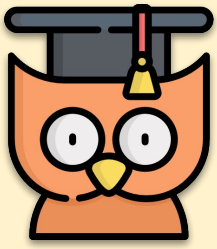
[the academy_of_code]

Grade 4
Unit 1

Shapes, Fills and Loading Images

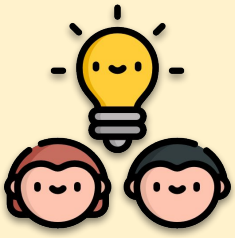


Learning Intentions

What we will learn in this lesson

Expert Tip

Top Advice from the Pro's

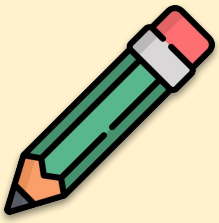


Thinking Time

Think about what is asked

Think and Talk

Discuss with a partner

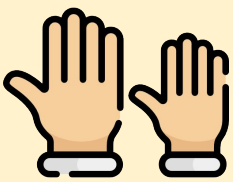


To Do

Lets Get Coding

Mouse

Click/Double Click



Ask your Tutor

Put up your hand when you need

Design

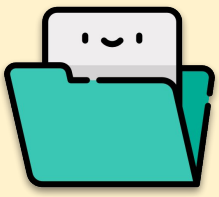
Time to get Creative

Checklist

Check over your work

Challenges

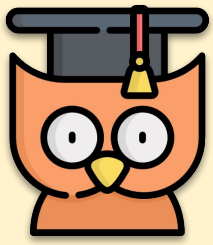
Extra work or Top Marks



Save

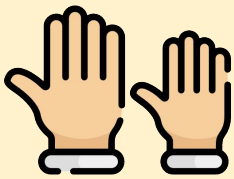
Save your work - MOST IMPORTANTLY

Lesson 1 - Getting Started



Learning Intentions:

- How to save run and open processing files - the saving conventions of file-save and file-save-as
- How to use shortcut keys to design your screen layout
- We will use our first two **functions**, **size()**, and **ellipse()**



REMEMBER: Put up your hand. We love to help!

Discussion Time

Before you get started think and chat about:

- What is coding?
- Is code the most important language in the world?
- What type of people code?

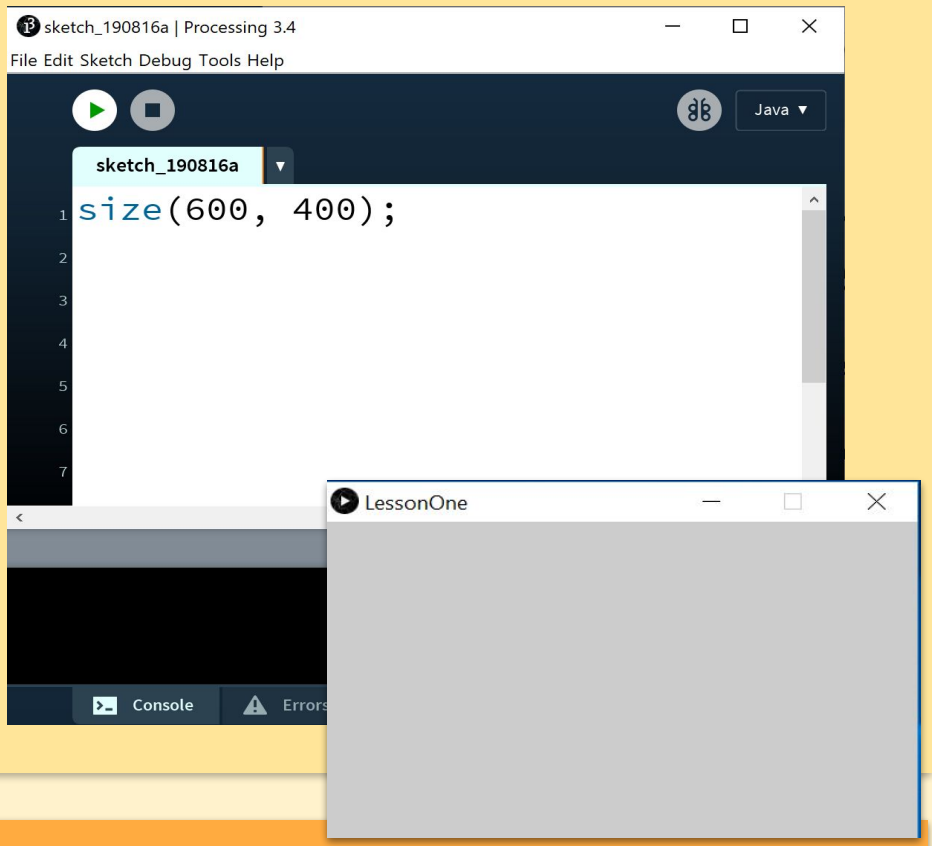
Watch these video on *The Art of Creative Coding - An Intro to Processing* and *Why schools don't teach code*

Let's Get Coding

Lets dive in. **Open Processing** (if you have trouble, tell your tutor) and write the following code **exactly** as it appears here paying attention to brackets and semicolons.

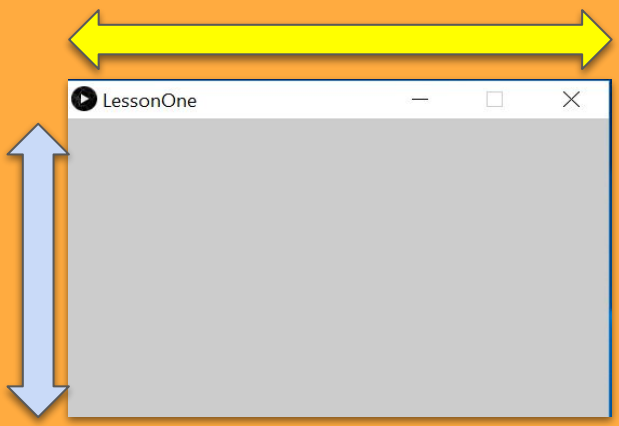
Press Play to run the sketch.

We have just created a **Display Window** on our computer that is **600 pixels** wide and **400 pixels** long.



Expert Tip

The first number is the **width** **size(600, 400);**
(the distance you go **across** from the left)



The second number is the **height** **size(600, 400);**
(the distance you come **down** from the top)

These **numbers** that we enter into our functions are called **PARAMETERS.**

Thinking Time

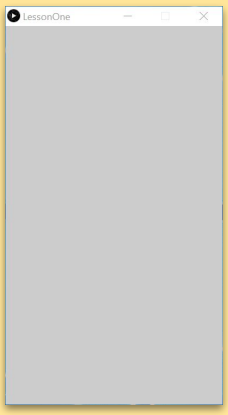
- Will the display window for **size(300, 500);** be wider or taller?
- Will the display window for **size(500, 300);** be wider or taller?

To Do

First Processing challenge: Two display windows

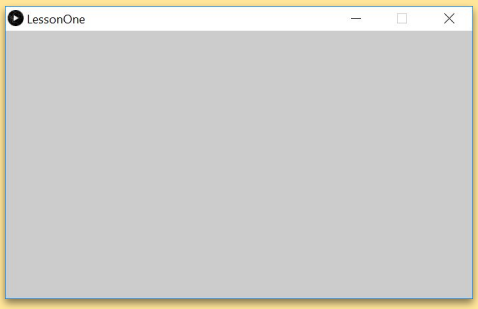
Imagine you are an App Developer

Create a sketch, changing the size **parameters** so that the display window is a similar shape to a smartphone screen



Now imagine you're a Games Developer

Create a sketch, changing the size **parameters** so that the display window is a similar shape to a widescreen TV/Monitor:

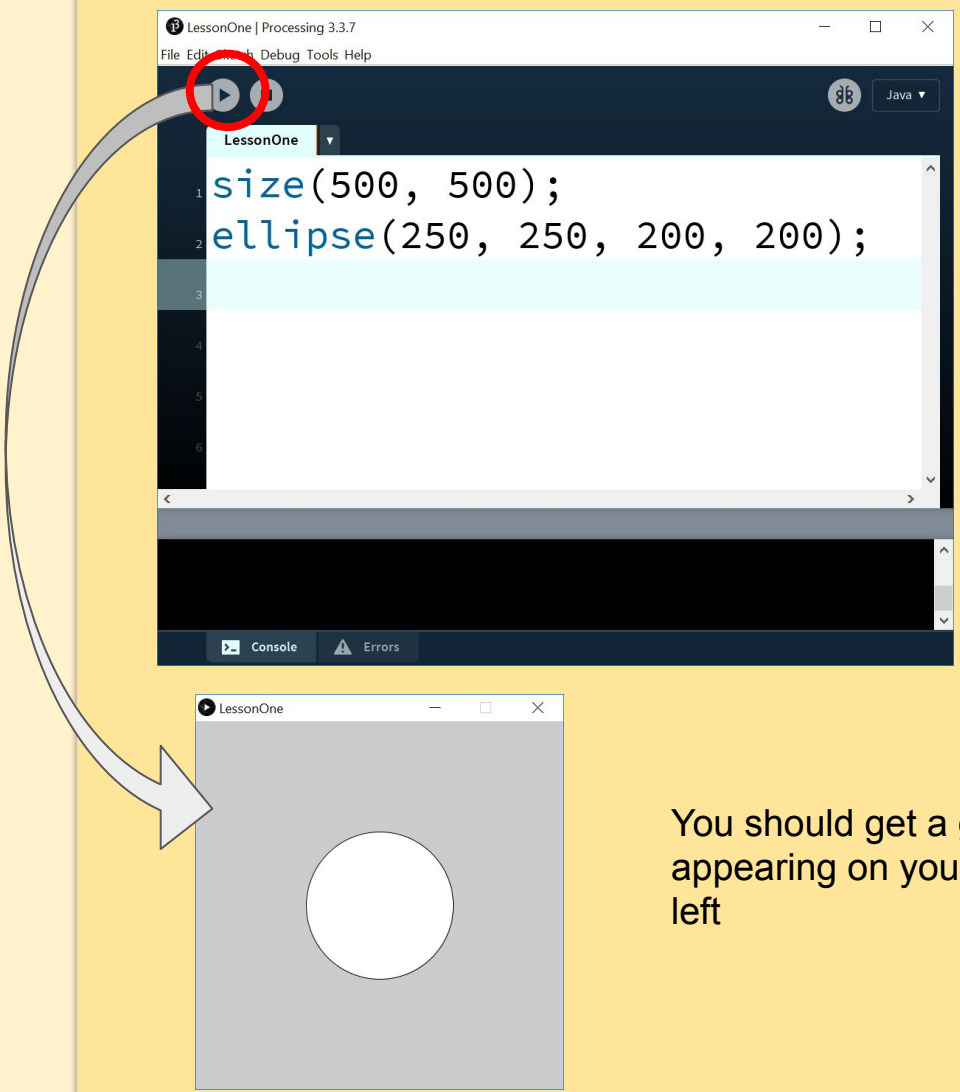


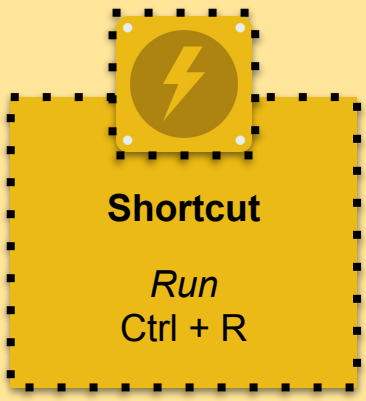
Sketching in the Display window

So far we have designed screens for our sketches. Now we are going to code to make a shape appear in the Display Window using our second **function**:

ellipse() - A posh name for a circle/oval

Write the two lines of code below in the **text editor** then click on the ***run button (or try Ctrl + r)***





You should get a grey box with a white circle appearing on your screen, like the one to the left

Thinking Time

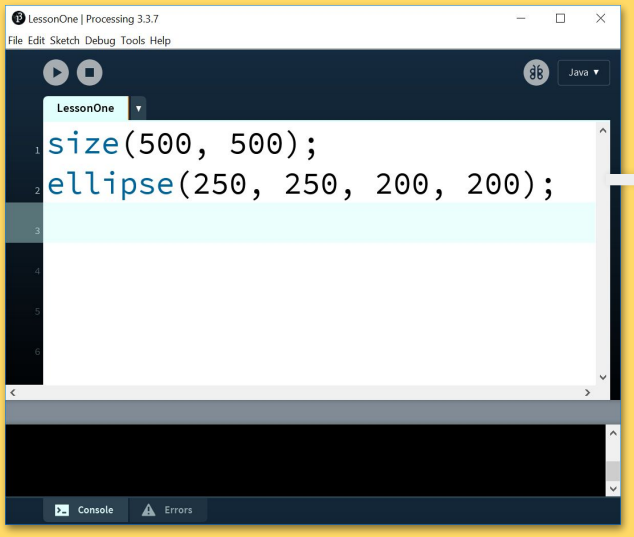
Decoding the **function**: **ellipse()**

What do you think the four numbers mean?

Congratulations

You have created a sketch in Processing

The ellipse() function in pictures



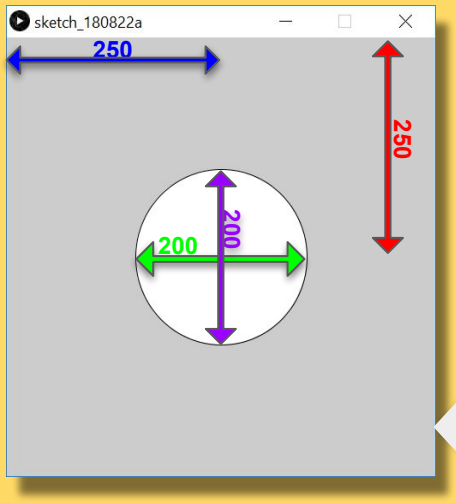
How far across the screen the centre of the circle is

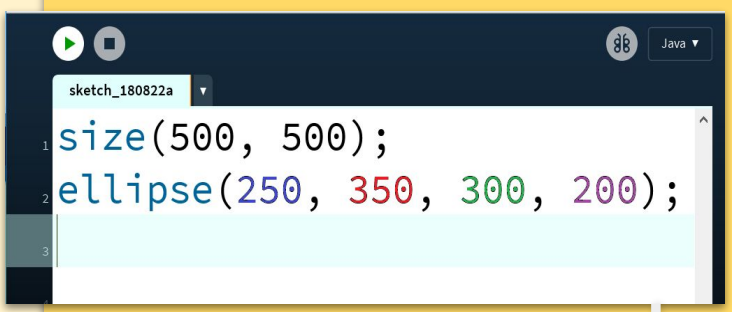
The width of the circle

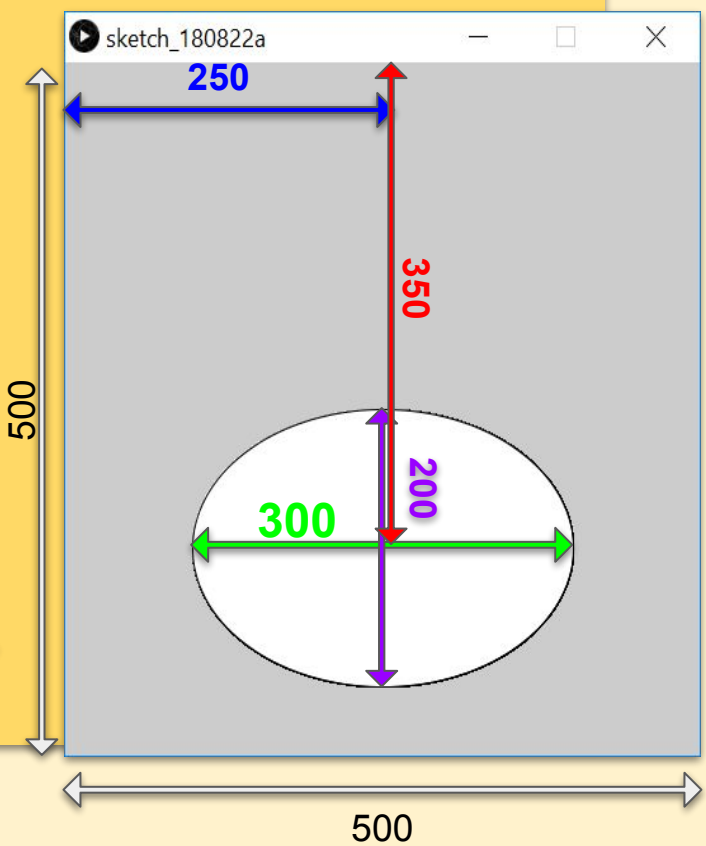
```
ellipse(250, 250, 200, 200);
```

How far down the screen the centre of the circle is

The height of the circle



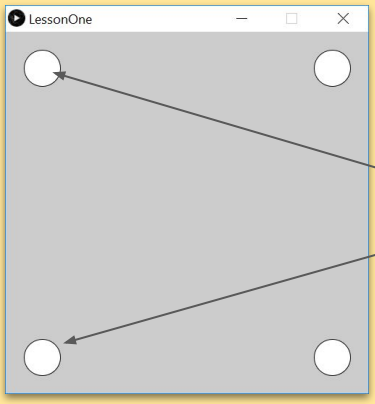




Lets get Coding!

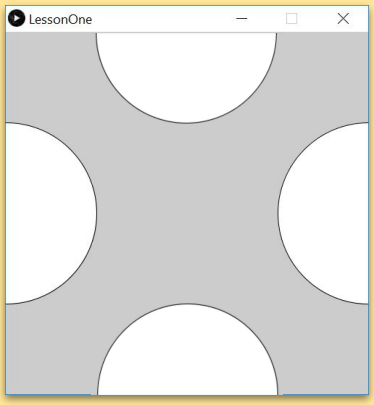
Use the **ellipse() function** to complete the following tasks. If you are confused, ask a tutor.

Draw a circle/oval in each corner of the display window




```
size(500,500);
ellipse(50,50,50,50);
ellipse(50.....
```

Now draw four circles that are cut in half by the sides of the screen.



Shortcut

COPY
Ctrl + C

PASTE
Ctrl + V

Challenge 1

Change the 4 circles so that they're different shapes and sizes. Try using **rect()** instead of **ellipse()** to make a rectangle. We'll learn more about these later.

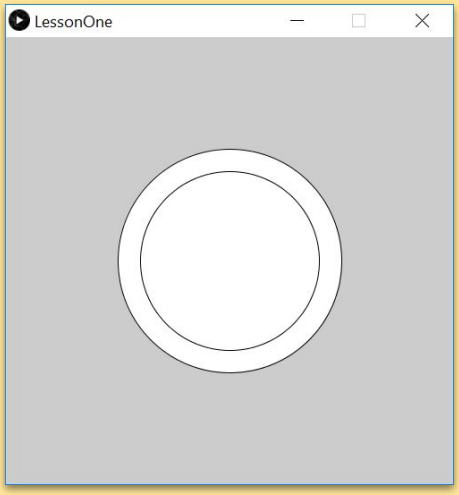
Challenge 2

Make a dice face for each of the six numbers on a dice. Save each one as different files in the same project folder. (Remember **Copy** and **Paste** to save yourself the work!)

Saving your file

Now is a good time to save up!

Draw a circle inside a larger circle. ***Hint**: They'll have the same centres*



Challenge 3

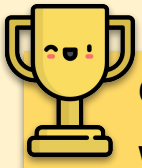
Building on from the last target, make a bigger **canvas** (window) that will take many circles from big to small. It'll look something like this.

Try following a pattern where your circle's diameter gets less making a number pattern (e.g. 300, 280, 260, 240 etc).

Remember: Only the size parameters will change, the position won't.

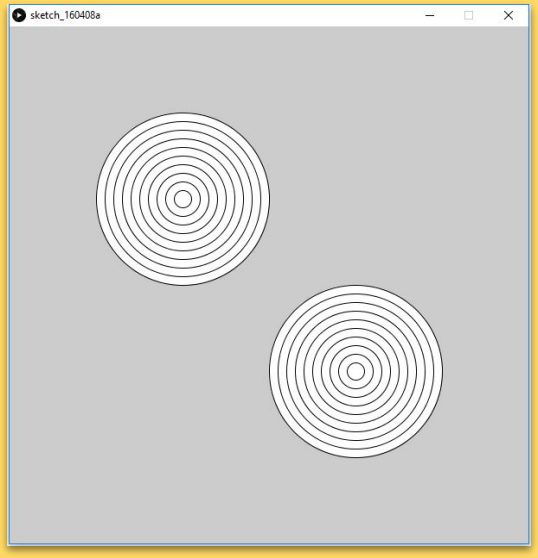
Saving your file

Now is a good time to save up!



Challenge 4 & 5

With the target you have created in Challenge 3, make a bigger canvas and try and recreate these pictures.



For the picture on the right, you'll be using **rect()** instead of **ellipse()** (to draw the rectangles) try and see if you can make this drawing. However, be careful! Rectangles are positioned from the top-left corner, not the centre, so this one is a lot trickier and takes a little bit more calculation!

Expert Tip

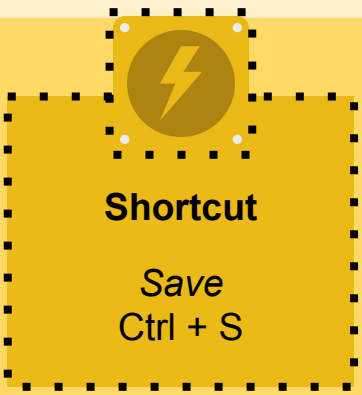
As you know, in Processing rectangles start at the top-left corner. However, if you want to change that so they start at the centre (like ellipses) put the following code above where you draw your shapes:

```
rectMode(CENTER);
```

Saving your file

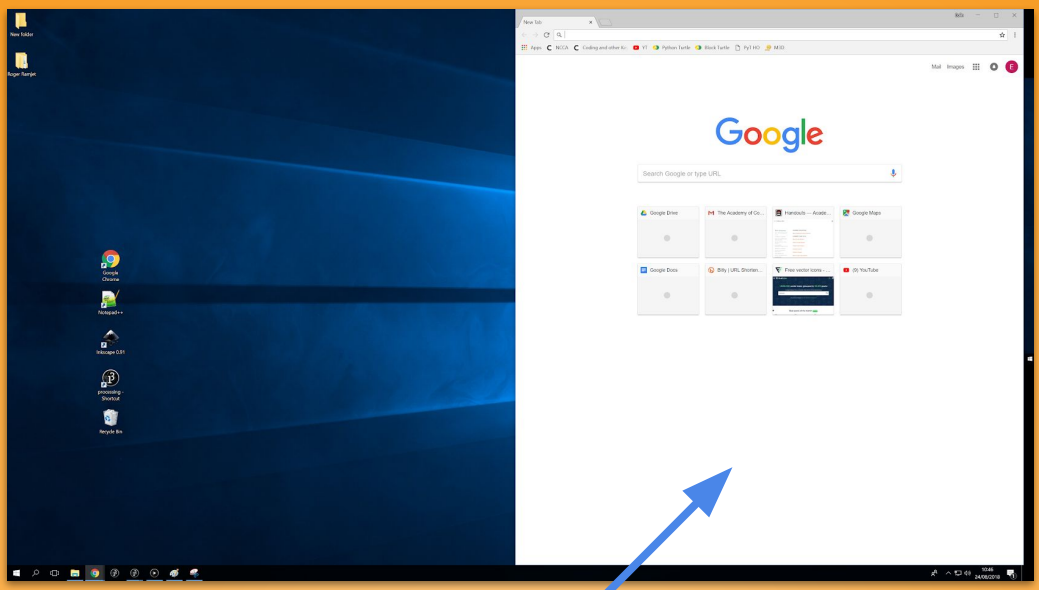
Now is a good time to save up!

Click **File** and **Save As** or try this shortcut ------->



Expert Tip

Making the browser window fill the right half of the desktop

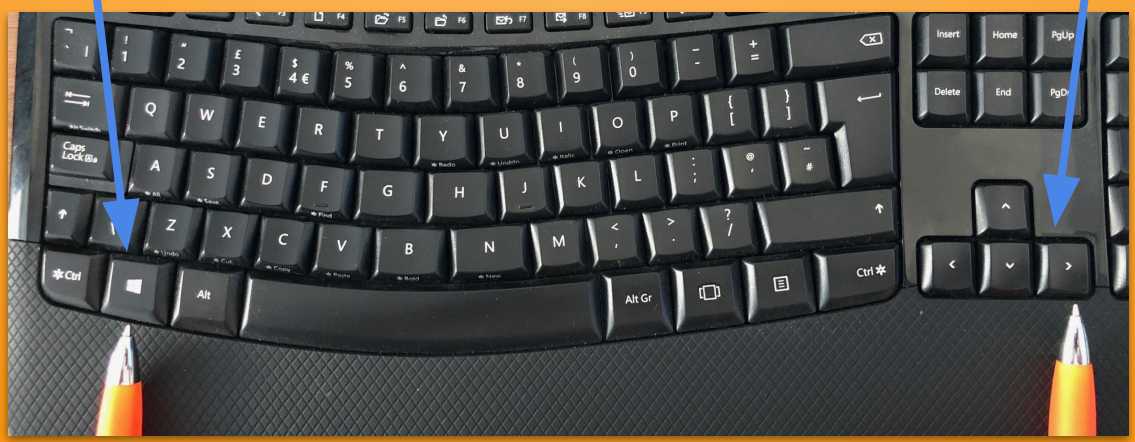


1

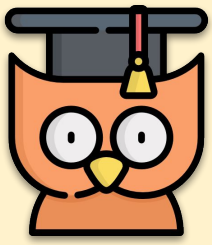
Left click on the ***browser window***

Press and hold the **Windows key** then press the **Right** or **Left** arrow key **at the same time**

2



Lesson 2 - Working with Fills

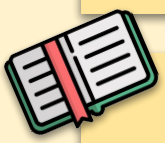


Learning Intentions:

- We will recap over our first two **functions**, **size()**, and **ellipse()**
- We will see how to colour the things that are on screen, using **fill()**

Discussion Time

Is code the most important language in the world? Have a chat and see if this video changes your perception.



Colour Sketches:

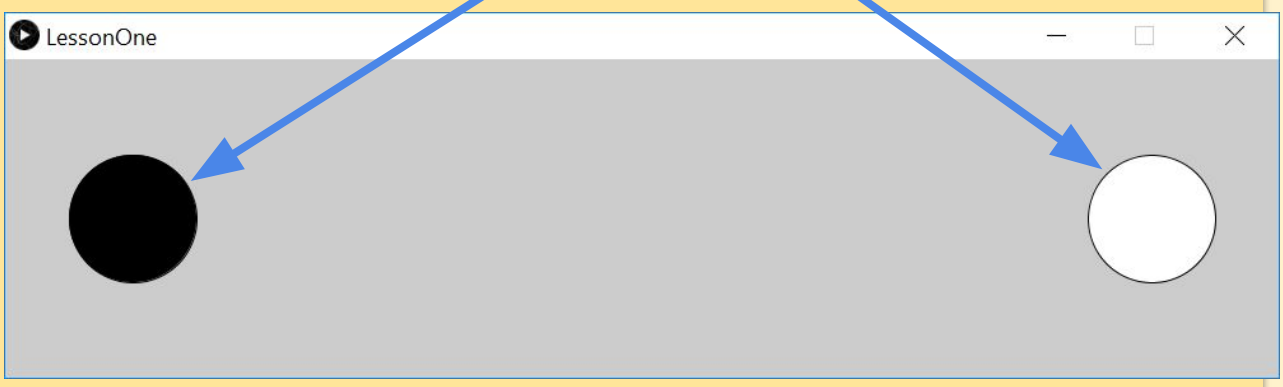
There are two main colour schemes used with **fill()**

- greyscale (black, white and everything in between)
- RGB colour

In processing and other coding languages, colours go from **0** to **255**

When dealing with greyscale

- **0** is black (no white) **fill(0);**
- **255** is white (no black): **fill(255);**



0 **255**

Lets Get Coding

We're going to look at making greyscale ellipses using **fill();**

1 Write the code below in the **text editor** then click on the ***run button*** ***(or try Ctrl + r)***

```
size(400, 400);
fill(0);
ellipse(200, 200, 325, 325);
fill(255);
ellipse(200, 200, 250, 250);
fill(0);
ellipse(200, 200, 175, 175);
```

Shortcut

Tidy
Ctrl + T

Shortcut

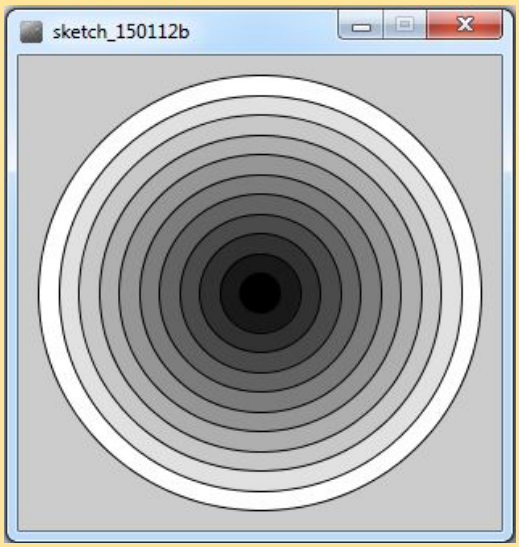
Run
Ctrl + R

You should get this sketch when it runs

2 **Now make this into a target**

Try following a pattern where your circle's diameter gets less making a number pattern (e.g. 300, 280, 260, 240 etc). **Remember:** Only the size parameters will change, the position won't.

Use **fill()** to make a series of circles that fade from white on the outside to black in the middle. You want to make something like the Display Window below.



Hint: Each circle is going to have its own fill. This should get less in a pattern. (e.g. 255, 235, 215, 195, 175.. etc)

Expert Tip

Get into the habit of Tidying, Saving and Launching at the same time using these three superfast shortcuts!

Shortcut

Tidy
Ctrl + T

Shortcut

Run
Ctrl + R

Shortcut

Save
Ctrl + S

Quick Challenge

If your finished try and see if you can use the function '**background()**' to change the colour of the background.

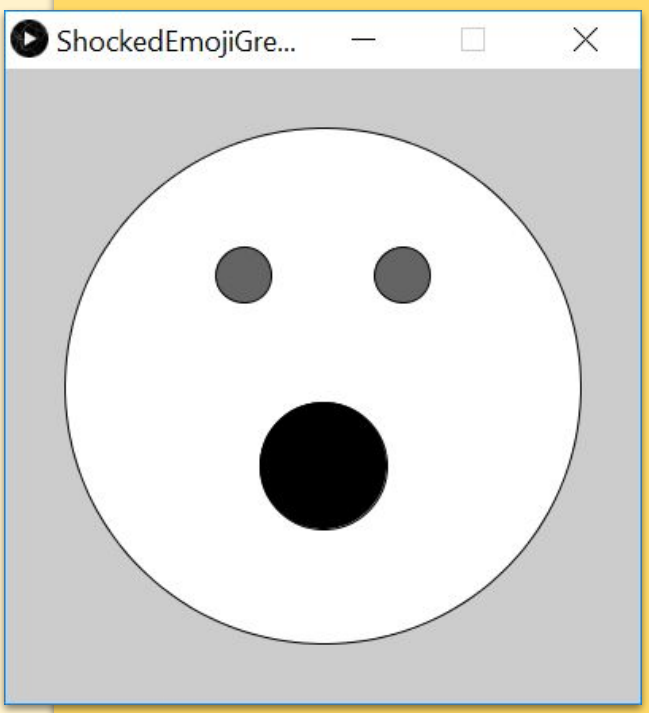
`Saving your file`

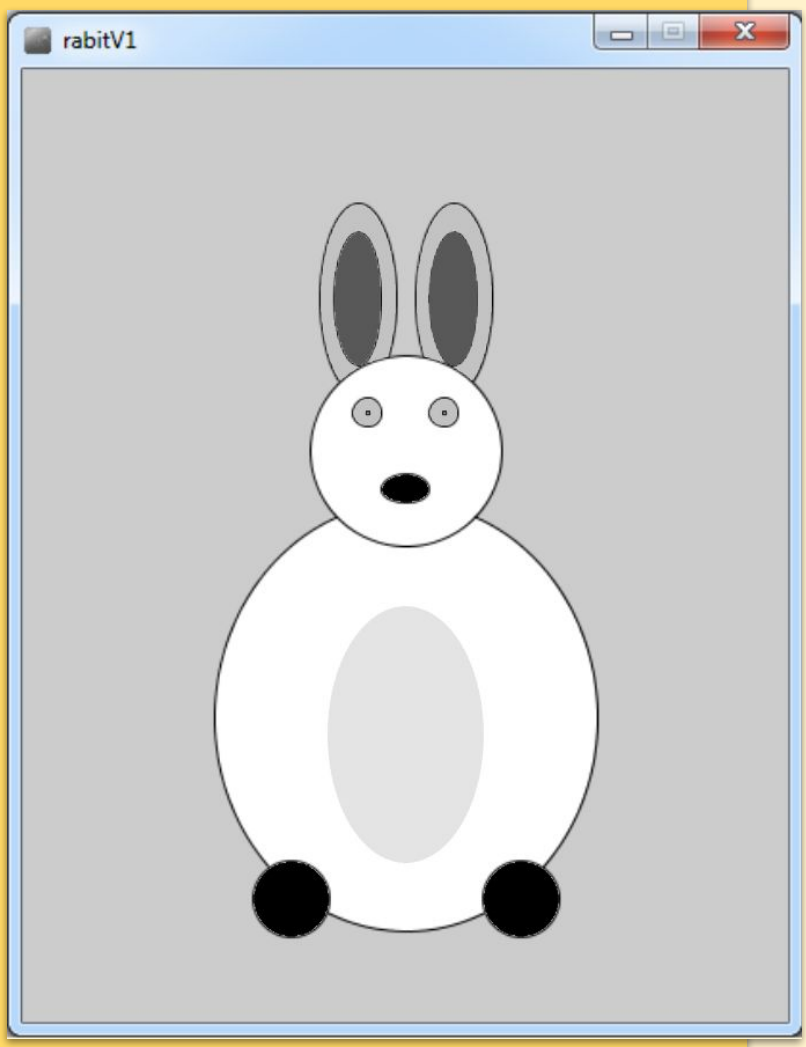
Now is a good time to save your work.

Try using **Ctrl + S.**

Making an emoji - The Beginners Challenge

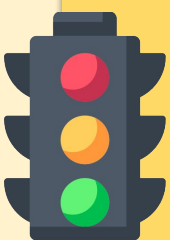
Once you have made both targets we will try something a little trickier - drawing a shocked emoji (like the one below) and colour it in in greyscale using fill(). When you have finished that, try the bunny on the right.





Grade 5 Power Up

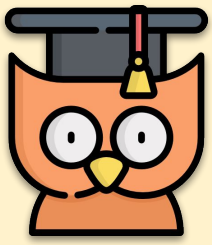
Use the triangle() function to give the rabbit a nose. *Make a colour version with moving eyes/ a waving hand (with booleans? - G5 U1)*



At this point, you should know how to:

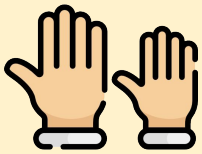
- design **Display Windows** for our sketches using **size();**
- made ovals and circles appear using **ellipse();**
    - Some of you may have tried the rect() function
- used **fill();** to shade ellipses from black through grey to white

Lesson 3 - Working with Colours



Learning Intentions:

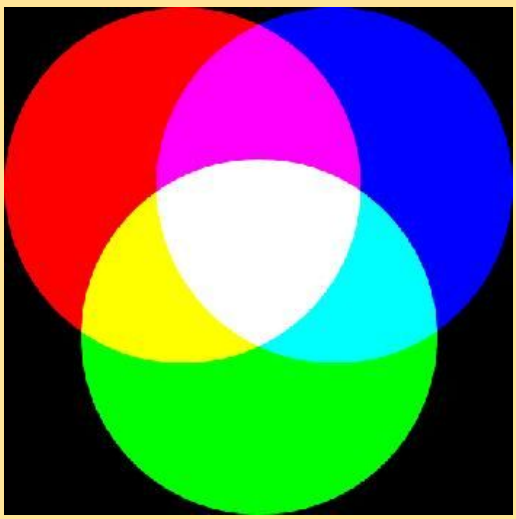
- We will see how to colour the things that are on screen, using **fill()**
- We will start to experiment with fill(R,G,B) to get different colours.



REMEMBER: Put up your hand and stay in your seat.
We love to help!

The RBG Colour space or RGB colour system

RGB Colours = **RED**,**GREEN**,**BLUE** colours. All colours on a computer's monitor/screen are made up by combining light from three colours.



RED,**GREEN**,**BLUE**

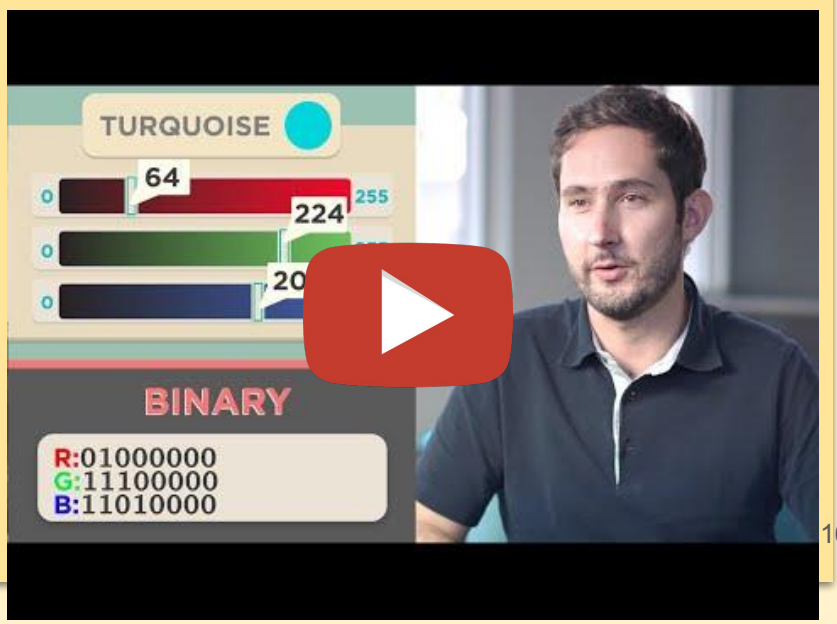
Black is **(0)** or **(0,0,0)**

White is **(255)** or **(255,255,255)**

Red is **(255,0,0);**

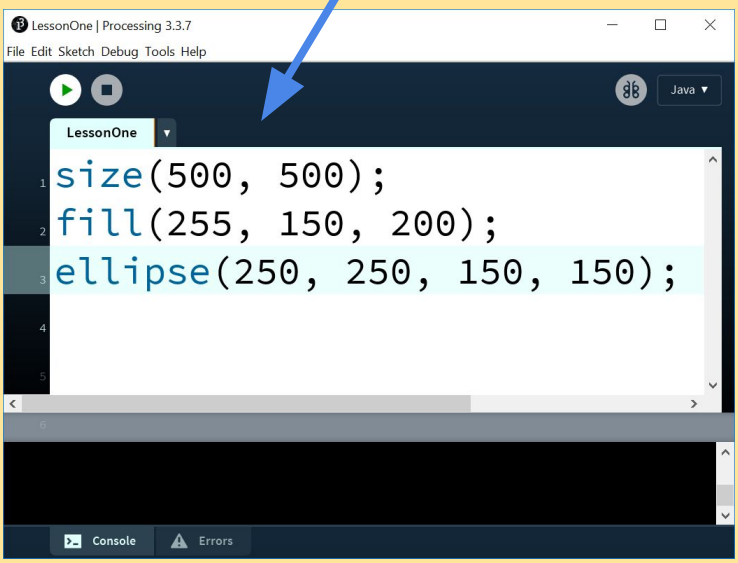
Green is **(0,255,0);**

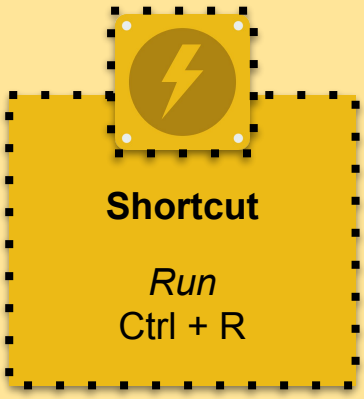
Blue is **(0,0,255);**



Multicolour Sketches

Write the code below in the **text editor**
then click on the ***run button***
(or try Ctrl + r)

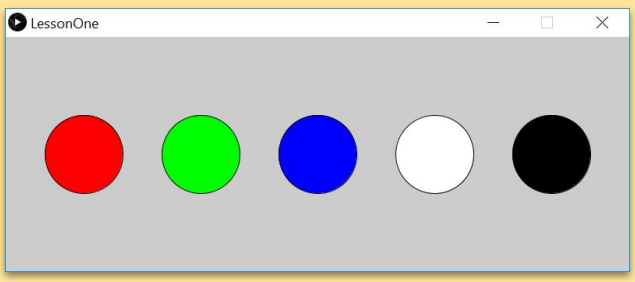




What colour does this make?

Thinking back to your previous lessons, see if you can make whats in this display window, five circles with each of these colours:

- Red
- Green
- Blue
- Black
- White

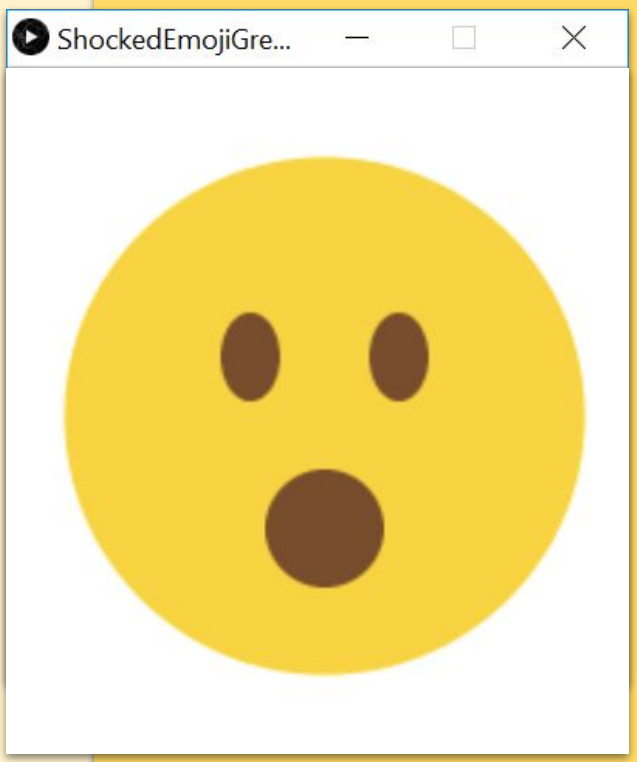


Try adding extra circles in of each of these colours

- Yellow (a combination of red and green)
- Magenta (a combination of red and blue)
- Cyan (a combination of green and blue)

Challenge

Remember the shocked emoji from the previous lesson? Try and make a coloured version of this emoji (like the one below) and colour it in these colours or in others using fill(). Save it as "**Lesson3ShockedEmojiRGB**"



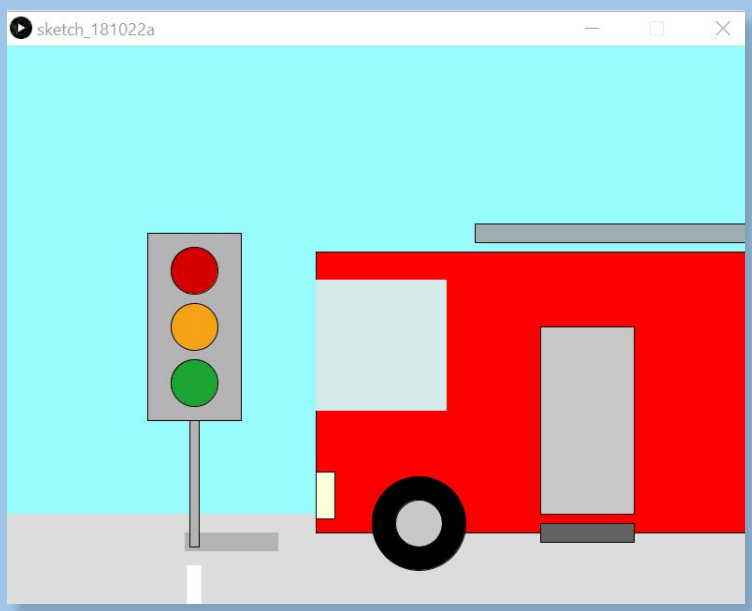
Hint: If you've saved your code properly you shouldn't have too start again.

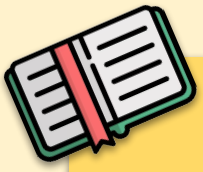
… and one last Hint:

Officially, yellow emoji faces are:
Red 249, Green 212 and Blue 66

Grade 5 Power Up

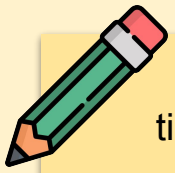
Remember the Fire truck from last year?
Make a vehicle that, using ***If statements,*** *moves with keyboard/mouse control.*





Gettin' a bit random()

Processing has an interesting function called random() which chooses a random colour for your ellipse each time you press run.



Use the **random()** function to colour an ellipse a random colour each time you press run.

You will do this by replacing **one or all three** numbers inside `fill()` with: `(random(0, 255))`

Examples:

```
fill((random(0, 255)),(random(0, 255)),(random(0, 255)));
```

This code would make the computer generate a random value (between 0 and 255) in each of the R, G and B parameters give you a totally different random colour each time.

```
fill((random(0, 255)),0,0);
```

This code would make the computer generate a random value (between 0 and 255) in just the Red parameters giving similar but slightly different random colours.

```
fill(0,(random(100, 200)),0);
```

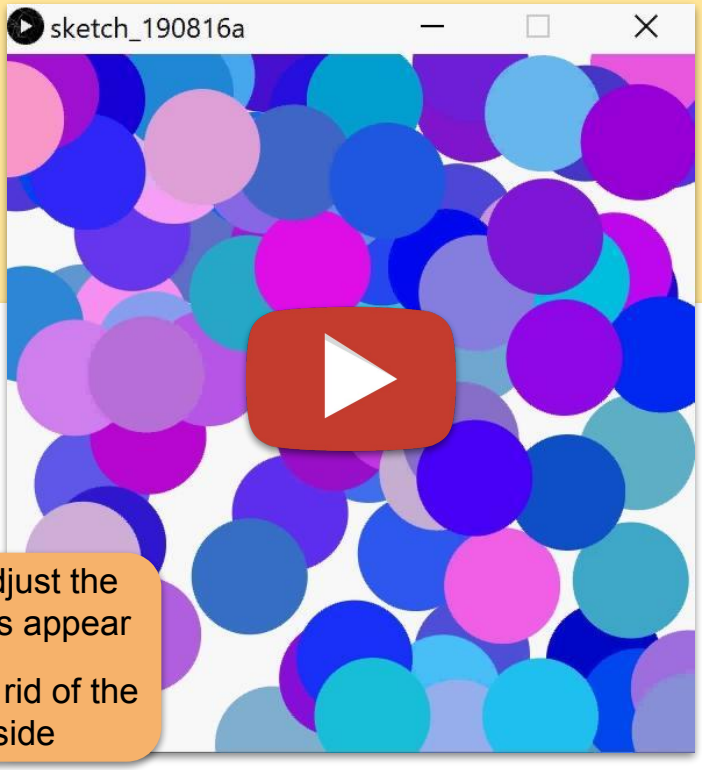
This code would make the computer generate a random value (but between 100 and 200) in just the blue parameters giving a random colour that is even more similar each time.

Change your emoji a code similar to one from above.

Now press run several times and you will see your ellipse changing to a random colour each time.

Crack the Pseudocode - Using random()

Using the random() function, try and get circles to appear similar to the given example.




```
void setup() {
  size(
  backg
}

void draw() {
  frameRate(10);
  noStroke();
  fill(
  ellipse(
}
```

A canvas size and a white background

frameRate() adjust the speed the circles appear

noStroke() gets rid of the circle's outside

random(255) random(200,255) random(100,200) random(600) random(600)

Grade 5 Power Up

Stroke, strokeWeight

Taking the above example:

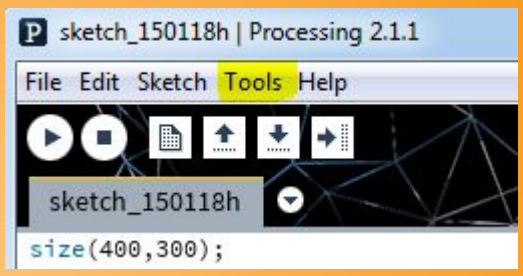
- Using an **conditional statement (if)** and **keyboard input**, expand your code so that when a certain colour is pressed, different random colours happen.
- Using a boolean (G5U1) have the circles running on a 'switch'
  - on/off
  - big/small

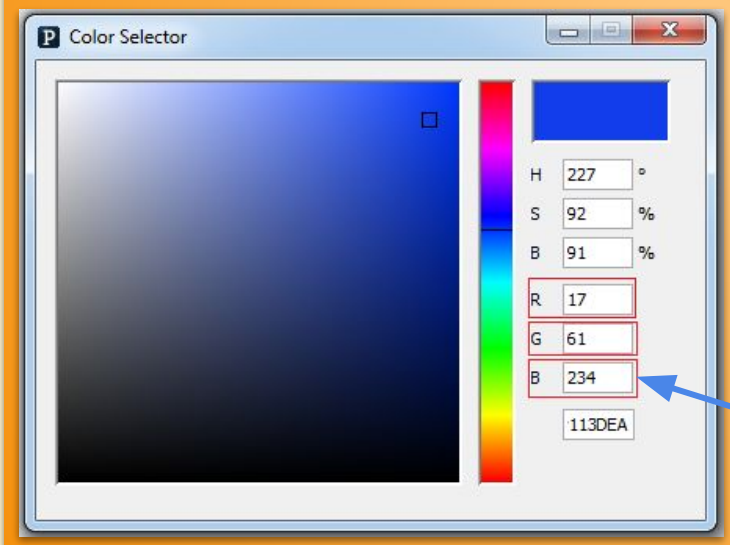
Expert Tip - Need a specific colour?

Processing can help us find whatever colour you want using the **Color Selector**.

Here's how to get to it:

- Select "**Tools**" in the Toolbar at the top
- Then select "**Color Selector**"



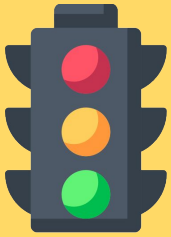


This is the colour selector, click on the colour you want and copy the **R,G** and **B** values and copy them into the `fill()` function.

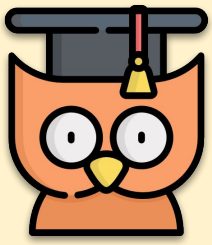
For this colour we'd take these three numbers and write:
`fill`**(17,61,234);**

Questions for the end of these lessons:

- ✔ How do you save a file? Where is the **Save As...**button?
- ✔ Which of these is a better filename: LessonOne or kodingiskool ?
- ✔ What is a **function**? Can you name three?
- ✔ What does **size()** do?
- ✔ What do we call the numbers that we enter into the **functions**?
- ✔ Can you describe using your own words what each number of **ellipse()**does?
- ✔ What is the range of colour used in Processing?
- ✔ What tool can we use to find the colours we want to use?
- ✔ What can we do with the **random() function** ?
- ✔ Where do we insert the **random() function** ?

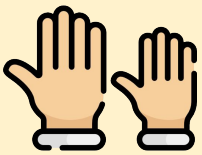


Lesson 4 - User Input



Learning Intentions:

- We will learn how to use **mouseX** and **mouseY** to programme user-controlled mouse input.
- We will learn about **void setup()** and **void draw()**



REMEMBER: Put up your hand and stay in your seat. We love to help!

First - create a new file called **LessonFourUserInput**

Thinking Time

Before we go on let's see if You can remember what all of these words and numbers mean.

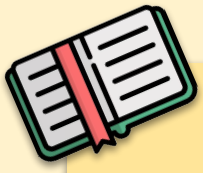
```
size(200,200);

ellipse(100,100,150,150);
```

Testing setup() and draw()

So far we've been working with images that, when run is pressed, a single image appears. In the real world we need interactive apps and software. This is where **setup()** and **draw()** come in.

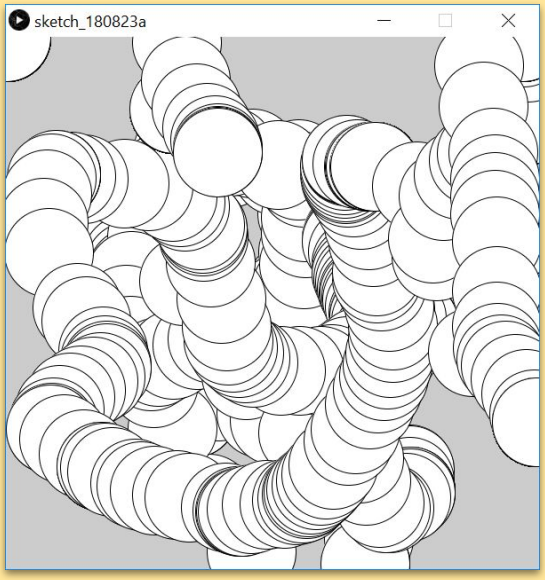
Let's give it a go. Type the following code:

```
void setup(){
 size(600,600);
}

void draw(){
 ellipse(mouseX, mouseY, 100, 100);
}
```



What is setup() and draw()

```
void setup(){
 size(600,600);
}

void draw(){
 ellipse(mouseX, mouseY, 100, 100);
}
```



This code gave us an **interactive** window. It's not the kind of application we could make much money from, but having **User input** makes it interactive, it's fun to mess around with!

void setup() and **void draw()** are what we call **code blocks**. They open and close with **curly brackets**. That is the actual technical definition for them!

Expert Tip

As soon as you use an opening bracket eg **"("**, **"{"** or **"["** close it straight away with its pair **")"**, **"}"** or **"]"**. Not doing this is the source of most problems for beginners.

- Whatever is inside **setup()** will run once when run is clicked.
- Whatever is inside **draw()** will keep running and running over and over again (60 times per second) until you close the application. This is why you made so many circles.

Tasks

Have a look at the code we wrote (above)

1. Replace **mouseX** with a number (say, 200). What happens now? Why?

2. Put **mouseX** back where it was, and now replace **mouseY** with a number. What happens? Is it different to what happened in Task 1?

3 Just for fun, switch `mouseX` and `mouseY`

eg `ellipse(mouseY, mouseX, 100, 100);`

What happens now? Why?

4 Now try having `mouseX` twice:

eg `ellipse(mouseX, mouseX, 100, 100);`

How does the circle move? Why? Try with two `mouseY`'s.

5 Try some new random variations yourself, for example:

```
ellipse(mouseX, mouseY, mouseX, mouseY);
ellipse(100, 100, mouseX, mouseY);
ellipse(mouseY, mouseX, mouseY, mouseX);
ellipse(mouseX, 100, 100, mouseY);
```

Etc.

6 Your circles leave a trail, a mark where they've been.

Change the background colour by insert **fill()** and **background()** into your code. See what happens when you put it before and after the text.

7 Use the rect() function that to write a code that has a moving coloured ellipse, that doesn't leave a trail, that has a different background colour to the rect.

Challenge 1

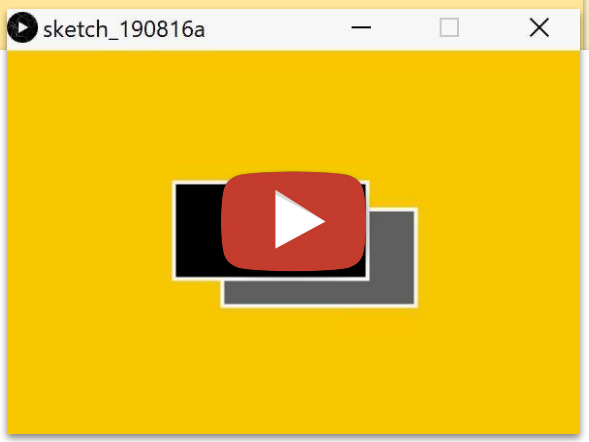
Write a code which features more than four shapes that move in different ways.

Challenge 2

Copy the code of a target, face or bunny that you made before and make this move around the screen!

Crack the Pseudocode - Built-In variables

mouseX and mouseY are what is called 'built in variables'. We'll deal more with variables in Unit 2. Other built in variables include **width** and **height**




```
void setup() {
  size(600, 400);
}

void draw() {
  background(255, 255, 0);

  rectMode(CENTER);
  strokeWeight(4);
  stroke(255);

  fill(100);
  rect(width - mouseX, height - mouseY, 200, 100);

  fill(0);
  rect(mouseX, height - mouseY, 200, 100);
}
```

A void setup containing a canvas size

void draw has a background(yellow) and the remainder of the code

rectMode(CENTER) makes rectangles behave as ellipses do

strokeWeight() thickness of the stroke (object outline)

stroke() colour of the stroke/outline

A grey fill

(width minus mouseX, height minus mouseY, 200, 100)

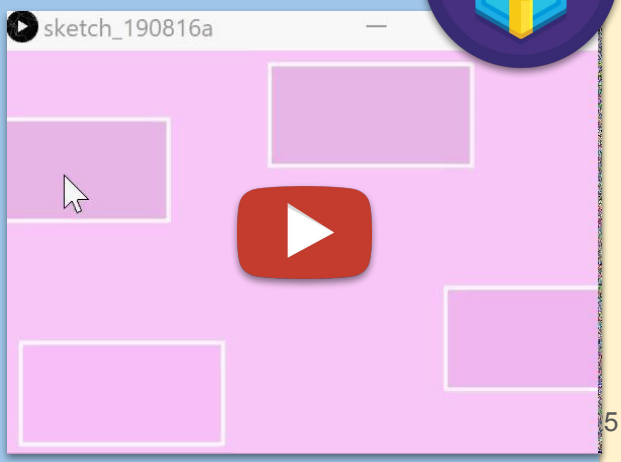
A black fill

(mouseX, height minus mouseY, 200, 100)

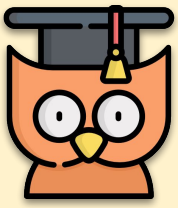
Grade 5 Power Up

Extra parameters in rect(), make the Rect rounded. *With the above example:*

- *Add a variable that controls the alpha value (4th parameter in fill function) and map it to mouseX/Y*
- *Add an extra few rectangles*

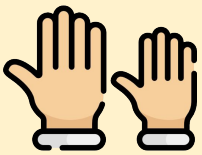


Lesson 5 - Background Image

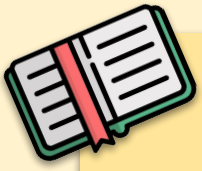


Learning Intentions:

- We will learn how to download and extract zip files
- We will see how to insert a background image into Processing
- We will code to take advantage of background image



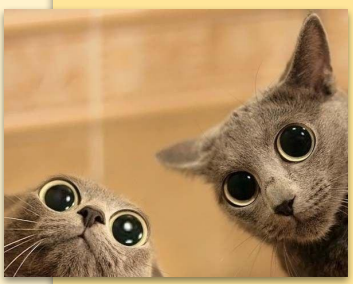
REMEMBER: Put up your hand and stay in your seat. We love to help!



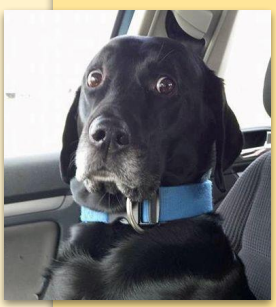
Loading External Images

So far in processing our sketches have had blank or coloured backgrounds.

In this lesson using the **function loadImage()** we will place a background image in our sketch, and using the function **resize()** we will resize the image so that it will fit in our sketch. We're going to take one of the images below and add flashing/moving eyes on to it.



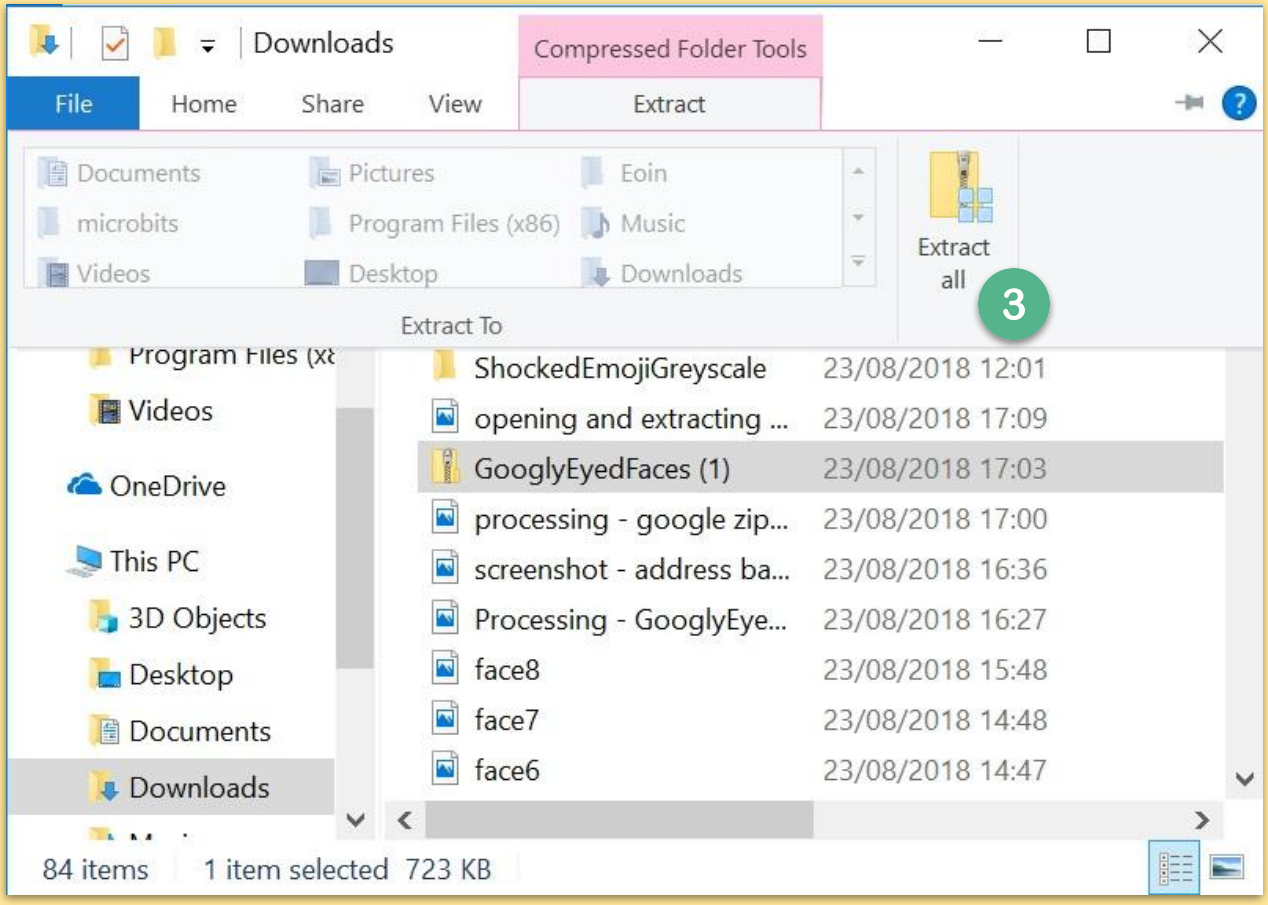




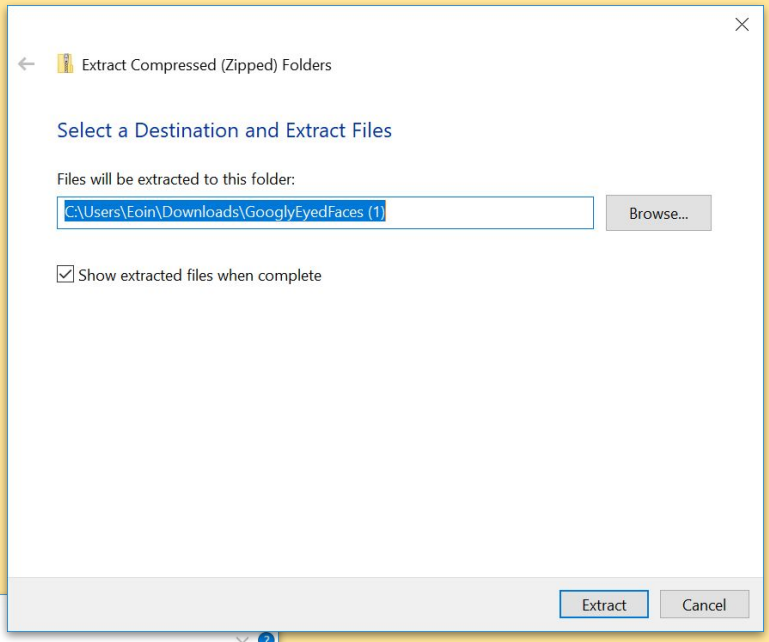

1. Go to **www.theacademyofcode.com/other-resources**

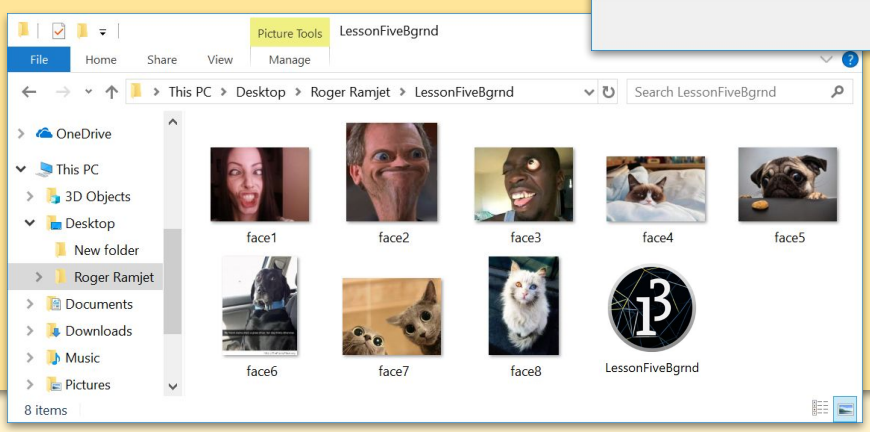
2. Download the pack called Googly Eyed Faces

3 Click on the downloaded file and click **Extract all** to extract the files.



4 Click on browse and find the location of your processing sketch (go to processing and **save** a sketch if you haven't done so yet). This will extract the 8 images ready to be programmed.





Now back to coding

Type the code below into your Sketch.

LessonFiveBgrnd | Processing 3.3.7

File Edit Sketch Debug Tools Help

LessonFiveBgrnd

```
PImage myImage;

void setup() {
  size(800,600);

  myImage = loadImage("face2.jpg");

  myImage.resize(width, height);
}

void draw() {
  image(myImage,0,0); // starts at top left, ends at bottom right
}
```

Expert Tip

Careful of the image file type

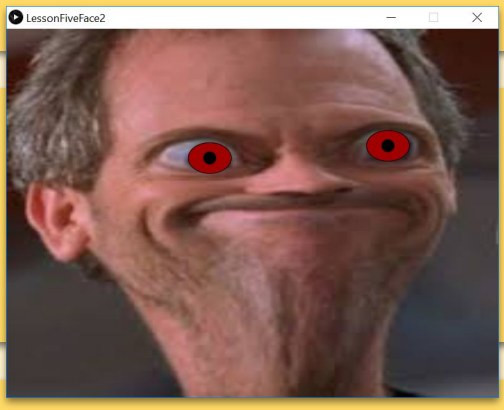
.png **.jpg** **.bmp**

Challenge 1

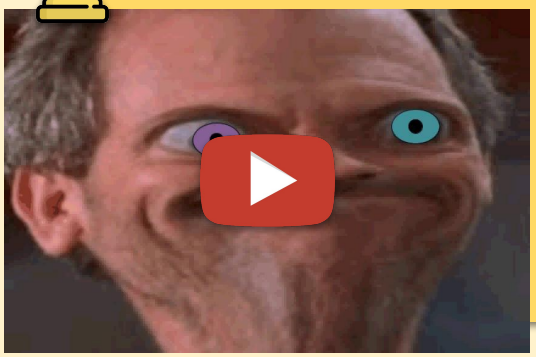
Make sure that the width and height (in size) match the pictures and that they are not skewed. Try the other face images by changing the file name face2.jpg and the size, pressing run each time

Challenge 2

Using the fill and ellipse functions, draw eyes on the face, taking time to position them.



Challenge 3



Use the **random function** to change the eyes and pupils inside the eyes of a face, so that it changes 60 times a second when run!