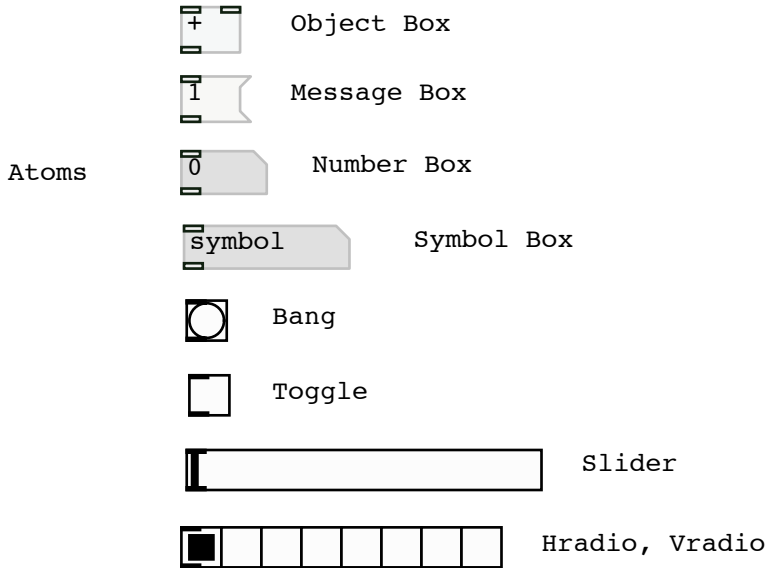


Terminology



The atoms have upper side entrances and exit at their bottom, so the reading of a patch is from the top down.

The use of the objects is different: a message box is clicked in order to trigger the message contained within. Number Box values are changed by holding the mouse with an upward or downward pull. The toggle is enabled / disabled by clicking on it. And a slider is drawn with the mouse from side to side.

Getting_Started

Effective Programming (patching) can only use the keyboard shortcuts. One hand is on the mouse, the other on the keyboard.

ctrl-1 Creates a new object:



Under OS X, all shortcuts will also work with the Apple key.

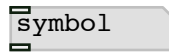
ctrl-2 generates a message box:



ctrl-3 generates a number box:




ctrl-4 produces a symbol field:




ctrl-5 generates a response, as it were a label

All other objects can be inserted via the menu "Put".

The Bang-object can be created with ctrl-1 and the entry "bng"

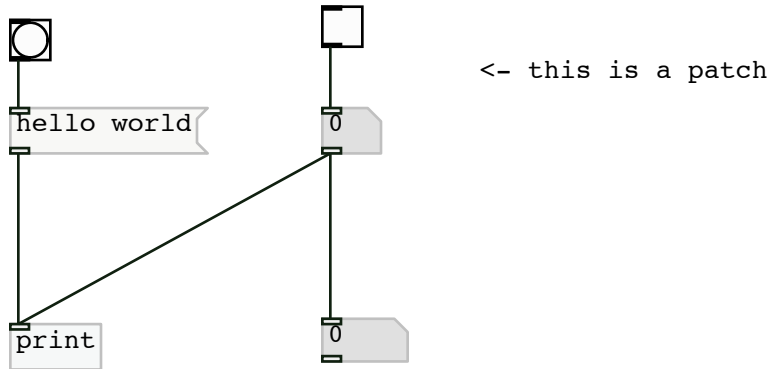
[bng] = 

the toggle object (on / off switch) can be generated with ctrl-1 and the entry "tgl"

[tgl] = 

Made a mistake? Inadvertently created a message box instead of a number box? Just to the side, and so the correct shortcut behind. Then you can delete the wrong object quickly.

Cables



You can connect outputs to inputs, not vice versa.

the object [print] "prints" the messages in the terminal window

A message box may be more than one message, separated by a comma to send one after another

With Apple-E (Ctrl-E on the PC), you can enter the edit mode.

task:

- 1 Try other possibilities of wiring.
- 2 Put a list of [print] objects and give them names.

Where_are_the_objects?

Pure Data is modular. Anyone can write "so-called" externals (libraries) and embed them in PD. That's why there is no PD toolbar.

How do I find objects with the function after I search?

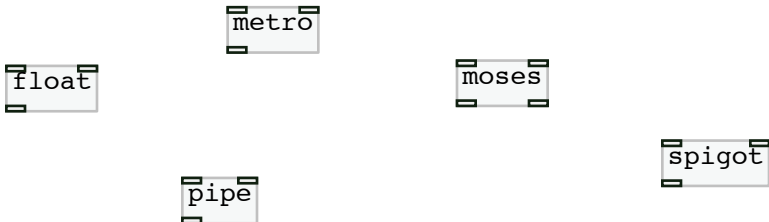
Under "Help" in the menu there is access to the built-in PD objects, tutorials, and help files.

Using the context menu you can find assistance and possible uses for the objects.

Under <http://iem.at/pd/pdb> finds the Pure Data Base, where you can search for keywords and get the appropriate download sites for libraries (externals)

If there are any questions about installation, configuration, etc are <http://www.puredata.org> recommends there is also a collection of the most frequently asked questions (FAQ).

The PD mailing list at <http://lists.puredata.info>



PureData_Compared

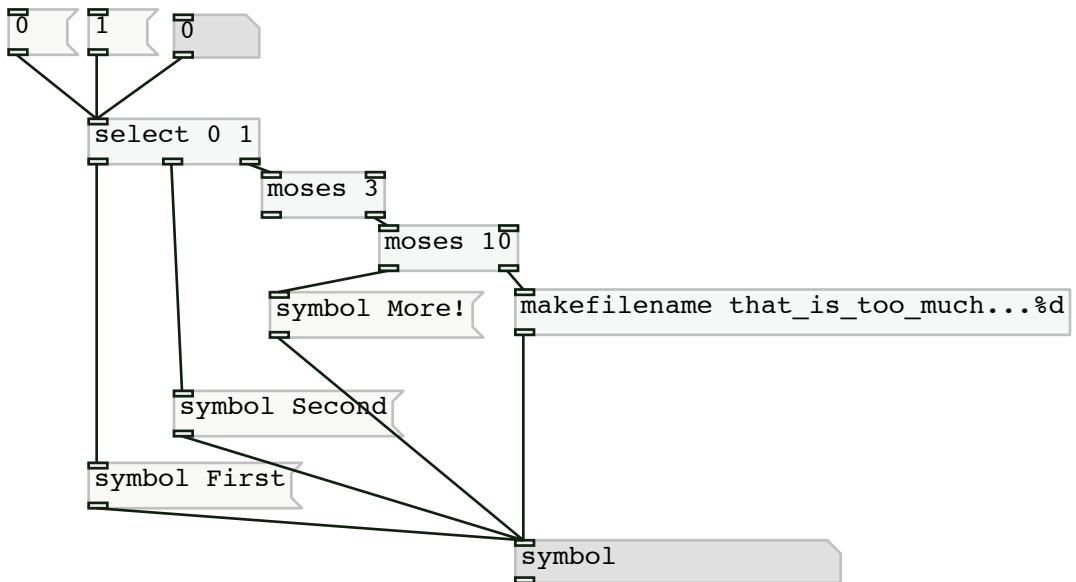
The visual "knitting" of a Pure Data patch is significantly different from other programming languages requiring other ways of thinking.

Here is a small translation course:

In Visual BASIC, a simple program would look like this:

```
IF input = 0 THEN
response.write("First")
ELSEIF input = 1 THEN
response.write("Second")
ELSEIF (input = 2 AND input < 10) THEN
response.write("More!")
ELSE
response.write("that_is_too_much..."& input)
END IF
```

Compared to the same program using Pure Data:



some PureData objects query which conditions:

`==` `!=` `>` `<` `>=` `<=` `sel` `route` `moses` `max` `min` `spigot` `demux`

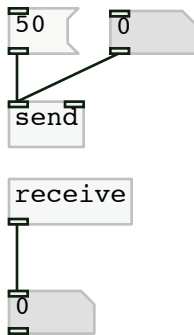
Program loops (loops) in Pure Data:

`until` zexy library: `repeat`

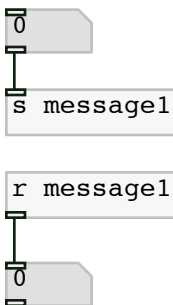
variable: `value` formulas: `exp` `pd example_to_exp`

[pd Advanced_double-click_here](#)

Send-and-Receive



Instead [send] can use the short form [s] are used:



task:

do more transmit and receive pairs

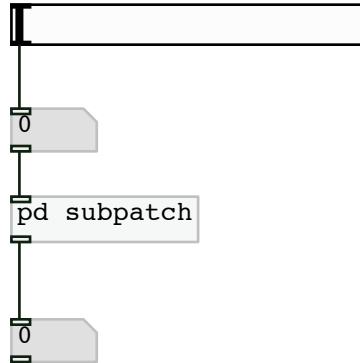
Copy / Paste works naturally in PD also. Simply select the field and press Ctrl-C (OS X: Apple-c) to copy and ctrl-v (OS X: Apple-v) for insert.

Note: Paste the copy of the reply inserts located directly on the original. To avoid this is the command has "Duplicate" (Ctl-d/Apfel-d).

To communicate between programs or computers you can use [OSC] (Open Sound Control), MIDI and [Netsend] / [Netreceive] and [NetClient] / [netserver]. We'll get to later still.

pd advanced

Subpatch



In addition to [send] / [receive] a patch under a patch of another possibility is to make clear about.

A subpatch is created by creating an object [pd ...]

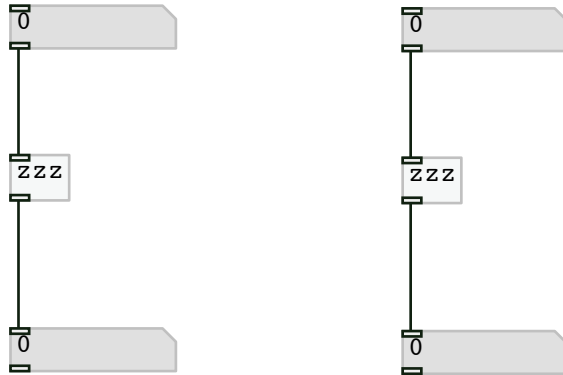
On (double) click context menu or "Open" has opened itself to the lower patch

Via the context menu item "Properties" Change customer can define the objects.

task:

Vera Endere the patch so that you can pay by the slider from 1 to 400 can be entered, and in the terminal window shows the number multiplied by pi.

Abstractions



Once a function is generated once used more than once, you should use abstractions.

Abstractions are Pd patches that are stored as a separate file and can be retrieved from Pd. Pd seeks to first in the current directory and all subdirectories that have been configured as a search path.

Because the search path can be extended over the menu "File-> path."

Here is an abstraction zzz.pd the same path as this patch.

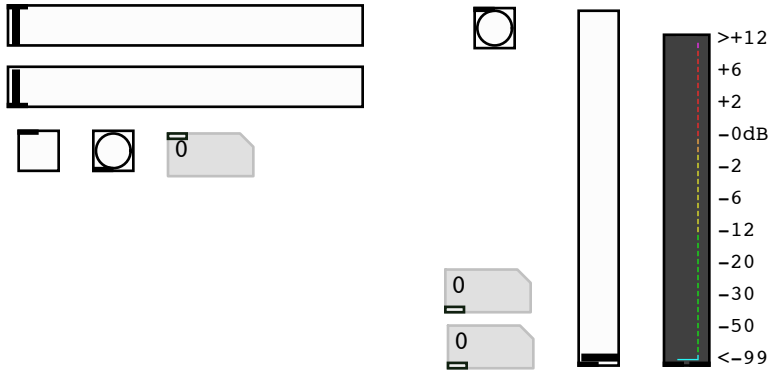
task:

a) Think of a "PDabs" directory for your abstractions to, for example, C :/ PDabs or / home / user / PDabs. Take this directory in your search path.

b) Add an abstraction "grad-to-rad.pd" the angular degrees to radians $0-360 \rightarrow 0-2 * \text{PI}$ (that is converted from 0 to 6.2831).

Tip: Convert (three set!) the only input to a range of 0-1

Object_Send-Receive



Order by clear graphical user interfaces (GUIs) can build with PD to use the wireless transmission options the PD controls.

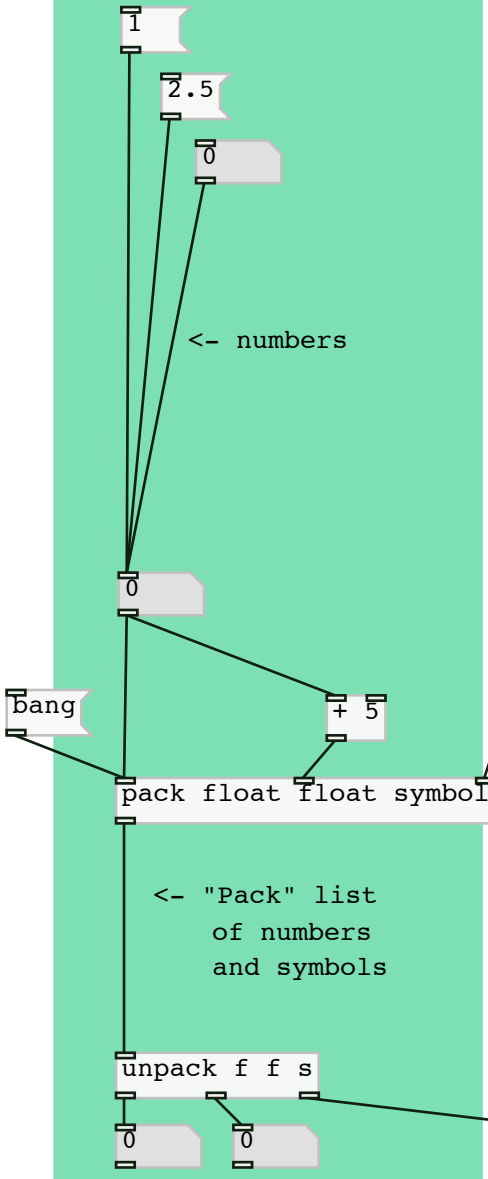
All PD objects that allow data entry can, on its own transmit and receive signals.

About the Object Properties (right-click, properties), you can set transmit and receive value.

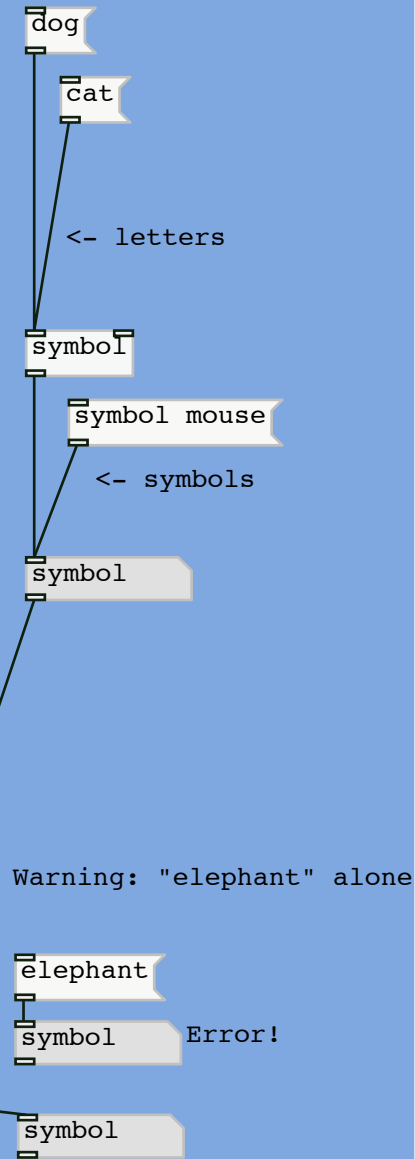
What indicates that an object has set a send or receiving value?

Signal_Types

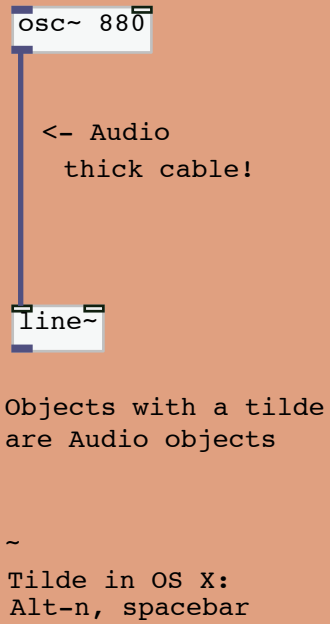
NUMBERS (floats)



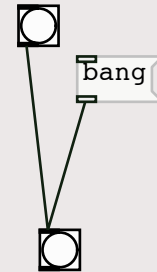
SYMBOLS



AUDIO



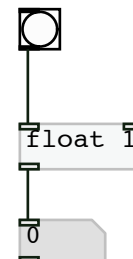
BANG



task:

The Object `float` stores a number. Fill in the patch at the right to build a simple counter.

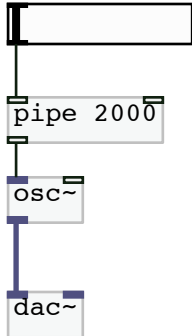
The brother of `[float]` is `[int]`. It stores only an integer, incoming floating point numbers are rounded.



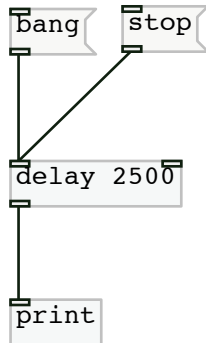
Metronome_Time_Delay

Four objects with which you can control a signal in time:

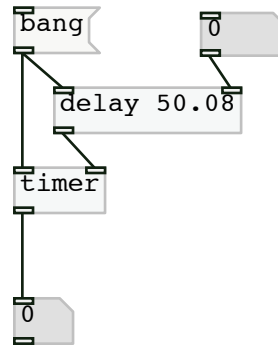
A "pipe":



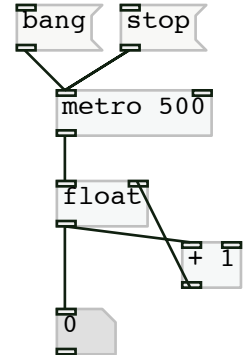
Delay:



Measurement of
Delay:



A metronome:



[pipe] stores an entire sequence of numbers and outputs them after a specified delay time in milliseconds.

[delay] delay a bang.

[timer] measures the distance between two bang.

[metro] specifies the clock.

Pd does not distinguish between integers and floating point numbers. Wherever integers can be entered are also decimal points possible. Note: in the Pd Dezimalstellentrennzeichen is a point.

By holding down the Shift key (Shift) can Change customer numbers in boxes decimals

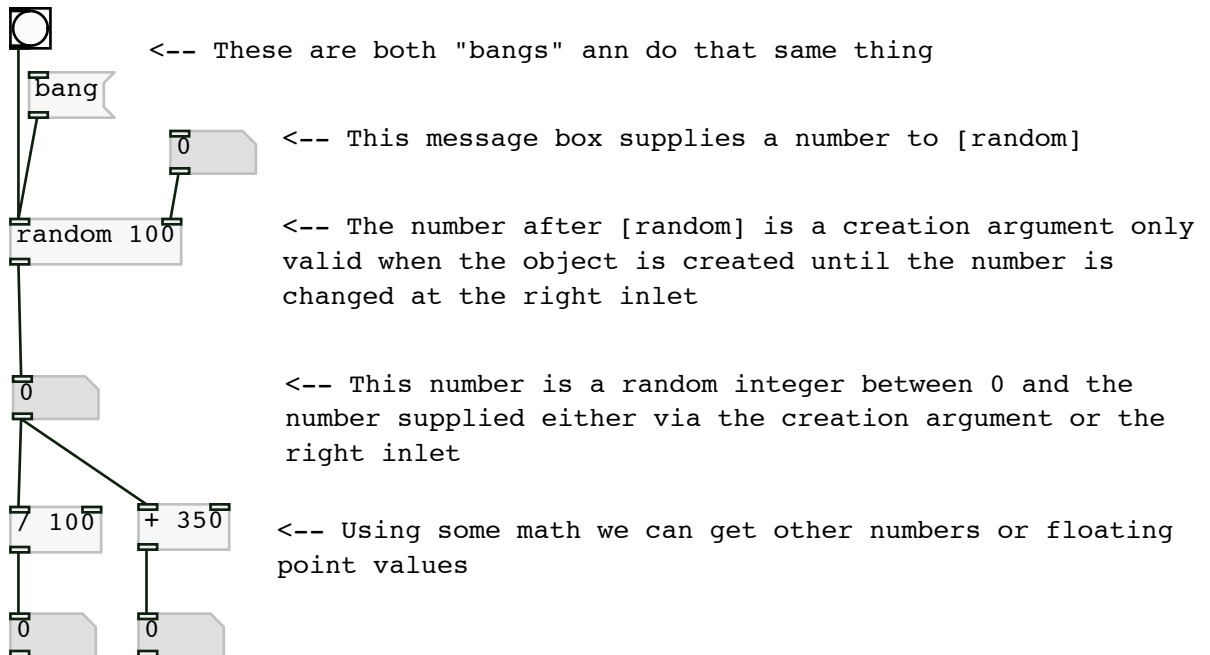
task:

Build a patch that plays a scale.

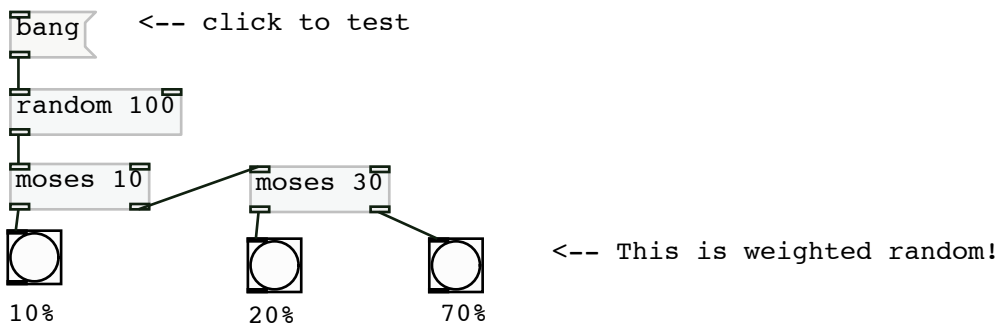
Advanced: to use the object `[mtof]` .

Randomization

the [random] object generates pseudorandom integers. You can use a creation argument such as in the example below or define a number using a numberbox on the right inlet. A bang to the left inlet will generate a random number from 0 to that number.



We can also use [random] with an object like [moses]. [moses] takes a number and outputs them at left if they're less than a control value, and at right if they're great or equal to it.



Sound_Generation

```
osc~ 880
```

```
noise~
```

```
dac~
```

Audio objects end with the tilde ~. On the Apple Keyboard with the tilde Alt-n, then the space bar can be achieved.

The audio output object [dac~] (digital to audio converter) has two standard Corridors (stereo).

[osc~] generates a cosine wave

task:

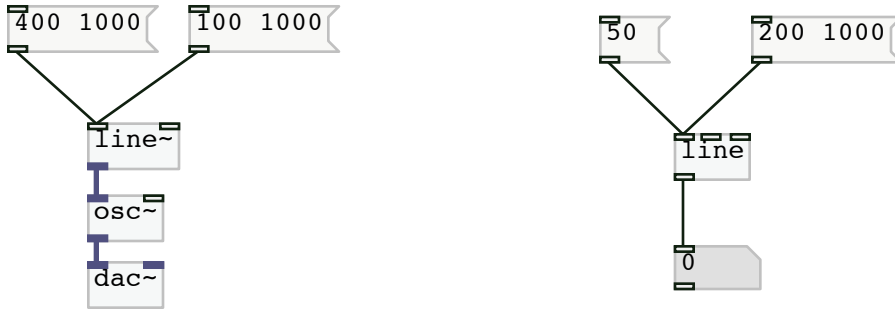
connect [osc~] with [dac~]

Vera Endere the frequency of the sine wave with a slider from 200 to 600 Hz

See you at the help-patch for object [dac~].

Audio_Analogies

There are some objects in two ways: as an audio object and a "normal" object. Here's an example:



[line ~] creates an audio ramp which is with [osc ~] to a frequency modulated (in Hz) and is sent to the audio output [dac ~].

[line] generates a sequence of numbers.

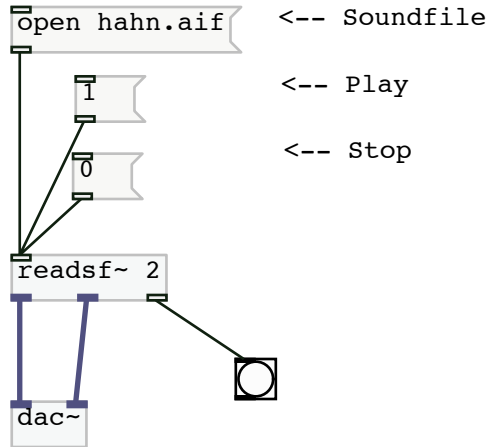
task:

Using a comma to separate multiple values, it is also in a message send sequence.

Let rise a tone of 100 to 250 Hz in 1, 5 seconds.

Use the object [sig ~] to produce a numerical value-ramp as an audio output with [line ~].

Play_a_Soundfile



[readsf ~] is an object to play sound files.

If the sound file is over there from his third outlet from a [bang].

task:

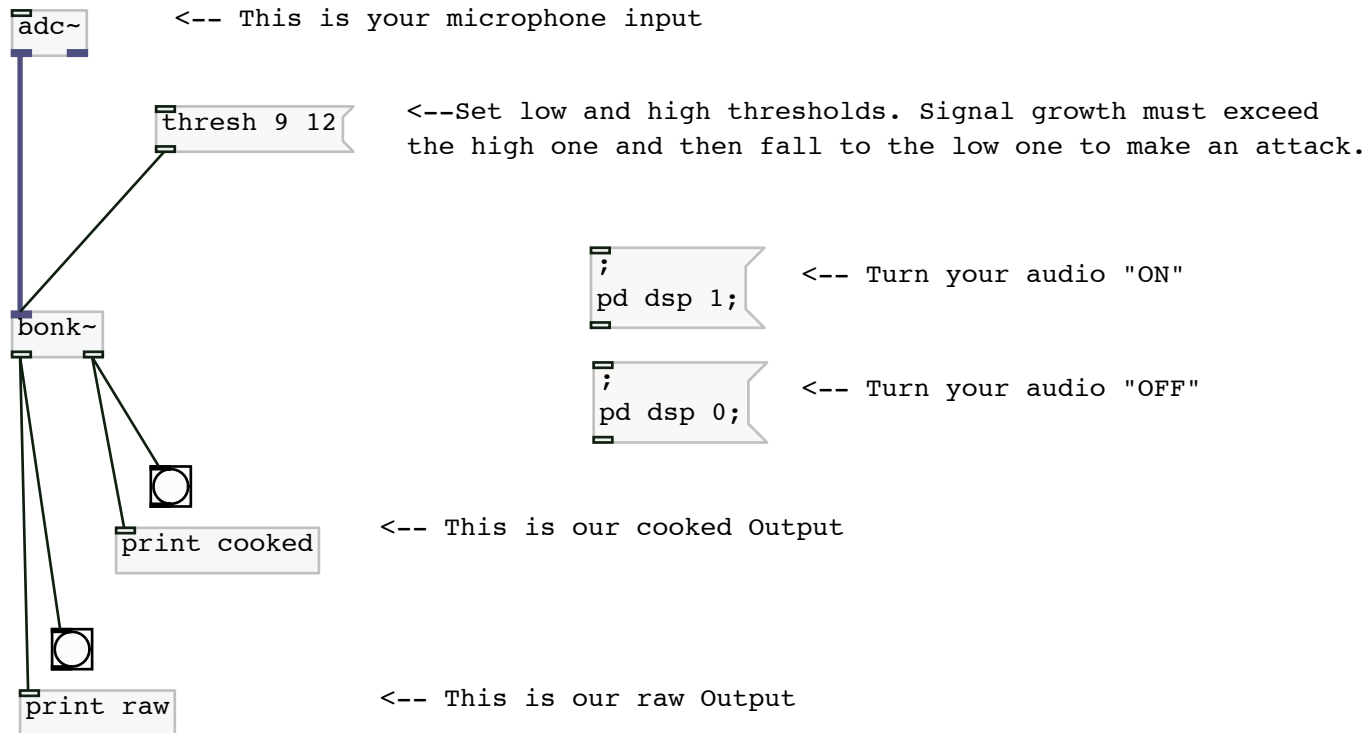
Let the cock crow without a break!

How do I open a dialog box, as I can select the sample?

See: `pd open_from_dialog`

Bonk~

The Bonk~ object takes an audio signal input and looks for "attacks" defined as sharp changes in the spectral envelope of the incoming sound.



There is plenty of information to be determined from the output. Please right click the object to see the help file and more information. there is also a lot more options to callibrate bonk~ and custom tailor to your application. **WARNING:** You might have lots of tweeking to do depeding on your environment!

task:

Build a patch that plays a note from a handclap.

Fiddle~

The Fiddle~ object estimates the pitch and amplitude of an incoming sound, both continuously and as a stream of discrete "note" events. [fiddle~] is pitch detection. It will turn an incoming audio signal into a stream of numbers.

