



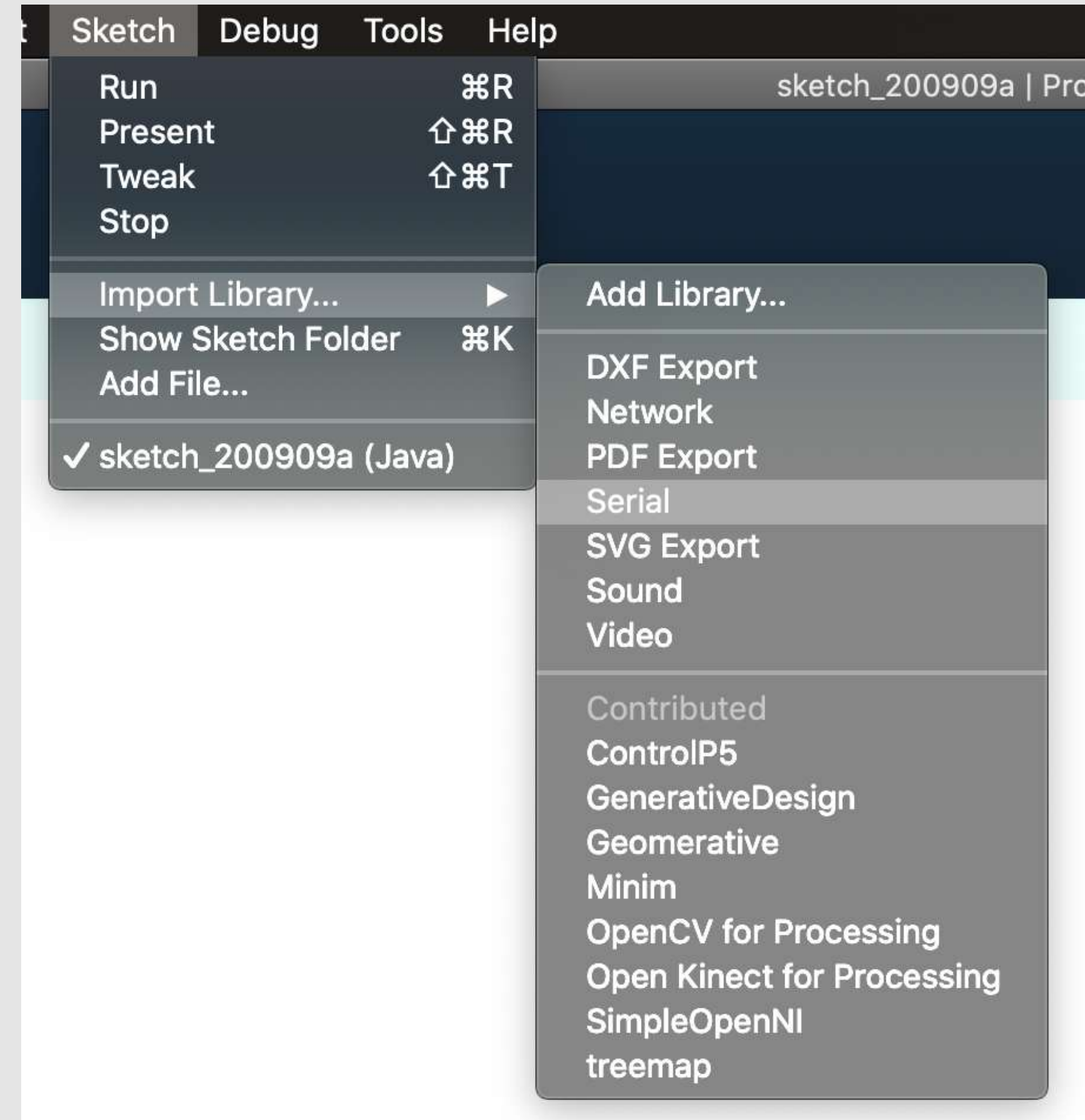
Processing Tutorial 4

Serial, Interactions, and more

Serial

Receive or send information between devices or programs through serial port

Use Serial library



Serial Read (Processing code)

Reading serial data from other source

Import Serial library

Creating object

Create a list of available serial ports. Mine is the 3rd in list so it's [2] following array logic. Then, initialise the serial port with 9600 baud rate (for communicating with other devices or software)

If serial info is sent, val will become the value of the sent data (be sure to check if your data is int, char, or string)

```
import processing.serial.*;
```

```
Serial myPort; // Create object from Serial class  
int val;       // Data received from the serial port
```

```
void setup()  
{  
  size(200, 200);  
  // I know that the first port in the serial list on my mac  
  // is always my FTDI adaptor, so I open Serial.list()[0].  
  // On Windows machines, this generally opens COM1.  
  // Open whatever port is the one you're using.  
  String portName = Serial.list()[2];  
  myPort = new Serial(this, portName, 9600);  
}
```

```
void draw()  
{  
  if ( myPort.available() > 0) { // If data is available,  
    val = myPort.read();         // read it and store it in val  
  }  
  background(255);              // Set background to white  
  fill(val);  
  
  rect(50, 50, 100, 100);  
  delay(100);  
}
```

Serial Read (Arduino code)

Reading serial data from other source

Set baud rate to 9600 to match with Processing baud rate

The code on Arduino is simple. You just need to send the data through Serial. In this case, the data is from a sensor attached on the Arduino

```
int a;

void setup() {
  Serial.begin(9600);
  pinMode(6, OUTPUT);
}

void loop()
{ digitalWrite(6, HIGH);
  delay(100);
  a = analogRead(A1);
  Serial.write(a);
}
```

Serial Write (Processing code)

Writing serial data to other source

When writing, you use the same setup code. But under void draw, you just need to have Processing write serial data when an action is made inside Processing.

```
void draw() {  
  background(255);  
  if (mouseOverRect() == true) { // If mouse is over square,  
    fill(204); // change color and  
    myPort.write('H'); // send an H to indicate mouse is over square  
  }  
  else { // If mouse is not over square,  
    fill(0); // change color and  
    myPort.write('L'); // send an L otherwise  
  }  
  rect(50, 50, 100, 100); // Draw a square  
}
```

Serial Write (Arduino code)

Writing serial data to other source

Set baud rate to 9600 to match with Processing baud rate

Receive data once there is serial data available for Arduino to read

If the Arduino receives the char 'H', the servo moves

```
#include <Servo.h>

Servo myservo;

char val;
int pos = 0;

void setup() {
  myservo.attach(9);
  Serial.begin(9600); // Start serial communication at 9600 bps
}

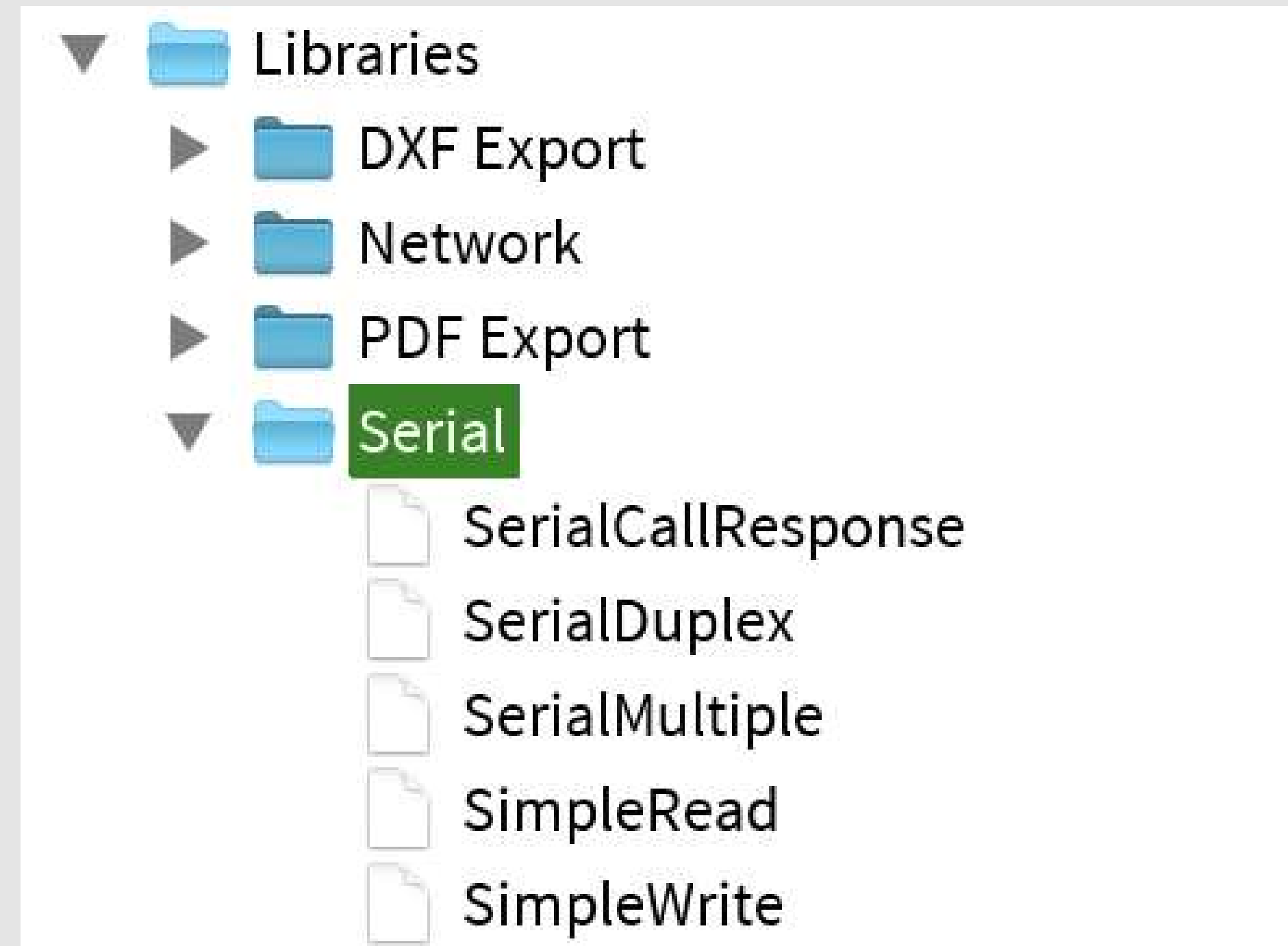
void loop() {
  while (Serial.available()) { // If data is available to read,
    val = Serial.read(); // read it and store it in val
  }

  myservo.write(pos);

  if (val == 'H') { // If H was received
    pos+=2;
    if (pos >= 180) {
      pos = 180;
    }
  } else {
    pos-=2;
    if (pos <= 0) {
      pos = 0;
    }
  }

  delay(100); // Wait 100 milliseconds for next reading
}
```

Serial Examples



- Examples to try on your own:**
- **Libraries > Serial > SimpleRead**
 - **Libraries > Serial > SimpleWrite**
 - **Libraries > Serial > SerialMultiple**

**You can use Serial to trigger many different things!
The possibilities are endless**

Interactions: A brief look

We have went through `mousePressed()` and `keyPressed()`

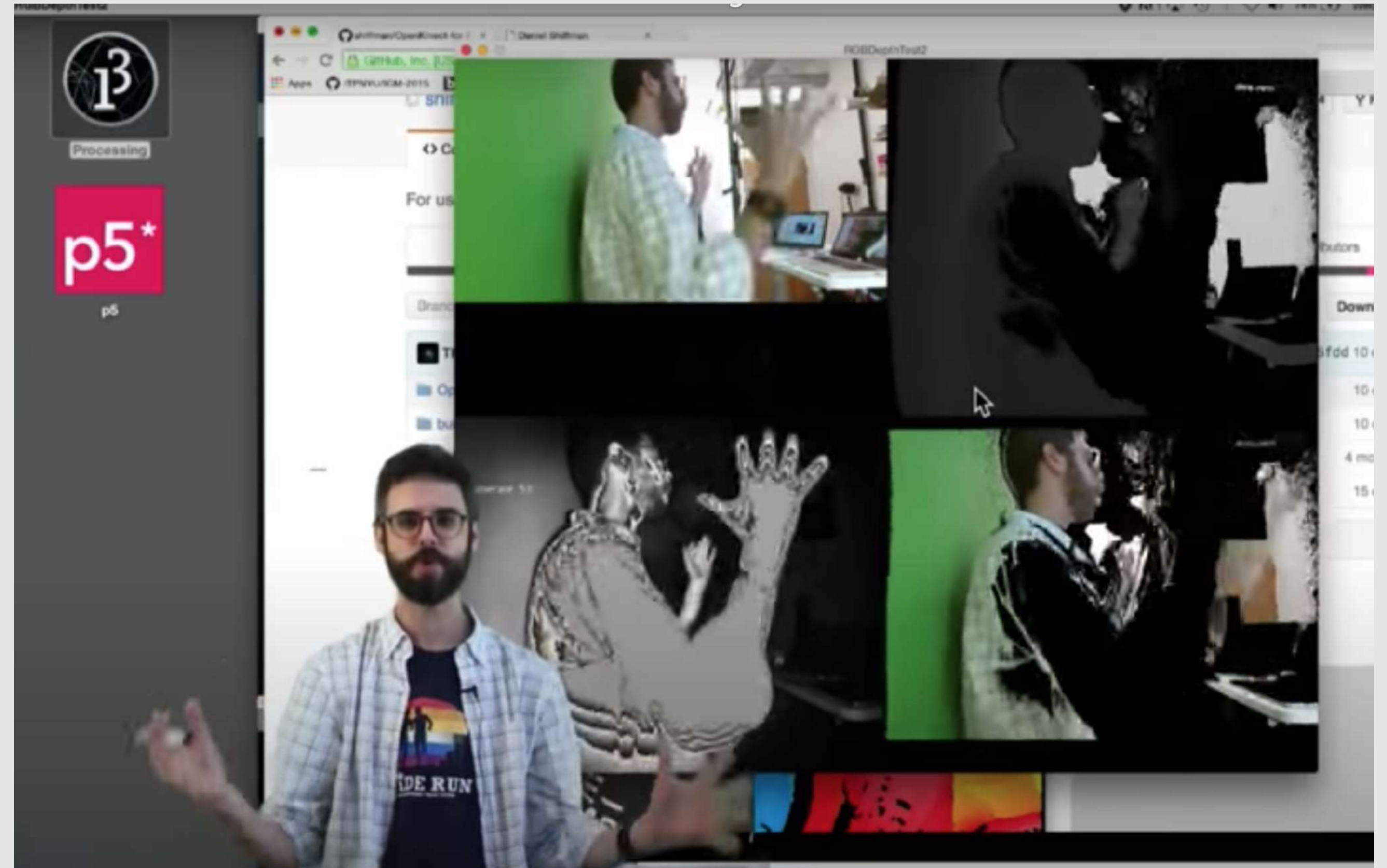
Other methods of interaction:

Serial connection

- Arduino + sensors
- Softwares like TouchDesigner, MaxMSP, etc

Other libraries

- Kinect
- OpenCV
- Leap Motion



Other Cool Examples

Contributed Libraries > OpenCV > FaceDetection
Topics > Advanced Data > ArrayListClass
Topics > Cellular Automata > GameOfLife
Topics > Fractals and L-Systems > PenroseSnowflake
Topics > Interaction > Reach2
Topics > Interation > Reflection2
Topics > Simulate > Flocking
Demo > Graphics > Yellowtail

Code Base from Generative Design Book

In OSS, you will find a link to the code base to try out the different sketches in the book. Use this if you are curious about using Processing to create generative art.

You need to first install the library called Generative Design by Hartmut Bohnacker and Benedikt Gross

Status	Name ▲	Author
	ComposingForEveryone ComposingForEveryone gives support f...	Guido Kramann
✓	GenerativeDesign A collection of various functions belonging to t...	Hartmut Bohnacker, Benedikt Gross
✓	Geomerative Extends 2D geometry operations to facilitate gener...	Ricard Marxer
	RiTa A library for experiments in generative natural language.	Daniel C. Howe
	ToxicLibs toxiclibs is an independent, open source library collect...	Karsten Schmidt
	XYscope XYScope is a library for Processing to render graphics on...	Ted Davis

Resources:

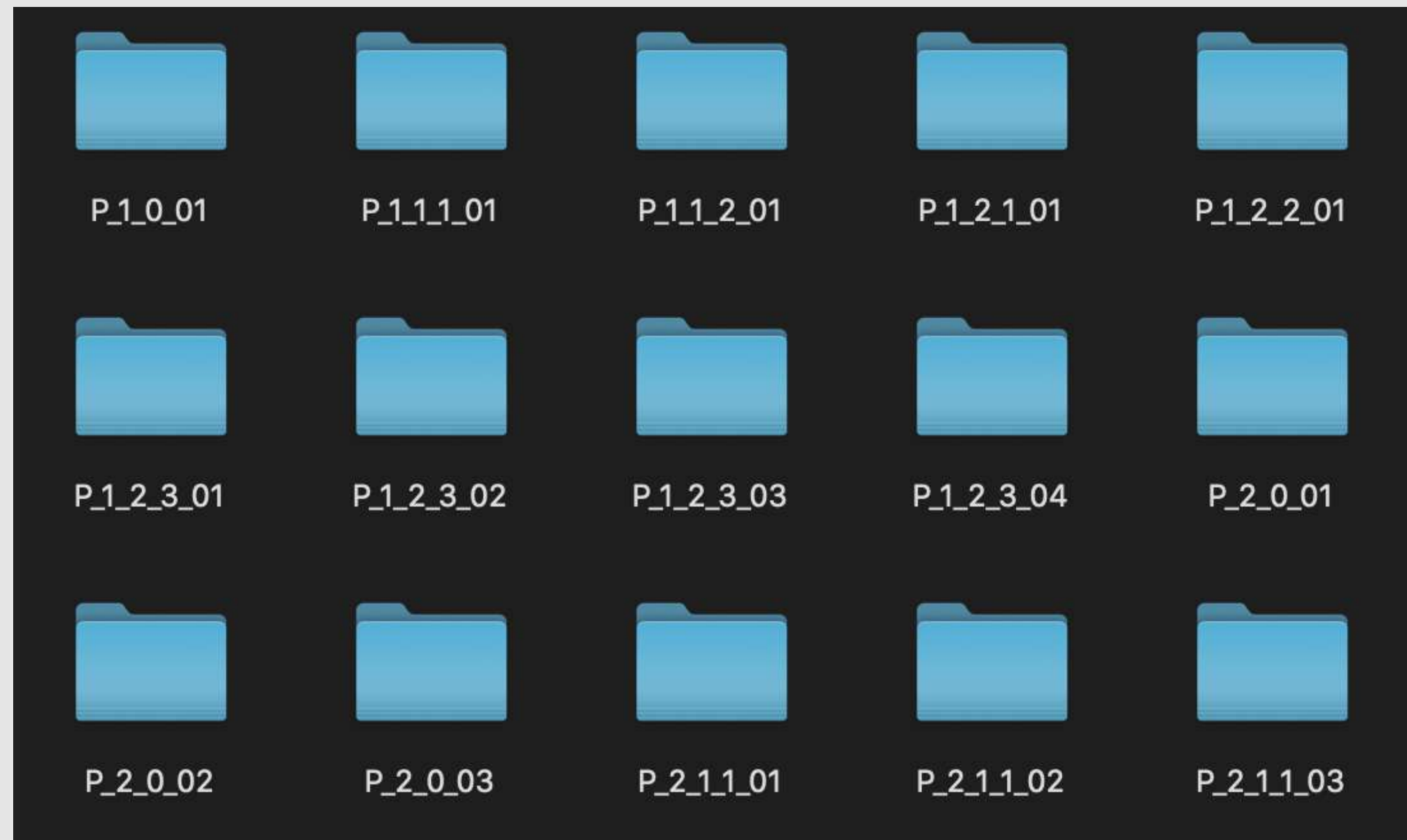
<https://onformative.com/work/book-generative-gestaltung>

www.generative-gestaltung.de (does not work)

Code Base from Generative Design Book

Use "Code Package - Processing 2" to begin

The folders are arranged by the sections within the book as seen from the image on the right. I have attached the content of the book in the zip file so you can also check what you are looking at.



Other Resources

kinect with Daniel Shiffman

https://www.youtube.com/watch?v=QmVNgdapJJM&ab_channel=TheCodingTrain

OpenCV Blob detection with Daniel Shiffman

https://www.youtube.com/watch?v=h8tk0hmWB44&list=PLRqwX-V7Uu6bw0bVn4M63p8TMJf30hGy8&index=4&ab_channel=TheCodingTrain

Thank you!