

# 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4週

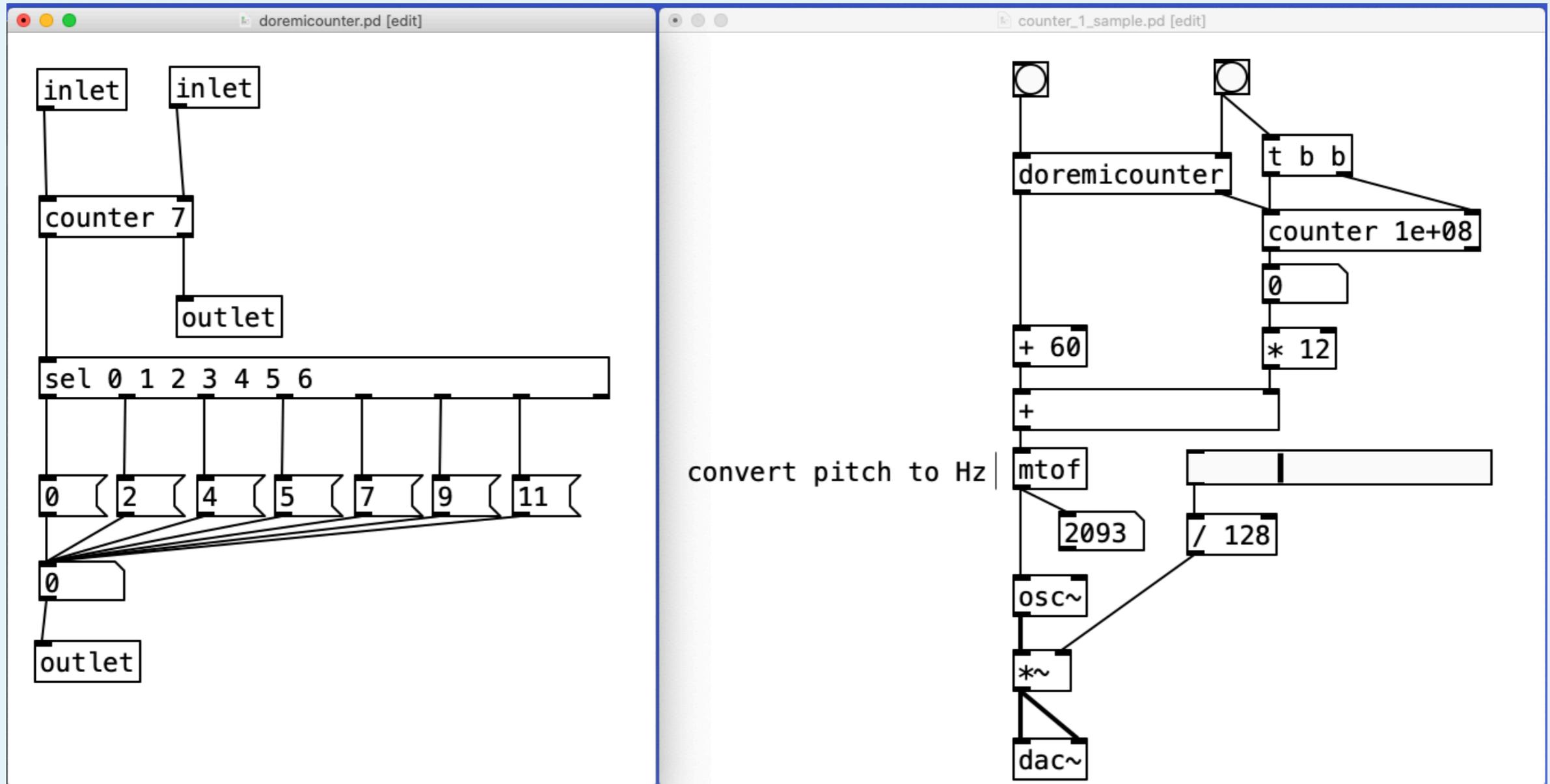
松浦 知也 ([teach@matsuuratomoya.com](mailto:teach@matsuuratomoya.com) / [teach.matsuuratomoya.com](http://teach.matsuuratomoya.com))

# Puredata入門 基本操作2

# セーブ

- Cmd+Sでファイルを保存
- 拡張子は.pd
- ファイルをダブルクリックすると直接Pdが立ち上がるはず

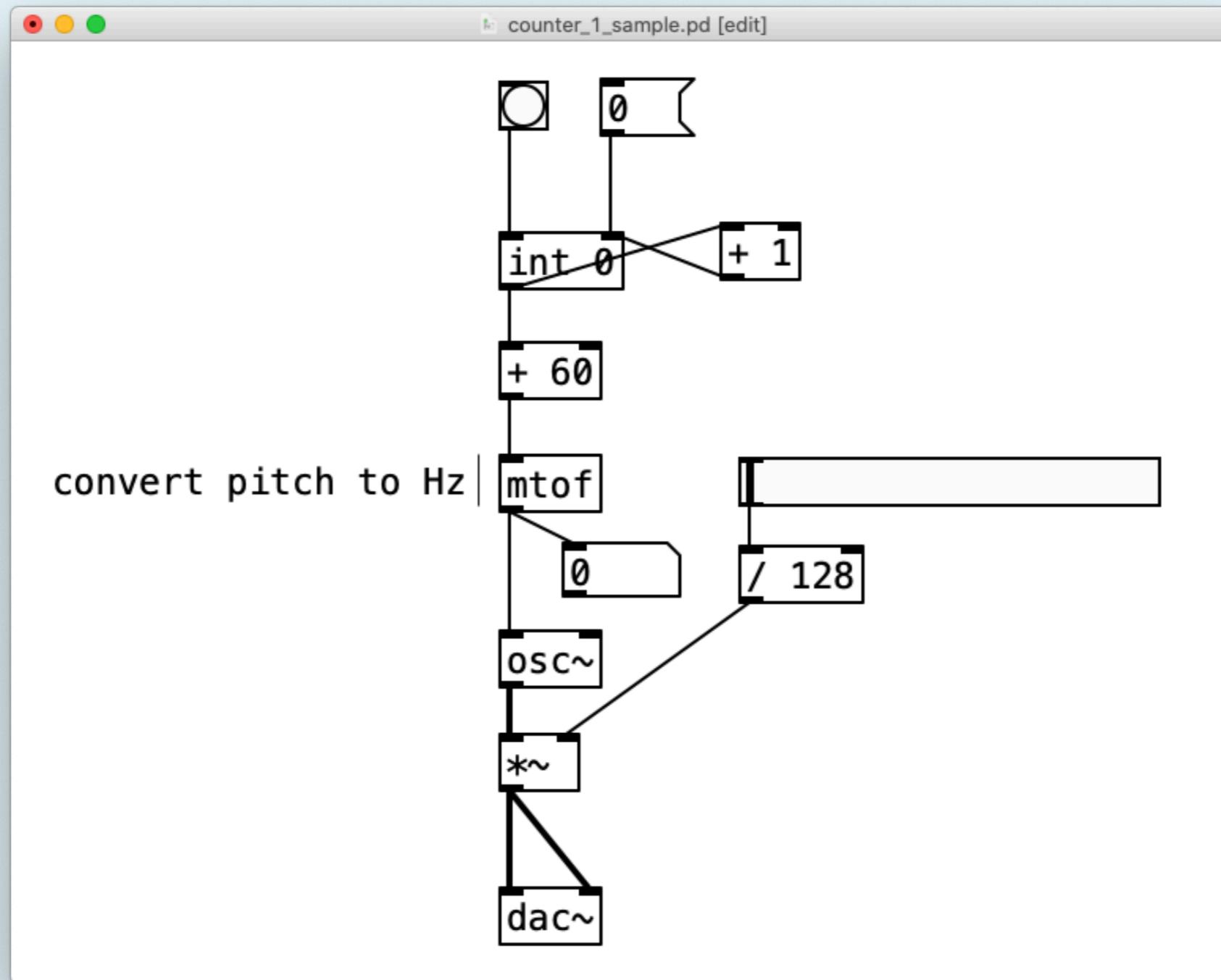
# 今日作るもの



- クリックするたびドレミファソラシドとピッチを上げていくカウンター

# ステップ1

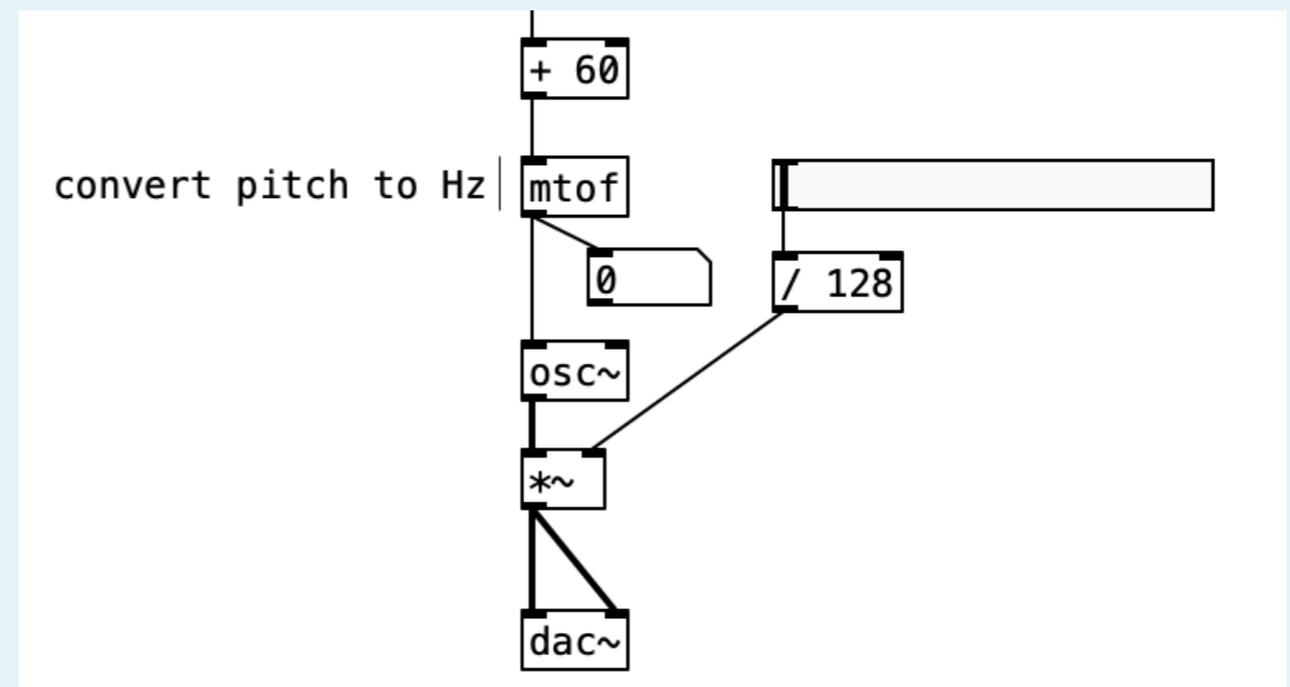
## 半音ずつ上がるカウンター



# ステップ1

## 下準備

- [mtof]オブジェクト：ドレミ（midiノート番号）を周波数に変換してくれる。60が中央のド
- 周波数を[osc~]に接続
- [\*~]とスライダーで音量を制御
- スライダーの範囲は標準0~127なので[/ 127]する（プロパティから範囲変更してもよい）



# ステップ1

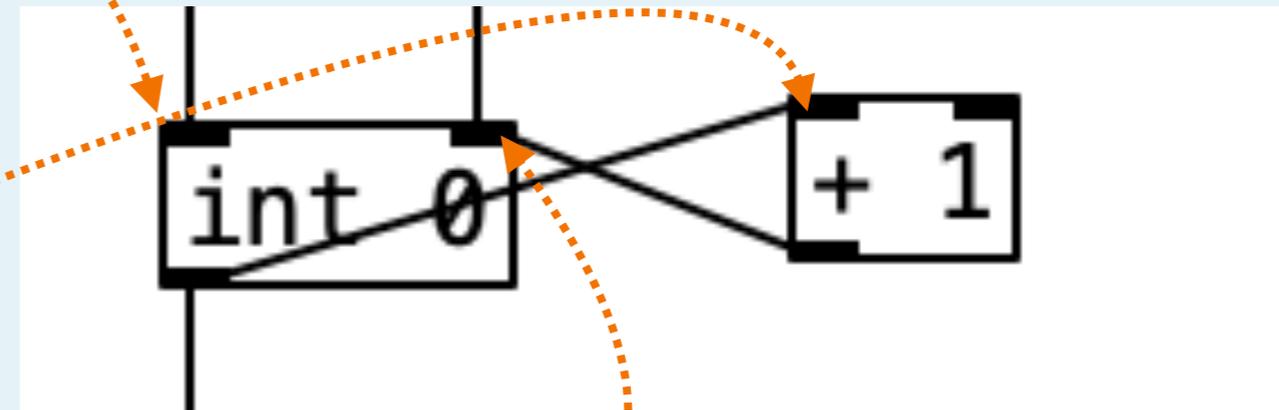
## タスキがけ：カウンターの基本系

1. [int 0]の左inletにbangが入ってくると中に保存されている数値（初期値0）を出力

2. [+ 1]に数字が入り1を足した数値を出力

3. [int 0]の右inletに1足された数値が入り、中に保存されている数値を書き換え（出力はしない）

4. 1.へ戻る

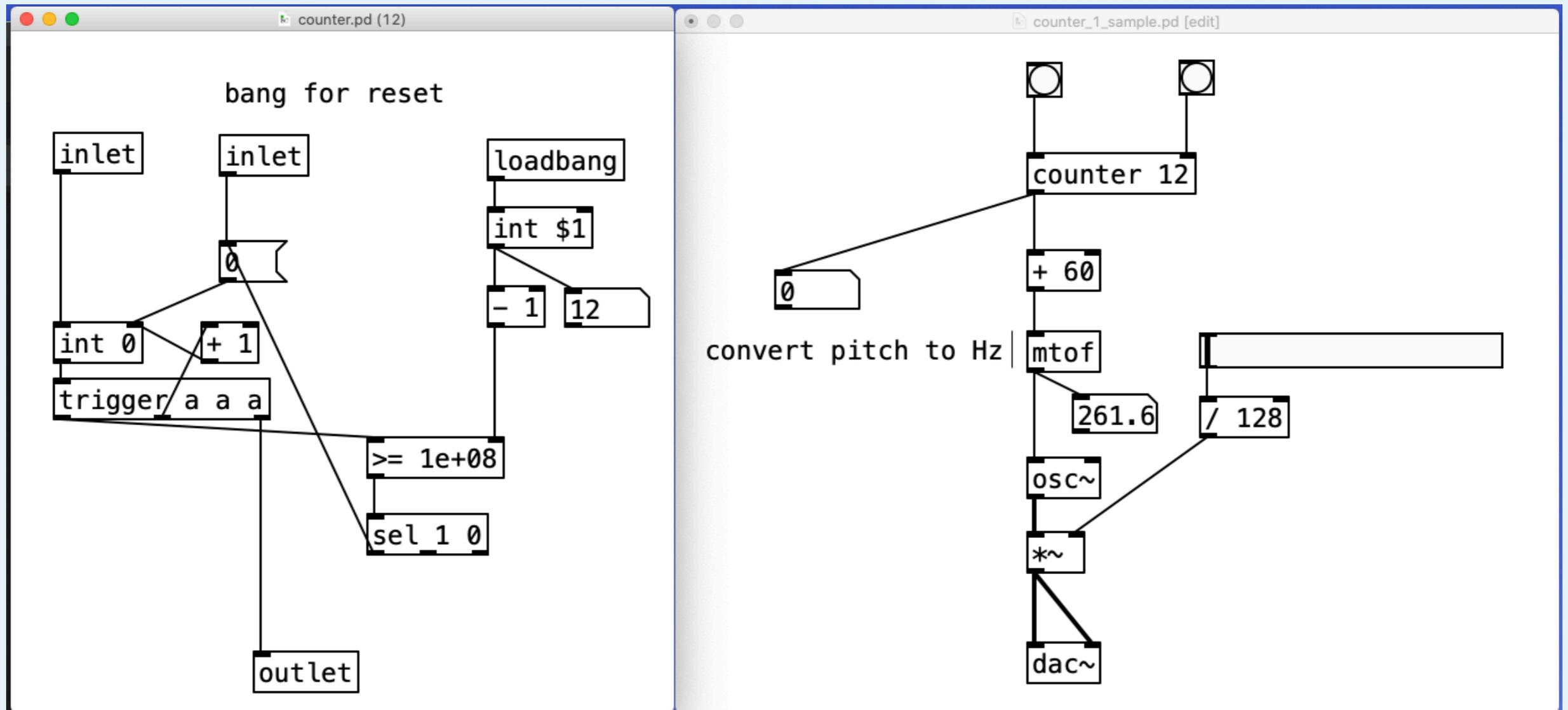


# Hot/Cold Inlet

- 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）

# ステップ2

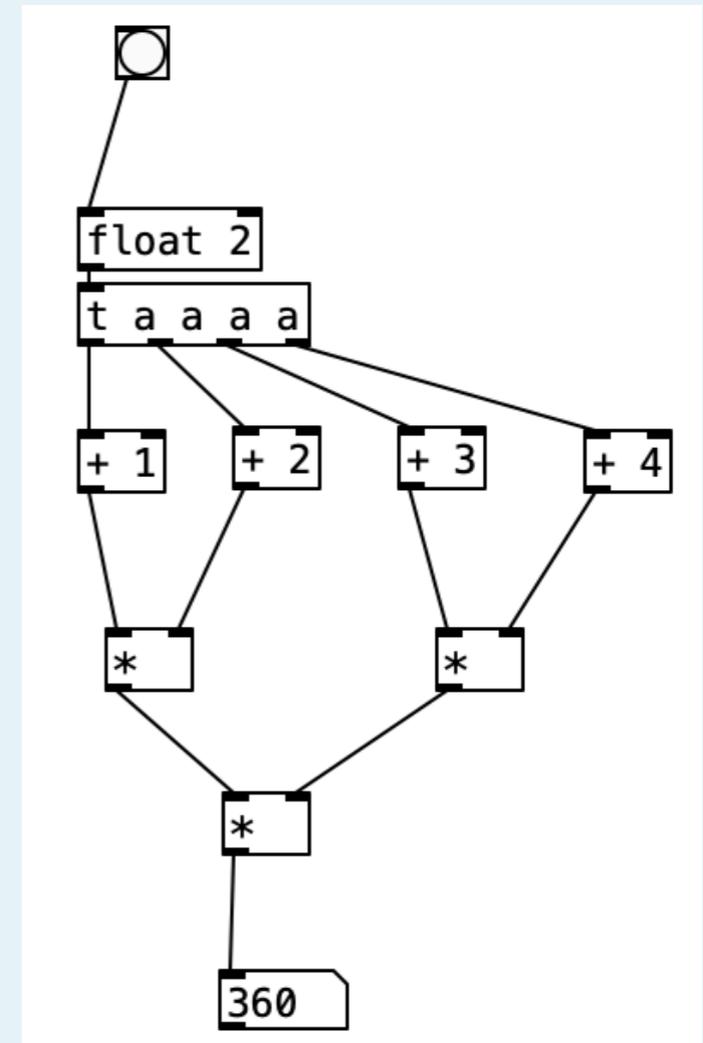
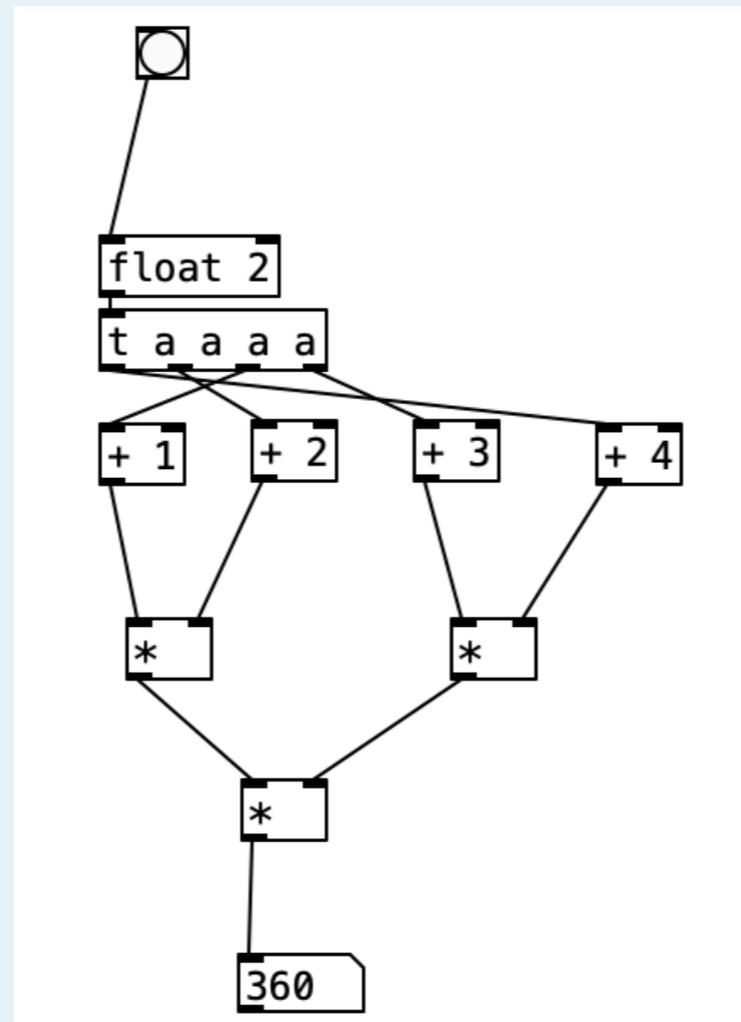
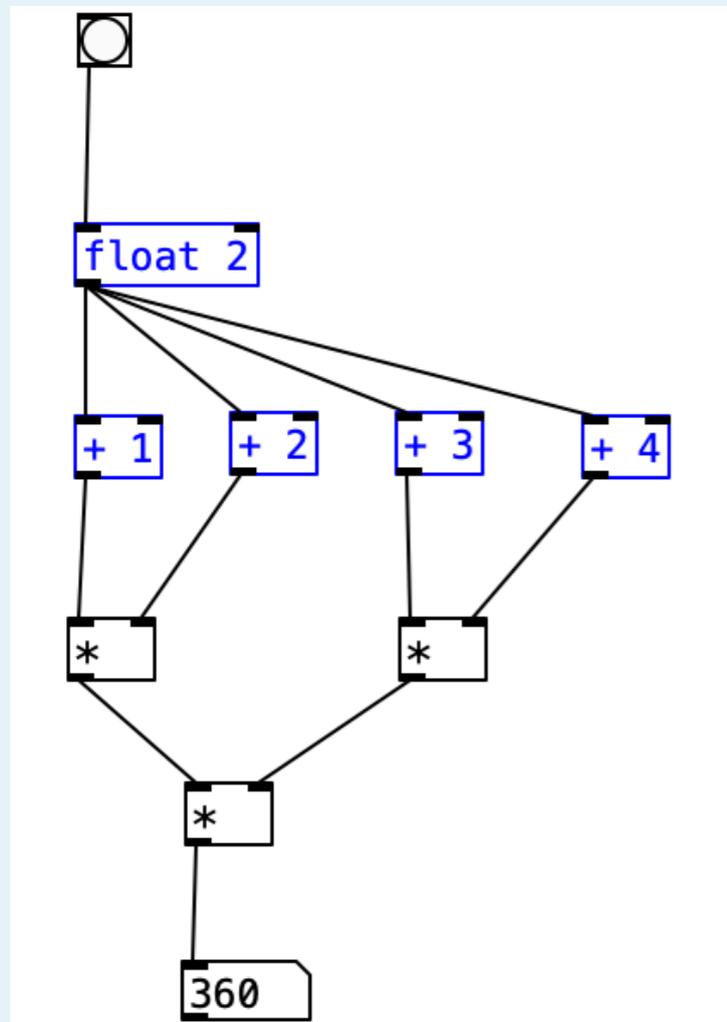
## 1オクターブでループするカウンター



# 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

# [select]([sel])オブジェクト

- [sel 0 1 2 3 4]というオブジェクトを作ると、outletが5個のオブジェクトができる
- 入ってきた値が0,1,2,3,4のいずれかだったら、それぞれ対応しているoutletからbangを出す
- 5や100,"symbol"などどれにも当てはまらない値だったら一番右のアウトレットにその値を流す
- [>][>=][<][<=][==][!=]などの比較オブジェクトと[sel 0 1]をコンビで使おう
- 普通のプログラミングにおけるif文（条件分岐）の機能に相当

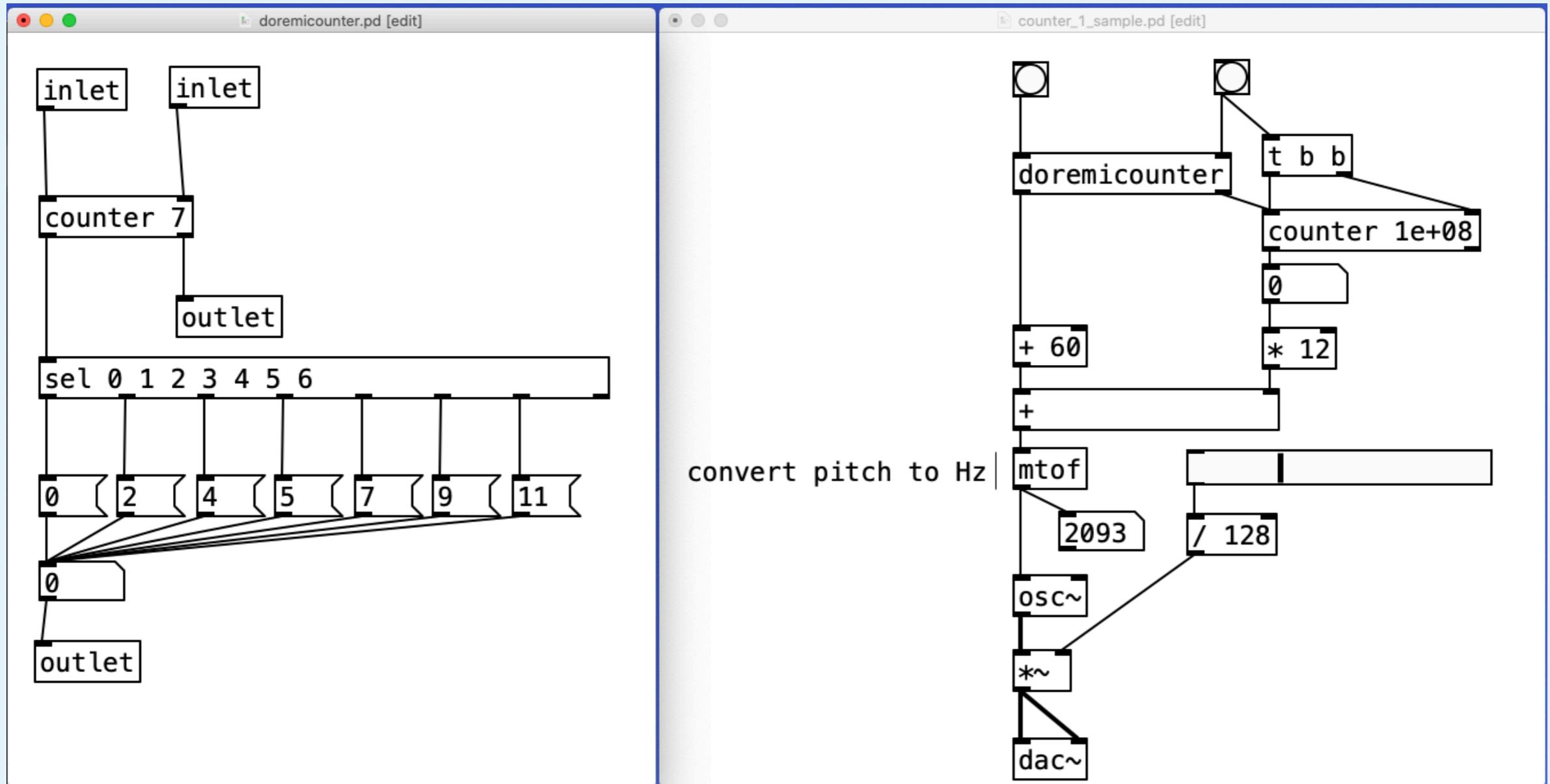
# Abstraction

- 自分で作った.pdファイルをオリジナルのオブジェクトとして読み込むことができる
- 例えばmain.pdというファイル上で[counter]というオブジェクトを作ると、main.pdと同じディレクトリにあるcounter.pdが読み込まれる
- [inlet][outlet]オブジェクトを使って入出力を生やすことができる

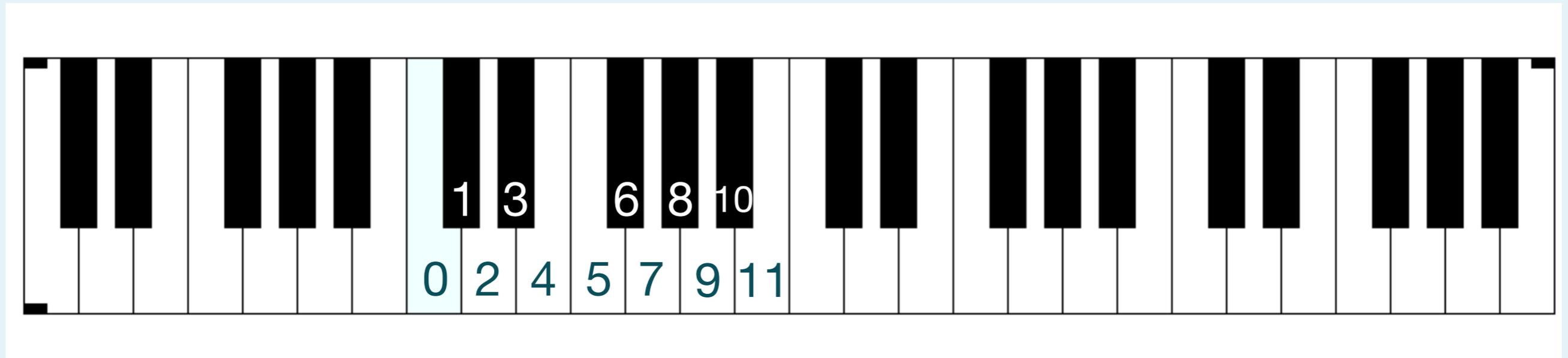
# Argument

- Inlet/outletで制御する代わりに、[counter 12]のように引数を指定してAbstractionの挙動を変えることもできる
- counter.pdを編集して、[int \$1]という形で\$マークに続けて1,2,3を指定すると、Abstractionで呼び出された時の引数 (Argument) を取得できる

# ステップ3



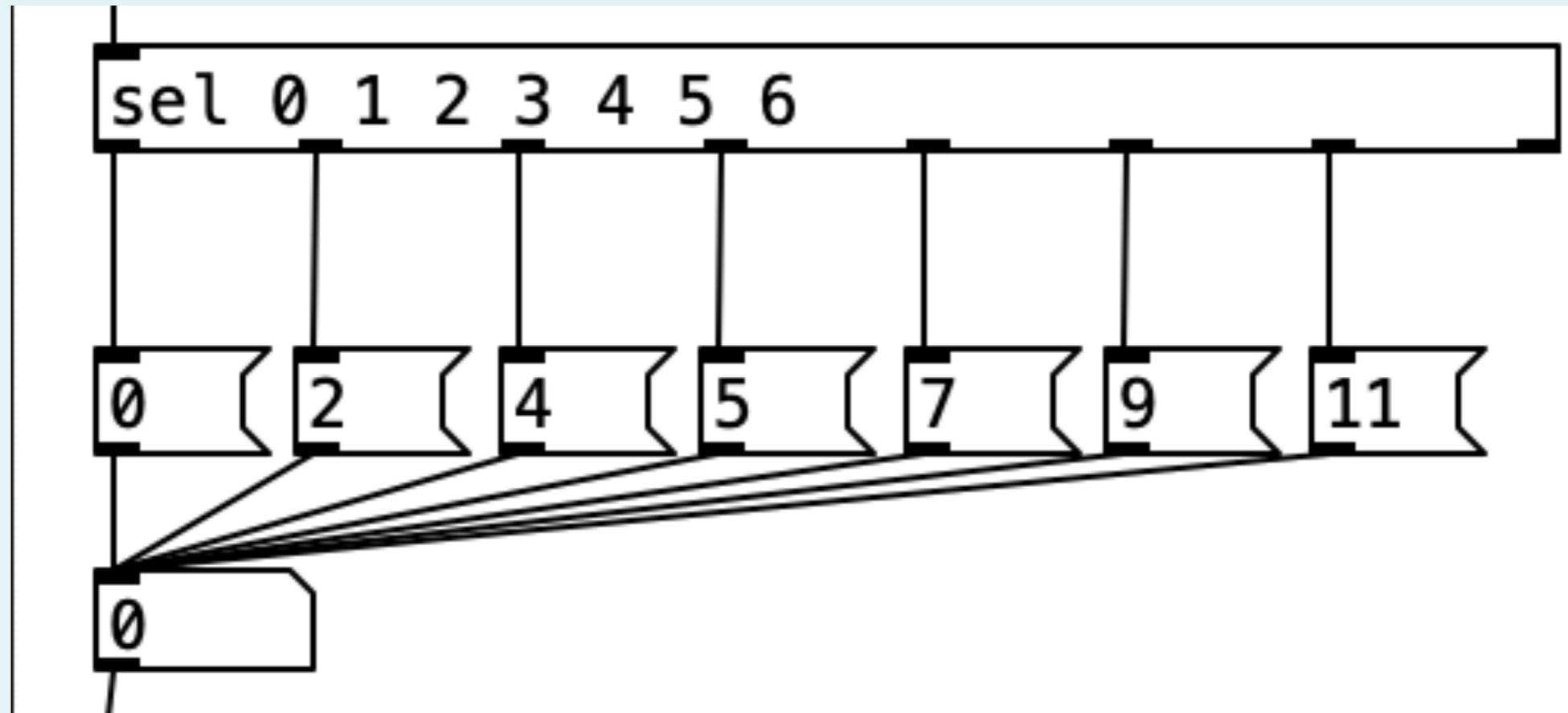
# 半音階からドレミに



- 白鍵と半音0~12の対応関係は図のようになっている
- 0,1,2,3,4,5,6 を
- 0,2,4,5,7,9,11へうまくマッピングしてあげるのがよさそう

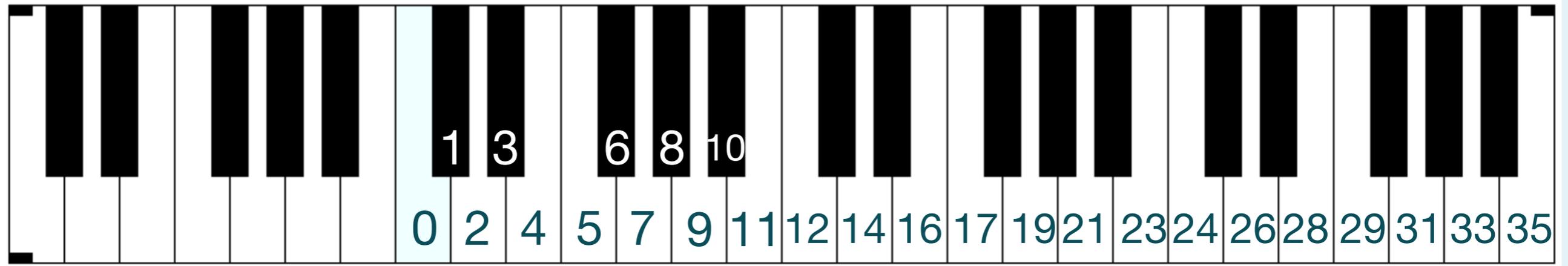
# 半音階からドレミに

↓0~6が入力される

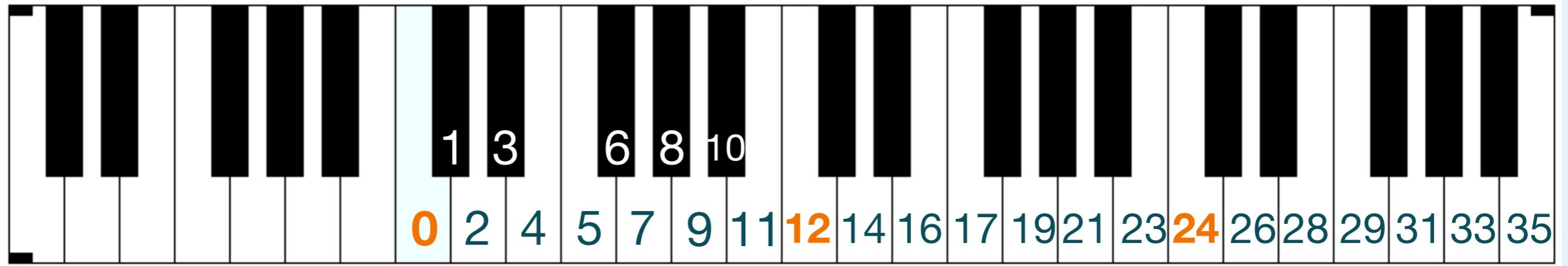


- とりあえずselで愚直に実装

# オクターブ上がるようにしたい

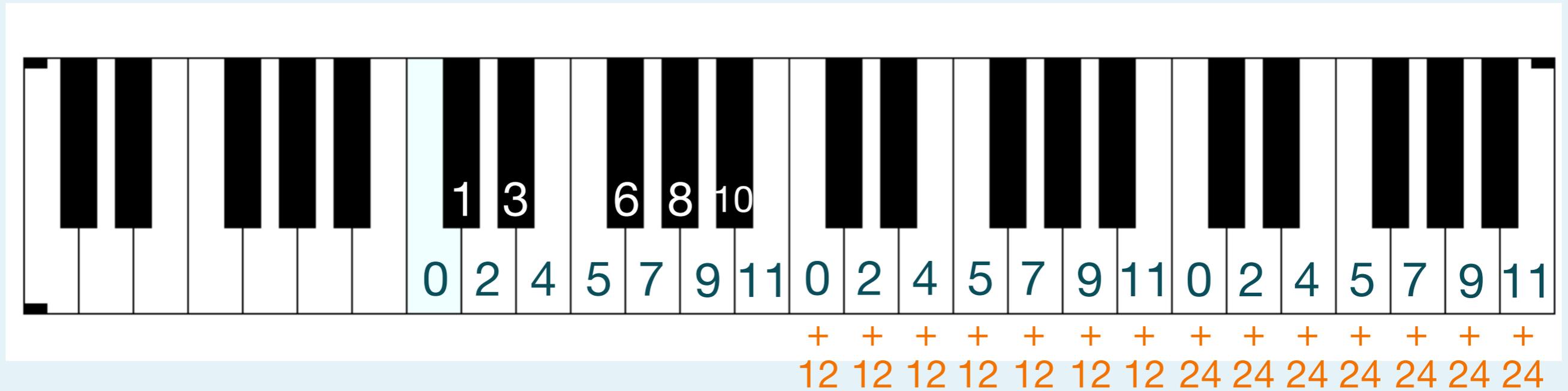


# オクターブ上がるようにしたい



- オクターブの関係に着目しよう

# オクターブ上がるようにしたい

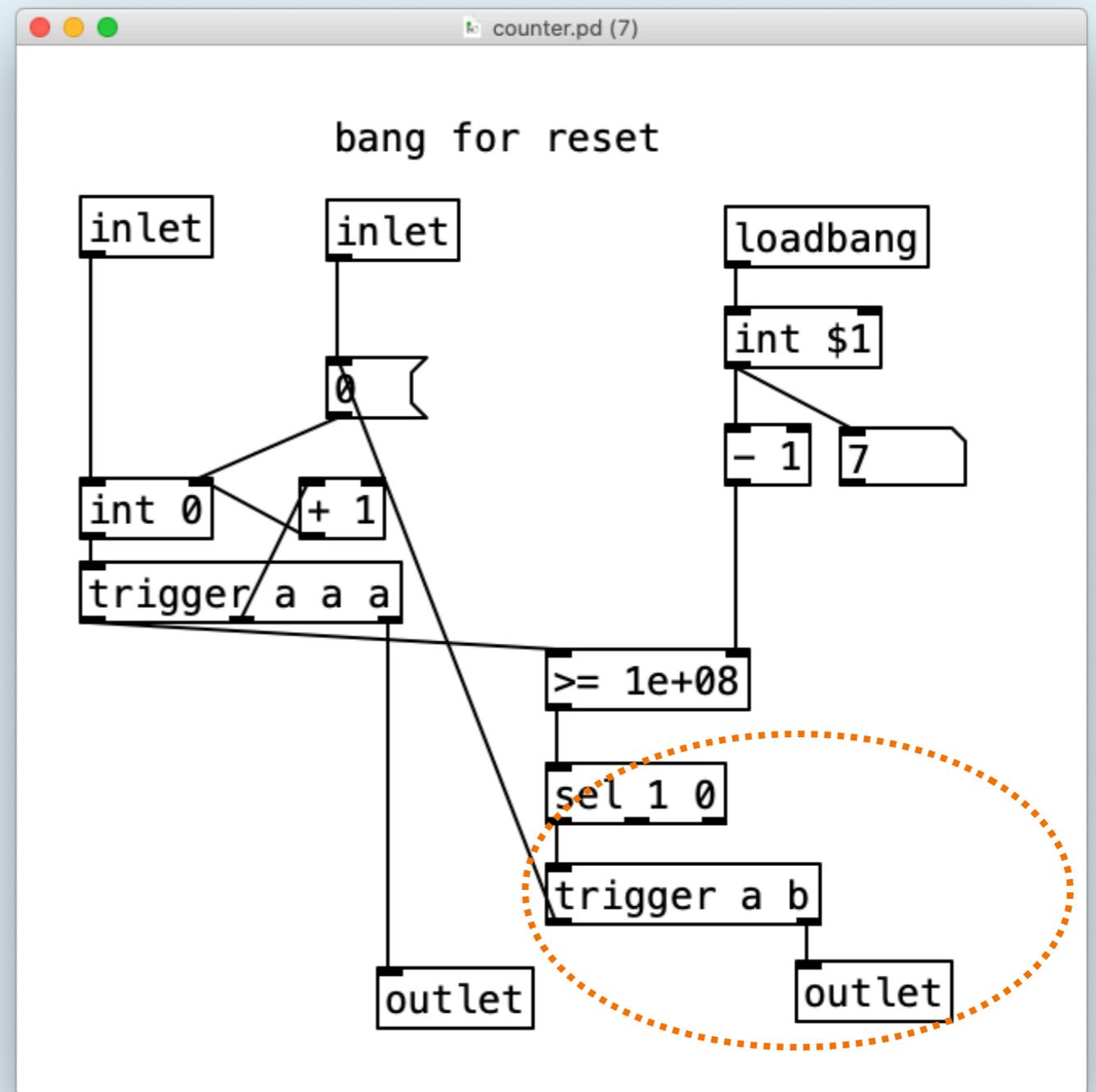


- オクターブ上がるごとに12足すと考えれば良い



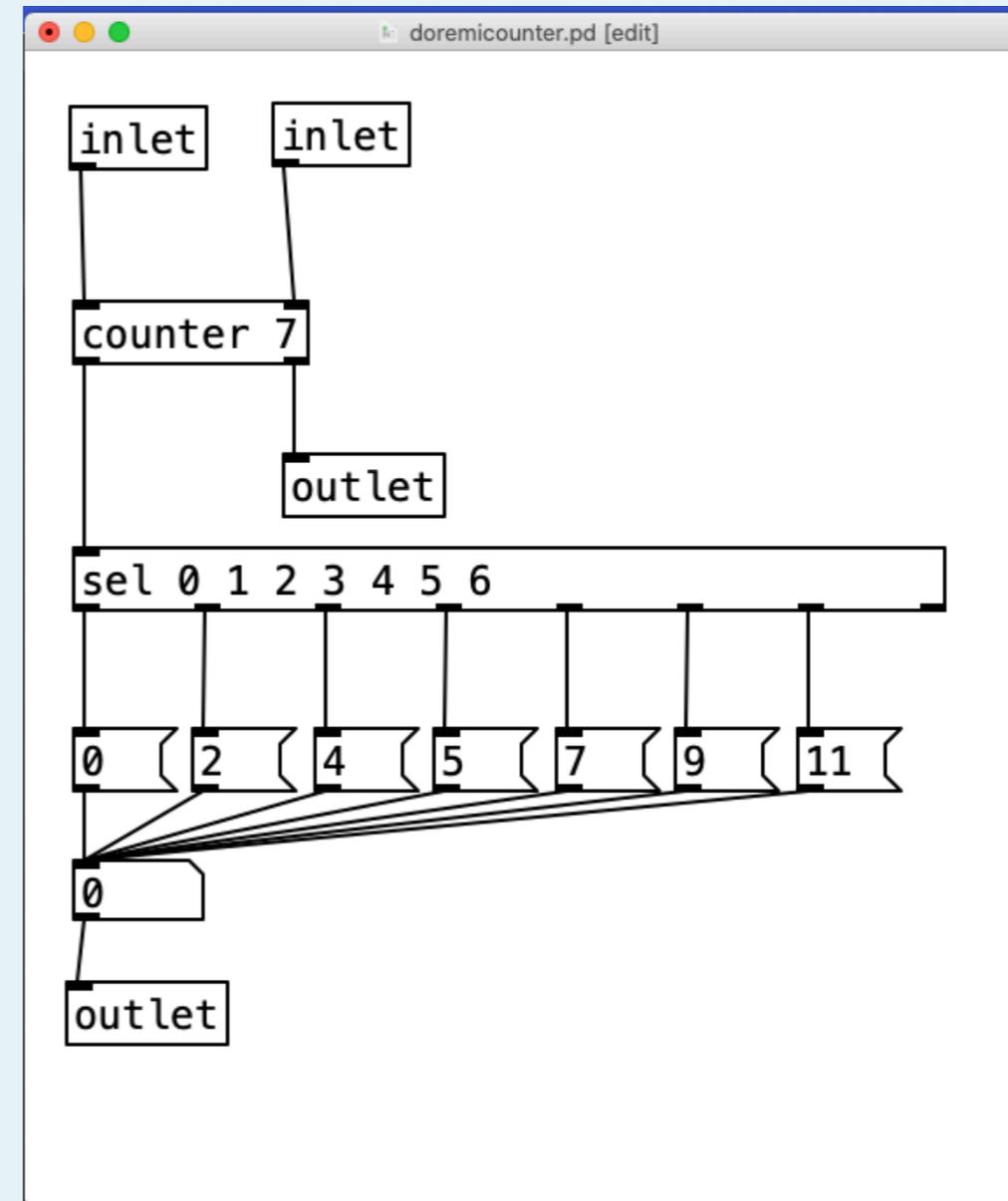
# オクターブ上がるようにしたい

- Counter.pdに1回ループするたびbangを出力するoutletを新たに追加しよう
- N進数の桁上がりを表現するのに使われるのでcarry outと呼んだりする



# オクターブ上がるようにしたい

- 改造したcounter.pdを使って doremicounter.pdを右の図のように作る
- 0,2,4,5,7,9,11を順番に出力し、1ループしたら右アウトレットからbangを出力



# ステップ3

- doremicounterのcarry outをオクターブ用に新たに作ったcounter（上限は99999999とか最大値を入力しておこう）に繋ぐ

