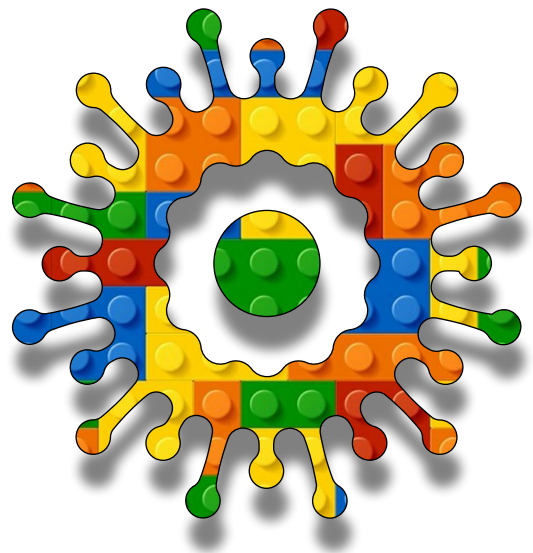


Artificial
Intelligence
Module B
Unit #1
p5.js code
snippets 3





Module B Unit #1 p5.js code snippets 3

Introduction to code snippets part 3

The index.html file

Sketch B1.1	drawing a rectangle
Sketch B1.2	drawing an ellipse
Sketch B1.3	drawing triangles
Sketch B1.4	measuring the distance
Sketch B1.5	in the name of the colour



Introduction to code snippets part 3

We are going to cover some essential coding snippets relevant to the next section on pre-trained models.



Sketch B1.1 drawing a rectangle

We have covered most of the shapes you will use or need except for this one, the **rectangle**. A rectangle has four arguments. The first two are similar to those of the square and they are the co-ordinates of the rectangle; the third is the width of the rectangle, and the fourth is the height of the rectangle.

```
function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
  rect(100, 100, 200, 50)
  rect(100, 200, 20, 100)
  rect(200, 200, 100, 150)
}
```



Notes

Drawing three rectangles, the arguments are:



x position



y position



horizontal dimension



vertical dimension



Challenge

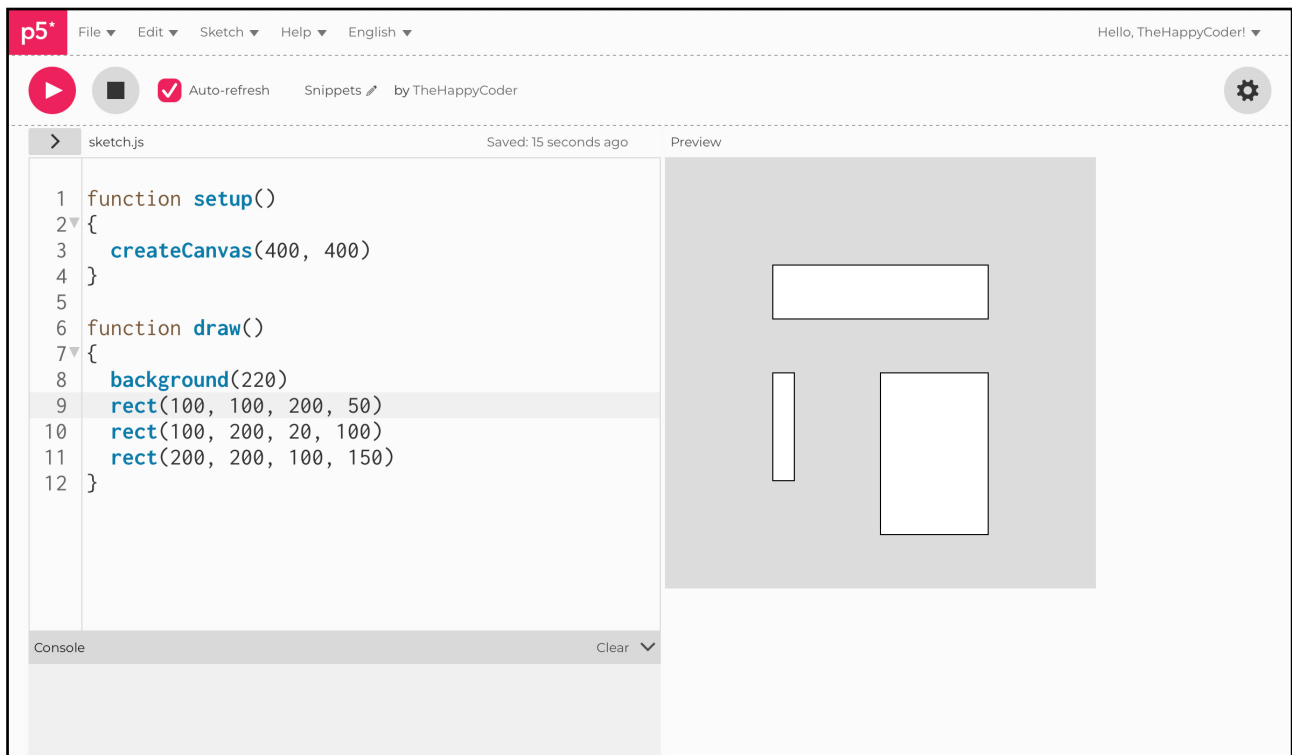
Add the **rectMode()** to put the co-ordinates in the centre of the rectangle.

Code Explanation

```
rect(100, 100, 200, 50)
```

Draws a rectangle at top left hand corner 100 from the left, 100 down and with a width of 200 and height of 50

Figure B1.1 drawing rectangles





Sketch B1.2 drawing an ellipse

Another shape that we can draw is an **ellipse**, which is similar to a circle with an extra argument.

```
function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
  ellipse(200, 100, 150, 50)
  ellipse(100, 200, 20, 100)
  ellipse(200, 250, 100, 200)
  ellipse(300, 200, 20, 100)
}
```



Notes

We draw four ellipses, the arguments are:



x position



y position



horizontal dimension



vertical dimension



Challenge

Make a pattern with them

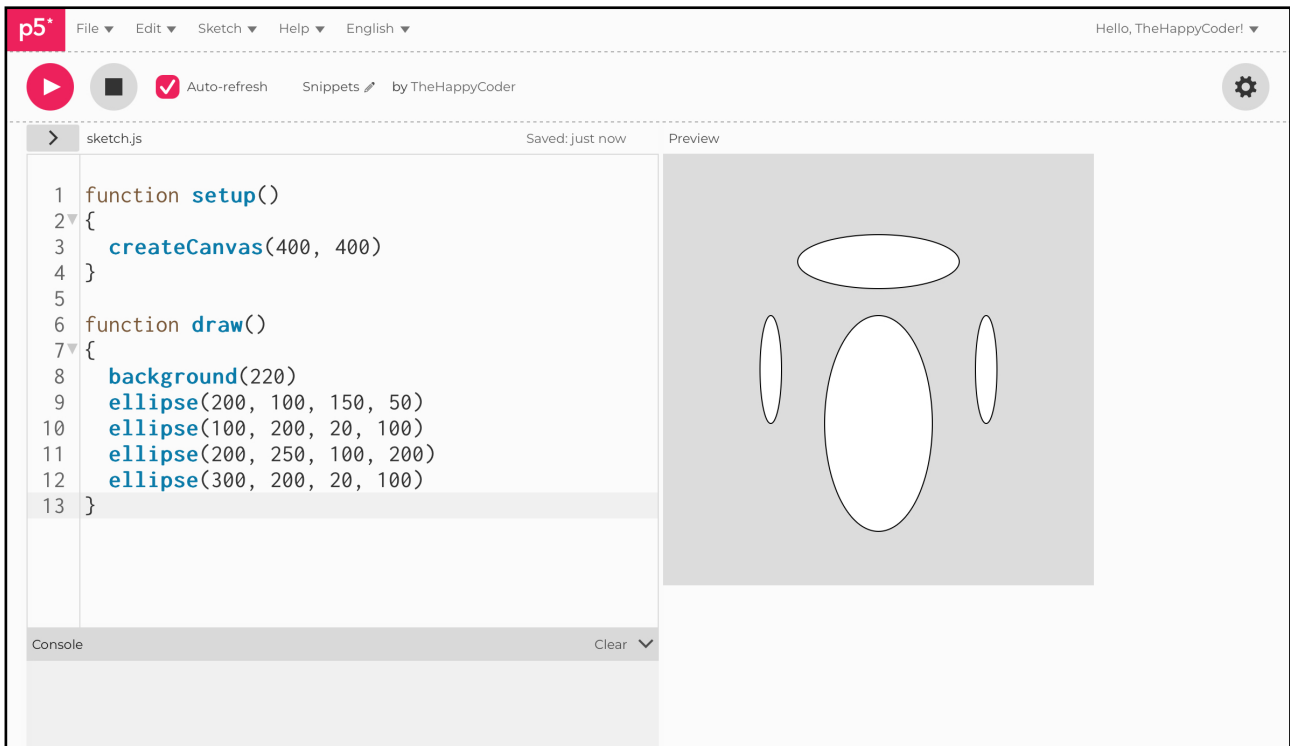


Code Explanation

```
ellipse(200, 100, 150, 50)
```

First two arguments are the x and y co-ordinates, third is the width of ellipse and the fourth is the height of the ellipse

Figure B1.2





Sketch B1.3 drawing triangles

The triangle has six arguments for the three co-ordinates at each corner.

```
function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
  triangle(200, 100, 100, 190, 300, 190)
  triangle(200, 300, 100, 210, 300, 210)
}
```



Notes

The triangle requires a bit more thought organising the co-ordinates. I often sketch them out first on a piece of paper to help plan them out. The arguments are:

- 中 x1 position
- 中 y1 position
- 中 x2 position
- 中 y2 position
- 中 x3 position
- 中 y3 position



Challenge

Make some more

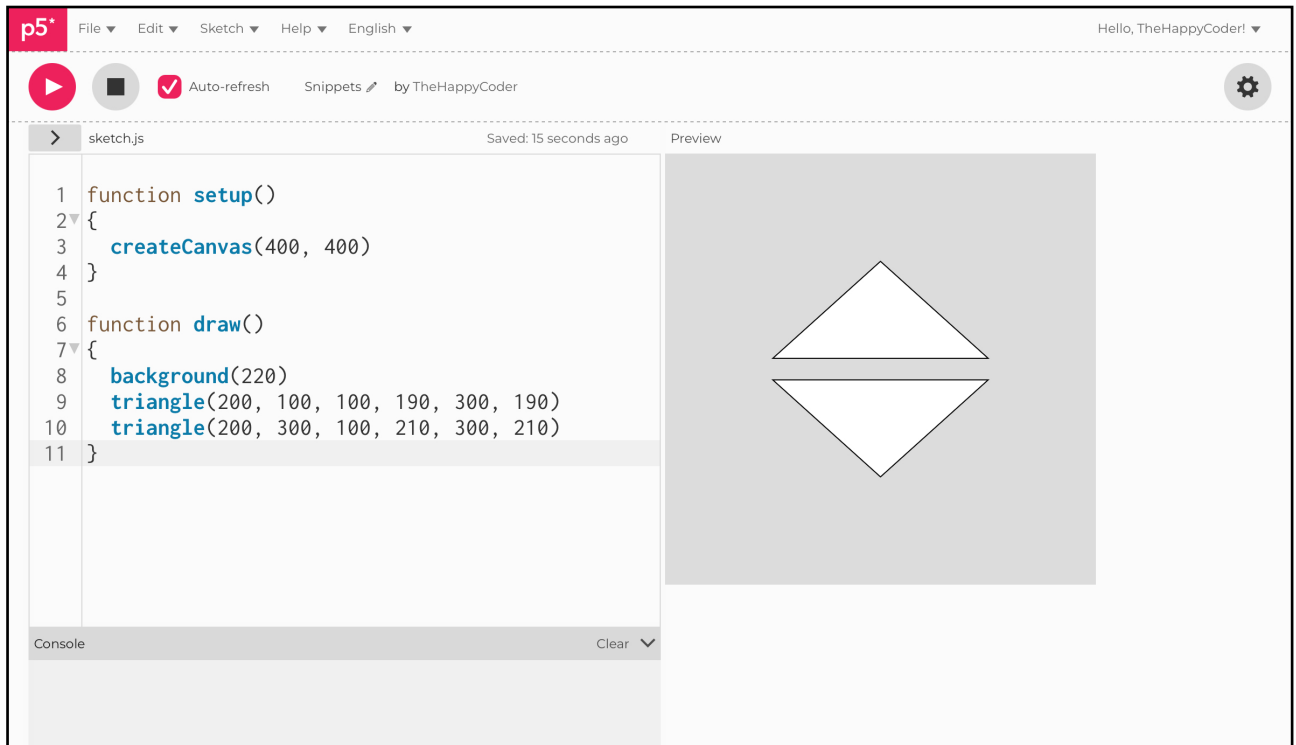


Code Explanation

```
triangle(200, 100, 100, 190, 300, 190)
```

The three co-ordinates for the three corners of the triangle

Figure B1.3





Sketch B1.4 measuring the distance

! Starting a new sketch

If we want to know the distance between two points, for instance, the centre of two circles, we can use a `dist()` function that calculates the distance between those two points. This is useful when looking at collisions or in our ml5.js example later using it to draw a circle. Here we are measuring the distance between two circles and writing the value on the canvas.

```
let x1
let y1
let x2
let y2

function setup()
{
  createCanvas(400, 400)
  textSize(32)
}

function draw()
{
  background(200)
  x1 = width/2
  y1 = height/2
  x2 = mouseX
  y2 = mouseY
  let d = dist(x1, y1, x2, y2)
  text(d, 50, 50)
  circle(x1, y1, 25)
  circle(x2, y2, 25)
}
```



Notes

Calculates the distance between two points. The version of `dist()` with four parameters calculates distance in two dimensions. The version of `dist()` with six parameters calculates distance in three dimensions; this is handy if you were using 3D shapes.



x1 position of point 1



y1 position of point 1



x2 position of point 2



y2 position of point 2

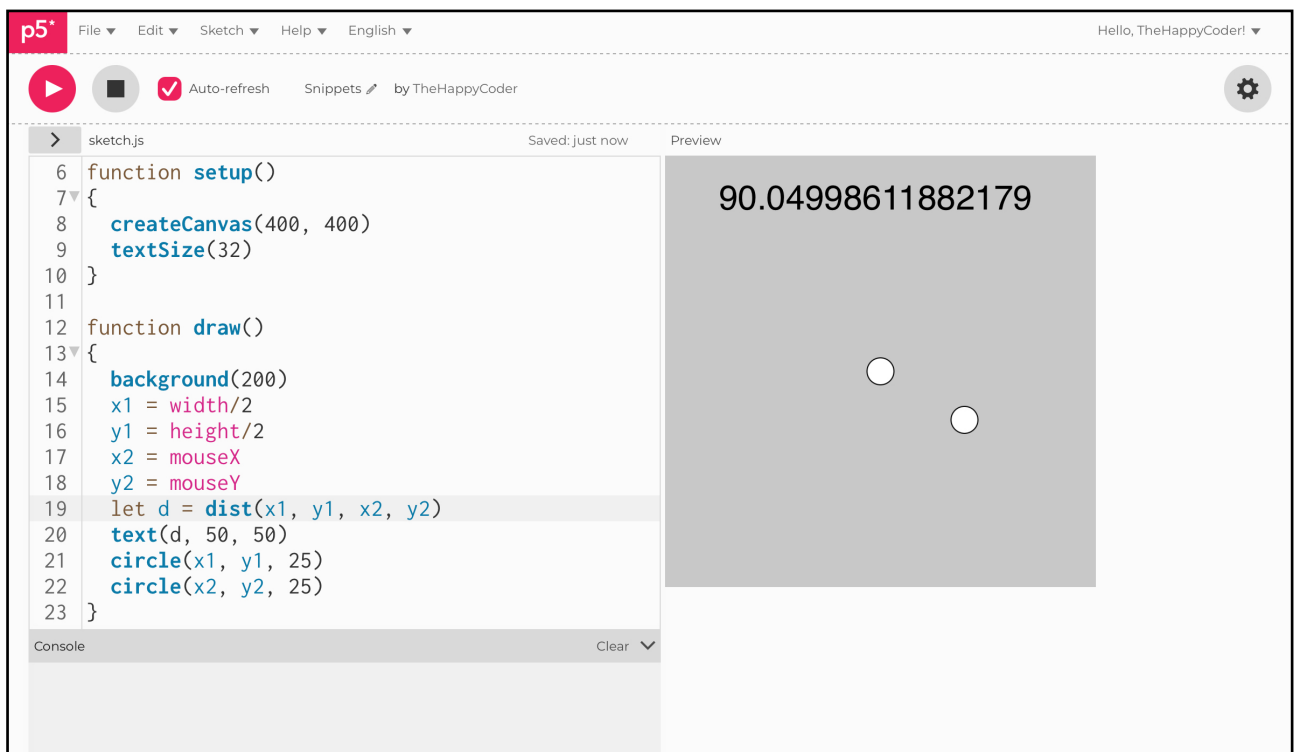


Code Explanation

```
let d = dist(x1, y1, x2, y2)
```

Calculating the distance (d) between two pairs of co-ordinates (x1, y1) and (x2, y2)

Figure B1.4





Sketch B1.5 in the name of the colour

! Starting another new sketch

We can, if we wish, just use the name of the colours. There is a wide range of colours, too numerous to mention them here. The name must be in speech marks.

```
function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background('orange')
  fill('green')
  circle(100, 100, 100)
}
```



Notes

When you put the name of the colour in speech marks, it gives you a square indicator to the colour. There are quite a few colour names. If you use 'lightgreen' with no gap, you get light green. Some colours will take the word dark, for instance, **darkred**. There are other names such as **teal**, **magenta**, and so on. You can use single or double quotes. It is useful if you just want a simple colour rather than trying to remember the **RGB** values.



Challenges

1. Just experiment.
2. Look at **Creative Coding module B unit #6 colour charts and pickers** in the resources section of www.elegantAI.org.

Code Explanation

<code>background('orange')</code>	Gives you an orange background
<code>fill('green')</code>	Fills the circle green

Figure B1.6

