Build the next Great Video Game Using the Hottest Tools

Lesson 4

First, we create a new file. In the video, I call this file `sketch4.js`. It is very important to include the `.js` extension because `.js` is a flavor of the JavaScript language.

Next, we go into our `index.html` page in order to link to our new `sketch4.js` file. The `index.html` page is the first file that a web browser looks for, therefore it is important that we link to it.

As this is not an `html` course, you are not required to understand `html`. All we are doing is linking the `index.html` page to our `.js` file.

We do this on line 14 of the `index.html` page.

```
1 <!DOCTYPE html>
2 <html>
3   <head>
4     <meta charset="UTF-8">
5     <title>sketch</title>
6     <script src="libraries/p5.js" type="text/javascript"></script>
7
8     <script src="libraries/p5.dom.js" type="text/javascript"></script>
9
10    <script src="libraries/p5.sound.js" type="text/javascript"></script>
11
12    <script src="lib/p5.play.js" type="text/javascript"></script>
13
14    <script src="sketch4.js" type="text/javascript"></script>
15
16    <style> body {padding: 0; margin: 0;} canvas {vertical-align: top;} </style>
17  </head>  <body>  </body>
```

In our `sketch4.js` file, here is the code that we start off with.

```
function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(180);
}
```

We create a new canvas, with a `width` of `800` pixels and a `height` of `600` pixels. We also set the `background` color to a medium shade of grey.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
}
```

Next, we create `2` variables. The `x` variable has the value of `50`, as does the `y` variable.

Then, using the values of the `x` and `y` variables, we create a circle with the `ellipse()` command, at `x`: `50`, `y`: `50`, with a `width` and `height` of `70` pixels each.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  x = x + 1;
}
```

Every time the `draw()` function loops, we add `1` to the `x` variable. This means that the first time the circle is drawn, `x` is `50`. The second time the circle is drawn, `x` is `51`, and so on. In this way, the circle moves across the canvas.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  x = x + 5;

}
```

We change the code so that we subtract `5` from `x`. The circle now moves in the opposite direction, going left.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  x = x - 5;

}
```

By subtracting `5` from `x`, we increase the speed at which the circle animates left. It now animates `5` times faster. Next, we'll add a `constrain`.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  x = x - 5;
  if(x < -35){
      x = width;
  }
}
```

In our `if` statement, whenever the `x` position is less than `-35`, that is, the center of the circle has animated to the left side of the canvas, we give `x` the value of `width`, sending it to the right side of the canvas.

Here again, the issue we have is that the entire circle does not animate off the canvas.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  x = x - 5;
  if(x < -35){
      x = width + 35;
  }
}
```

We change the `constrain` so that when the `x` value is `-35`, that is, the entire circle has animated off the left side of the canvas, we send it right. The circle then begins to animate fully off the right side of the canvas, with its `x` position at `width` + `35` pixels.

The circle fully animates offscreen and loops.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  x = x - 5;
  /*
  if(x < -35){
      x = width + 35;
  }
  */
}
```

We `comment out` the `constrain` and the `update`. The code is rendered inactive and the value of `x` does not change. The circle no longer animates.

Next, we will animate the circle upwards in the `y` direction.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  x = x - 5;
  /*
  if(x < -35){
      x = width + 35;
  }
  */
}
```

Each time the `draw()` function loops, we subtract `1` from `y`, which animates the circle up the canvas.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  y = y - 3;
  /*
  if(x < -35){
      x = width + 35;
  }
  */
}
```

If the value of `y` is less than `-35`, that is, the circle is at the top of the canvas, we set the value of `y` to the `height` of the canvas.

Again, we have the issue of the circle not completely animating offscreen. We'll modify the code in order to fix this problem.

```
var x = 50;
var y = 50;

function setup(){
  createCanvas(800, 600);
}

function draw(){
  background(100);
  ellipse(x, y, 70, 70);
  y = y - 3;
  if(y < -35){
      y = height + 35;
  }
  /*
  if(x < -35){
      x = width + 35;
  }
  */
}
```

Now, when the value of `y` is less than `-35`, that is, when the circle fully animates off the top of the canvas, we set the value of `y` to the `height` of the canvas `+` `35` pixels, the bottom of the canvas.

The circle fully animates off the top of the canvas back to the bottom.