```
size(canvas width, canvas height);
```

This function sets the size of the “canvas” or window of the Processing sketch.

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

This function draws a rectangle with the top-left corner at the x,y position passed to the function when you call it, and a width and height in pixels.

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

This function sets the color of whatever shapes you draw after calling the function.

```
fill(color variable);
```

The fill function can also be passed a color variable.

A color variable can be initialized and assigned a value like this:

```
color yourColor = color(30, 50, 100);
```

This can be broken down like so:

```
color variableName = color(red, blue, green);
```

```
line(x1, y1, x2, y2);
```

The line function draws a line between two points on the canvas, x1,y1 and x2,y2.

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

The ellipse function creates an oval of a width and height in pixels from the passed variables, with a center at the x,y coordinate passed.

```
background(red color, green color, blue color);
```

The background function is used to fill the screen with a color, either as a color variable or from 3 red, blue, and green values – just like the fill function.

```
strokeWeight(line thickness);
```

The strokeWeight function is used to set the thickness of points, lines, and the borders of shapes.

```
stroke(red color, green color, blue color);
```

The stroke function is used to set the color of strokes. It is used like the background and fill functions.

```
noStroke();
```

If you don’t want strokes, call this function in setup().

```
save(“filename in quotes and ending in .jpg”);
```

The save function is used to save an image to the same folder the source code is used in. **This function is necessary for the assignment.**

```
translate(push origin this many pixels in x, push origin this many pixels in y);
```

The translate function shifts the origin of the x,y coordinate system to the x,y coordinate passed, for all shape functions called after this translate function. So, the following:

```
    translate(50, 50);
    ellipse(0, 0, 100, 100);
```

Will draw an ellipse of a width and height of 100 at 50,50 instead of 0,0 – **because the translate function shifts the entire coordinate system to use the x,y coordinates passed as the origin.**

```
println(“text to print to console ” + a variable);
```

The println function will print text to the console, which is a small box underneath the source code in Processing. You can “concatenate” or add variable numbers to the end of the text within the quotation marks by using the plus sign and a variable name, like above.