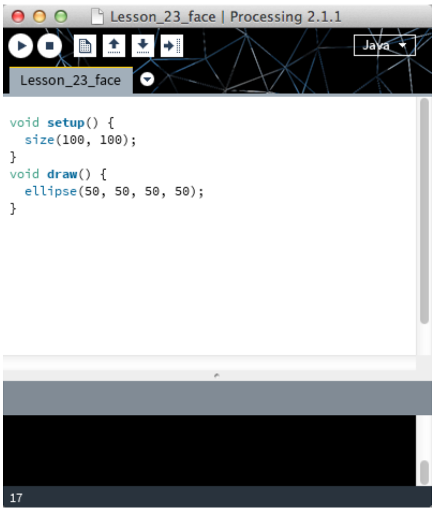
2 - Drawing in Processing

Overview: We are going to learn how to a draw and colour shapes in Processing. How to place shapes on the screen and then creating a character.

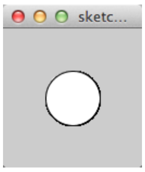
STEP 1: RECAP SESSION 1 - SETUP AND DRAW

Lets take a look at the Processing windows.

The display window opens up when you hit the Run button on the toolbar.



```
void setup() {
  size(100, 100);
}
void draw() {
  ellipse(50, 50, 50, 50);
}
```



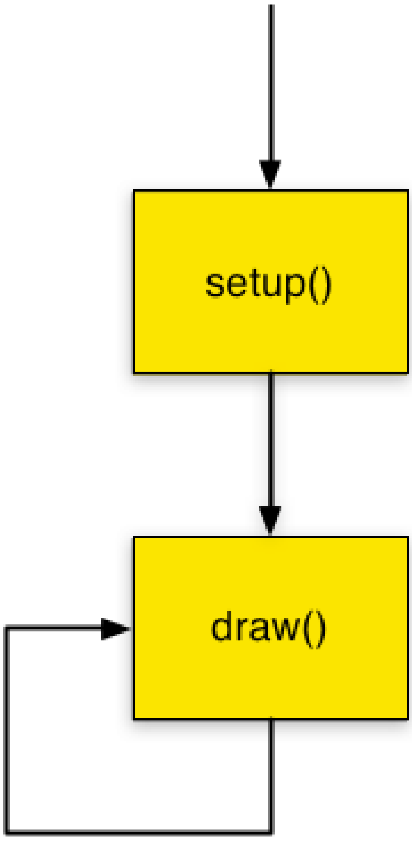
Display Window

setup() and draw() are special functions in Processing.

setup() is called only once per sketch run, at the start. It can be used to define the base sketch setup.

draw() is called on each frame: a sketch is like a movie, it is made of successive frames, images, and their quick succession (by default around 60 per second) makes the eye believe there is a continuous movement.

```
void setup() {

}

void draw() {


}
```

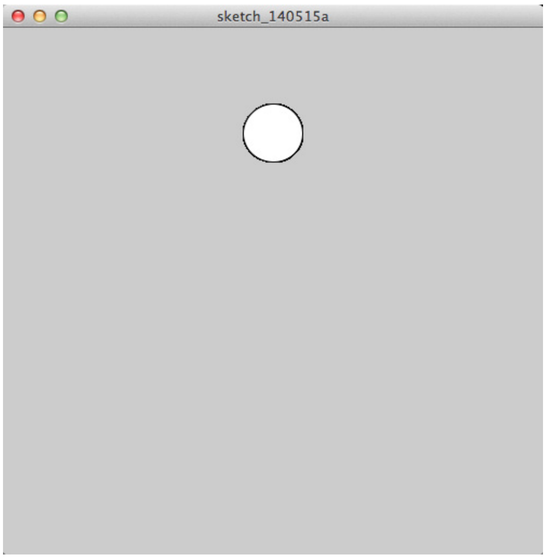


STEP 2: DRAWING SHAPES

Adding 2D shapes in processing is very simple. All that you need is to add one line of code to your draw ( ) function.

```
void setup() {
  size(500, 500);
}

void draw() {

  ellipse(250, 100, 55, 55);
}
```

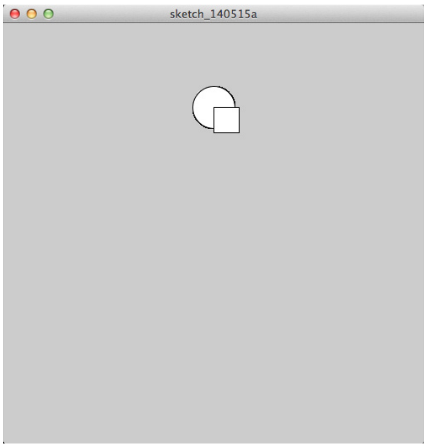
When you hit run can you see an ellipse shape draw on your window?



Adding four sided shapes, squares and rectangles are just as simple as circles.

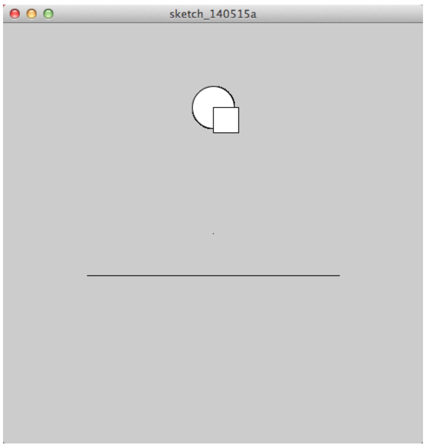
Lets add a square to your code, place it under your ellipse:

```
void setup() {
  size(500, 500);
}

void draw() {

  ellipse(250, 100, 55, 55);
  rect(250, 100, 30, 30);
}
```



Lets see how we draw lines and points.

```
line(100, 300, 400, 300);
point(250, 250);
```



note Do you see the line and point under your other shapes?

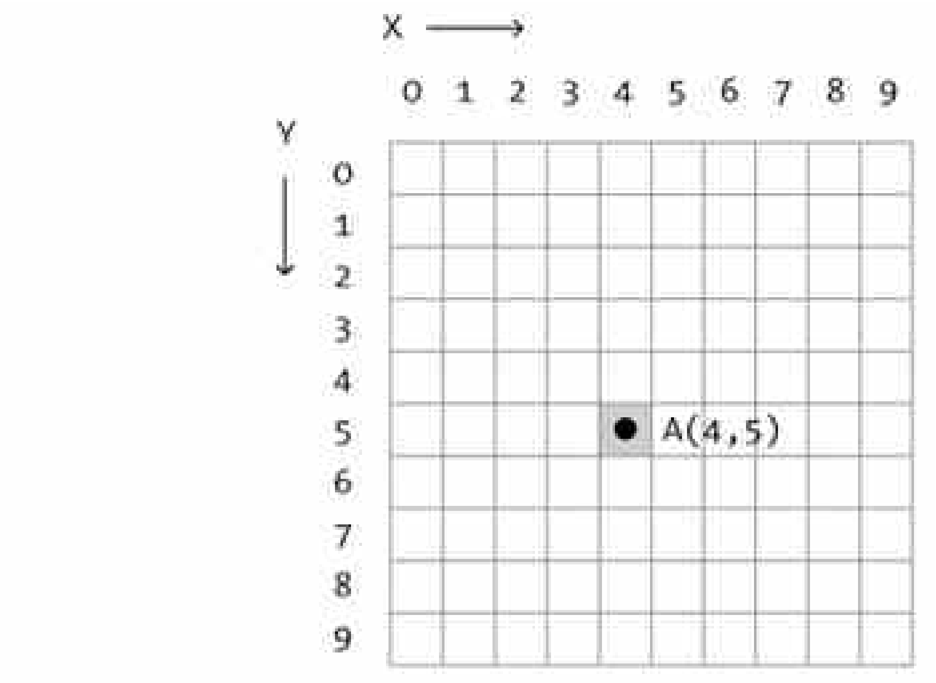
STEP 3: PLACING SHAPES

A point() is the easiest of the shapes and a good place to start. To draw a point, we only need an x and y coordinate.

```
point(x, y);
```

Example:

```
point(4, 5);
```



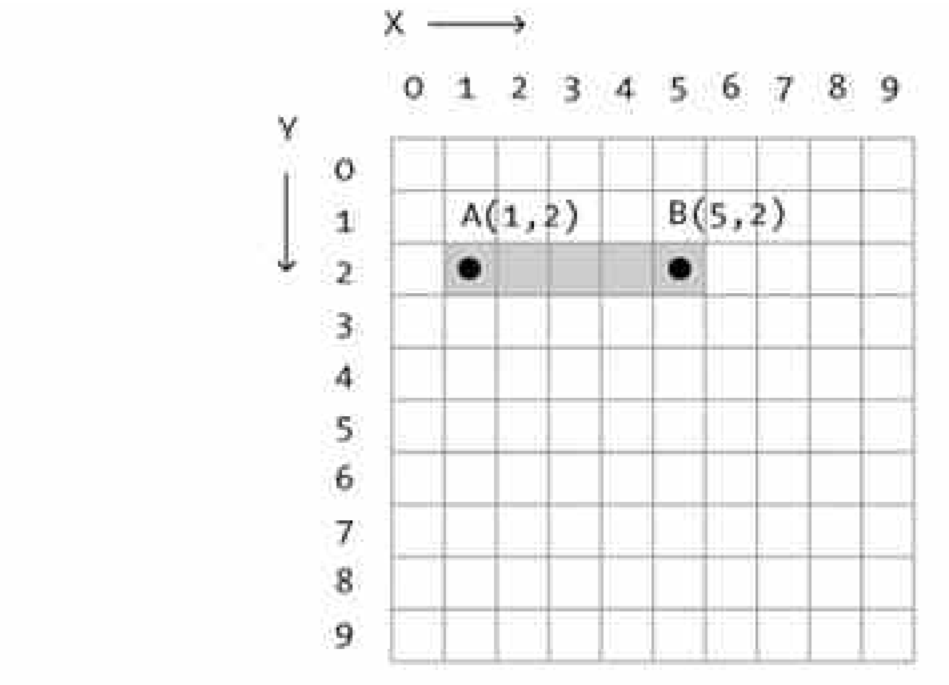
STEP 2: SHAPES: LINE

A line() isn't terribly difficult either and simply requires two points: (x1,y1) and (x2,y2):

```
line(x1, y1, x1, x2);
```

Example:

```
line(1, 2, 5, 2);
```



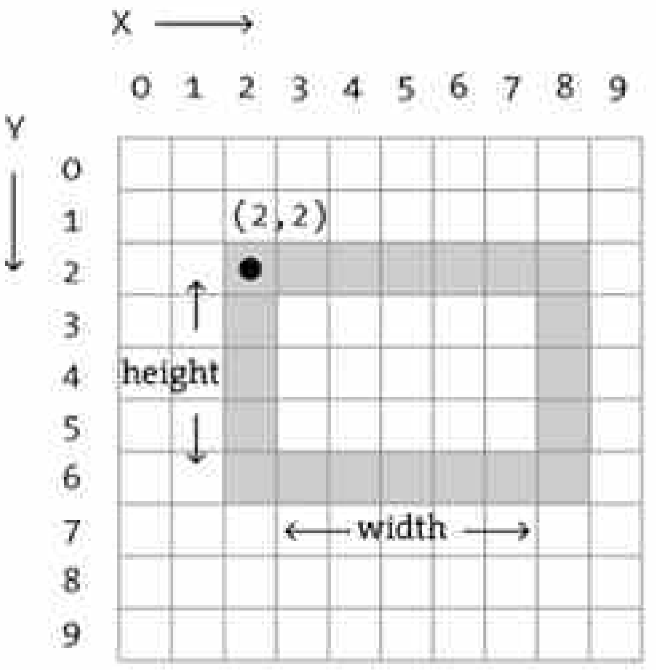
STEP 3: SHAPES: RECTANGLE

Once we arrive at drawing a rect(), things become a bit more complicated. In Processing, a rectangle is specified by the coordinate for the top left corner of the rectangle, as well as its width and height.

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

Example:

```
rect(2, 2, 7, 5);
```

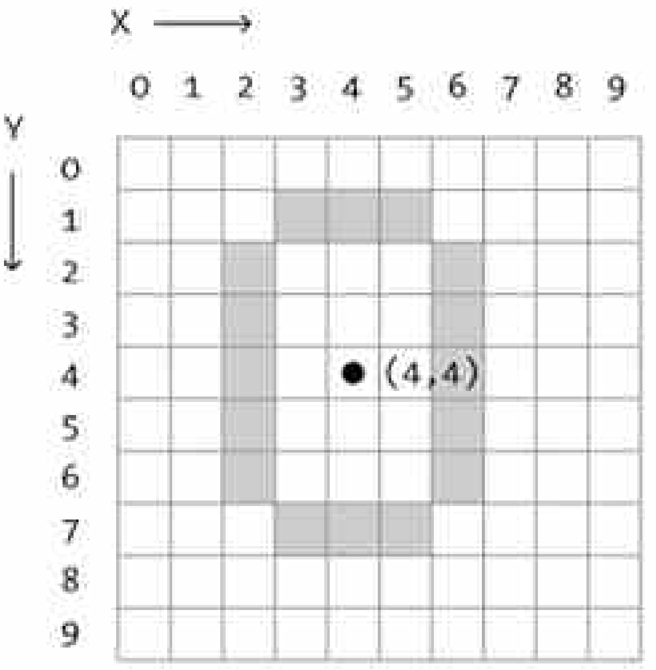


Once we have become comfortable with the concept of drawing a rectangle, an ellipse() is a snap. In fact, it is identical to rect() with the difference is that an ellipse is drawn where the bounding box of the rectangle would be. The default mode for ellipse() is "CENTER", rather than "CORNER."

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

Example:

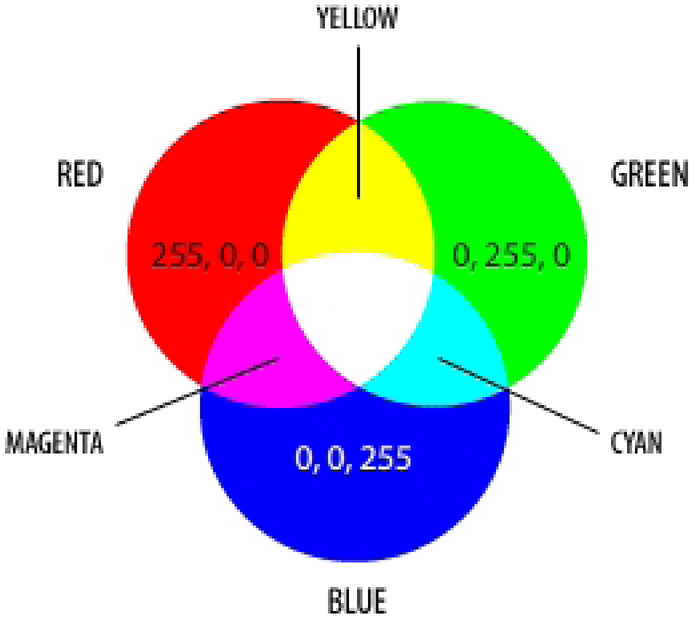
```
ellipse(4, 4, 4, 7);
```



note Now we can understand how to draw and place our shapes in the code, make the existing shapes bigger and move them to another place on your Processing Window.

STEP 4: ADDING COLOUR

In order to define colours, Processing, like most programming languages use the RGB system.
RGB stands for “RED GREEN BLUE”, which are the 3 primary colours in the digital world.
Each colour can be defined by a combination of 3 values.
The first one representing the amount of RED
The second one for the amount of GREEN
The third one representing the amount of BLUE.



For example:
- Black: 0, 0, 0
- White: 255, 255, 255
- Red: 255, 0, 0
- Brown: 165, 42, 42
- Cyan: 0, 255, 255
- Gold 255, 215, 0
- Yellow: 255, 255, 0

etc…

There are more than 16 million possible combinations so time to be creative!

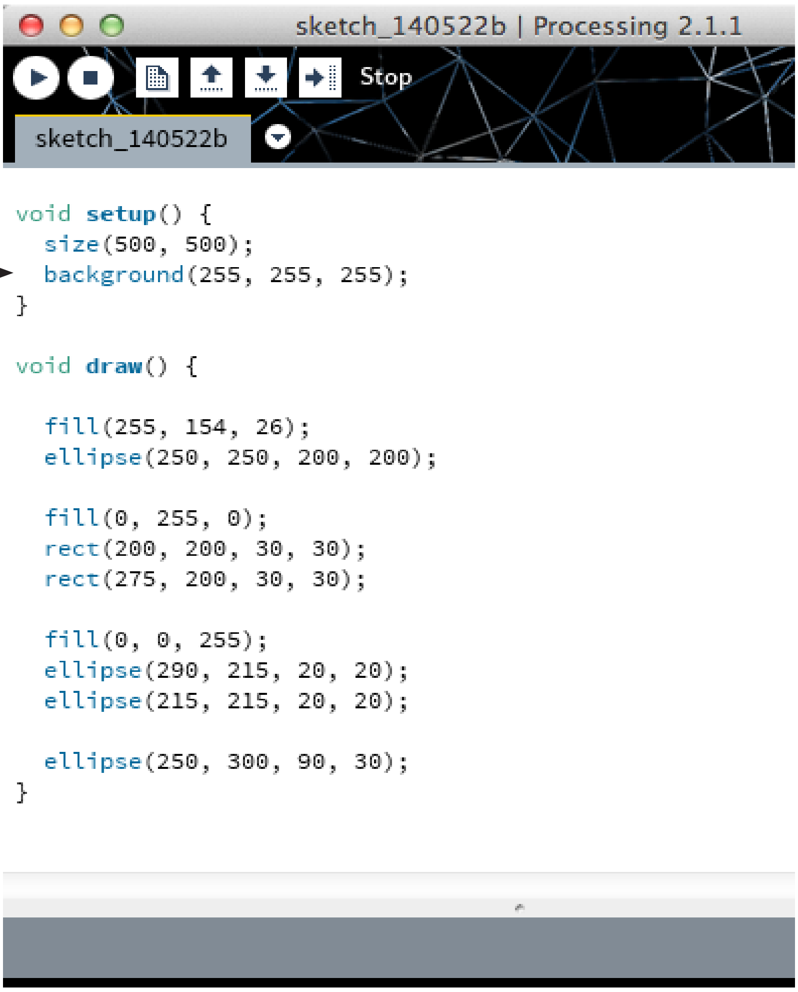
You can specify the color of your ellipse and rectangle, by adding in fill in your draw() function.

```
fill(2, 23, 100);
```

What happens when you add fill to your code?
Can you change the colour of two of your shapes?

STEP 5: LETS MAKE A FACE

Now we have know how draw shapes, where to place them and colour to the background.



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

void draw() {

  fill(255, 154, 26);
  ellipse(250, 250, 200, 200);

  fill(0, 255, 0);
  rect(200, 200, 30, 30);
  rect(275, 200, 30, 30);

  fill(0, 0, 255);
  ellipse(290, 215, 20, 20);
  ellipse(215, 215, 20, 20);

  ellipse(250, 300, 90, 30);
}
```

This is adding background colour, 255, 255, 255 is white.

The main face

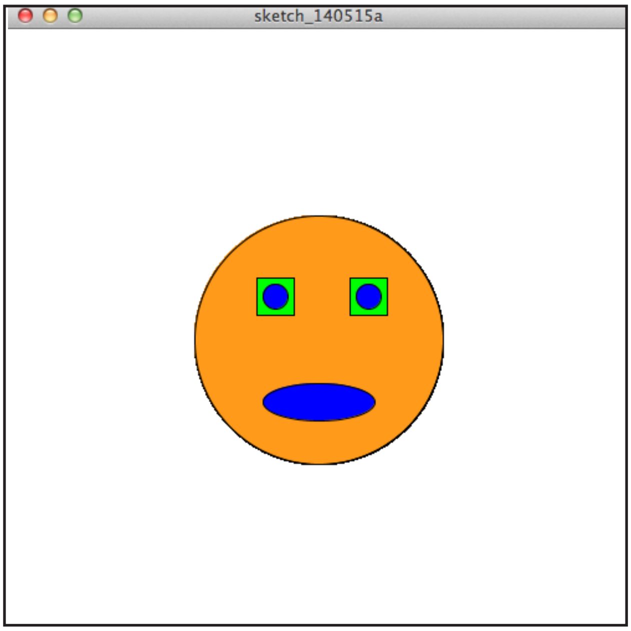
Large part of the eyes

Small part of the eyes

Mouth

Now its your turn to make a face!

Experiment using different shapes and colours!



If you would like to experiment with different shapes you can try these:

arc()
quad()
triangle()

You can goto the Processing reference page for more information on 2D shapes:

http://processing.org/reference/