Processing:

- Software toolkit that can be used for graphical programming, data visualization, physical computing, games, mobile, web and desktop applications, etc
- You can very quickly draw to the screen in code
- You can pretty much do anything with it
- Started around 2001, java based

Java translation:

- Class extends PApplet
  - Draw, setup, settings need to be implemented
- All methods other than main are not static
- Processing library need to be imported
  - All dependencies exist in libs folder
- main method has special line:
  - PApplet.main("ProcessingTest");

First Processing example

Every processing program we write will have these 3 methods:

```
/** set up size */
public void settings(){
   size(500, 500);
}

/** runs once at the beginning */
public void setup() {
   background(0);
}

/** runs over and over again as long as the program is still running */
public void draw() {
   ellipse(mouseX, mouseY, 30, 30);
}
```

Methods:

1. Settings

Settings is used to set the width and height in pixels for our sketch

```
/** set up size */
public void settings(){
   size(500, 500);
}
```

2. Setup

Setup runs one time and prints to the screen.

```
/** runs once at the beginning */
public void setup() {
   background(0);
}
```

3. Draw

Once setup is done, draw runs *continuously* until the program is closed. Like a while loop. You don't *have* to use this, if you just want to draw once to the screen.

```
/** runs over and over again as long as the program is still running */
public void draw() {
   ellipse(mouseX, mouseY, 30, 30);
}
```

Shapes:

```
// x, y, width, height
ellipse(mouseX, mouseY, 30, 30);

// x, y, width, height
rect(mouseX, mouseY, 30, 30);

// x1, y1, x2, y2
line(mouseX, mouseY, 30, 30);
```

Events:

- mousePressed and keyPressed booleans tell your program when the user clicks (you can also make a mousePressed() method)
- mouseX and mouseY are variables for the current cursor position.

```
if (mousePressed){
   ellipse(mouseX, mouseY, ellipseSize, ellipseSize);
}
```

```
if (keyPressed) {
   if (key == 'b') {
      fill(0);
   }
} else {
   fill(255);
}
ellipse(mouseX, mouseY, ellipseSize, ellipseSize);
```

Pixels:

- pixels[] is a one-dimensional array, so how can we get the "row" and "column" or coords of each pixel?
- index = row * width + columns

```
public void setup(){
     loadPixels();
     for(int row = 0; row < height; row++) {
        for(int column = 0; column < width; column++) {
           int i = row * width + column;
           if (row % 10 == 0) {
              pixels[i] = color(255, 255, 255);
           } else {
              pixels[i] = color(0, 0, 0);
           }
        }
     }
     updatePixels();
  }
```