

IMCA 221
Programming for Artists
Winter 2026

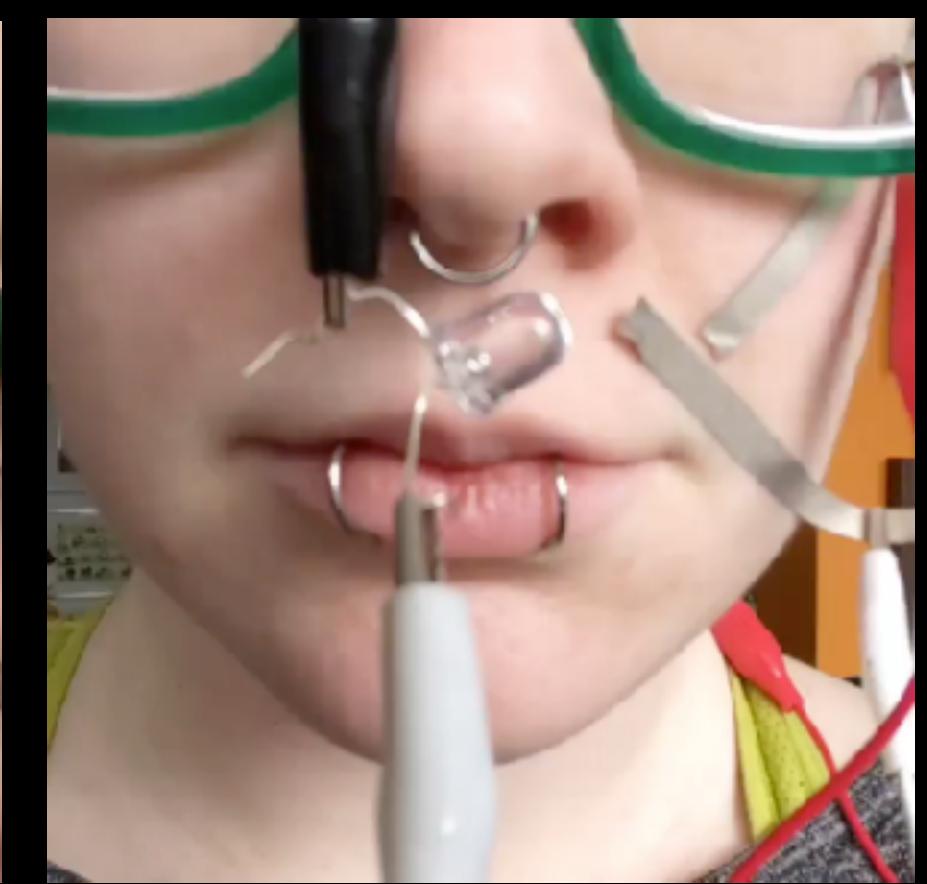
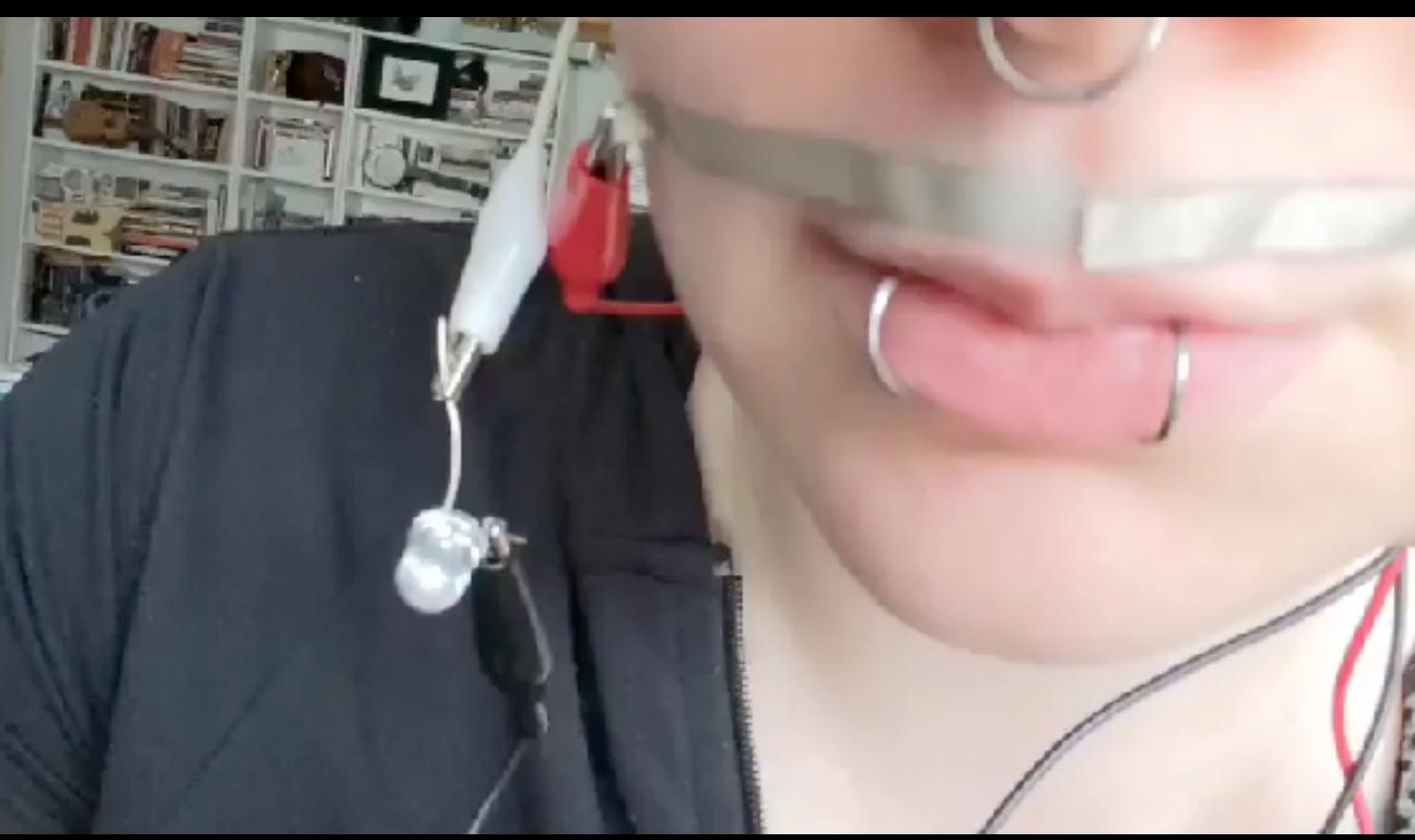
dr lee wilkins
l.wilkins@concordia.ca

**Class is on Zoom even in the classroom
for sharing, find the details on Moodle**

<https://moodle.concordia.ca/>

Download the slides!

Introduce Yourself in this slide









YES

ARCANA

NO

A B C D E F G H I J K L M N O
P Q R S T U V W X Y Z

X
1 2 3 4 5 6 7 8 9 0

.....
GOOD BYE





POWER

ALL
ABOUT
WIRES

LED
LOVE

LEE WILKINS

Open Source Hardware Association





Get Help!

Technician: James Schidlowsky

Office: EV-S2-618

E-mail:

james.schidlowsky@concordia.ca

HYBRID LAB SCHEDULE

WINTER 2026 SEMESTER

TIME	MON	TUE	WED	THU	FRI
8:30					
9:00					
9:30					
10:00					
10:30	OPEN LAB* 10:30-13:00	OPEN LAB* 10:30-16:30	OPEN LAB* 10:30-13:00	IMCA 221 08:30-12:30	
11:00					
11:30					
12:00					
12:30					
13:00					
13:30					
14:00					
14:30					
15:00					
15:30	IMCA 222 13:30-17:30	IMCA 322 13:30-17:30	IMCA 322 13:30-17:30	OPEN LAB* 13:00-16:30	SCUL 251 13:30-17:30 (OCCASIONALLY)
16:00					
16:30					
17:00					
17:30					

* OPEN LAB :

- Please see the IMCA technician (James) in EV-S2.618 for access, tools and help.
- Computers require a CDA account. Installed software includes Max/MSP, Arduino IDE, Adobe suite, Processing, Python, etc. Please log out before leaving.
- Soldering station, fume extractor, bench power supply: use with caution! Please turn off after use.

**Reserve private studios / critique spaces
via the Patron Portal**

<https://www.concordia.ca/finearts/cda/equipment/tutorial.html#gsc.tab=0>

Accommodations

Center for Students with Disabilities: [https://www.concordia.ca/students/
accessibility.html](https://www.concordia.ca/students/accessibility.html)

You need to tell me if you need an extension!

Student Success Center: <https://www.concordia.ca/students/success.html>

Mental Health Services: [https://www.concordia.ca/health/mental-health/
services.html#appointments](https://www.concordia.ca/health/mental-health/services.html#appointments)

Health Services: <https://www.concordia.ca/health/medical/clinic.html>

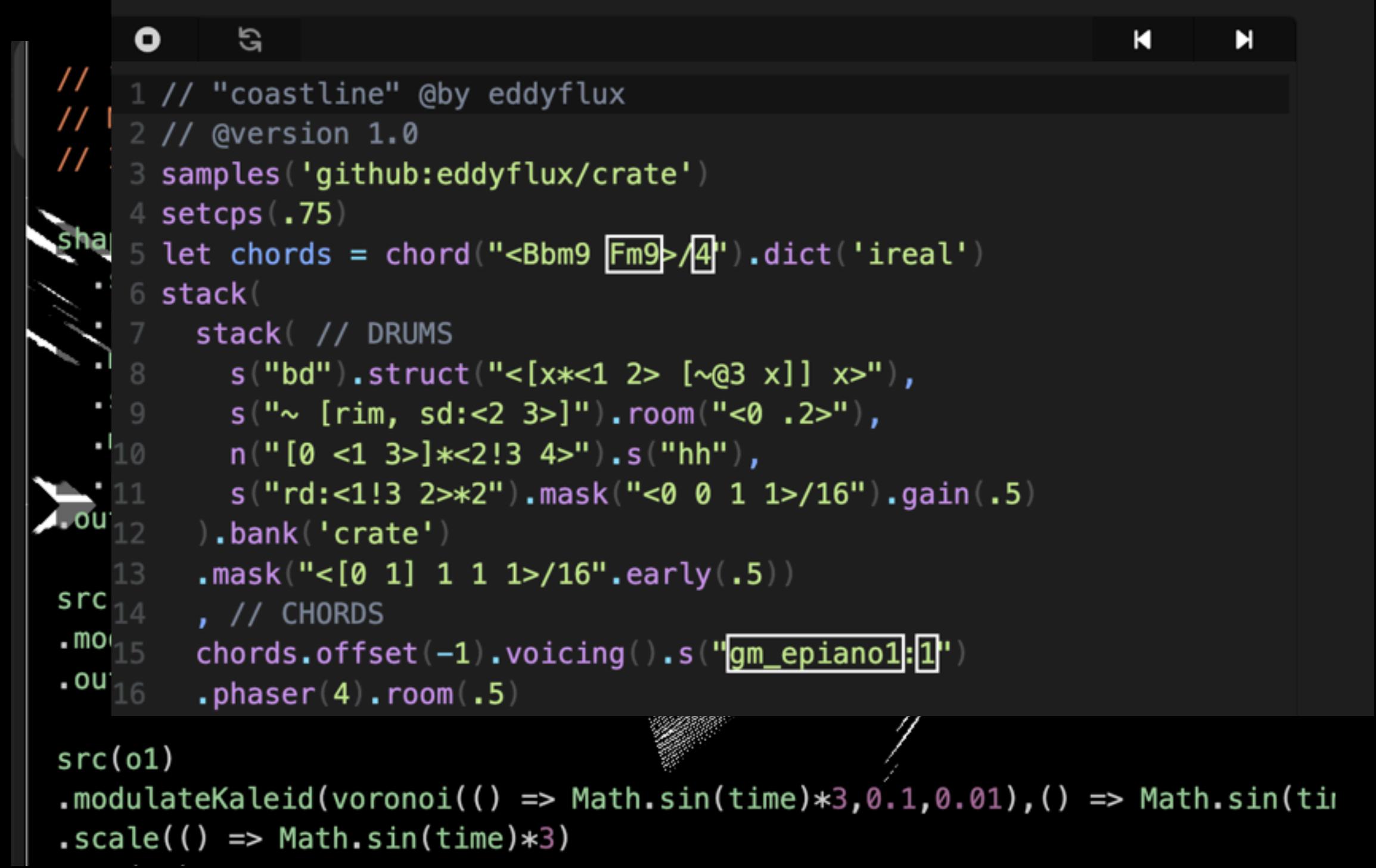
Learning Objectives

- Learn the basics of interactive installation using MaxMSP
- Learn the basics of Arduino based interactive installation
- Explore the possibilities of interactive installation in a variety of contexts
- Learn how to constructively critique work
- Learn to document your work

What is programming for artists?

- Controlling audio, video or physical objects using your computer
- Automatically sensing the world and outputting it in different ways
- Experimenting with computation and code to enhance your existing practice
- Remember, you can make it your own!

Strudel



A screenshot of a code editor window displaying Strudel code. The code is a musical composition script. It starts with comments indicating it's a coastline piece by eddyflux and version 1.0. It uses samples from a GitHub repository and sets the CPS value to 0.75. The main structure is a stack of multiple audio processing chains. One chain includes a drum pattern (bd), rim, and hh, followed by a masked and gain-controlled signal. Another chain involves chords offset by -1, voicing, and a piano sample (gm_epiano1). The final output is a phaser effect with a room setting of 0.5. The code is annotated with line numbers from 1 to 16.

```
// "coastline" @by eddyflux
// @version 1.0
samples('github:eddyflux/crate')
setcps(.75)
let chords = chord("<Bbm9 Fm9>/4").dict('ireal')
stack(
  stack( // DRUMS
    s("bd").struct("<[x*<1 2> [∼@3 x]] x>"),
    s("~ [rim, sd:<2 3>]").room("<0 .2>"),
    n("[0 <1 3>]*<2!3 4>").s("hh"),
    s("rd:<1!3 2>*2").mask("<0 0 1 1>/16").gain(.5)
  ).bank('crate')
  .mask("<[0 1] 1 1 1>/16".early(.5))
, // CHORDS
  chords.offset(-1).voicing().s("gm_epiano1:1")
.ou
  .phaser(4).room(.5)

src(o1)
.modulateKaleid(voronoi(() => Math.sin(time)*3,0.1,0.01),() => Math.sin(ti
.scale(() => Math.sin(time)*3)
```

[https://
strudel.cc/
workshop/
getting-started/](https://strudel.cc/workshop/getting-started/)

```
// licensed with CC BY-NC-SA 4.0 https://creativecommons.org/licenses/by-nc-sa/4.0/
// Mahalia H-R
// IG: @mm_hr_

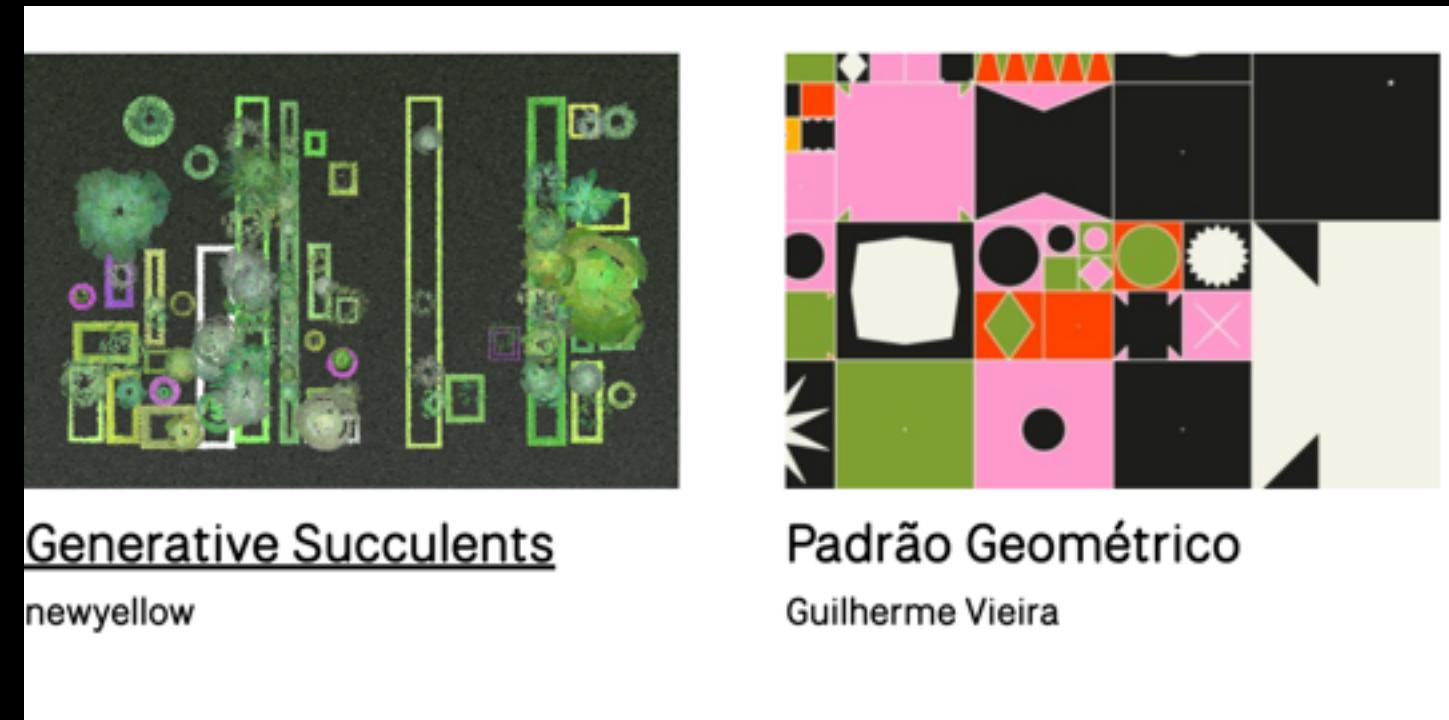
shape(20,0.1,0.01)
  .scale(() => Math.sin(time)*3)
  .repeat(() => Math.sin(time)*10)
  .modulateRotate(o0)
  .scale(() => Math.sin(time)*2)
  .modulate(noise(2,0))
  .rotate(0.1, 0.9)
  >out(o0)>

src(o0)
  .modulate(osc(500,0,0))
  .out(o1)

src(o1)
  .modulateKaleid(voronoi(() => Math.sin(time)*3,0.1,0.01),() => Math.sin(time)*10)
  .scale(() => Math.sin(time)*3)
```

Hydra

[https://
hydra.ojack.xyz/](https://hydra.ojack.xyz/)



processing.org and P5.js

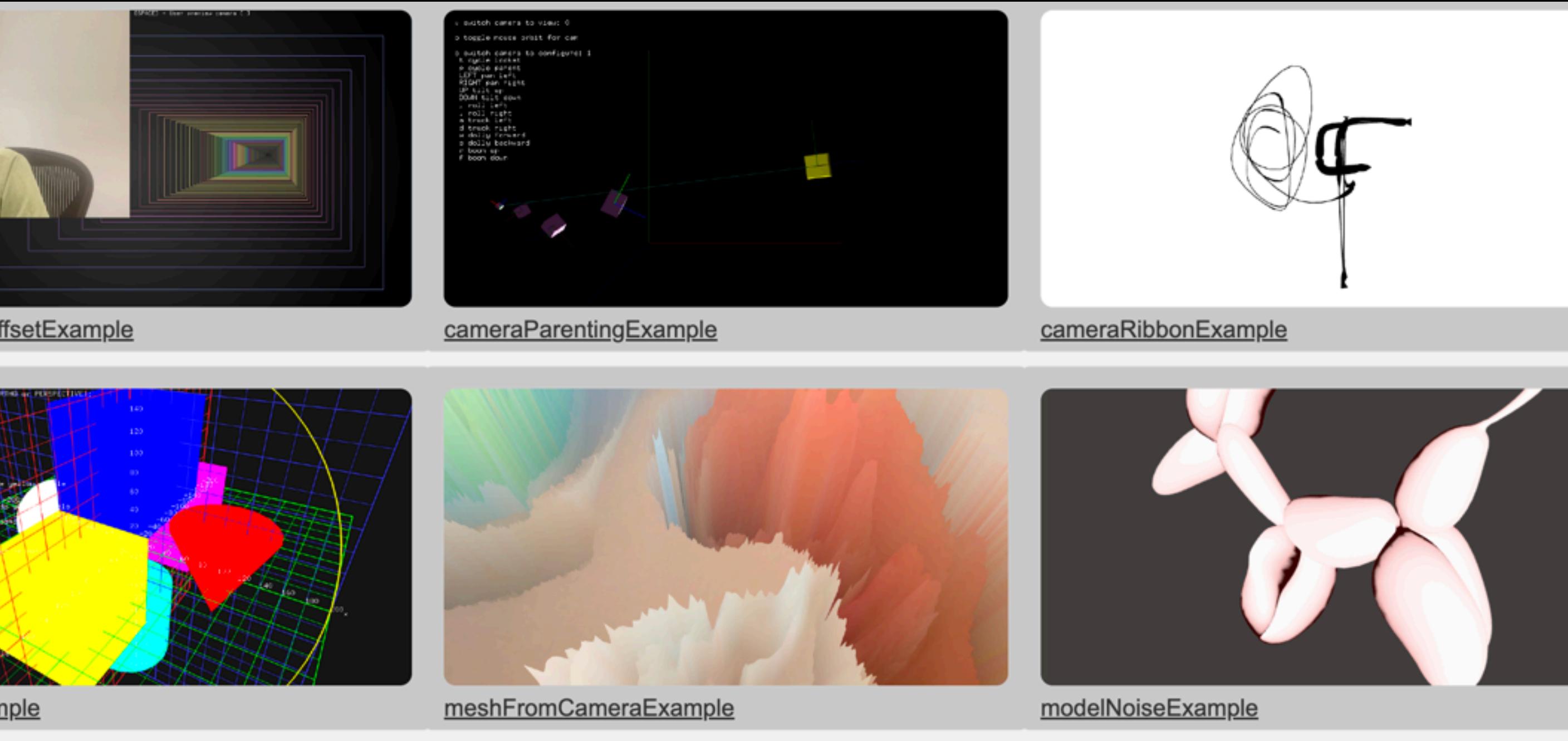
Open Processing: <https://openprocessing.org/>

<https://editor.p5js.org/>

<https://p5js.org/>

```
1 let weavingDraft;
2 const w = 700;
3 const h = 700;
4
5 const nbThreads = 64;
6 const nbOfShafts = 4;
7
8 const nbTieups = 4;
9 const cellSize = w / (nbThreads + nbTieups + 3); // Because 3 gu
10 const padding = cellSize;
11
12 const gps = [61.235148, 21.481852]; // Onkalo
13
14 const lengthDraw = Math.round((h - ((nbTieups + 3) * cellSize))
15
16 function setup() {
17   const canvas = createCanvas(w, h);
18   canvas.parent('canvas-container');
```

Open Frameworks

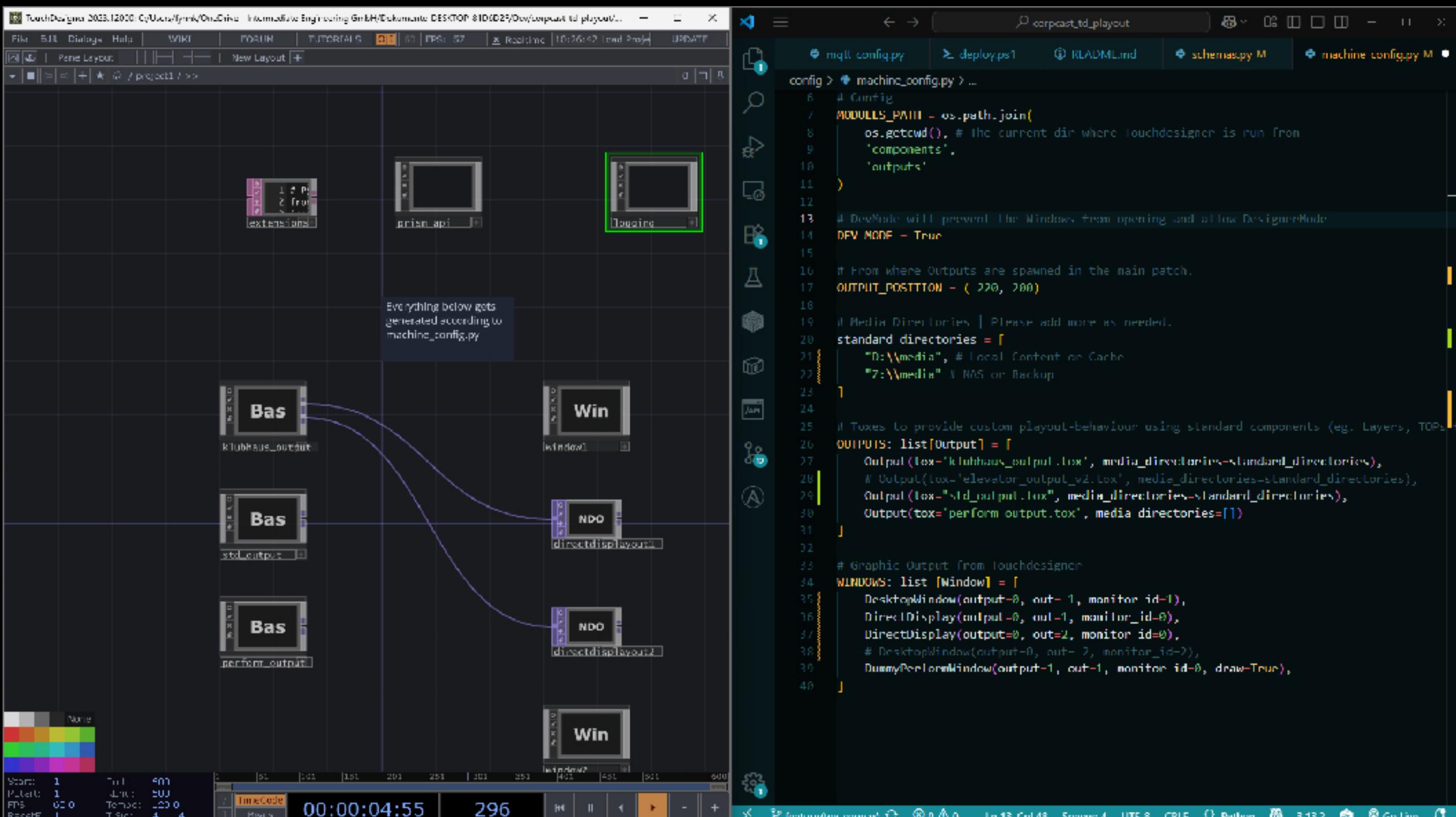


[https://
openframeworks.cc/](https://openframeworks.cc/)

[https://
openframeworks.cc/
examples/](https://openframeworks.cc/
examples/)

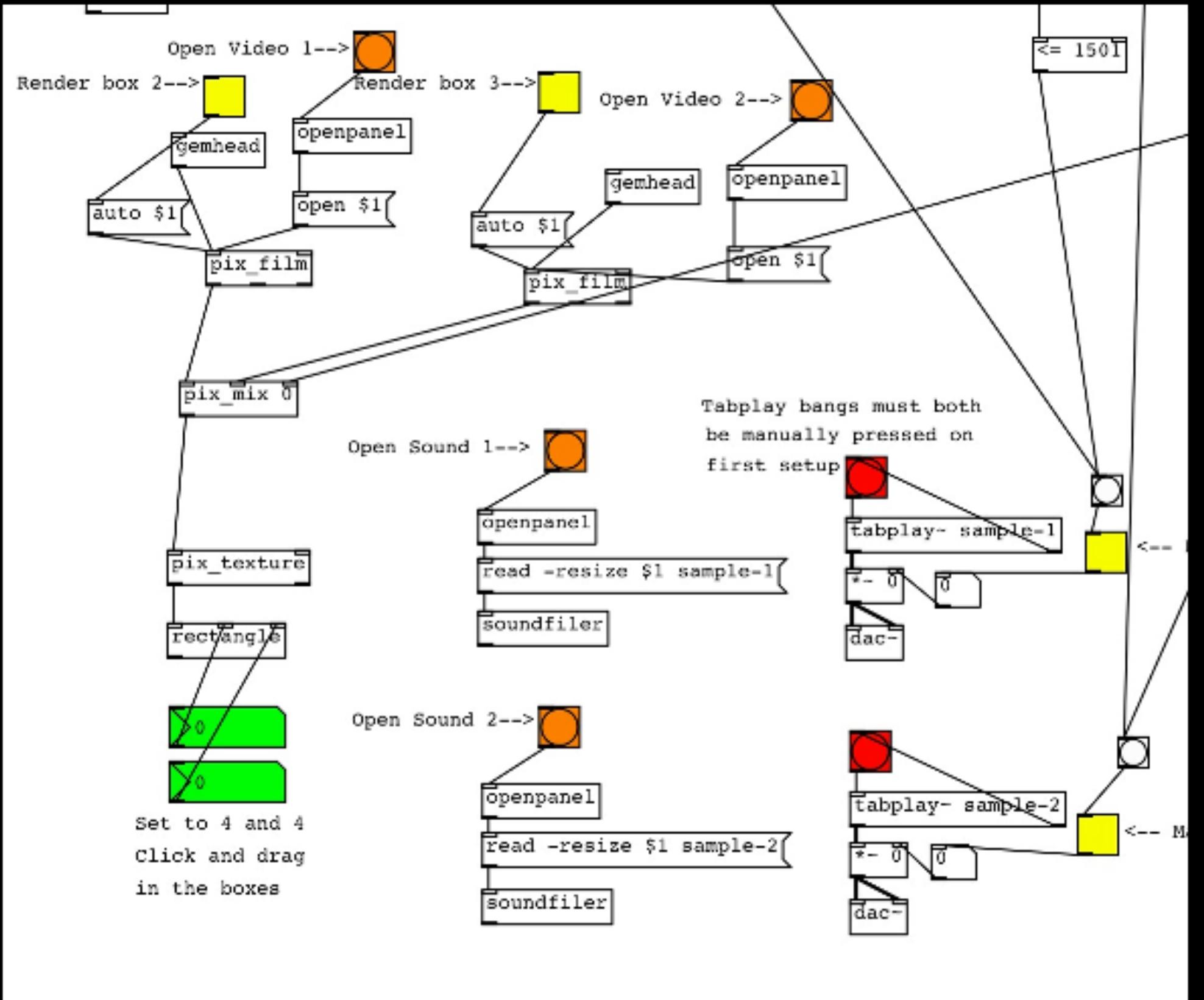
Touch Designer

[https://
derivative.ca/](https://derivative.ca/)



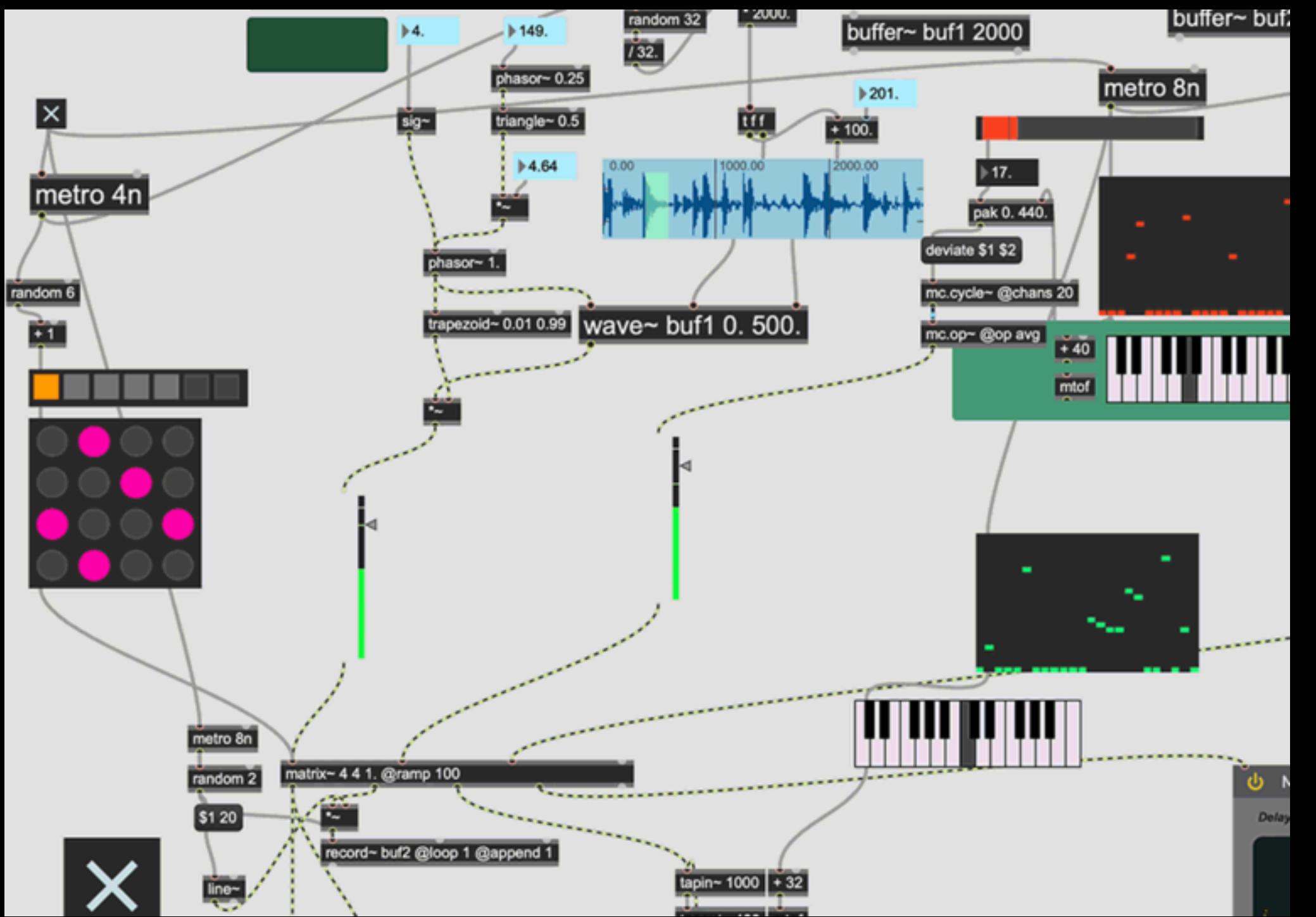
Pure Data

[https://
puredata.info/](https://puredata.info/)

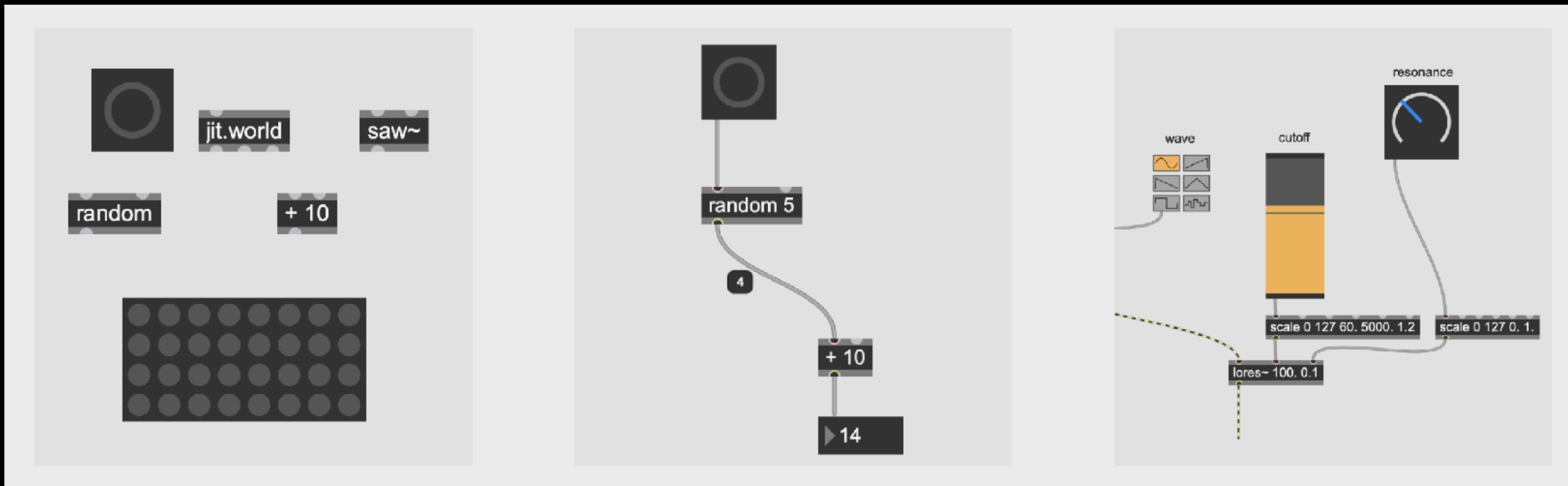




Max 8

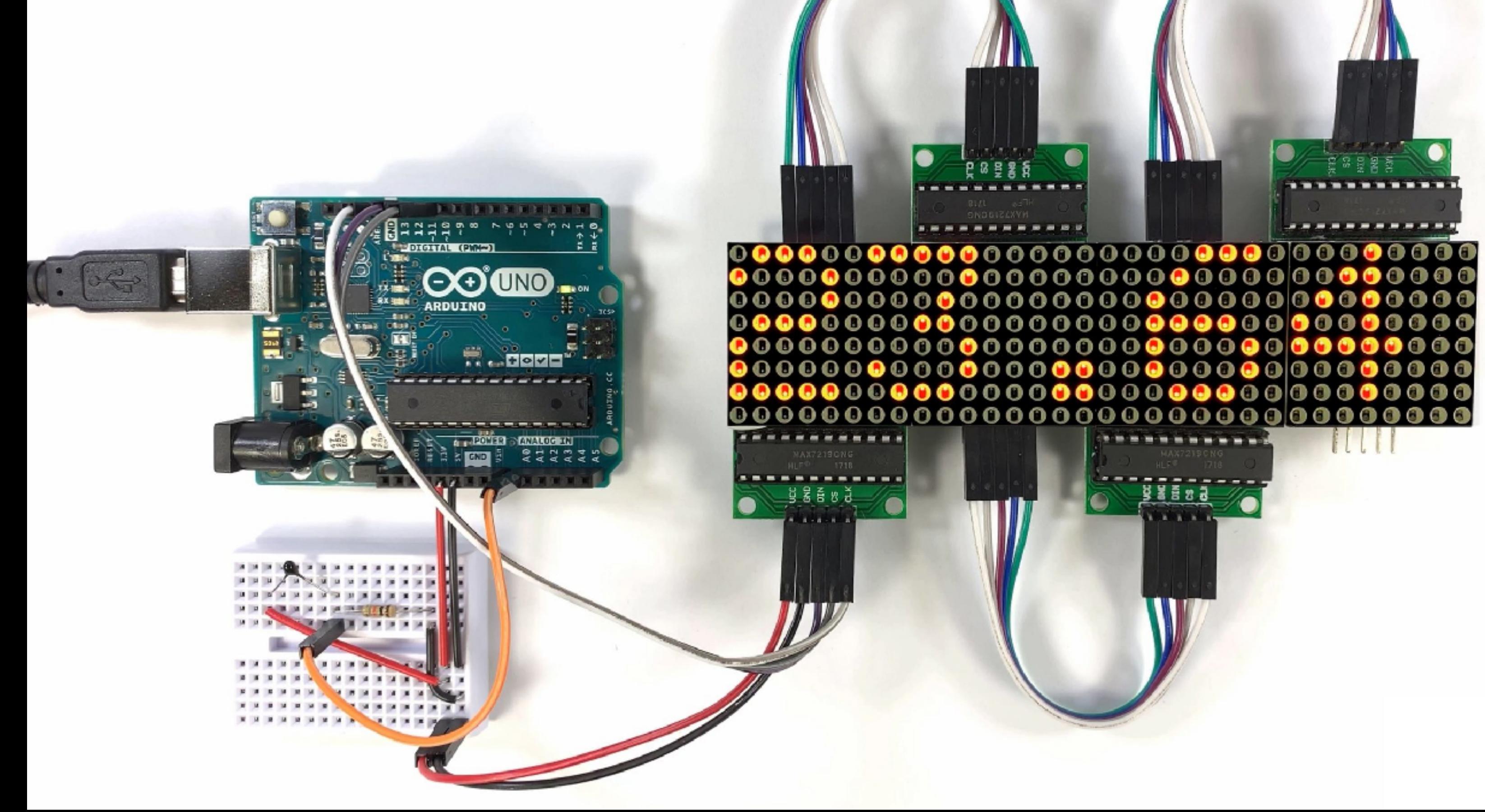


Max 8



Arduino

A physical platform for prototyping
circuits and sensors.



This is a 200 level class. You are not expected to have programmed before, and you are not being graded on how complex your code is.

This course exists to help expand how you think about interactivity and give you the tools to expand your own work. Take your time, explore, and you will do great!

Grade Outline

Main Projects

Mid-term: TRANSFORM (groups) 25% February 26

Final project (individual) 35% April 9

 Prototype 10% March 26

 Final 25%

In class work:

Audio Experiment 10% January 29

Video Experiment 10% February 12

Sensor Experiment 10% March 19

Participation 10%

Grade Breakdown

25% Exploration & Functionality:

The goal of this grade is to help you push your boundaries but work within your capabilities. Having a more complex project is part of the grade, but it is more important that your work functions. Be intentional and spend time trying to understand your problems instead of abandoning things that don't work.

A good way to excel in this area is to explore a new technique we did not touch on in class OR expand meaningfully on something we touched on in class.

25% Creativity & Concept:

Every project should have a concept. Concepts don't have to be elaborate or in depth, they can be as simple as evoking a feeling, exploring a moment, explaining a subject or cause, sharing something you like, making the viewer feel something briefly. Keep your concepts simple as you start. This grade is measured by how effective your concept is on the viewer so don't try anything complex when you start. Some really simple ideas: colours, nostalgia for a particular time, surprise, de-contextualize a space, etc. As you get better at concepts, you can explore larger ideas. Always have a concept, no matter how simple.

A good way to excel in this area is to have a clear concept that is evident to the viewers without explanation.

Grade Breakdown

25% Execution & presentation quality:

This is measured by how well presented your work is. This is aside from the creativity, but how well it is physically or digitally assembled. How would it look in a gallery?

A good way to excel in this area is to make sure your work is polished and clean. This can vary widely depending on the work. Clean up your cables, good presentation, well made objects or assets etc. Be mindful of how the work is shown and viewed. Its okay if you can't make it perfect, but have an idea of how you'd like to finish the work.

25% Documentation:

Project is documented as outlined below. Remember: if you don't document your project, nobody else will know it happened!

Part of your documentation grade is how you present your work in class during critique. Come ready to discuss your work and tell us about your process and purpose.

Handing in work

All exercise and projects should be handed in using the project documentation template on Moodle / GitHub.

Download it and fill in a title, image, screen shot, project description and link to a video. Upload a .pdf.

Remember: Every time you see someone else's work, you're looking at their documentation! Learning good documentation is important!

Class Structure

Outside of class: Practice & explore

Part 1: Live coding + Concepts

Part 2: Studio time and exercises

**You are expected to work 1-3 hours
outside of class time per week.**

Software

MaxMSP/Jitter: download at <https://cycling74.com/> for free trial (you can't save your work) or use a lab computer with a full licence.

Arduino: download at <https://www.arduino.cc/> on your computer or use at the lab. Arduino physical hardware can be found in EV-S2-618 with James.

Zoom: <https://zoom.us/download> used for in-class video tutorials.

Moodle: : <https://moodle.concordia.ca/moodle/login/index.php> used to post all of the class materials, assignments, and hand-ins

Weekly Breakdown

January 15 Week 1 - Introduction, Studio Tour + intro to MaxMSP.

January 22 Week 2 - Using Audio, sampling, making a basic synth.

January 29 Week 3 - More audio, sequencers, studio time & Audio Experimentation due by midnight & in class sharing.

February 5 Week 4 - Using Jitter, video input/output, controlling video - REMOTE CLASS -

February 12 Week 5 - More video, Open CV, Vizzie 3D objects, studio time & Video Experimentation due by midnight & in class sharing.

February 19 Week 6 - Using Midi controllers, stage lights and DMX.

February 26 Week 7 - Mid Term Project due - Group Project (Presentations in class).

March 5 BREAK :).

March 12 Week 8 - Arduino & sensors introduction.

March 19 Week 9 - More sensors & Sensor Exploration due by midnight & in class sharing.

March 26 Week 10 - Prototype due, in class discussion.

April 2 Week 11 - Studio time.

April 9 Week 12 - Final Critique.

Critique Policy

Critiques are critical opportunities for artistic development and growth. Students are expected to be both supportive and constructively critical with each other. You are expected to engage actively in critique and be prepared to engage deeply with concept and technical aspects of projects.

There are both technical and conceptual aspects to a class like this. We should be prepared to discuss both during critique.

Late Work Policy

All projects are expected to be handed in by midnight of the due date. Feel free to make changes after critique or finish documentation. Students will be deducted 5% per day after that. If there is any reason why you can't meet the deadline, notify the instructor via email or in class. The sooner you notify the instructor, the sooner they can help you find a solution or avoid late marks. If you need help accessing student services or resources, please reach out.

On Plagiarism

In Simple Words:

Do not copy, paraphrase or translate anything from anywhere without saying where you obtained it.

In a class like this, you will frequently use other people's code and examples. This is totally fine, but you should indicate where you get it from AND change it meaningfully. Work that is a direct copy from a tutorial will not be accepted, even if it is cited. Work based on a tutorial should be indicated and linked.

PAY YOUR FEES

1. Log into the Student Hub OR ([https://adsys2.concordia.ca/OFAF/
pages/ Default](https://adsys2.concordia.ca/OFAF/pages/Default))

Moodle Tour!

Resources:

Max Documentation: <https://docs.cycling74.com/max8>

Max Cookbook <https://music.arts.uci.edu/dobrian/maxcookbook/>

Andrew Robinson Video Tutorials <https://www.youtube.com/@AndrewRobinson26>

Inspo

Behnaz Farah

**Blindness: [https://
behnazfarahi.com/
blindness/](https://behnazfarahi.com/blindness/)**



**Iridescence: [https://
behnazfarahi.com/
Iridescence/](https://behnazfarahi.com/Iridescence/)**



Daniel Rozin: Mirrors

[https://
www.youtube.com/
watch?
v=kV8v2GKC8WA&t=
203s](https://www.youtube.com/watch?v=kV8v2GKC8WA&t=203s) & [https://
www.smoothware.co
m/danny/](https://www.smoothware.com/danny/)



The Stone Pad instrument was played on tracks from the latest album by Jhon William Castaño Montoya, a Colombian violinist and researcher.

[https://www.dezeen.com/
2014/05/07/fabrica-stone-
pad-musical-instrument/](https://www.dezeen.com/2014/05/07/fabrica-stone-pad-musical-instrument/)



Radom International
Rain Room: it's
raining, but you
won't get wet

[https://
www.youtube.com/
watch?v=EkvazlZx-
F0](https://www.youtube.com/watch?v=EkvazlZx-F0)

Bill Vorn



© Bill Vorn & LP Demers 2015, photo by Gregory Bohnenblust

[https://billvorn.concordia.ca/
menuall.html](https://billvorn.concordia.ca/menuall.html)

[https://www.youtube.com/
watch?v=JaUAVo8PBJ4](https://www.youtube.com/watch?v=JaUAVo8PBJ4)

MaxMSP tools:

[https://billvorn.concordia.ca/
research/software/
lifetools.html](https://billvorn.concordia.ca/research/software/lifetools.html)



Martin Messier

[https://
martinmessier.art](https://martinmessier.art)

/

Mutek - Village Numérique



📍 [Le Parterre](#)

18 h - 23 h

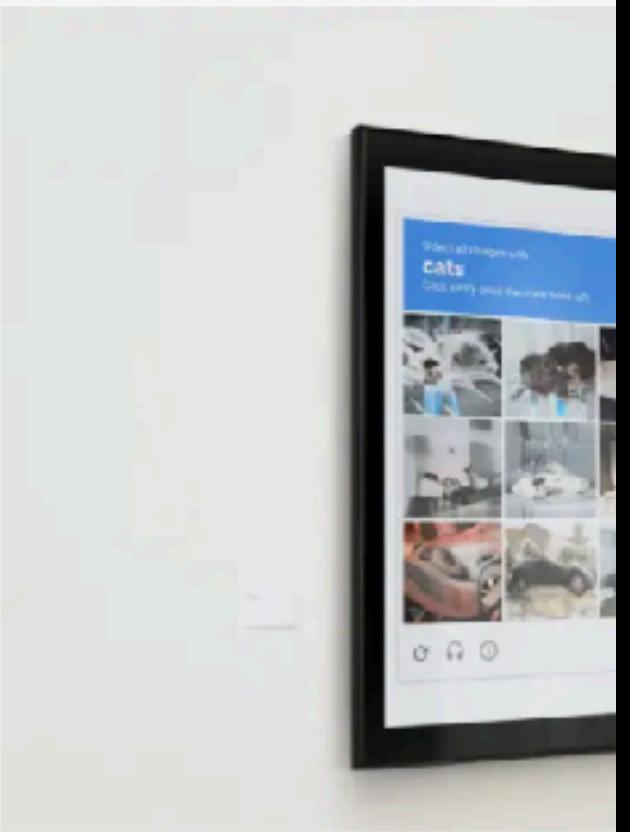
In camera

5

📍 [Le Parterre](#)

18 h - 23 h

I'M NOT A ROB



[https://village-
numerique.mutek
.org/fr/parcours](https://village-numerique.mutek.org/fr/parcours)



Myriam Bleau

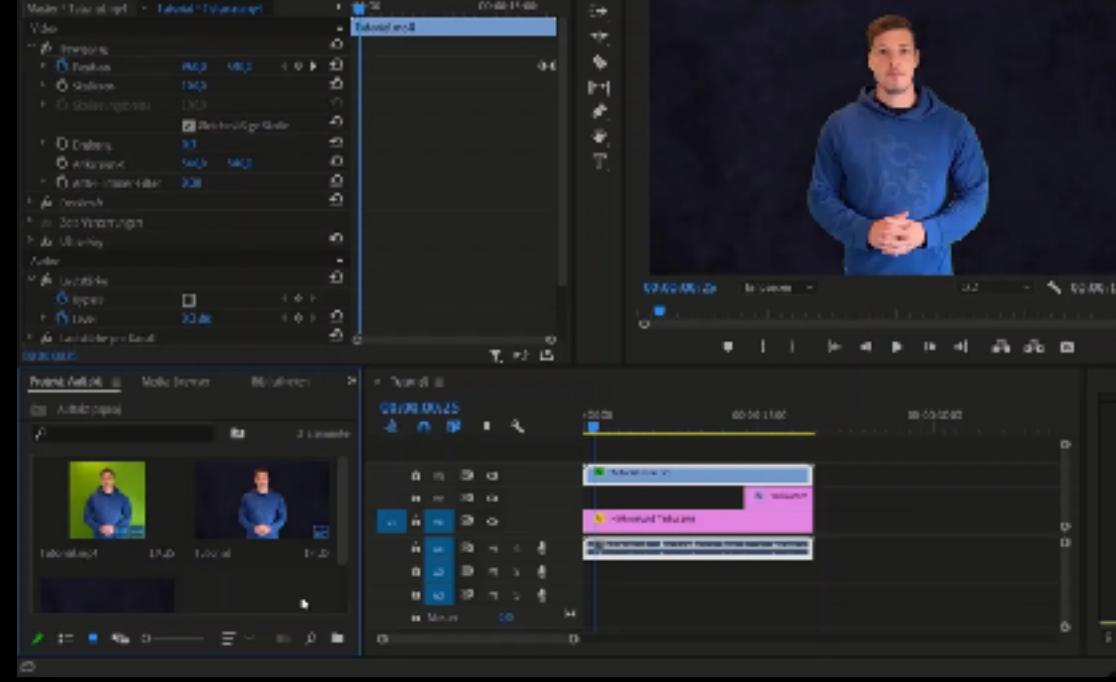
https://

www.myriambleau.com/

Imogen Heap



[https://
www.youtube.co
m/watch?v=ci-
yB6EgVW4&t=53
s](https://www.youtube.com/watch?v=ci-yB6EgVW4&t=53s)



Camera



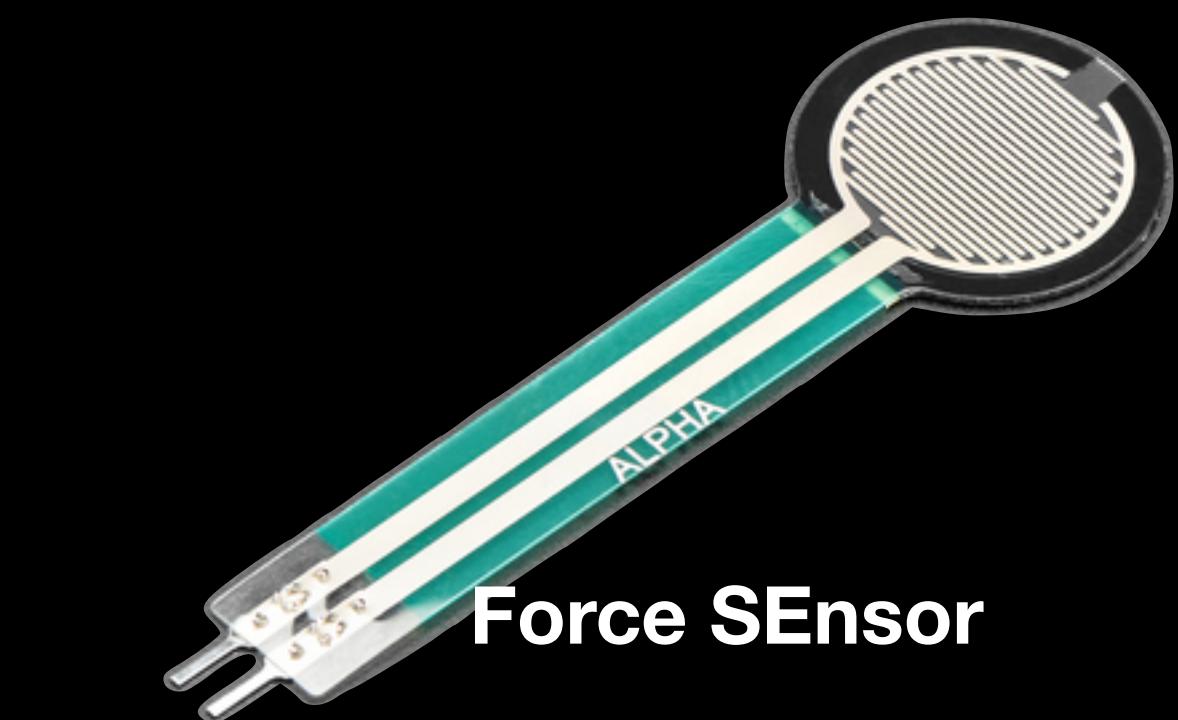
Sound File



Synthesizer



Lighting



Force SEnsor



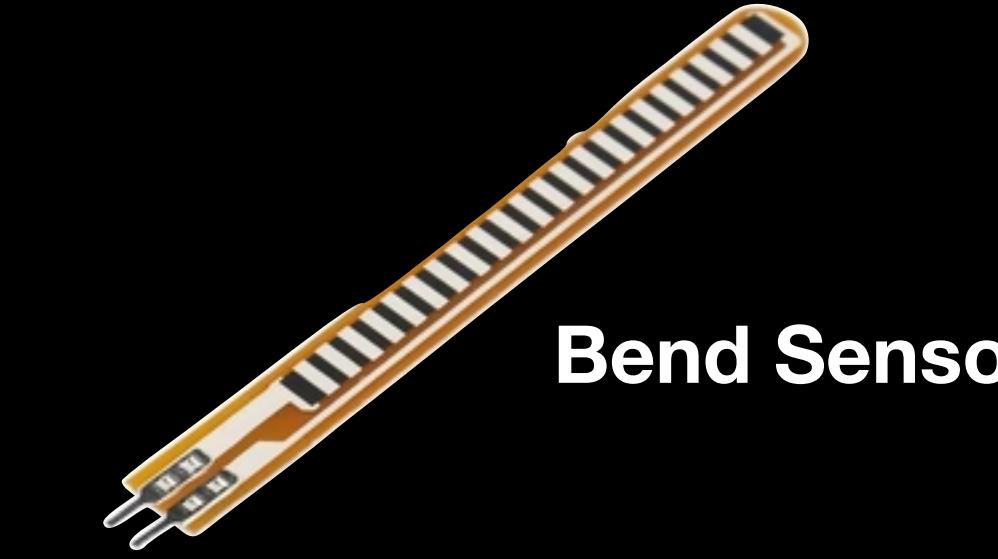
Buttons



Ableton Live



Arduino



Bend Sensor

MaxMSP can help us interface between
these devices, and more!

Chat GPT: Please don't.

AI is a great tool once you know how to code. It can help you get things done faster. But its difficult when you are learning because you can't tell what it did, it makes it hard to debug. I can't help you as effectively if you used chatgpt.

What can max do?

- Create, manipulate, and enhance sounds
(music, ambient sounds, trigger sounds)
- Mix, control, and filter videos and 3D assets
- Interface between real world digital objects, ex
Use sensors to influence music, objects, lights,
sound and video
- Connect different hardware and software
across networks

The Basics

- I am not expecting you to do complex things. We're here to explore and expand our existing practice.
- If you want to explore more and write complex code, please do!
- Take the time to learn the basics, you will understand more in the end.
- Sometimes we are learning together, programming is big!

Relax. Take your time!

**Nobody can program while stressed.
Take breaks, think, be patient.**

**The only way you will get better is by
doing.**



Terminology

Patch: A max program

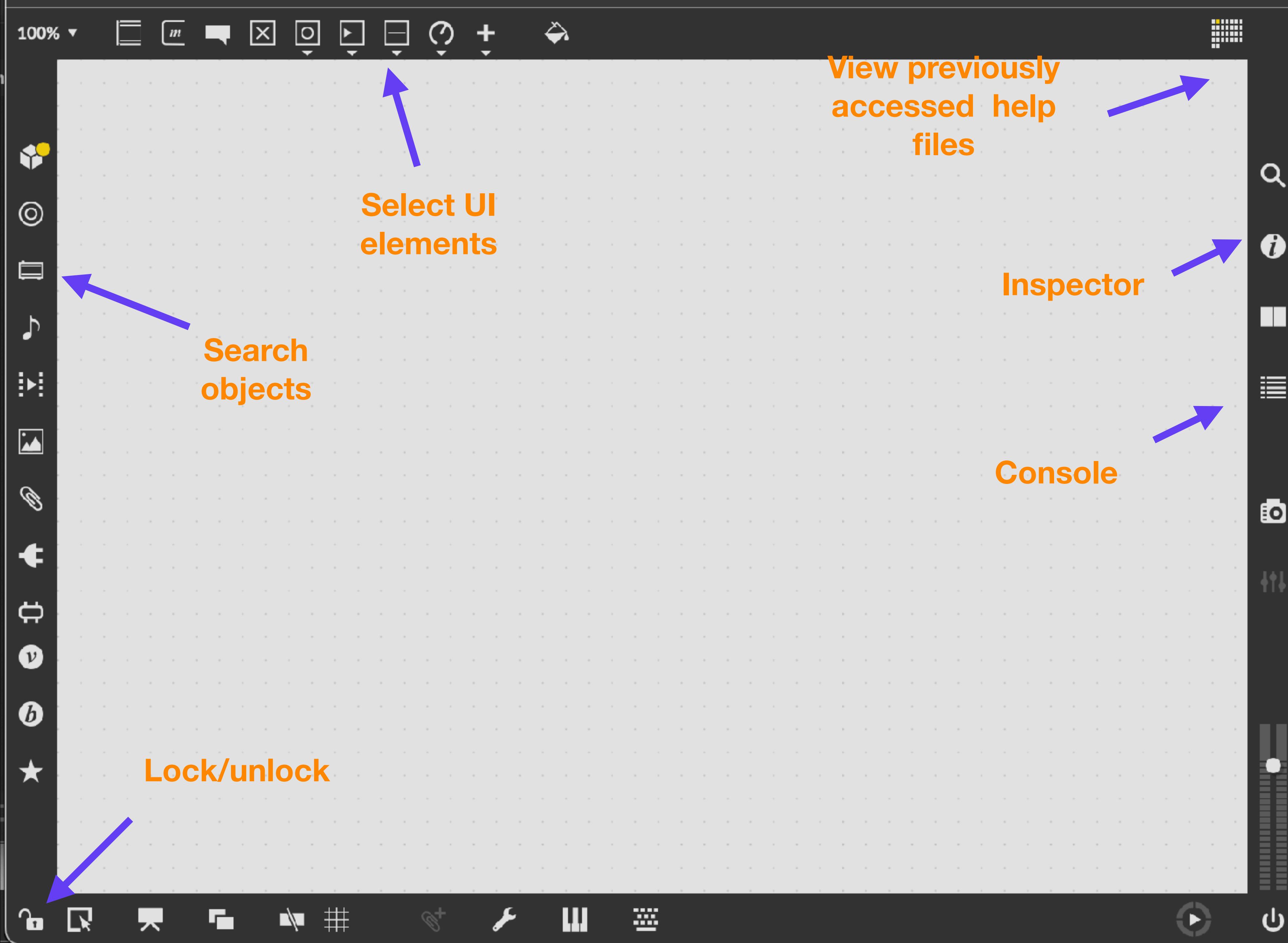
Objects: Blocks that do something

Patch Wires: Connect each block

Lock/Unlock: The patch is editable or not
editable

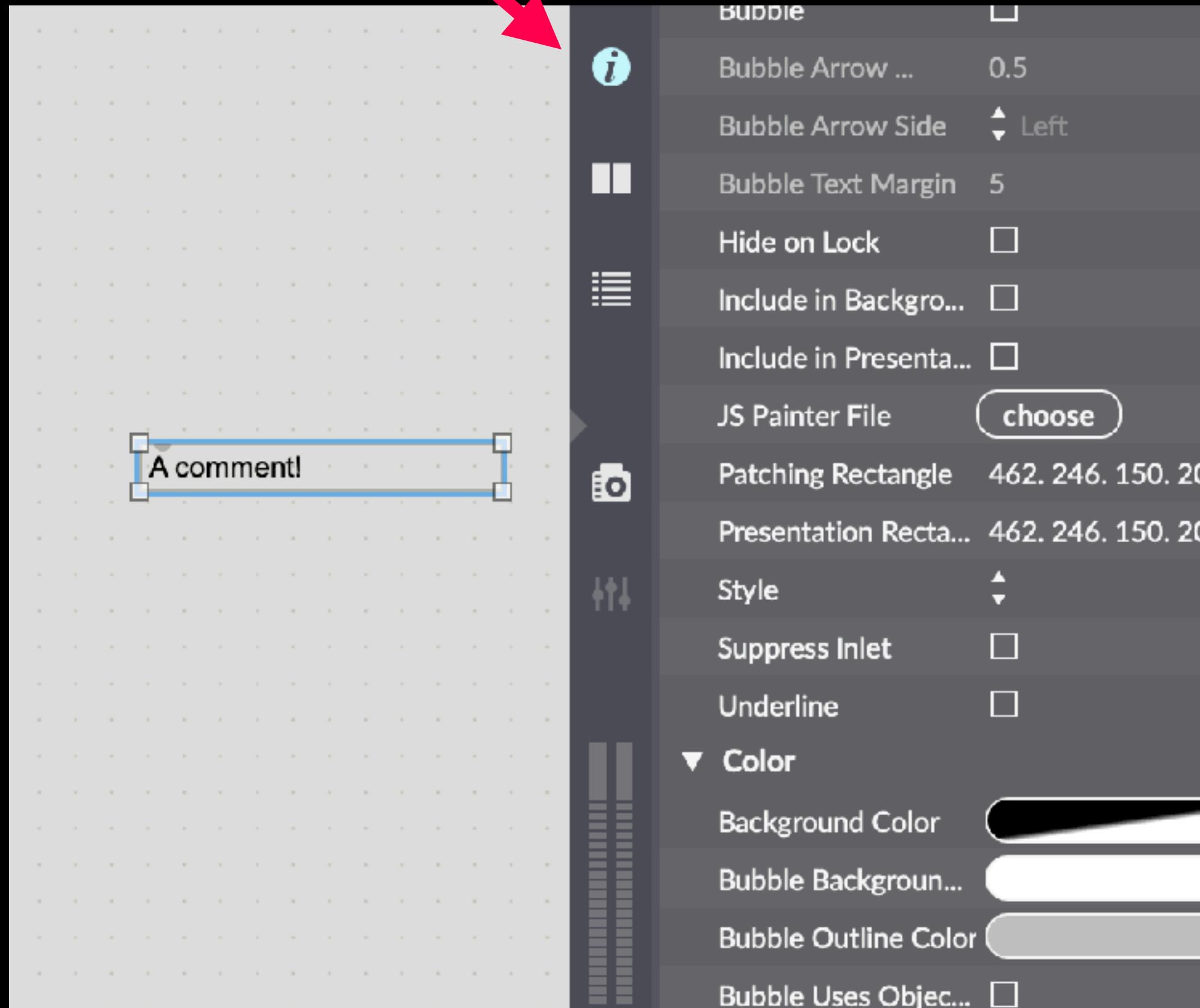
Bang: A signal sent often to active an object

Console: a text output useful for debugging

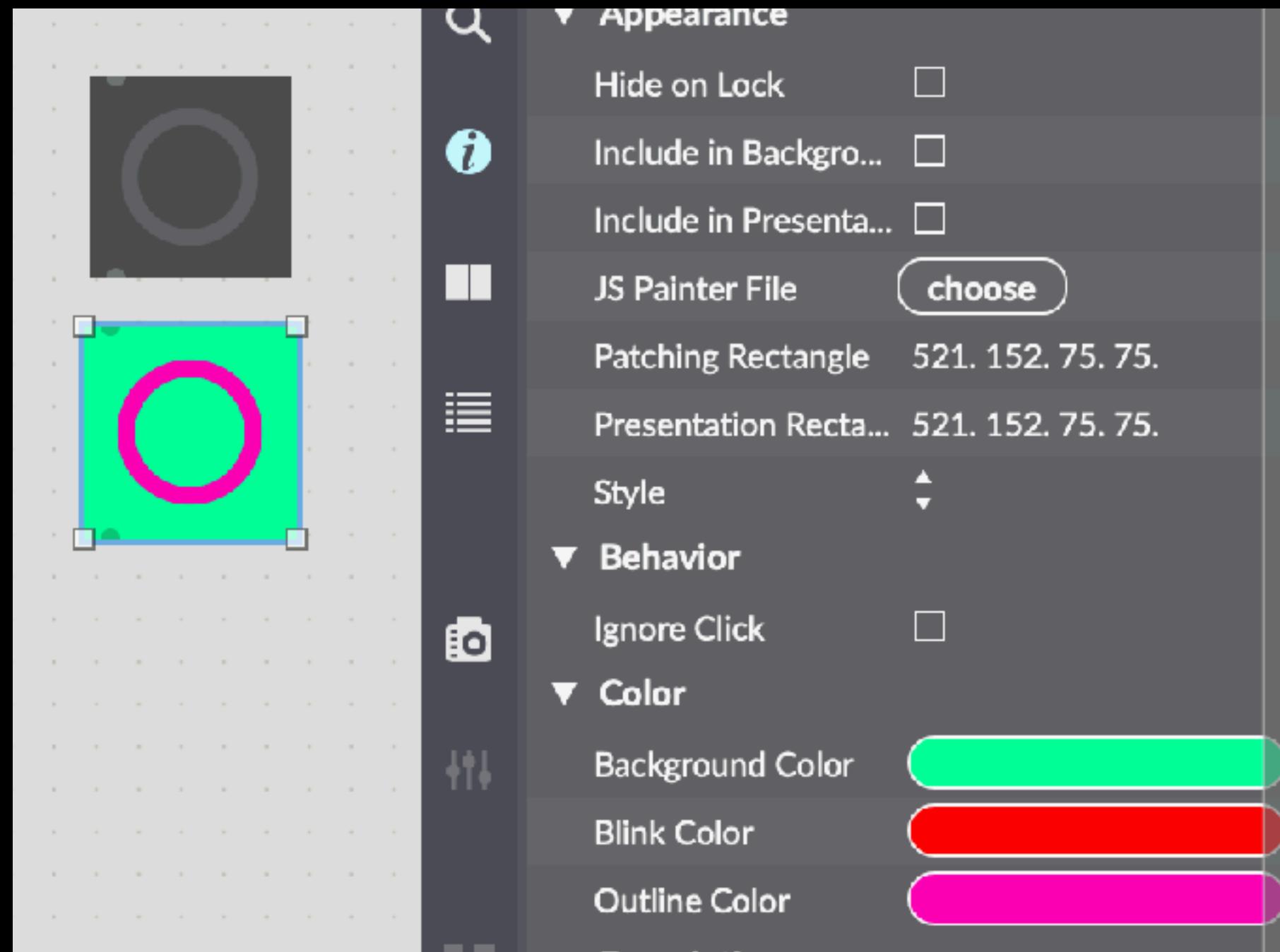


Comment: help describe your patch in a readable way. Press “c” to make a comment. Type what you want to appear. Select the comment and press the inspector. You can change how it looks here.

inspector



**Button: Buttons can
be used to activate
things in your patch.
Press “b” to make a
button. You can also
edit it in the inspector!**



MaxMSP is both an interface and code. Locking and unlocking your patch lets you use or edit it.

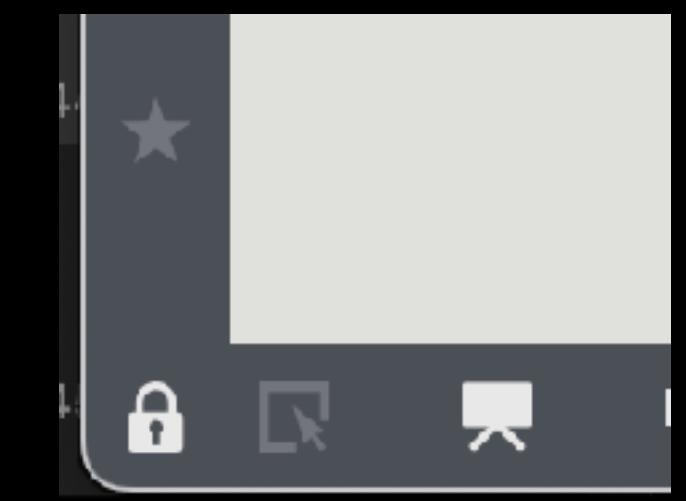
To use the button, lock your patch (bottom left) or ctrl+click. Press the button, wow!

When a patch is unlocked, you can edit it. Locked patches are in presentation mode.

Unlocked



Locked



Lets look at the button...

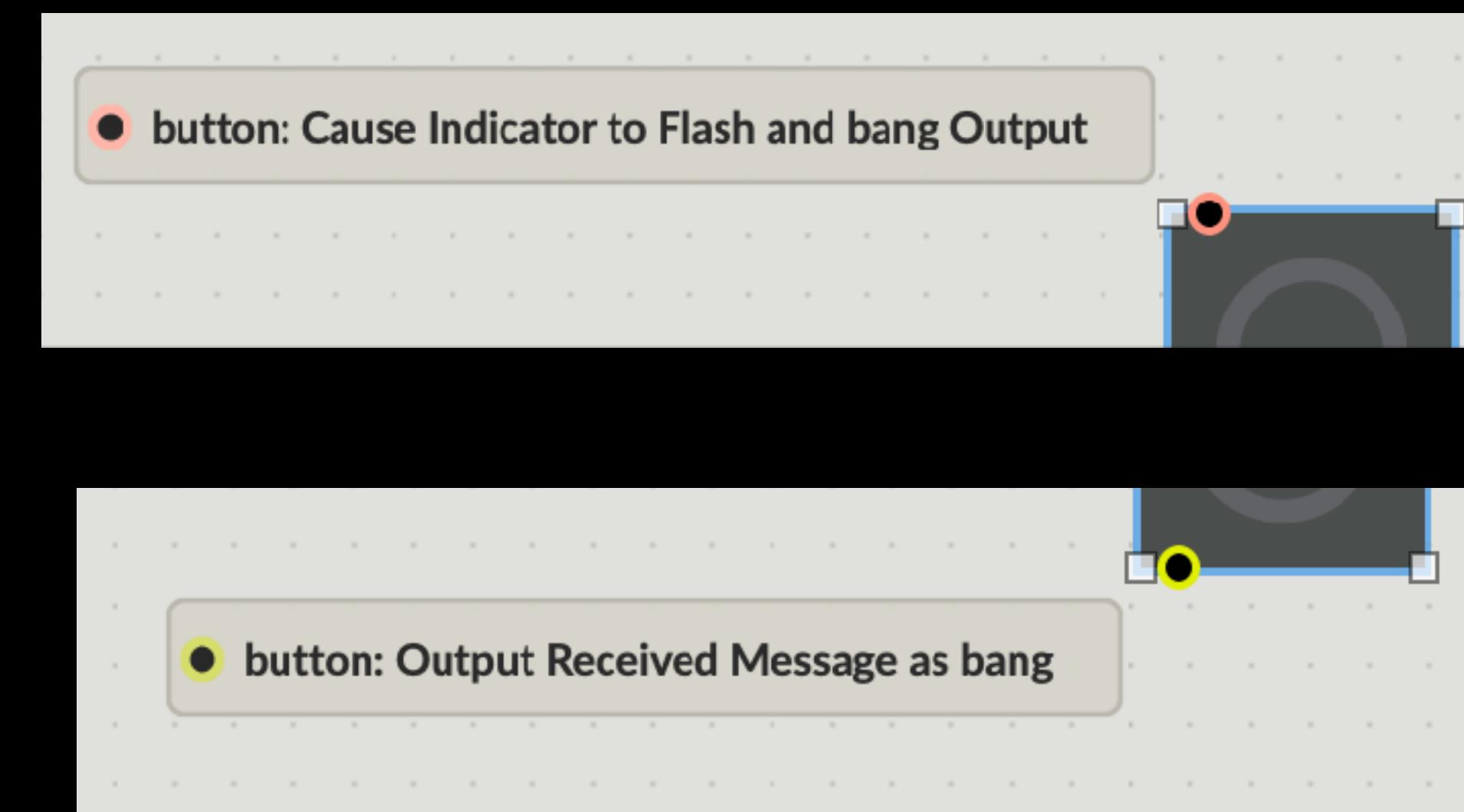
The button as an input and an output.

Hover over each to get more information, which tells you how to use it.

The top input can trigger the Button.

The bottom output can make something happens when the button is triggered.

Other objects will have different inputs and outputs.



All objects have:

Inlets: Data that the object accepts to perform its function
Some are RED and some are BLUE. We will get into this a bit later.

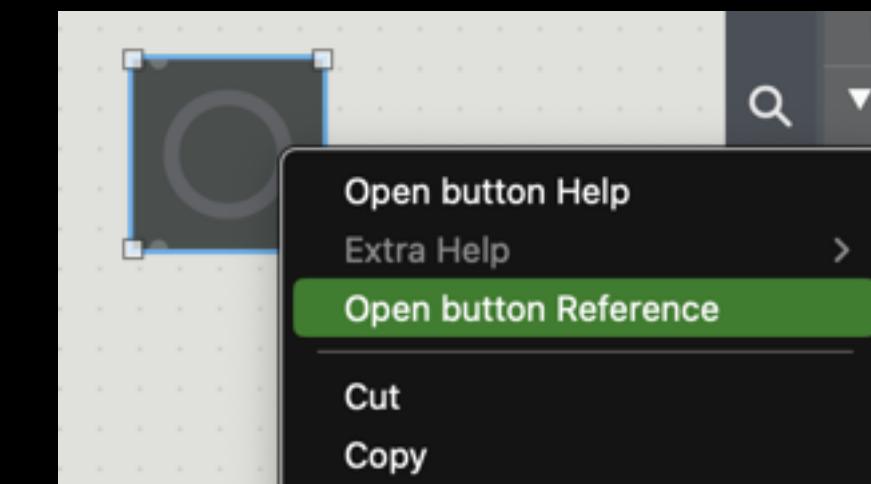
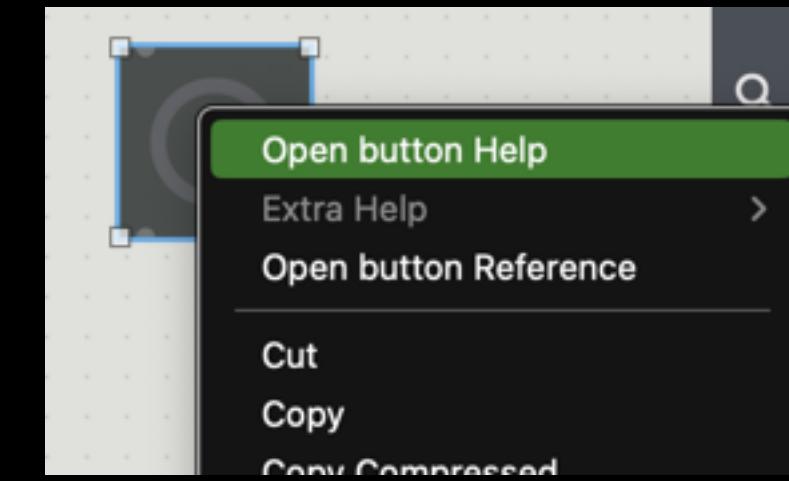
Outputs: Data the object sends out to another object or output. Outputs are YELLOW.

Some have:

Arguments: Pieces of data to make the object have a specific behaviour

Get Help!

Right click on an object and use the **HELP** or **REFERENCE**. They're so useful and include interactive examples.



First Connection

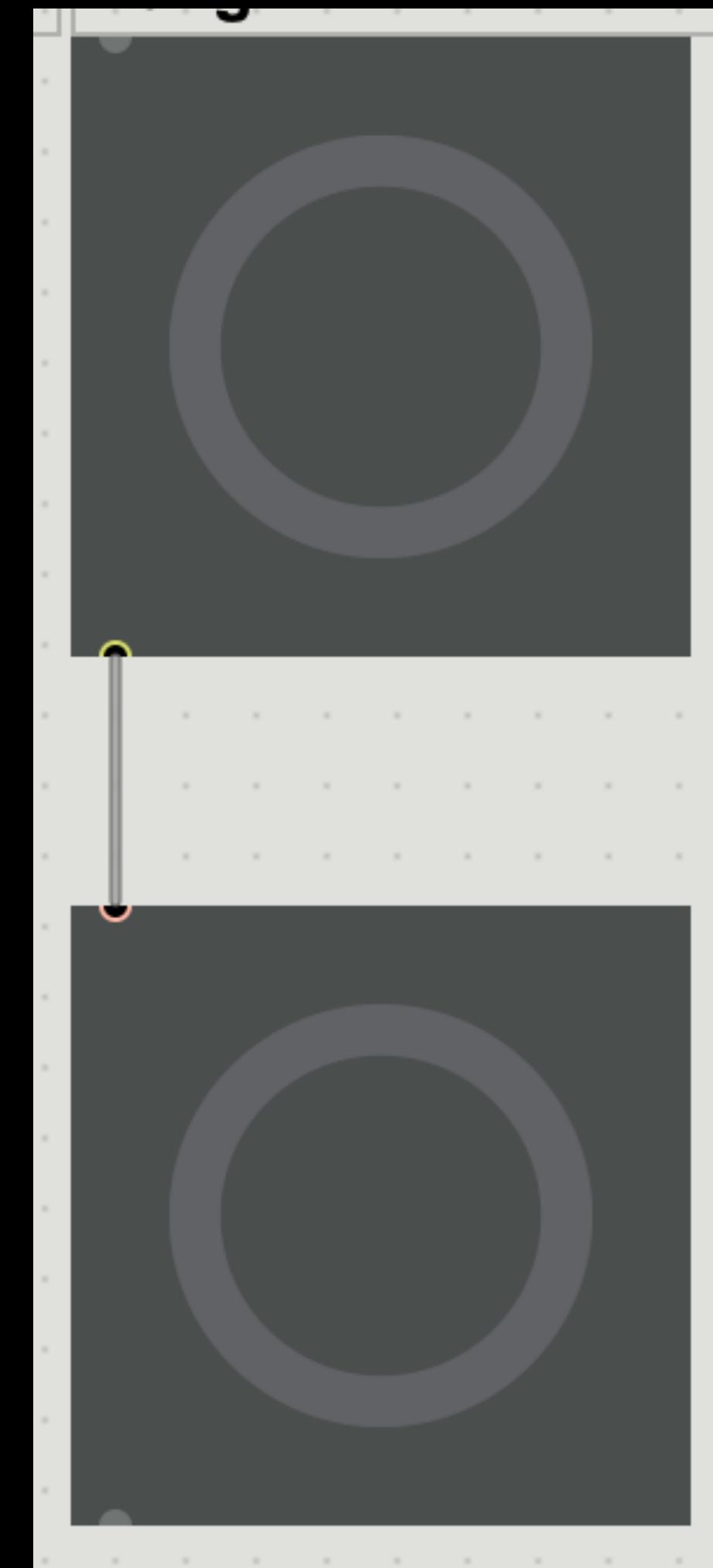
Make two buttons.

Connect the **OUTPUT** of one to
the **INPUT** of the other.

This is called a patch cord.

Lock your patch, press the
button!

One button is triggering the other,
so cool.



Every week we will have examples on Moodle. However, more examples will be available on the class GitHub

Either save the file OR copy and paste the code into MaxMSP

Open first_Patch.maxpat

Button: Press once, sends a bang once.

Toggle: Press once, sends a 1 and a bang, press again sends a 0 and a bang.

Dial: Sends a bang every time it changes.

Hello Max!

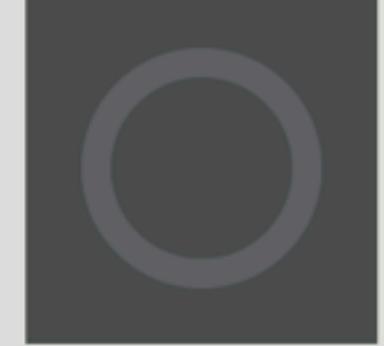
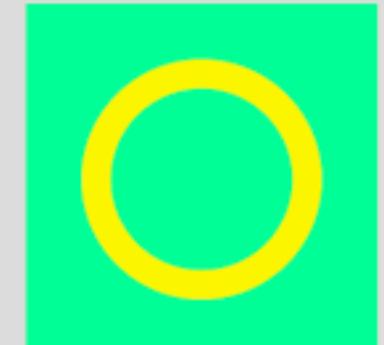
This is a comment.
Press C to make a comment

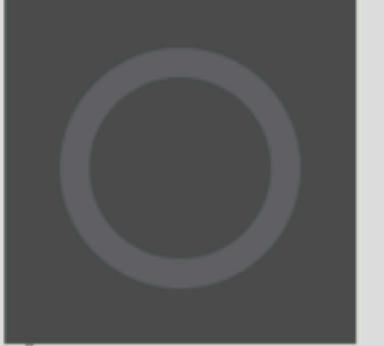
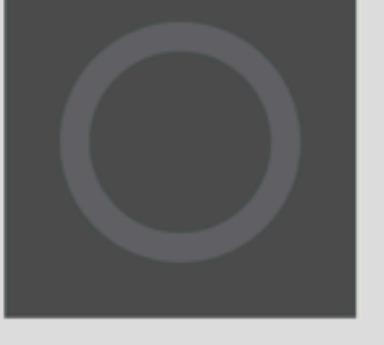
Lock / Unlock
Unlock the patch in the bottom left to edit it
Lock it to interact
Ctrl+click is a shortcut

Make Objects
Type N in your unlocked patch to make a new object.
Start typing the object name, hit enter to make it or select it from the dropdown.

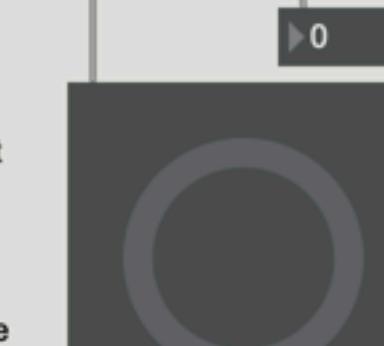
Learn More
Right click on an object to open the help menu
Click the "I" to edit the properties of an object in the inspector

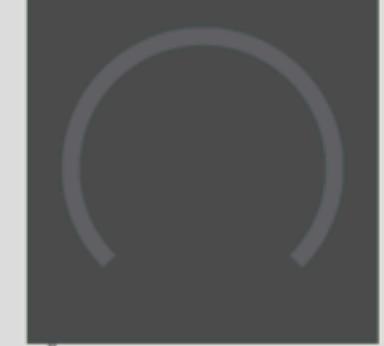
Inputs/Outputs
Every object has inputs/outputs
Yellow: Outputs
Red: Hot Inputs (triggered when they are on)
Blue: Cold inputs (triggered when they receive a message)

Button
This is a button.
Make it by pressing "n" for new object, then typing button
Or, press b for a shortcut!



Bang
All max objects have inputs and outputs.
A button outputs a signal called a bang.
A patch cord can connect inputs and outputs.
Here I have connected a button's output to another button's input.
The bang is received by the button!



Toggle
This is a toggle.
Type "N" for a new object and "toggle" or T as a shortcut
A toggle outputs a BANG and also a number, a 0 or a 1.
Press once to turn it on, and again to turn it off.



Dial
This is a dial.
Type "N" for a new object and "dial" as a shortcut
It outputs a bang every time it changes.



Use the inspector to change how things look, or their parameters

Print is a way of displaying data that can help you understand how your program is working.

Print messages appear alongside warnings in the console.

- Lets make a print object. Press n and then start typing print

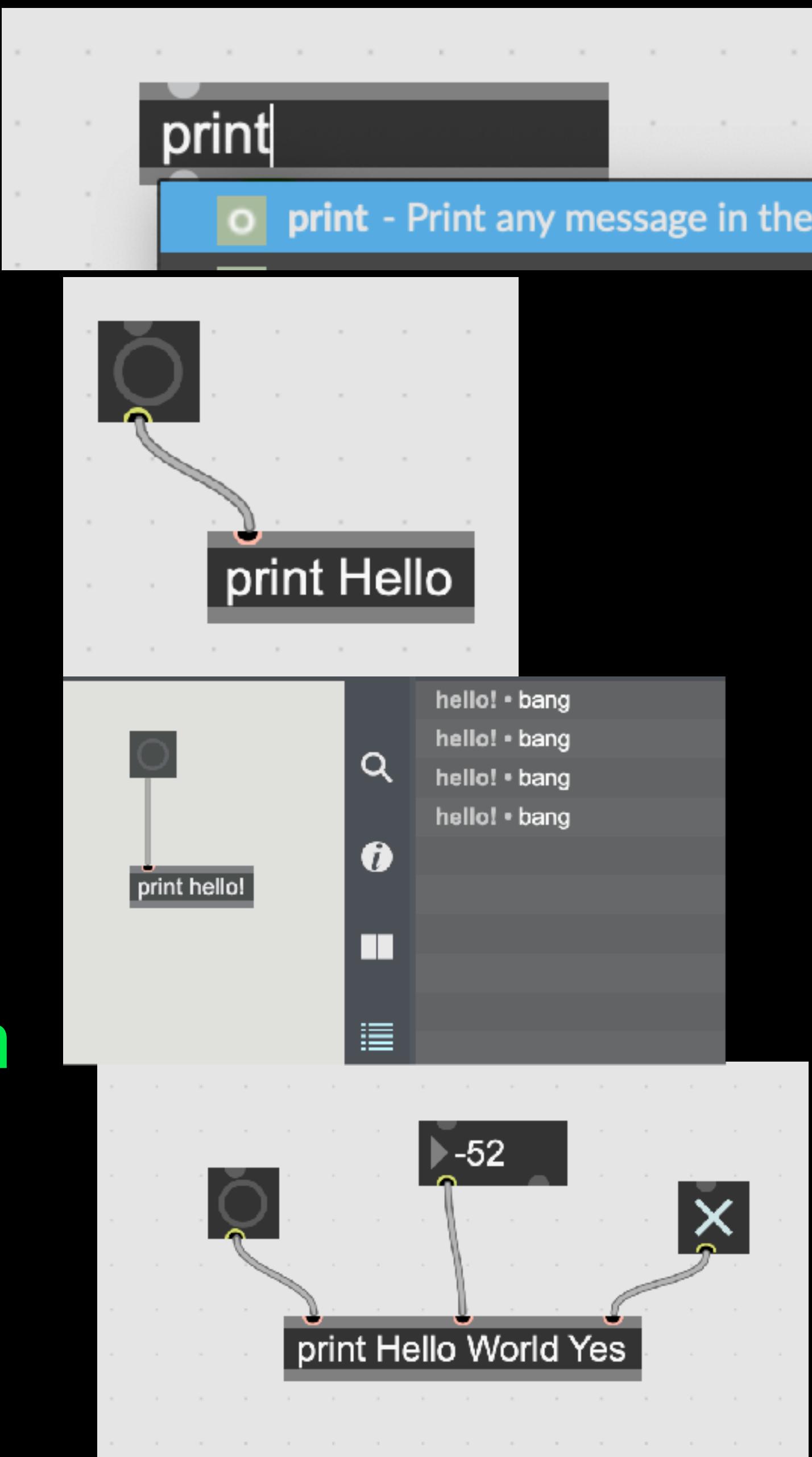
- Lock your patch and open the console

- Nothing happens! Why? You need to trigger a change in your inlet

- We can try a button. Press b, connect the button to the print inlet. Lock the patch and press the button!

- Now try with a number and a toggle

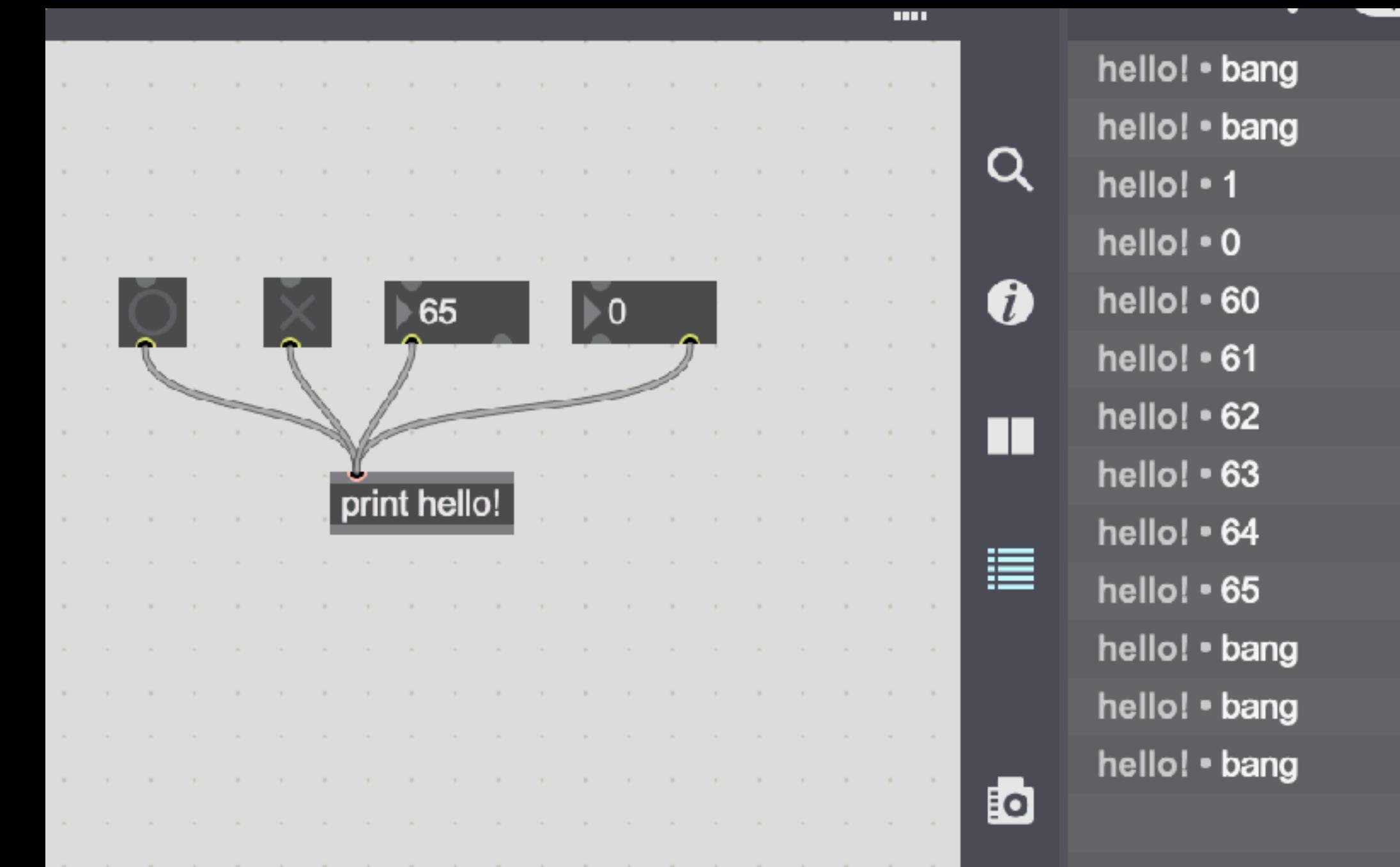
- A new outlet gets created for each word



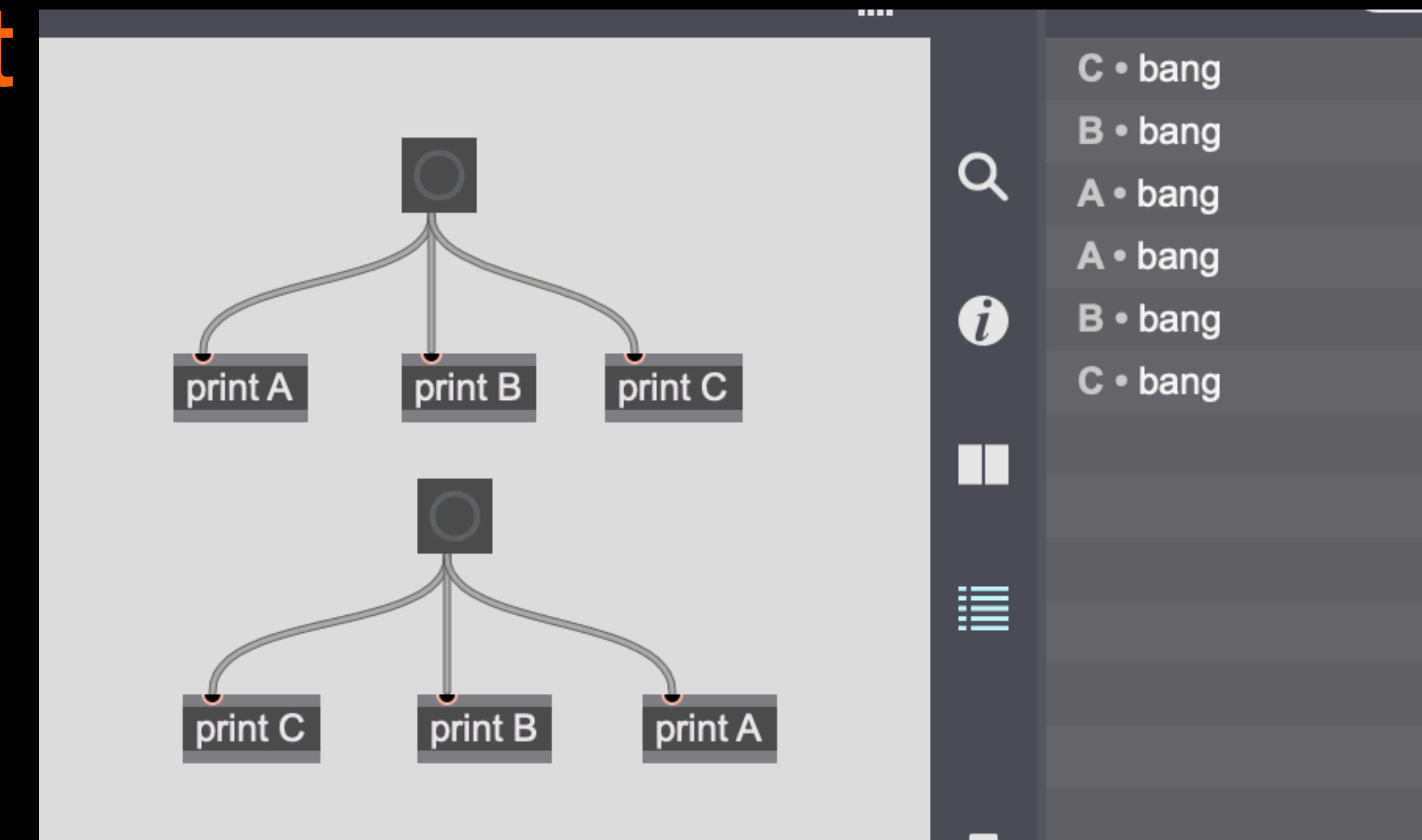
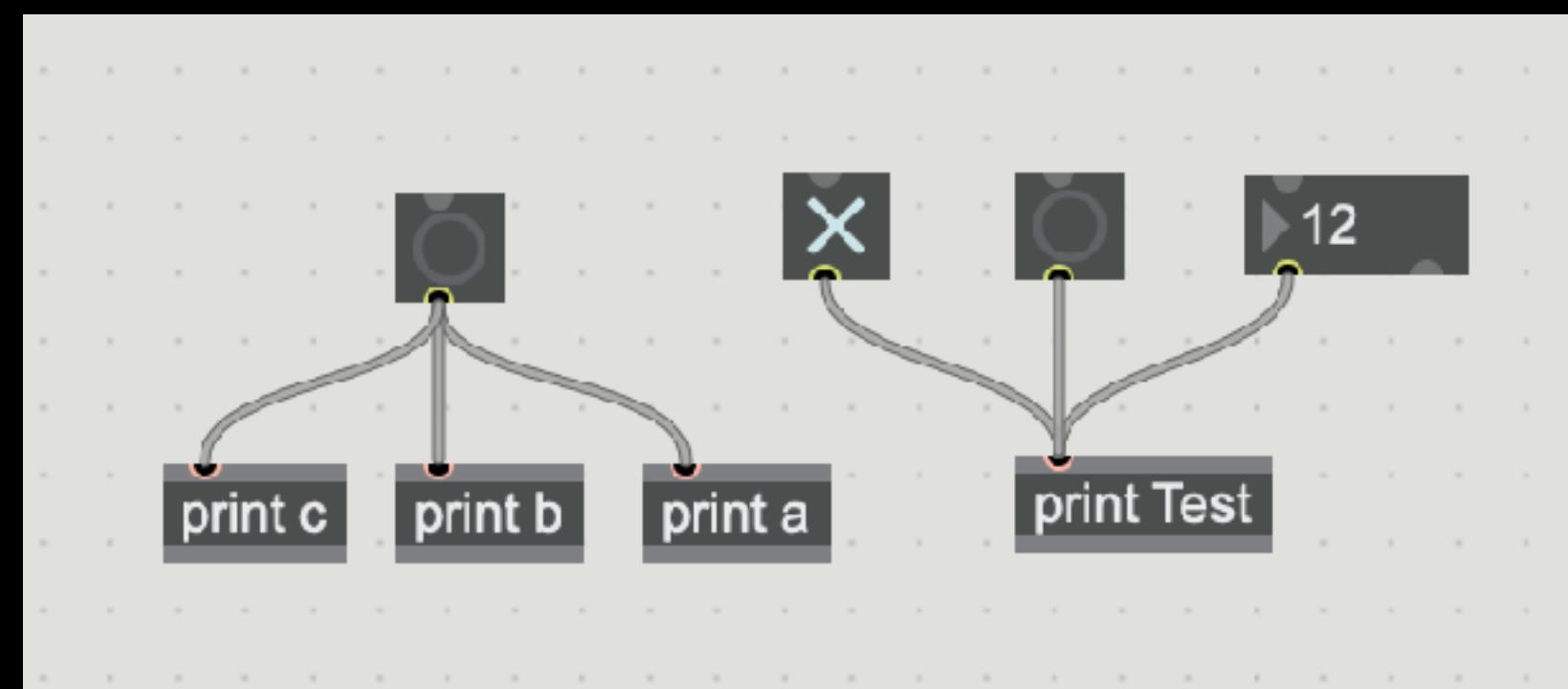
Print

-We can see print output different results with different inputs.

- Button is always a bang
- Toggle is on/off
- number left inlet is the number in the box
- Number right inlet is bang on tab



Flow right > left

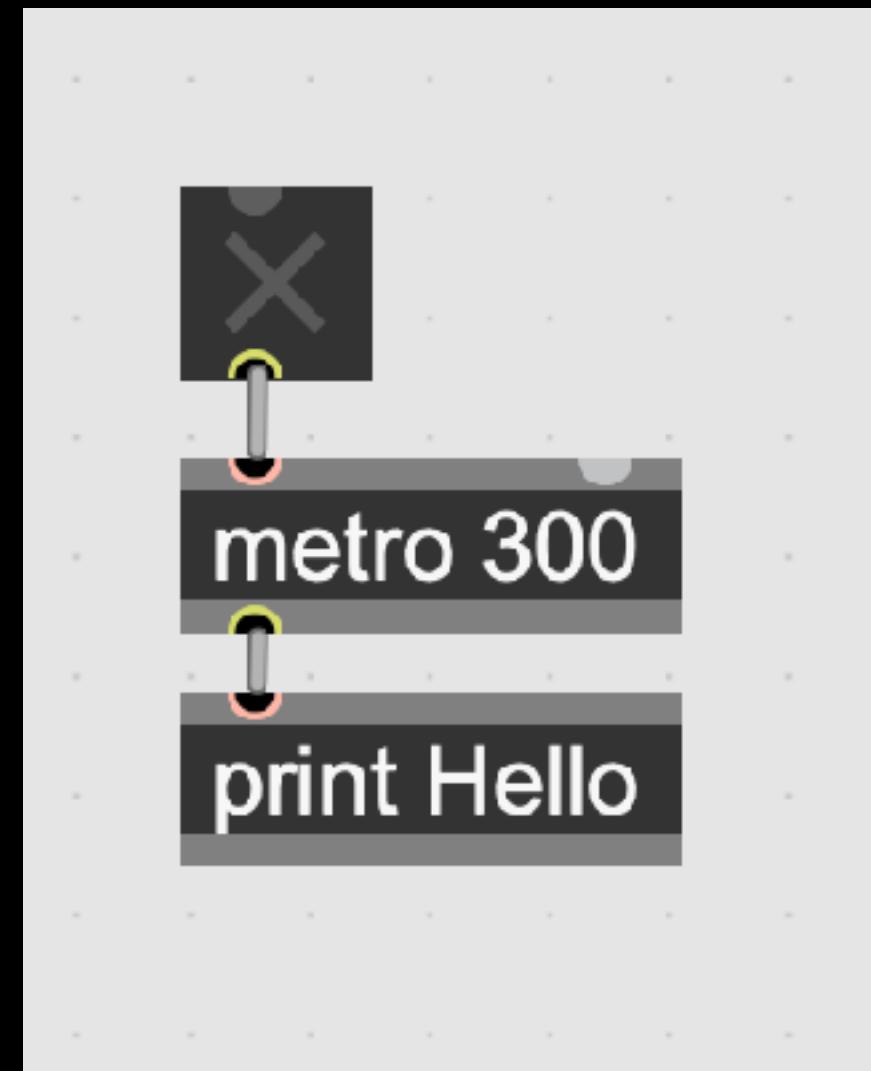


Lets get more dynamic with... metro! Short for metronome.



Metro has 1 argument, which is the number of milliseconds between each output. Default is 50 if you put nothing

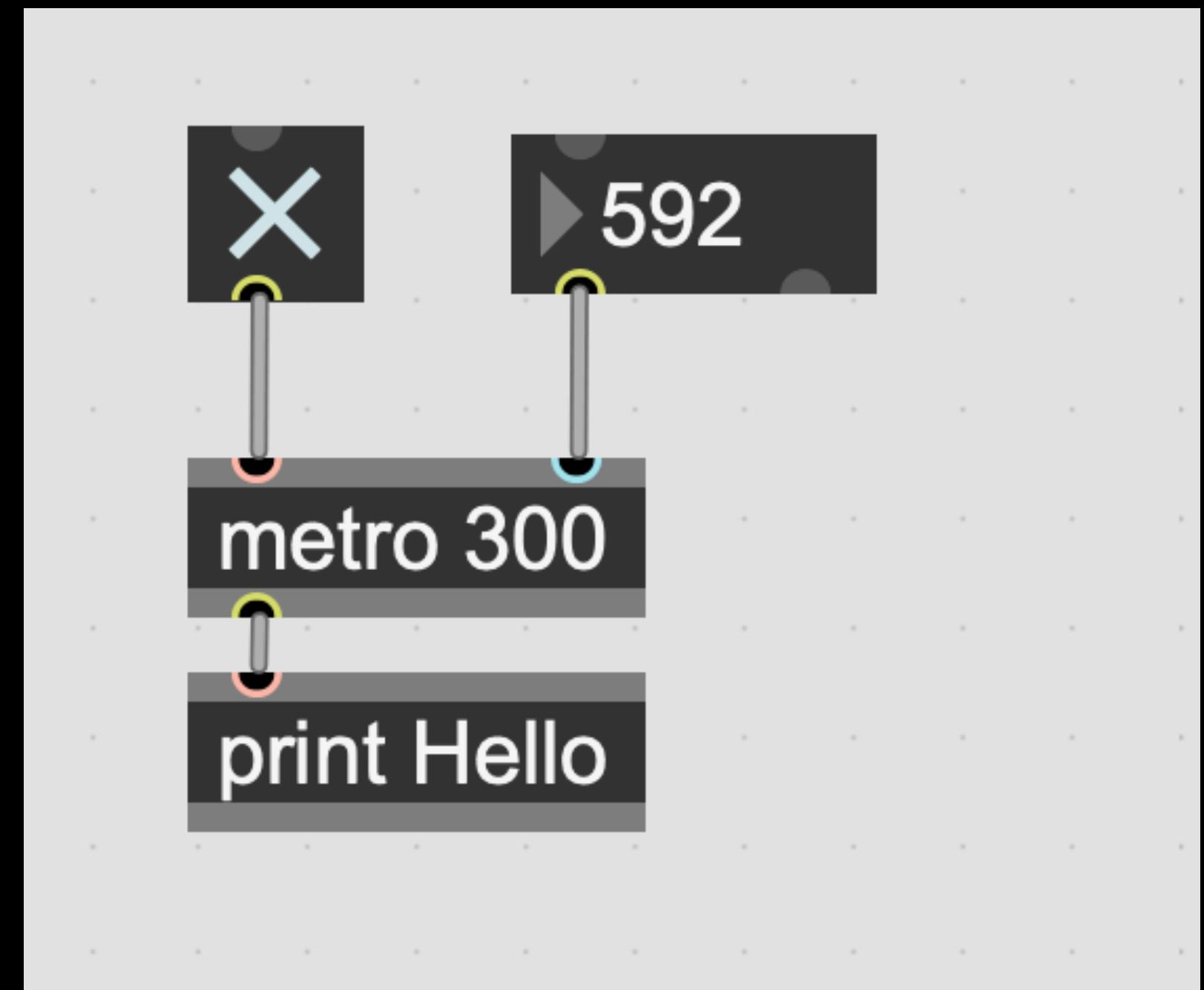
We can use the TOGGLE object to turn it on or off. Lock your patch and try it.



Metro

I can add a number object that lets me change the internal. Lock your patch and try this.

** Why does it update if it is a cold inlet? Because the metronome is continuous!



Shortcuts:

n: new object

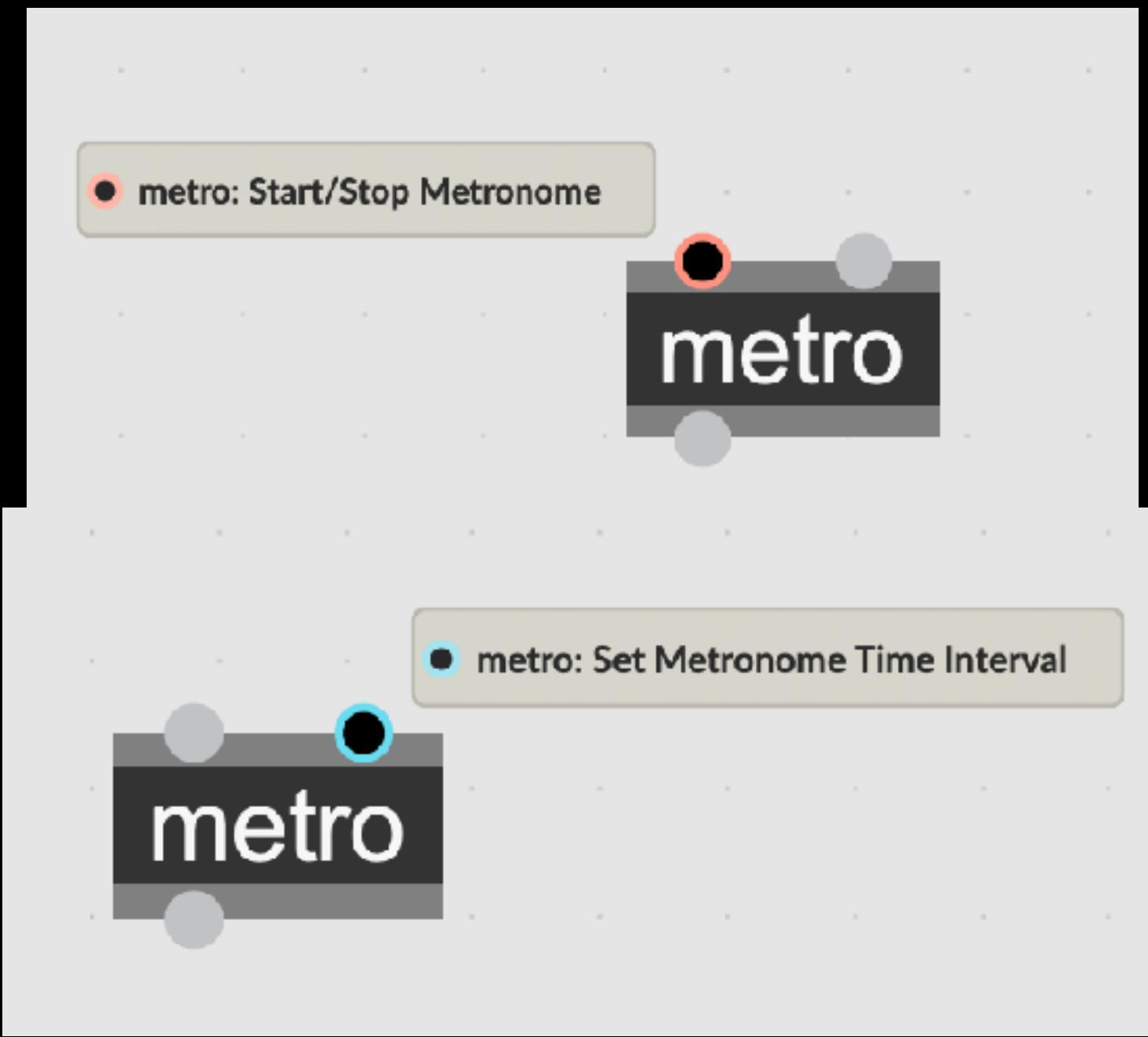
b: new button

t: new toggle

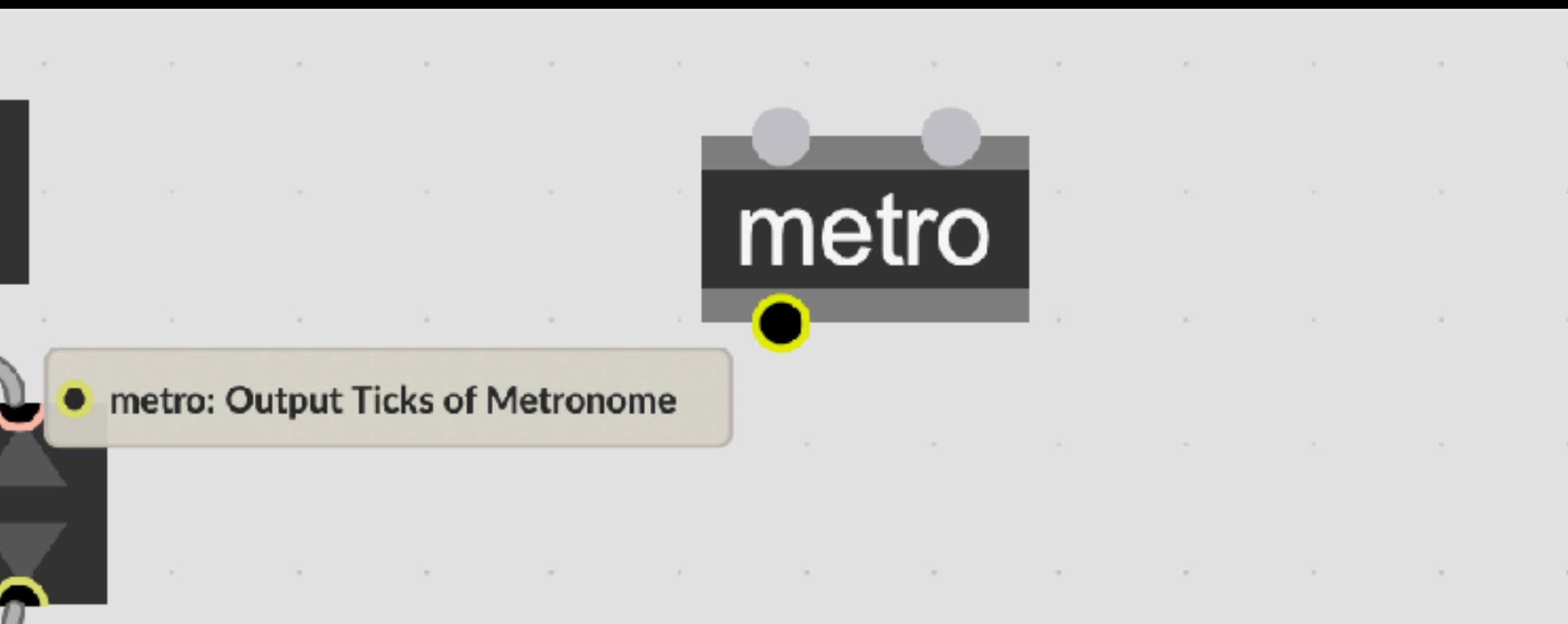
Ctrl + click lock/unlock

Make sure your patch is unlocked!

Red: Hot Inlets triggers outputs when they come in



Blue: Cold Inlets: setting a variable not NOT triggering a response



Yellow: Outputs data to whatever is patched to it

Hover over an outlet/inlet to see more information about it

Make your first annoying sound

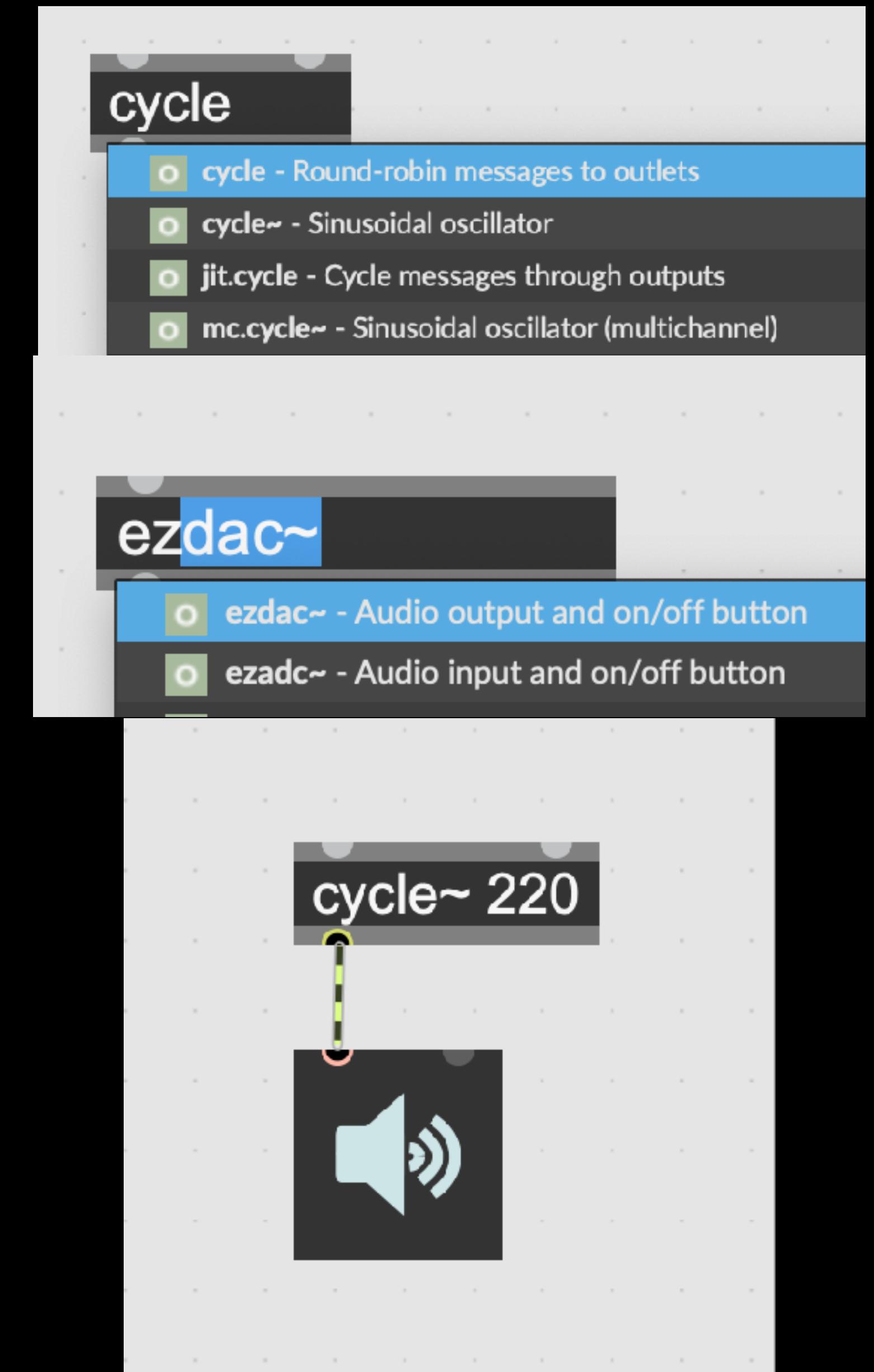
- Create a **cycle** object, its an oscillator.

Make sure to use the one with ~ because that means audio signals! I put 220 for the frequency.

- Create an **ezdac~** object. Its a simple digital to audio converter. It makes sound

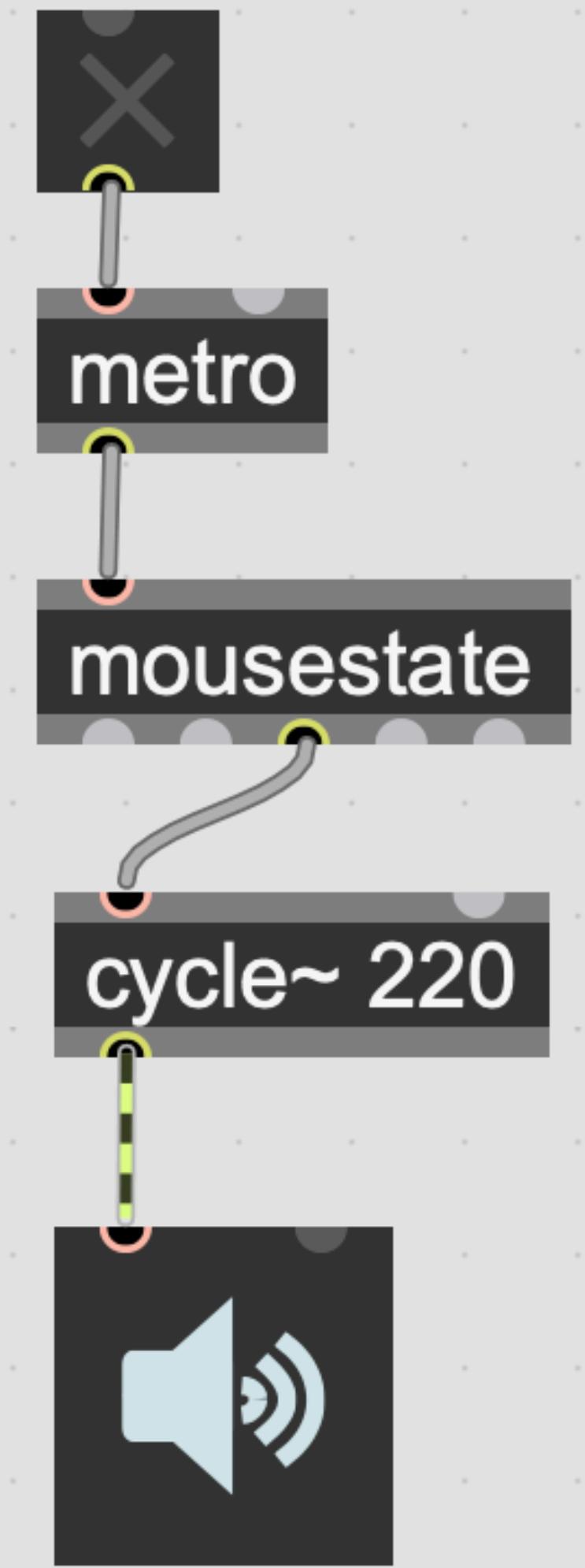
- Connect them, see it uses a **YELLOW AND BLACK** patch cable, this means its an audio signal patch

- Unlock your patch a d enjoy



Integrate mouse position

- Mouse state has a variety of different outputs, check them out in Help
- But it needs to be updated, so we'll use a metro
- Metro needs to be toggled on and off!

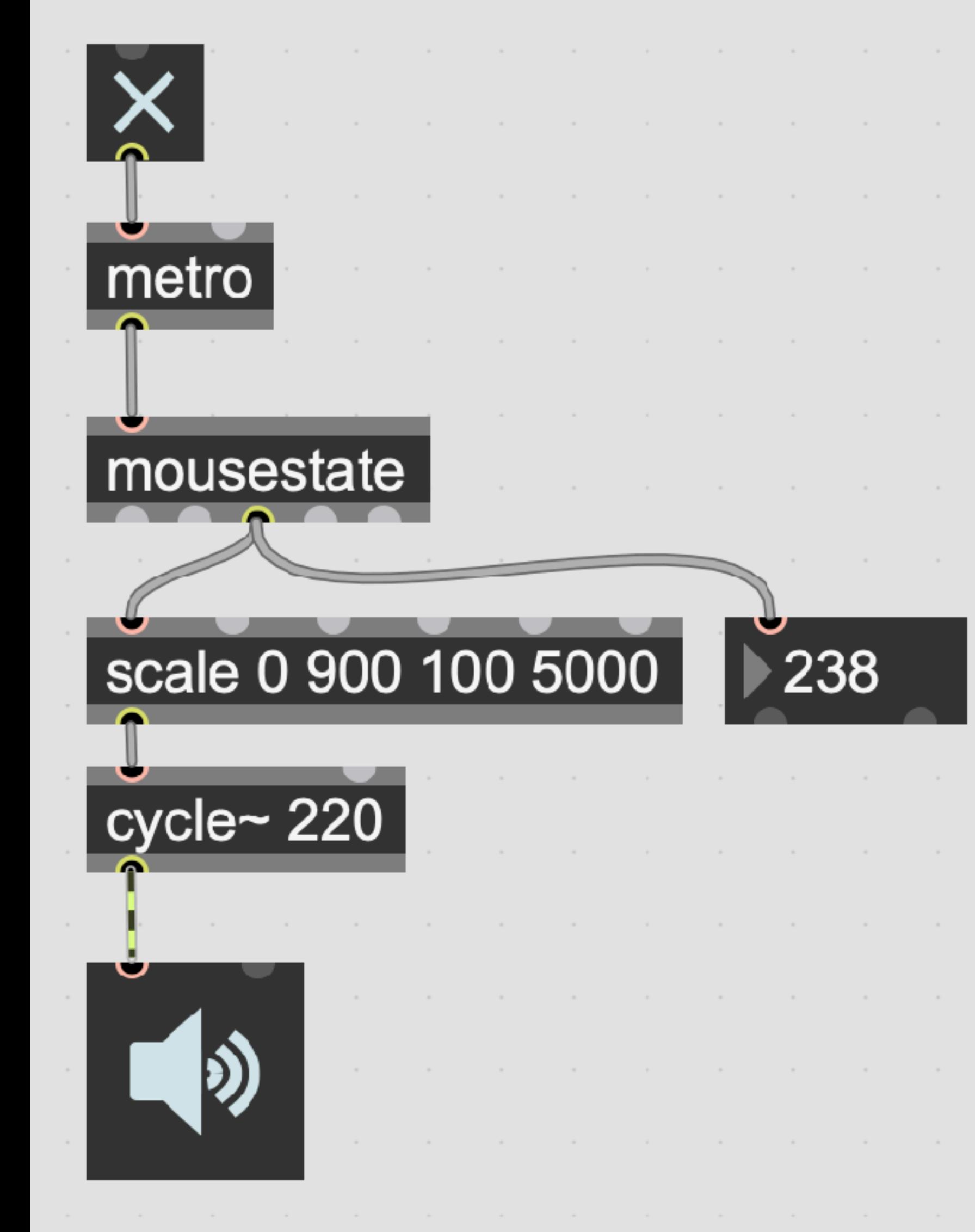


Scale to change the sound

- Frequencies are the note of the sound. Here we're using whatever mouse is. Lets look..

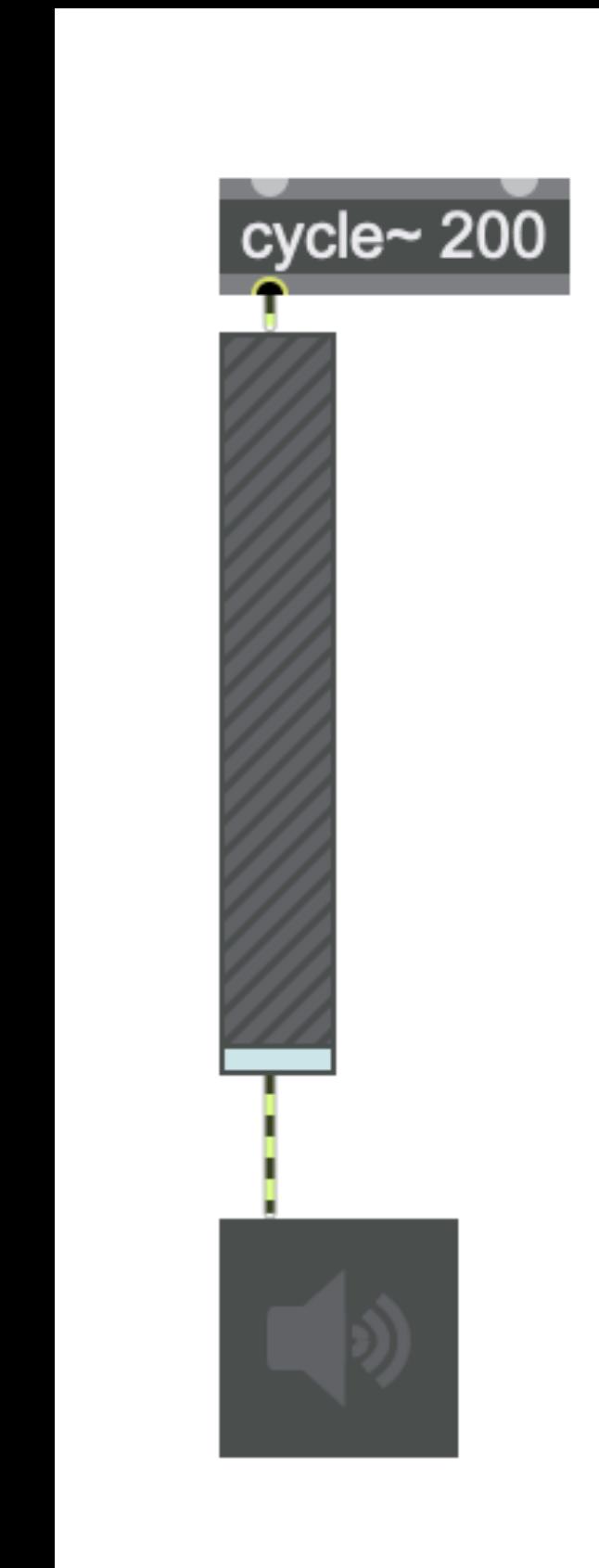
I get between 0 and 900 for the vertical value

- I can use scale to change that into a much wider range, I get a more interesting sound!



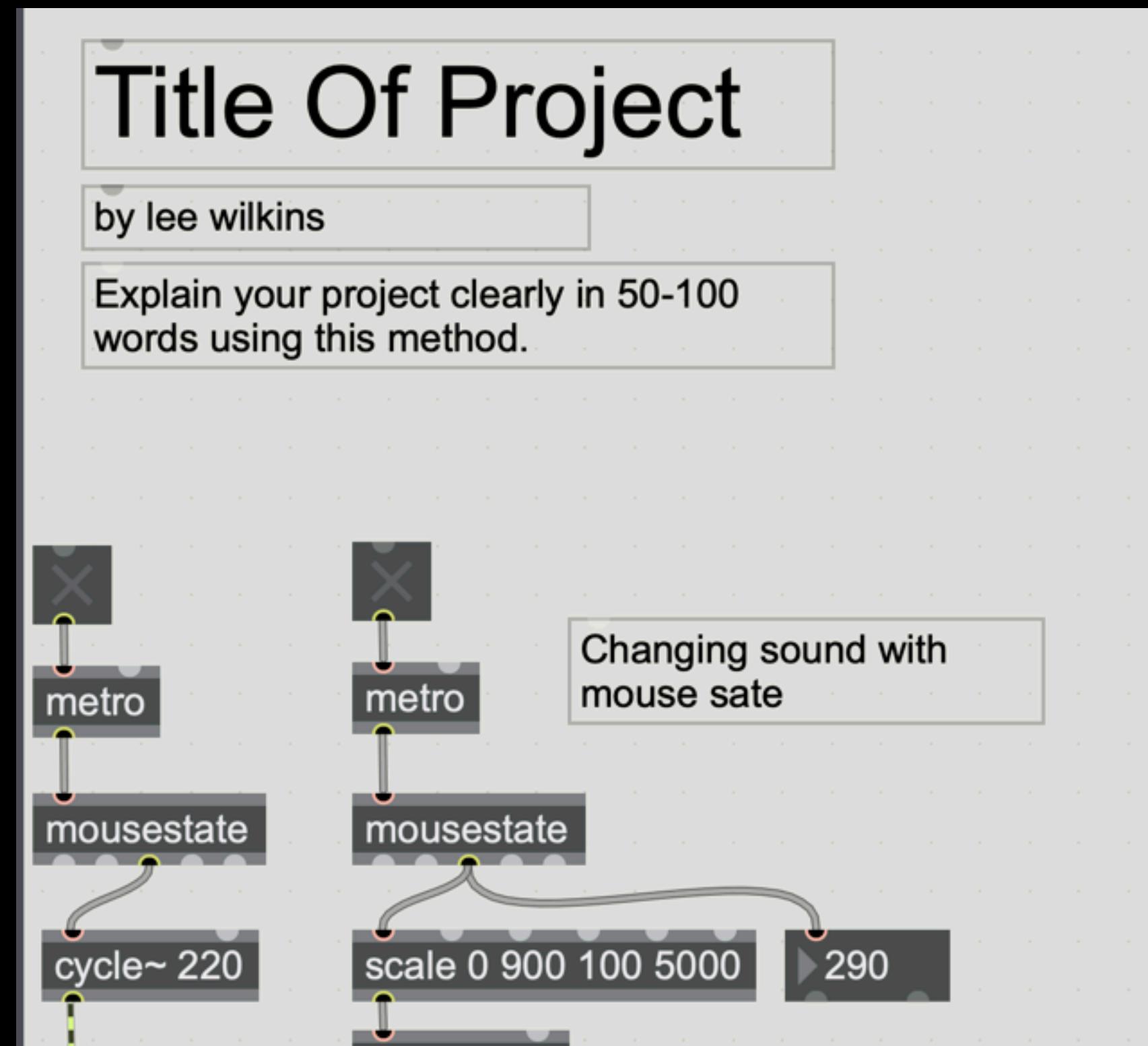
**Click on the audio icon
to turn off the sound or
add a Gain object.**

**You may need to
change your audio
output when you add
headphones. Find the
menu under Options >
Audio Status**



Comments & Layout

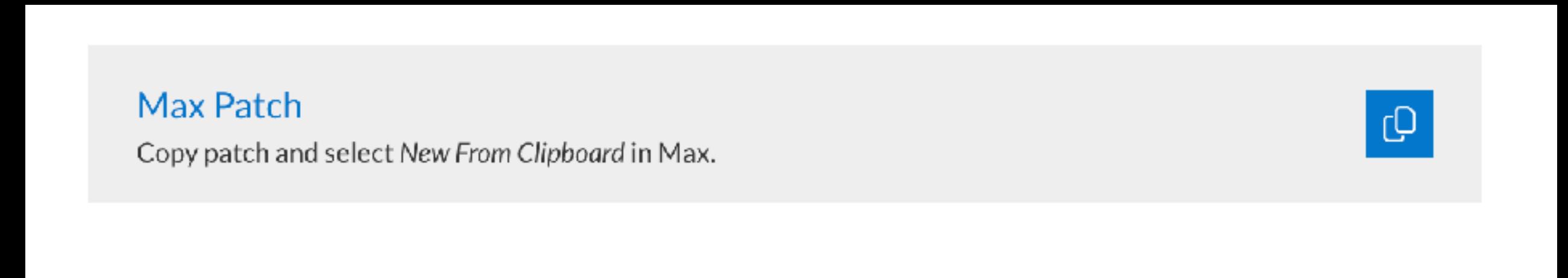
All project should have clear comments, title and authorship. Use the Comment object.



You can copy Max code and paste it directly into your patch as text.

CMD + V (paste)

```
{  
  "boxes": [  
    {  
      "box": {  
        "maxclass": "gain~",  
        "patching_rect":  
          [ 385.792366743087769,  
            357.923513054847717, 22.0, 140.0 ],  
        "outlettype": [ "signal", "" ],  
        "multichannelvariant": 0,  
        "id": "obj-114",  
        "parameter_enable": 0,  
        "numinlets": 1,  
        "numoutlets": 0  
      }  
    }  
  ]  
}
```



Using max / Saving

- MaxMSP is on all of the lab computers here and in the studios
- You can purchase a license either month by month or lifetime
- You can download MaxMSp for free, but you can't save. What you can do is copy the code and save it on your computer, GitHub, or email it to yourself and paste it back in later.

Handing in projects

-A saved file (File > Save as Project) .maxpat

A PDF that contains project documentation. See [Project_Documentation_Template](#) on Moodle. It contains:

- A clear, strong image of your project.
- A screen capture of your max patch
- A link to video or audio recording of your project working (can be a video or a link to a private video on YouTube, Vimeo or Google)
- A 50-100 word explanation of your project inside your patch Be sure to name files properly (no untitled-1.zip) All files are expected to be cleaned up and arranged in a reasonable, legible way. Videos should be clear, well light and show your project working.

Be sure to name files properly (no untitled-1, no screen shot 2343242342134)

All files are expected to be cleaned up and arranged in a reasonable, legible way. Videos should be clear, well light and show your project working.

Homework:

Start thinking about Audio Experiment

Open every file in Max Basics, read all of the comments and play with the patch. Try to recreate them yourself!