SOUND DESIGN

Alessandro Fiordelmondo
alessandro.fiordelmondo@labatrentino.it

3P
INTRODUZIONE A PURE DATA
PD

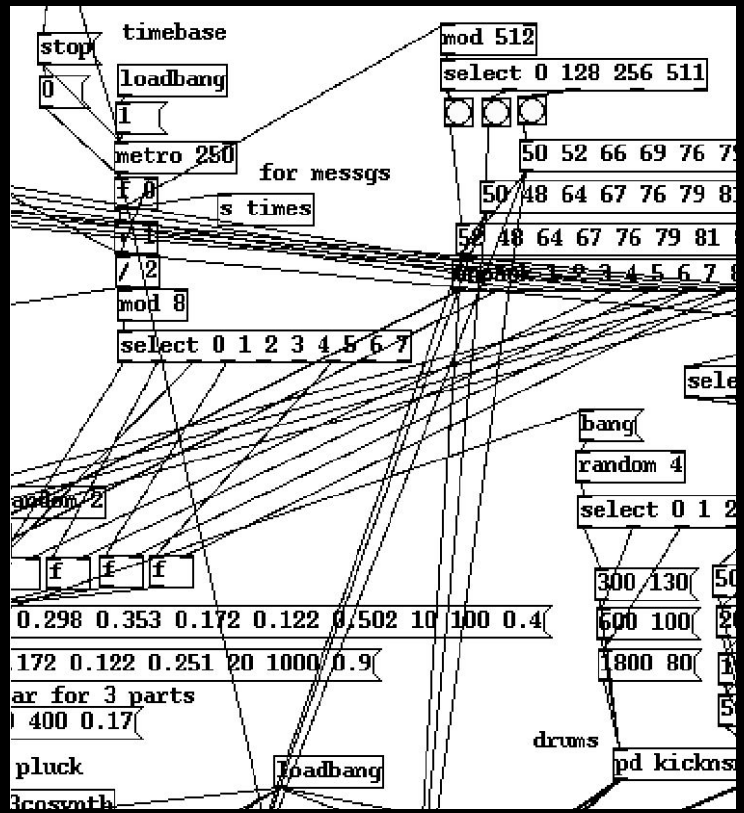
LABA Libera Accademia Belle Arti - Trentino
2021/2022

PURE DATA

Real-time graphical programming environment for audio and graphical processing

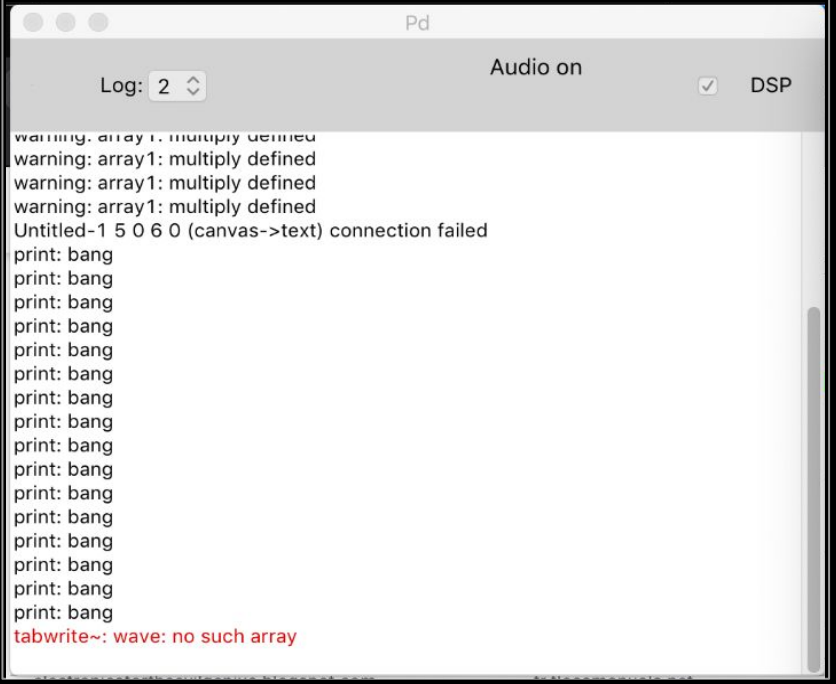
Open Source

Cross-platform software (personal computer, smartphone, embedded platform)



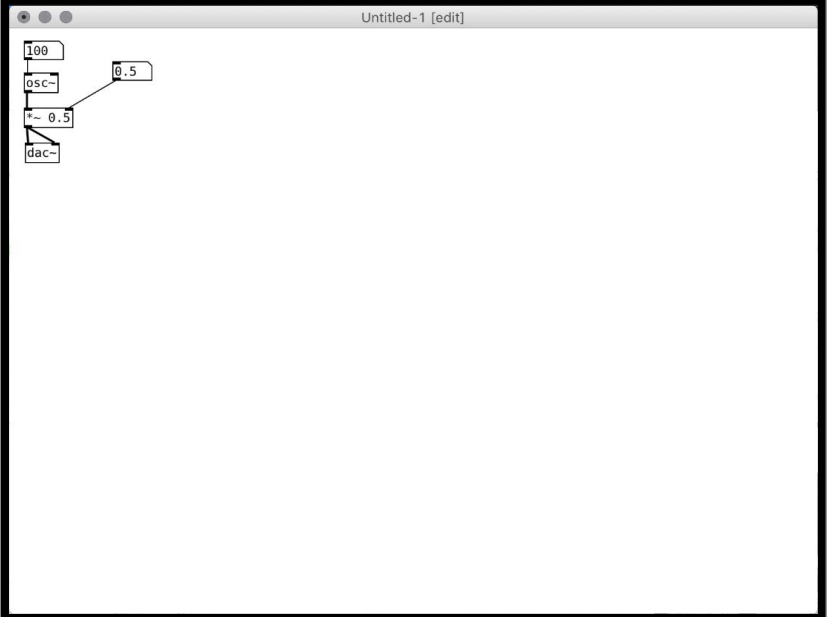
PURE DATA ENVIRONMENT

CONSOLE



errors, messages, data, etc

PATCH



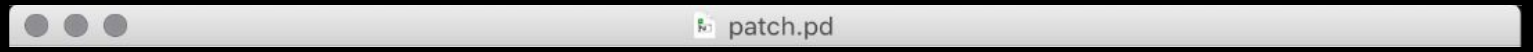
Development Environment

PATCH - EDIT MODE

The patch can run in:

- **Edit mode** users can develop the patch by adding and link boxed

- **Lock mode** users can interact with the boxed in the patch



To switch from one mode to another go in *edit* on the menu bar and click *Edit Mode*.

By shortcut: **cmd+e** (macOS) **ctrl+e** (windows / linux)

HELLO WORLD

Add *boxes* into the patch while the patch is in edit mode.

Go to *put* on the menu bar; the opened window shows the *boxes* you can add to the patch. By clicking on one of them, the *box* is automatically added.

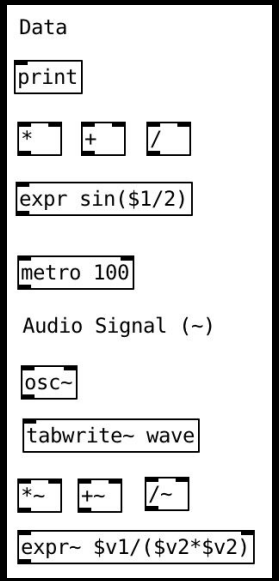
To add a specific type of box you can also use the shortcut provided in the same window, after the name of the object.

For the **object** boxes, you have to digit the class of the specific object (‘print’ as shown image)

To print the message “Hello World”, you have to switch the patch to lock mode and click the message box.

TYPES OF BOX

object

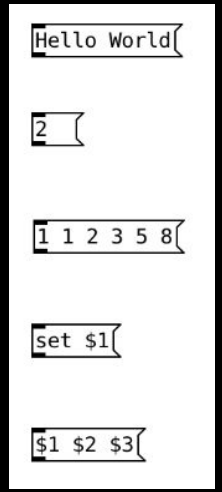


When you put the object box into the patch, it appears with a dashed border. By typing a specific text into the box a determined object is created.

Each object is referred to as a class with certain inlets, outlets, methods, arguments and attributes.

The objects can be of **data type** - for numbers and symbols - or **signal type** - for an audio signal.

message

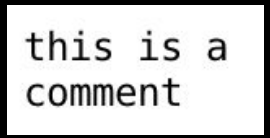


Message boxes interpret the text as a message to send whenever the box is activated (by an incoming message or with the mouse). They handle numbers, symbols and numbers/symbols lists.
The $ is a message variable that is substituted with an input value

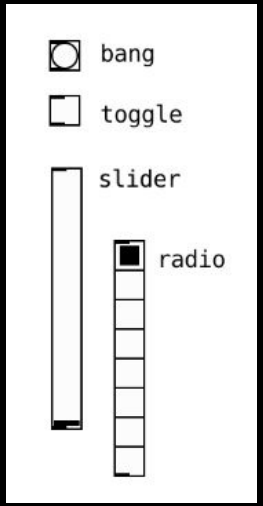
number box

Number boxes allow you to display numbers or to enter numbers using the mouse and keyboard. When a number arrives at the number box's inlet, it is displayed and sent to the outlet

comment



GUI



GUI boxes come in many forms including bangs, toggles, sliders, and so on

Bang sends a "bang" message with which many processes are activated.

Toggle alternatively sends 1 or 0 (on/off, true/false)

Slider works as a mixer's fader with numeric values range from 0 to 127.

with **Radio** you can select a list of integer value

- **Data**:
  - **numeric**: 32-bit floating-point
  - **symbol**: non-numeric value (e.g. a string)

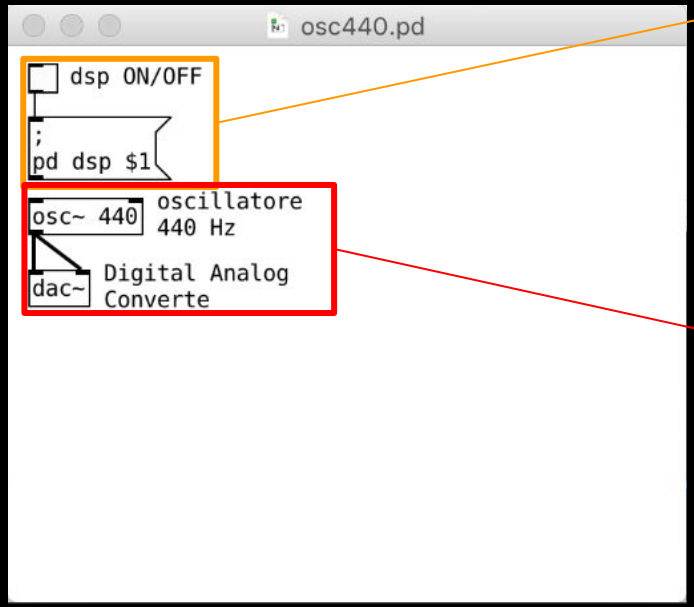
  Data can be grouped in number and/or symbol **lists**. The term **array** specifically refers to a *numeric array* often used with wavetable oscillator and oscilloscope.

- **Audio Signal:**

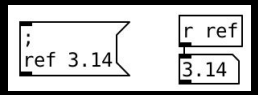
  Pd's audio signals are internally kept as 32-bit floating-point numbers, so you have all the dynamic range you could want. The sample-rate (44.1kHz, 48kHz, ...) and bit-depth (16, 24, .. bits) depend on you sound card.

  Audio computations in Pd are carried out by "tilde objects" such as "osc~" whose names conventionally end in a tilde character to warn you what they are.

OSCILLATOR 440 HZ



enable/disable DSP
the use of semicolons allows sending messages to a specific object without the link. The first symbol following the semicolon is the "destination reference". What follows is the message. Usually, we use the object [r] (receiver object) to catch the message with the specific "destination reference".



In osc440.pd patch case we send the message "**dsp $1**" to a hidden object **pd** (an internal object of the software). The variable **$1** takes the input of the message that is the output of the **toggle**, therefore we send alternatively "**dsp 0**"/"**dsp 1**" which disable/disable the PD's dsp.

the **osc~** object is a digital oscillator and generates a cosinusoidal wave (a sinusoid with pi/2 phase, 90°)

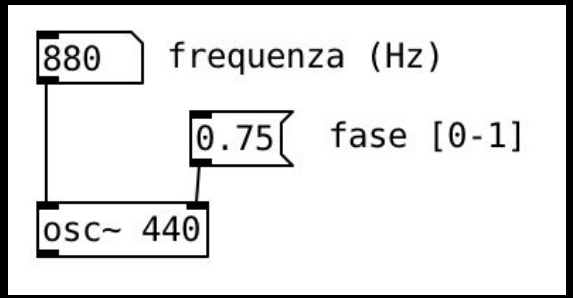
the **dac~** (Digital Analog Converter) converts the digital signal into analog signal that we can listen to through our speakers or headphone. Practically, it's our audio output.

DSP (*Digital Signal Processing*). By enabling the DSP we turn on the audio computation (it allows listen to, record and process an audio signal). The Digital Signal is a sequence of number that represent an audio sample.

The DSP in the hardware is the sound card's microprocessor optimised for the digital signal processing.

In PD we can set our sound card, the sample-rate and the block-size in the **Audio Settings** subwindow. In the main menu go to *Media* and click on *Audio Settings...*

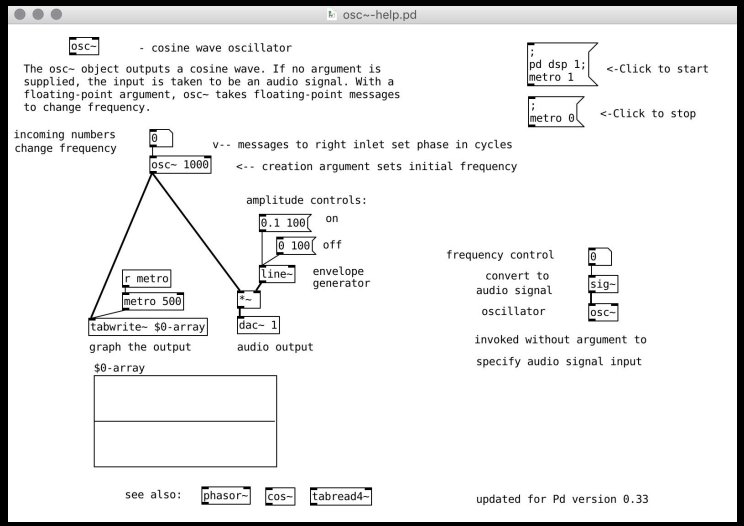
INLETS, OUTLETS AND ARGUMENTS



Each object has a different number of inlets (dashes on the top border) and outlets (dashed on the bottom border) with different functionality.

In the osc~ object with the left inlet we can set the frequency in Hz, with the right inlet we can set the phase in cycles (from 0 to 1, where 1 is equal to 2pi or 360°)

The arguments are defined after the object class name, and they define the object's initialization values



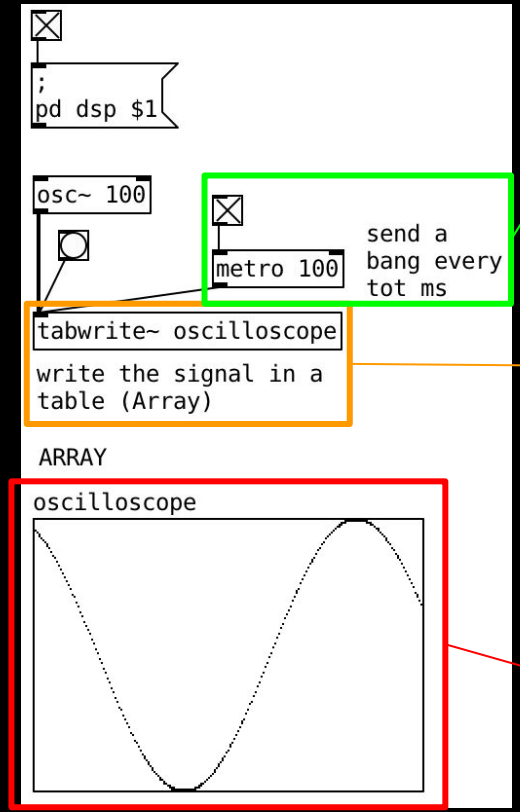
osc~ help

in the osc~ 440 specify the initial frequency

For more info abobject (about inlets, outlets, arguments, etc) refer to the pd's Help

HELP (right click on the object and click "help" on the pop-up menu)

AUDIO SIGNAL VISUALIZATION



the **metro** object send a bang message every tot milliseconds (specified by the argument or set through the right inlet).

In this case we can update our oscilloscope every 100 milliseconds

the **tabwrite~** object convert the signal in a sequence of number (array) every time we send a bang. In the argument we have to specify the array's name to fill. tabwrite~ convert a number of samples equal to the size of the determined array.

Numeric Array in a table.

To see our signal we have to store it as a sequence of number (a numeric array).

When we create an array (from the *put* menu) we have to chose the name (e.g. "oscilloscope") and the size (e.g. 512).