

# Lesson 3: Colour

## Using colour

To colour shapes or text we use the `fill()` function. Using this function is like picking up a pencil - **every shape drawn after `fill()` will be that colour until you use `fill()` again to pick a different colour.**

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

### Examples to try out!

```
size(200,200);
fill(100);      //←set the colour of shapes below to 100 (dark grey)
ellipse(150,50,100,100);
ellipse(50,150,75,75);
```

Type this in and run it. You get two circles, both dark grey (why is 100 dark grey?)

- Now try putting the fill between the two circles, like this:

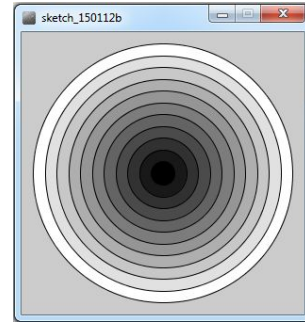
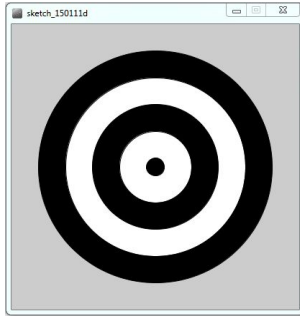
```
size(200,200);
ellipse(150,50,100,100);
fill(0);        //← set the colour of shapes below to 0 (black)
ellipse(50,150,75,75);
```

Run this - the first circle will be white (default) and the second black. We can also use the fill more than once. Let's go back to the target and make every second circle black:

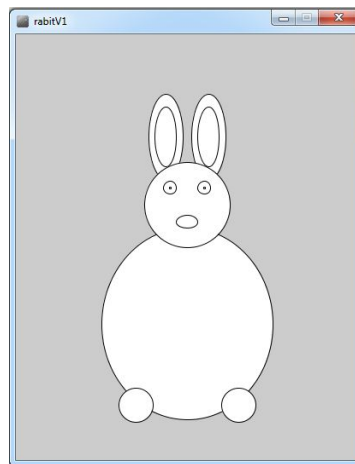
```
fill(0);          //← set colour below to 0 (black)
ellipse(200,200,325,325); //← draw outer circle
fill(255);        //← set colour below to 255 (white)
ellipse(200,200,250,250); //← draw inner circle
fill(0);          //← set colour below to 0 (black)
ellipse(200,200,175,175); //← draw further inner circle
...
```

**Tasks:**

1. Create two new programs, starting the file name with “**Lesson3Target**”
2. Using the code above as a starting point, produce the image below on the left
3. Use fill to make a series of circle that fade from white at the outside to black at the middle (below right)



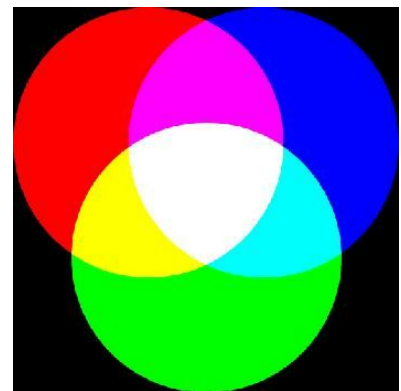
4. Once you have made both targets we will try something a little trickier - draw a bunny rabbit (like the one below) and colour it in using fill(). Save it as “**Lesson3Bunny**”

**Fill using Colour - Red, Green and Blue (RGB)**

We can also use `fill()` with three numbers. Each number tells processing how much **red** (the first number), **green** (the second number) and **blue** (the third number) to add.

Using a combination of red, green and blue we can get every colour. See right for an example.

**Note:** as you can see from the picture, if you give the maximum for each number (255) you get white. If you give the minimum on each (0) you get black.



The amounts for red, green and blue go from **nothing (0)** to as **much as possible (255)**, the same way that white did when we were doing greyscale colour.

### Example

```
size(400,300);  
fill(0,255,0); //← no red, full green, no blue  
rect(50,50,300,200);
```

### Tasks:

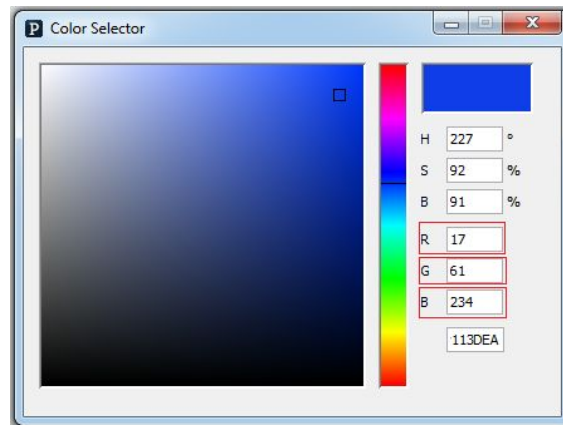
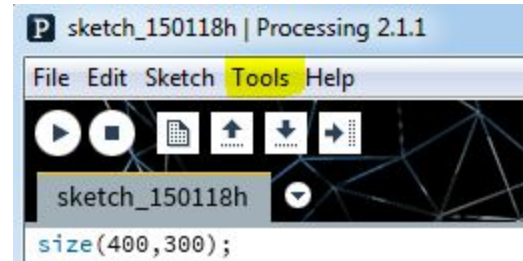
1. Save the example in a new program with a name starting with “**Lesson3**”.
2. Change the example code to make the rectangle:
  - a. red
  - b. blue
  - c. black
  - d. white
  - e. yellow (a combination of red and green)
  - f. pink (a combination of red and blue)
  - g. cyan (about 50% green and 50% blue)
3. Finally, open up “Lesson3Bunny” again and add *proper* colour this time!

## The Colour Selector:

Processing helps us out here, by giving us a way to get the Red, Green and Blue values for all colours.

Its called the Colour Selector. Here's how to get to it:

- Go to the toolbar (at the top of the processing app)
- Select "Tools" (see right)
- Then select "Color Selector"
- This is the colour selector, click on the colour you want and copy the R,G and B values (shown in a box below) and copy them into the `fill()` function.



## Task:

Using all of you've learned so far, make a new project, starting the file name with "**Lesson3Robot**" and **draw a robot**.

Let your imagination run wild! (The robot on the right is a good starting point)

Make sure you use at least one:

- ellipse
- rect
- greyscale fill command (one parameter)
- colour fill command (three parameters)

