

Introduction to P5.JS

To get started go to: <https://p5js.org/> and click on the link 'Try it out now in the new p5.js Web Editor!'

Your file will look like this when you first open the web editor.

```
function setup() {  
  createCanvas(400, 400);  
}  
  
function draw() {  
  background(220);  
}
```

There are two functions that are included in the set up for us. These have been named `setup` and `draw`.

Functions are blocks of code that are given a name. This means we can reuse the same lines of code rather than writing them out every time we need to use them.

Any lines of code inside the curly braces `{}` of the `setup` function are run once when you press play. At the moment it only contains the line:

```
createCanvas(400,400);
```

So, when you press play a canvas will be created which is 400 pixels wide and 400 pixels tall. Try changing these numbers eg.

```
createCanvas(600,600);
```

Press play again. What happens?

Any lines of code inside the curly braces `{}` of the `draw` function are run over and over many times from when you press play. This will make more sense when you ANIMATE your sketches!

SIZES AND PIXELS

Sizes in p5.js are written in pixels.

1cm is equal to around 40 pixels.

10cm is equal to around 400 pixels.

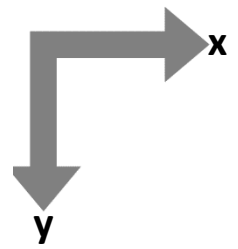
DRAWING SHAPES

Circle:

```
circle(x, y, size);
```

Rectangle:

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



ADDING COLOUR

R	G	B
178	31	53
216	39	53
255	116	53
255	161	53
255	203	53
255	249	53
0	117	58
0	158	71
22	221	53
0	82	165
0	121	231
0	169	252
104	30	126
125	60	181
189	122	246

To fill shapes with colours add the fill line of code on the line before your shape is drawn.

```
fill(225,0,0);  
circle(200,200,50);
```

Writing the lines above will draw a red circle.

```
fill(red, green, blue);
```

This uses **RGB** (Red, Green, Blue) colour. To create the colour you want, mix together different amounts of red, green and blue by entering numbers between 0 and 255 into the fill function.

Pure Colours:

`fill(0,0,0);` = Black

`fill(255,255,255);` = White

`fill(255,0,0);` = Pure Red

`fill(0,255,0);` = Pure Green

`fill(0,0,255);` = Pure Blue

The picture to the left demonstrates the variety of colours you can mix - try some of these and then try to mix some of your own!

If you want an empty/not coloured shape use `noFill()` instead.

Introduction to P5.JS

SHAPE OUTLINES

To change the colour of the outline of the shape use:

```
stroke(0,0,0);
```

This works in much the same way as `fill(,_,_)` as the three inputs are RGB values as before and you use it just before the shape is drawn.

To change the width of the outline you can use:

```
strokeWeight(5);
```

This also goes just before the shape is drawn, and the one input it takes is the width of the outline in pixels. Try numbers between 1 and 50.

INTERACT WITH THE MOUSE

Using `mouseX` and `mouseY` in place of the horizontal and vertical position inputs on our shapes will make the shape follow your mouse across the screen.

For example add a circle with the following line:

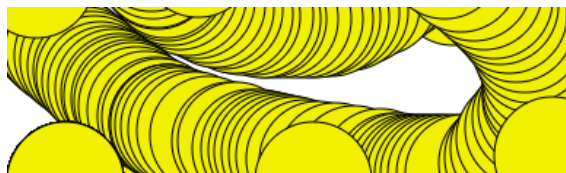
```
circle(mouseX, mouseY, 40);
```

ANIMATE

If you want to see how `mouseX` and `mouseY` work more clearly; try moving the `background()` line from the draw function into the setup function. Something like this:

```
function setup() {  
  createCanvas(400, 400);  
  background(255);  
}  
  
function draw() {  
  fill(240,240,0);  
  circle(mouseX,mouseY,40);  
}
```

Press play and try wiggling your mouse across the canvas. You might get something like this:



A circle is being drawn every time the draw function runs. The background acts like an eraser and it overwrites the circles when it is drawn in. When we move it from draw to setup it only happens once at the beginning and is no longer erasing each circle before the next one is drawn.

SET THE FRAME RATE

If you want to slow down the animation you can add the line

```
frameRate(2);
```

Add this within the setup function. The number in the brackets will be how many times the draw function is run per second.

Try using numbers between 1 and 30.

RANDOM NUMBERS

To add an element of surprise to your coded sketches you can use random numbers. In place of any number you can use `random()`

To get a random number between 0 and 255 use: `random(0,255)`

To get a random number between 100 and 200 use: `random(100,200)`

HOW TO SAVE YOUR SKETCH

Either sign up and create an account using an email. You can then save your sketches by pressing Ctrl + s. Or you can copy and paste your entire file into notepad and save this – copy it back over when you need it!