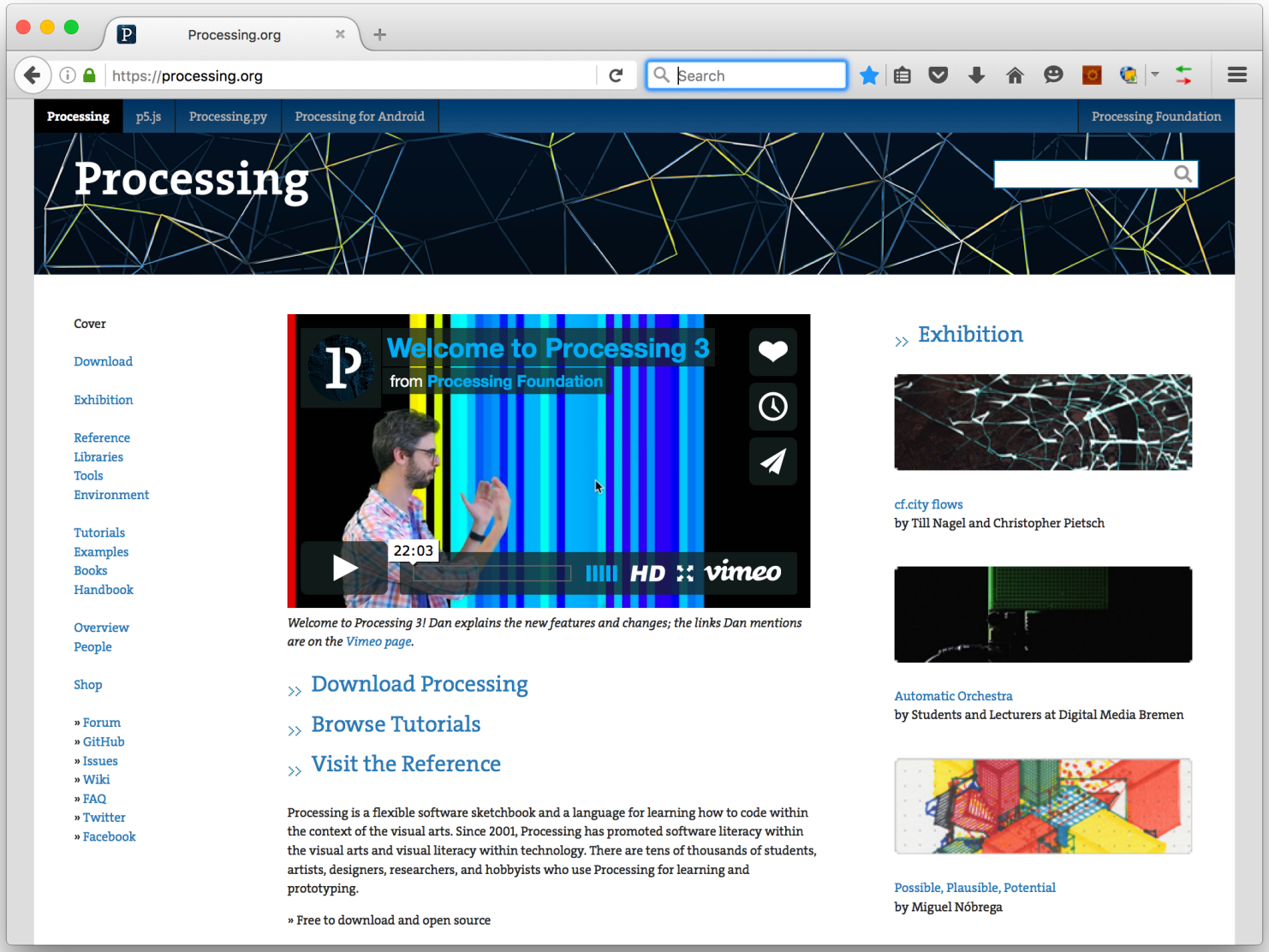
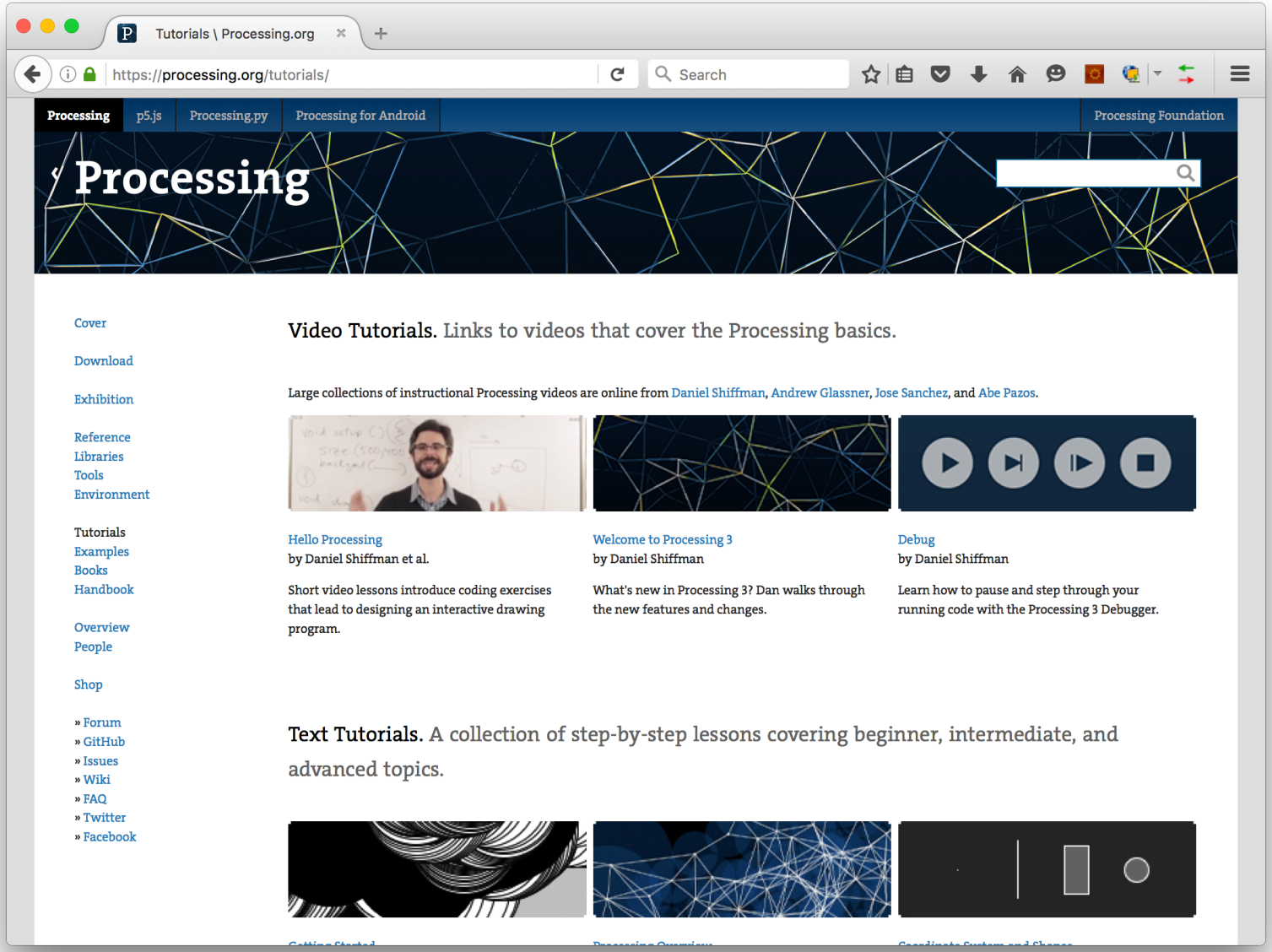
Programming in Java

with Processing

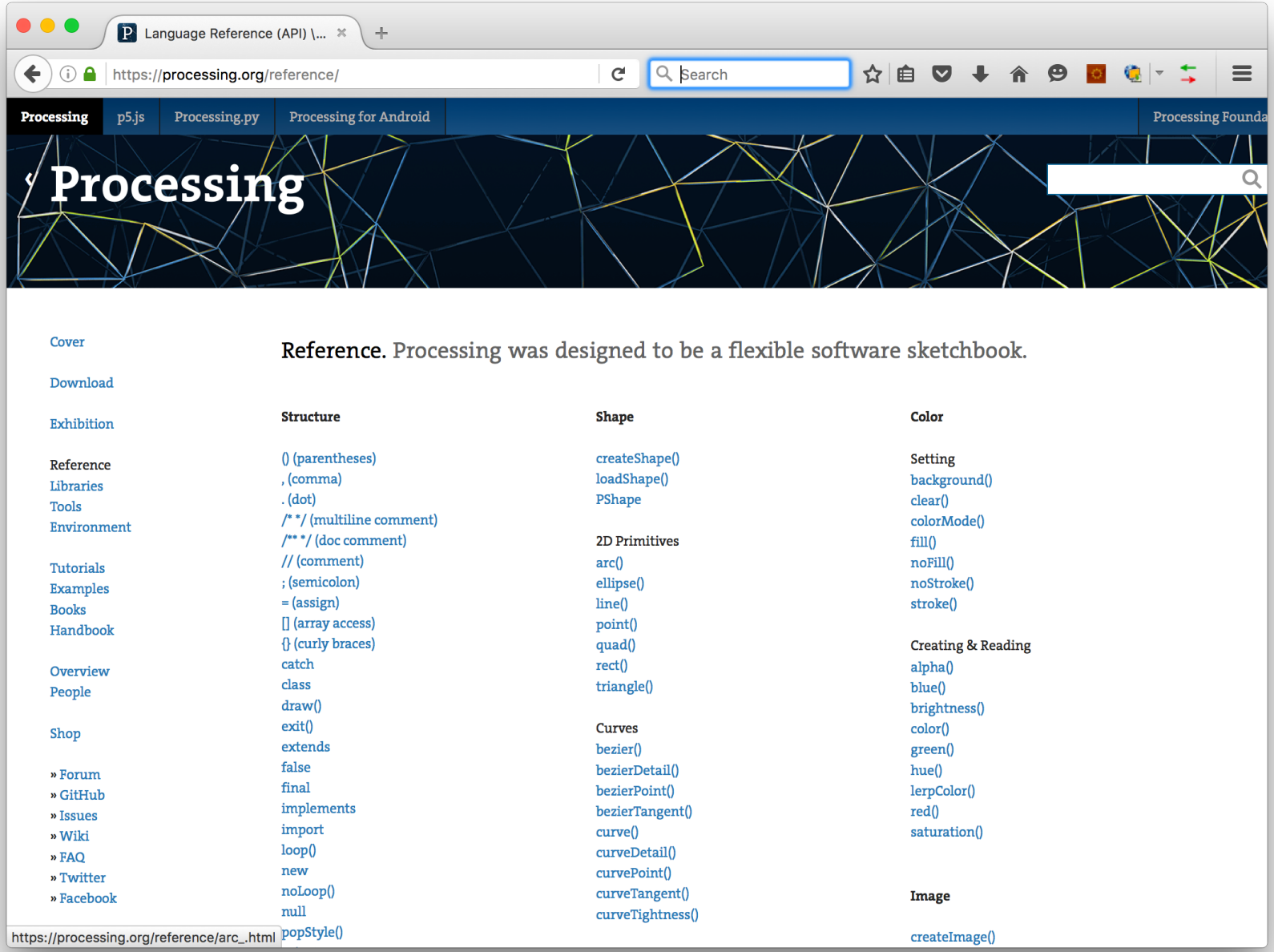
https://processing.org/



https://processing.org/tutorials/



https://processing.org/reference/



Name

println()

Examples

```
String s = "The size is ";
int w = 1920;
int h = 1080;
println(s);
println(w, "x", h);

// This program writes to the console:
// The size is
// 1920 x 1080
```

```
print("begin- ");
float f = 0.3;
int i = 1024;
print("f is " + f + " and i is " + 1024);
String s = " -end";
println(s);

// This program writes to the console:
// "begin- f is 0.3 and i is 1024 -end"
```

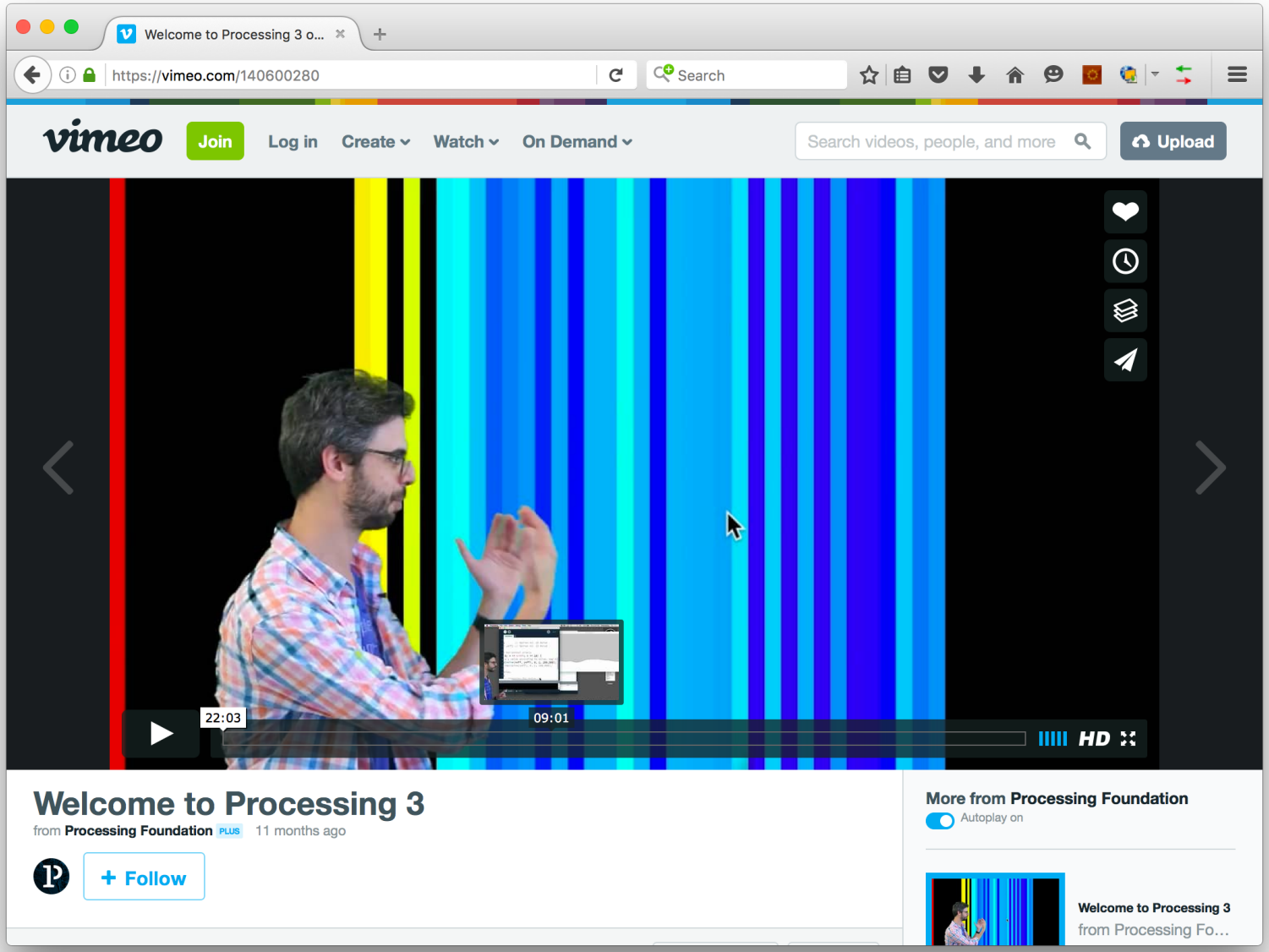
Description

The `println()` function writes to the console area, the black rectangle at the bottom of the Processing environment. This function is often helpful for looking at the data a program is producing. Each call to this function creates a new line of output. More than one parameter can be passed into the function by separating them with commas. Alternatively, individual elements can be separated with quotes ("") and joined with the addition operator (+).

Before Processing 2.1, `println()` was used to write array data to the console. Now, use `printArray()` to write array data to the console.

https://processing.org/reference/println_.html

https://vimeo.com/140600280



» Forum
» GitHub
» Issues
» Wiki
» FAQ
» Twitter
» Facebook

Text Tutorials. A collection of step-by-step lessons covering beginner, intermediate, and advanced topics.

Getting Started
by Casey Reas and Ben Fry

Welcome to Processing! This introduction covers the basics of writing Processing code.

Level: Beginner

Processing Overview
by Ben Fry and Casey Reas

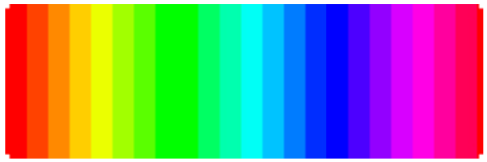
A little more detailed introduction to the different features of Processing than the Getting Started tutorial.

Level: Beginner

Coordinate System and Shapes
by Daniel Shiffman

Drawing simple shapes and using the coordinate system.

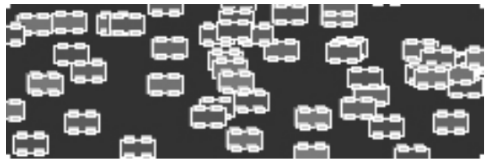
Level: Beginner



Color
by Daniel Shiffman

An introduction to digital color.

Level: Beginner



Objects
by Daniel Shiffman

The basics of object-oriented programming.

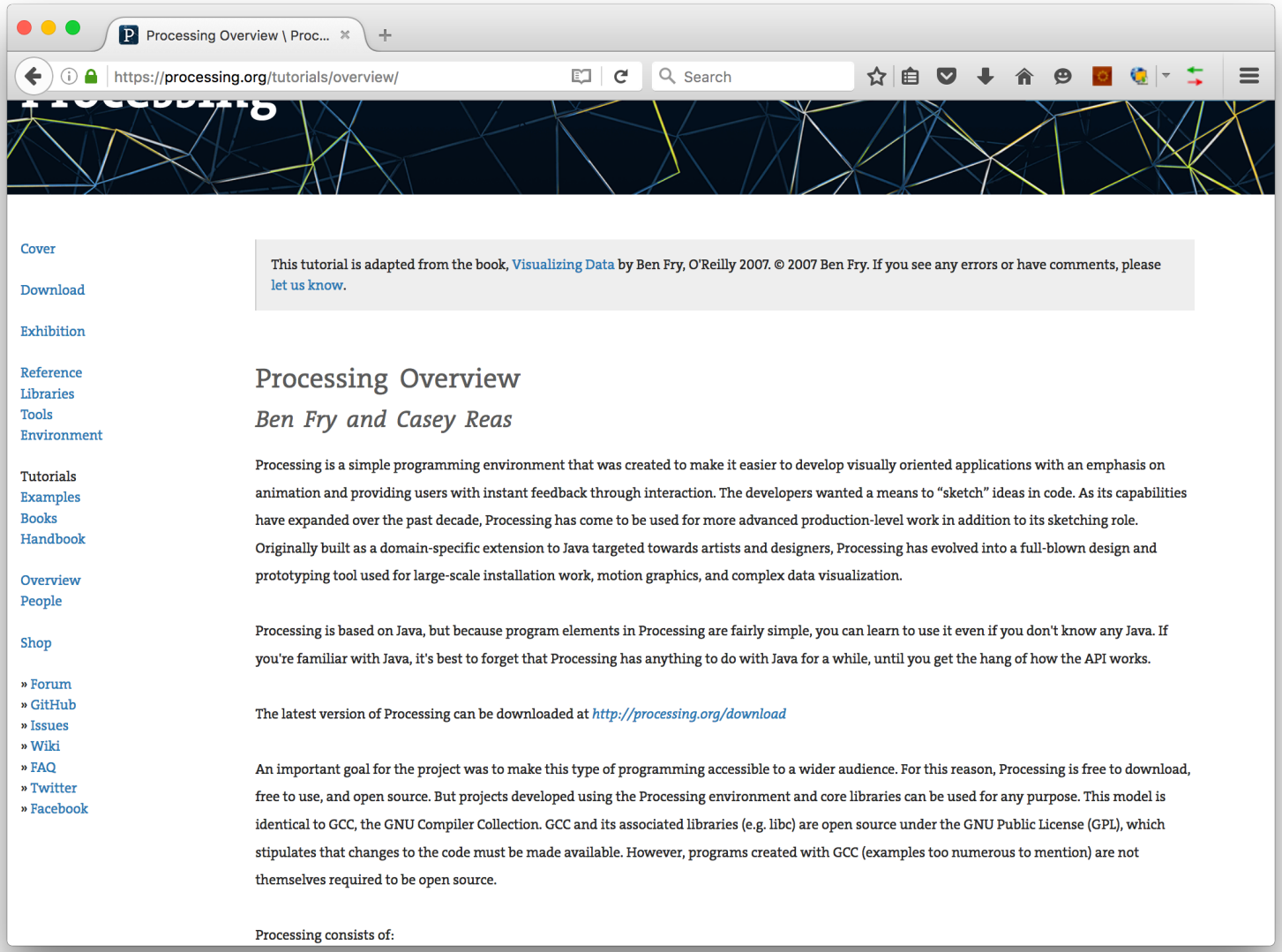
Level: Beginner

Interactivity
by Casey Reas and Ben Fry

Introduction to interactivity with the mouse and keyboard.

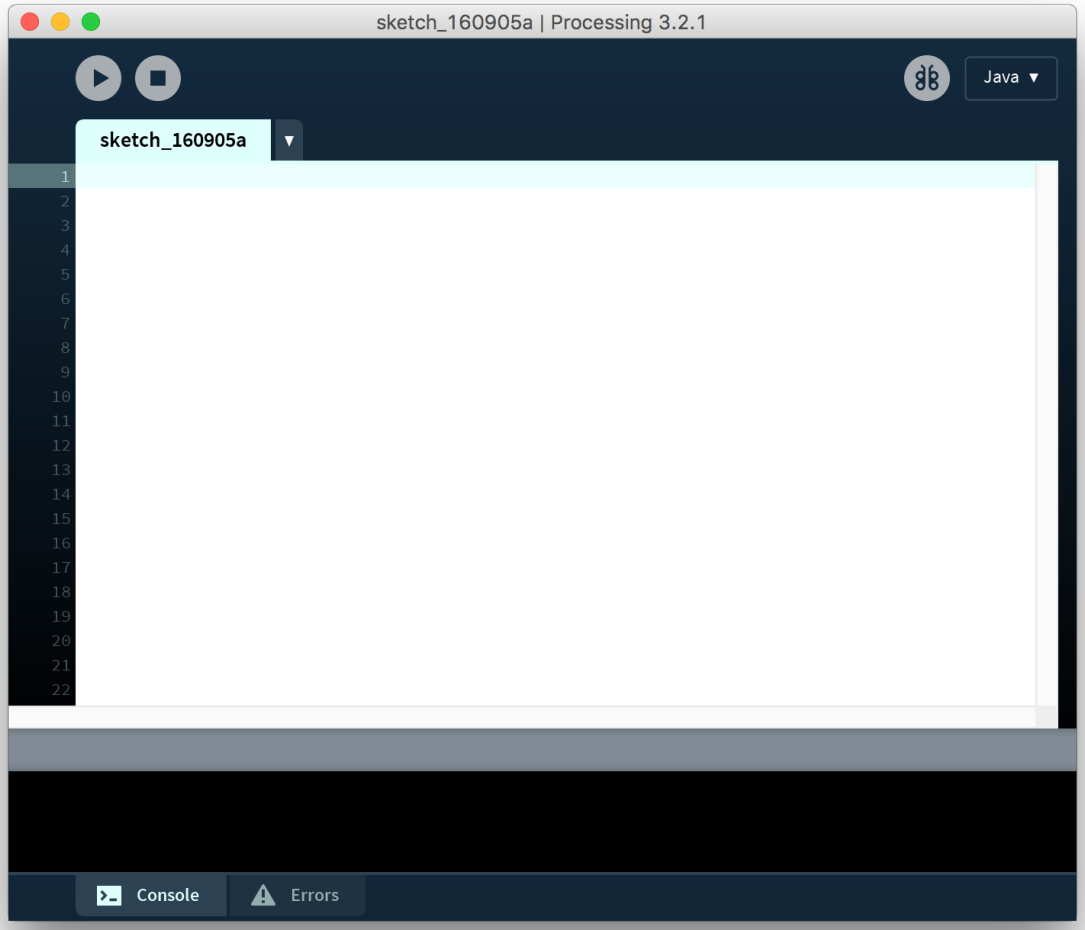
Level: Beginner

Processing Overview

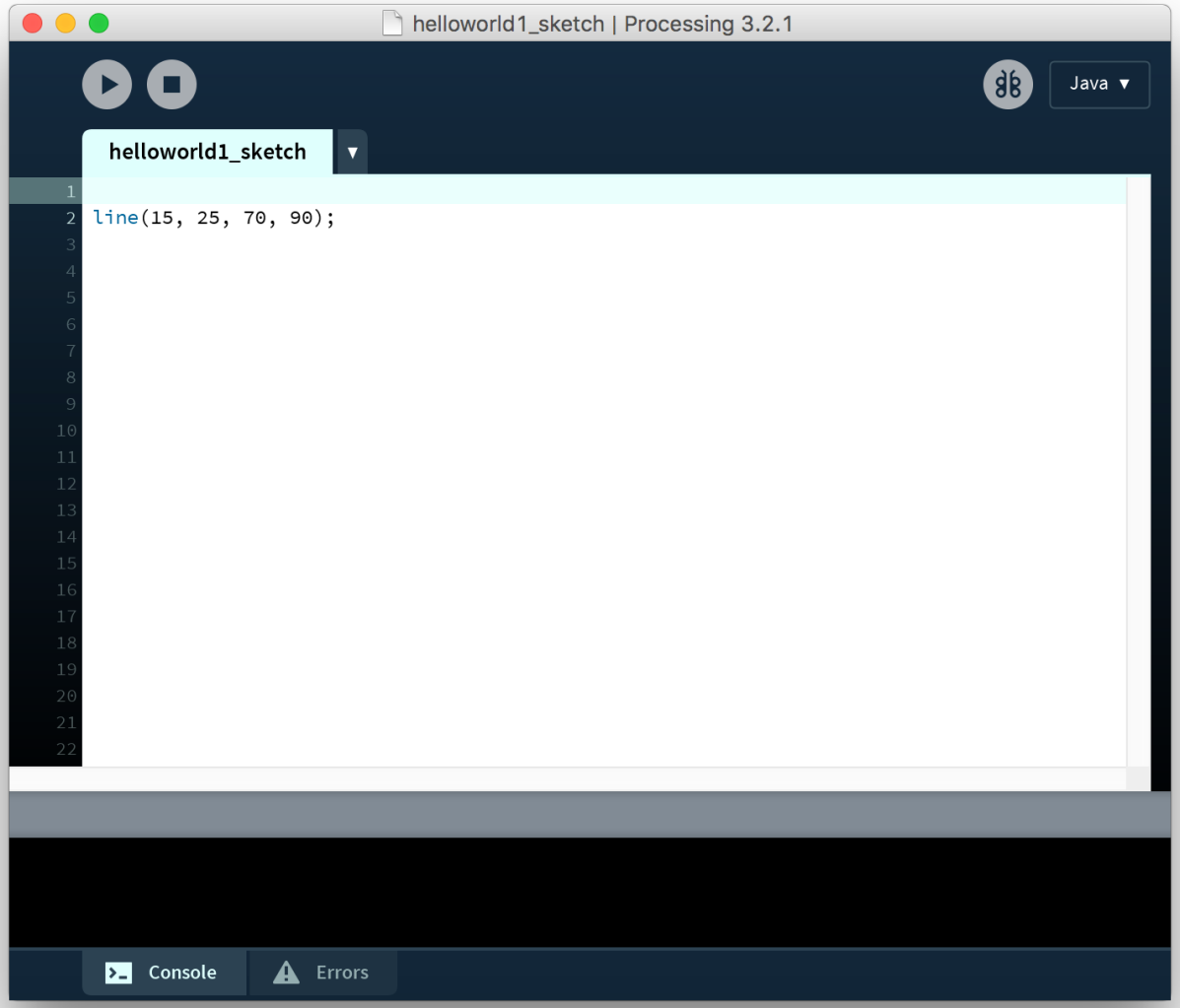


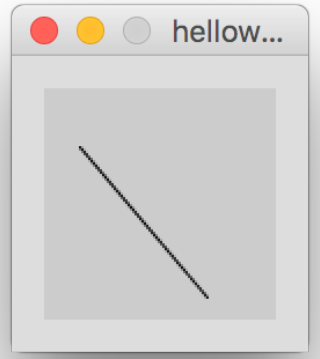
https://processing.org/tutorials/overview/

Processing 3 | Interactive IDE

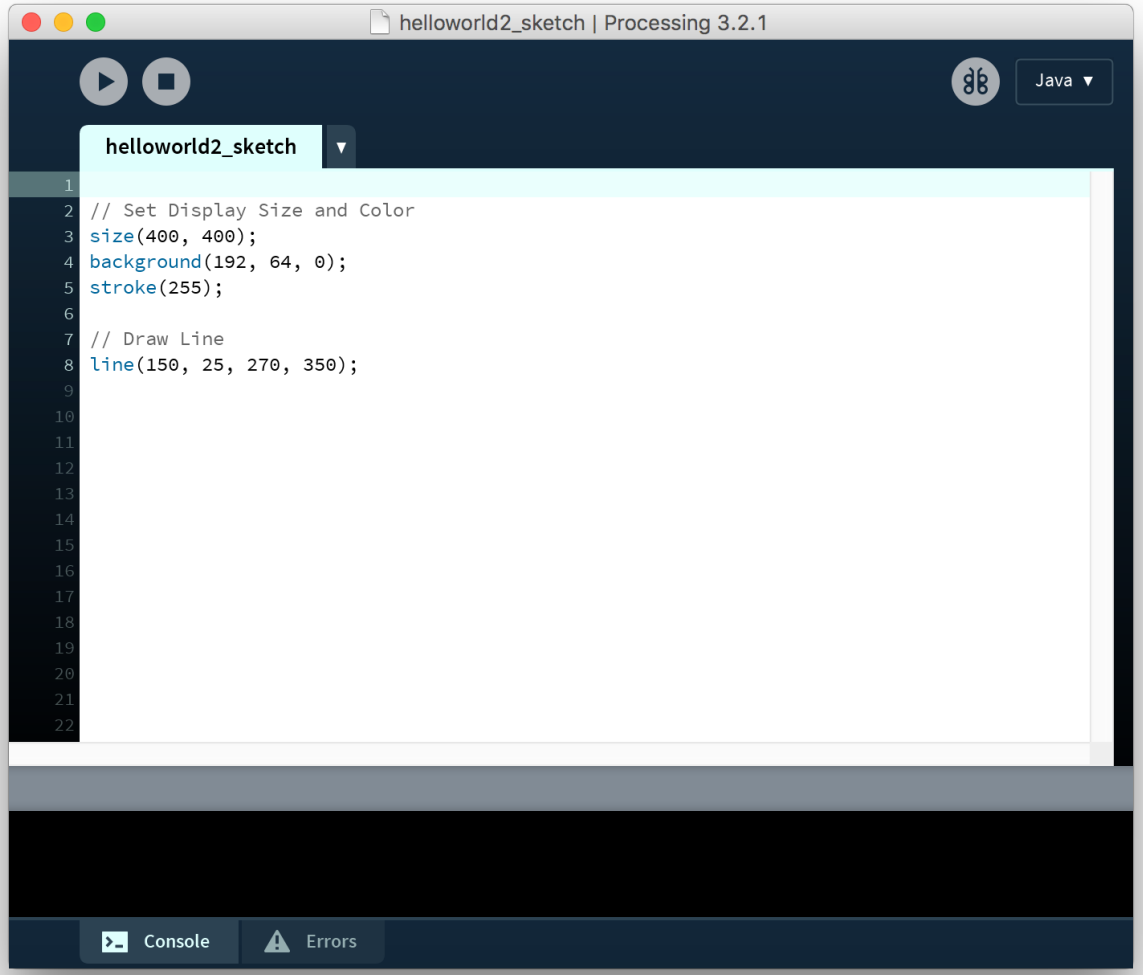


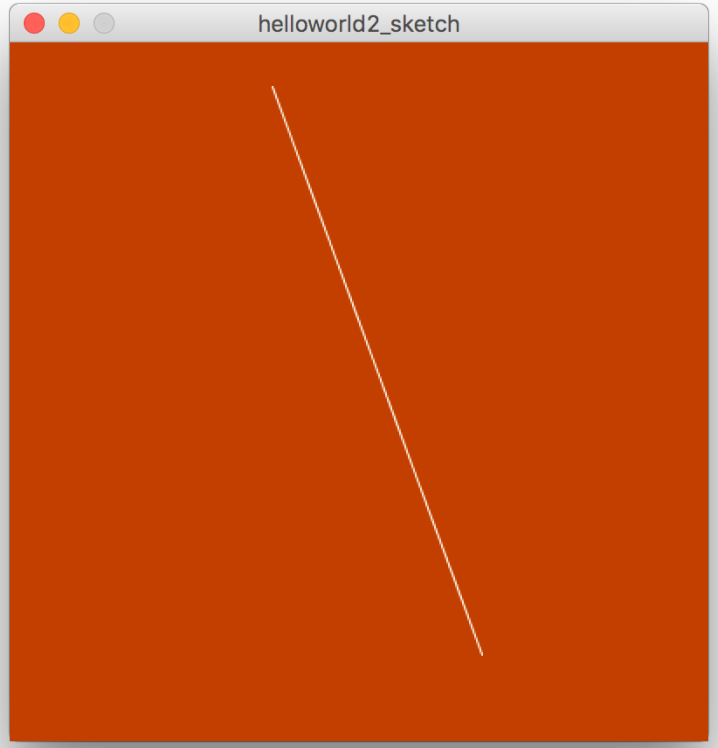
Simple Hello World



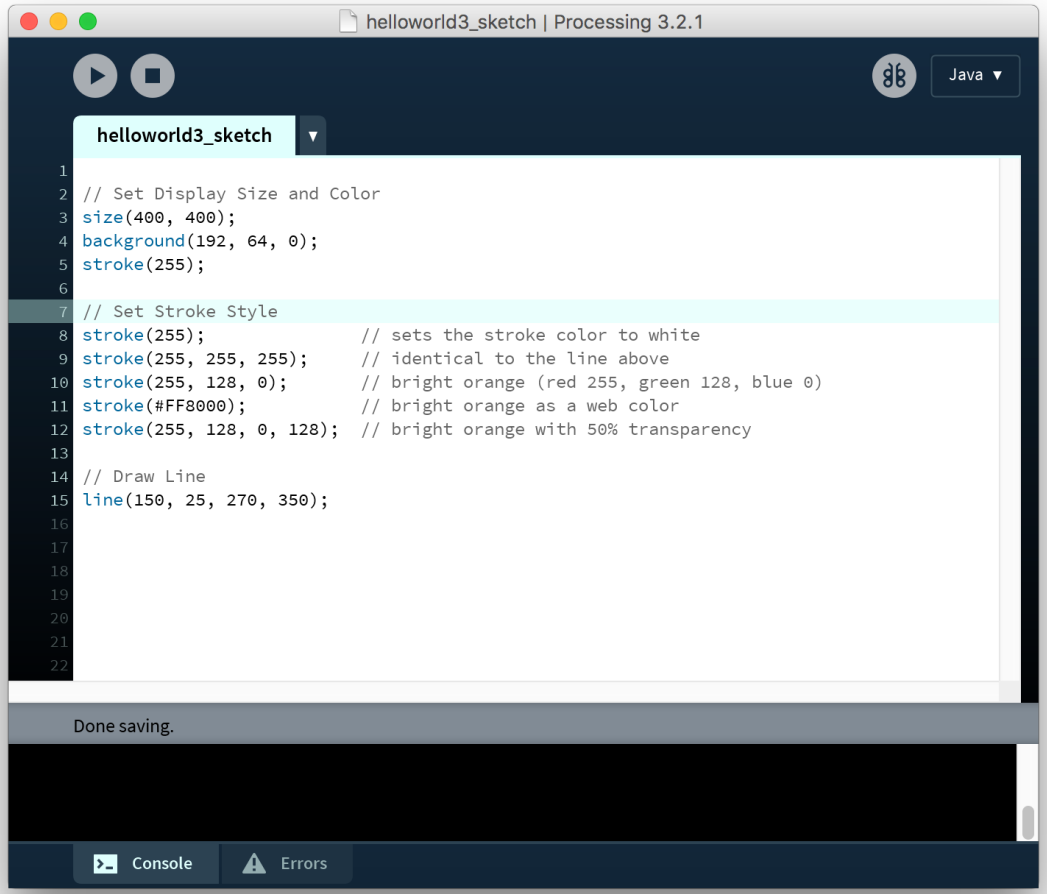


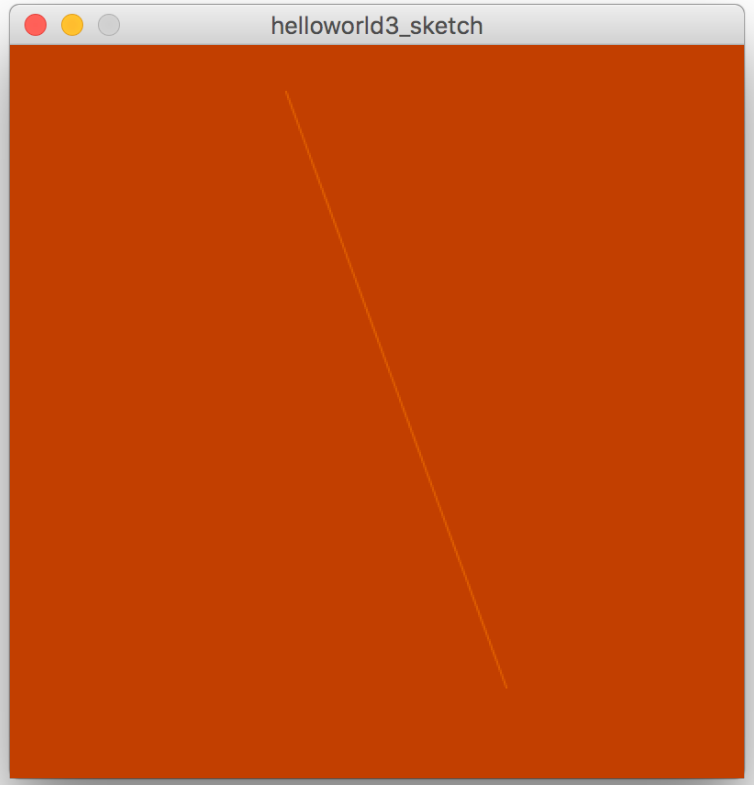
Another Hello World



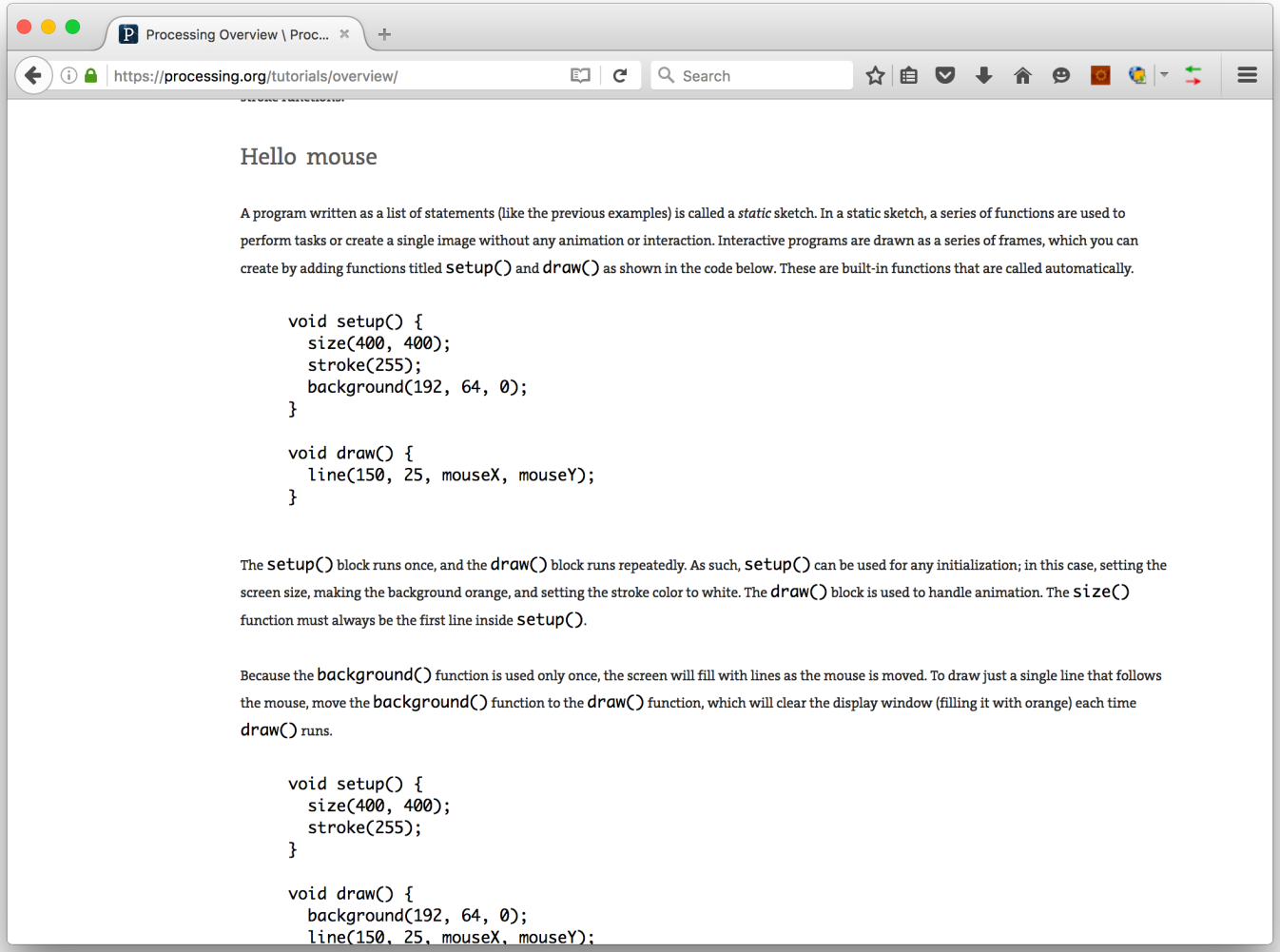


Add Some Style!



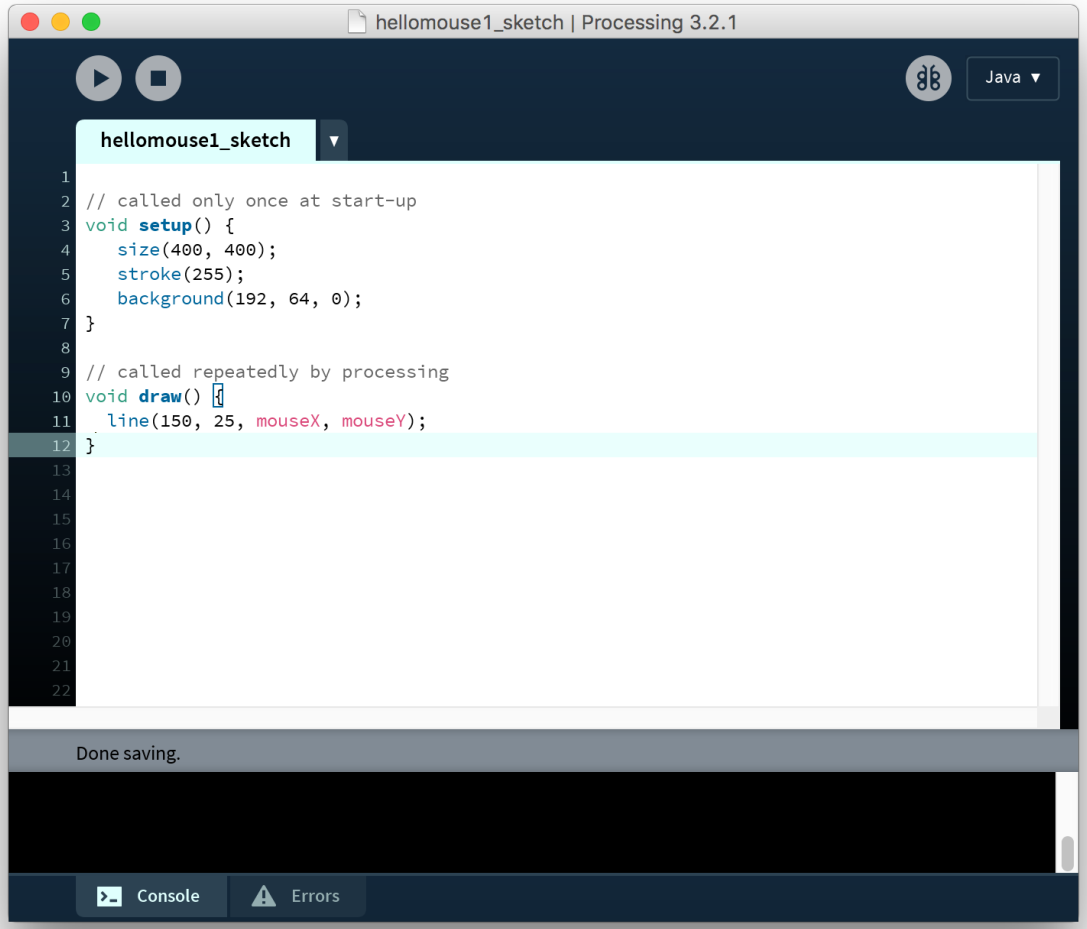


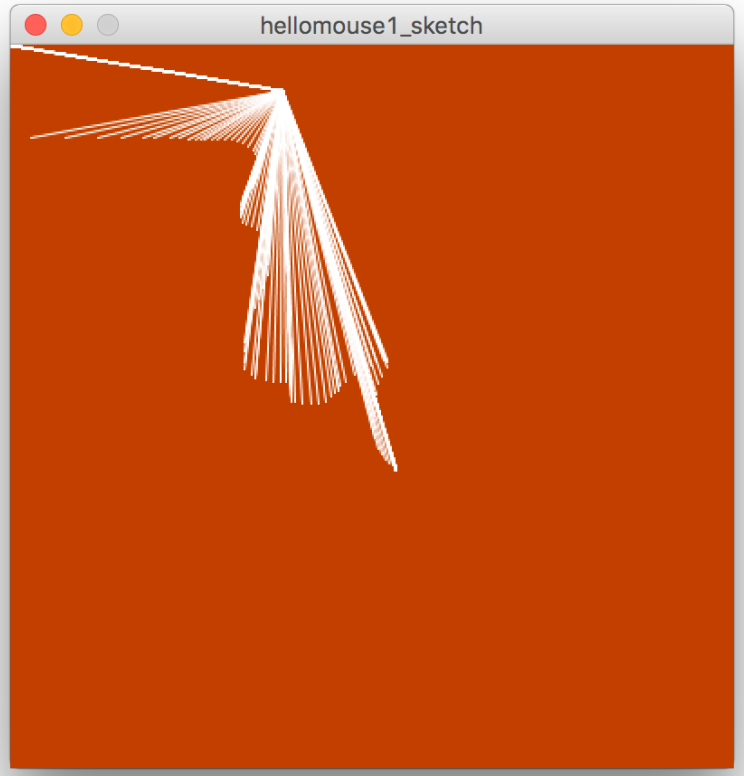
Hello Mouse



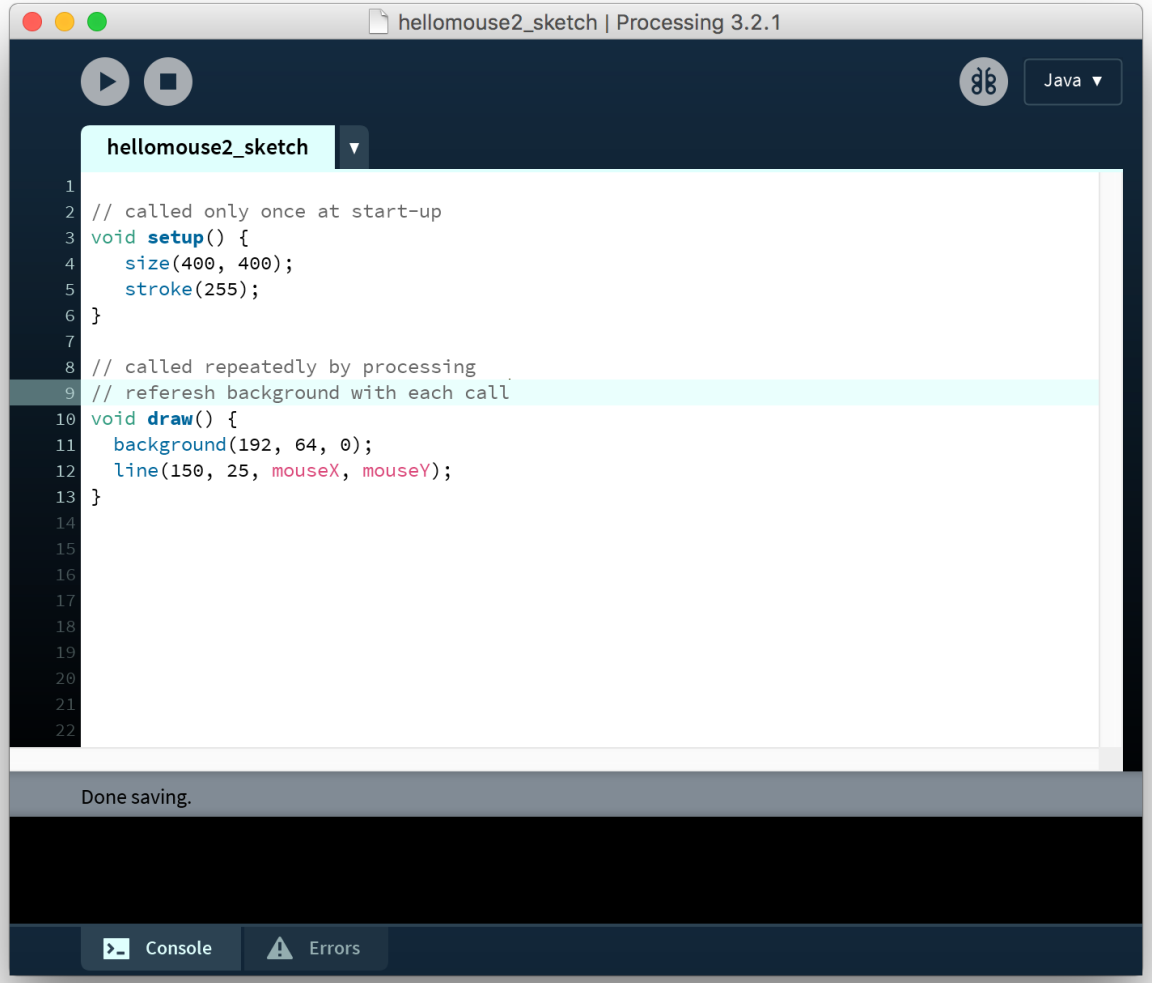
https://processing.org/tutorials/overview/

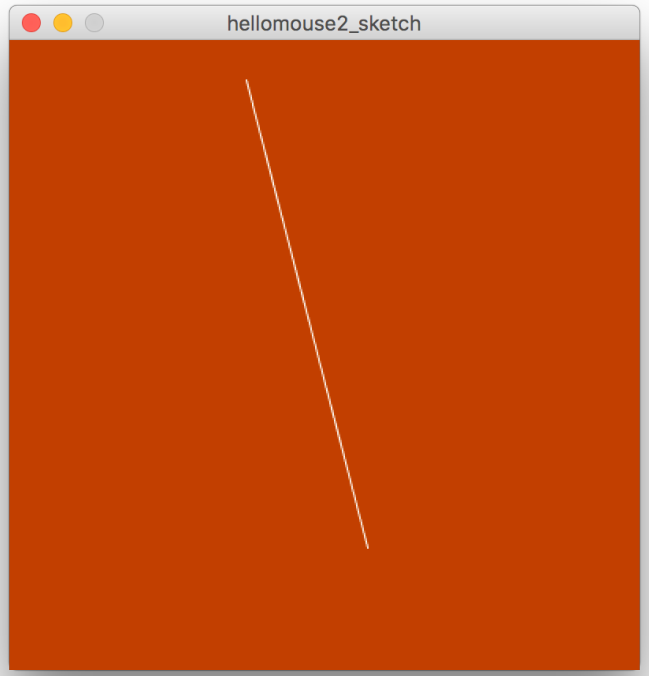
Setup and Draw



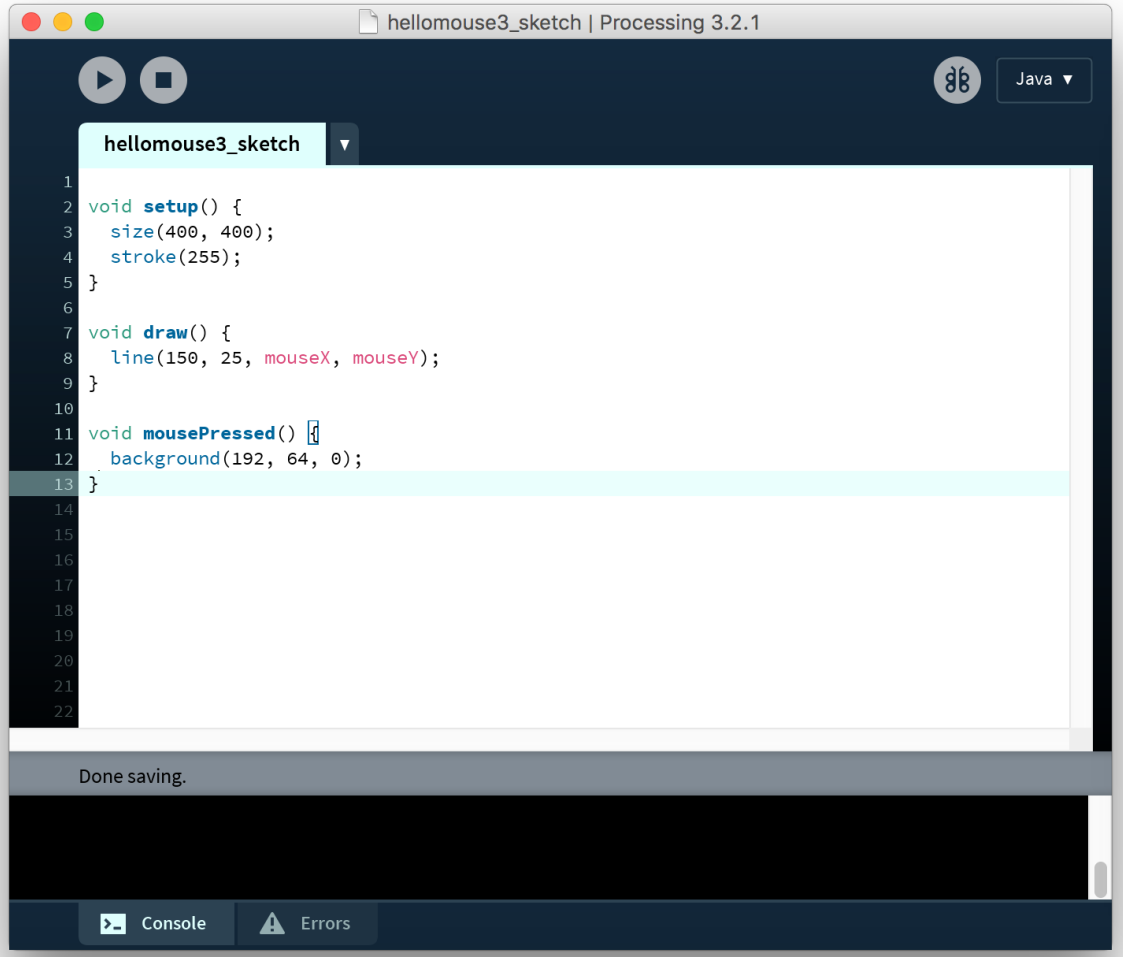


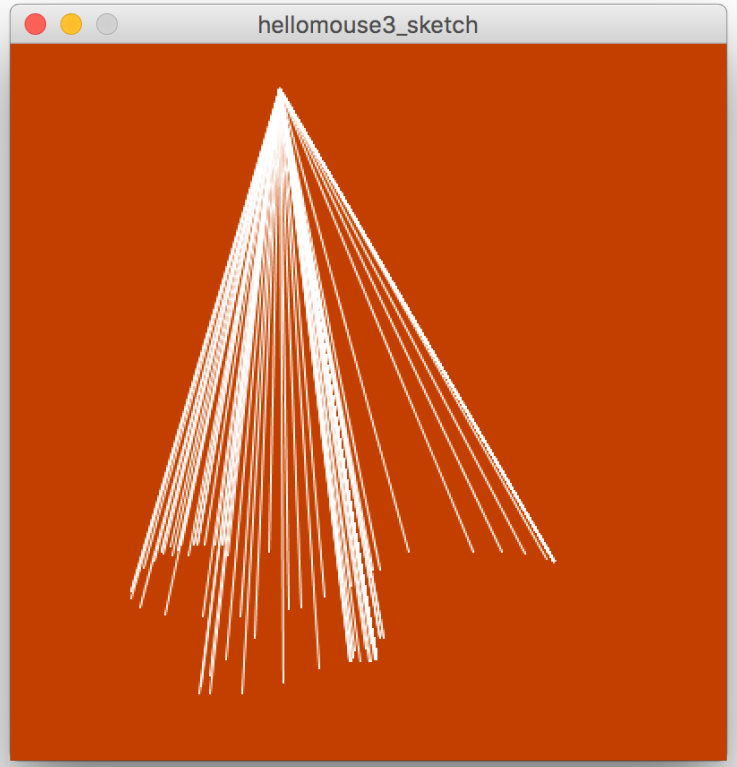
Following Line



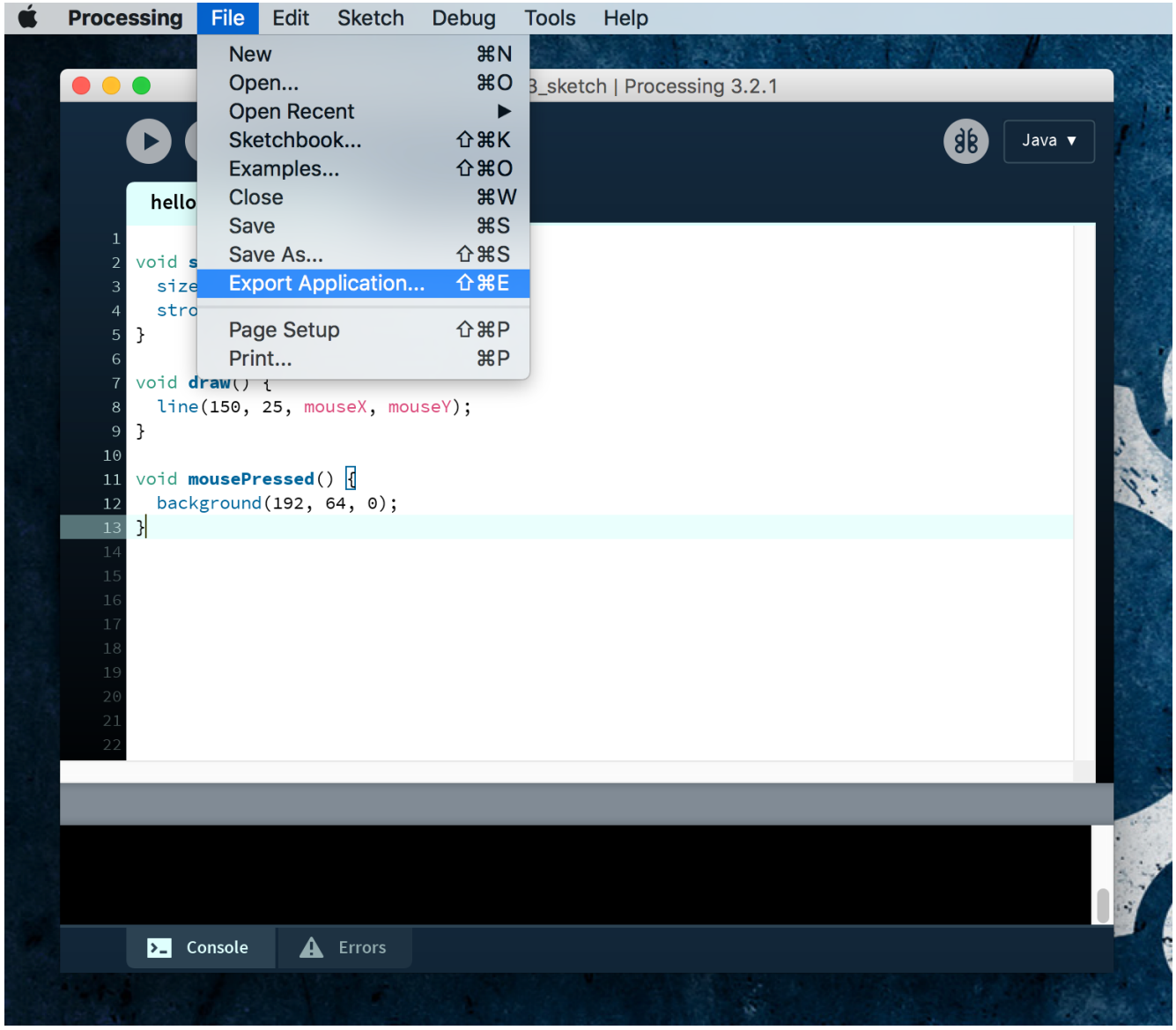


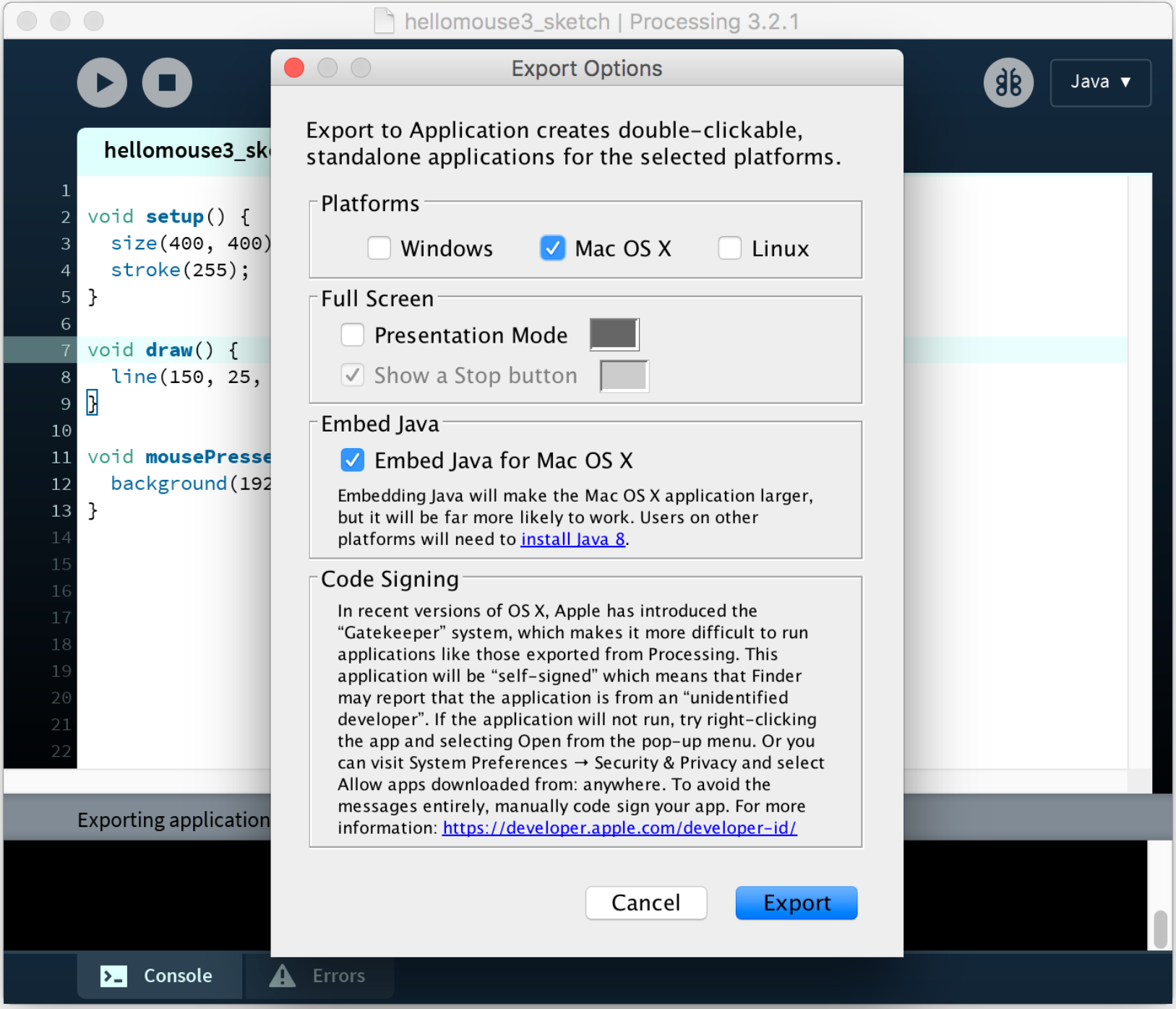
Using Mouse Press



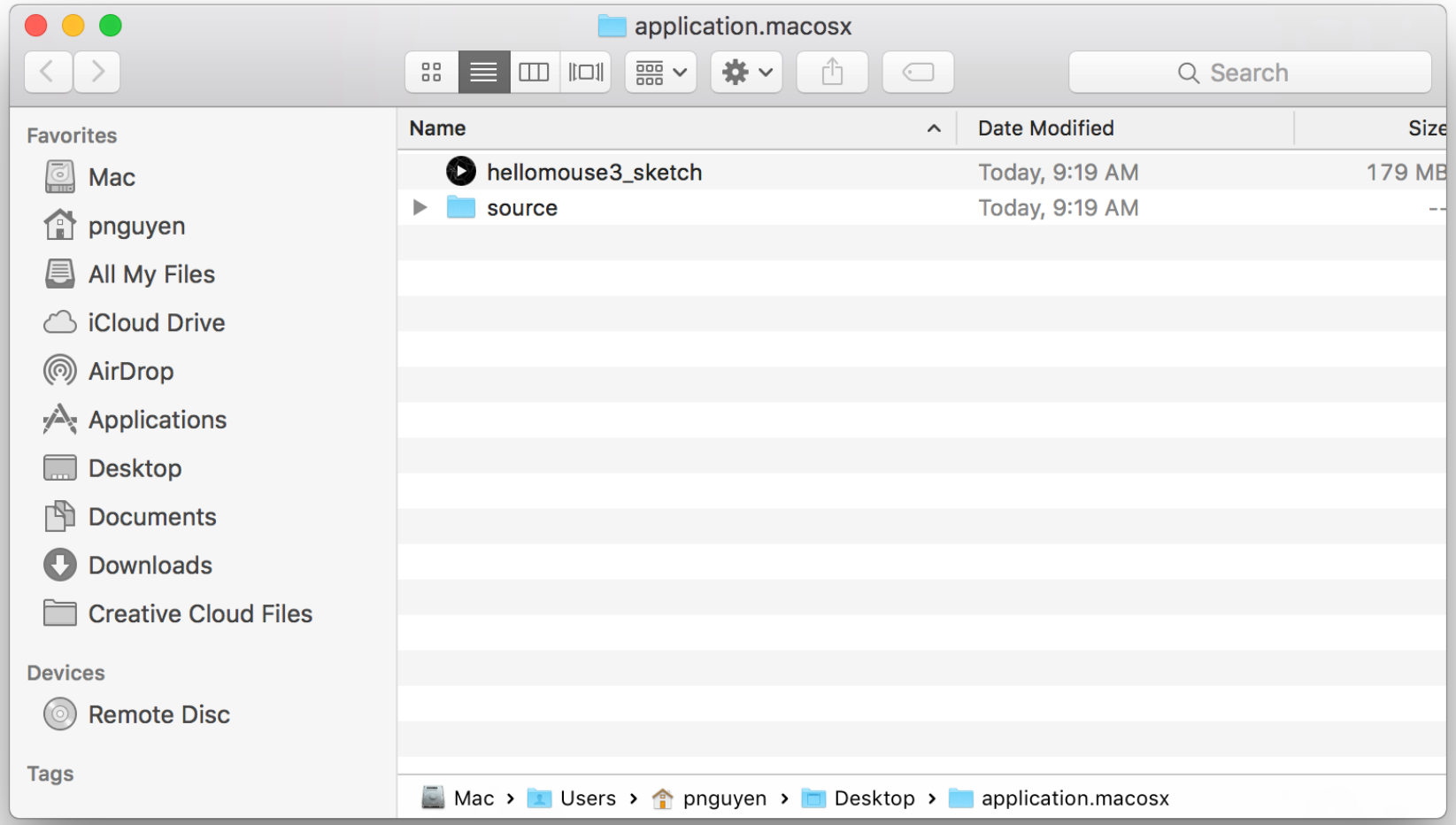


Exporting Application

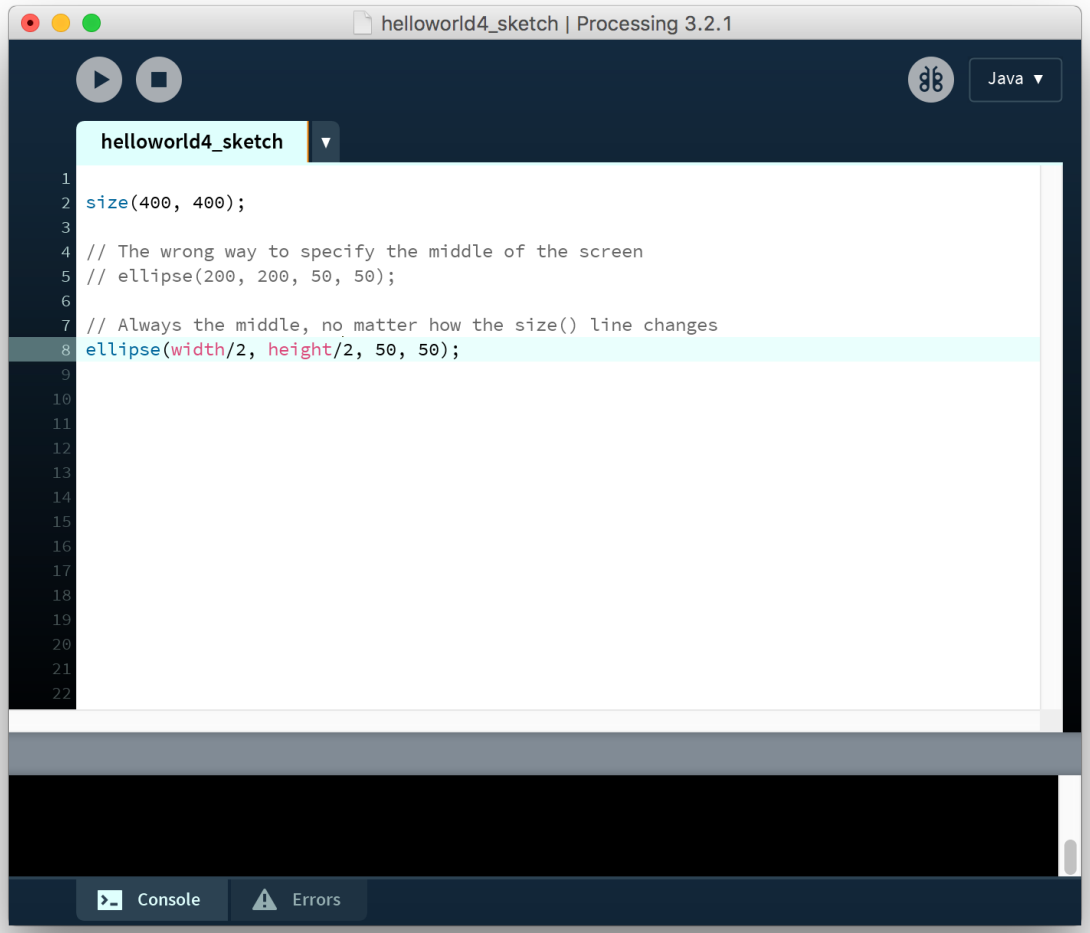


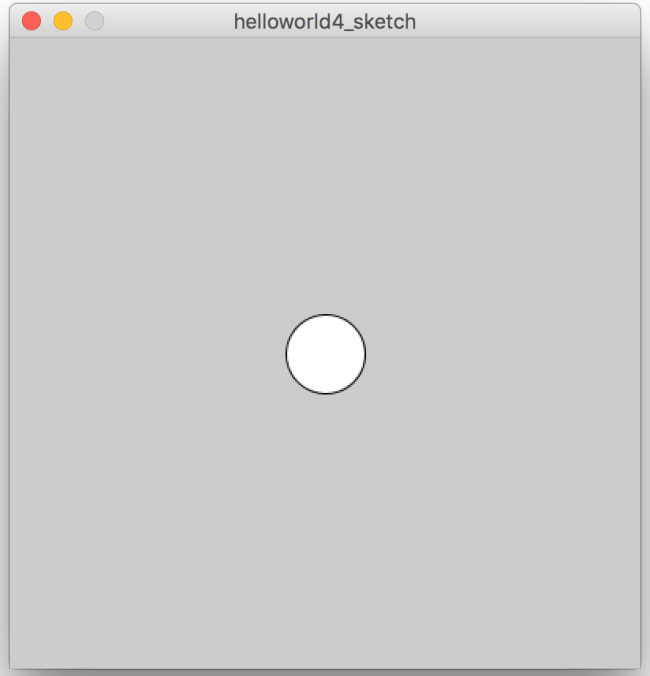

hellomouse3_sketch | Processing 3.2.1
Java
hellomouse3_sk
1 void setup() {
2 size(400, 400)
3 stroke(255);
4 }
5
6 void draw() {
7 line(150, 25,
8 }
9
10 void mousePresse
11 background(192
12 }
13
Exporting application
Console
Errors
Export Options
Export to Application creates double-clickable, standalone applications for the selected platforms.
Platforms
Windows
Mac OS X
Linux
Full Screen
Presentation Mode
Show a Stop button
Embed Java
Embed Java for Mac OS X
Embedding Java will make the Mac OS X application larger, but it will be far more likely to work. Users on other platforms will need to install Java 8.
Code Signing
In recent versions of OS X, Apple has introduced the "Gatekeeper" system, which makes it more difficult to run applications like those exported from Processing. This application will be "self-signed" which means that Finder may report that the application is from an "unidentified developer". If the application will not run, try right-clicking the app and selecting Open from the pop-up menu. Or you can visit System Preferences → Security & Privacy and select Allow apps downloaded from: anywhere. To avoid the messages entirely, manually code sign your app. For more information: https://developer.apple.com/developer-id/
Cancel
Export

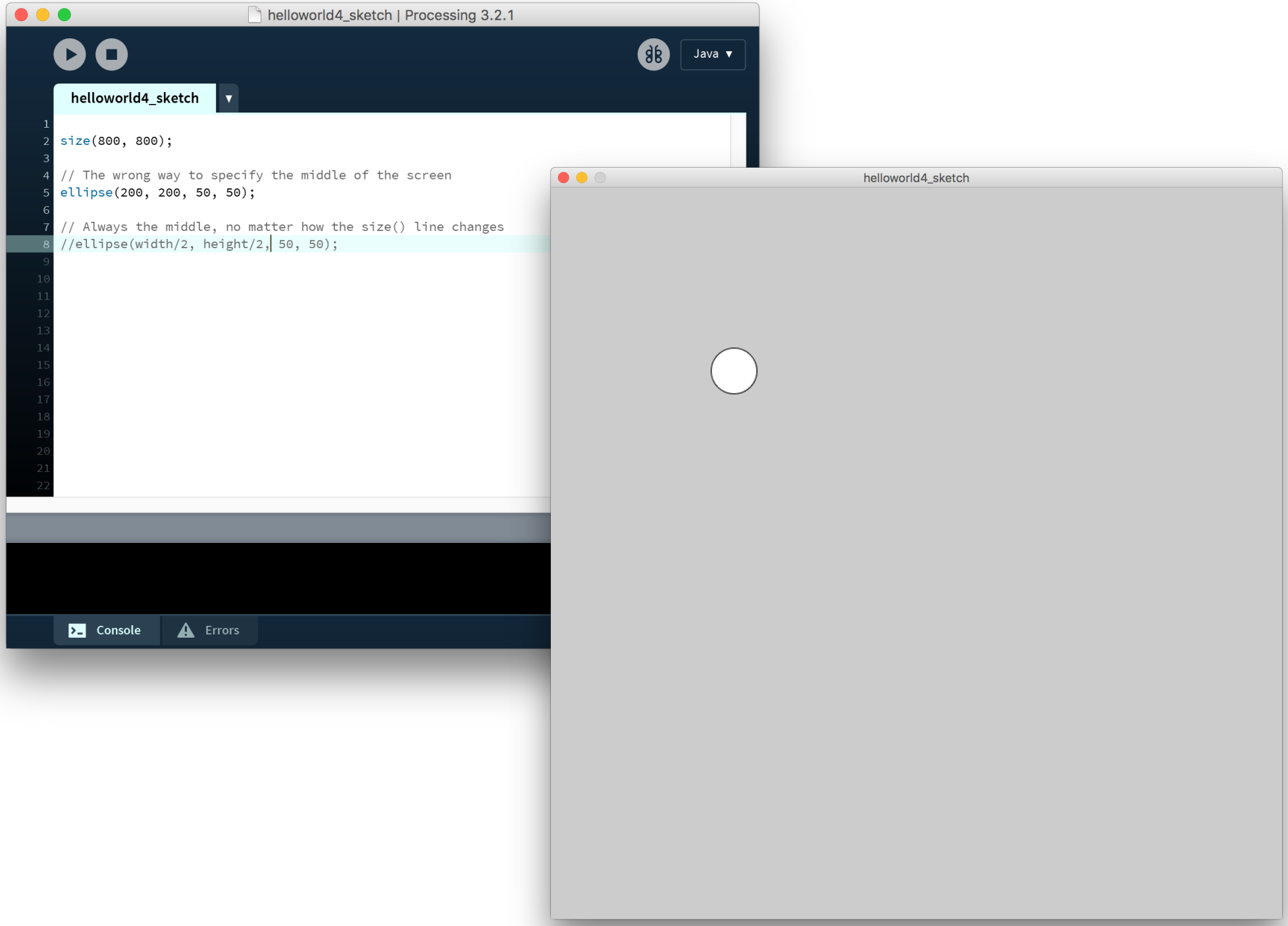
Exported Application | Click to Run



Sizing


helloworld4_sketch | Processing 3.2.1
Java
helloworld4_sketch
size(400, 400);
// The wrong way to specify the middle of the screen
// ellipse(200, 200, 50, 50);
// Always the middle, no matter how the size() line changes
ellipse(width/2, height/2, 50, 50);
Console
Errors


helloworld4_sketch


helloworld4_sketch | Processing 3.2.1
Java
helloworld4_sketch
size(800, 800);
// The wrong way to specify the middle of the screen
ellipse(200, 200, 50, 50);
// Always the middle, no matter how the size() line changes
//ellipse(width/2, height/2, 50, 50);
Console
Errors
helloworld4_sketch

Loading Data & Images

Loading and displaying data

One of the unique aspects of the Processing API is the way files are handled. The `loadImage()` and `loadStrings()` functions each expect to find a file inside a folder named *data*, which is a subdirectory of the sketch folder.

Advanced Topic: Notes on the data folder

The *data* folder addresses a common frustration when dealing with code that is tested locally but deployed over the web. Like Java, software written with Processing is subject to security restrictions that determine how a program can access resources such as the local hard disk or other servers via the Internet. This prevents malicious developers from writing code that could harm your computer or compromise your data.

The security restrictions can be tricky to work with during development. When running a program locally, data can be read directly from the disk, though it must be placed relative to the user's "working directory," generally the location of the application. When running online, data must come from a location on the same server. It might be bundled with the code itself (in a JAR archive, discussed later; or from another URL on the same server). For a local file, Java's `FileInputStream` class can be used. If the file is bundled in a JAR archive, the `getResource()` function is used. For a file on the server, `URL.openStream()` might be employed. During the journey from development to deployment, it may be necessary to use all three of these functions.

With Processing, each of these scenarios (and some others) is handled transparently by the file API functions. By placing resources in the data folder, Processing packages the files as necessary for online and offline use.

helloworld5_sketch | Processing 3.2.1

Java ▾

helloworld5_sketch

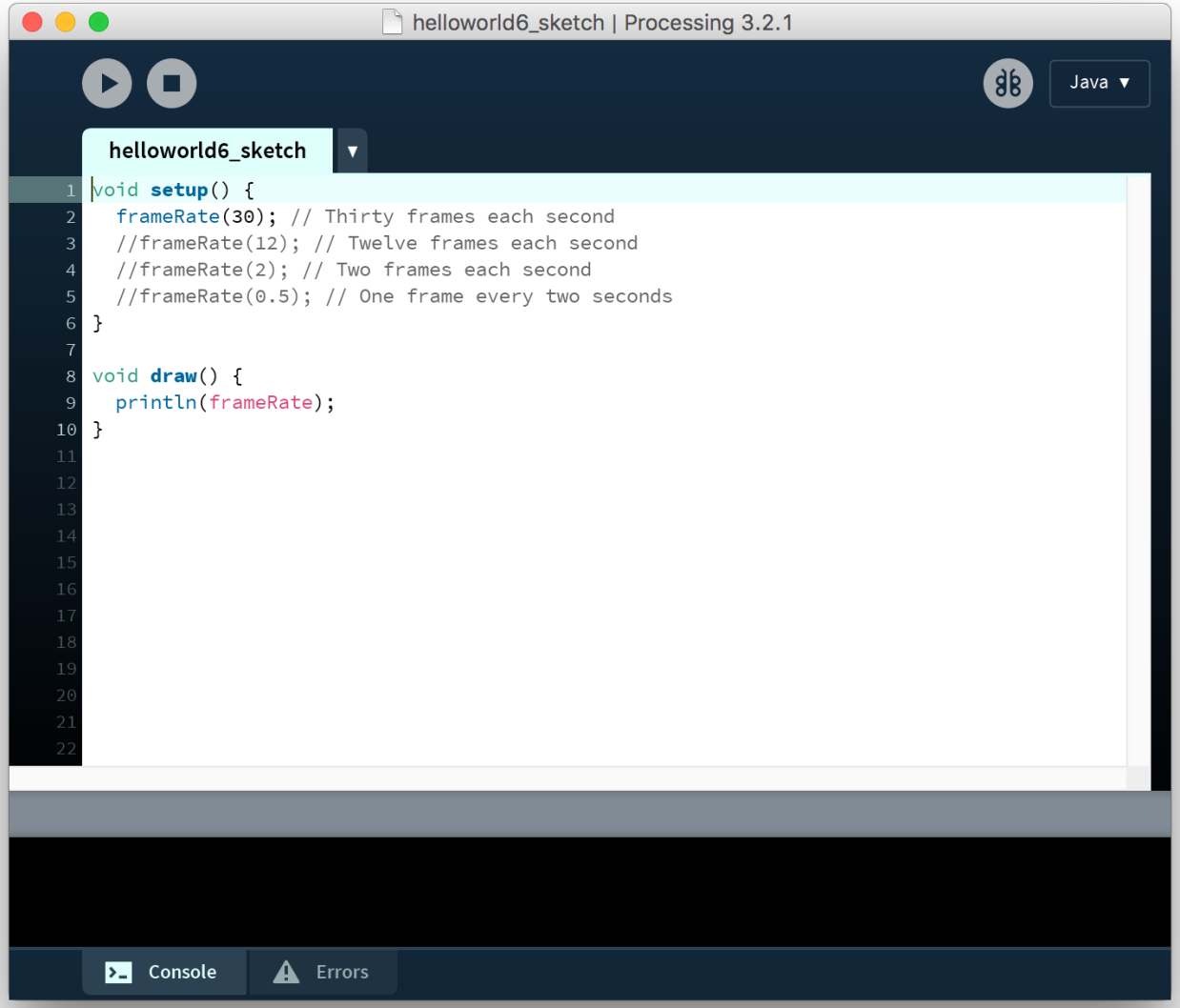
```
1
2  background(255) ;
3  size(800, 800) ;
4
5  // Examples of loading a text file and a JPEG image
6  // from the data folder of a sketch.
7
8  String[] lines = loadStrings("helloworld.txt");
9  PImage image = loadImage("gumball.jpg");
10
11 println( "Image Width: " + image.height ) ;
12 println( "Image Height: " + image.width ) ;
13 printArray( lines ) ;
14
15 // Displays the image (centered on canvas) at its actual size
16 image(image, (width-image.width)/2, (height-image.height)/2);
17
18
19
```

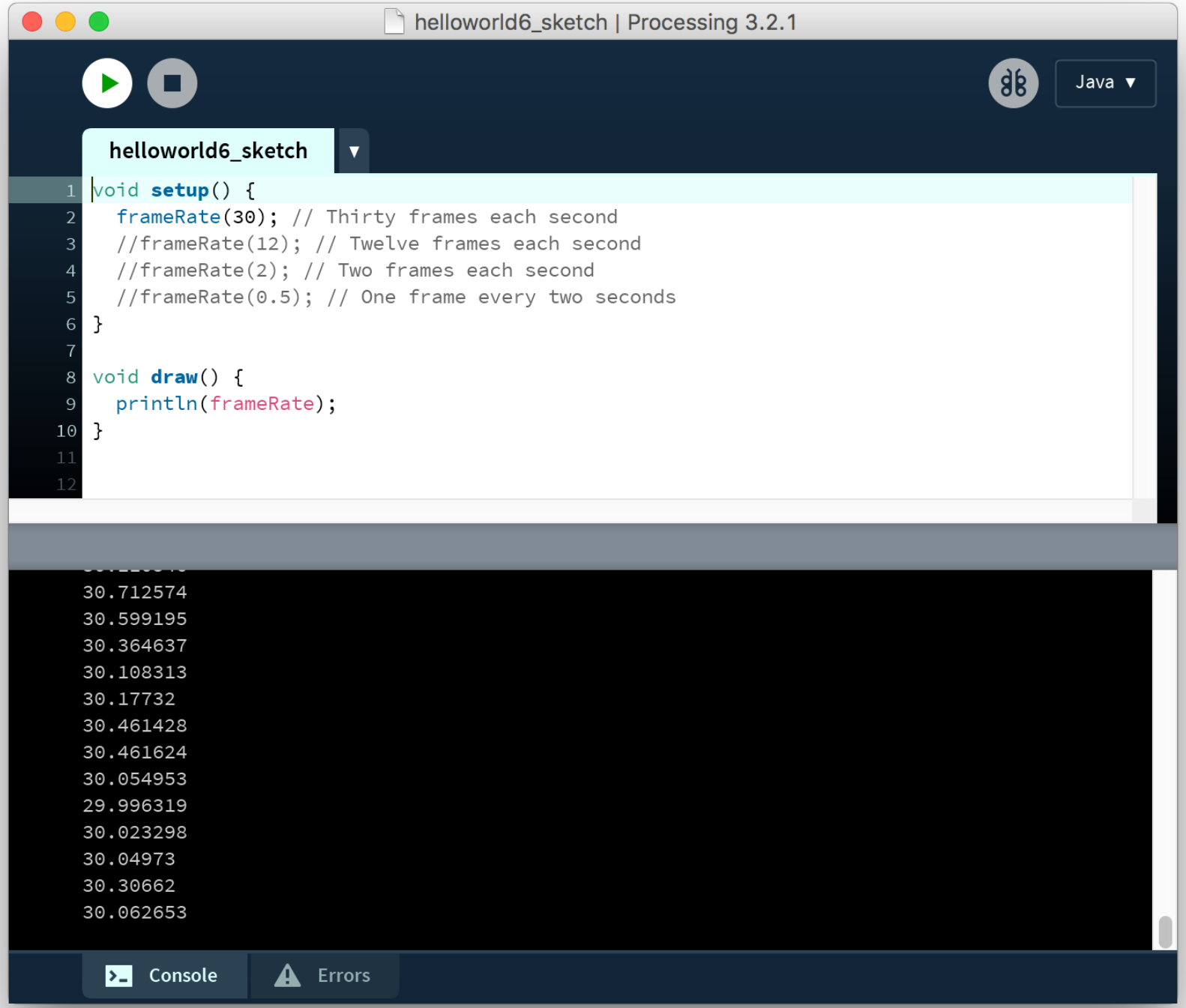
```
Image Width: 553
Image Height: 570
[0] "Hello World!"
[1] "Processing is Cool!"
```

Console Errors

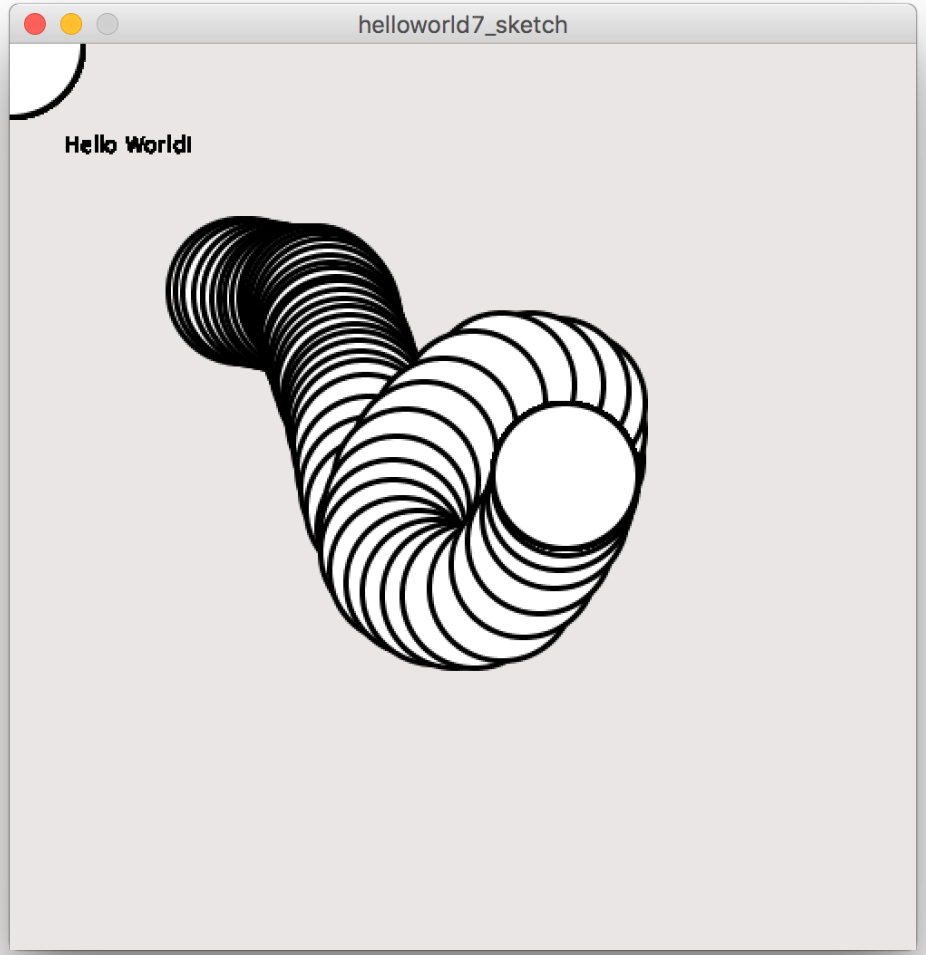
helloworld5_sketch

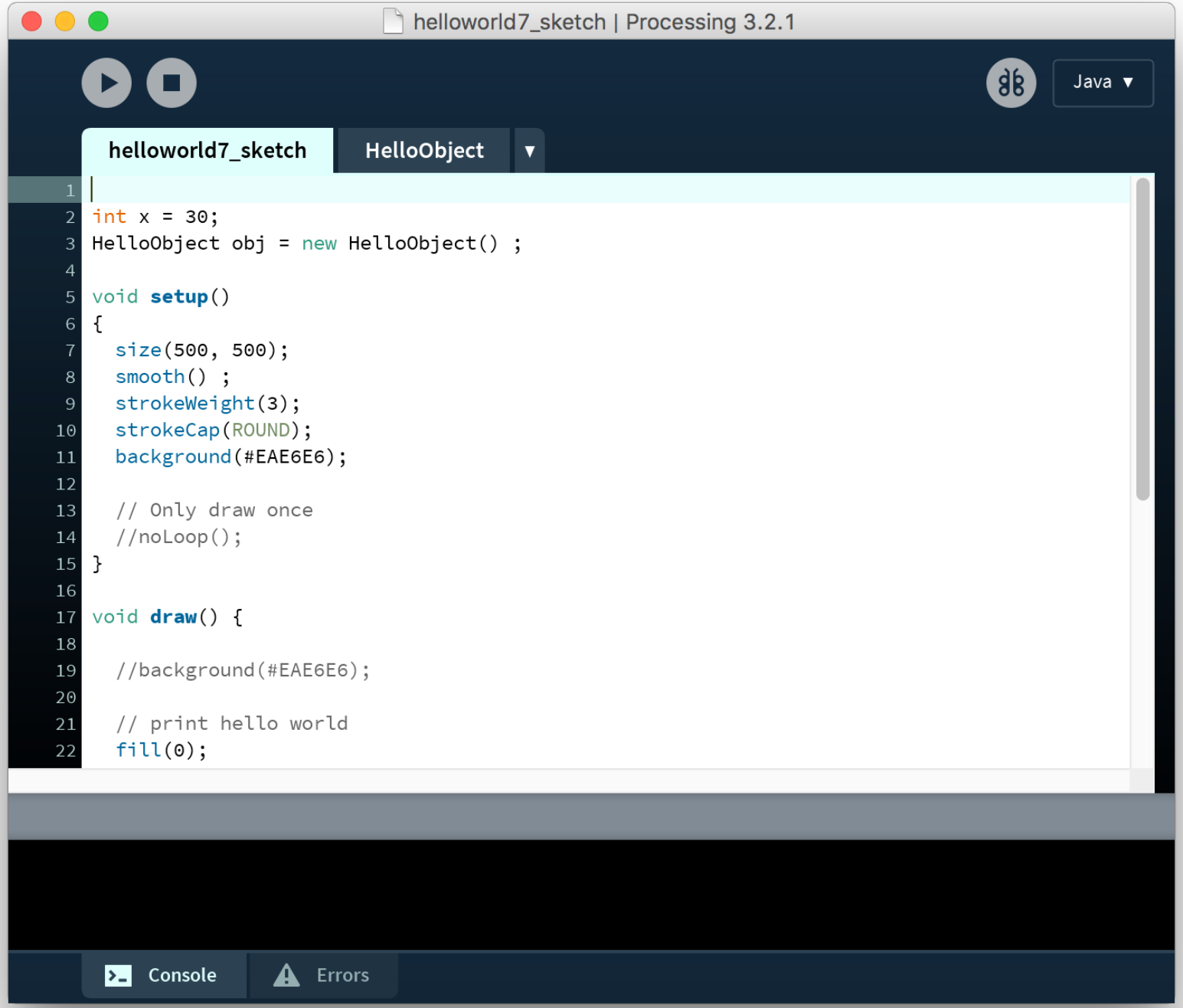
Frame Rates

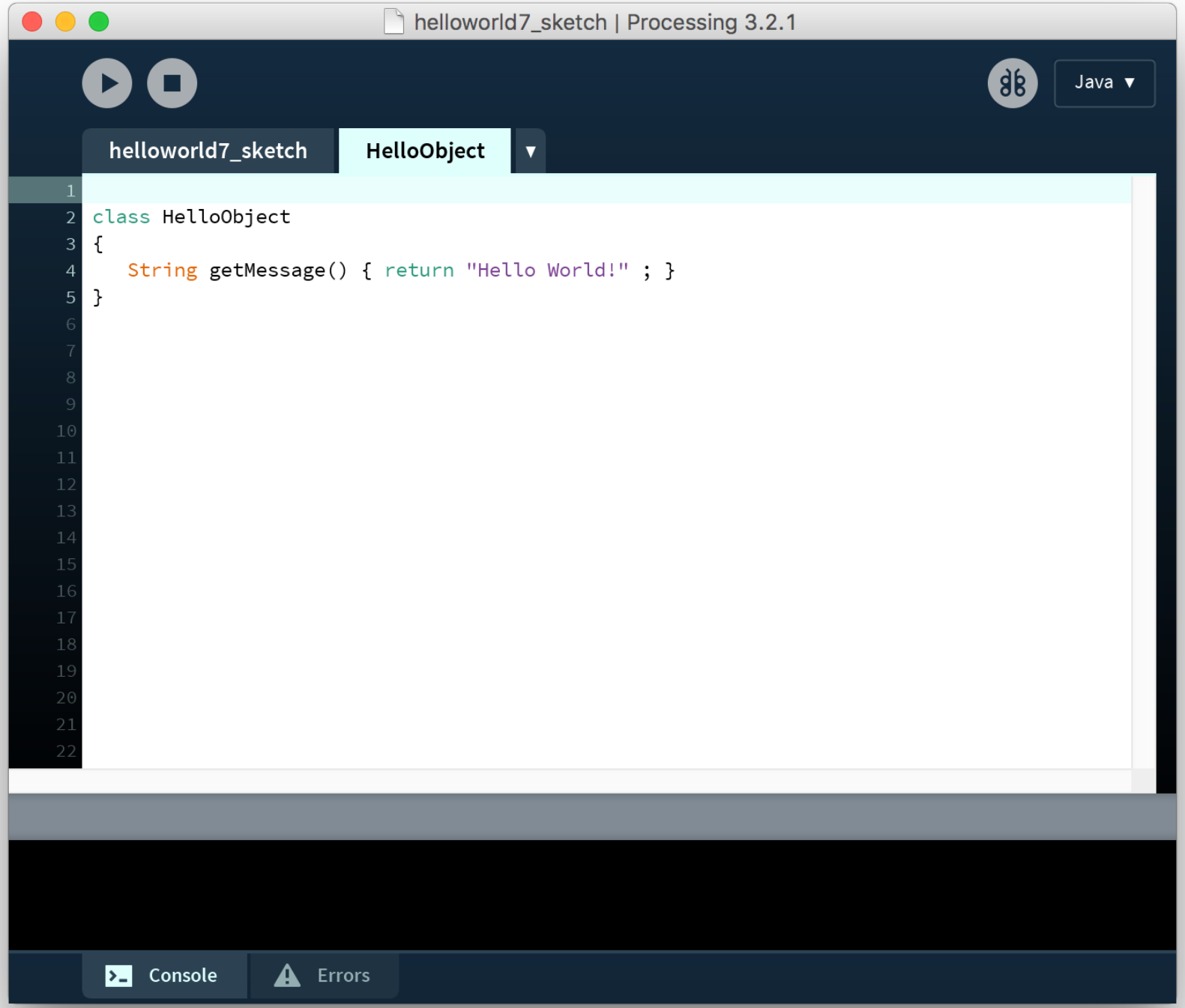

helloworld6_sketch | Processing 3.2.1
Java
helloworld6_sketch
void setup() {
frameRate(30); // Thirty frames each second
//frameRate(12); // Twelve frames each second
//frameRate(2); // Two frames each second
//frameRate(0.5); // One frame every two seconds
}
void draw() {
println(frameRate);
}
Console
Errors


helloworld6_sketch | Processing 3.2.1
Java
helloworld6_sketch
void setup() {
frameRate(30); // Thirty frames each second
//frameRate(12); // Twelve frames each second
//frameRate(2); // Two frames each second
//frameRate(0.5); // One frame every two seconds
}
void draw() {
println(frameRate);
}
30.712574
30.599195
30.364637
30.108313
30.17732
30.461428
30.461624
30.054953
29.996319
30.023298
30.04973
30.30662
30.062653
Console
Errors

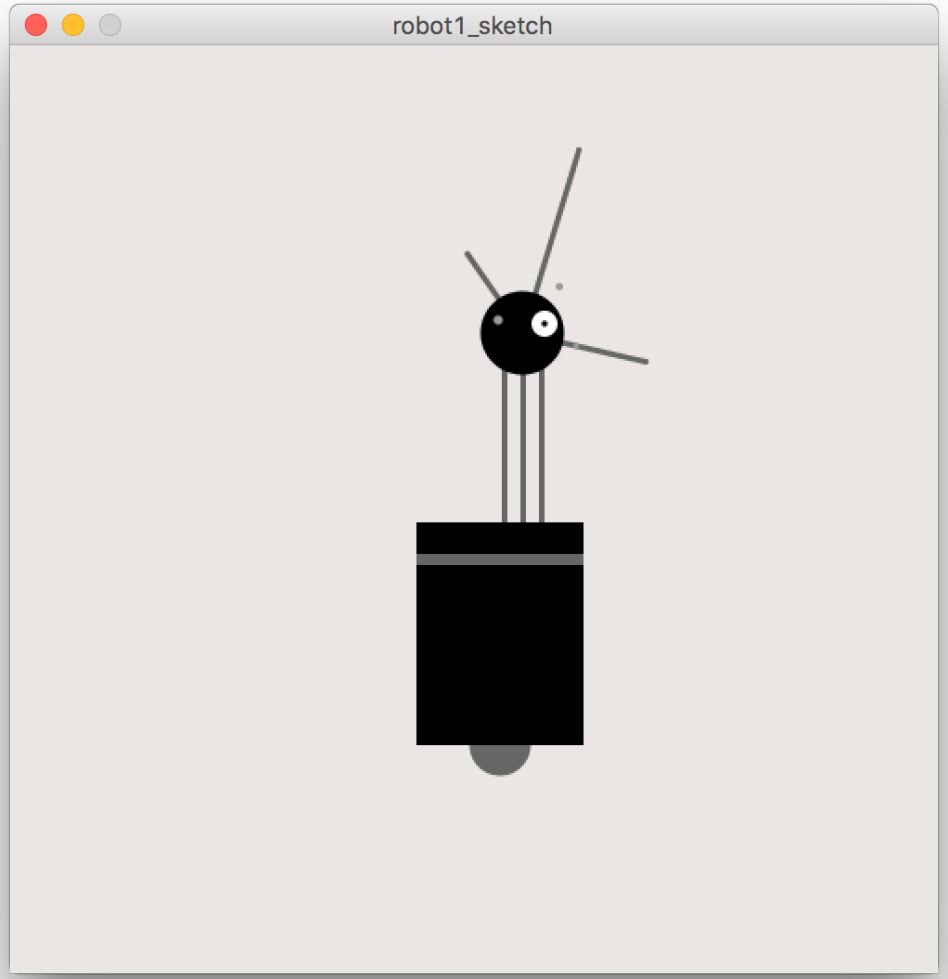
Hello World with Java Objects

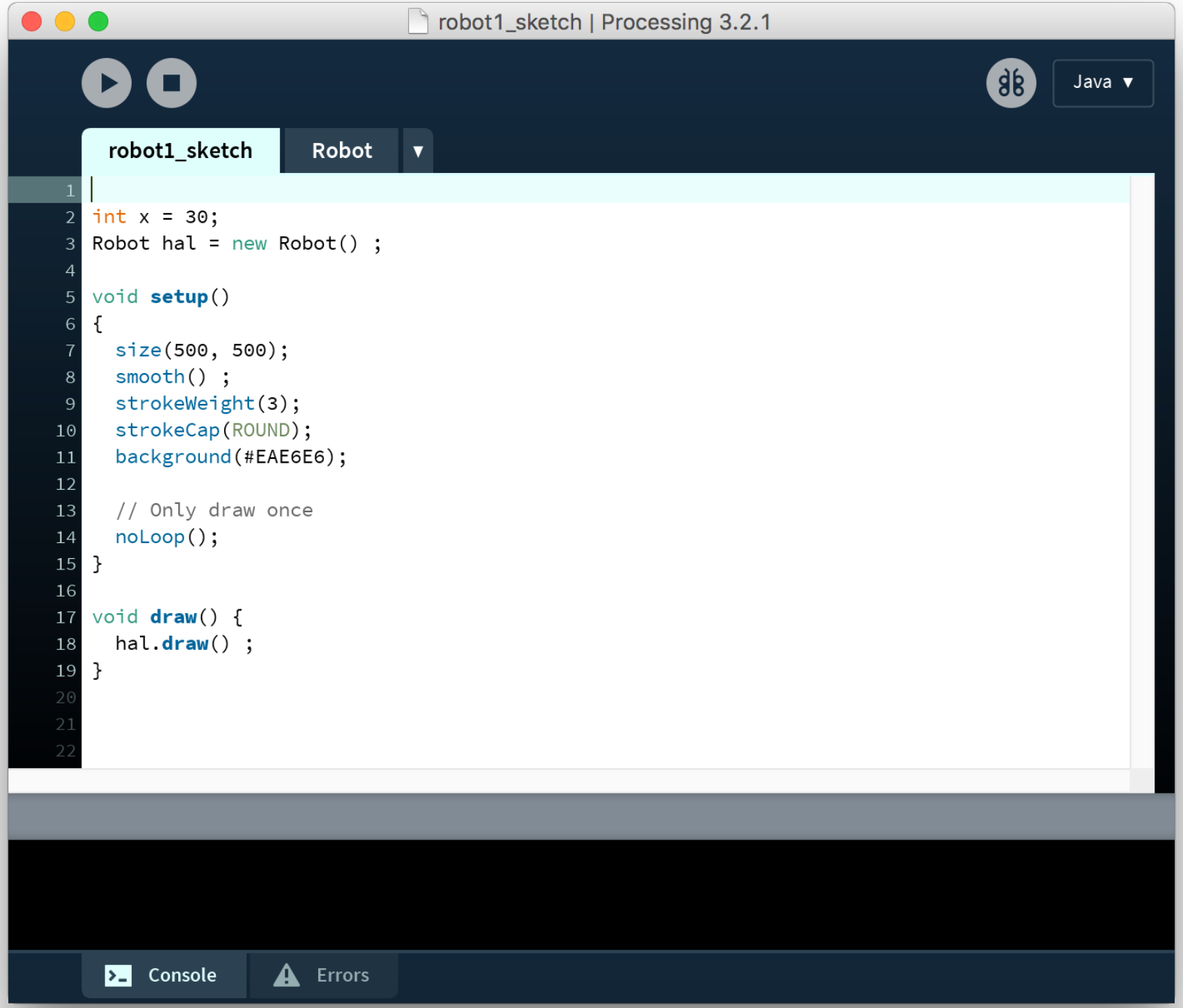


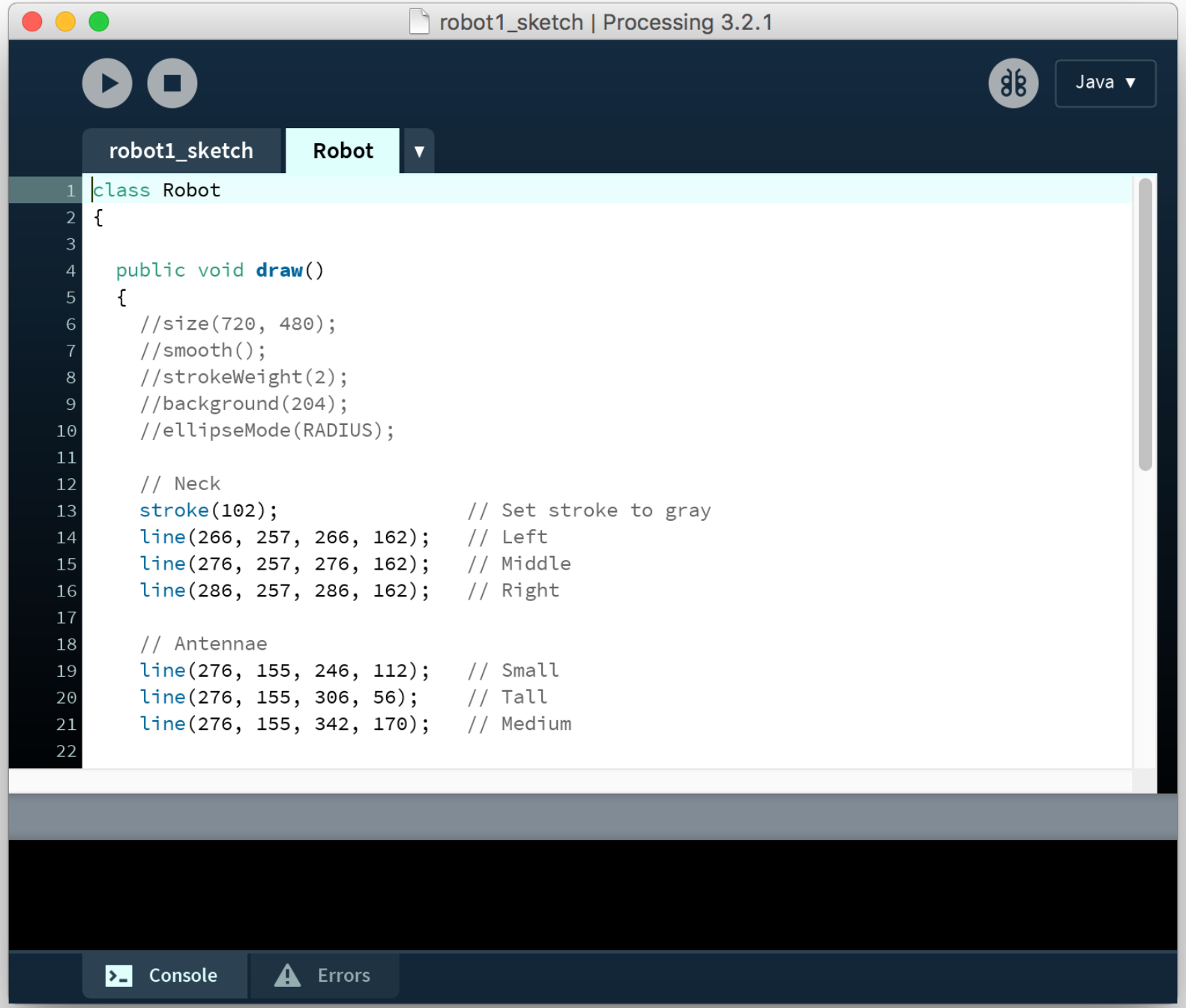

helloworld7_sketch | Processing 3.2.1
Java
helloworld7_sketch
HelloObject
int x = 30;
HelloObject obj = new HelloObject() ;
void setup()
{
size(500, 500);
smooth() ;
strokeWeight(3);
strokeCap(ROUND);
background(#EAE6E6);
// Only draw once
//noLoop();
}
void draw() {
//background(#EAE6E6);
// print hello world
fill(0);
Console
Errors


helloworld7_sketch | Processing 3.2.1
Java
helloworld7_sketch
HelloObject
class HelloObject
{
String getMessage() { return "Hello World!" ; }
}
Console
Errors

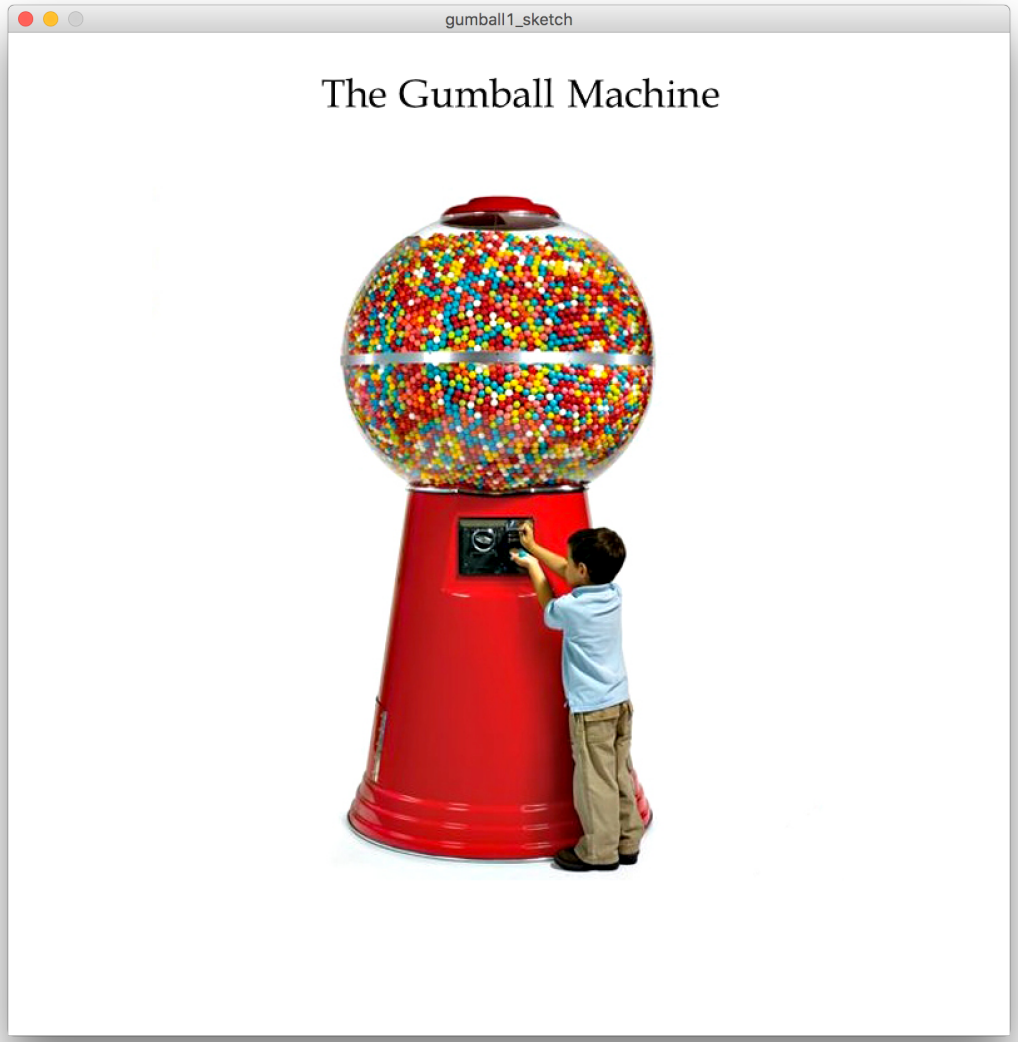
Hello HAL (the Robot)

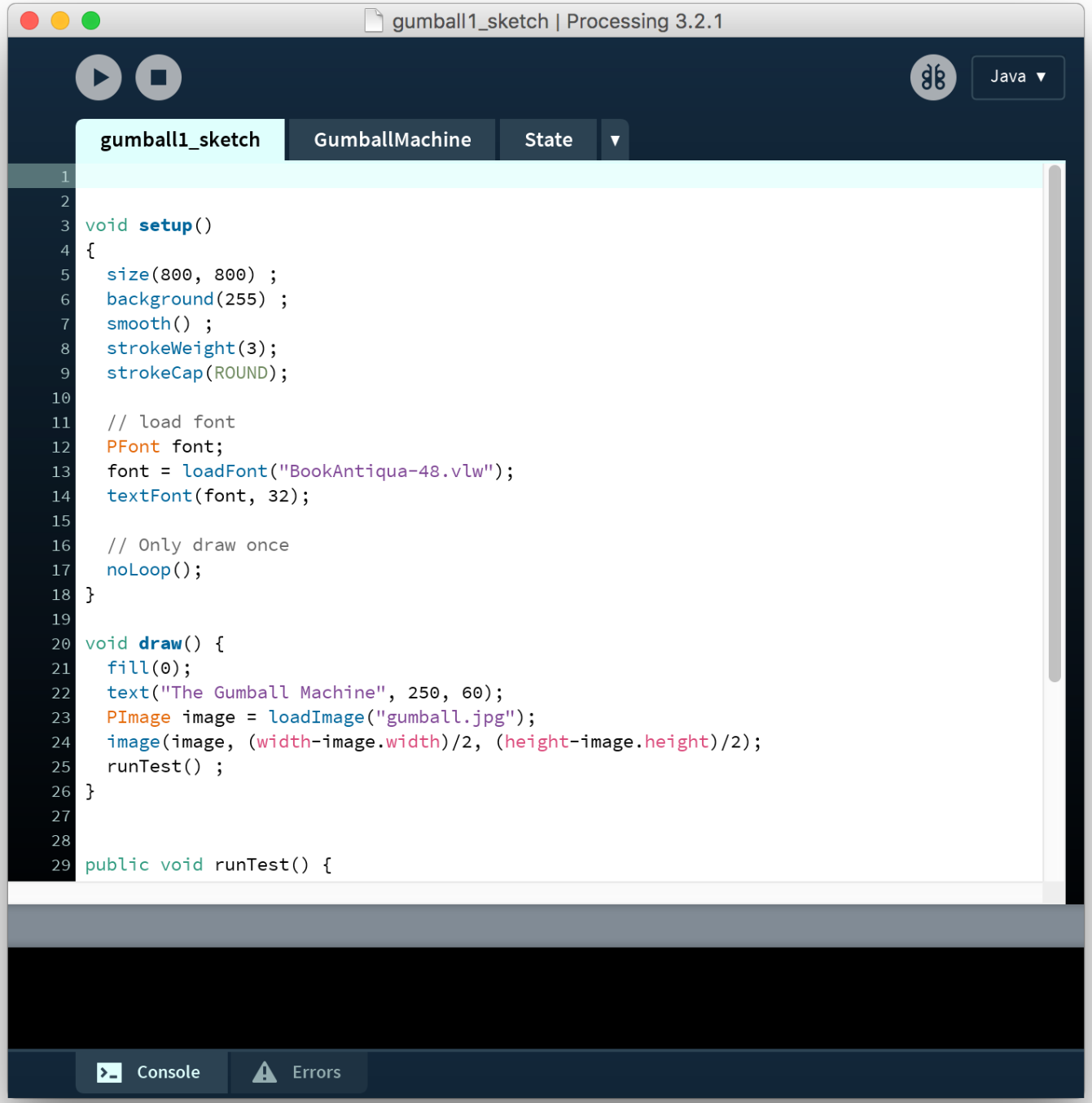


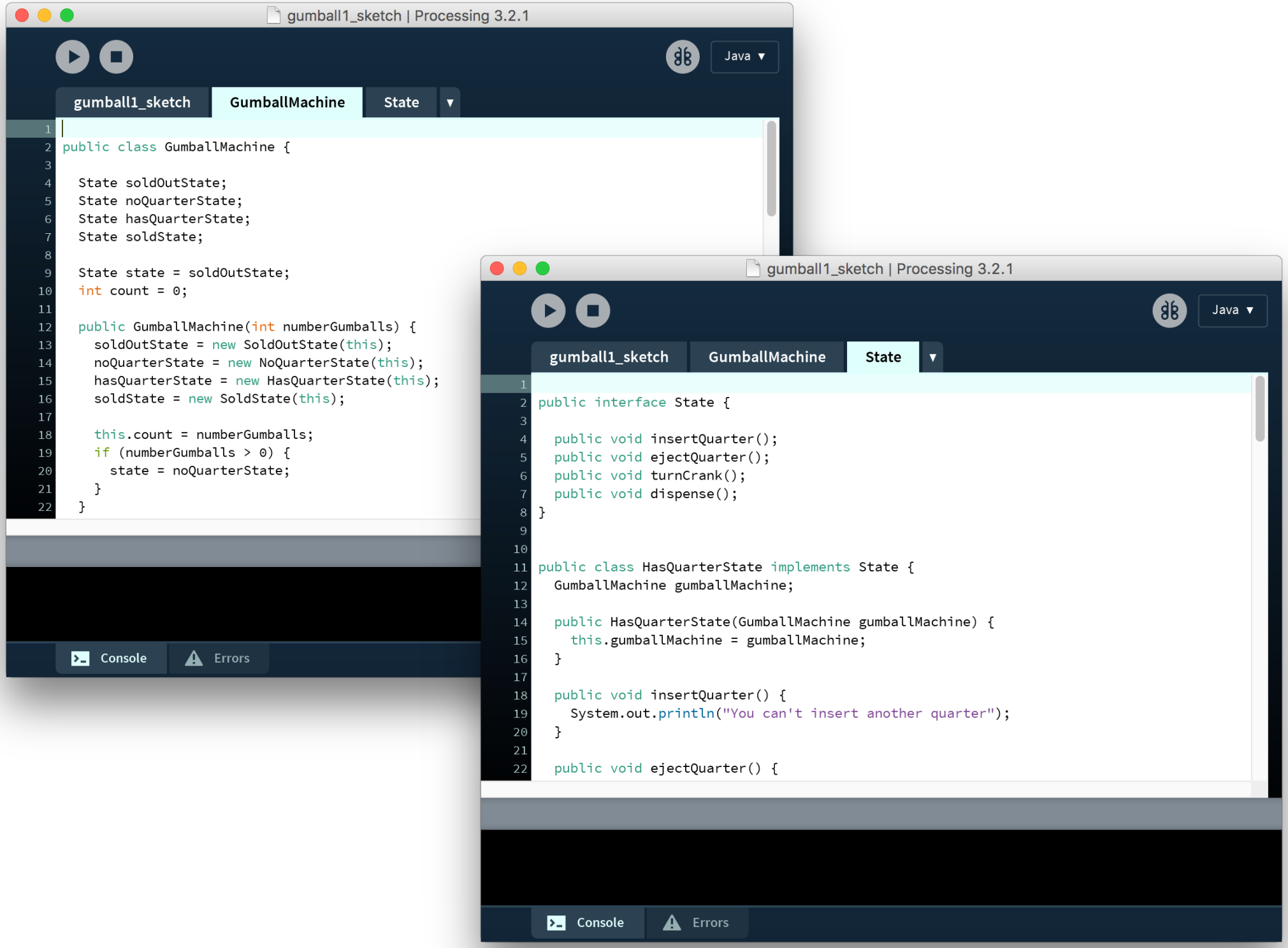

robot1_sketch | Processing 3.2.1
Java
robot1_sketch
Robot
int x = 30;
Robot hal = new Robot() ;
void setup()
{
size(500, 500);
smooth() ;
strokeWeight(3);
strokeCap(ROUND);
background(#EAE6E6);
// Only draw once
noLoop();
}
void draw() {
hal.draw() ;
}
Console
Errors


robot1_sketch | Processing 3.2.1
Java
robot1_sketch
Robot
class Robot
{
public void draw()
{
//size(720, 480);
//smooth();
//strokeWeight(2);
//background(204);
//ellipseMode(RADIUS);
// Neck
stroke(102); // Set stroke to gray
line(266, 257, 266, 162); // Left
line(276, 257, 276, 162); // Middle
line(286, 257, 286, 162); // Right
// Antennae
line(276, 155, 246, 112); // Small
line(276, 155, 306, 56); // Tall
line(276, 155, 342, 170); // Medium
Console
Errors

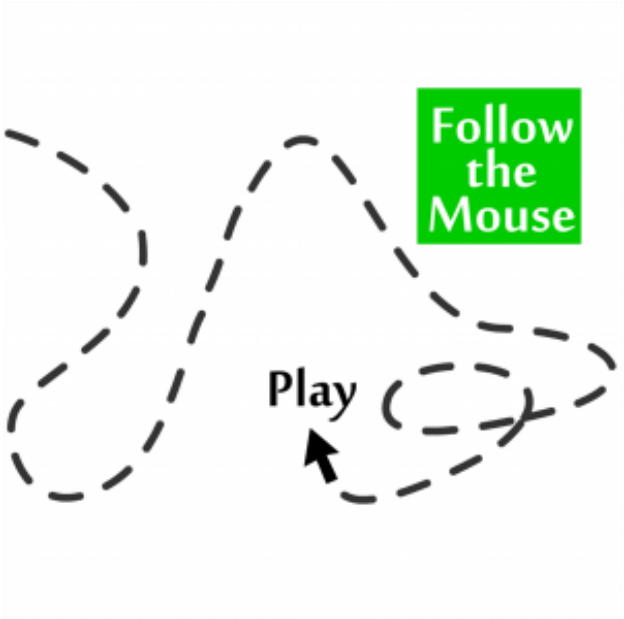
Gumball Machine (v1)

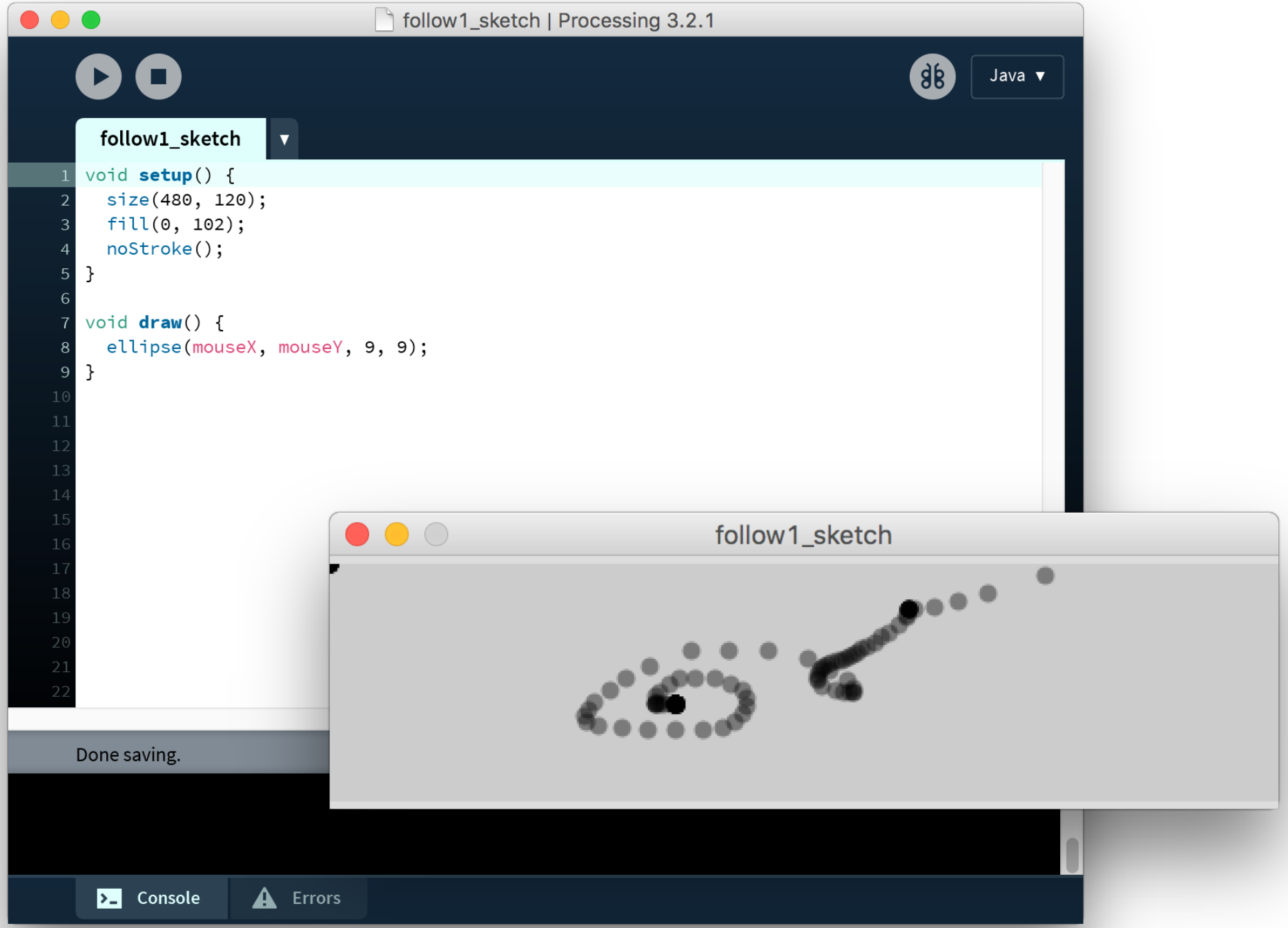


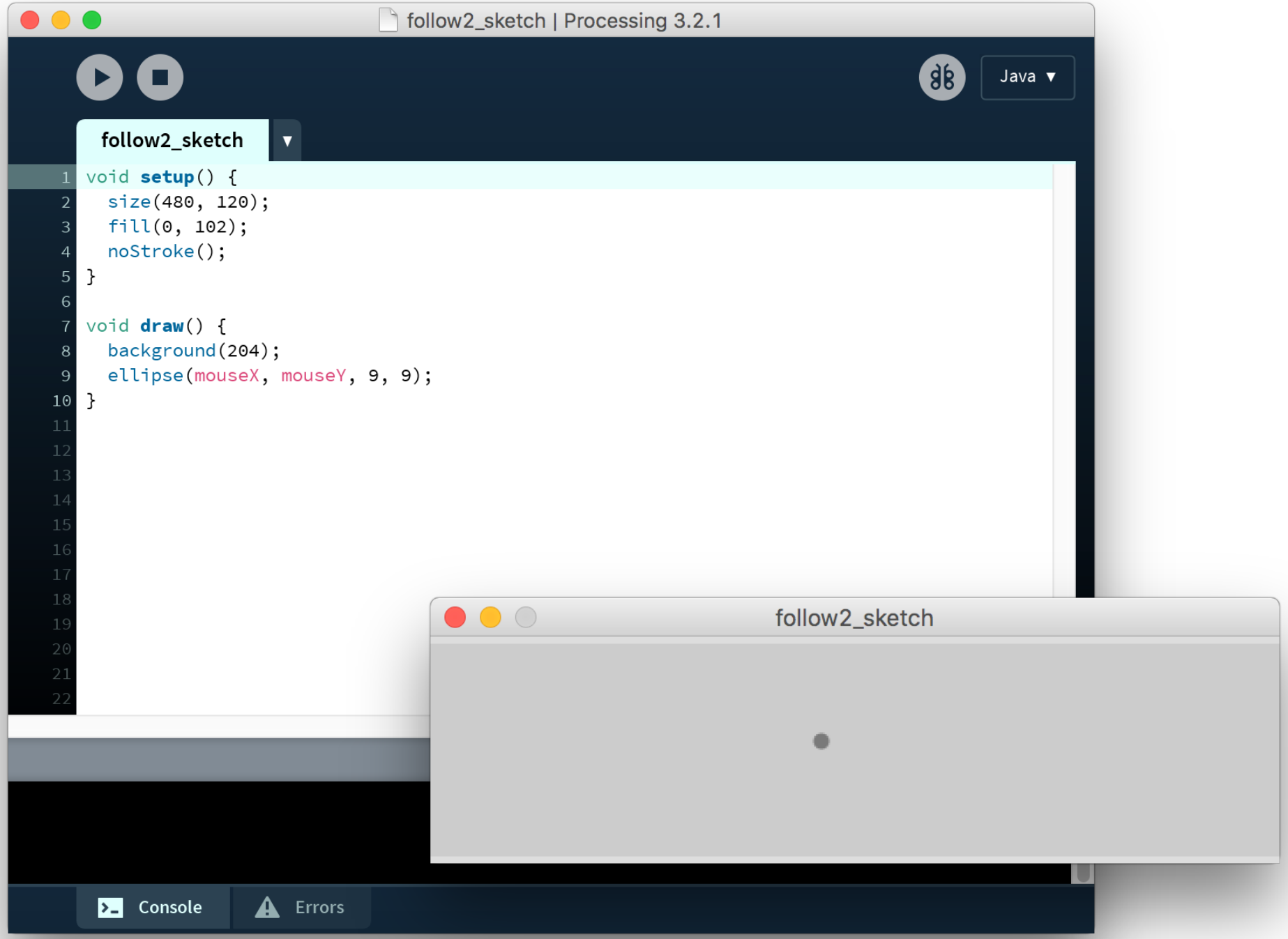

gumball1_sketch | Processing 3.2.1
Java
gumball1_sketch
GumballMachine
State
1
2
3 void setup()
4 {
5 size(800, 800) ;
6 background(255) ;
7 smooth() ;
8 strokeWeight(3);
9 strokeCap(ROUND);
10
11 // load font
12 PFont font;
13 font = loadFont("BookAntiqua-48.vlw");
14 textFont(font, 32);
15
16 // Only draw once
17 noLoop();
18 }
19
20 void draw() {
21 fill(0);
22 text("The Gumball Machine", 250, 60);
23 PImage image = loadImage("gumball.jpg");
24 image(image, (width-image.width)/2, (height-image.height)/2);
25 runTest() ;
26 }
27
28
29 public void runTest() {
Console
Errors


gumball1_sketch | Processing 3.2.1
Java
gumball1_sketch
GumballMachine
State
public class GumballMachine {
State soldOutState;
State noQuarterState;
State hasQuarterState;
State soldState;
State state = soldOutState;
int count = 0;
public GumballMachine(int numberGumballs) {
soldOutState = new SoldOutState(this);
noQuarterState = new NoQuarterState(this);
hasQuarterState = new HasQuarterState(this);
soldState = new SoldState(this);
this.count = numberGumballs;
if (numberGumballs > 0) {
state = noQuarterState;
}
}
Console
Errors
gumball1_sketch | Processing 3.2.1
Java
gumball1_sketch
GumballMachine
State
public interface State {
public void insertQuarter();
public void ejectQuarter();
public void turnCrank();
public void dispense();
}
public class HasQuarterState implements State {
GumballMachine gumballMachine;
public HasQuarterState(GumballMachine gumballMachine) {
this.gumballMachine = gumballMachine;
}
public void insertQuarter() {
System.out.println("You can't insert another quarter");
}
public void ejectQuarter() {
Console
Errors

Follow Me




follow1_sketch | Processing 3.2.1
Java
follow1_sketch
void setup() {
size(480, 120);
fill(0, 102);
noStroke();
}
void draw() {
ellipse(mouseX, mouseY, 9, 9);
}
follow1_sketch
Done saving.
Console
Errors


follow2_sketch | Processing 3.2.1
Java
follow2_sketch
void setup() {
size(480, 120);
fill(0, 102);
noStroke();
}
void draw() {
background(204);
ellipse(mouseX, mouseY, 9, 9);
}
follow2_sketch
Console
Errors

follow3_sketch | Processing 3.2.1

Java

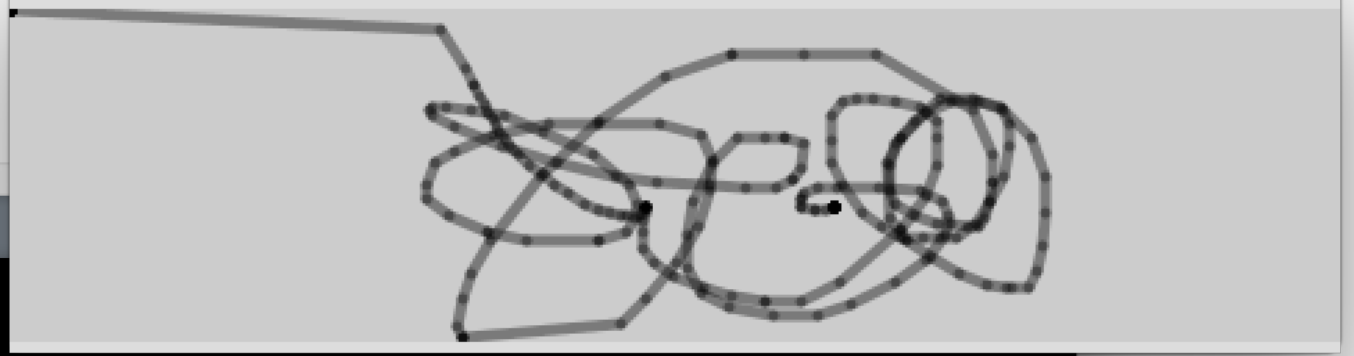
follow3_sketch

```
void setup() {
  size(480, 120);
  strokeWeight(4);
  stroke(0, 102);
}

void draw() {
  line(mouseX, mouseY, pmouseX, pmouseY);
}
```

Done saving.

Console Errors

follow3_sketch

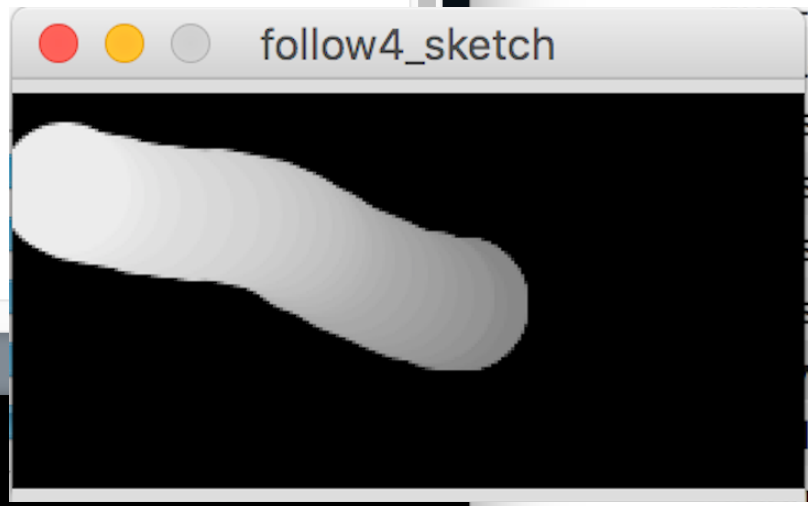


follow4_sketch | Processing 3.2.1

Java ▾

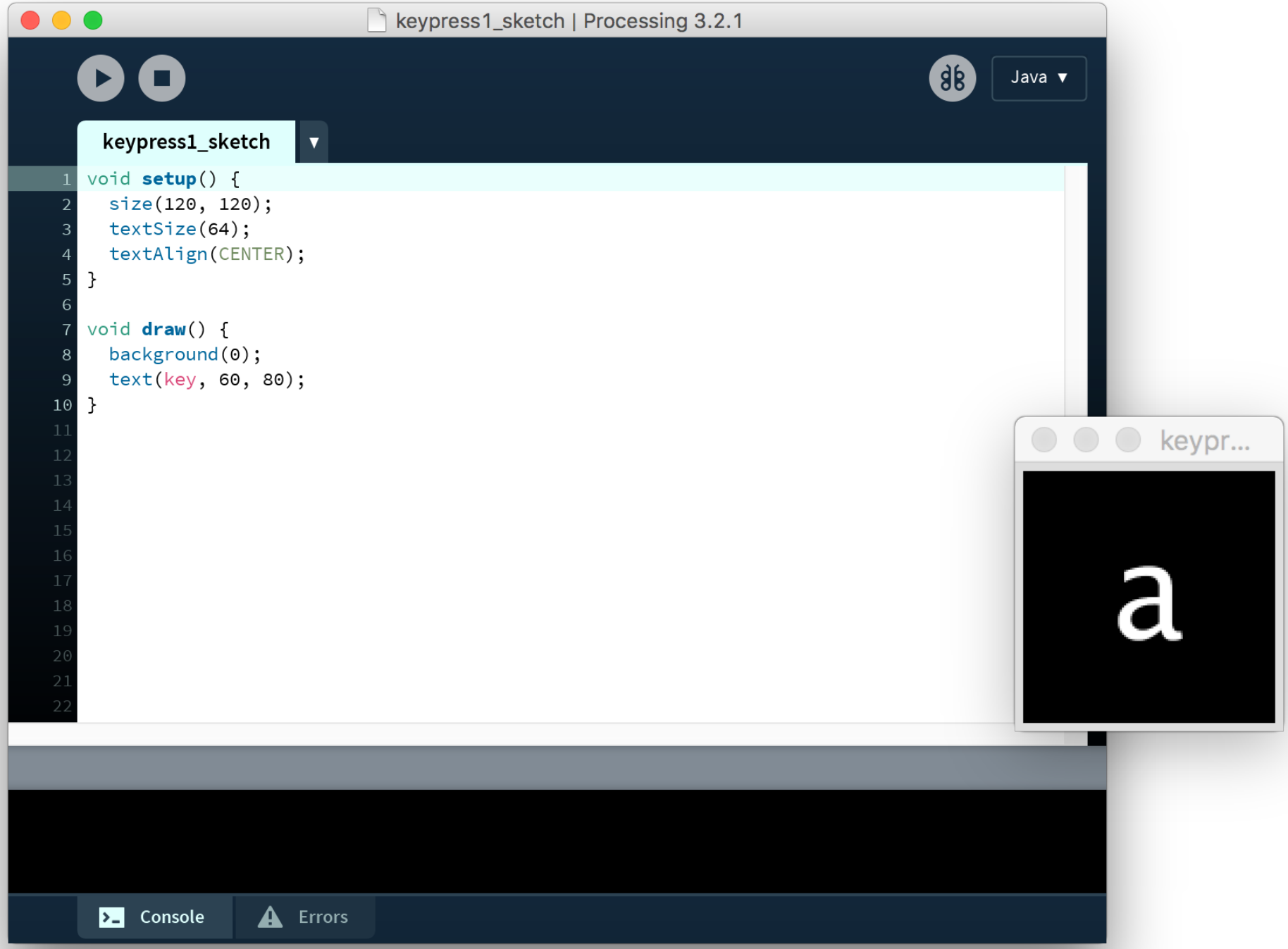
follow4_sketch ▾

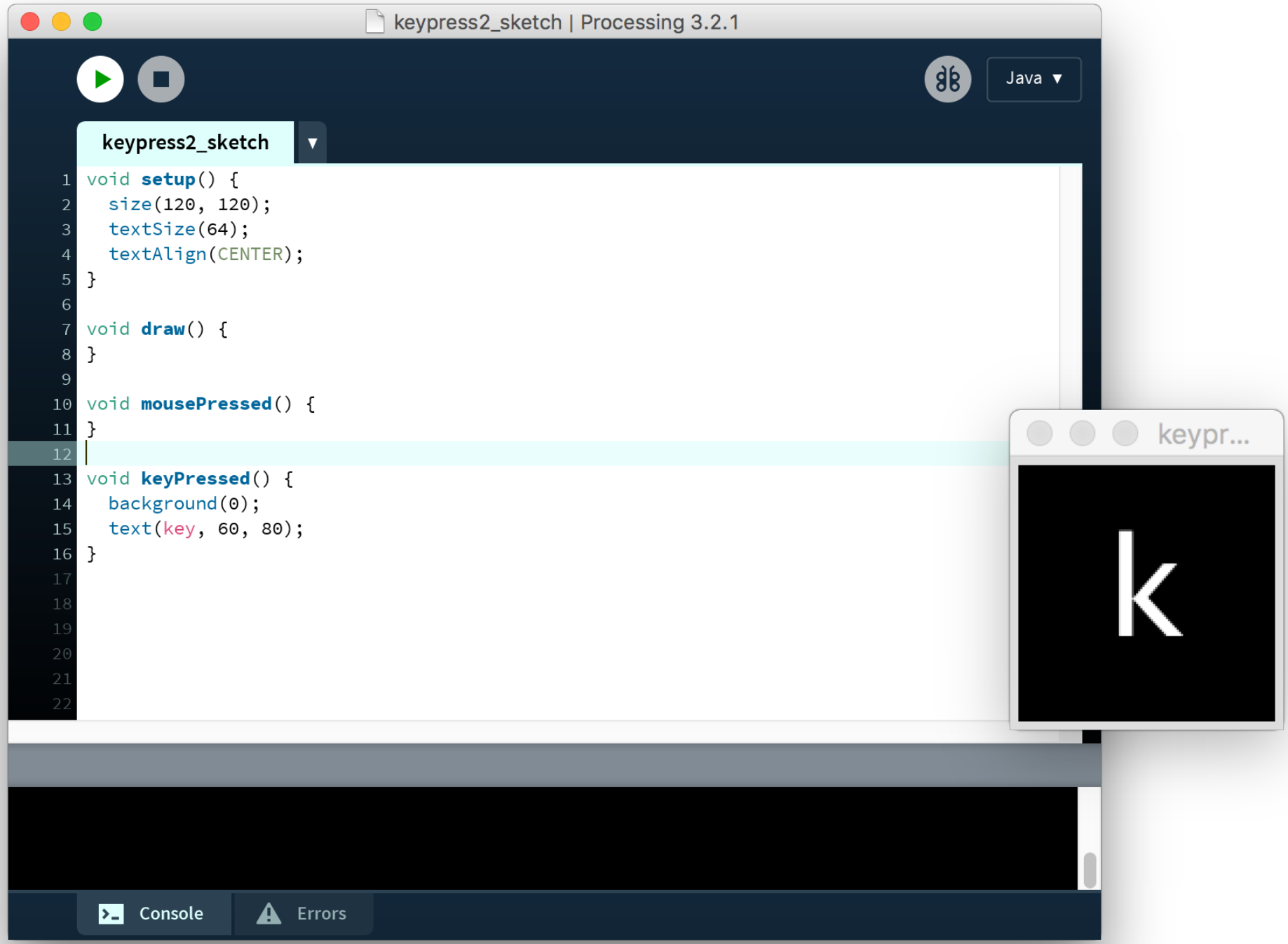
```
int num = 60;
int[] x = new int[num];
int[] y = new int[num];

void setup() {
  size(240, 120);
  noStroke();
}

void draw() {
  background(0);
  // Copy array values from back to front
  for (int i = x.length-1; i > 0; i--) {
    x[i] = x[i-1];
    y[i] = y[i-1];
  }
  x[0] = mouseX; // Set the first element
  y[0] = mouseY; // Set the first element
  for (int i = 0; i < x.length; i++) {
    fill(i * 4);
    ellipse(x[i], y[i], 40, 40);
  }
```

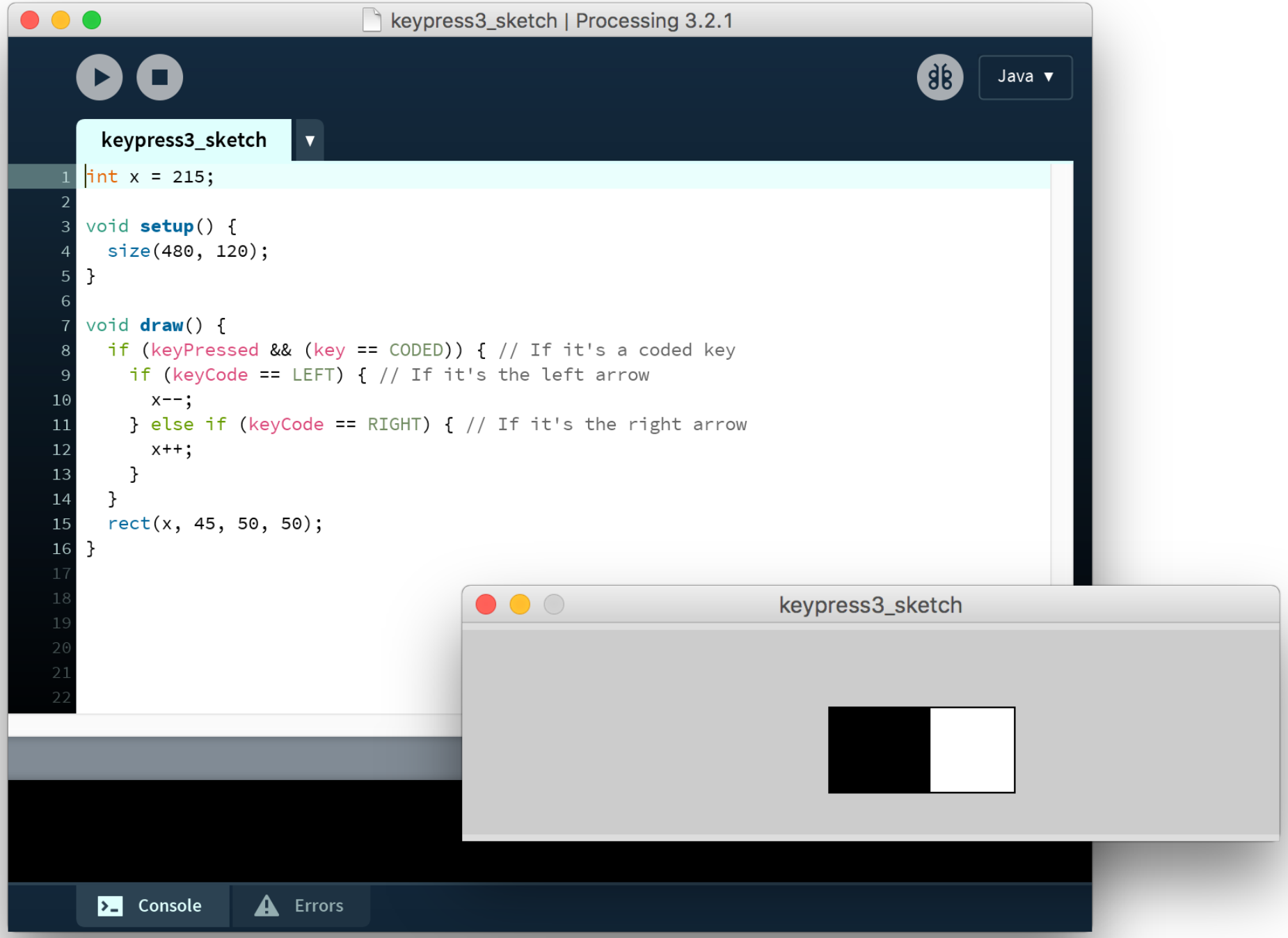


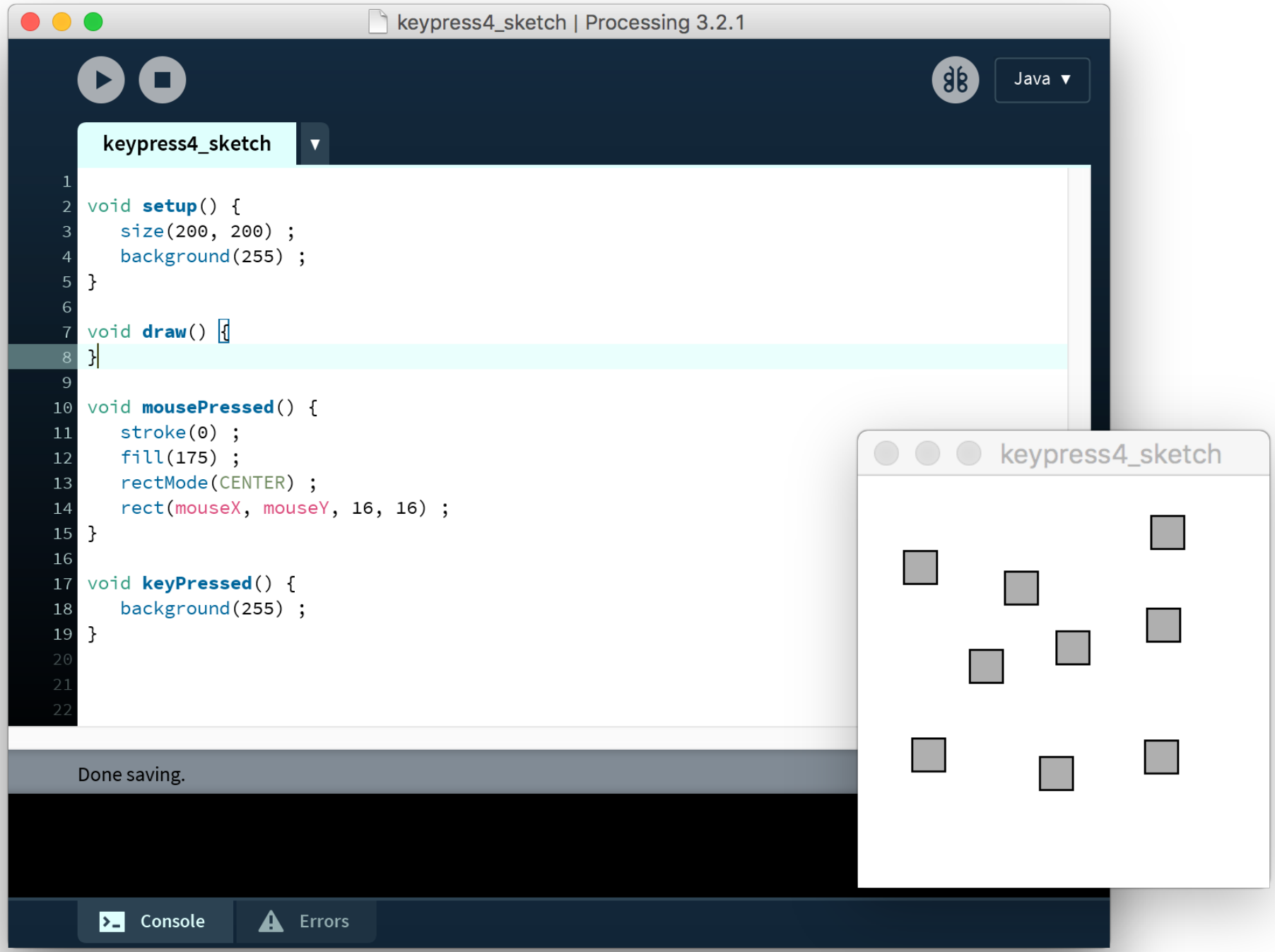
Console Errors

Keyboard Events

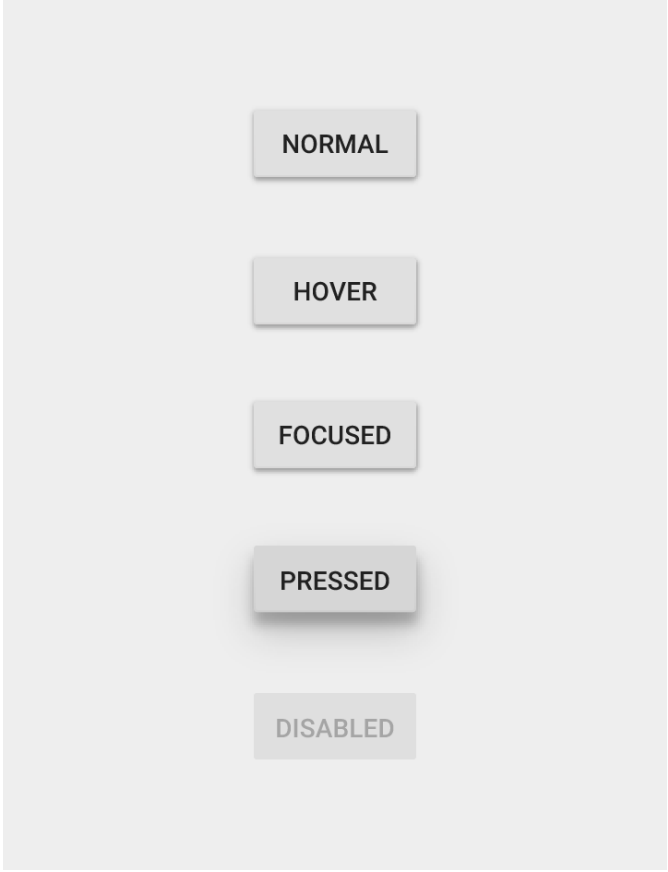

keypress1_sketch | Processing 3.2.1
Java
keypress1_sketch
void setup() {
size(120, 120);
textSize(64);
textAlign(CENTER);
}
void draw() {
background(0);
text(key, 60, 80);
}
Console
Errors
keypr...
a

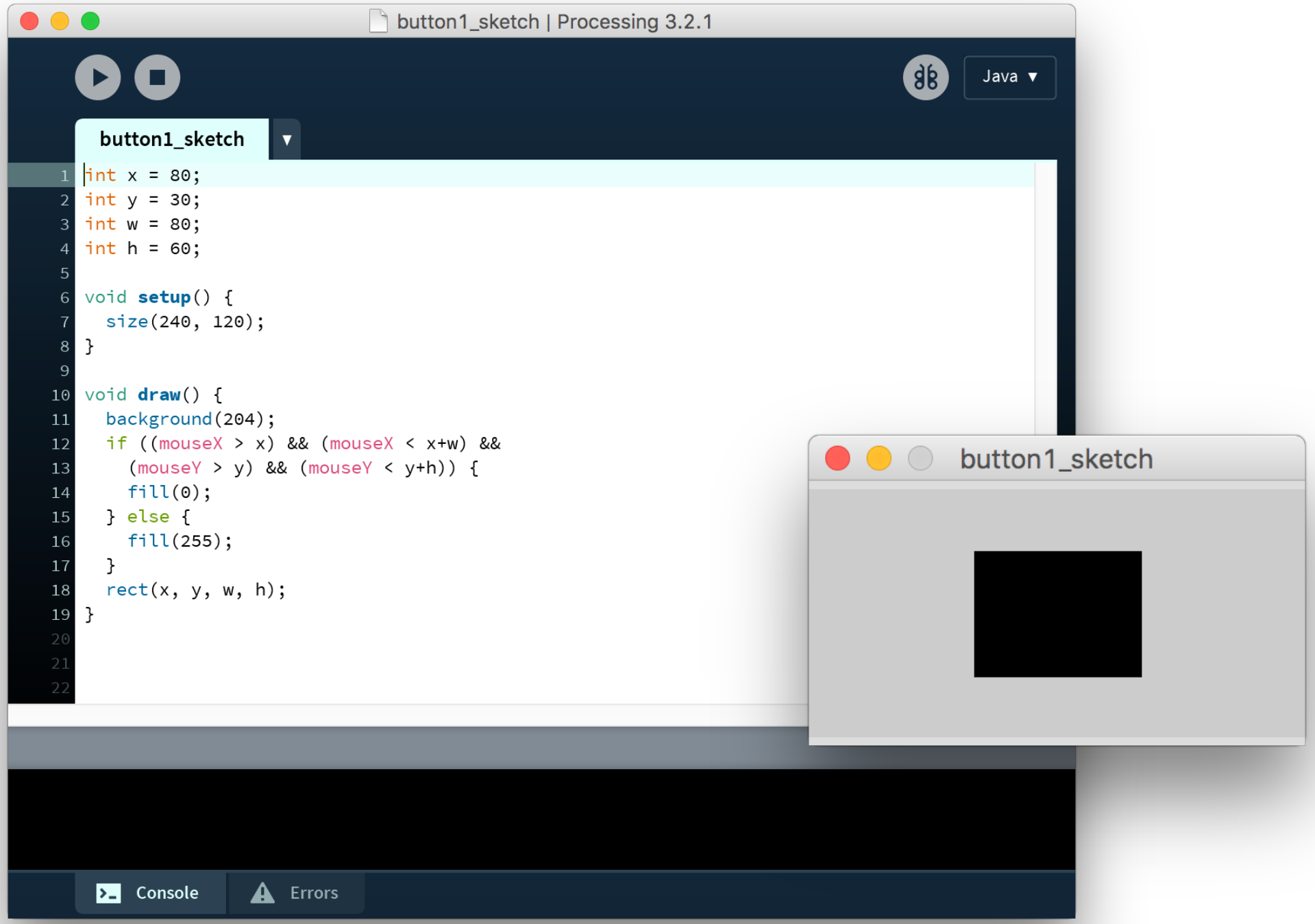

keypress2_sketch | Processing 3.2.1
Java
keypress2_sketch
void setup() {
size(120, 120);
textSize(64);
textAlign(CENTER);
}
void draw() {
}
void mousePressed() {
}
void keyPressed() {
background(0);
text(key, 60, 80);
}
Console
Errors
keypr...
k

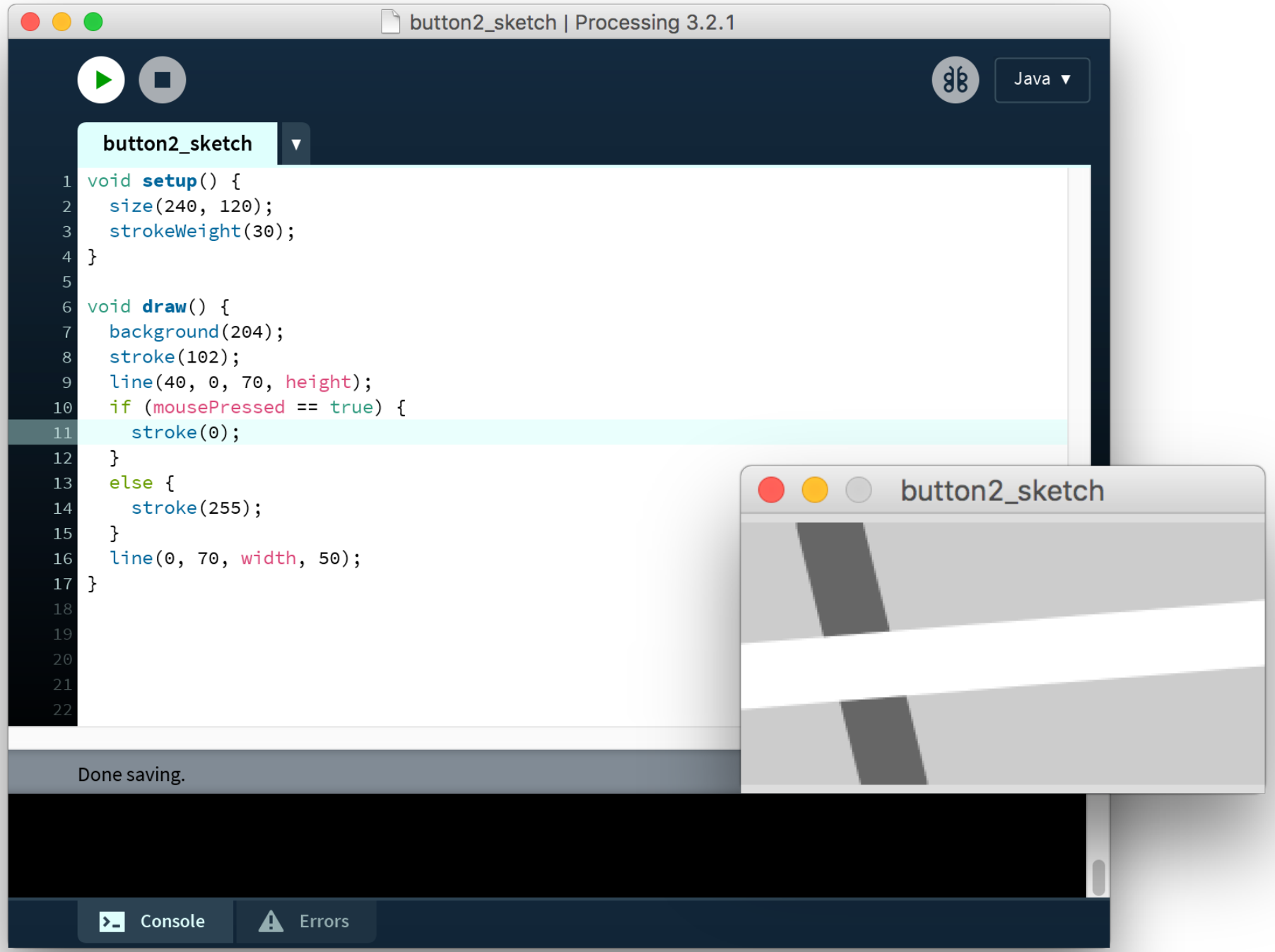

keypress3_sketch | Processing 3.2.1
Java
keypress3_sketch
int x = 215;
void setup() {
size(480, 120);
}
void draw() {
if (keyPressed && (key == CODED)) { // If it's a coded key
if (keyCode == LEFT) { // If it's the left arrow
x--;
} else if (keyCode == RIGHT) { // If it's the right arrow
x++;
}
}
rect(x, 45, 50, 50);
}
keypress3_sketch
Console
Errors

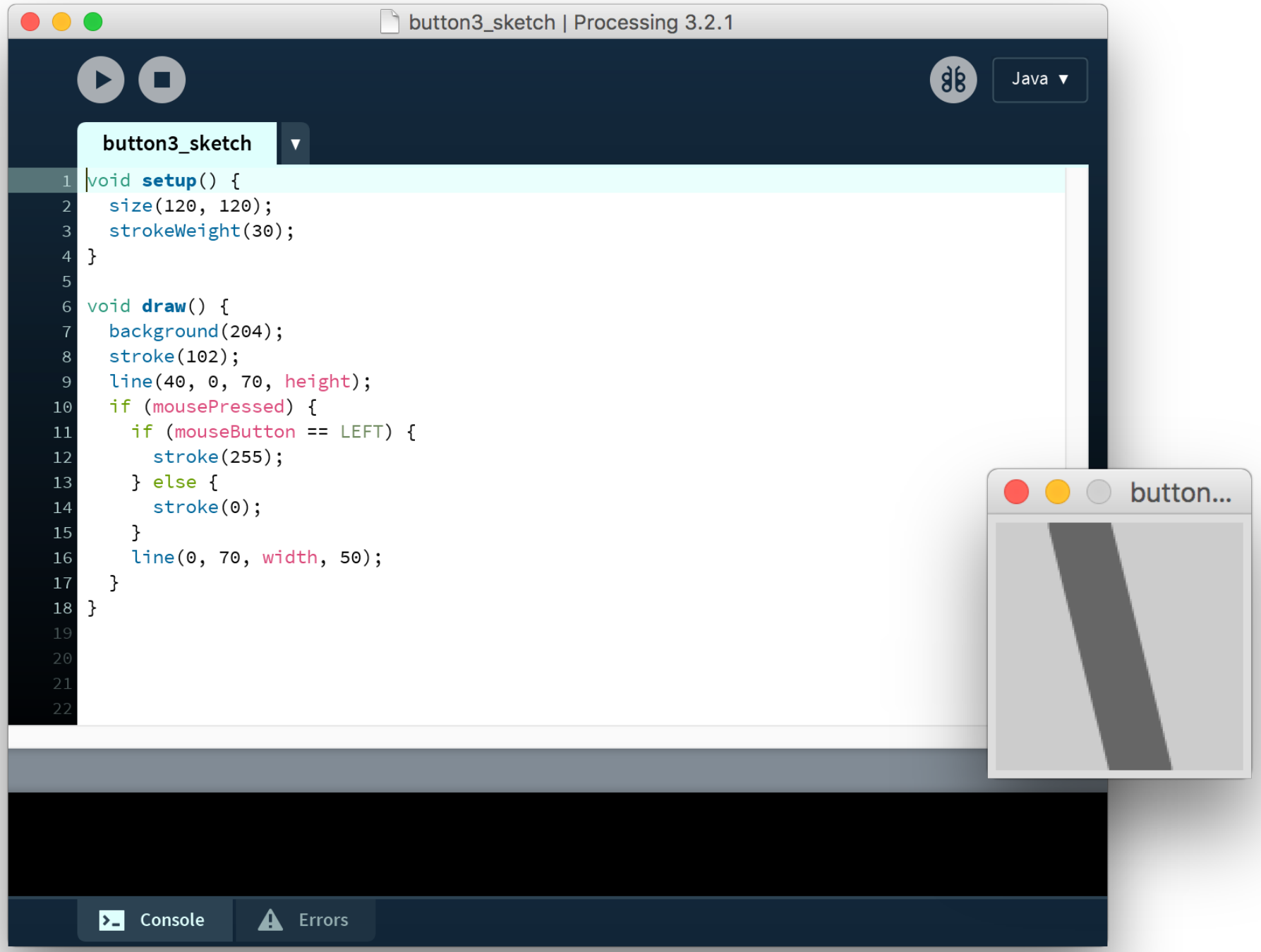

keypress4_sketch | Processing 3.2.1
Java
keypress4_sketch
1
2 void setup() {
3 size(200, 200) ;
4 background(255) ;
5 }
6
7 void draw() {
8 }
9
10 void mousePressed() {
11 stroke(0) ;
12 fill(175) ;
13 rectMode(CENTER) ;
14 rect(mouseX, mouseY, 16, 16) ;
15 }
16
17 void keyPressed() {
18 background(255) ;
19 }
20
21
22
Done saving.
Console
Errors
keypress4_sketch

Making UI Buttons

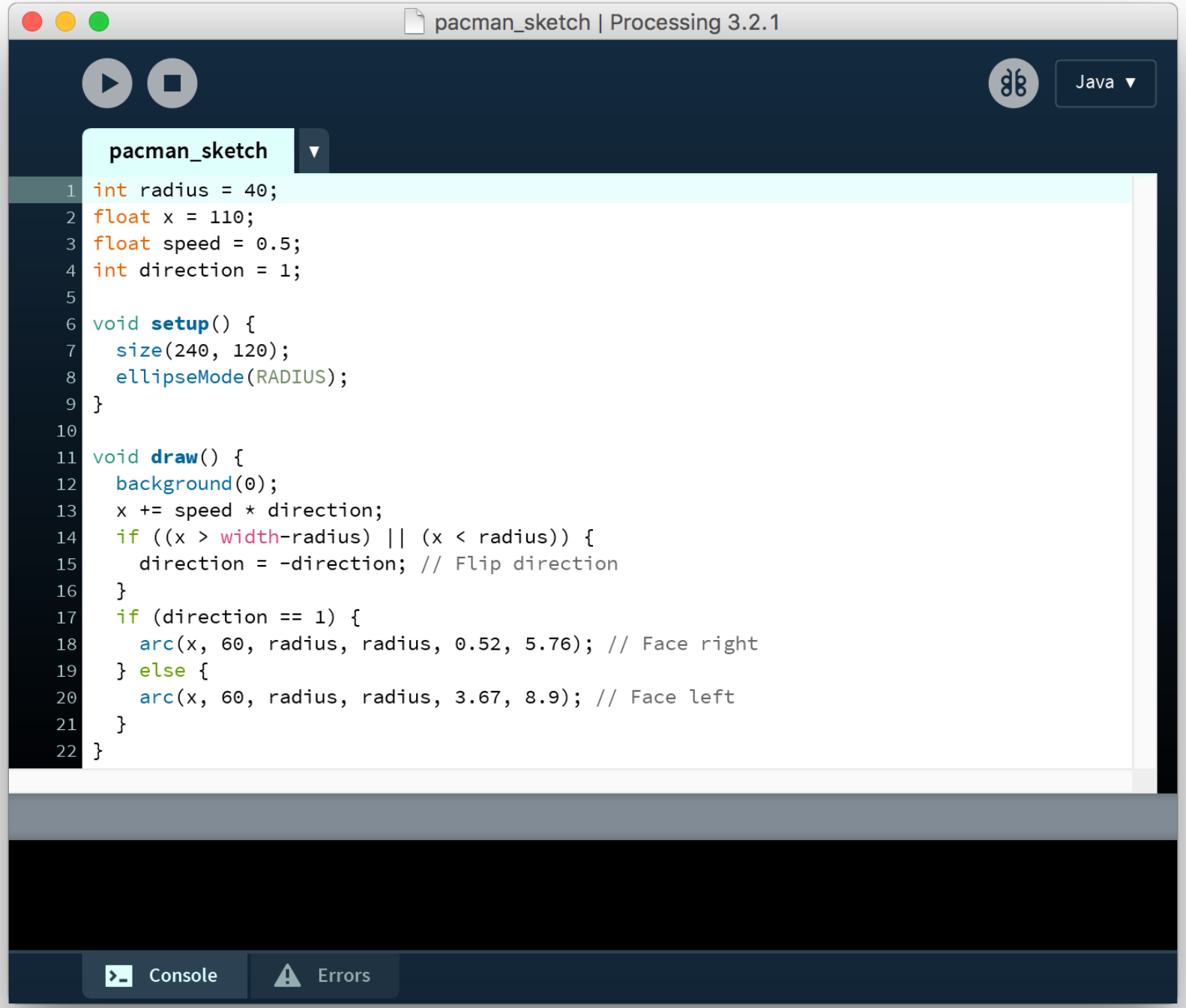



button1_sketch | Processing 3.2.1
Java
button1_sketch
int x = 80;
int y = 30;
int w = 80;
int h = 60;
void setup() {
size(240, 120);
}
void draw() {
background(204);
if ((mouseX > x) && (mouseX < x+w) &&
(mouseY > y) && (mouseY < y+h)) {
fill(0);
} else {
fill(255);
}
rect(x, y, w, h);
}
Console
Errors
button1_sketch

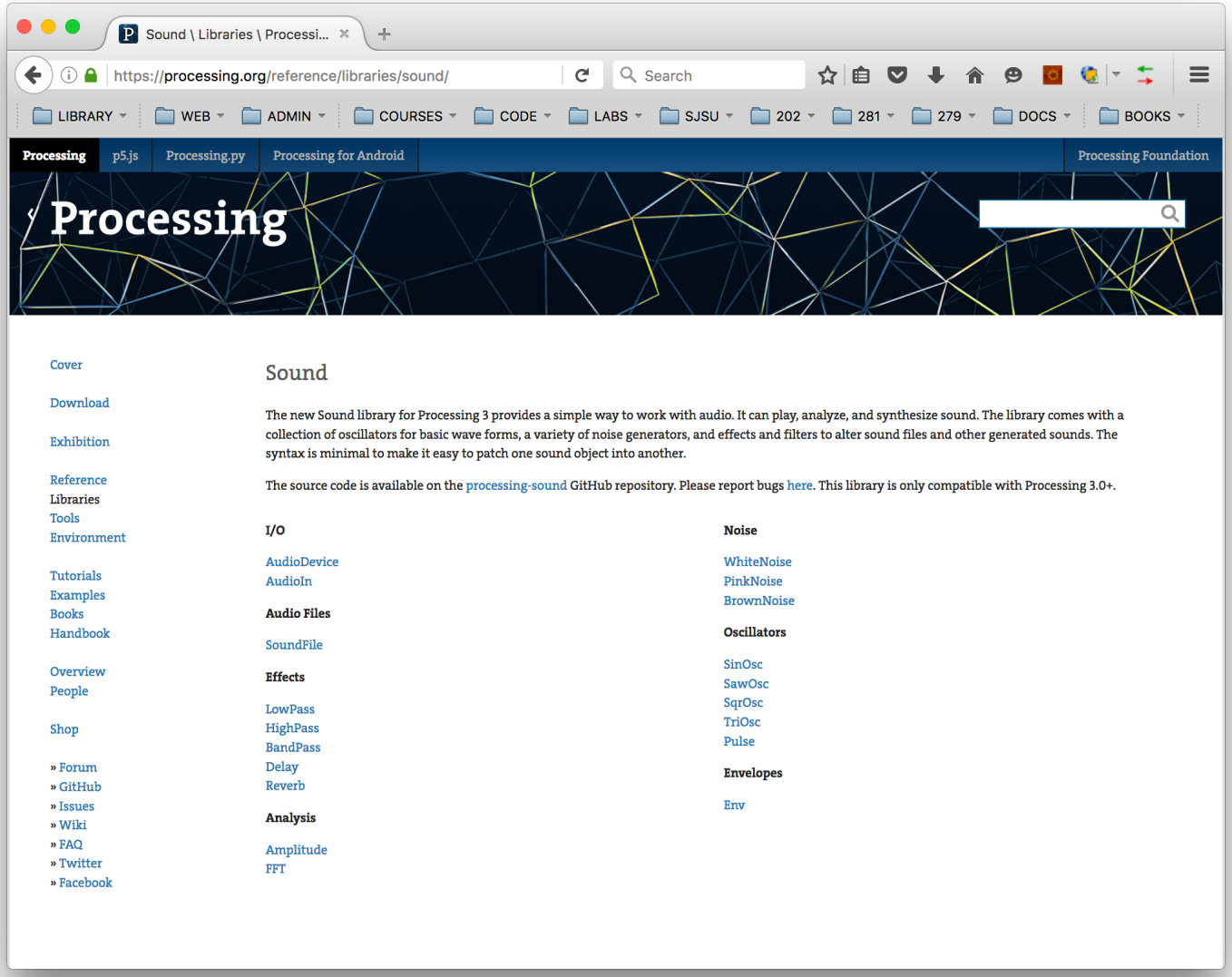

button2_sketch | Processing 3.2.1
Java
button2_sketch
void setup() {
size(240, 120);
strokeWeight(30);
}
void draw() {
background(204);
stroke(102);
line(40, 0, 70, height);
if (mousePressed == true) {
stroke(0);
}
else {
stroke(255);
}
line(0, 70, width, 50);
}
Done saving.
Console
Errors
button2_sketch


button3_sketch | Processing 3.2.1
Java
button3_sketch
void setup() {
size(120, 120);
strokeWeight(30);
}
void draw() {
background(204);
stroke(102);
line(40, 0, 70, height);
if (mousePressed) {
if (mouseButton == LEFT) {
stroke(255);
} else {
stroke(0);
}
line(0, 70, width, 50);
}
}
button...
Console
Errors

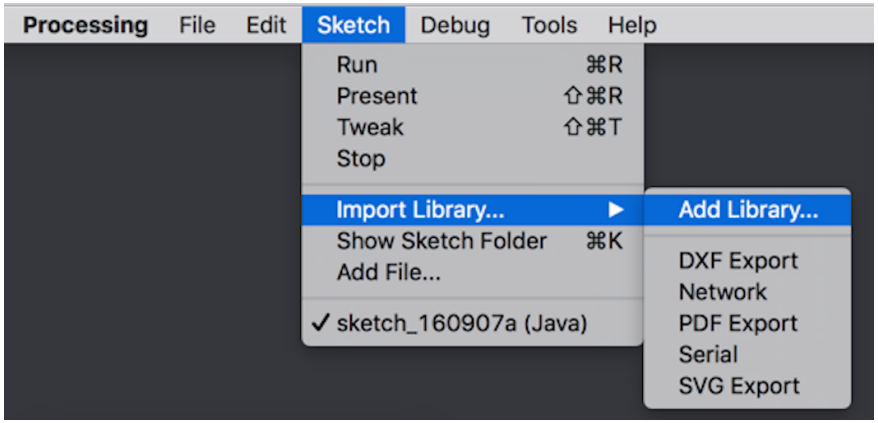
Pacman

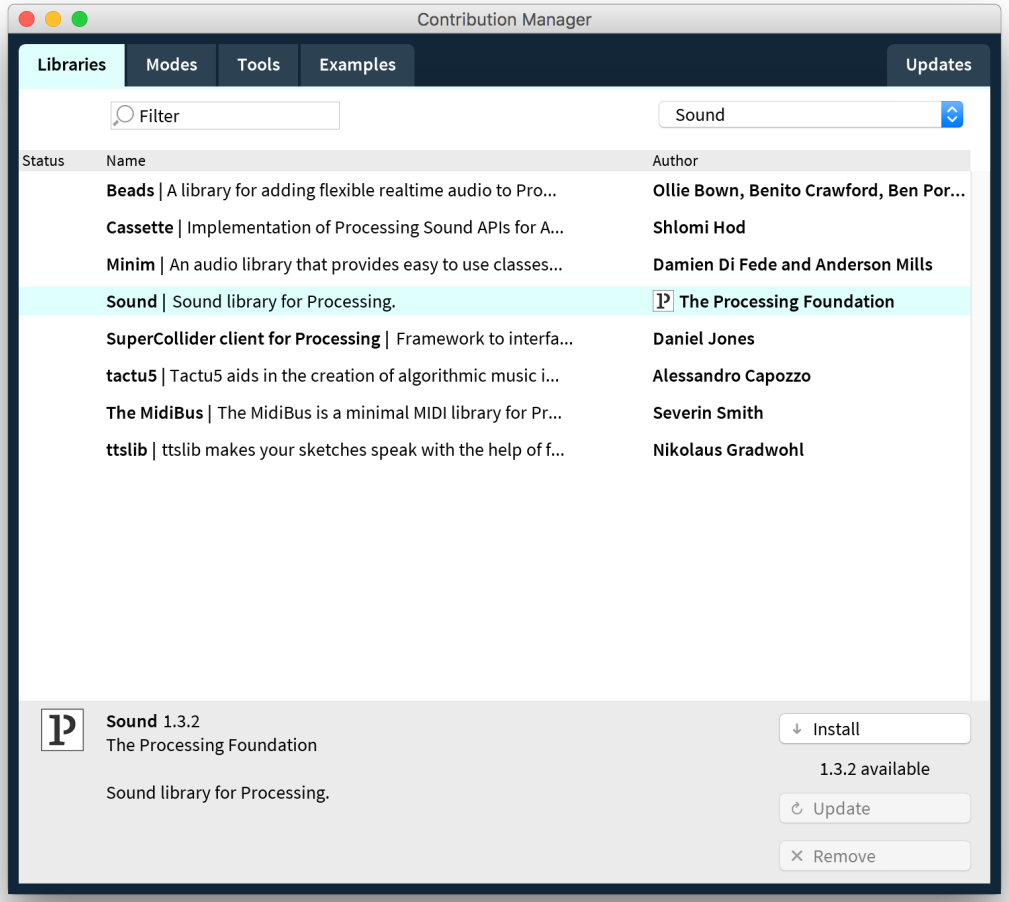

pacman_sketch | Processing 3.2.1
Java
pacman_sketch
int radius = 40;
float x = 110;
float speed = 0.5;
int direction = 1;
void setup() {
size(240, 120);
ellipseMode(RADIUS);
}
void draw() {
background(0);
x += speed * direction;
if ((x > width-radius) || (x < radius)) {
direction = -direction; // Flip direction
}
if (direction == 1) {
arc(x, 60, radius, radius, 0.52, 5.76); // Face right
} else {
arc(x, 60, radius, radius, 3.67, 8.9); // Face left
}
}
Console
Errors

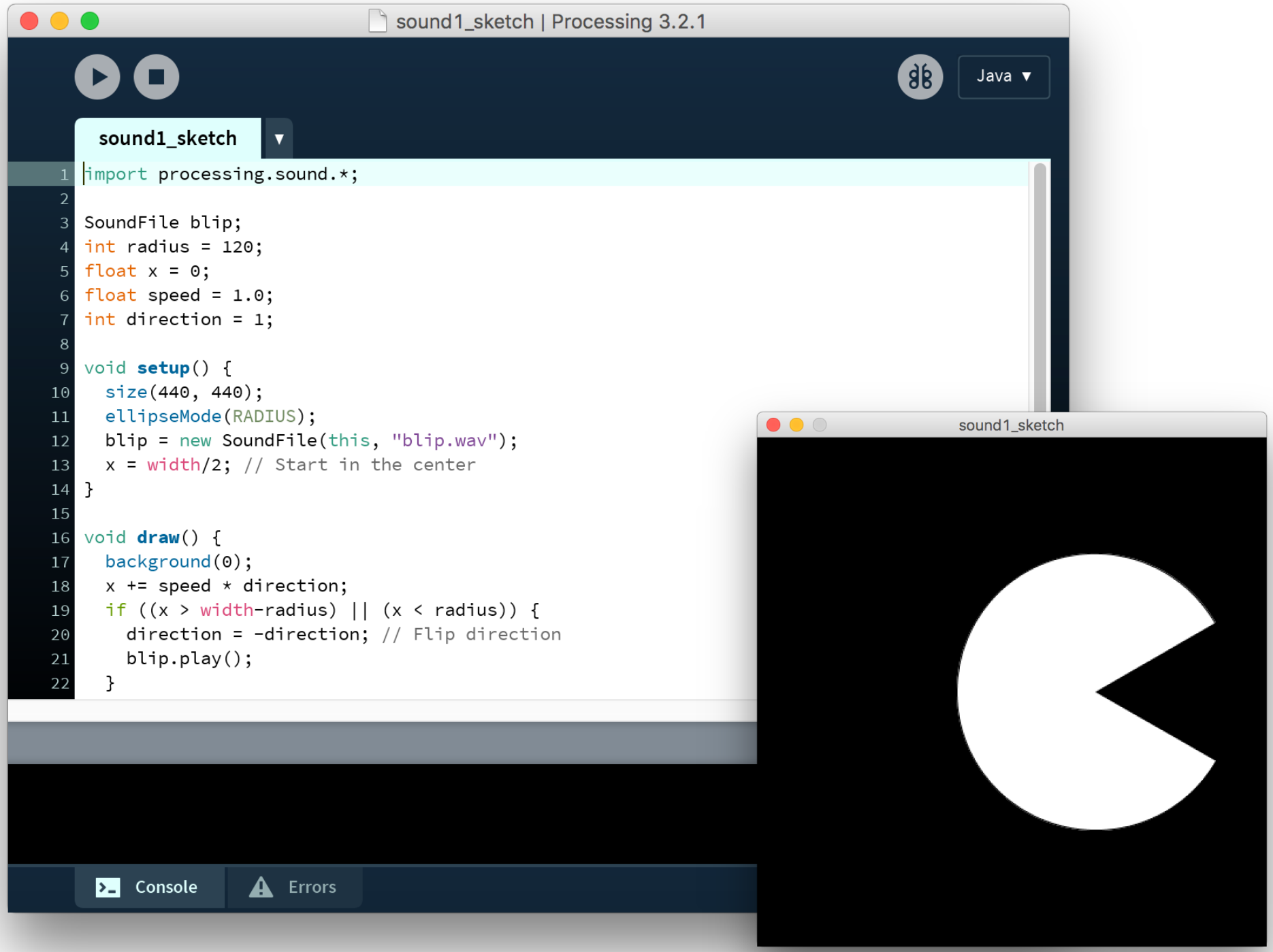
Playing Sound

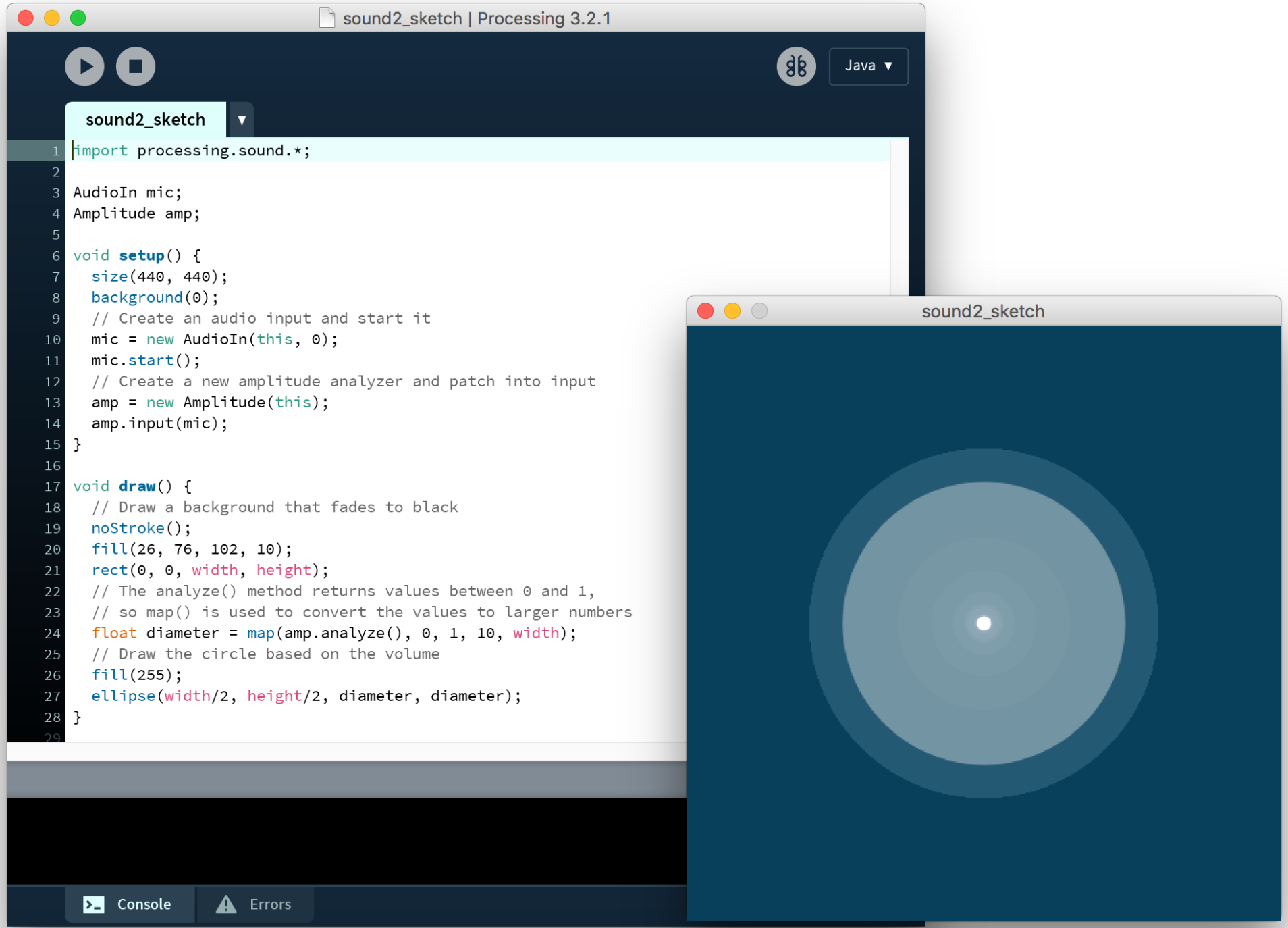


Sound Library Not Installed by Default!

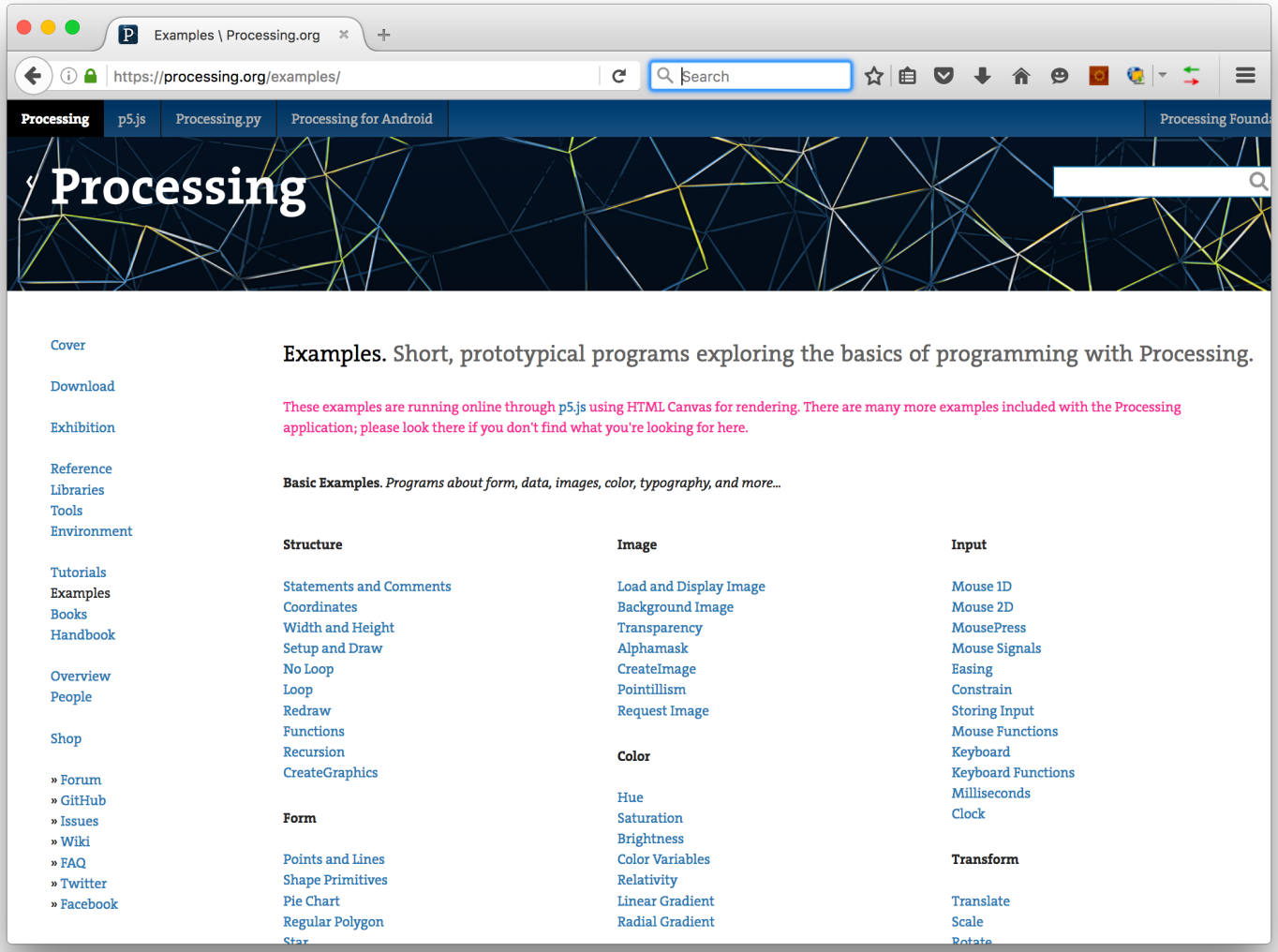




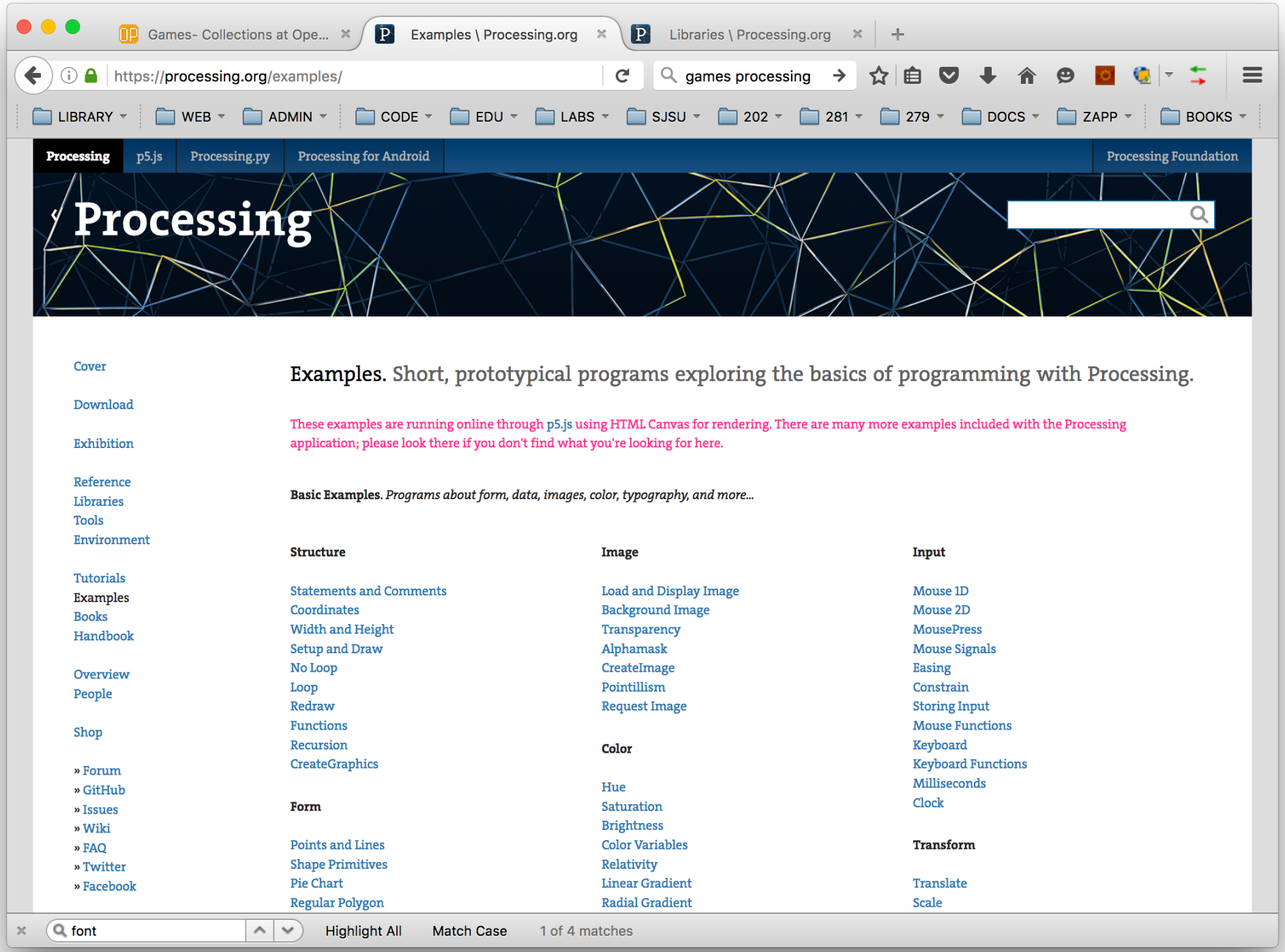

sound1_sketch | Processing 3.2.1
Java
sound1_sketch
import processing.sound.*;
SoundFile blip;
int radius = 120;
float x = 0;
float speed = 1.0;
int direction = 1;
void setup() {
size(440, 440);
ellipseMode(RADIUS);
blip = new SoundFile(this, "blip.wav");
x = width/2; // Start in the center
}
void draw() {
background(0);
x += speed * direction;
if ((x > width-radius) || (x < radius)) {
direction = -direction; // Flip direction
blip.play();
}
Console
Errors
sound1_sketch


sound2_sketch | Processing 3.2.1
Java
sound2_sketch
import processing.sound.*;
AudioIn mic;
Amplitude amp;
void setup() {
size(440, 440);
background(0);
// Create an audio input and start it
mic = new AudioIn(this, 0);
mic.start();
// Create a new amplitude analyzer and patch into input
amp = new Amplitude(this);
amp.input(mic);
}
void draw() {
// Draw a background that fades to black
noStroke();
fill(26, 76, 102, 10);
rect(0, 0, width, height);
// The analyze() method returns values between 0 and 1,
// so map() is used to convert the values to larger numbers
float diameter = map(amp.analyze(), 0, 1, 10, width);
// Draw the circle based on the volume
fill(255);
ellipse(width/2, height/2, diameter, diameter);
}
Console
Errors
sound2_sketch

More Examples



https://processing.org/examples/


Games- Collections at Ope...
Examples \ Processing.org
Libraries \ Processing.org
https://processing.org/examples/
games processing
LIBRARY
WEB
ADMIN
CODE
EDU
LABS
SJSU
202
281
279
DOCS
ZAPP
BOOKS
Processing
p5.js
Processing.py
Processing for Android
Processing Foundation
Processing
Cover
Download
Exhibition
Reference
Libraries
Tools
Environment
Tutorials
Examples
Books
Handbook
Overview
People
Shop
» Forum
» GitHub
» Issues
» Wiki
» FAQ
» Twitter
» Facebook
Examples. Short, prototypical programs exploring the basics of programming with Processing.
These examples are running online through p5.js using HTML Canvas for rendering. There are many more examples included with the Processing application; please look there if you don't find what you're looking for here.
Basic Examples. Programs about form, data, images, color, typography, and more...
Structure
Statements and Comments
Coordinates
Width and Height
Setup and Draw
No Loop
Loop
Redraw
Functions
Recursion
CreateGraphics
Form
Points and Lines
Shape Primitives
Pie Chart
Regular Polygon
Image
Load and Display Image
Background Image
Transparency
Alphamask
CreateImage
Pointillism
Request Image
Color
Hue
Saturation
Brightness
Color Variables
Relativity
Linear Gradient
Radial Gradient
Input
Mouse 1D
Mouse 2D
MousePress
Mouse Signals
Easing
Constrain
Storing Input
Mouse Functions
Keyboard
Keyboard Functions
Milliseconds
Clock
Transform
Translate
Scale
font
Highlight All
Match Case
1 of 4 matches

Games- Collections at Ope... | Examples \ Processing.org | Libraries \ Processing.org

www.openprocessing.org/collection/25

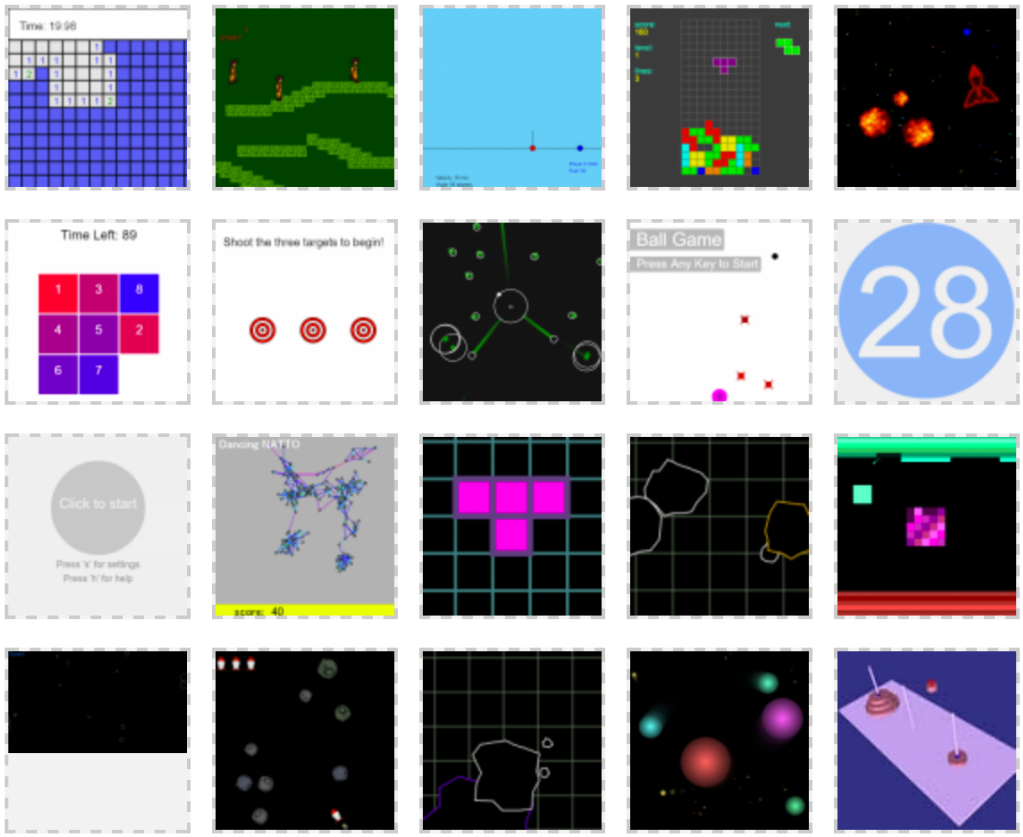
games processing

LIBRARY | WEB | ADMIN | CODE | EDU | LABS | SJSU | 202 | 281 | 279 | DOCS | ZAPP | BOOKS

OpenProcessing

browse classrooms collections books go plus+ jobs search

login - sign up



sketches from Games

This collection includes games that are built with Processing. Enjoy! (and feel free to submit sketches that I missed)

collection created by Sinan Ascioglu

includes sketches by Art Merrideth (4), oggy (2), Greg Wittmann (3), Lambert Wang (4), Mark Gillespie (1), jacques maire (1), Ricardo Gambirasio (1), Jarod Searle (1), Yuri Socher Bichibichi (1), Tetsuya Matsuno (1), MacKenzie Bates (1), Juan Irache (1), Robert D. (1), N. Strang (1), haroflow (1), adrian trevino (1)

submit a sketch from your portfolio

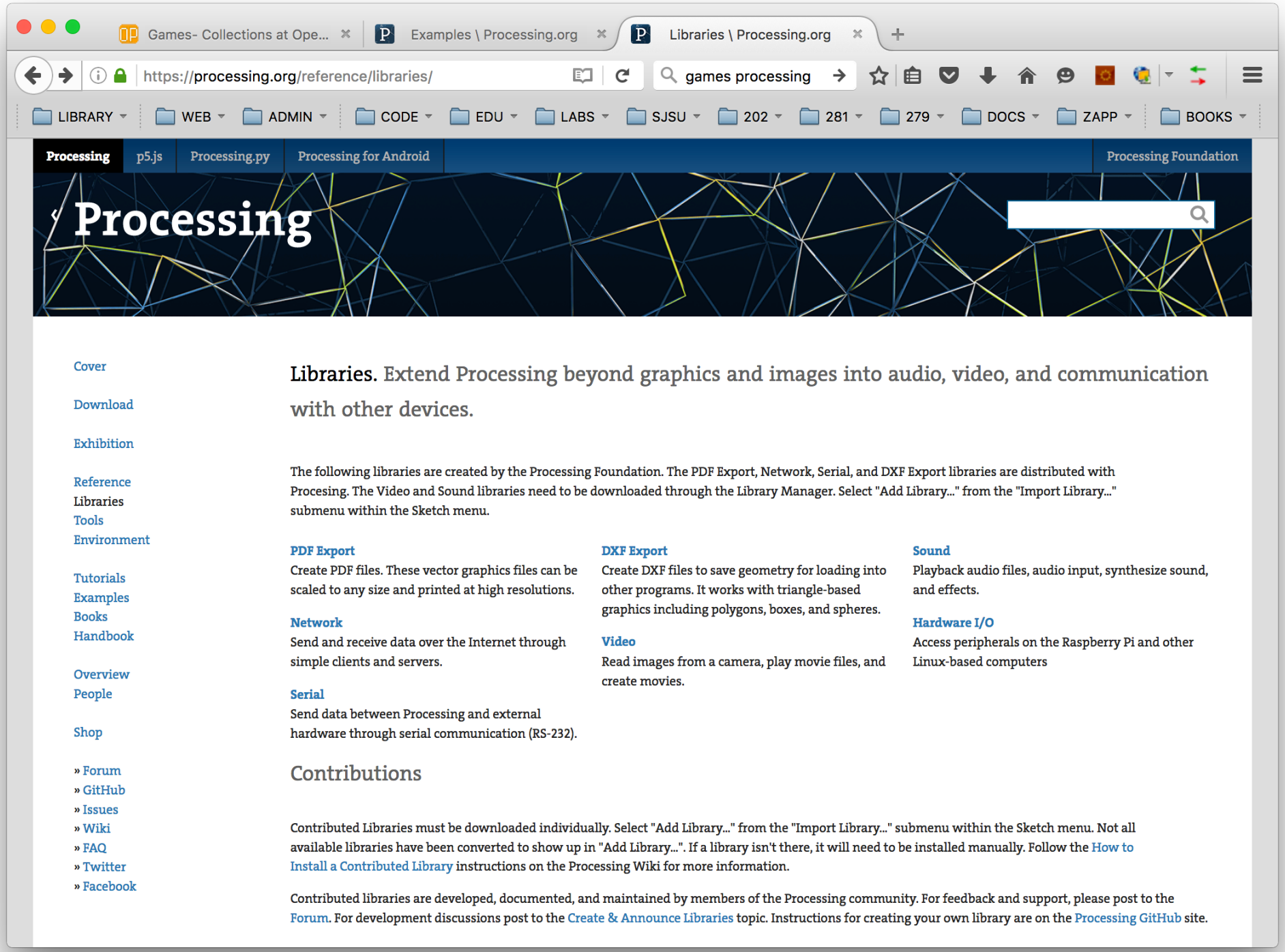
Select a Sketch

or you can enter the visualID of the sketch below

http://openprocessing.org/sketch/ Go

Share Subscribe to RSS

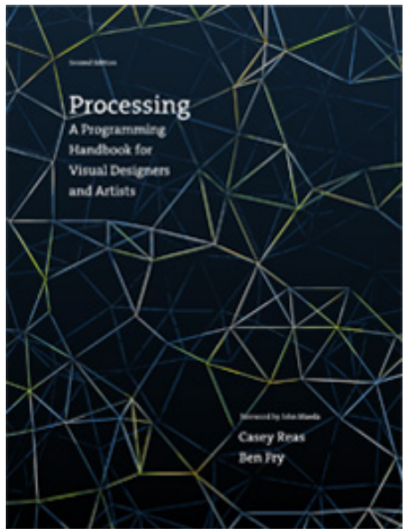
Font Highlight All Match Case Phrase not found


Games- Collections at Ope...
Examples \ Processing.org
Libraries \ Processing.org
https://processing.org/reference/libraries/
games processing
LIBRARY
WEB
ADMIN
CODE
EDU
LABS
SJSU
202
281
279
DOCS
ZAPP
BOOKS
Processing
p5.js
Processing.py
Processing for Android
Processing Foundation
Processing
Cover
Download
Exhibition
Reference
Libraries
Tools
Environment
Tutorials
Examples
Books
Handbook
Overview
People
Shop
» Forum
» GitHub
» Issues
» Wiki
» FAQ
» Twitter
» Facebook
Libraries. Extend Processing beyond graphics and images into audio, video, and communication with other devices.
The following libraries are created by the Processing Foundation. The PDF Export, Network, Serial, and DXF Export libraries are distributed with Procesing. The Video and Sound libraries need to be downloaded through the Library Manager. Select "Add Library..." from the "Import Library..." submenu within the Sketch menu.
PDF Export
Create PDF files. These vector graphics files can be scaled to any size and printed at high resolutions.
Network
Send and receive data over the Internet through simple clients and servers.
Serial
Send data between Processing and external hardware through serial communication (RS-232).
DXF Export
Create DXF files to save geometry for loading into other programs. It works with triangle-based graphics including polygons, boxes, and spheres.
Video
Read images from a camera, play movie files, and create movies.
Sound
Playback audio files, audio input, synthesize sound, and effects.
Hardware I/O
Access peripherals on the Raspberry Pi and other Linux-based computers
Contributions
Contributed Libraries must be downloaded individually. Select "Add Library..." from the "Import Library..." submenu within the Sketch menu. Not all available libraries have been converted to show up in "Add Library...". If a library isn't there, it will need to be installed manually. Follow the How to Install a Contributed Library instructions on the Processing Wiki for more information.
Contributed libraries are developed, documented, and maintained by members of the Processing community. For feedback and support, please post to the Forum. For development discussions post to the Create & Announce Libraries topic. Instructions for creating your own library are on the Processing GitHub site.

Processing

Cover
Download
Exhibition
Reference
Libraries
Tools
Environment
Tutorials
Examples
Books
Handbook
Overview
People
Shop
» Forum
» GitHub
» Issues
» Wiki
» FAQ
» Twitter
» Facebook

Books. Processing books cover topics from programming basics to visualization. Browse this page to find the right books for you.



Processing: A Programming Handbook for Visual Designers, Second Edition
Casey Reas and Ben Fry.
Published December 2014, The MIT Press. 720 pages. Hardcover.
» Order from MIT Press
» Order from Amazon

The second edition of the Handbook has been thoroughly updated, influenced by the seven years of Processing being taught in classrooms, computer labs, and studios since the first edition. Every chapter has been revised, and added chapters introduce new ways to work with data and geometry. New "synthesis" chapters offer discussion and worked examples of such topics as sketching with code, modularity, and algorithms. Interviews have been added that cover a wider range of projects and history. "Extension" chapters are now offered online so they can be updated to keep pace with technological developments in such fields as computer vision and electronics.

If you are an educator, you can request a desk/exam copy from the MIT Press website.

Make: Getting Started with Processing, Second Edition
Casey Reas and Ben Fry.
Published September 2015, Maker Media. 238 pages. Paperback.

Getting Started with Proce... | Contact Us: Step 1 of 2 | T... | Learning Processing 2nd E...

shop.oreilly.com/product/0636920031406.do

Search

LIBRARY | WEB | ADMIN | CODE | EDU | LABS | SJSU | 202 | 281 | 279 | DOCS | ZAPP | BOOKS

O'REILLY

Shop Books & Videos

Search

Your Account

Shopping Cart 0 items $0.00

Home | Shop Video Training & Books | Radar | Safari Books Online | Conferences

Browse Subjects | Learning Paths | Video Training | New | Upcoming | Early Release | Bestselling | Ebooks

1-800-889-8969 / 707-827-7019 / orders@oreilly.com

Search Inside and Read

Larger Cover

Getting Started with Processing, 2nd Edition

A Hands-On Introduction to Making Interactive Graphics

By Casey Reas, Ben Fry

Publisher: Maker Media, Inc
Final Release Date: September 2015
Pages: 238

★★★★★ 5.0

Read 1 Review | Write a Review

Processing opened up the world of programming to artists, designers, educators, and beginners. This short book gently introduces the core concepts of computer programming and working with Processing. Written by the co-founders of the Processing project, Reas and Fry, Getting Started with Processing shows you how easy it is to...

Full description

Buying Options

Immediate Access - Go Digital what's this?

Ebook: **$12.99** Add to Cart
Formats: DAISY, ePub, Mobi, PDF

Print & Ebook: **$27.49** Add to Cart

Print: **$24.99** 1 Add to Cart

Safari Books Online - Read now >

Essential Links

To Order

Books: Call toll-free (800) 405-1619 in the US and Canada. Call (401) 658-4226 from South America and Asia.

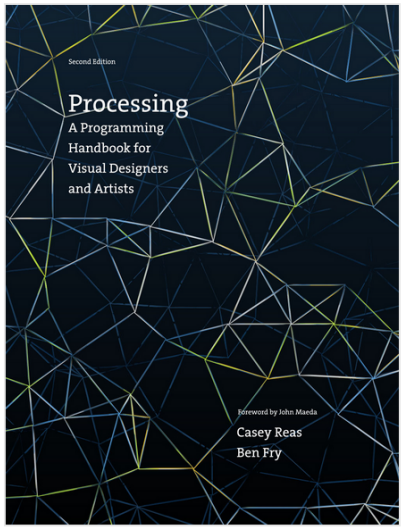
Journals: Call (617) 253-2889 for worldwide ordering. To see complete contact info, Use the contact form at right.

General Press Inquiries

The MIT Press
One Rogers Street
Cambridge, MA 02142-1209 USA
Tel: (617) 253-5646
Fax: (617) 258-6779

Textbook Request: Step 1 of 2

You are requesting:



Processing, Second Edition

by Casey Reas and Ben Fry

Please tell us the region from which you are requesting this textbook and/or instructor materials.

- ○ United States

Getting Started with Proce... | Contact Us: Step 1 of 2 | T... | Learning Processing 2nd E...

learningprocessing.com

Search

LIBRARY | WEB | ADMIN | CODE | EDU | LABS | SJSU | 202 | 281 | 279 | DOCS | ZAPP | BOOKS

DANIEL SHIFFMAN
LEARNING PROCESSING
THE NATURE OF CODE

EXAMPLES
EXERCISES
VIDEOS
DOWNLOADS
RESOURCES
GET THE BOOK

LEARNING PROCESSING

2ND EDITION

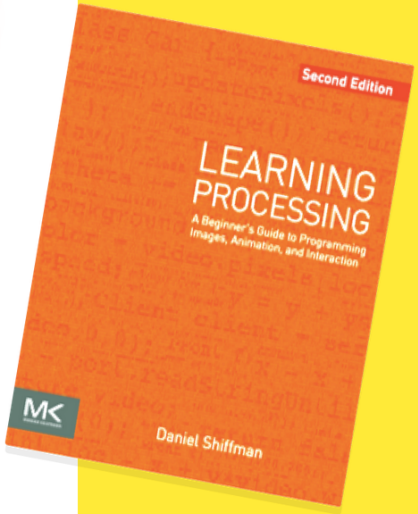
Find a friend in code

This book tells a story. It's a story of liberation, of taking the first steps towards understanding the foundations of computing, writing your own code, and creating your own media without the bonds of existing software tools. This story is not reserved for computer scientists and engineers. This story is for you.

BUY ON AMAZON | DOWNLOAD SAMPLE

CODE FOR ALL EXAMPLES IN PROCESSING
CODE FOR ALL EXAMPLES IN P5.JS

Like Share Tweet



SKETCH MADE WITH P5.JS
WEBSITE ON GITHUB