

# GENERATIVE ART WITH PROCESSING

**Imanol Gómez**  
imanolgomez.net  
13-14 February 2016



# INTRODUCTION

Me

<http://imanolgomez.net/>



# Casey Reas





# Quayola



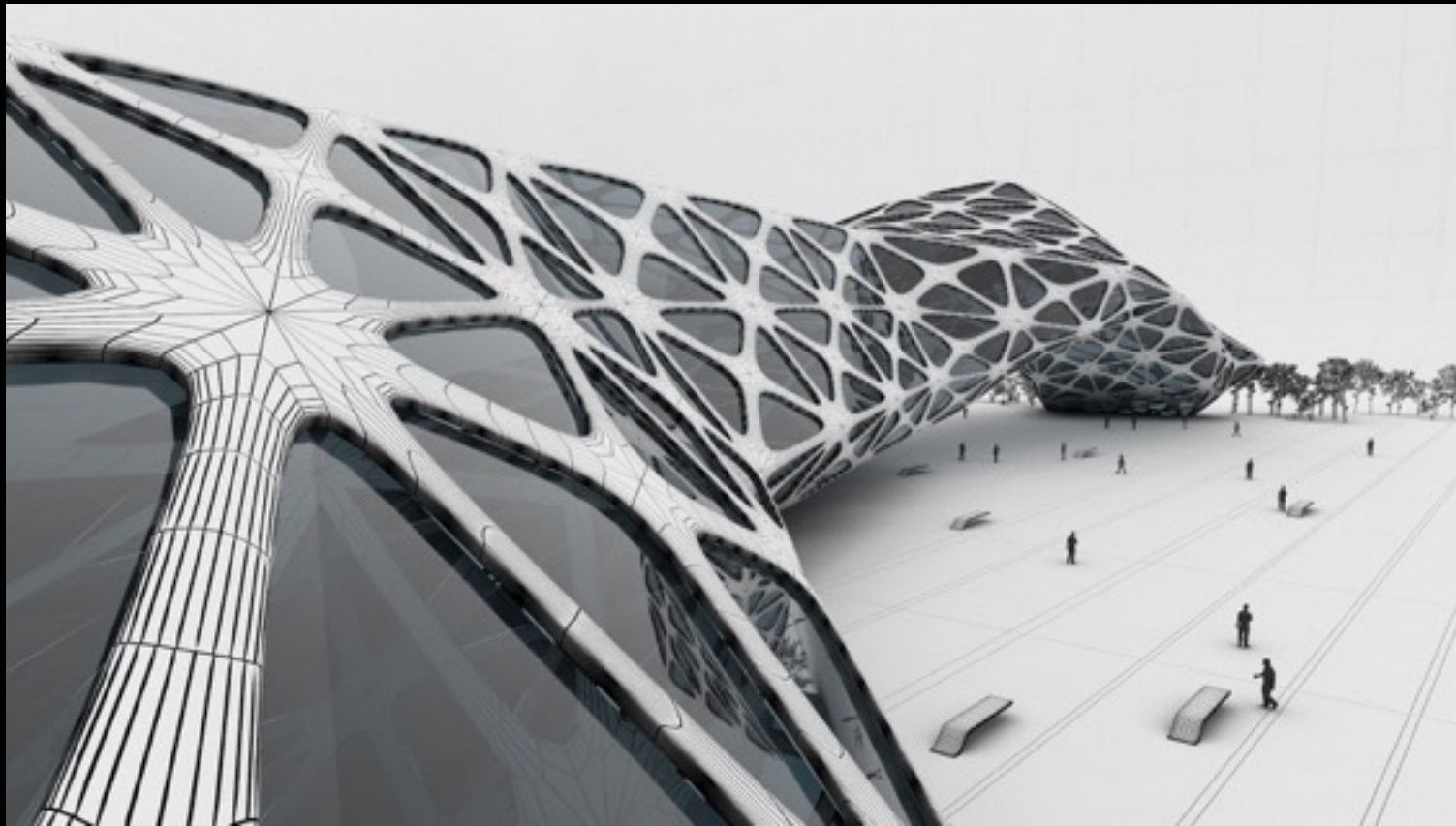


Riyoji Ikeda



# Projects

# Architecture





# Jewellery





# Creative Coding



processing.org





[https://github.com/ImanolGo/  
GenerativeArtWorkshop](https://github.com/ImanolGo/GenerativeArtWorkshop)

# Getting Started





Java ▼

sketch\_151105a ▼

1  
2  
3  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16  
17  
18  
19  
20  
21  
22  
23

Problem

Tab

Line



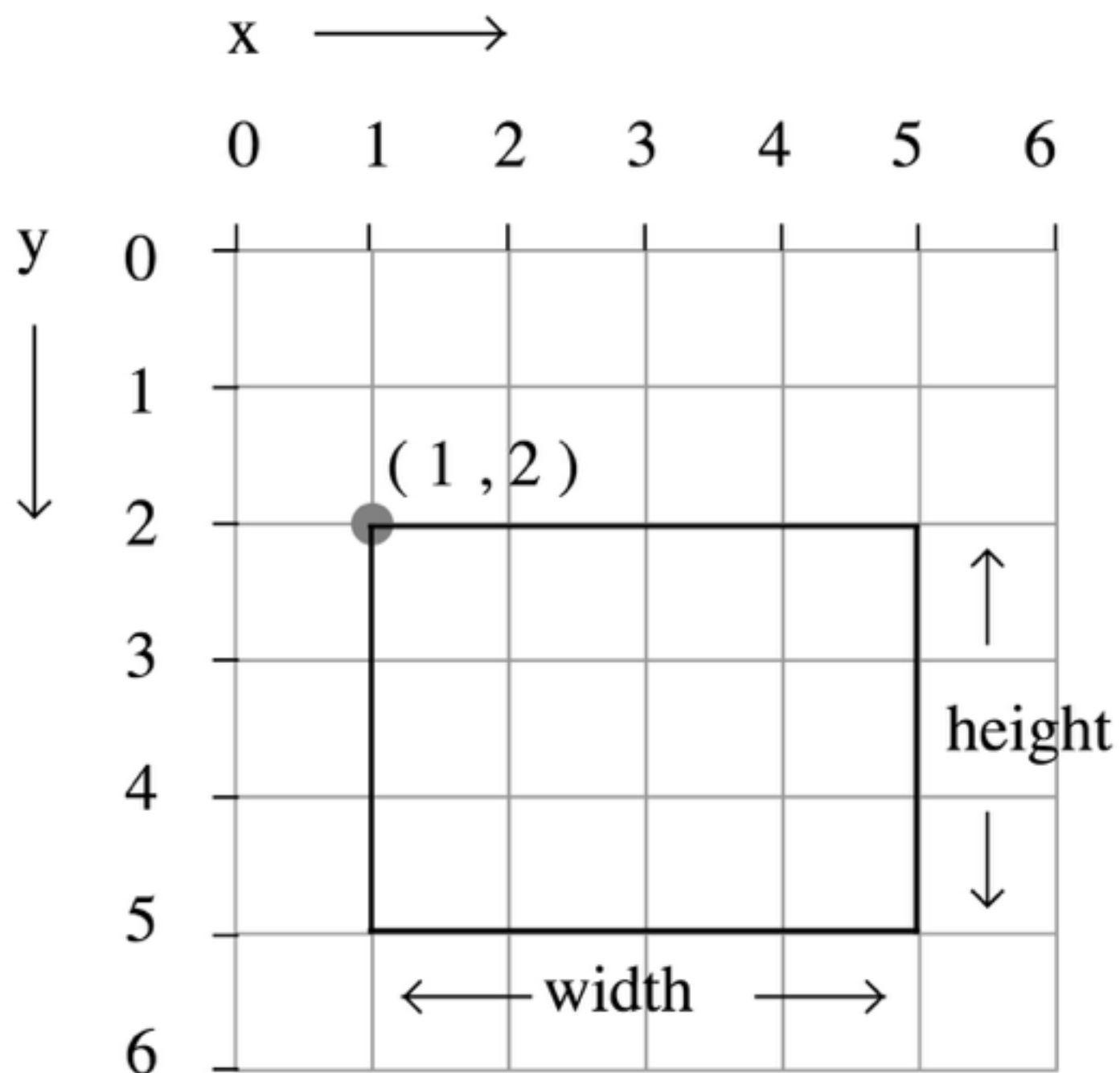
Console



Errors

# I. Shapes

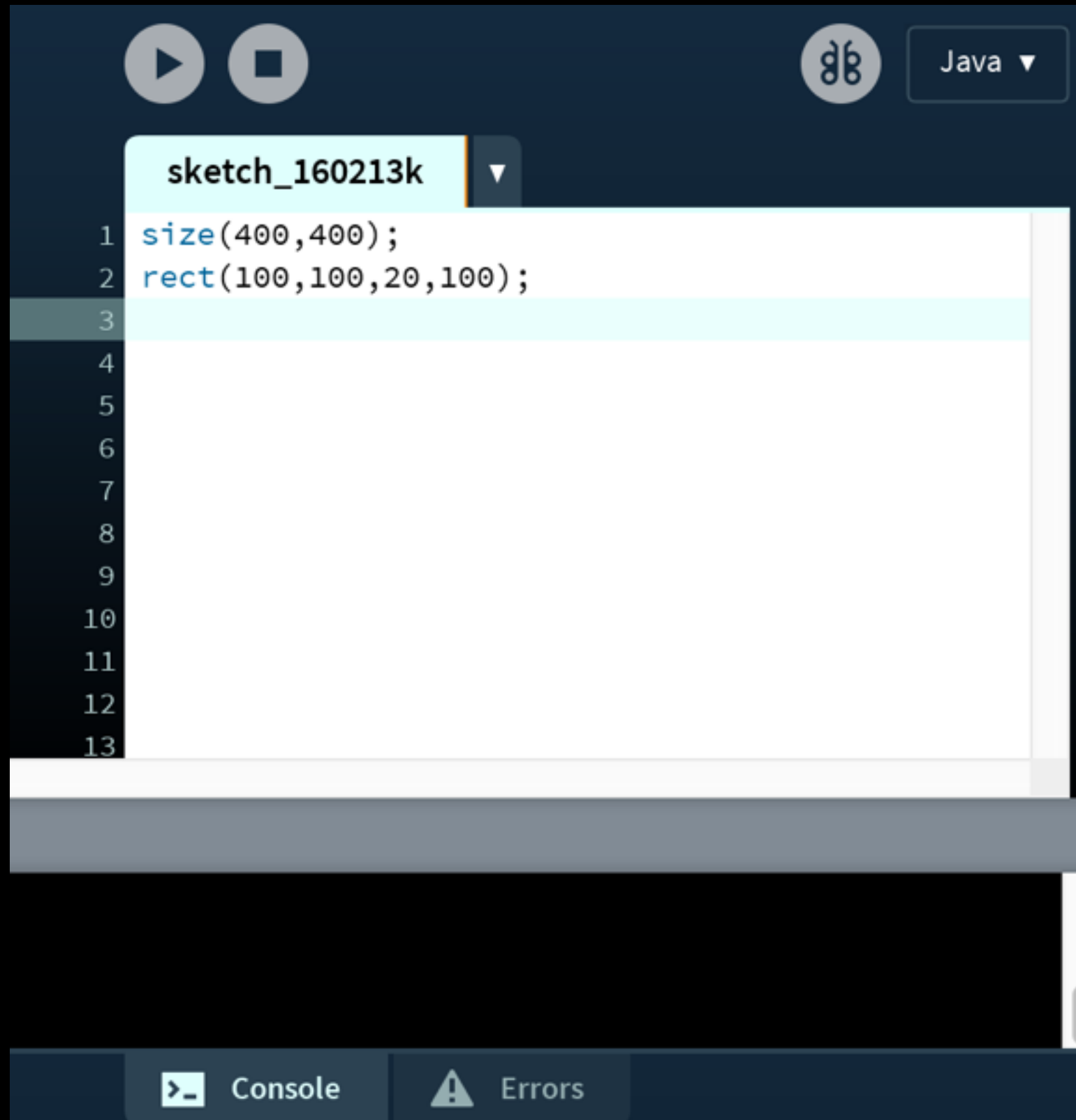




`rect( x , y , width , height ) ;`

Example: `rect ( 1 , 2 , 4 , 3 ) ;`

# size(), rect()







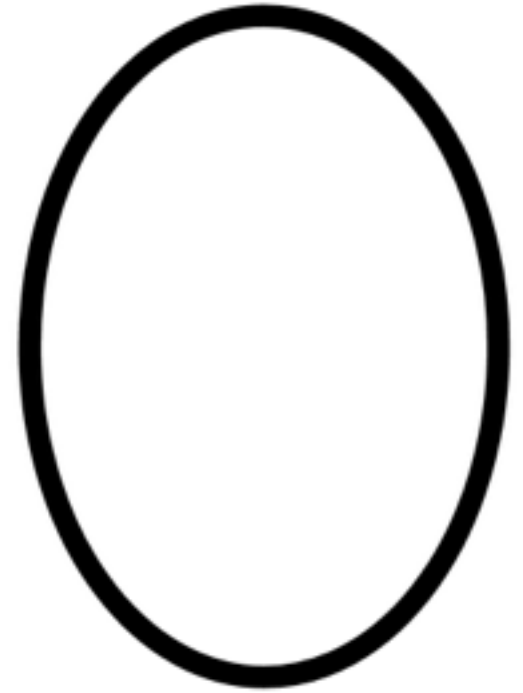
Point



Line



Rectangle

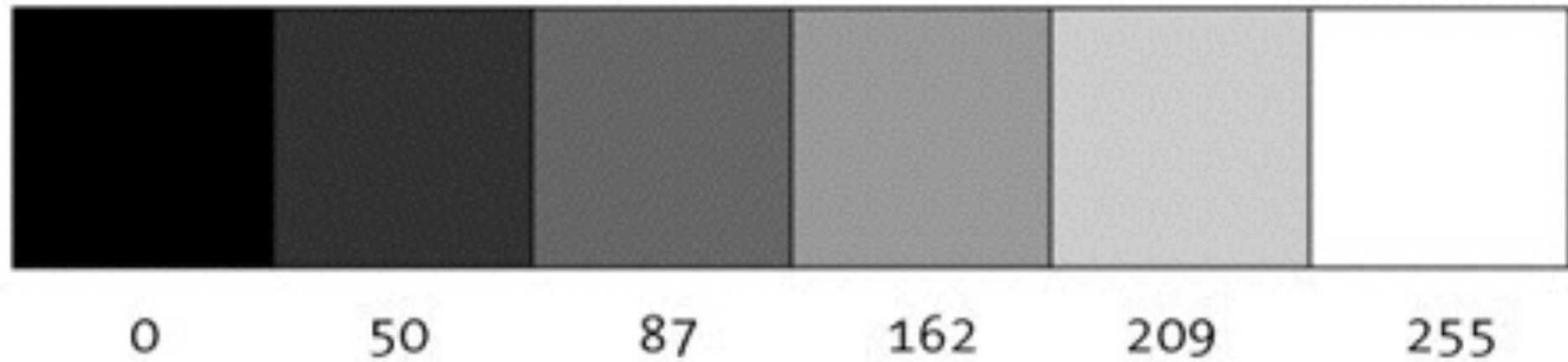


Ellipse

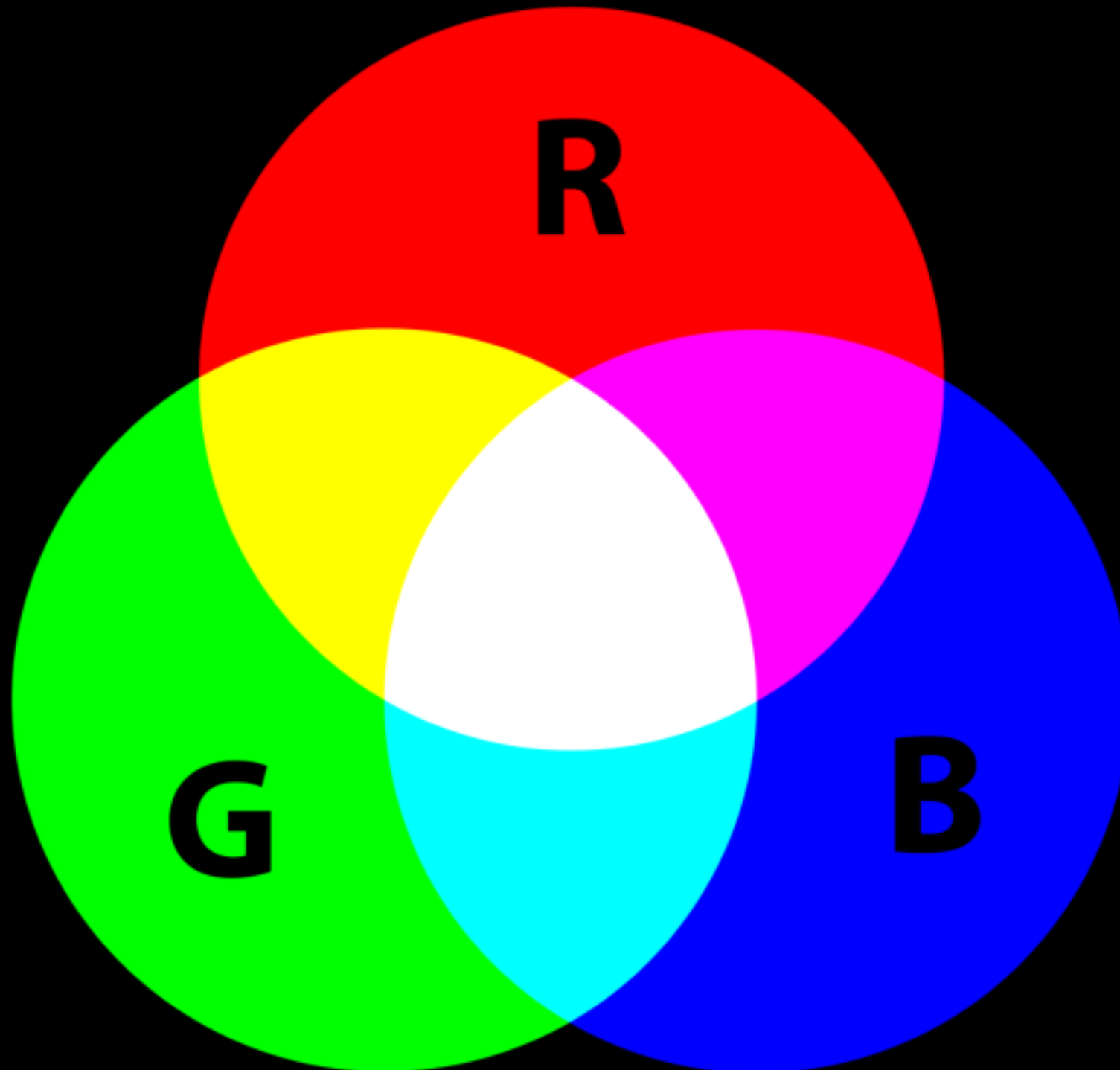
Colour



# Colour values

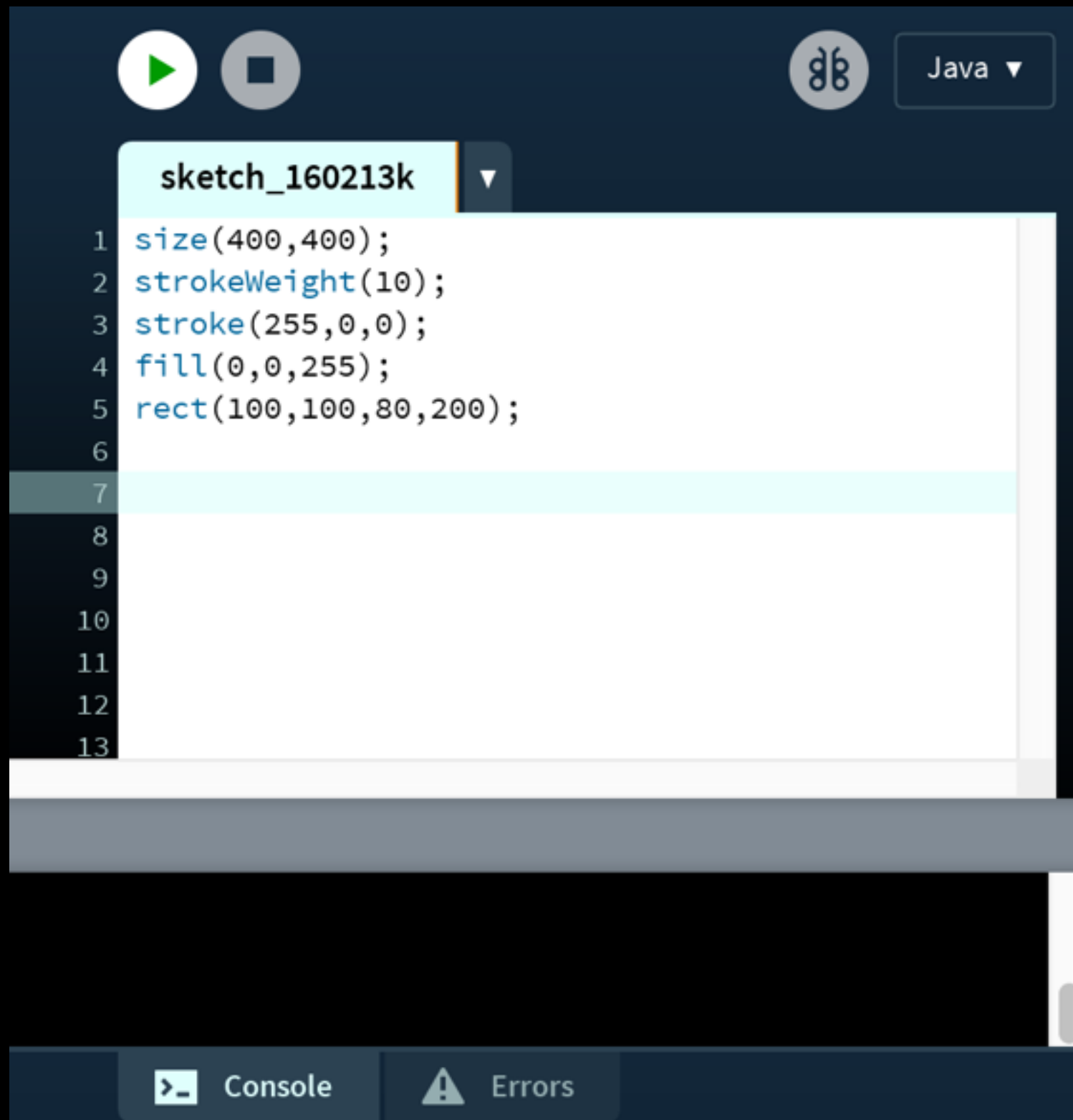


# Red, Green, Blue

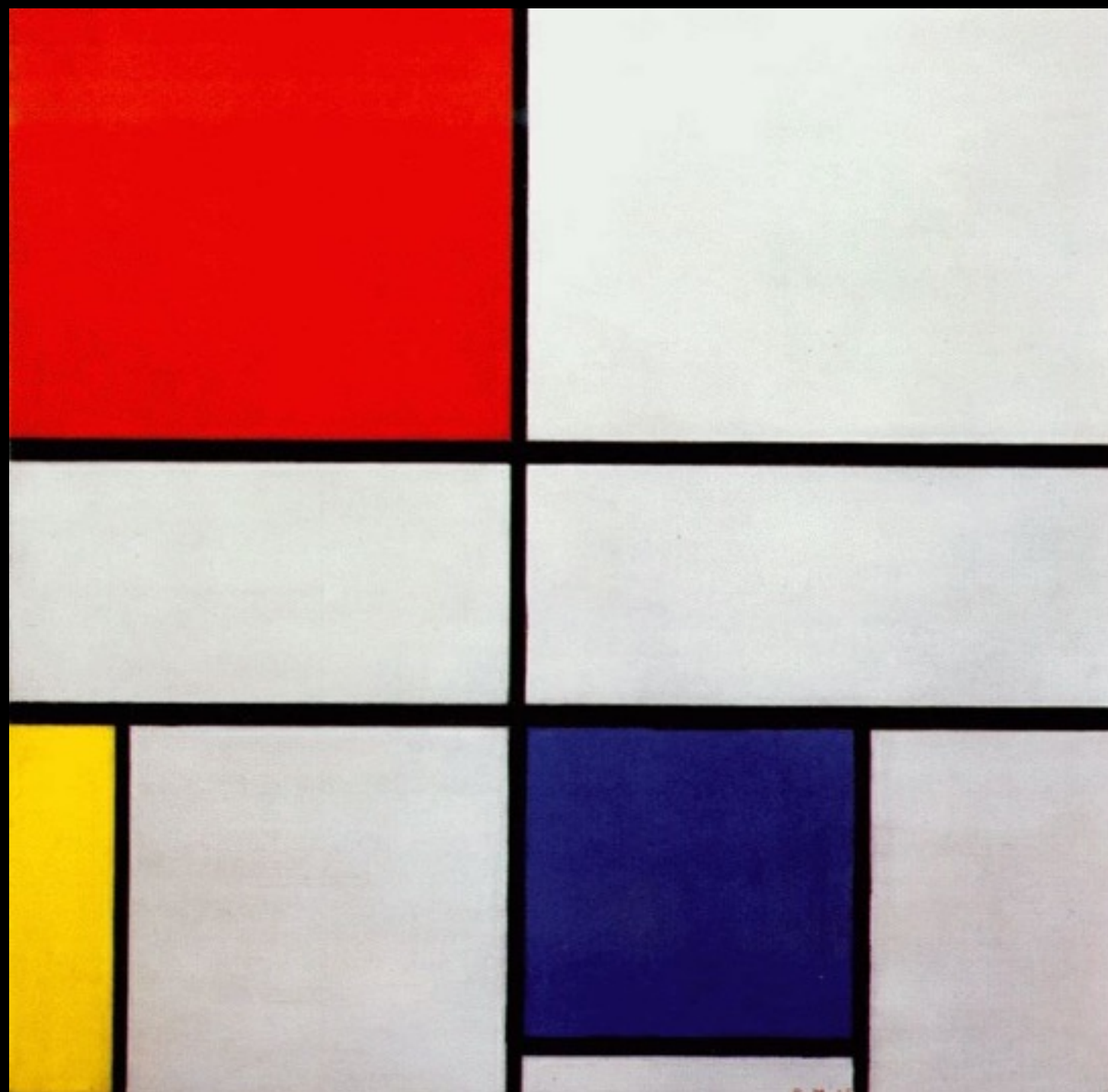




# stroke(), strokeWeight, fill()



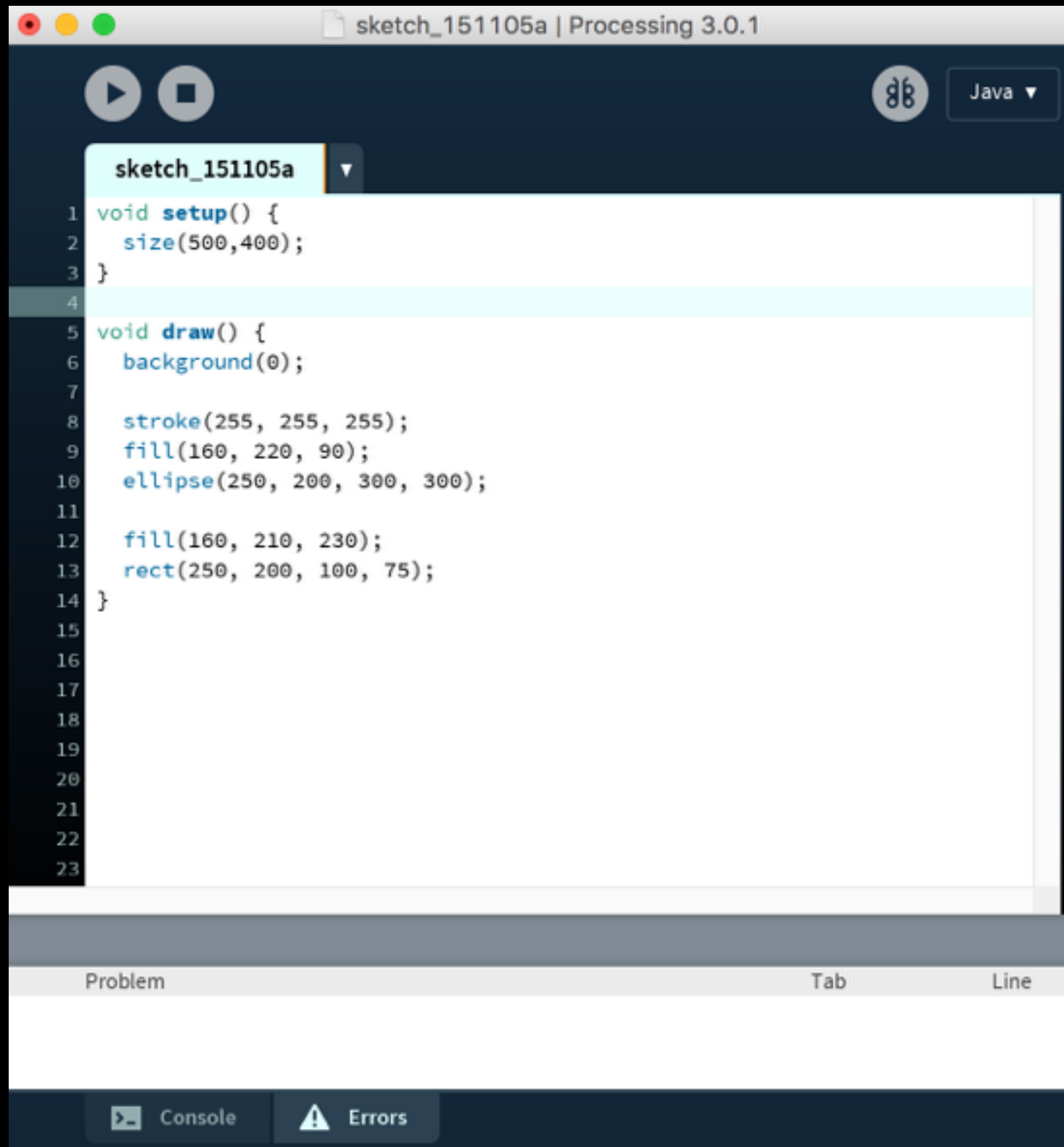
# Exercise: Mondrian



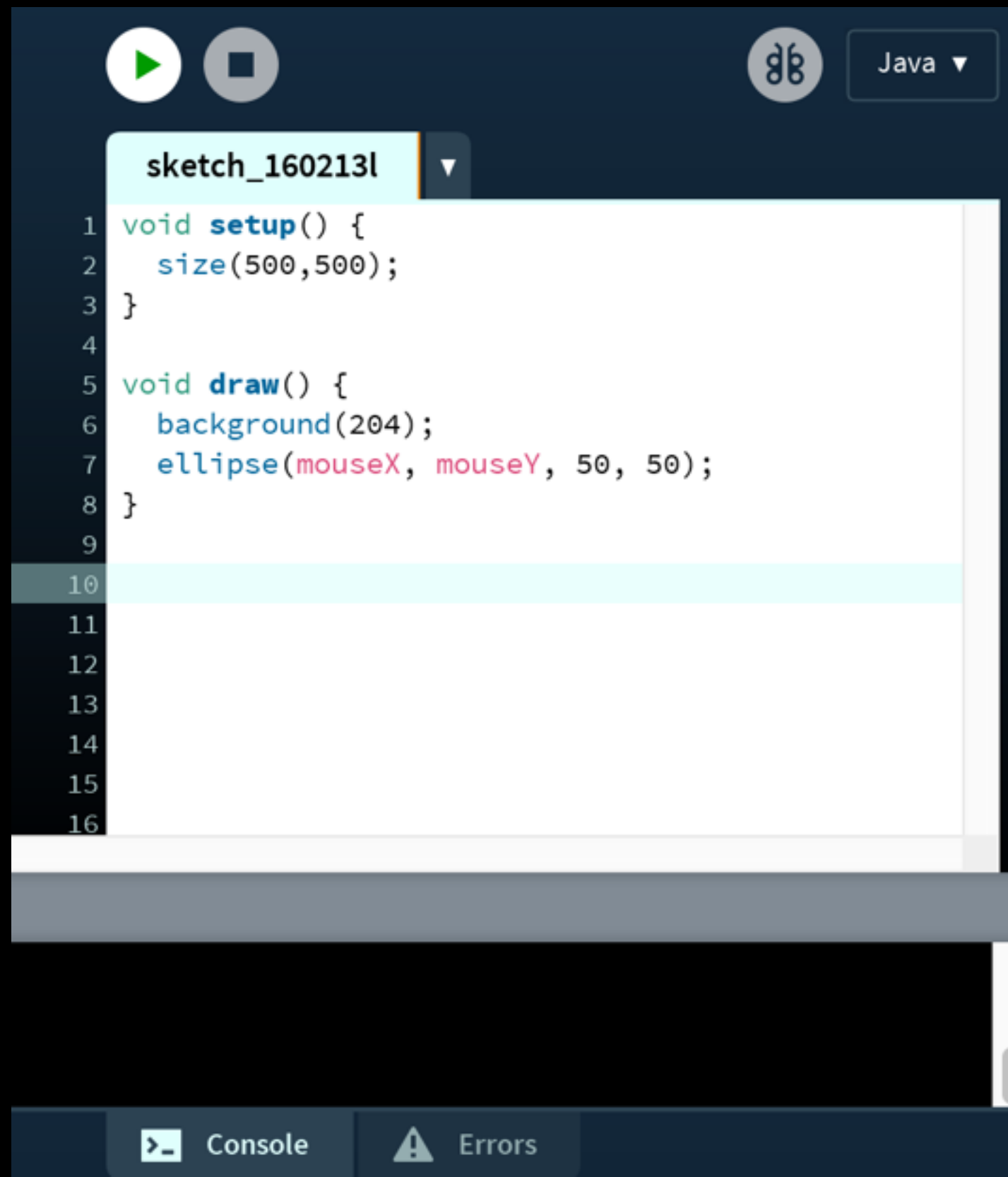
# III. Interactivity



# setup(), draw()



# Mouse

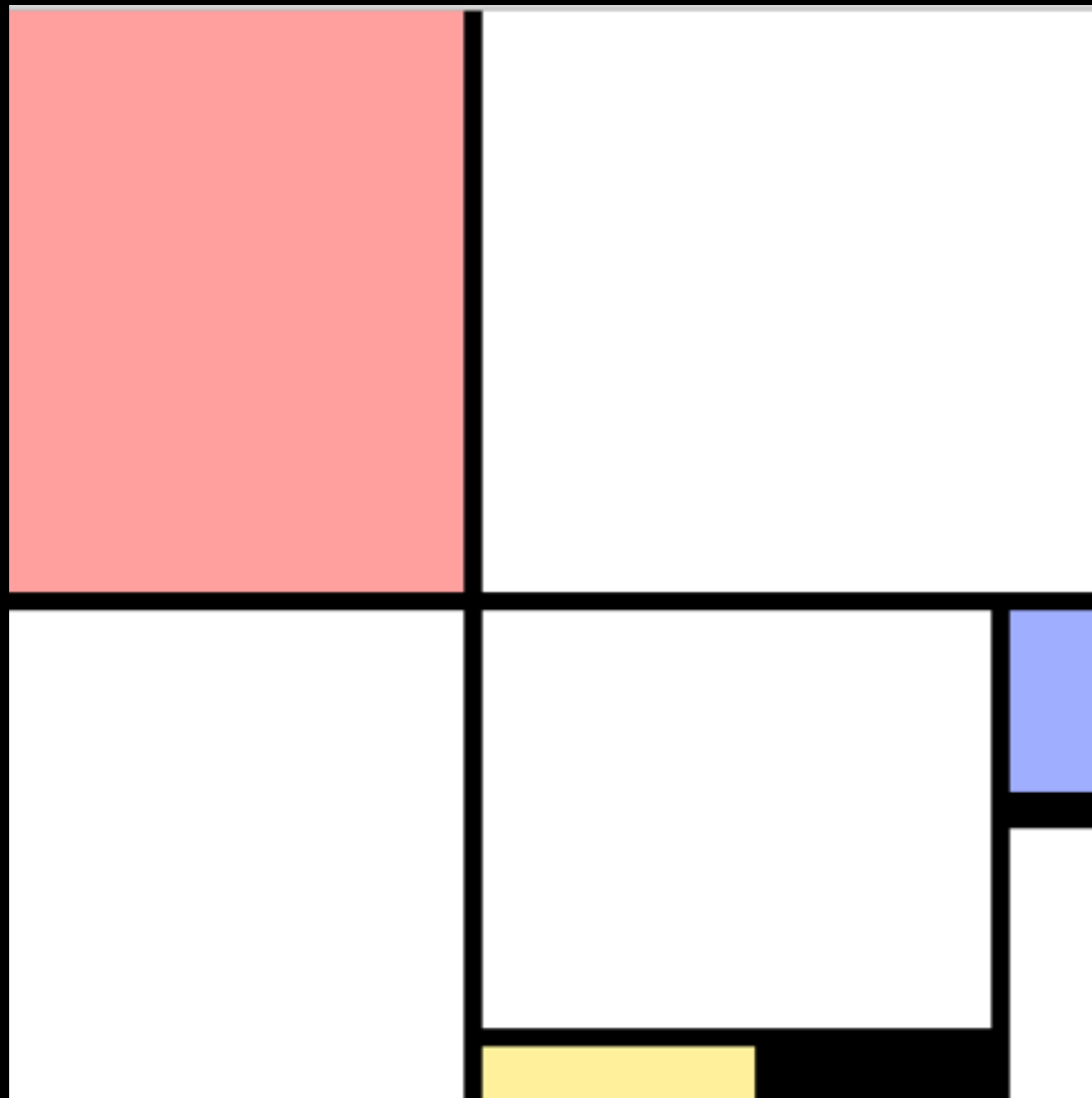


The screenshot shows the Arduino IDE interface. At the top, there are buttons for running (a green play icon) and stopping (a grey square icon). To the right, there is a button with a double-rectangle icon and a dropdown menu labeled 'Java'. Below this, a tab labeled 'sketch\_160213l' is active. The main text area contains the following code:

```
1 void setup() {  
2   size(500,500);  
3 }  
4  
5 void draw() {  
6   background(204);  
7   ellipse(mouseX, mouseY, 50, 50);  
8 }  
9  
10  
11  
12  
13  
14  
15  
16
```

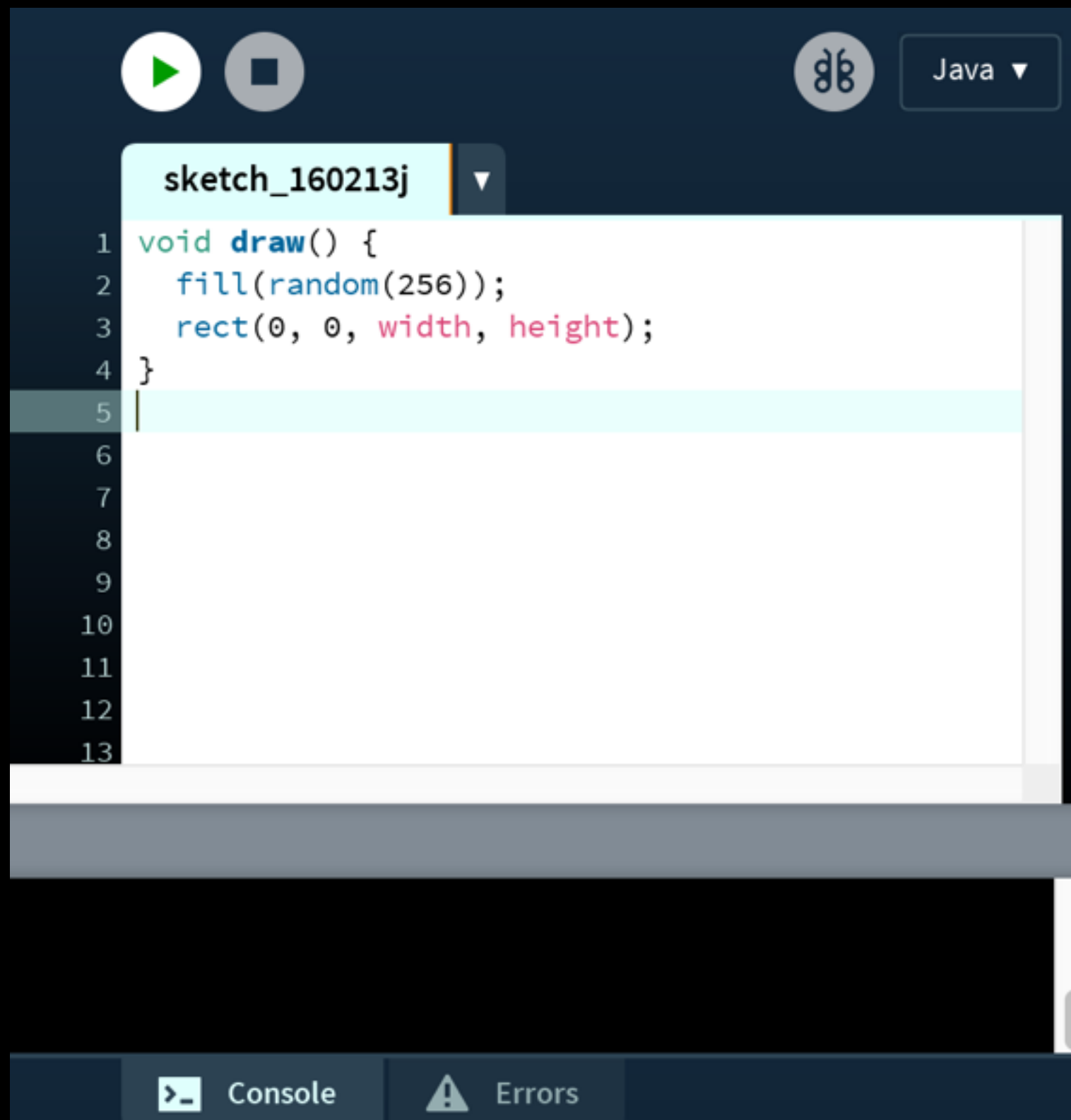
At the bottom of the IDE, there are two tabs: 'Console' (with a terminal icon) and 'Errors' (with a warning triangle icon).

# Exercise: Reactive Mondrian

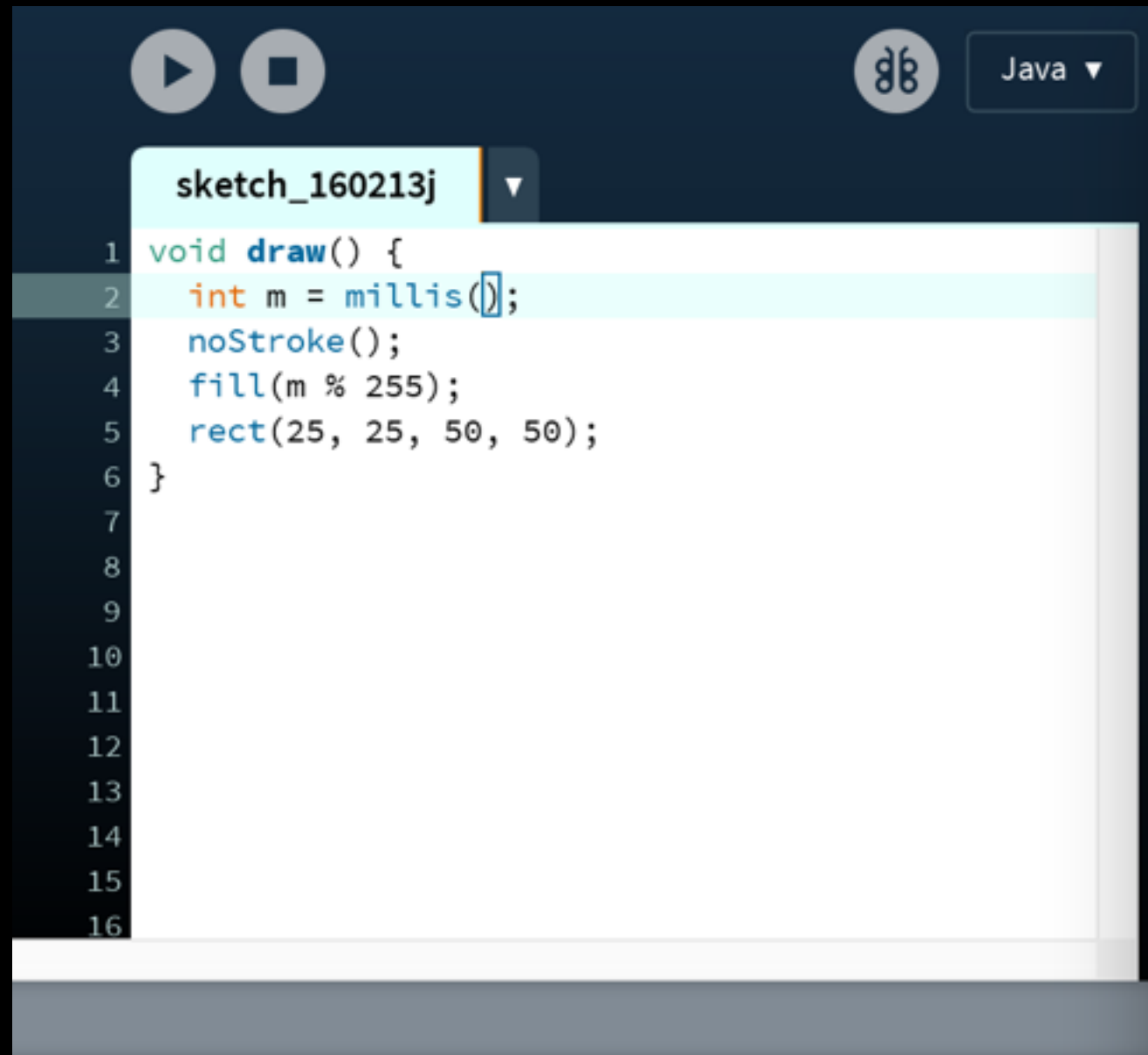




# random()

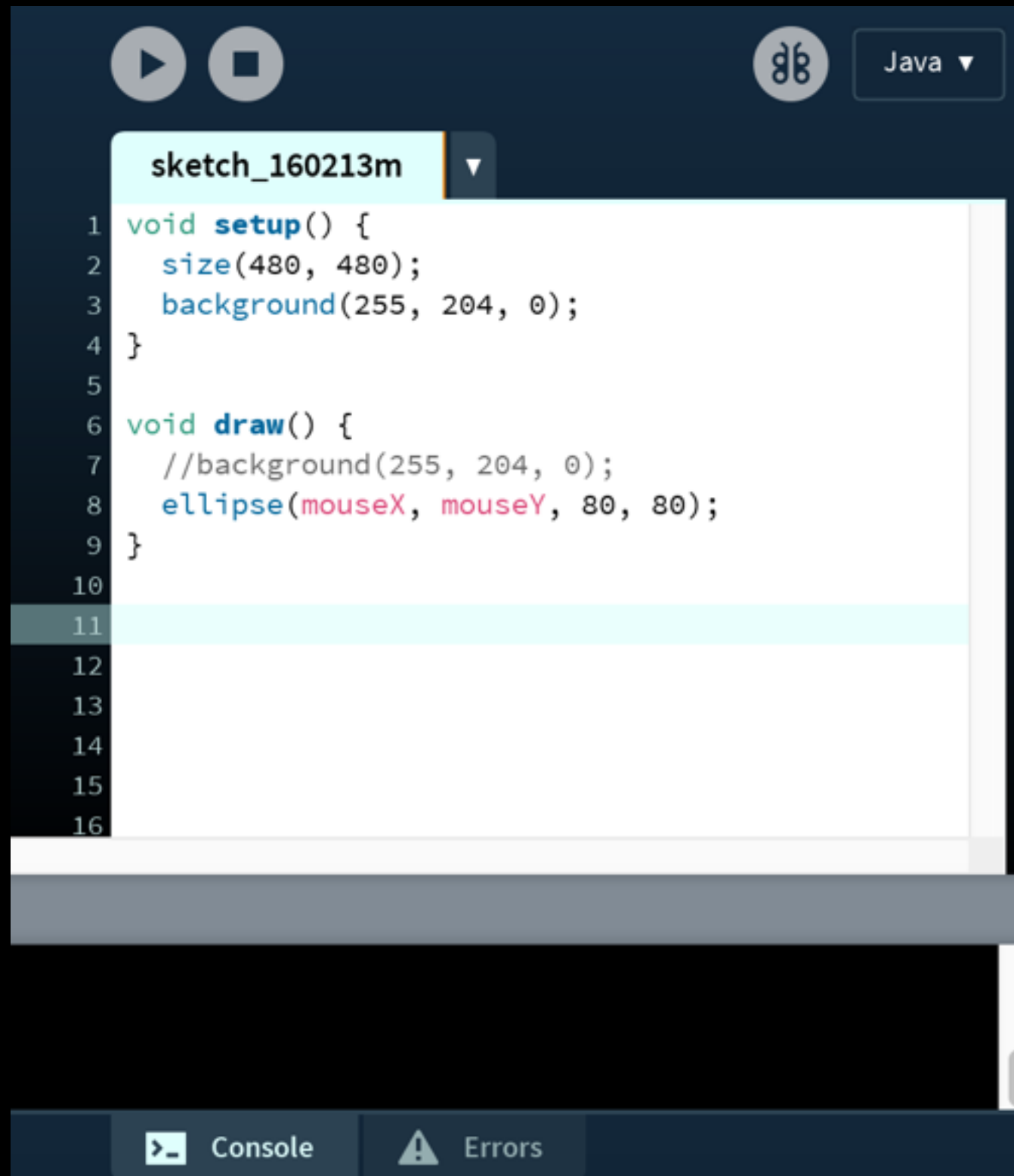


# millis()



```
sketch_160213j
1 void draw() {
2   int m = millis();
3   noStroke();
4   fill(m % 255);
5   rect(25, 25, 50, 50);
6 }
7
8
9
10
11
12
13
14
15
16
```

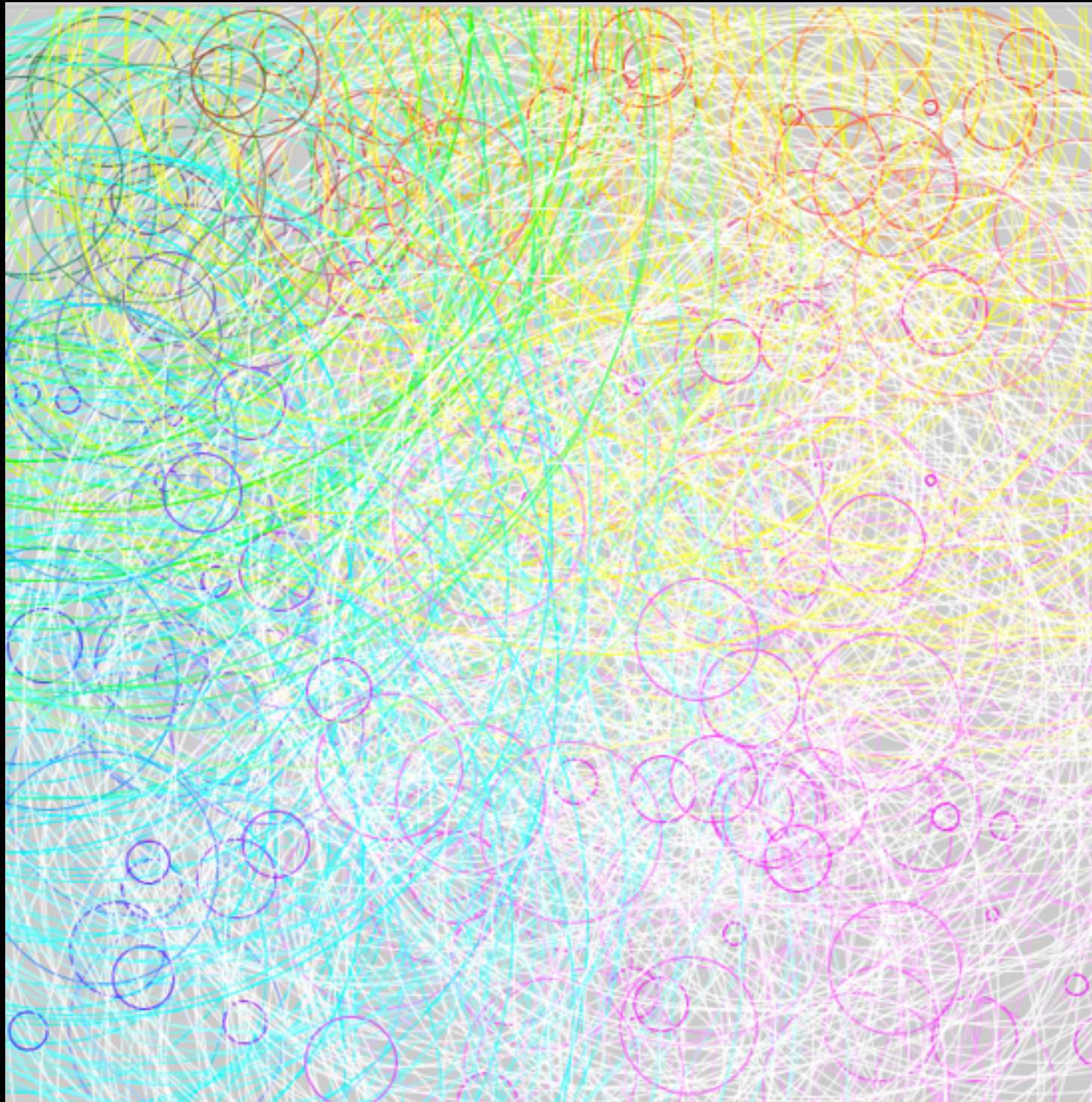
# background()



```
1 void setup() {  
2   size(480, 480);  
3   background(255, 255, 255);  
4 }  
5  
6 void draw() {  
7   //background(255, 255, 255);  
8   ellipse(mouseX, mouseY, 80, 80);  
9 }  
10  
11  
12  
13  
14  
15  
16
```



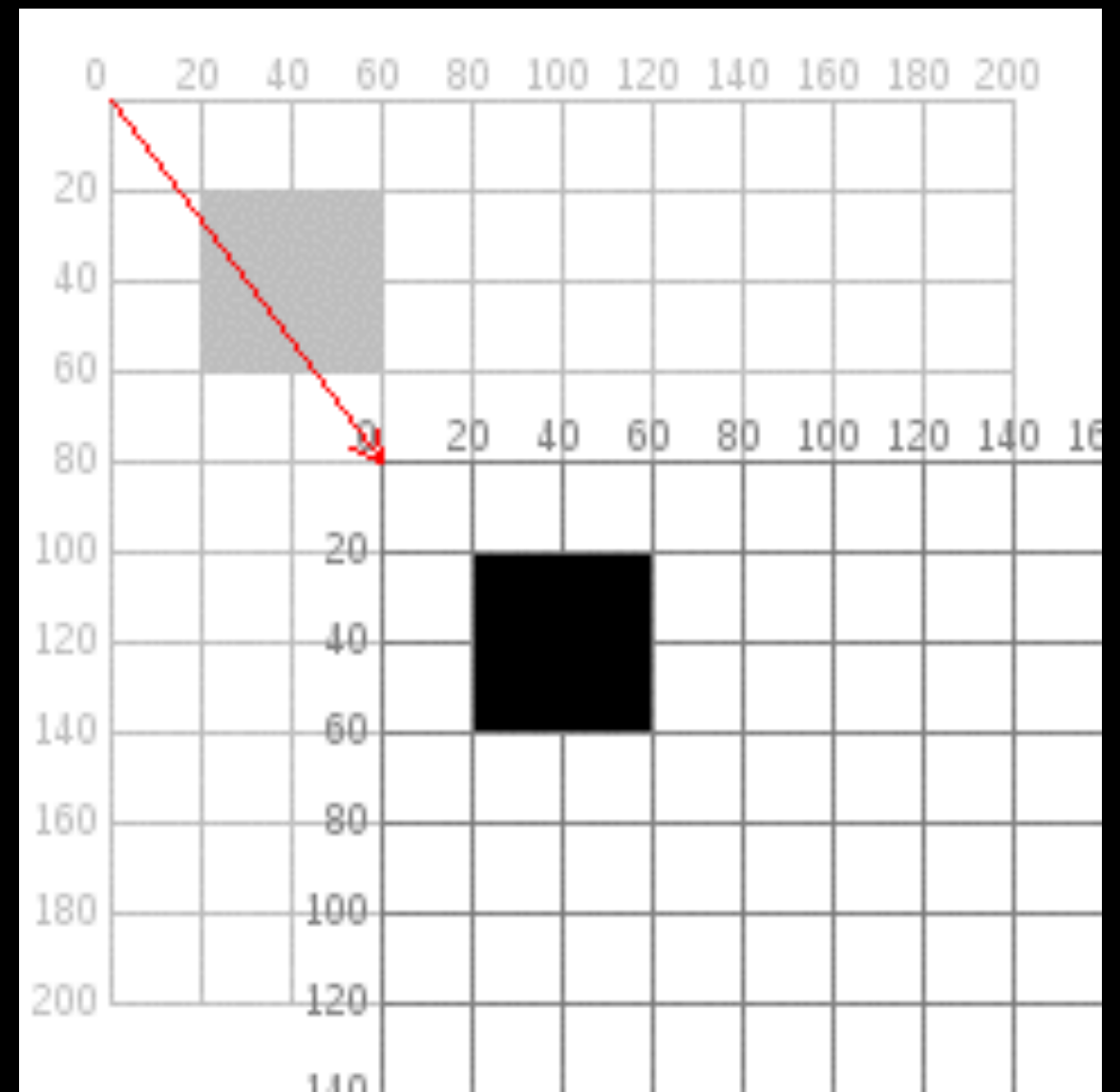
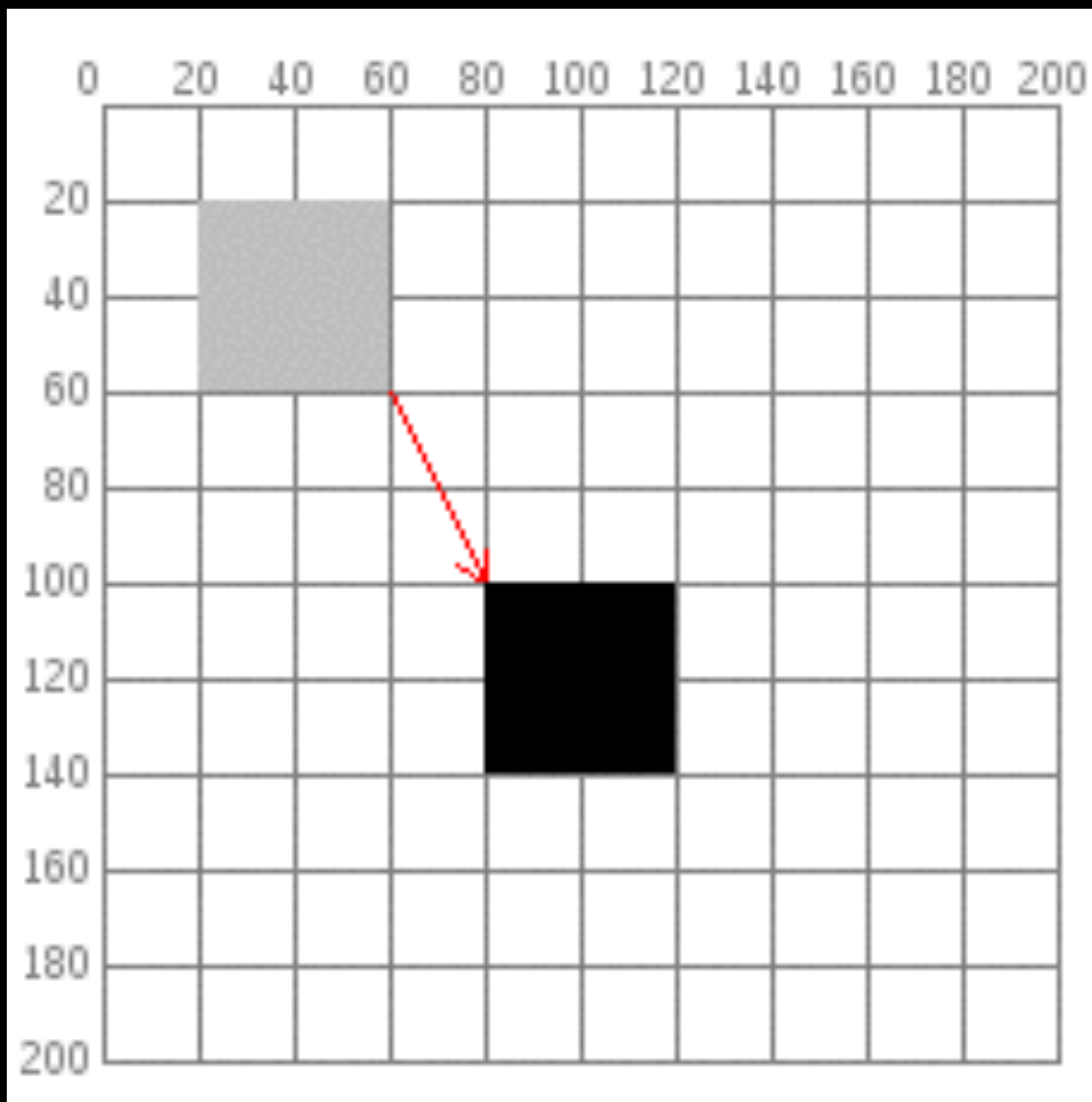
# Exercise: Circles



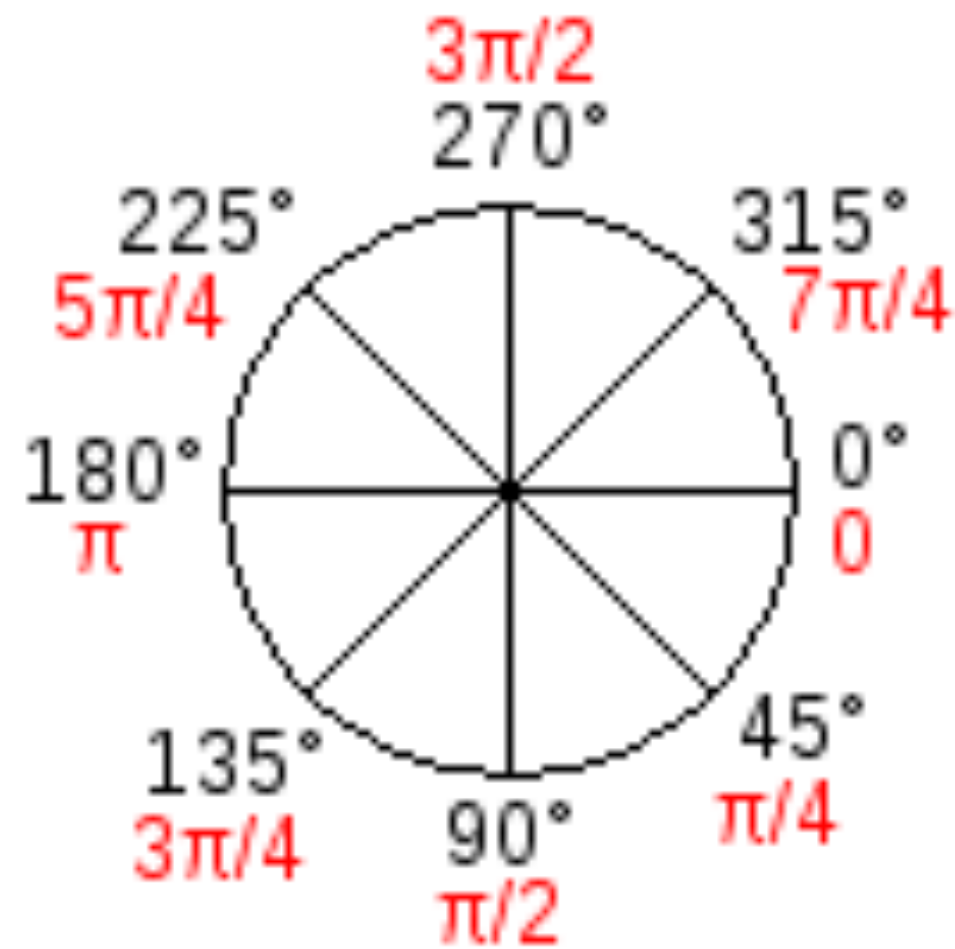


# III. Transformations

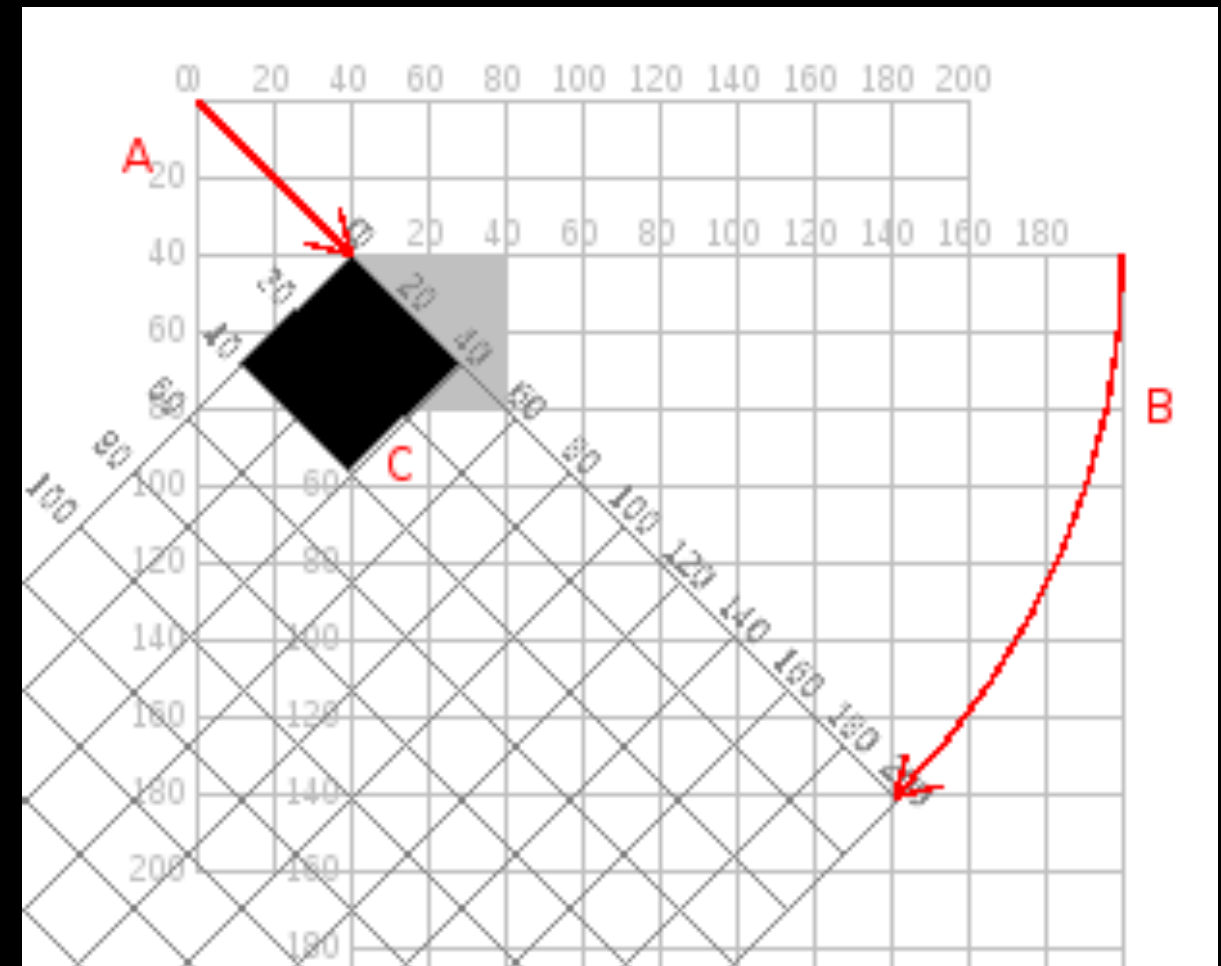
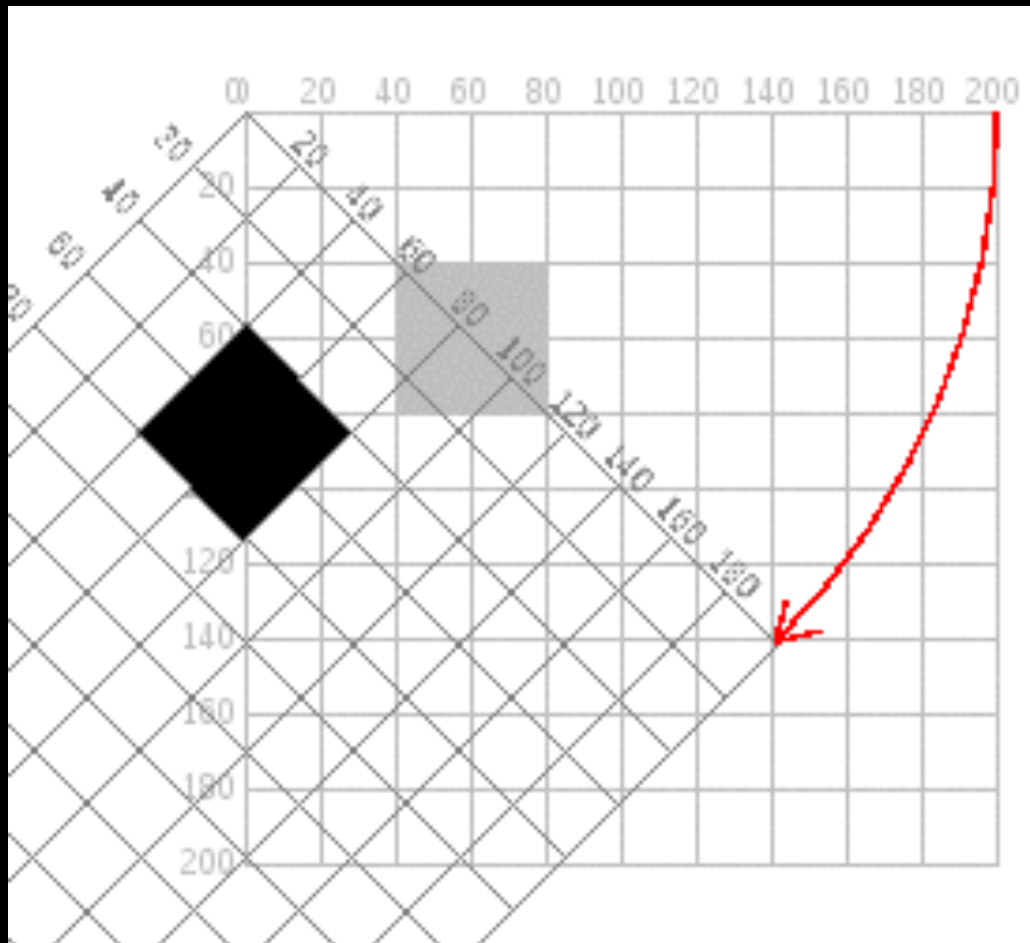
# Translate



# Rotation

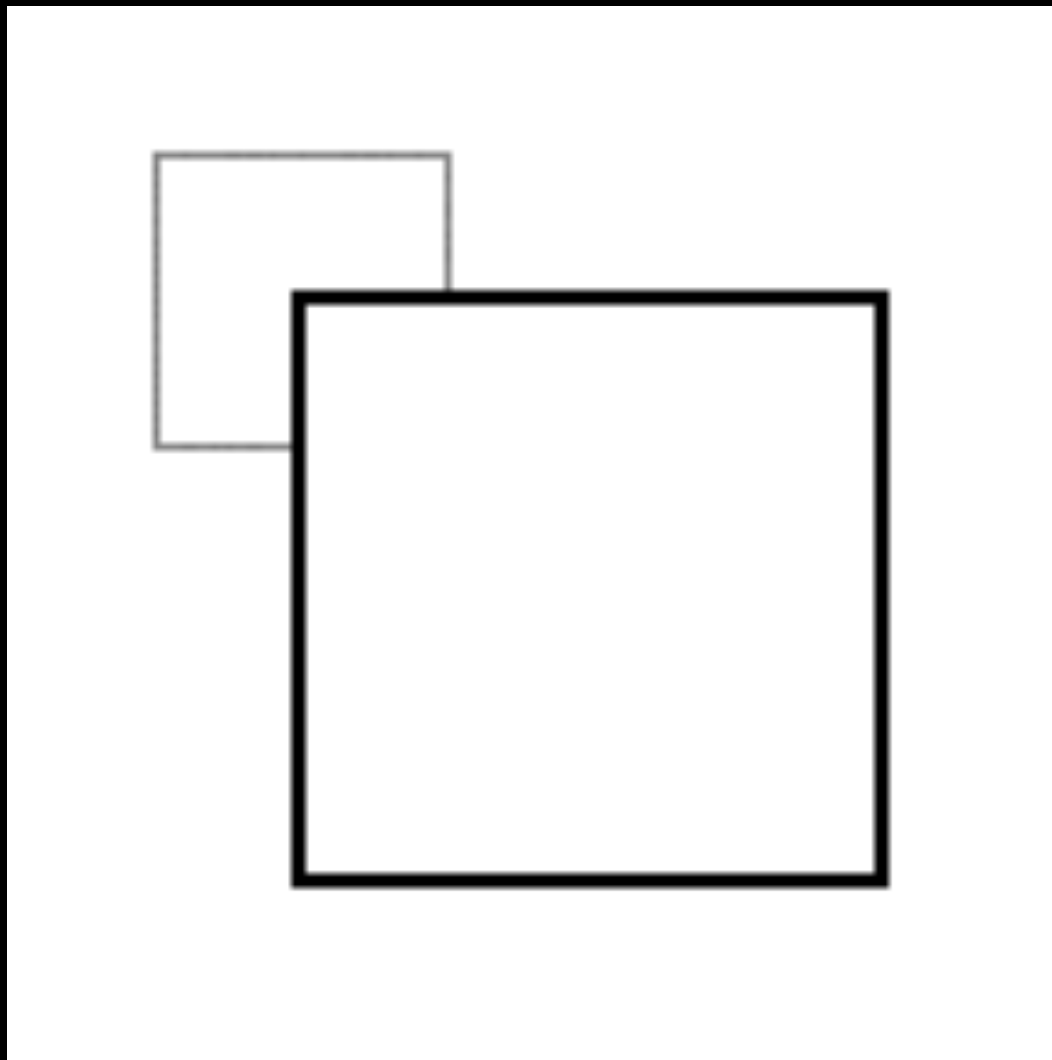


# Rotation



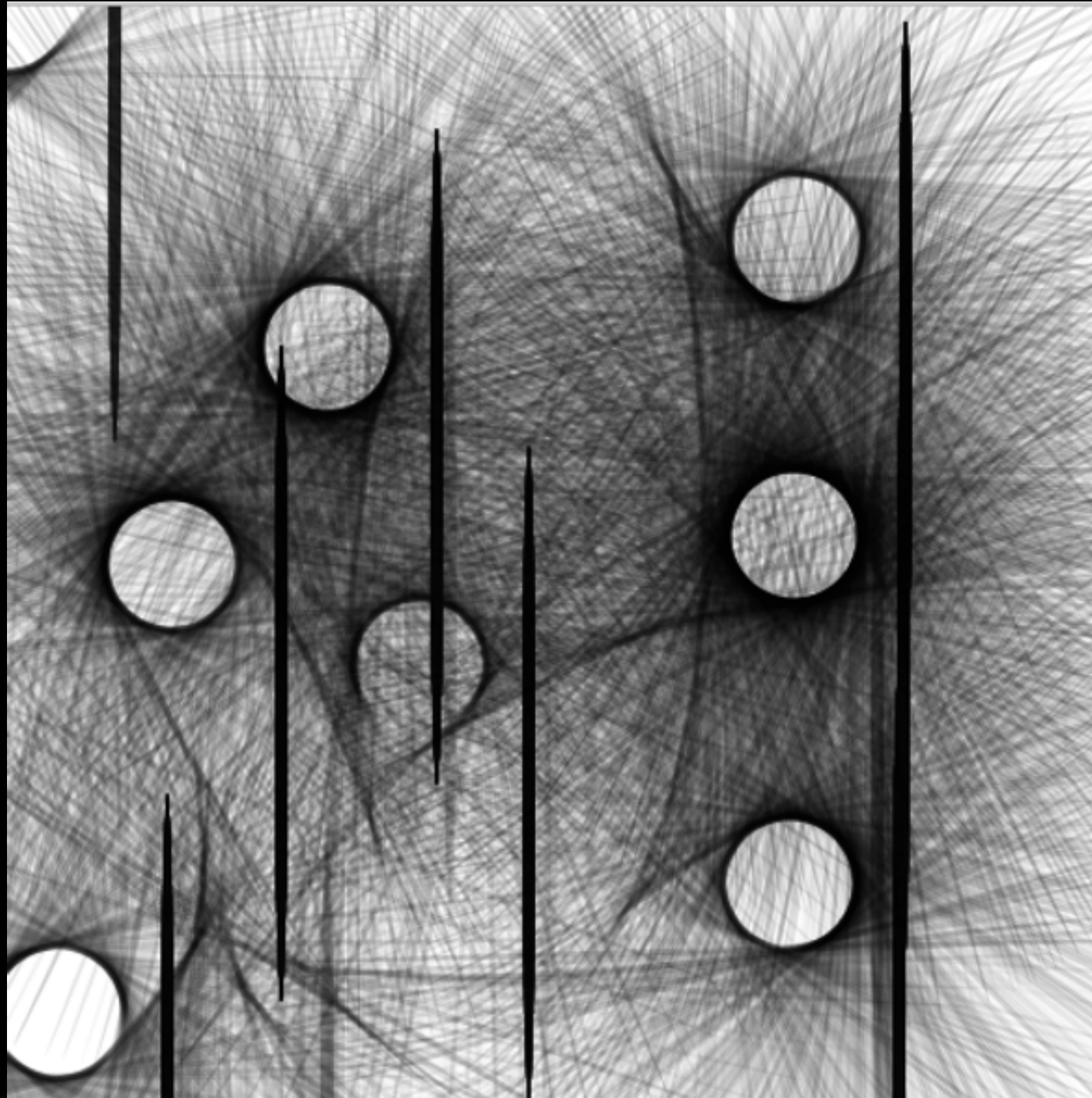


# Scaling



# The Transformation Matrix

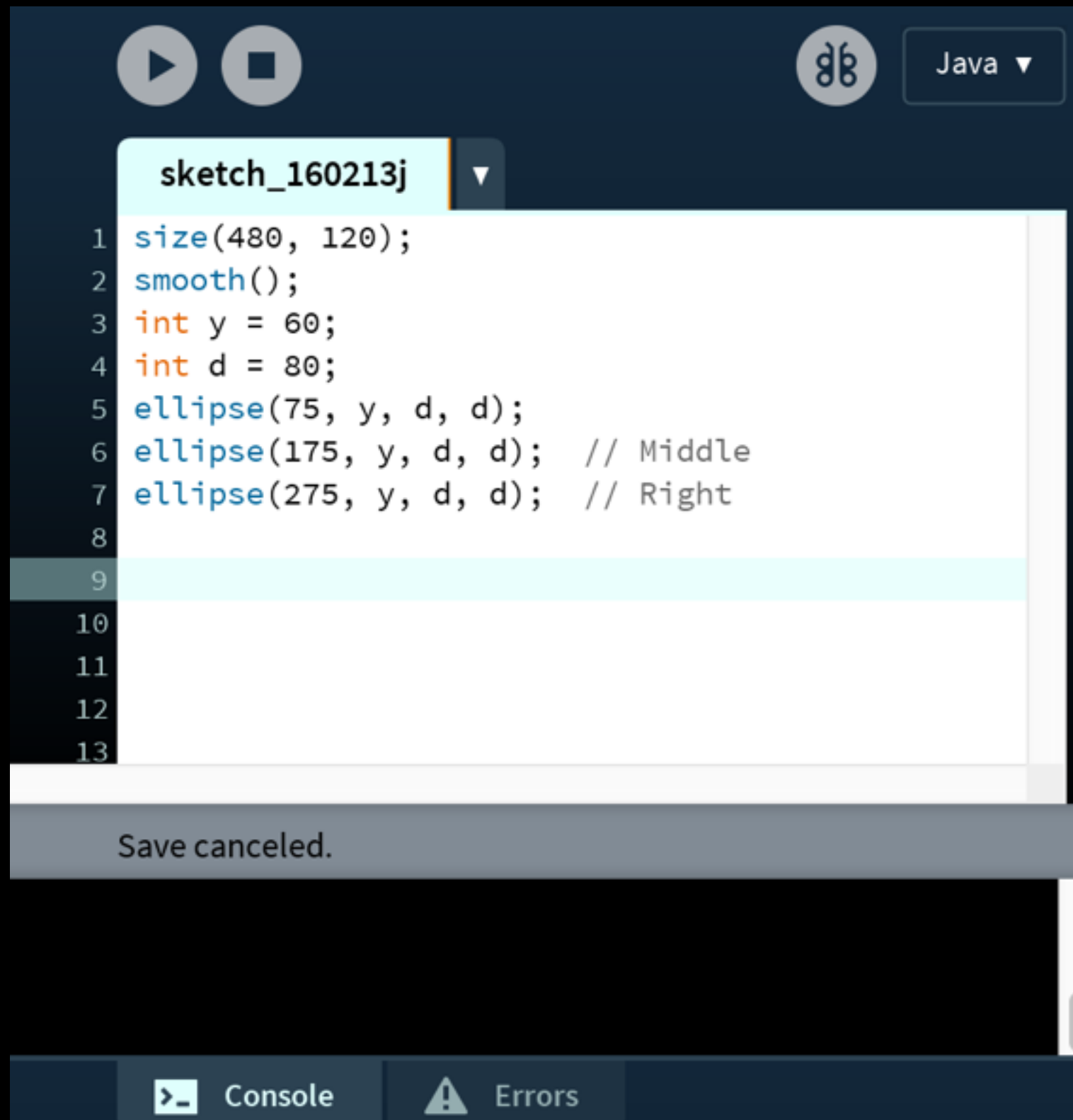
# Exercise: Transformations



# IV. Main Tools



# Variables

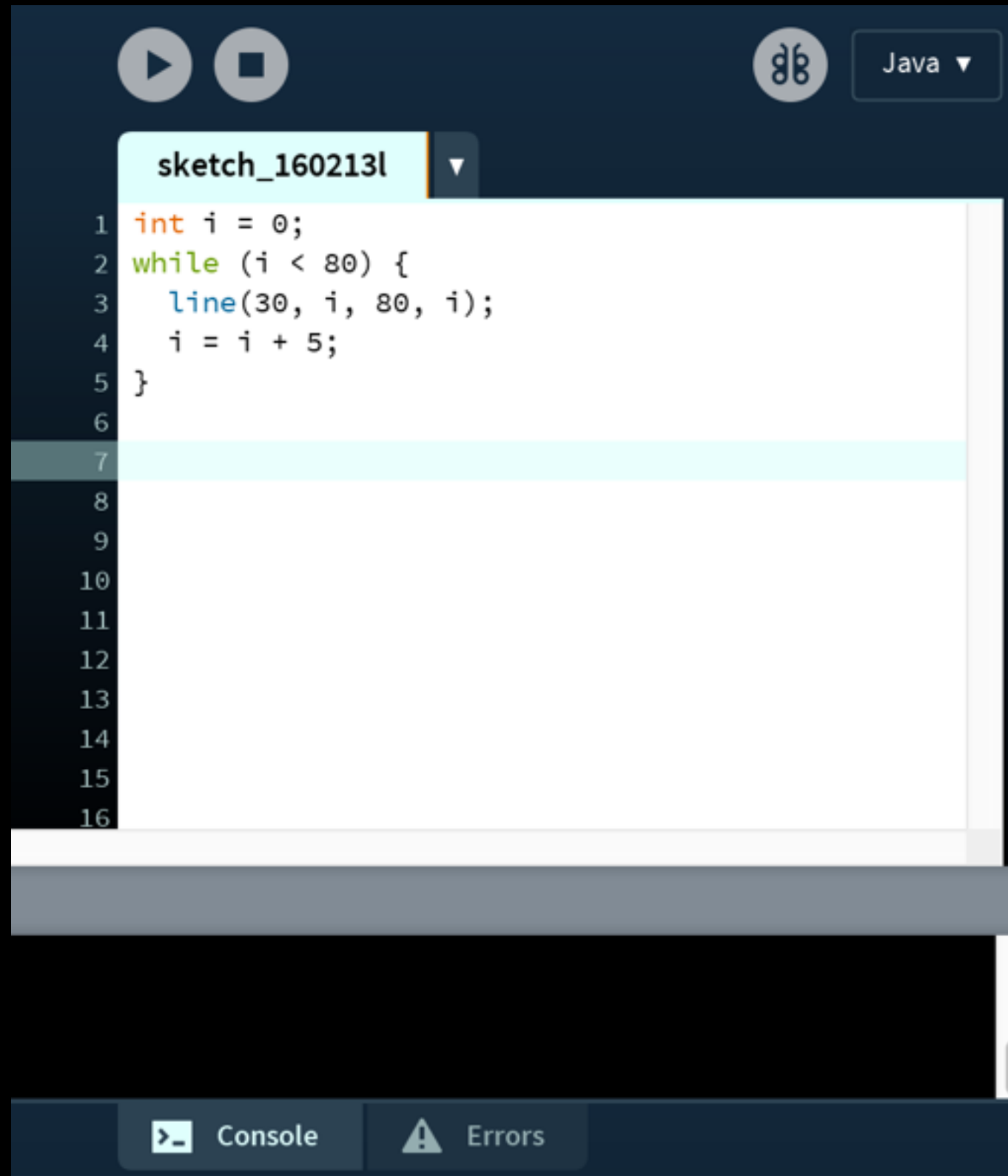


The screenshot shows the Arduino IDE interface. At the top, there are buttons for running (play icon) and stopping (square icon) the sketch, a button with a double infinity-like symbol, and a language dropdown menu set to 'Java'. Below this is a tab labeled 'sketch\_160213j'. The main text area contains the following code:

```
1 size(480, 120);  
2 smooth();  
3 int y = 60;  
4 int d = 80;  
5 ellipse(75, y, d, d);  
6 ellipse(175, y, d, d); // Middle  
7 ellipse(275, y, d, d); // Right  
8  
9  
10  
11  
12  
13
```

Below the code editor, a gray message bar displays the text 'Save canceled.'. At the bottom of the IDE, there are two tabs: 'Console' and 'Errors'.

# while

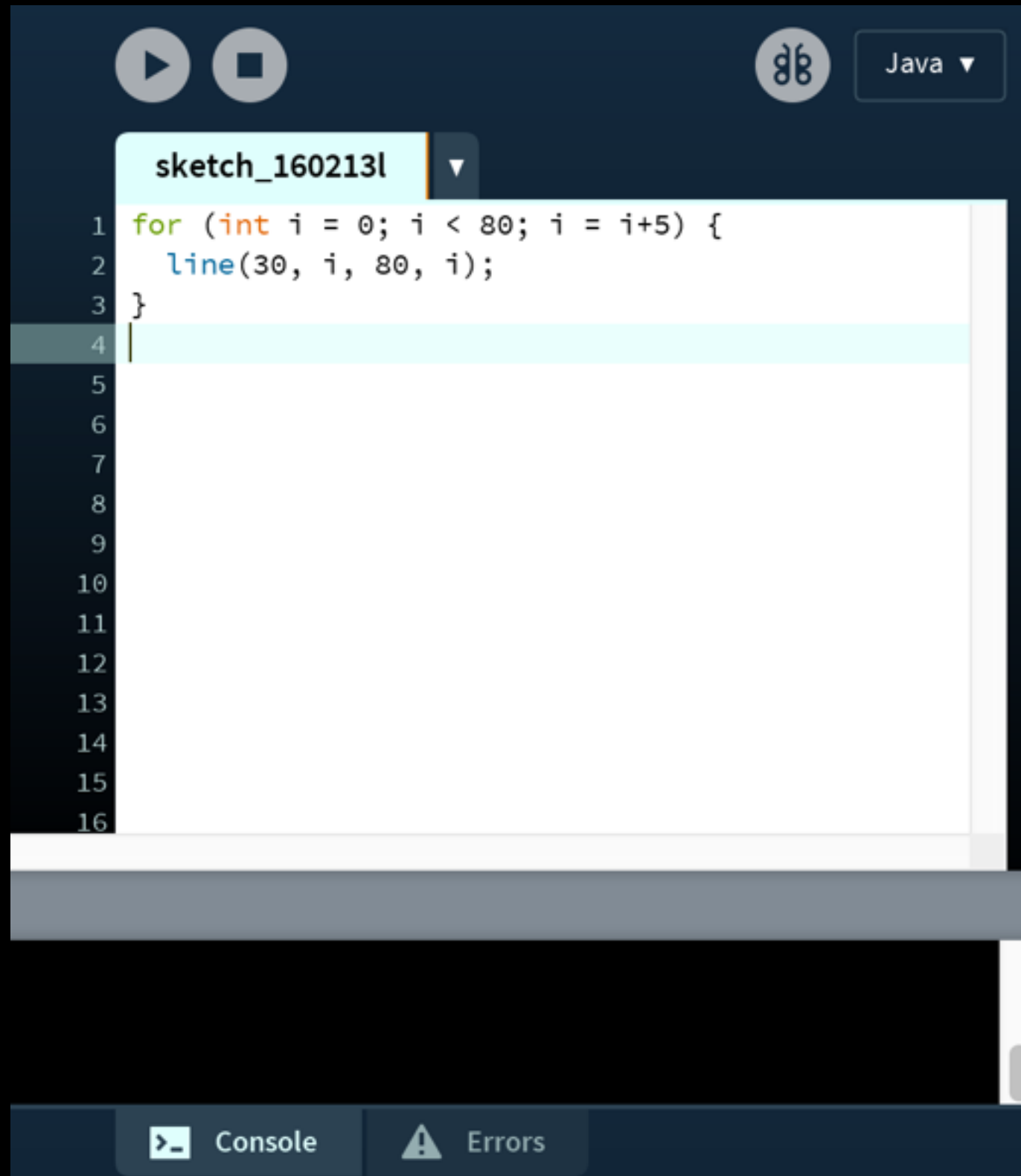


The screenshot shows an IDE window with a tab labeled "sketch\_160213l". The code is written in Java and uses a `while` loop. The code is as follows:

```
1 int i = 0;
2 while (i < 80) {
3     line(30, i, 80, i);
4     i = i + 5;
5 }
6
7
8
9
10
11
12
13
14
15
16
```

The IDE interface includes a top bar with a play button, a stop button, a palette icon, and a language dropdown set to "Java". The bottom bar contains tabs for "Console" and "Errors".

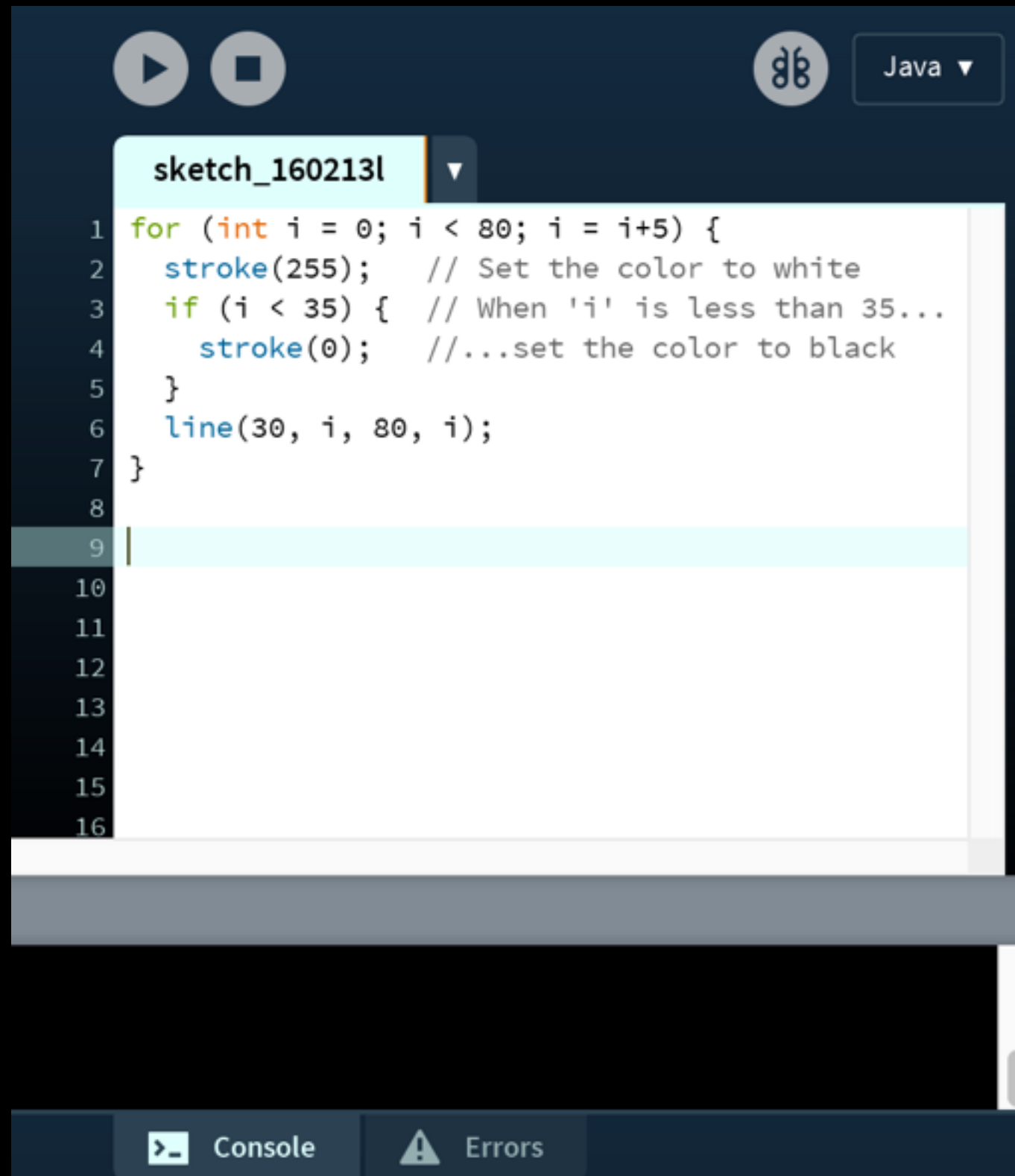
# for



The screenshot shows an IDE window with a dark theme. At the top, there are icons for running (a play button) and stopping (a square button), followed by a circular icon with two vertical bars and a dropdown menu showing 'Java'. Below this is a tab labeled 'sketch\_160213l'. The main editor area contains a Java code snippet with line numbers 1 through 16 on the left. The code is a for loop that draws a horizontal line. The loop starts at line 1, continues to line 2, and ends at line 3. Line 4 is highlighted with a light blue background. The bottom of the IDE has a panel with two tabs: 'Console' and 'Errors'.

```
1 for (int i = 0; i < 80; i = i+5) {  
2   line(30, i, 80, i);  
3 }  
4  
5  
6  
7  
8  
9  
10  
11  
12  
13  
14  
15  
16
```

# if



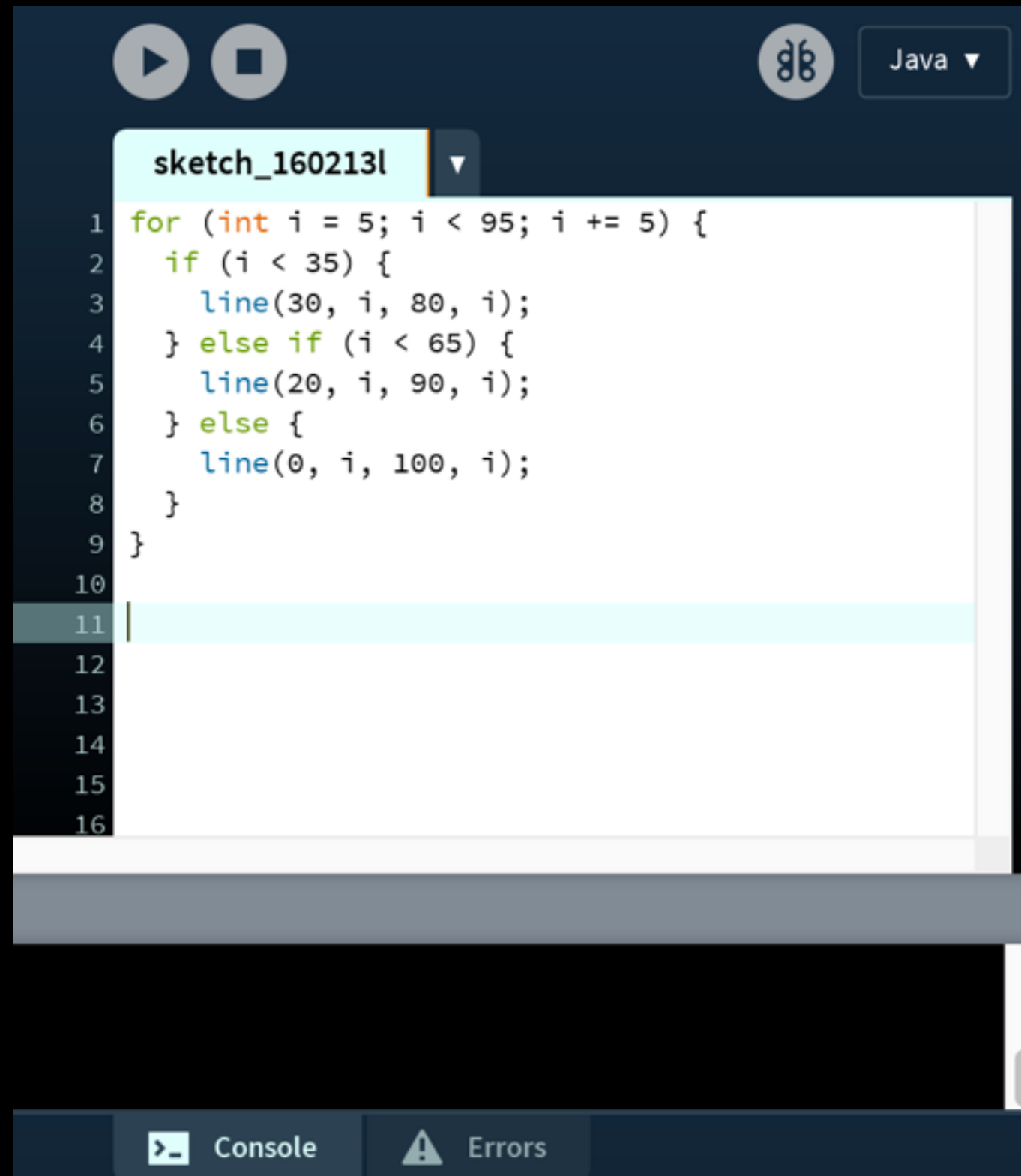
The screenshot shows a Java IDE interface. At the top, there are icons for running (a play button) and stopping (a square button), followed by a circular icon with two vertical bars and a dropdown menu set to 'Java'. Below this is a tab labeled 'sketch\_160213l'. The main area contains a code editor with the following Java code:

```
1 for (int i = 0; i < 80; i = i+5) {  
2     stroke(255);    // Set the color to white  
3     if (i < 35) {    // When 'i' is less than 35...  
4         stroke(0);    //...set the color to black  
5     }  
6     line(30, i, 80, i);  
7 }  
8  
9  
10  
11  
12  
13  
14  
15  
16
```

The code is color-coded: 'for' is orange, 'int' is blue, 'if' is green, 'stroke' is blue, and 'line' is blue. Line numbers 1 through 16 are visible on the left side of the editor. The line containing the 'if' statement (line 3) is highlighted with a light blue background. At the bottom of the IDE, there are two tabs: 'Console' and 'Errors'.



# else

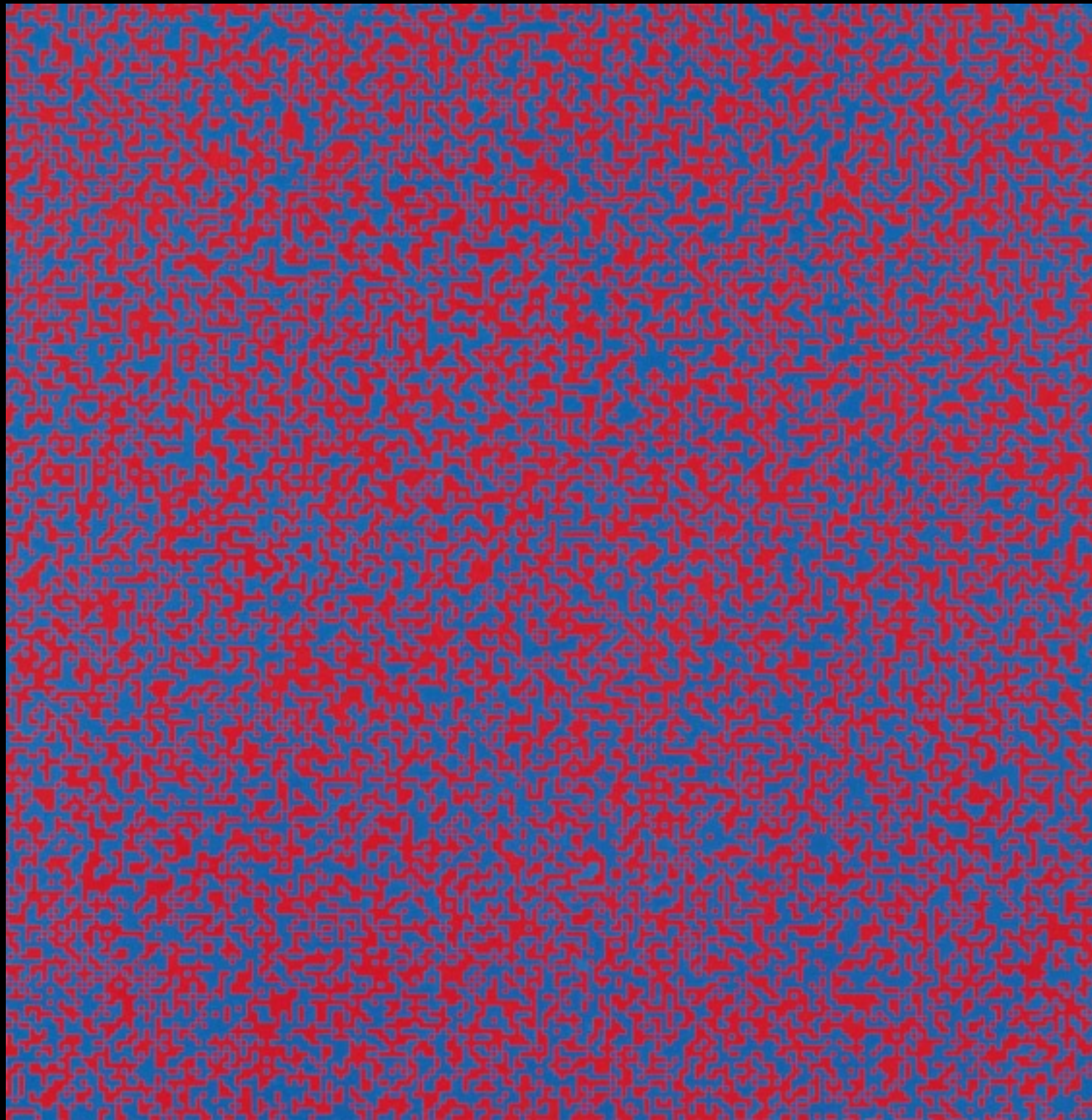


The screenshot shows an IDE window with a tab labeled "sketch\_160213l". The code is written in Java and uses a for loop with an if-else structure. The code is as follows:

```
1 for (int i = 5; i < 95; i += 5) {  
2     if (i < 35) {  
3         line(30, i, 80, i);  
4     } else if (i < 65) {  
5         line(20, i, 90, i);  
6     } else {  
7         line(0, i, 100, i);  
8     }  
9 }  
10  
11  
12  
13  
14  
15  
16
```

The IDE interface includes a top bar with a play button, a stop button, a search icon, and a language dropdown set to "Java". The bottom bar has tabs for "Console" and "Errors".

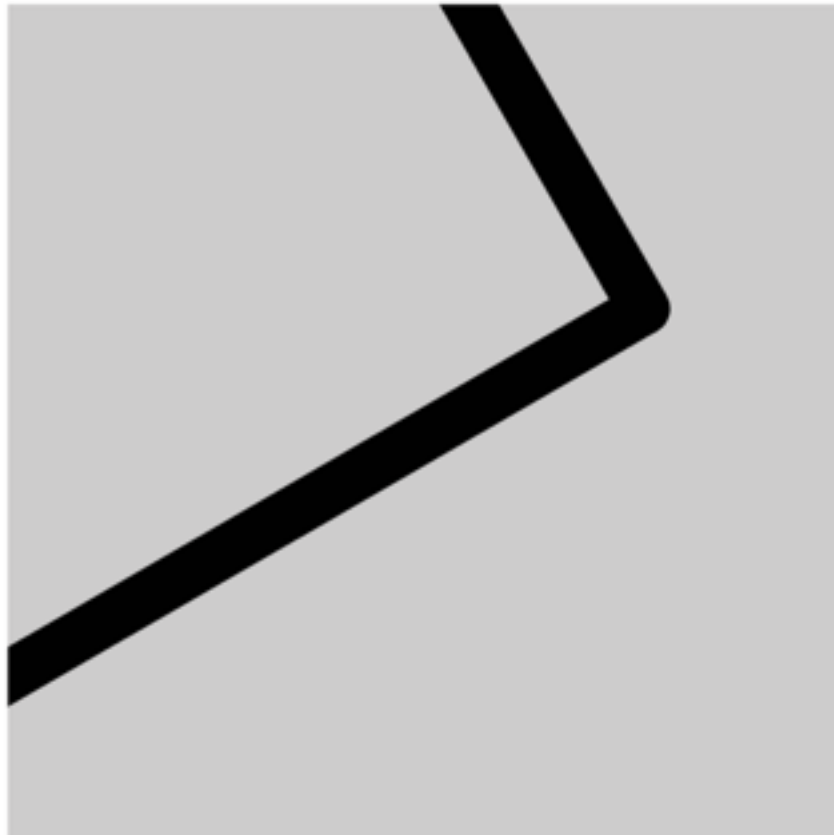
# Exercise: Morellet



V. Print

# Vector vs Raster

Vector image enlarged 800%



Raster image enlarged 800%



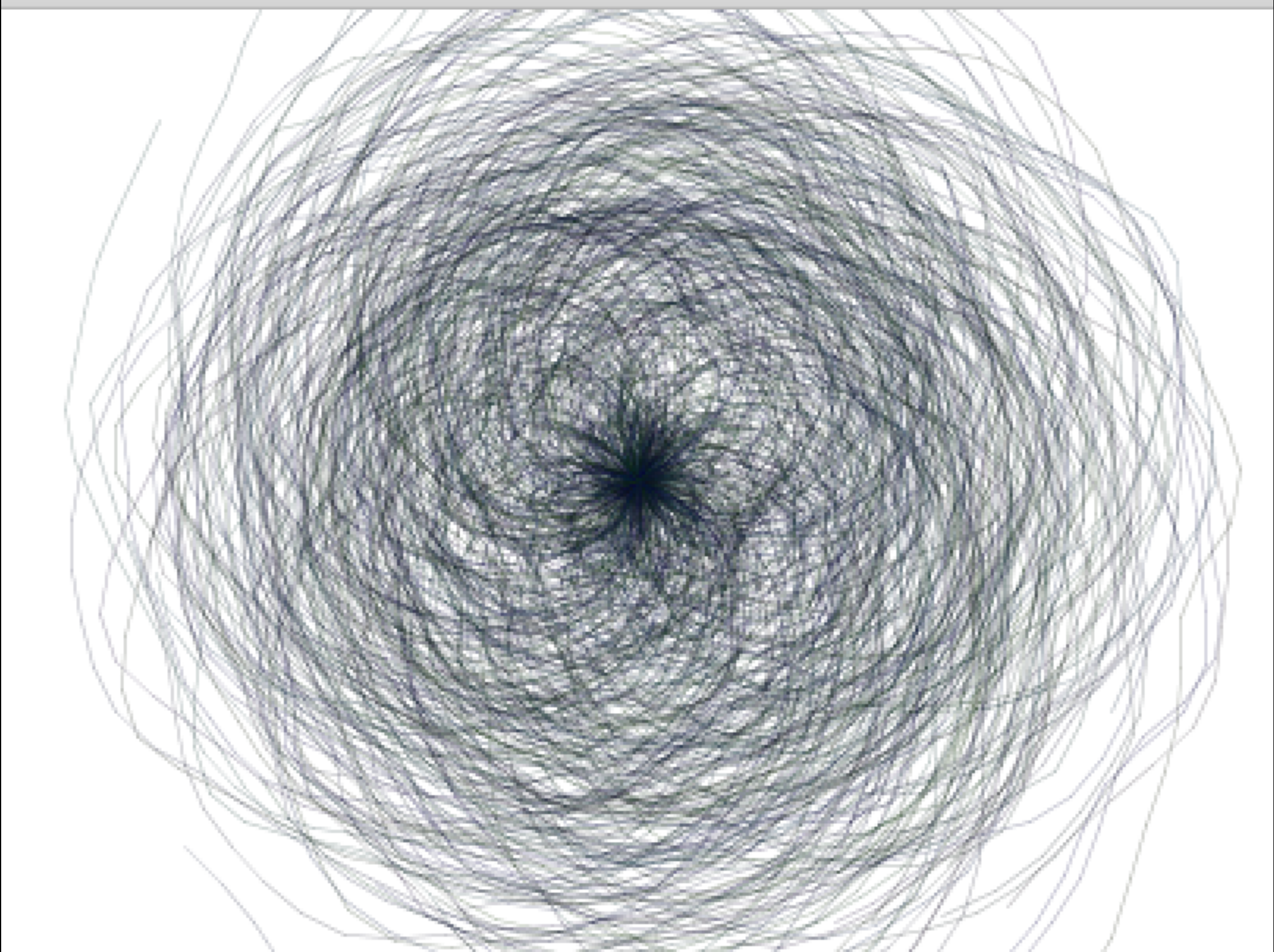
# VI. Noise





# Noise Function

# Exercise: Noise



# VII. Images

# Getting Started



# Image Processing

Pixels

How the pixels look:

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24

How the pixels are stored:

0	1	2	3	4	5	6	7	8	9	.	.	.		
---	---	---	---	---	---	---	---	---	---	---	---	---	--	--

Exercise: Render to PDF

# VIII. Parametric



Goodbye!

# Questions?

**Imanol Gómez**

[imanolgomez.net](http://imanolgomez.net)

[yo@imanolgomez.net](mailto:yo@imanolgomez.net)