Art With Processing

An Intro to Programming Logic

Sarah Bradburn

What's a program?

- A series of statements that makes a computer do something.

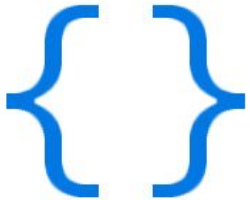
- Today you will learn the basics!

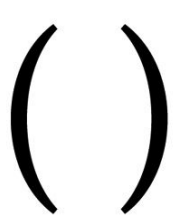
How do you write a program?

- Be ok with not understanding everything
- When you do want to understand something, there are a few basic techniques:
  - Google it!
  - Experiment
  - Ask someone else
- Make mistakes, have fun!

Syntax

- Just like English, programming languages have rules.
  - And your program won't run if you don't follow them!
- Don't stress about syntax today, we care more about the concepts!





Functions

- A function is a statement that performs some action
- You can *call* a function
  - function();
- and you can *implement* a function
  - void function() {

    Write code here

    }

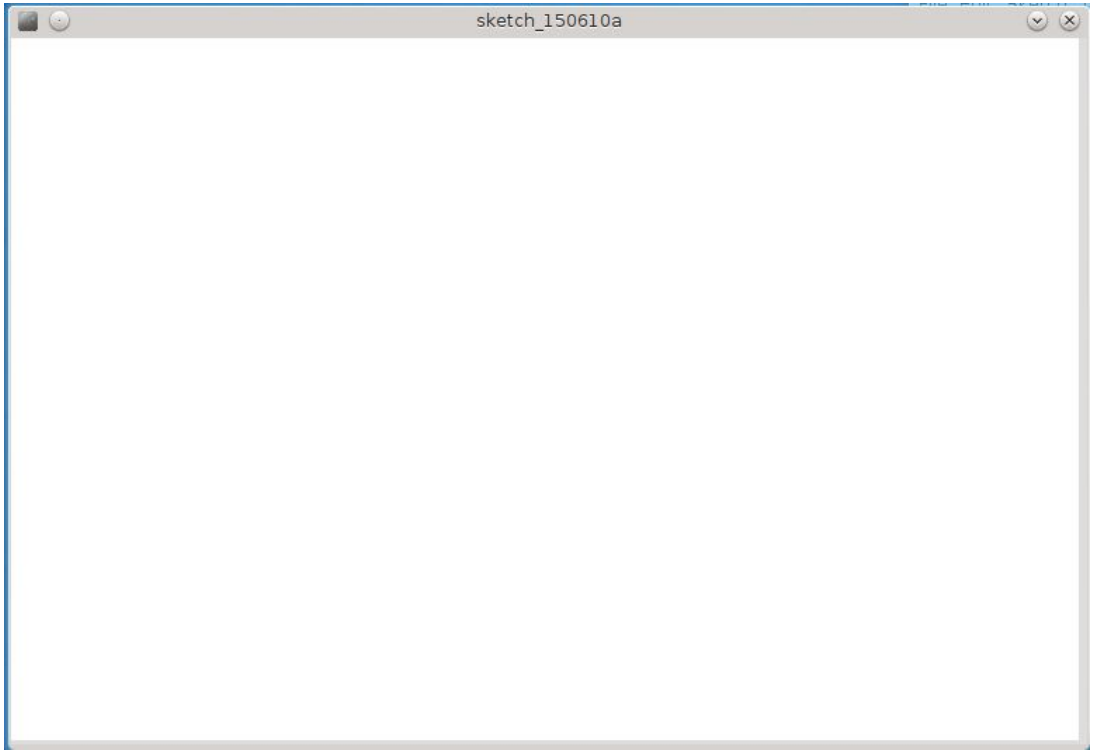
Using the setup() function

First, we will implement the setup() function to create a background.

Note: setup() is a special function in processing

```
void setup() {
  size(760, 500);
  noStroke();
  background(255);
}
```

Using the `setup()` function



Loops

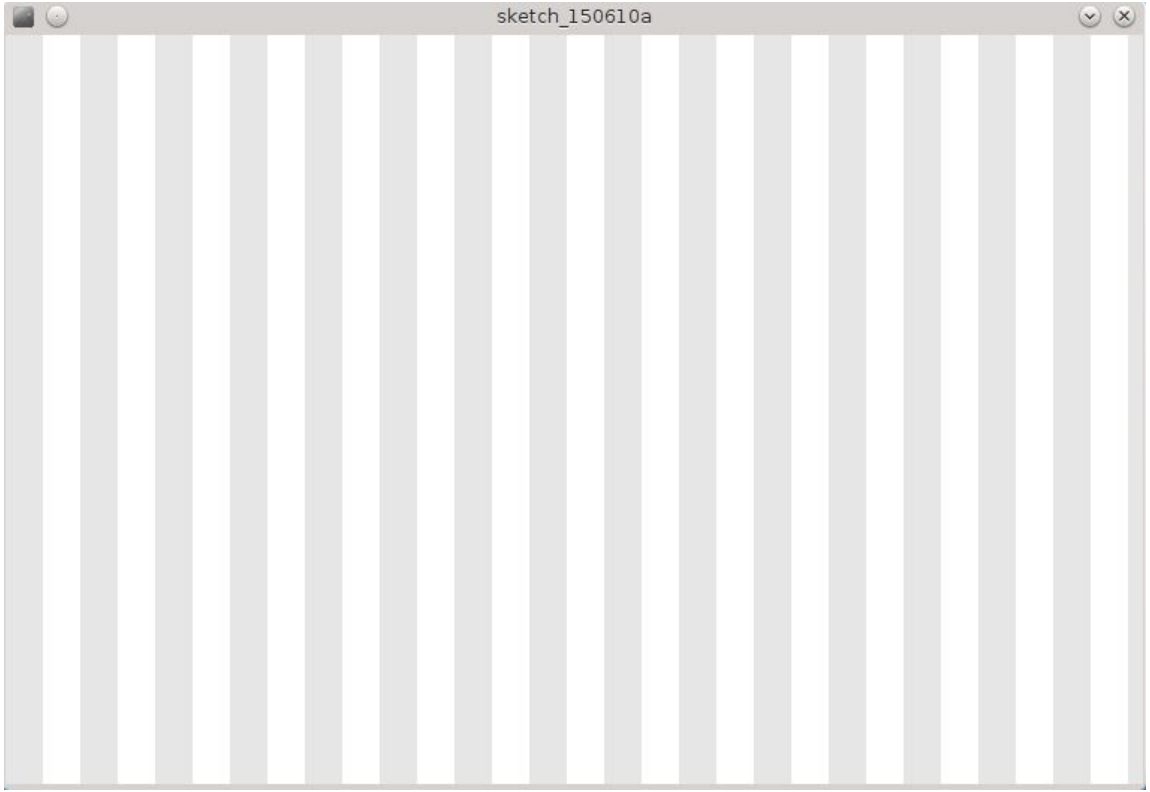
- Computers are great at doing repetitive tasks
- What could we do in a loop?
- Today we will use a *for loop* (this loop has fancy syntax)

Adding Patterns

Right now it just makes a white box. Lets add a pattern! Inside of setup, we will draw rectangles to make bars.

```
fill(230);
for(int i = 0; i < width; i = i + 50) {
  rect(i, 0, 25, height);
}
```

Adding Patterns



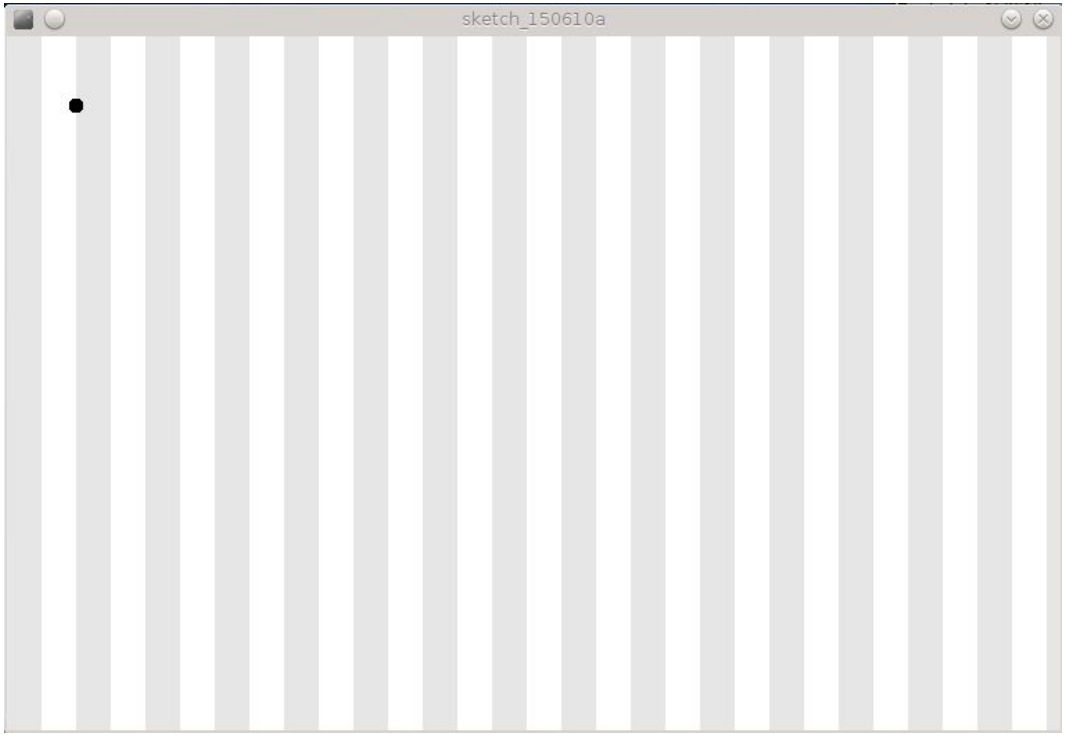
Code Check!

Using the draw() function

Processing asks the program to draw() about 60 times per second! If you want to do something over and over again, you will do it in draw().

```
void draw() {
  fill(0);
  ellipse(50, 50, 10, 10);
}
```

Using the draw() function

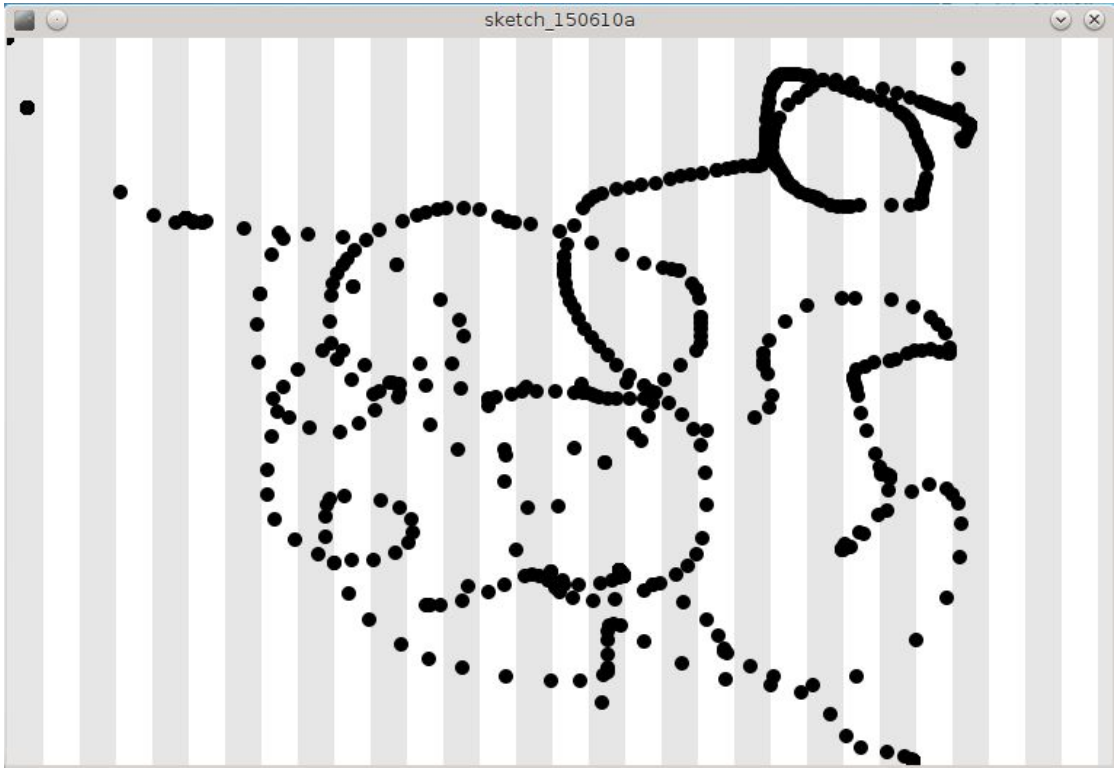


Following the mouse

We can find the x and y coordinates of our mouse and draw an ellipse at that position.

```
void draw() {
  fill(0);
  ellipse(mouseX, mouseY, 10, 10);
}
```

Following the mouse



Variables

Variables are used to make a value more dynamic.

At the very beginning of your program (before the setup function) add a variable called size.

```
int size = 10;
```

Back to draw()

Now we can use our `size` variable to define our ellipse!

```
void draw() {
  fill(0);
  ellipse(mouseX, mouseY, size, size);
}
```

Back to draw()

Change the value of `size` to `20` and see what happens.

```
int size = 20;
```

Adding a fill value

We will use another variable to define the color of the ellipses.

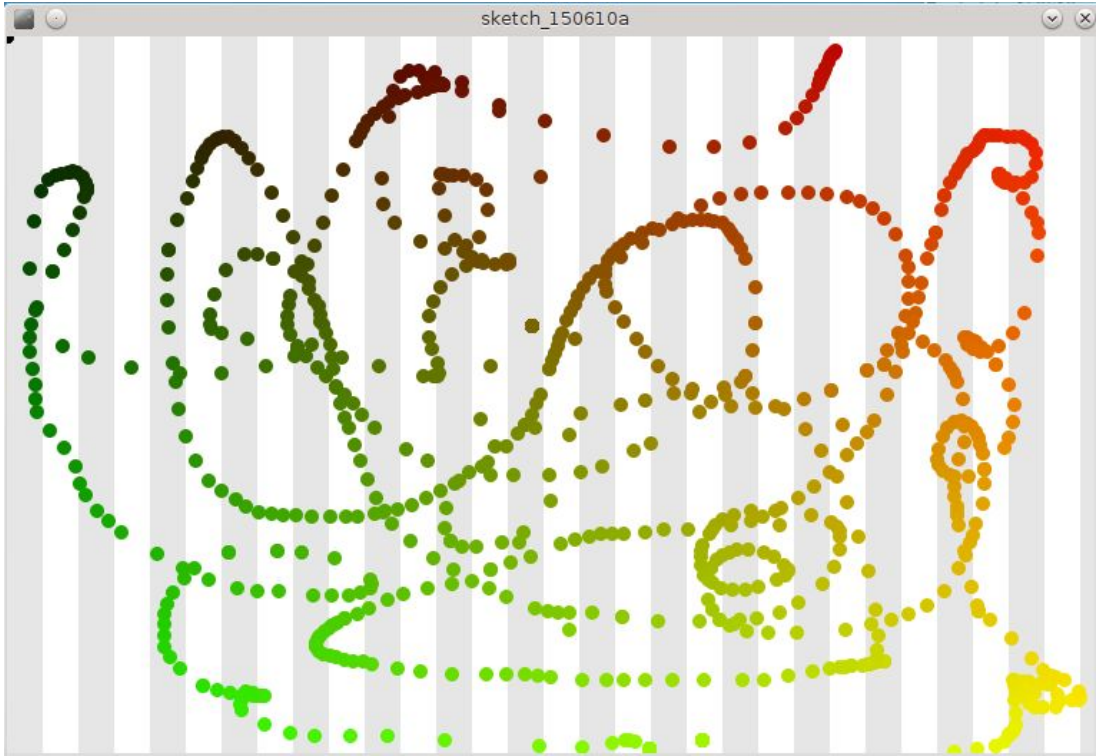
Underneath size, define a new variable called blueValue

```
int blueValue = 0;
```

Inside of `draw()` (before you draw the ellipse), change `fill(0)` to

```
fill(mouseX/3, mouseY/2, blueValue);
```

Adding a fill value



Code Check!

Using `mousePressed()` to manipulate your variables

The `mousePressed()` function will be called every time the mouse is clicked. We will use this change your variables.

```
void mousePressed() {
  size = size + 10;
}
```

Conditionals

- A conditional allows us to make a decision based on a *condition*
- If *it is snowing* then I will have a snowball fight

Using an if statement

What if we want a right click and a left click to do different things?

```
if(mouseButton == LEFT) {
  size = size + 10;
}
if(mouseButton == RIGHT) {
  blueValue = blueValue + 20;
}
```

Code Check!

```
int size = 10;
int blueValue = 0;

void setup() {
  size(760, 500);
  noStroke();
  background(255);

  fill(230);

  for (int i = 0; i < width; i = i + 50)
    rect(i, 0, 25, height);
}
```

```
void draw() {
  fill(mouseX/3, mouseY/2, blueValue);
  ellipse(mouseX, mouseY, size, size);
}

void mousePressed() {
  if(mouseButton == LEFT) {
    size = size + 10;
  }

  if(mouseButton == RIGHT) {
    blueValue = blueValue + 20;
  }
}
```

How do you write a program?

- Be ok with not understanding everything
- When you do want to understand something, there are a few basic techniques:
  - Google it!
  - Experiment
  - Ask someone else
- Make mistakes, have fun!

What Next?

Khan Academy

Add more to your program!

-Try different shapes

-Experiment with the fill colors

-Make the shapes get smaller

-Make the background different

-Use the keyboard

-Experiment with the for loop

-Experimentation is the best part of programming

- Learn what works, learn what doesn't!

-The possibilities are endless! Use the processing API and talk to your neighbor or a tutor!

-Play with the built in processing examples!

You can use the Processing API to find more functions!

https://processing.org/reference/