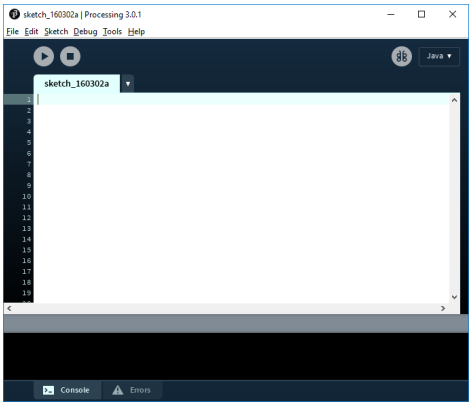
What will we be learning in this lesson?

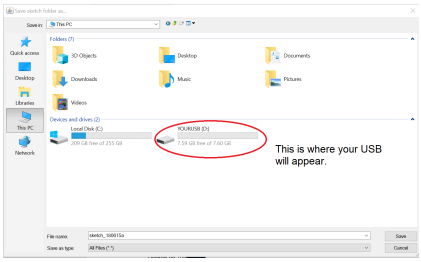
- The basic elements of Processing. We will learn how to save, run and open a file,
- We will use our first two functions, `size()` and `ellipse()`.
- We will see how to colour the things that are on screen, using `fill()`.



How to save your file:

To start, we're going to see how to save your coding file. If you look at your screen right now, you'll see a window like the one to the right. This is Processing!

To save a file in Processing, you first need to click the **File** button in the top left. Then click **Save As...**. A new screen should pop up, this is your saving screen.



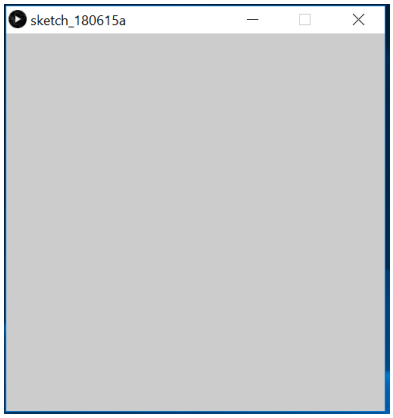
Now you need to find your USB. It's normally on the menu on the left hand side of the save screen, but sometimes you need to find it under **this PC**. Make sure you double click into your USB.

Once you've gotten onto the USB, you need to name the file. Pick a name reflecting what you're working on. For example, the file that you work on for this lesson should be called **LessonOne**. Then, click the save button and the computer will save your file.

It's important to remember to always save your file before you start coding.

Writing our first code:

Now, we're gonna write our first bit of code. We're going to make a window appear on the screen, using an important function: `size()`.



Try writing this line of code below and click the run button ( ).

```
size(500, 500);
```

You should get a grey box appearing on your screen, like the one to the right. Congrats, you've done some successful coding.

We have just created a new window on our computer that is 500 pixels wide and 500 pixels long. How do we know it's 500 x 500? These are the numbers that we put into our `size()` function. The two numbers inside the brackets of the `size()` function determine the dimensions of the window that is created. The first number is the **width** and the second is the **height**. These numbers that we enter into our function are called parameters.

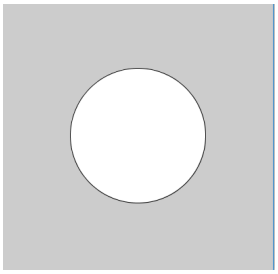
Task:

Use the `size()` function to answer the following questions. If you are confused, ask a teacher:

1. Will the window for `size(300, 500)` be wider or taller?
2. Will the window for `size(500, 300)` be wider or taller?
3. What dimensions will make the window full screen? *Hint: Screens tend to be wider*

Creating a window is kinda neat, but now we're gonna try and draw something in the window. We'll do this, using our other basic function, `ellipse()`.

Drawing Circles:

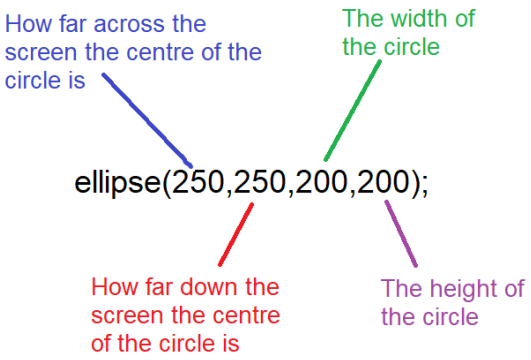


Try writing these lines of code below and click the run button ( ).

```
size(500, 500);
ellipse(250, 250, 200, 200);
```

You should see a white circle appear in the middle of your window. Let's break down what the four numbers do.

The first two parameters control where the centre of the ellipse will appear on the screen. The first deals with the horizontal position and the second deals with the vertical position.

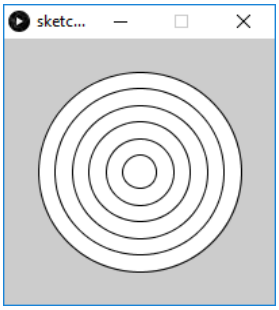


The second pair of numbers decide the size of the circle. The third number changes the width of the circle and the last number changes the height.

Task:

Try and draw the following circles. For each one, draw a new circle. When you are done, show a teacher so they can check that you understand everything.

1. Draw a circle / oval in each corner of the screen. They should all have different sizes.
2. Draw a circle that is cut in half by the sides of the screen.
3. Draw a circle inside a larger circle. *Hint: They'll have the same centres*



Bonus task:

If you're flying ahead and want a bit more of a challenge, or are refreshing yourself, try this bonus task: creating a target. It'll look something like this. If your able to do that easily enough, try making more targets dotted around the screen.

Time to colour in our circles:

We're now going to introduce the last function for this lesson, `fill()`. We use this function to colour our circles. When we use `fill()`, all the circles drawn after the function, will be coloured with the colour we picked. There are two main colour schemes used with `fill()`, grayscale and full colour. To start with, let's do grey scale.

NB: We recommend you open a new processing file and name it, LessonOneGreyScale

In Processing, colours go from **0** to **255**. For example, when dealing with greyscale (fancy name for black and white), **0 is black** (no white) and **255 is white** (100% white). This is shown in the following diagram:

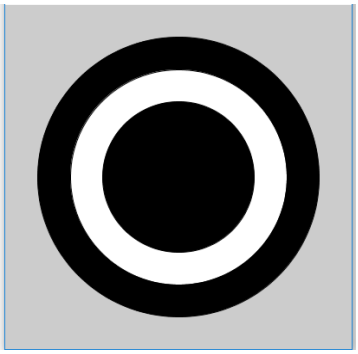
0 255

We'll try colouring in some circles using grayscale.

Write out the following code:

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

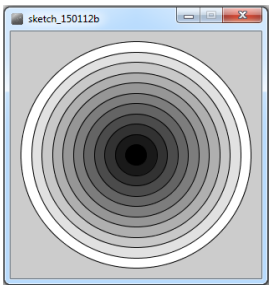
You should get the following when it runs.



Task:

1. Try and complete the target shape from the sample.

Hint: The size of the circle is decreasing by the same amount each time



2. 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 diagram. *Hint: Each circle is going to have its own fill.*

Moving onto multicolour:

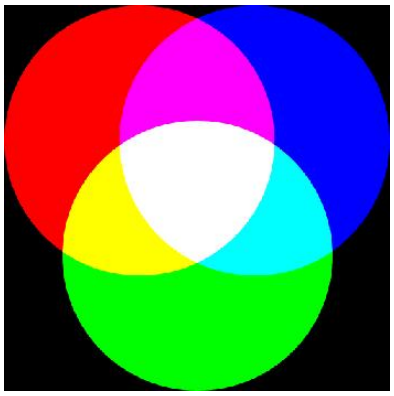
It's possible to use more colours than just black, white and grey. We can also use `fill()` with three numbers. Each number can go from 0 to 255, the same as with greyscale. They each tell processing how much **red** (the first number), **green** (the second number) and **blue** (the third number) to add. Using a combination of these, we can get every possible colour.

NB: We recommend you open a new processing file and name it, LessonOneRBG

Task:

1. Colour a circle each of these colours, You don't have to do a new circle each time.

- Red
- Green
- Blue
- Black
- White
- Yellow (a combination of red and green)
- Magenta (a combination of red and blue)
- Cyan (a combination of green and blue)



Bonus task:

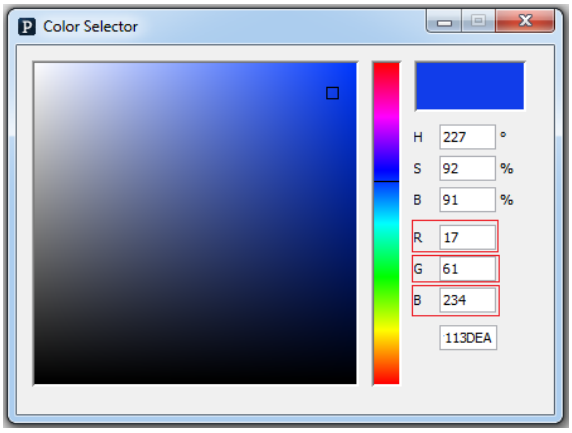
If you're flying ahead and want a bit more of a challenge, or are refreshing yourself, try this bonus task. Use the `random()` function to colour a circle a random colour each time you press run. You'll do this by putting `int(random(0, 255))` inside `fill()`.

The Colour Selector:

Processing helps us out here, by giving us a way to get the Red, Green and Blue (RGB) values for all colours. We can use a tool called the Colour Selector.

Here's how to get to it: Go to the toolbar at the top of the processing app and select **Tools**. Then select **Color Selector**

This is the colour selector, click on the colour you want. It will appear in the small box by the right hand corner. You can then copy the RGB values into the `fill()` function.



Questions for the end of the lesson?

- How do you save a file? Where is the **Save As...** button?
- Which of these is a better filename: LessonOne or kodingiskool?
- What's 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?