

# Programming Patterns

## Creative Coding

**Goal:** Familiarize students with p5.js basics. Introduce them to variables, conditional statements, and color theory.

**Motivation:** Show students “aesthetic” color palettes/images and see if they can recognize what different abstract painting are representing.

**CT Themes:** Abstraction, Problem Decomposition, Algorithmic Design

### Description

You will experiment with shapes and color in p5.js, be challenged to create different patterns and learn some basic concepts from color theory.

### Part 1 – Experimentation

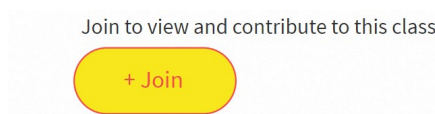
1. Log in to the Computer.

Username: Password:

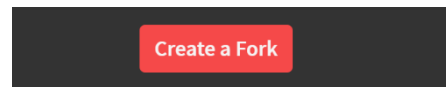
-----

2. Go to <https://www.openprocessing.org/sketch/736493>

Click “Join”  
Create an account.  
Class Code: BCB592



3. Scroll down to “Lesson 1” and click on the sketch “p5.js Basics”.  
On the upper right-hand side click the fork button and select “Create a Fork”



4. Go to the **edit code** tab.



5. Experiment with the code! Change the value of numbers and see what happens. To run the code, press the play button.



## Part 2 – Explanation

**Function:** *a set of instructions which executes a desired task and returns the result to the user.*

You may have noticed that `rect()`, `ellipse()`, `triangle()` and `fill()` all have parenthesis after them and take in a set of numbers (called *parameters*) which tell the computer what to do. These are all functions.

For example, the `ellipse()` function takes in four parameters. The first two parameters control the position of the shape (x and y), and the second two control the size (width and height).

```
6 function draw() {  
7   ellipse(400, 100, 50, 80);  
8 }
```

### ellipse()

#### Example



```
ellipse(56, 46, 55, 55);
```

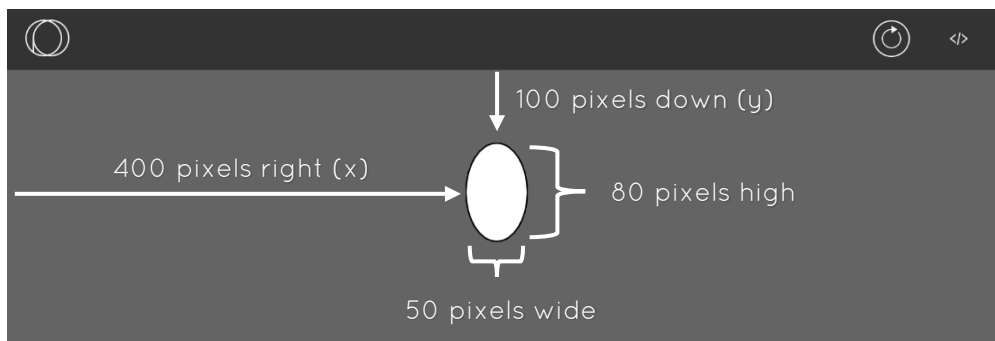
[edit](#) [reset](#) [copy](#)

#### Description

Draws an ellipse (oval) to the screen. An ellipse with equal width and height is a circle. By default, the first two parameters set the location, and the third and fourth parameters set the shape's width and height. If no height is specified, the value of width is used for both the width and height. If a negative height or width is specified, the absolute value is taken. The origin may be changed with the `ellipseMode()` function.

#### Syntax

```
ellipse(x, y, w, [h])
```



The parameters of the `fill()` function control the red, green, and blue (RGB) values for the color of the object. The color defined will be assigned to every shape after the `fill()` function unless `fill` is redefined. The RGB values range from 0-255.

For help with RGB colors, use

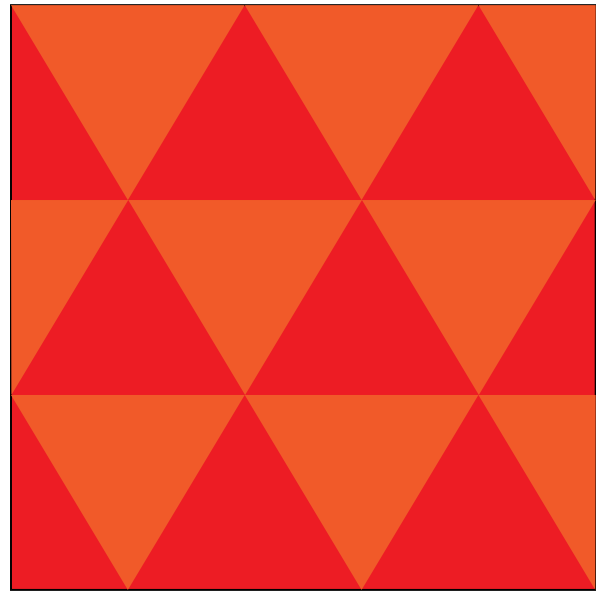
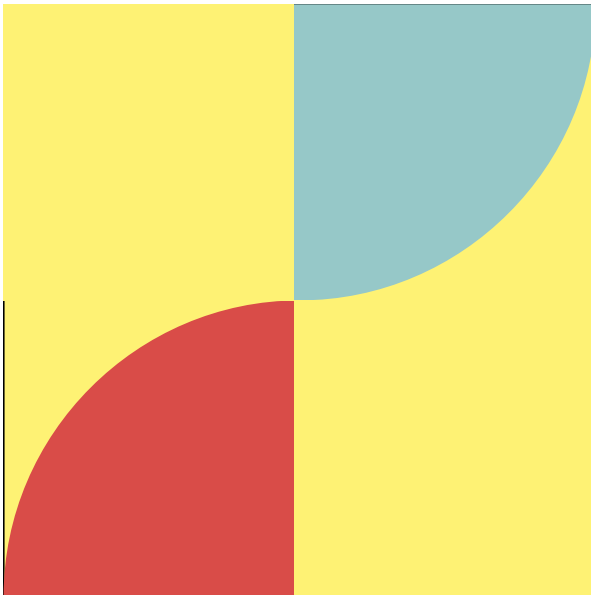
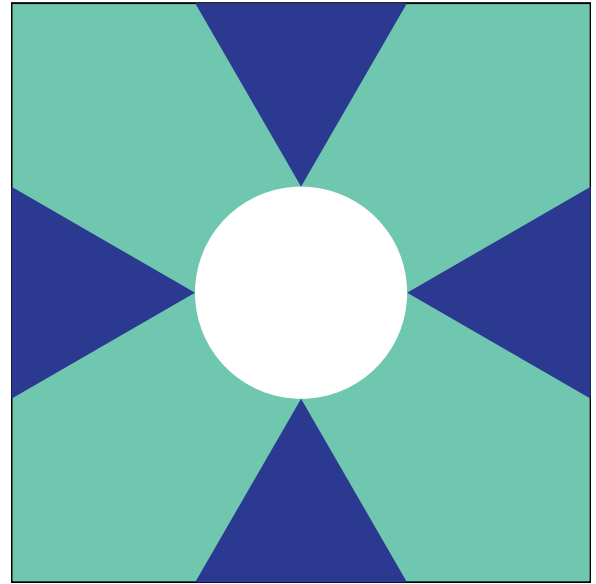
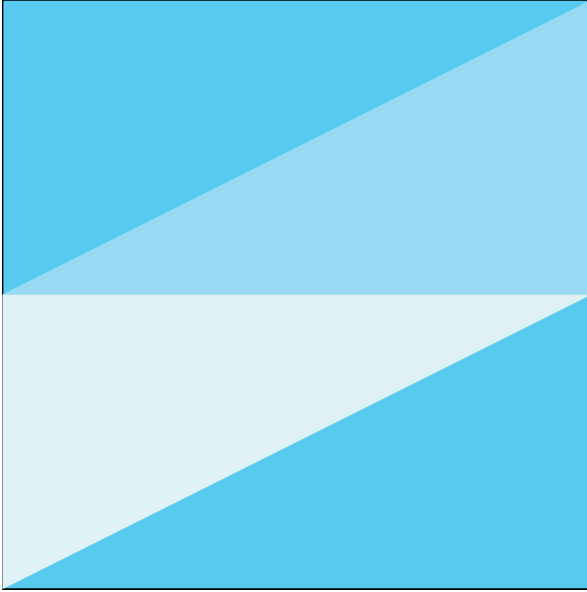
[https://www.rapidtables.com/web/color/RGB\\_Color.html](https://www.rapidtables.com/web/color/RGB_Color.html)

There are many different functions for creating shapes and colors in p5.js. For a more complete list of definitions and explanations of the different p5.js functions, go to:

<https://p5js.org/reference/>

### Part 3 – Challenge

Using what you've learned, recreate one of the following images:



### Part 4 – Concept

You may have noticed that you find certain color combinations to be more pleasing to look at than others. Often, these color palettes result from relationships which exist between colors. Here are a few common relationships used to develop color palettes:

## Monochromatic

Consists of a single hue (such as red, blue, yellow etc.)  
And can vary by:

- Tint: amount of white
- Tone: amount of grey
- Shade: amount of black



Monochromatic Blue



Monochromatic Yellow



## Analogous

Consists of three-four colors next to each other on the color wheel and variations in tint, tone, and shade.



Analogous Red, Orange, Yellow



Analogous Blue, Green, Lime



## Triadic

Consists of three colors equal distance to each other on the color wheel.

Triadic Red, Blue, Yellow



Color relationships are a good way to become familiar with creating color schemes, but don't feel limited by them. There are many different ways to think about color!

## Part 5 – Create

Using everything you've learned, create your own image!