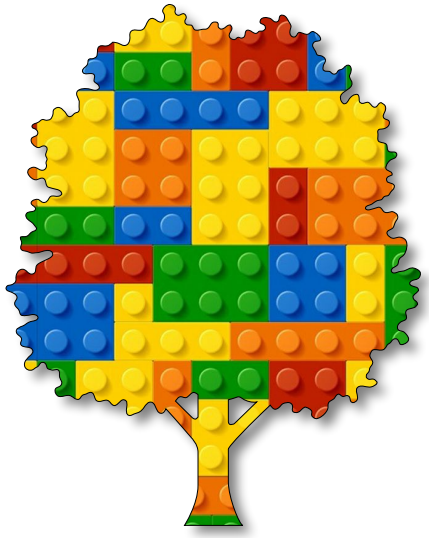
Algorithmic Art

Module A

Unit #4

lots of lines



Content

Module A Unit #4 lots of lines

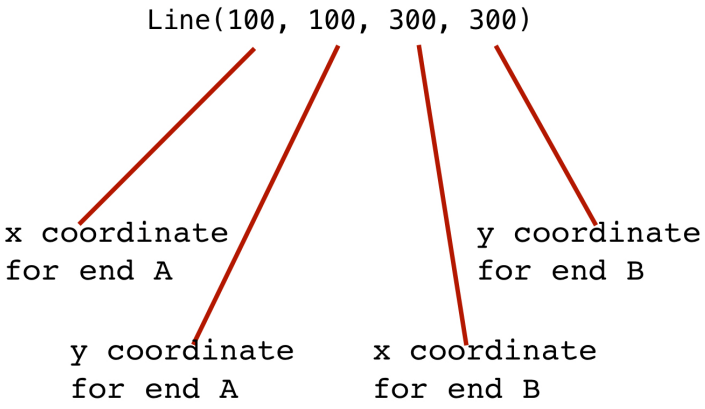
Introduction to drawing lots of lines

We have drawn circles now to introduce lines. To draw a line, we need four arguments; these are two sets of coordinates. A set for each end of the line; it is that simple. If we have an end A and an end B for the line() function, it will look like below in Fig. 1. This draws a line between the coordinates (100, 100) and (300, 300).

Key concepts:

- line() function
- if() statement
- noLoop()

Figure 1: line() function



Sketch A4.1 lines

❗ start a new sketch

We can draw lots of other shapes, but in this section, we will draw lines and create simple image using just lines. To reiterate, the line() function has four arguments: the first two are the x and y co-ordinates of one end of the line, and the other two arguments are the x and y co-ordinates of the other end of the line.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  line(100, 100, 300, 300)
}
```

🗒️ Notes

Not a very exciting creation but there is plenty we can do with it.

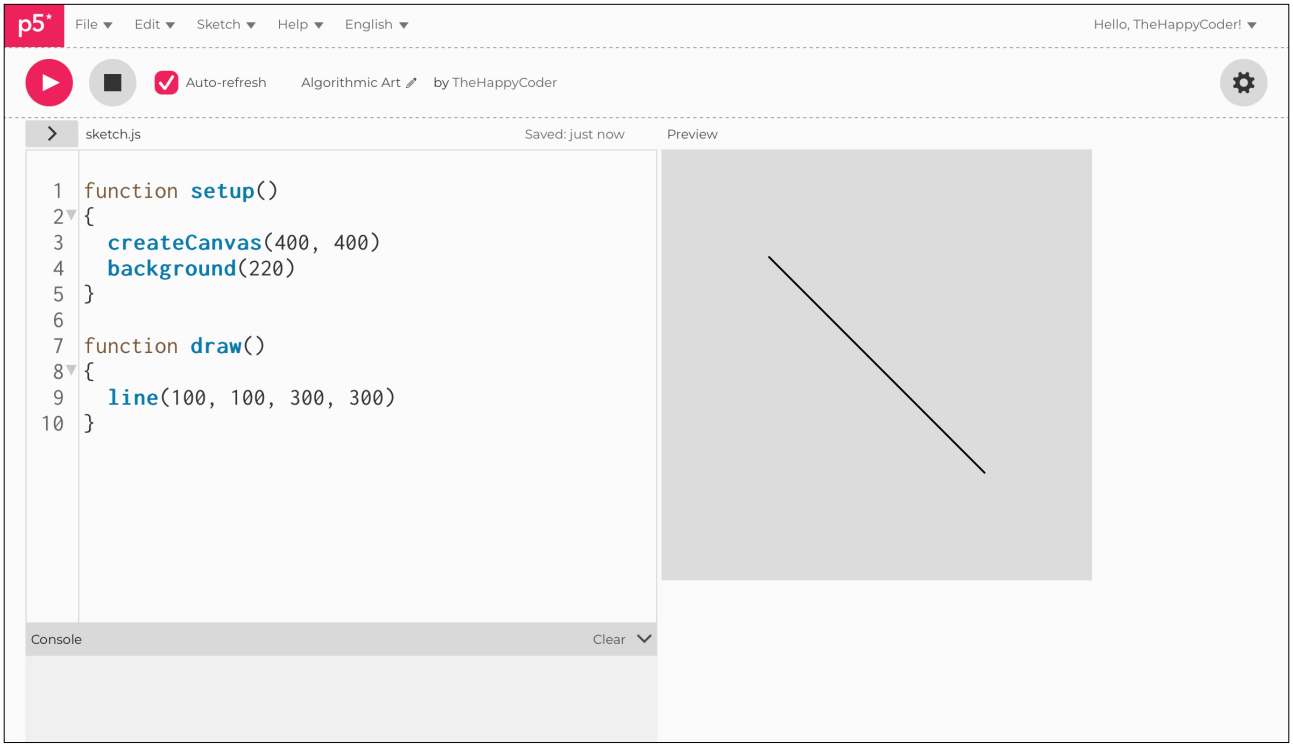
🌻 Challenge

Try other values for the coordinates

🛠️ Code Explanation

`line(100, 100, 300, 300)`	Drawing a line from the top left (100, 100) to the bottom right (300, 300)

Figure A4.1


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
1 function setup()
2 {
3 createCanvas(400, 400)
4 background(220)
5 }
6
7 function draw()
8 {
9 line(100, 100, 300, 300)
10 }
Console
Clear

Sketch A4.2 a row of lines

Here, we are going to draw lots of vertical lines 10 pixels apart. We first create a variable for the x component of the coordinates and put that in the line() function. We want the same x value for each end.

```
let x = 10

function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  line(x, 100, x, 300)
  x += 10
}
```

🗒️ Notes

To add 10 each time we have used an abbreviation (+=). We could have written it long hand like so: x = x + 10. We have cheated a little because the programme is still drawing the lines; it never stops, which is never a great idea. In the next bit, we will use a while() loop to stop it.

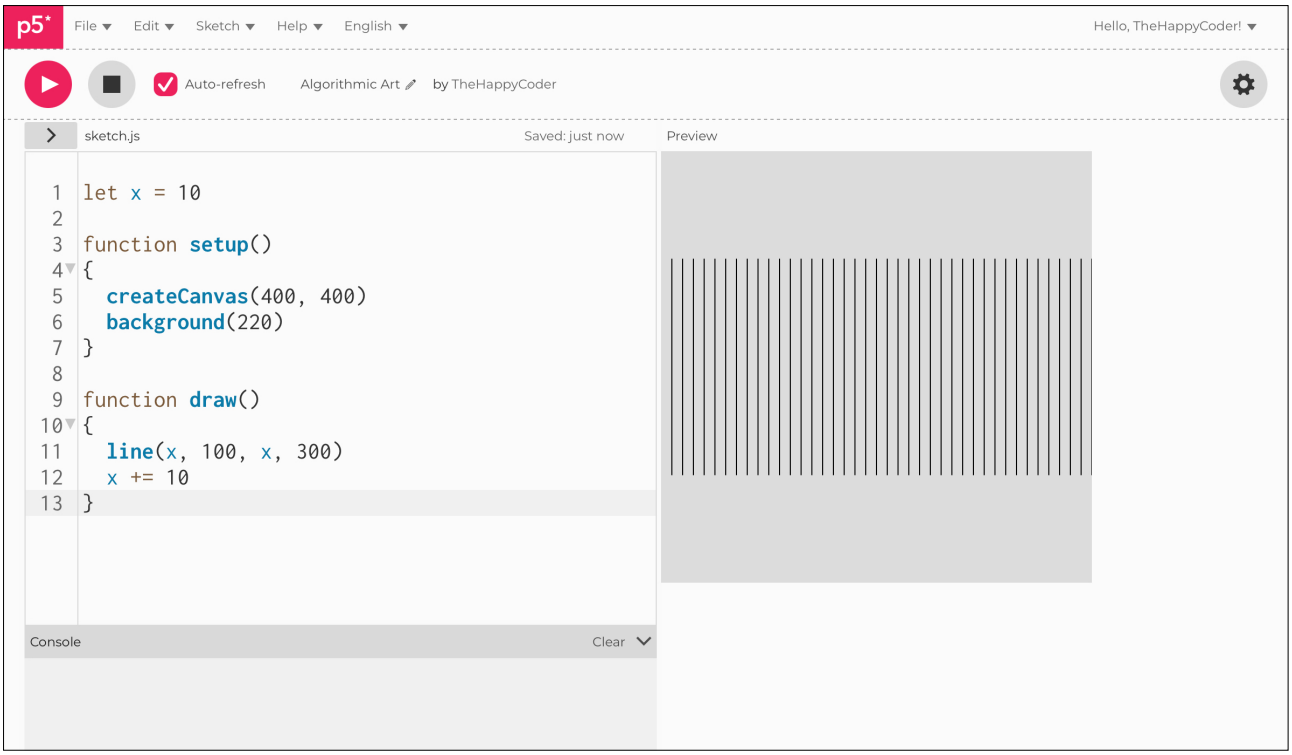
🌻 Challenge

Try a different spacing value other than 10

🛠️ Code Explanation

let x = 10	Create a variable for x and give it a starting value of 10
line(x, 100, x, 300)	Draws a line with the same x value which starts at 10 and increases by 10, the y values remain the same for each line
x += 10	Adds 10 each time. We could write x = x + 10

Figure A4.2


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 10
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
line(x, 100, x, 300)
x += 10
}
Console
Clear

Sketch A4.3 limiting lines

Here we have a `while()` loop. In this example, the lines are drawn while `x` is less than `400`. When `x` reaches `400`, then the loop stops. The loop only works while `x` is less than (`<`) `400`. To put the code inside the curly brackets, you can just cut and paste.

```
let x = 10

function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  while (x < 400)
  {
    line(x, 100, x, 300)
    x += 10
  }
}
```

🗒️ Notes

The `while()` loop uses a conditional statement (`<`). While `x` is less than `400`, the condition is `true` and it draws the line, adds `10` to `x`, and repeats. The condition is considered to be `false` when the value of `x` is `400` or above (and the function stops).

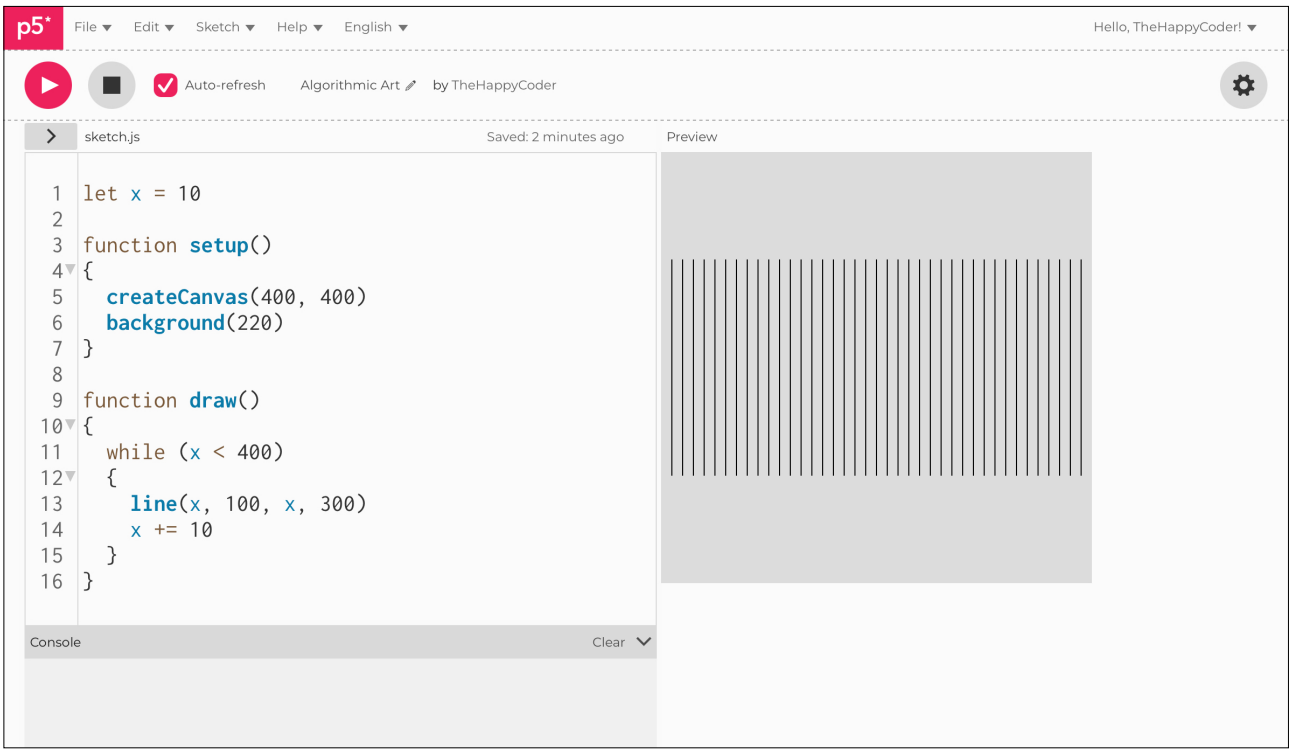
🌻 Challenges

Change the conditional value to `100`

🛠️ Code Explanation

`while (x < 400)`	It checks to see if x has exceeded 400

Figure A4.3


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: 2 minutes ago
Preview
let x = 10
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
while (x < 400)
{
line(x, 100, x, 300)
x += 10
}
}
Console
Clear

Sketch A4.4 colour and thickness

Adding some thickness and colour to the lines.

```
let x = 10

function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  strokeWeight(5)
  stroke(200, 0, 0)
  while (x < 400)
  {
    line(x, 100, x, 300)
    x += 10
  }
}
```

🗒️ Notes

We can give the line extra weight (thickness) as well as colour. This means we can manipulate the lines in many creative and interesting ways.

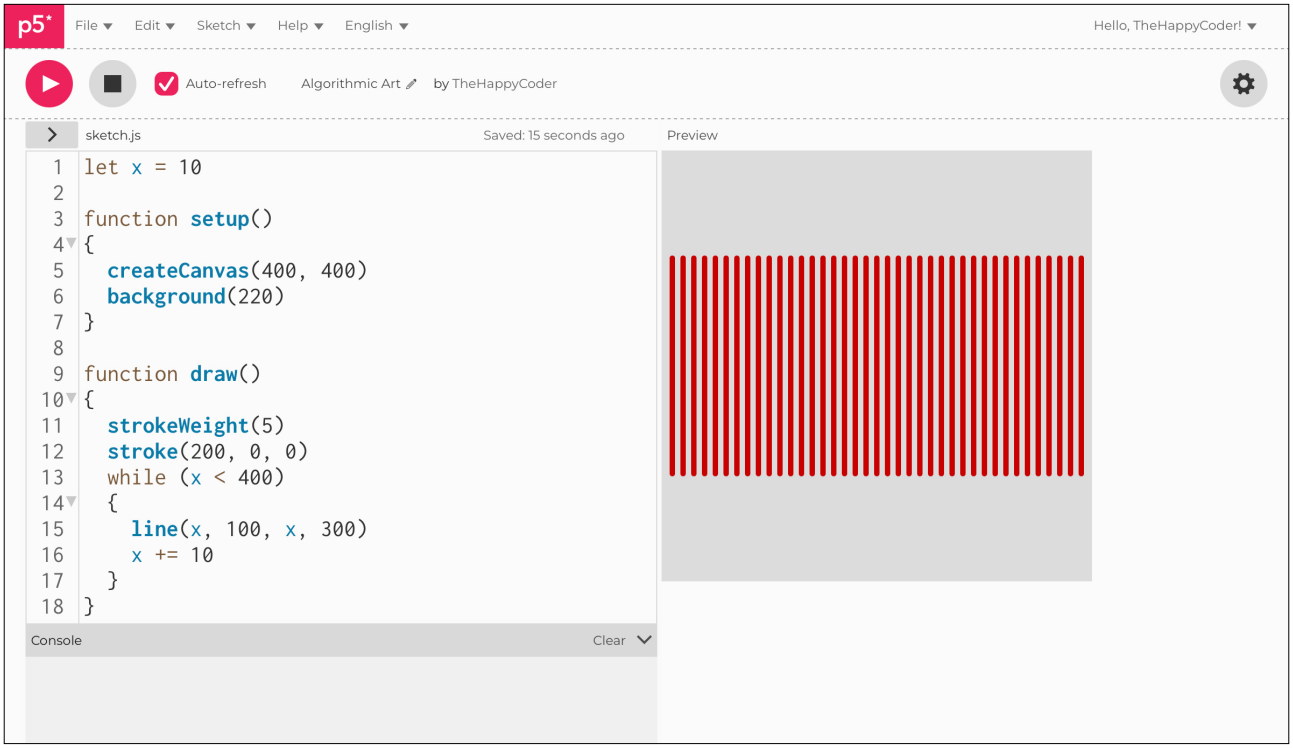
🌻 Challenges

1. Try alpha as well
2. Make it full height

🛠️ Code Explanation

`strokeWeight(5)`	Determines how thick the line is
`stroke(200, 0, 0)`	Determines the colour of the line

Figure A4.4


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: 15 seconds ago
Preview
let x = 10
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
strokeWeight(5)
stroke(200, 0, 0)
while (x < 400)
{
line(x, 100, x, 300)
x += 10
}
}
Console
Clear

Sketch A4.5 incrementing the colour

We can introduce another variable to reduce the amount of red. We will call this `increment` and subtract it from the initial value of `200` in steps of `5`.

❗ we move the `stroke()` function inside the `while()` loop

```
let x = 10
let increment = 5

function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  strokeWeight(5)
  while (x < 400)
  {
    stroke(200 - increment, 0, 0)
    line(x, 100, x, 300)
    x += 10
    increment += 5
  }
}
```

🗒️ Notes

We are decreasing the red element of the RGB by `5` on each iteration inside the `while()` loop, thus making it darker.

🌻 Challenge

1. Can you think of another way to do it (hint: use `-=` instead)
2. Try to achieve the opposite, from dark red to a light red

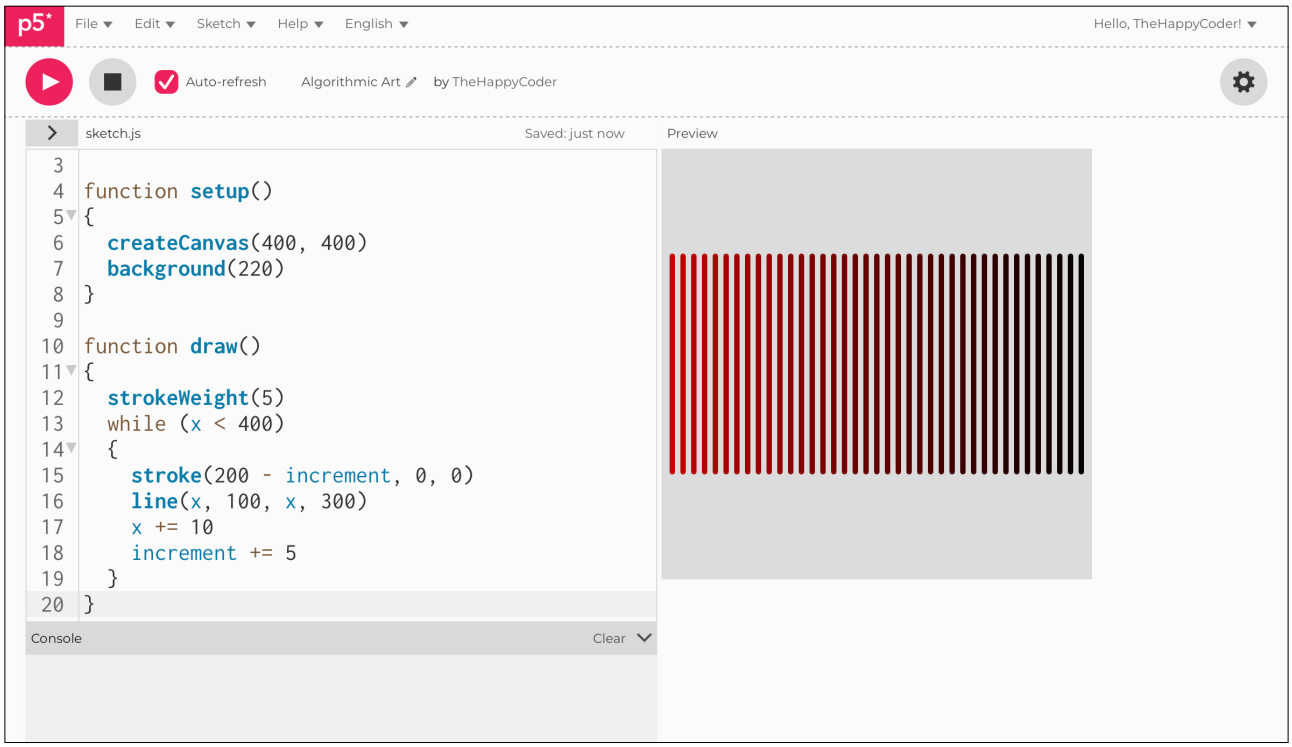
🛠️ Code Explanation

`let increment = 5`	Create a variable called increment
`stroke(200 - increment, 0, 0)`	Reduce the amount of red by that amount
`increment += 5`	Keep reducing the red each iteration of the loop

Hint solution:

```
let x = 10
let increment = 200

function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  strokeWeight(5)
  while (x < 400)
  {
    stroke(increment, 0, 0)
    line(x, 100, x, 300)
    x += 10
    increment -= 5
  }
}
```

Figure A4.5



Sketch A4.6 more blue

Let's add the increment to the amount of blue as the red value decreases.

```
let x = 10
let increment = 5

function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  strokeWeight(5)
  while (x < 400)
  {
    stroke(200 - increment, 0, increment)
    line(x, 100, x, 300)
    x += 10
    increment += 5
  }
}
```

🗒️ Notes

The power of variables

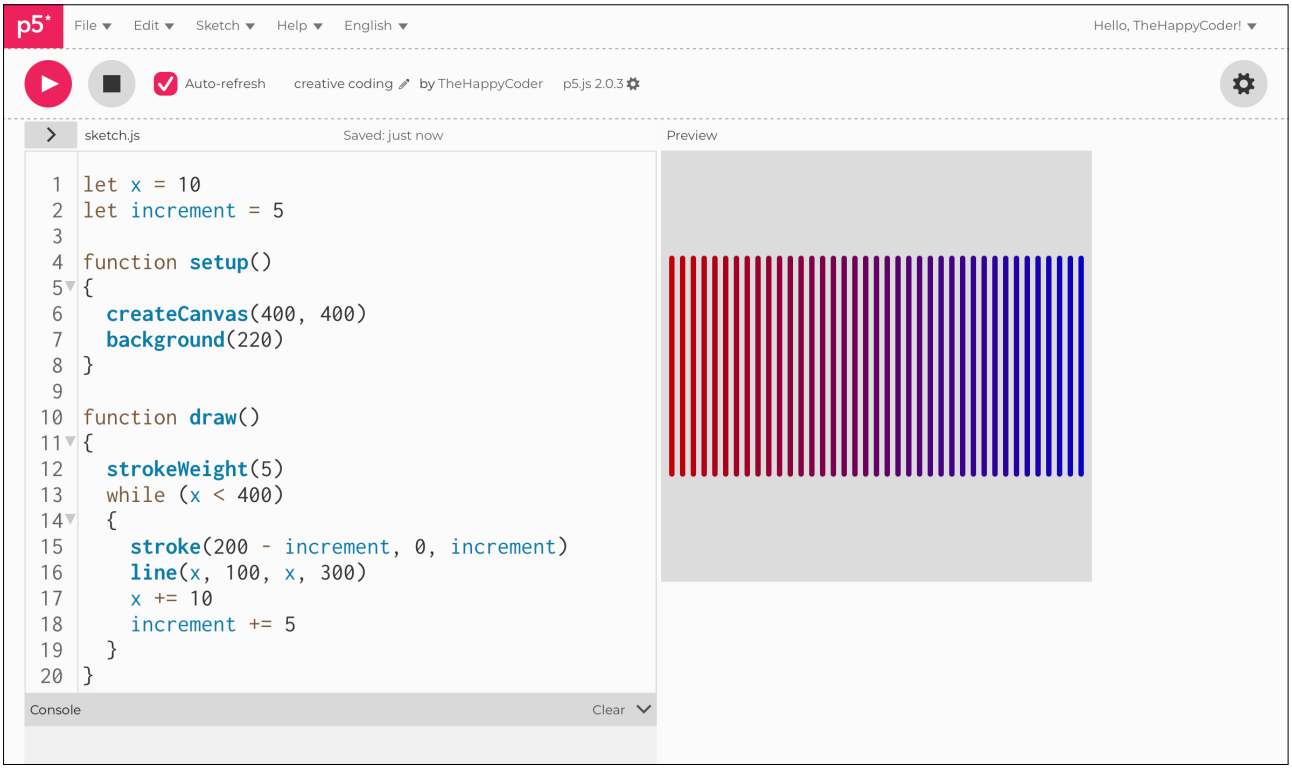
🌻 Challenges

1. Try other colours
2. Try alpha

🛠️ Code Explanation

`stroke(200 - increment, 0, increment)`	Decreases the red and increases the blue at the same time

Figure A4.6


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
creative coding
by TheHappyCoder
p5.js 2.0.3
sketch.js
Saved: just now
Preview
let x = 10
let increment = 5
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
strokeWeight(5)
while (x < 400)
{
stroke(200 - increment, 0, increment)
line(x, 100, x, 300)
x += 10
increment += 5
}
}
Console
Clear

Sketch A4.7 random colour and other stuff

We can start to play around a bit. Notice that you don't always have to create variables when using random. Where appropriate, you can put them straight into the function.

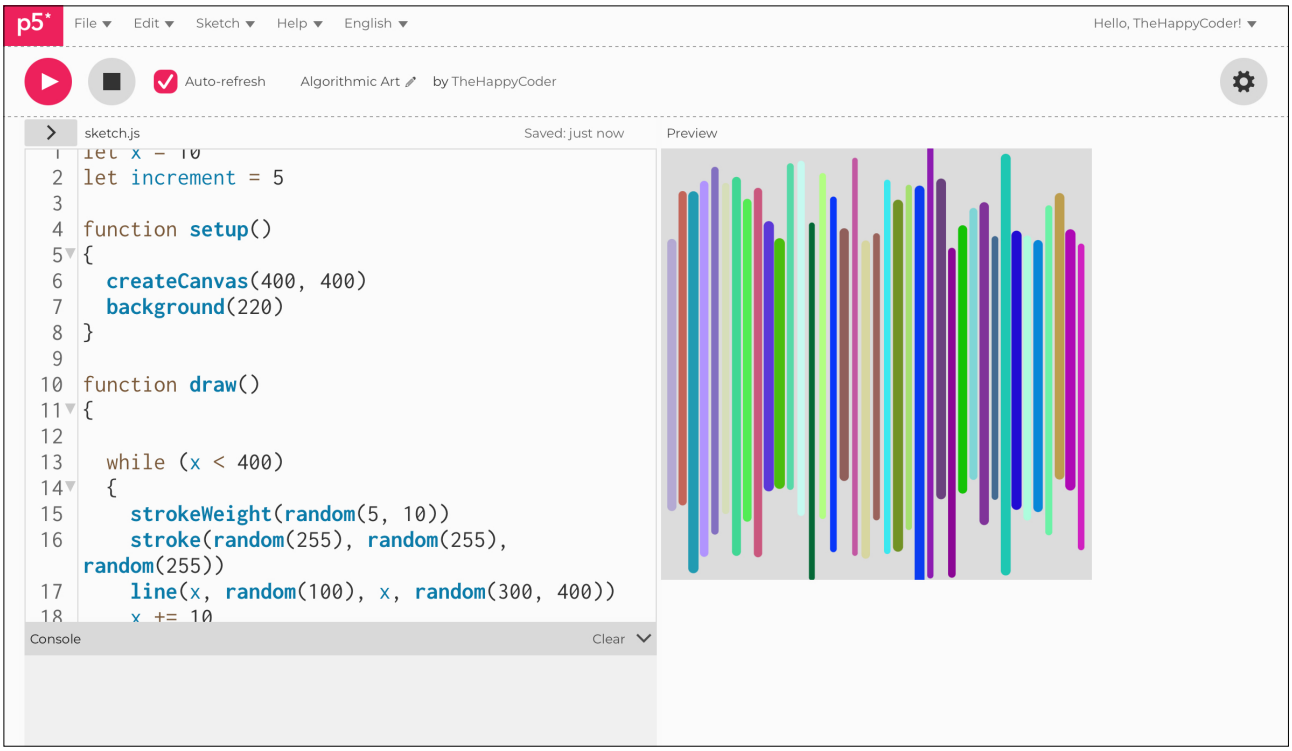
❗ Move the `strokeWeight()` into the `while()` loop and randomise it, also notice we aren't using `increment` anymore.

```
let x = 10
let increment = 5

function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  while (x < 400)
  {
    strokeWeight(random(5, 10))
    stroke(random(255), random(255), random(255))
    line(x, random(100), x, random(300, 400))
    x += 10
    increment += 5
  }
}
```

🗒️ Notes

The code should be fairly obvious to you now, creates a pleasing effect

Figure A4.7


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 10
let increment = 5
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
while (x < 400)
{
strokeWeight(random(5, 10))
stroke(random(255), random(255), random(255))
line(x, random(100), x, random(300, 400))
x += 10
Console
Clear

Sketch A4.8 newish sketch

❗ start a new sketch with additional lines highlighted.
Here we have a faint line drawn randomly from the left side to the right side.

```
function setup()
{
  createCanvas(400, 400)
  background(255)
}

function draw()
{
  strokeWeight(0.1)
  line(0, random(400), 400, random(400))
}
```

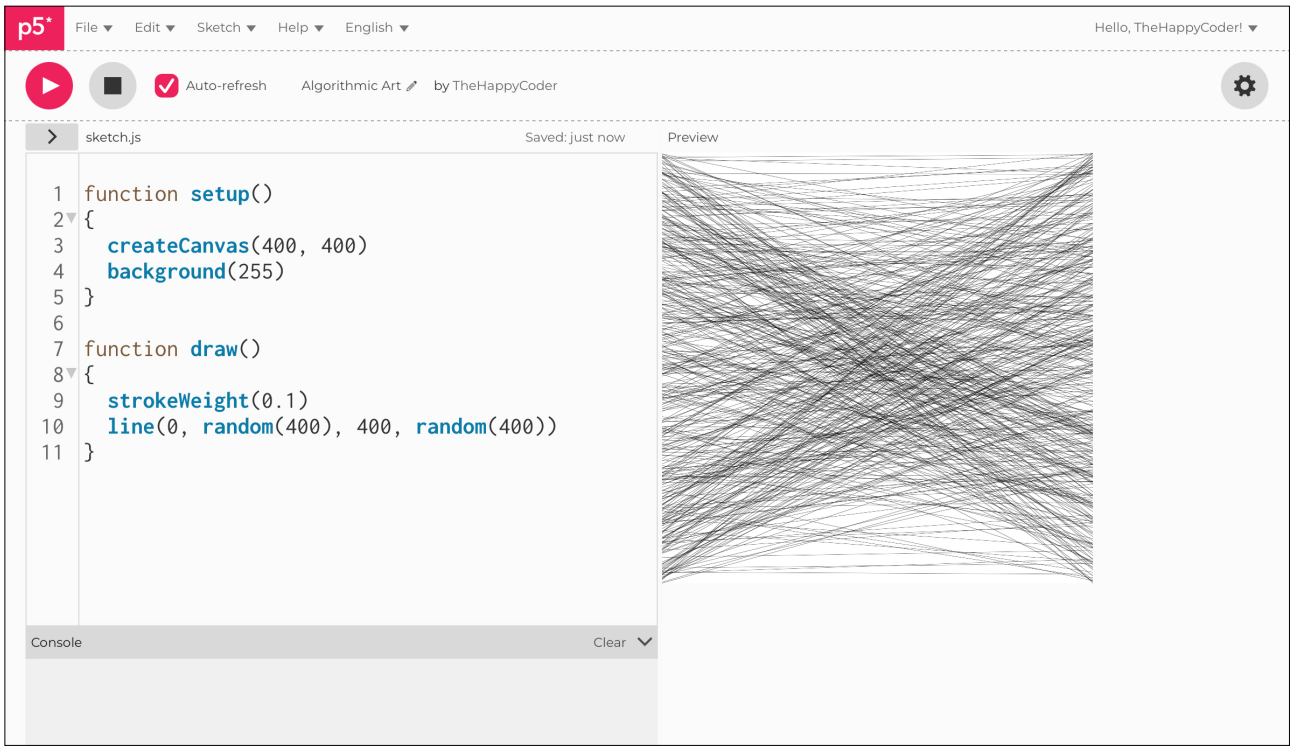
🗒️ Notes

This will continue infinitum

🛠️ Code Explanation

`strokeWeight(0.1)`	Gives a very thin weight to the line

Figure A4.8


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
1 function setup()
2 {
3 createCanvas(400, 400)
4 background(255)
5 }
6
7 function draw()
8 {
9 strokeWeight(0.1)
10 line(0, random(400), 400, random(400))
11 }
Console
Clear

Sketch A4.9 more random lines

Let's add another random line generator. This time, top to bottom.

```
function setup()
{
  createCanvas(400, 400)
  background(255)
}

function draw()
{
  strokeWeight(0.1)
  line(0, random(400), 400, random(400))
  line(random(400), 0, random(400), 400)
}
```

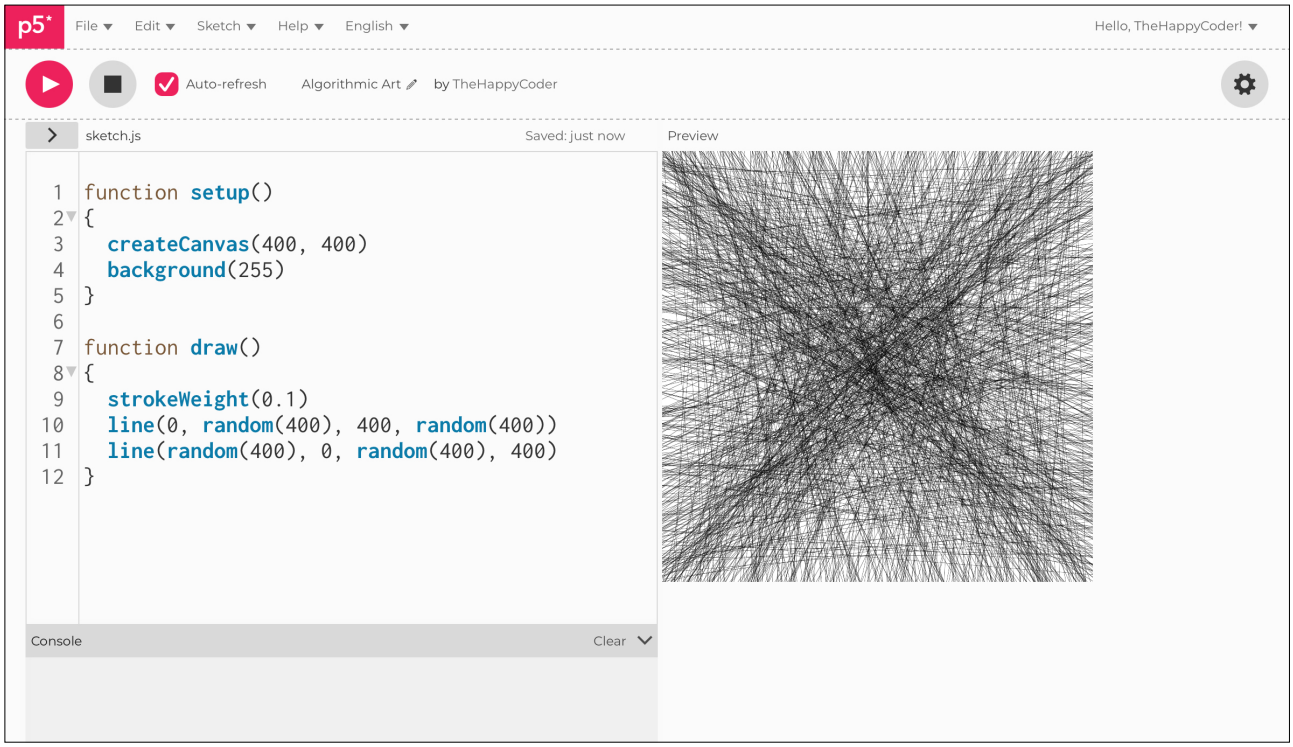
🗒️ Notes

The mesh of lines fills up the canvas rapidly. What we would like to do is stop it before it is completely blacked out.

🌻 Challenge

Can you think of a way to stop it after, say, 300 lines?

Figure A4.9


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
1 function setup()
2 {
3 createCanvas(400, 400)
4 background(255)
5 }
6
7 function draw()
8 {
9 strokeWeight(0.1)
10 line(0, random(400), 400, random(400))
11 line(random(400), 0, random(400), 400)
12 }
Console
Clear

Sketch A4.10 we need to count the lines

We can create a random mesh. First, let's create a variable called `count` to count the number of lines.

```
let count = 0

function setup()
{
  createCanvas(400, 400)
  background(255)
}

function draw()
{
  strokeWeight(0.1)
  line(0, random(400), 400, random(400))
  line(random(400), 0, random(400), 400)
  count++
}
```

Notes

This does nothing except count the lines (two lines for every `count` increase) on each iteration of the `draw()` loop.

Sketch A4.11 if() statement

Another solution to limiting the number of lines is to use something called an `if()` statement. It has a condition attached; in this case, it is the `==` sign, which means `if it is equal to`. When that condition is `true`, then it activates the code inside the curly brackets `{...}`. In this instance, we want to stop drawing any more lines, so we use a function called `noLoop()`, which is pretty obvious what it does.

```
let count = 0

function setup()
{
  createCanvas(400, 400)
  background(255)
}

function draw()
{
  strokeWeight(0.1)
  line(0, random(400), 400, random(400))
  line(random(400), 0, random(400), 400)
  count++
  if (count == 300)
  {
    noLoop()
  }
}
```

🗒️ Notes

The `if()` statement is similar to the `while()` loop; it is used more often than the `while()` loop. You can string lots of `if()` statements together and have what is called `if()...else()` statements (more on that later).

🌻 Challenges

3. Change the number of lines.
4. Change the weight and add alpha instead.
5. Have different colours.

🛠️ Code Explanation

`if (count == 300)`	Checks to see if this condition is true
`noLoop()`	If it is true then it does this

Figure A4.11