



Processing

# What exactly is Processing?

---

- Processing is a programming language, development environment, and online community. Since 2001, Processing has promoted software literacy within the visual arts and visual literacy within technology. Initially created to serve as a software sketchbook and to teach computer programming fundamentals within a visual context, Processing evolved into a development tool for professionals. Today, there are tens of thousands of students, artists, designers, researchers, and hobbyists who use Processing for learning, prototyping, and production.

# Processing Features

---

- Free to download and open source
- Interactive programs with 2D, 3D, .jpg, .png, or PDF output
- OpenGL integration for accelerated 3D
- For Linux, Mac OS X, and Windows
- Over 100 libraries so far to extend the core software and functionality
- Well documented, with many books and communities.

# Why Processing? Why code?

---

- mission statement:

Processing seeks to ruin the careers of talented designers by tempting them away from their usual tools and into the world of programming and computation. Similarly, the project is designed to turn engineers and computer scientists to less gainful employment as artists and designers.

- Processing is geared toward creating visual, interactive media for use as a stand-alone program, incorporation into larger projects, or web-based projects.
- You learn the fundamental logic of programming in a fun, quick way. These skills begin to translate into other languages such as html, javascript, and jQuery, making you a stronger artist, designer, and developer.
- It's fun and pretty darn cool.

# Strange Things People Have Done With Processing

---

- Radio Head - House of Cards
- MIT Media Lab's Generative Logo
- Chronograph projected software mural  
for the New World Center in Miami

# Let's Get Making

- Opening Processing takes you to the PDE: Processing Development Environment.
- It is an IDE: Integrated Development Environment.

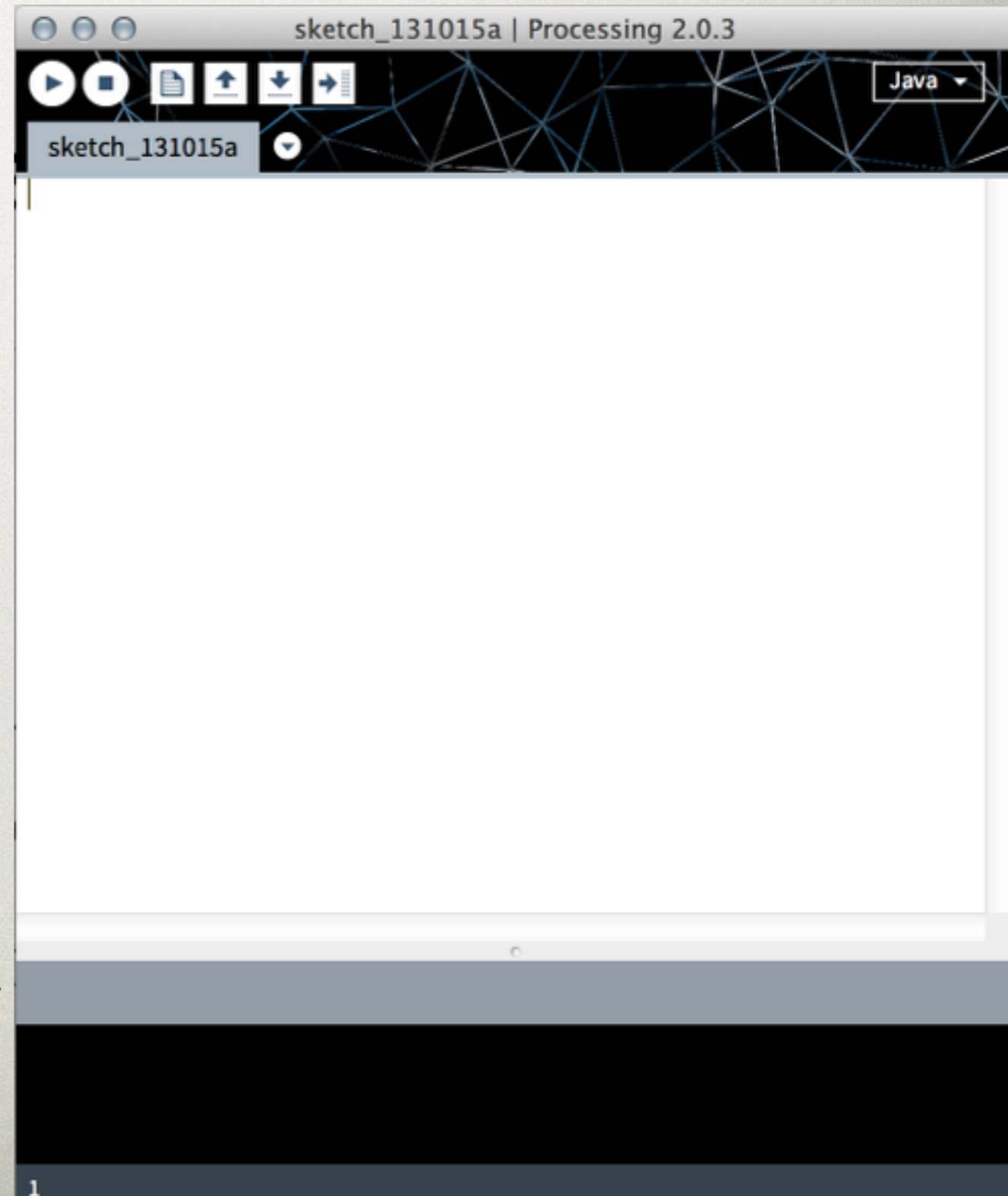
Toolbar ->

Tabs ->

Text editor ->

Message area ->

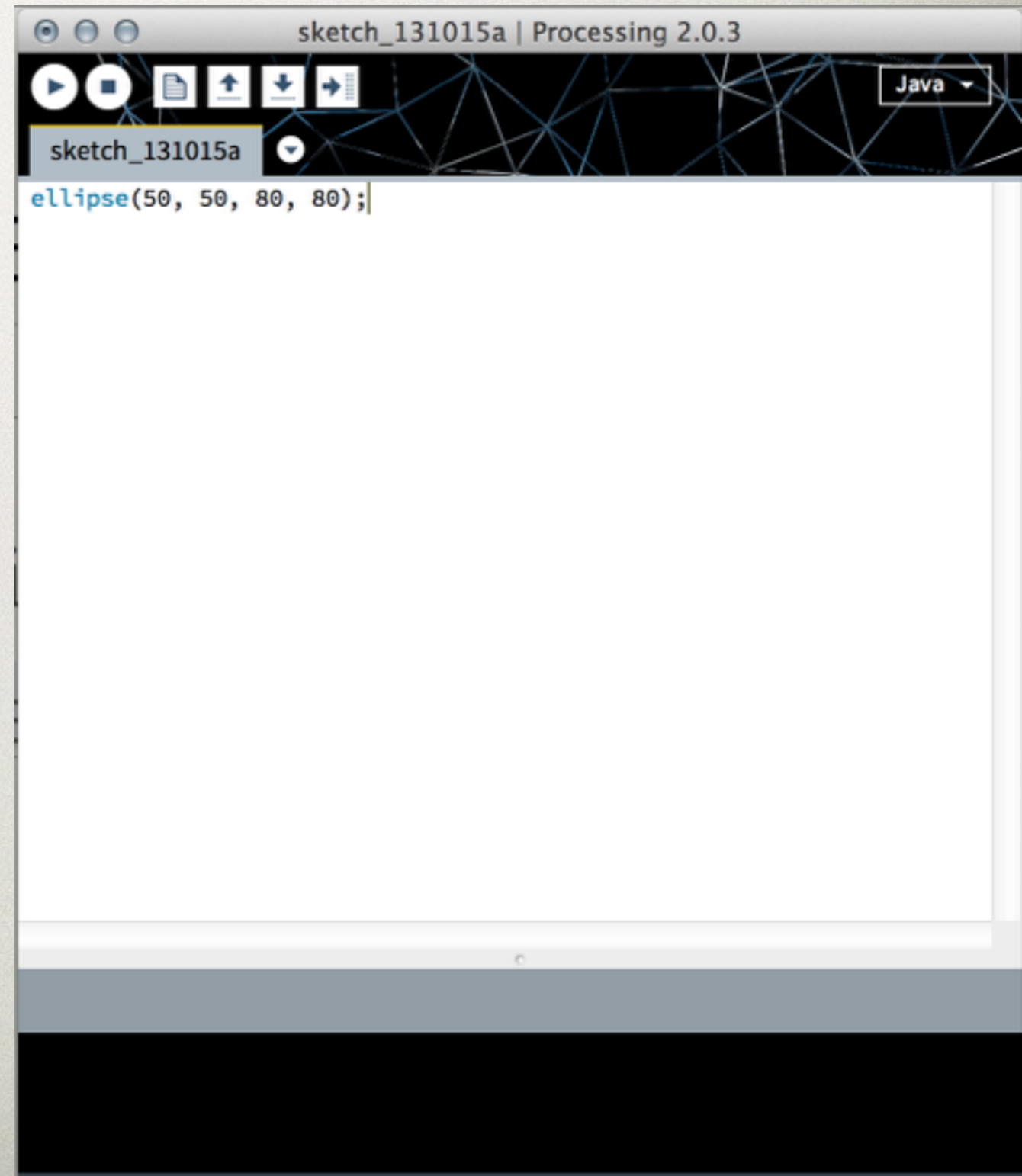
Console ->

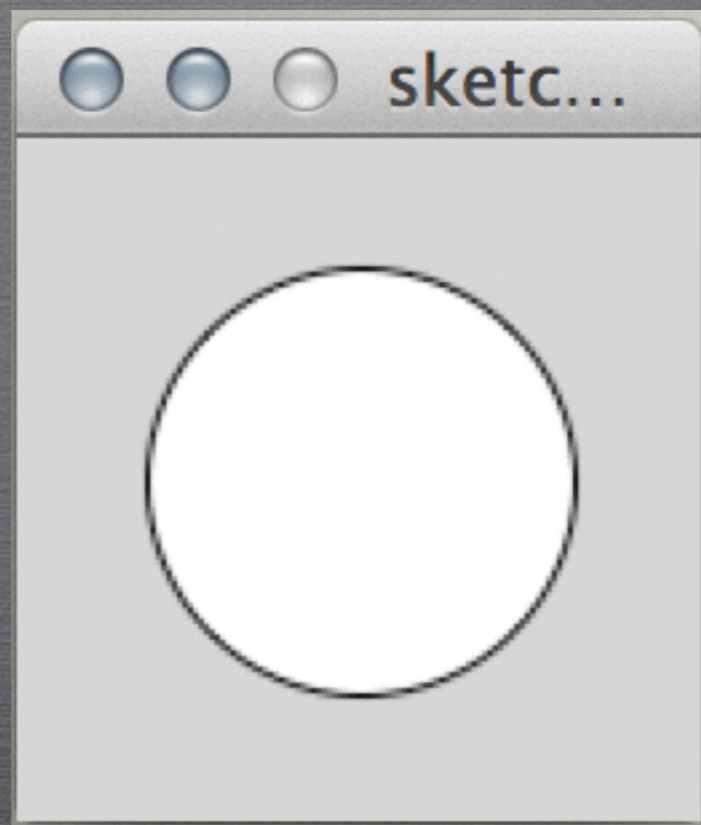


# Hello World!

---

- In the editor, type the following:
- `ellipse(50, 50, 80, 80);`
- Press the play button





You've written a  
computer program!



# I Thought Processing Was Interactive!

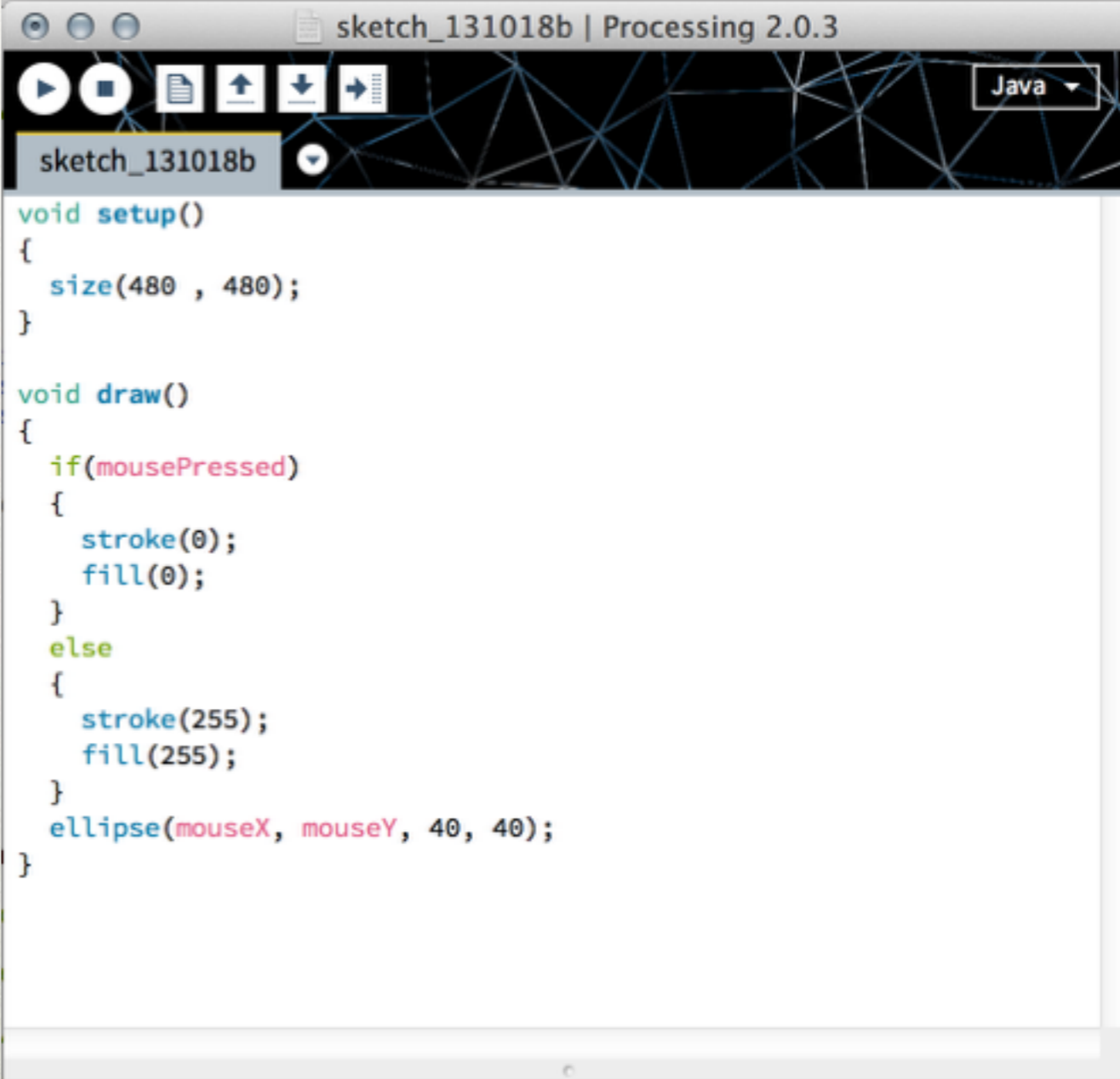
---

- What we just created is static, meaning, it doesn't change over time or involve interactivity.
- By introducing the `setup()` & `draw()` methods along with some built-in functionality, we can create a simple interactive sketch.

# Basic Draw Program

```
void setup()
{
  size(480, 480);
}

void draw()
{
  if(mousePressed)
  {
    stroke(0);
    fill(0);
  }
  else
  {
    stroke(255);
    fill(255);
  }
  ellipse(mouseX, mouseY, 40, 40);
}
```

A screenshot of the Processing IDE window titled "sketch\_131018b | Processing 2.0.3". The window shows a code editor with the following code:

```
void setup()
{
  size(480 , 480);
}

void draw()
{
  if(mousePressed)
  {
    stroke(0);
    fill(0);
  }
  else
  {
    stroke(255);
    fill(255);
  }
  ellipse(mouseX, mouseY, 40, 40);
}
```

The IDE interface includes a toolbar with icons for running, stopping, saving, and other functions, and a language dropdown menu set to "Java".



An Interactive  
Program!

# Let's take This just A Little Further...

```
int value = 0;
void setup()
{
  size(480, 480);
}
void draw()
{
  if(mousePressed)
  {
    if(value==1)
    {
      stroke(value);
      fill(value);
      ellipse(mouseX, mouseY, 40, 40);
    }
    else
    {
      stroke(value);
      fill(value);
      ellipse(mouseX, mouseY, 40, 40);
    }
  }
}
void keyPressed()
{
  if(value==0)
  {
    value=255;
  }
  else
  {
    value=0;
  }
}
```



```
draw_keys | Processing 2.0.3
Java
draw_keys
int value = 0;
void setup()
{
  size(480, 480);
}
void draw()
{
  if(mousePressed)
  {
    if(value==1)
    {
      stroke(value);
      fill(value);
      ellipse(mouseX, mouseY, 40, 40);
    }
    else
    {
      stroke(value);
      fill(value);
      ellipse(mouseX, mouseY, 40, 40);
    }
  }
}
void keyPressed()
{
  if (value == 0)
  {
    value = 255;
  }
  else
  {
    value = 0;
  }
}
```



Mouse & Key  
Interaction

# EXPORT AN APPLICATION

- You can export a .exe file directly from the Processing IDE:  
file ->  
export  
application,  
or use the  
export button  
in your tool  
bar.





Processing.js

# Export a Javascript File file

---

- With the introduction of Processing.js, you can export to a .js file which can then be included along with your .pde file into your website directory for interactive, javascript based canvases.
- You must download Processing.js in addition to Processing.
- Once installed, select JavaScript from the drop down menu at the top right.
- Select export for web.
- <http://processingjs.org/>



# Things to Check Out

---

- Processing has tons of great examples built in for you to run and learn from file -> examples.
- The main websites, [processing.org/](http://processing.org/) [processingjs.org](http://processingjs.org) has great tutorials and examples of its main functions and methods.
- Processing has a great online community. [OpenProcessing.org](http://OpenProcessing.org) is my favorite. You can share sketches and look at other peoples' sketches. You can even look at the source code and since Processing is an open-source project, OpenProcessing allows full access to code and you are encouraged to use what you need to achieve your projects.



Thanks!