~`

updated for Pd version 0.33

- オブジェクトを右クリック（編集/実行モード問わず）してHelpを選択
- 出てきたHelp自体もパッチなので編集/コピーできる（ただし上書き保存しないよう注意）

UIオブジェクト

- [bng] ボタン(Cmd+Shift+b)
- [tgl] 0/1切替ボタン(Cmd+Shift+t)
- [hslider][vslider]横/縦スライダー
- [hradio][vradio]ラジオボタン



InletとOutlet

- 全てのボックスは上側のInletでメッセージ/シグナルを受け取り、各オブジェクトの機能に応じて下側のOutletからメッセージ/シグナルを出力する。
- 同じオブジェクトでもArgumentの内容でInlet/Outletの数が変わる場合がある

MessageとSignal

- オブジェクト同士を繋ぐ線にはMessageとSignalの2種類がある。
- 主にMessageを離散的な処理に、Signalを連続的な音声信号に使う。SignalはDSPがオンになっていないと動かない。
- InletによってMessageのみ受け付けるものもある。
- [+]と[+~]、[*]と[*~]のように、同じ機能でもMessageとSignalで別々にオブジェクトが存在することがある。慣例的に、Signalを扱うオブジェクトの名前には~(tilde、チルダ)が付いている。
- MessageをSignalに変換するには[sig~]、[line~]、SignalをMessageに変換するには[snapshot~]などを使う。

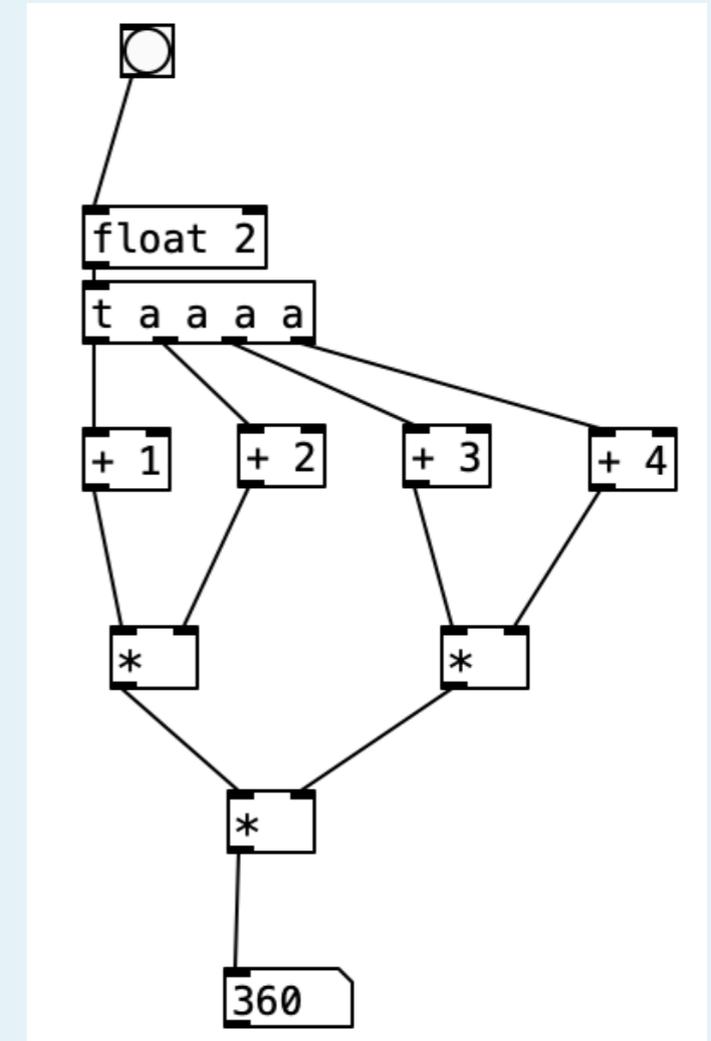
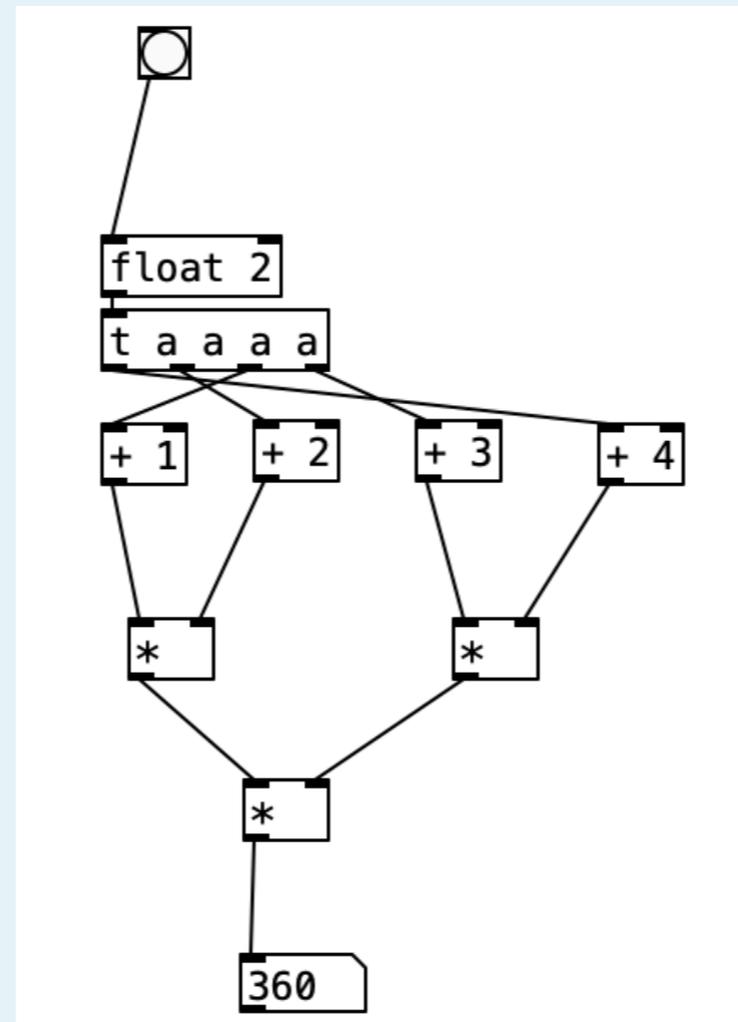
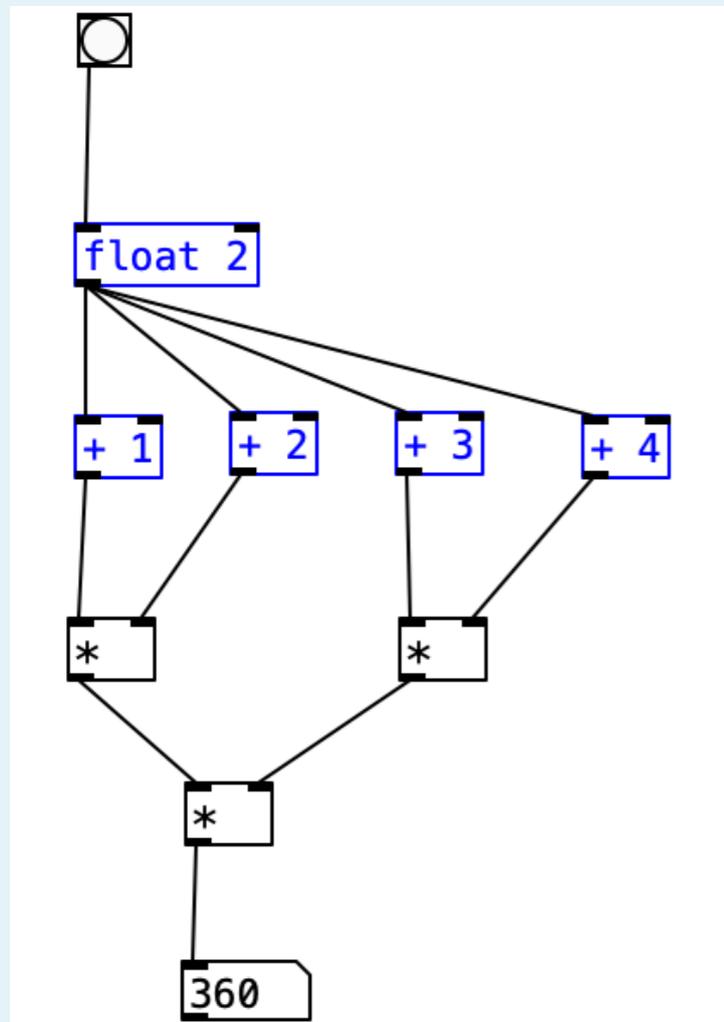
Hot/Cold Inlet

- メッセージを受け取るInletは、以下の2種類ある
 - Outletからメッセージをトリガーする **Hot Inlet**
 - 内部状態を変更するだけで出力はしない **Cold Inlet**
- 基本的には、一番左のInletがHot、それ以外はCold（例外あり）

メッセージの処理順の制御

- 1つのアウトレットから複数メッセージが出ている場合(fan-out)、**初めに繋いだオブジェクト**から順番にメッセージが出力される。
- 同じ見た目でも、処理の順番が変わってしまって計算結果が変わってしまうことがある
- [trigger]オブジェクトで処理の順番を制御できる。[trigger any any any]とすると、入力されたメッセージを3つのアウトレットに、右から順番に出力する。([t a a a]と略せる)

Cmd+T : “Triggerize”



- 一つのアウトレットから複数メッセージが出ている場所を選択してCmd+Tを押すとtriggerオブジェクトを挟んでくれる。右から順に処理されるように順番を変更する

よく使うので省略できるもの

bang	b
float	f
int	i
send hoge	s hoge
receive hoge	r hoge
send~ fuga	s~ fuga
receive~ fuga	r~ fuga
trigger any bang float symbol	t a b f s
select any bang float symbol	sel a b f s

(余談：Maxでもほぼ同様に省略ができるが、send~とreceive~だけはなぜか省略できないので注意)