

# Handout

## Links

- [openprocessing.org/user/161462](https://openprocessing.org/user/161462) - My openprocessing profile where you can view all the sketches we will be covering
- [p5js.org/reference/](https://p5js.org/reference/) - The official p5js reference with all the commands, and what they do

## Cheat sheet

### Shapes

```
line(startX, startY, endX, endY);  
ellipse(x, y, diameter); // drawn from centre  
ellipse(x, y, width, height); // drawn from centre  
rect(x, y, width, height); // drawn from corner
```

### Colour

- [colours.neilorangepeel.com](https://colours.neilorangepeel.com) - Reference for named colours
- [webfx.com/web-design/color-picker](https://webfx.com/web-design/color-picker) - Colour picker
- Ways of expressing colour
  - Named colours, like: 'red', 'beige', 'violet'
  - Hex codes, like: '#FF4500', '#00FF7F'
  - Red, green, blue values, each goes from 0-255
  - Single value for grayscale
  - Hue, saturation, lightness values

```
fill(0); // Set the fill colour for shapes after it to black  
fill(150,50); // Set the fill colour to a transparent grey  
background('peachpuff'); // Set the background colour to the named colour 'peachpuff'  
stroke(255,0,0); // Set the stroke to red using red, green and blue values  
background('#D2691E'); // set the background to a chocolate-y colour using a hex code  
noFill(); // remove fill from shapes  
noStroke(); // remove stroke from shapes
```

## Randomness

```
random(); // Creates a random number between 0 and 1
random(0,255); // Creates a random number between 0 and 255
circle(200,200, random(50,200)); // Circle with random diameter between 50 and 200
fill(random(0,255), random(0,255), random(0,255)); // creates a random fill
random(['red', 'green', 'blue', 'yellow']); // selects a random colour from that list
stroke( random(['olive','darkcyan']) ); // randomly sets stroke to either olive or darkcyan
```

## Variables

A way of storing a value, like a box.

```
var startingDiameter = 100; // Stores the value 100 inside of the 'box' startingDiameter
ellipse(200,200, startingDiameter); // This ellipse has a diameter of 100
ellipse(200,200, startingDiameter + 100); // This ellipse will always be 100 pixels bigger than the
other one
```

## Functions

Group commands together

```
function draw() {
  drawThreeCircles(); // Draws three circles with random postions
  drawThreeCircles(); // Draws another three randomly positioned circles
  drawRandomlyColouredCircle(90,100,50); // Draws a randomly coloured circle at position 90,100 with a
diameter of 50
}

function drawThreeCircle() {
  ellipse(random(400), randome(400), 80);
  ellipse(random(400), randome(400), 80);
  ellipse(random(400), randome(400), 80);
}

function drawRandomlyColouredCircle(x,y,diameter) {
  fill(random(255), random(255), random(255));
  ellipse(x,y,diameter);
}
```

## If/Else Statement

Decide whether or not to do a command based on a condition

```
if (random()>0.5) {  
    ellipse(100,100,50);  
}  
else {  
    rect(100,100,50);  
}
```

This command will draw either a circle or a square, each will happen about half the time

## Loops

```
for (var count = 0; count < 10; count = count + 1) {  
    // Commands to do each loop  
}
```

This loop starts at 0, and ends at 9, going up by 1 each time it loops

```
for (var diameter = 50; diameter <= 200; diameter = diameter + 50) {  
    ellipse(100,100, diameter);  
}
```

This loop starts 50, ends at 200 (notice the sign is less than or equals this time) , and goes up 50 each time. Each time the loop runs it draws a circle with the diameter it's on. So it draws four circles: one with a 50 pixel diameter, one with a 100 pixel diameter, one with a 150 pixel and one with a 200 pixel diameter.

```
for (var x = 0; x < width; x=x+10) {  
    for (var y = 0; y < height; y = y+10) {  
        rect(x,y,10,10);  
    }  
}
```

These loops create a grid of 10 by 10 pixel squares that go across the width and height of the canvas