

Start here!

1. This paper uses the interactive coding environment from codeguppy.com. If you never used codeguppy.com before, please first register for a free account at codeguppy.com
2. Type codeguppy.com/code.html into the chrome web browser or click the big "CODE NOW" button from the main codeguppy.com page
3. Type in the code in the grey boxes below
4. When finished click on the play button (the rectangle with the triangle in it) at the top - to stop press the square button
5. Try the challenges in ***bold italics***
6. Or just play to see what effects you can create
7. Make a scene using the shapes (e.g. a beach or a house with a garden)
8. Make interesting patterns with different shapes

Note: This guide was provided to codeguppy.com by thehappycoder. If you are an advanced codeguppy.com user ready to graduate to p5.js sketches, you can find more p5.js tutorials and materials at www.thehappycoder.org

|1| Let's draw some shapes

```
background(100)
circle(300, 300, 50)
ellipse(350, 200, 150, 50)
square(100, 200, 100)
rect(50, 50, 500, 100)
triangle(500, 200, 350, 400, 500, 400)
line(200, 550, 600, 400)
```

We can draw 6 basic shapes: Circle, Ellipse, Square, Rectangle, Triangle, Line.

Draw 5 circles in different positions and different diameters. Do the same with rectangles.

The
Happy
Coder



|2| Let's colour the background

```
background(255, 0, 0)
circle(300, 300, 100)
```

The maximum value is 255, the minimum colour is 0.

Try different combinations of those three colours.

Here are some examples...

Yellow (255, 255, 0)

Orange (255, 100, 0)

Pink (255, 0, 100)

Try mixing them yourself.

The
Happy
Coder



|3| Let's color the circle

```
background(255, 0, 0)
fill(200, 200, 0)
circle(300, 300, 100)
```

You can now fill it with whatever color you want. This will work for all the other shapes except for a line.

Draw three circles and color each one differently

The
Happy
Coder



|4| Coloring the line

```
background(255, 0, 0)
stroke(0, 0, 255)
line(200, 550, 600, 400)
circle(300, 300, 100)
```

This color the lines in the same way. Here it will give you a blue line and a blue edge to any shape.

Draw three circles with a different color border round each one.



|5| Making the lines thicker

```
background(255, 0, 0)
stroke(0, 0, 255)
strokeWeight(5)
line(200, 550, 600, 400)
circle(300, 300, 100)
```

The `strokeWeight()` gives the thickness of the line in pixels.

Try different thicknesses

The
Happy
Coder



|6| Getting rid of the lines

```
background(255, 0, 0)
noStroke()
circle(300, 300, 100)
```

Now you can get rid of the line round the shape by using `noStroke()`.

Draw two circles, have a red border round one of them and no border round the other



|7| Creating a variable

```
var x = 100
var y = 300

background(200, 100, 0)

circle(x, y, 100)
```

Here we are creating variables. var is short for variable. x is a variable and y is a variable. We can replace the co-ordinates with x and y but we need to set up the variable at the start of the code.

Change the values from 100 and 300 to other values and could you make the diameter a variable as well. You can call it anything you like.



|8| Let's make some random dots

```
var x = 0
var y = 0
var r = 20

background(200, 100, 0)

function loop()
{
  x = random(800)
  y = random(600)
  circle(x, y, r)
}
```

We have three variables `x`, `y` and `r` (for radius). This will draw the circles (dots) at random positions. The `x` position is between 0 and 800 and the `y` is between 0 and 600.

Now make it draw random dots with random radix

The
Happy
Coder



|9| Let's make the color random

```
var x = 0
var y = 0

background(200, 100, 0)

function loop()
{
  x = random(800)
  y = random(600)
  fill(random(255), random(255), random(255))
  circle(x, y, 20)
}
```

It will pick a random number between 0 and 255 for each of the three colors in `fill()`.

Add a fourth random element:

`fill(random(255), random(255), random(255), random(255))`

What does it do?

The
Happy
Coder

