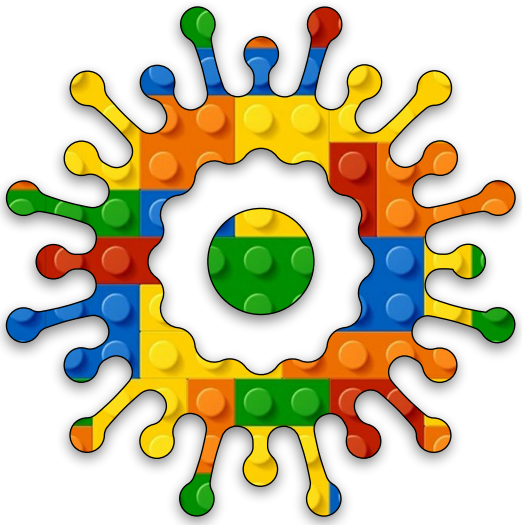
Creative Coding Module A Unit #8 10PRINT pattern



Content

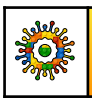
Module A Unit #8 10PRINT

Introduction to 10PRINT

It comes from an abbreviation of a single-line algorithm written in BASIC in the early 1980s: `10 PRINT CHR$(205.5+RND(1)); : GOTO 10`.

If you remember coding in BASIC, then you are possibly older than you look. This single line of code produces a random pattern, and this section uses this principle for creating random patterns.

This unit also introduces nested loops. There is such a huge scope to play with this, creating interesting designs and patterns with different shapes, colours, thicknesses, etc., so it is a great place to experiment.



Sketch A8.1 our starting sketch

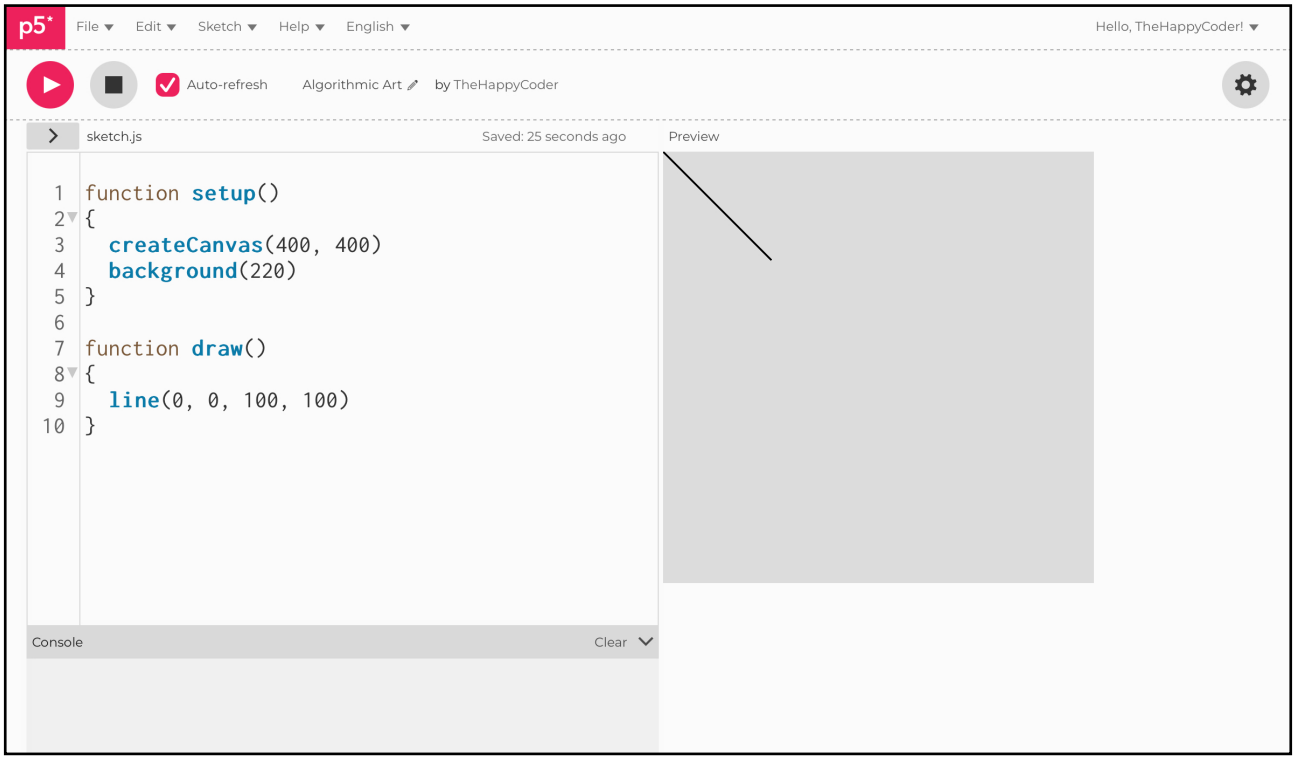
Drawing a line from (0, 0) to (100, 100).

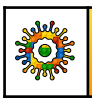
```
function setup()
{
  createCanvas(400, 400)
  background(220)
}

function draw()
{
  line(0, 0, 100, 100)
}
```

Notes

This is our starting point.

Figure A8.1


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: 25 seconds ago
Preview
1 function setup()
2 {
3 createCanvas(400, 400)
4 background(220)
5 }
6
7 function draw()
8 {
9 line(0, 0, 100, 100)
10 }
Console
Clear



Sketch A8.2 adding some variables

Adding some variables for x and y.

```
let x = 0
let y = 0

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

function draw()
{
  line(0, 0, 100, 100)
}
```

Notes

We will need them later.

Sketch A8.3 the line function

This does nothing for the line; yet, all will be revealed in due course.

```
let x = 0
let y = 0

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

function draw()
{
  line(x, y, x + 100, y + 100)
}
```

Notes

You should have the same effect.

Sketch A8.4 another variable

Instead of typing 100, we will create a variable called spacing and give that the value of 100, which will give us exactly the same result.

```
let x = 0
let y = 0
let spacing = 100

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

function draw()
{
  line(x, y, x + spacing, y + spacing)
}
```

Notes

We now have easy control of the main variables.

Sketch A8.5 using the variable

Let's now change the value of the `spacing` variable to `10` and give the canvas a nice yellow background. Also, make the line dark red.

```
let x = 0
let y = 0
let spacing = 10

function setup()
{
  createCanvas(400, 400)
  background(255, 255, 0)
  stroke(100, 0, 0)
}

function draw()
{
  line(x, y, x + spacing, y + spacing)
}
```

🗒️ Notes

Not a lot to show yet.

🌻 Challenge

Use different colours or line thickness.

Figure A8.5


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 0
let y = 0
let spacing = 10
function setup()
{
createCanvas(400, 400)
background(255, 255, 0)
stroke(100, 0, 0)
}
function draw()
{
line(x, y, x + spacing, y + spacing)
}
Console
Clear

Sketch A8.6 adding the spacing

To draw evenly spaced objects in a row, we'll loop through them. You could use a fixed value like 10, but using a variable allows you to change it once at the beginning, and it will automatically apply to all instances of that variable.

```
let x = 0
let y = 0
let spacing = 10

function setup()
{
  createCanvas(400, 400)
  background(255, 255, 0)
  stroke(100, 0, 0)
}

function draw()
{
  line(x, y, x + spacing, y + spacing)
  x += spacing
}
```

Notes

This gives us a nice row of lines.

Figure A8.6


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 0
let y = 0
let spacing = 10
function setup()
{
createCanvas(400, 400)
background(255, 255, 0)
stroke(100, 0, 0)
}
function draw()
{
line(x, y, x + spacing, y + spacing)
x += spacing
}
Console
Clear

Sketch A8.7 stop at the edge

However, the lines are being drawn beyond the canvas continually. What I would like to do is to start a new line when it gets to the right-hand edge. To do this, we use an `if()` statement. An `if()` statement checks to see if something has happened, and if it has, we tell it to do something different.

```
let x = 0
let y = 0
let spacing = 10

function setup()
{
  createCanvas(400, 400)
  background(255, 255, 0)
  stroke(100, 0, 0)
}

function draw()
{
  line(x, y, x + spacing, y + spacing)
  x += spacing
  if (x >= width)
  {
    x = 0
  }
}
```

Notes

The symbol `>=` means greater than or equal to. So if the value of `x` is greater than or equal to the width, it goes back to zero. So what you are

seeing is the red line starting again from the left-hand edge every time it reaches the right-hand edge. You can't see it happen because it is writing it over and over again on the same line in a continuous loop.

🌻 Challenge

You can check to see if it works by putting a smaller number instead of width.

🛠️ Code Explanation

`if (x >= width)`	Checks to see if x is greater then or equal to the width

Sketch A8.8 let's go down

But what I would like is to start a new line after it gets to the edge, and so we need to nudge it down by the same amount. To do this, we add a `spacing` value to `y`.

```
let x = 0
let y = 0
let spacing = 10

function setup()
{
  createCanvas(400, 400)
  background(255, 255, 0)
  stroke(100, 0, 0)
}

function draw()
{
  line(x, y, x + spacing, y + spacing)
  x += spacing
  if (x >= width)
  {
    x = 0
    y += spacing
  }
}
```

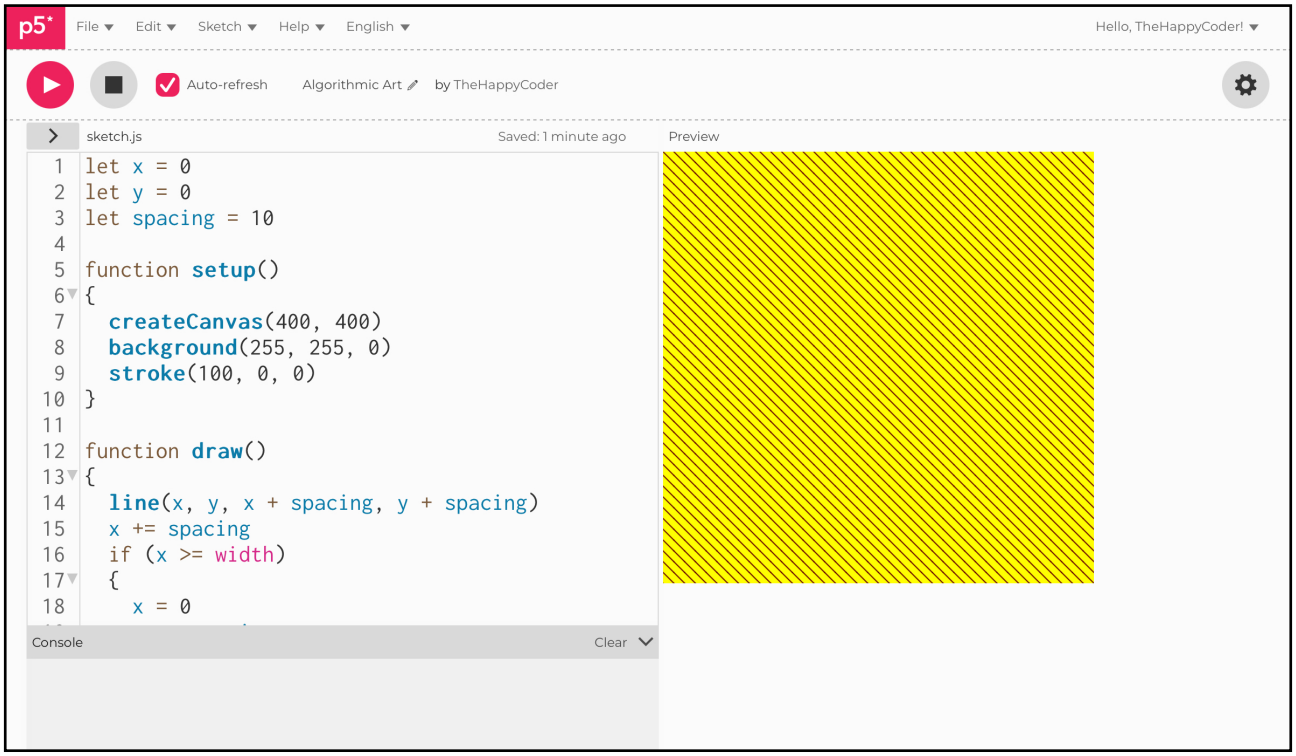
Notes

Every time we get to the edge, it starts a new line; however, it doesn't stop when it gets to the bottom. It is not critical, but neater if it did stop.

🛠️ Code Explanation

`y += spacing`	Adds the value of spacing each time, accumulating as it goes

Figure A8.8


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: 1 minute ago
Preview
let x = 0
let y = 0
let spacing = 10
function setup()
{
createCanvas(400, 400)
background(255, 255, 0)
stroke(100, 0, 0)
}
function draw()
{
line(x, y, x + spacing, y + spacing)
x += spacing
if (x >= width)
{
x = 0
Console
Clear

Sketch A8.9 stop when we get to the bottom

We use `noLoop()` when we get to the bottom; it stops the `draw()` function from looping round. You should get the same result as before.

```
let x = 0
let y = 0
let spacing = 10

function setup()
{
  createCanvas(400, 400)
  background(255, 255, 0)
  stroke(100, 0, 0)
}

function draw()
{
  line(x, y, x + spacing, y + spacing)
  x += spacing
  if (x >= width)
  {
    x = 0
    y += spacing
  }
  if (y >= height)
  {
    noLoop()
  }
}
```

Notes

Nothing new to see yet.

🛠️ Code Explanation

`if (y >= height)`	Checks to see if the y value has reached the bottom of the canvas
`noLoop()`	Instructs any loop to stop

Sketch A8.10 sloping the other way

Let's change it so that it draws the line sloping the other way. We can use nearly the same code.

```
let x = 0
let y = 0
let spacing = 10

function setup()
{
  createCanvas(400, 400)
  background(255, 255, 0)
  stroke(100, 0, 0)
}

function draw()
{
  line(x + spacing, y, x, y + spacing)
  x += spacing
  if (x >= width)
  {
    x = 0
    y += spacing
  }
  if (y >= height)
  {
    noLoop()
  }
}
```

🗒️ Notes

Now we have some changes, not a lot but some; the lines are sloping the opposite way. We just reorganised the line function. The logic is there, just follow it through.

Figure A8.10


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: 1 minute ago
Preview
8 background(255, 255, 0)
9 stroke(100, 0, 0)
10 }
11
12 function draw()
13 {
14 line(x + spacing, y, x, y + spacing)
15 x += spacing
16 if (x >= width)
17 {
18 x = 0
19 y += spacing
20 }
21 if (y >= height)
22 {
23 noLoop()
24 }
25 }
Console
Clear

Sketch A8.11 the other line 50%

We want to draw one line sloping one way 50% of the time and then sloping the other way 50% of the time in a random order. To achieve that, we can simply pick a number at random from 0 to 1, and if it is less than 0.5, do one of them; else, do the other.

```
let x = 0
let y = 0
let spacing = 10

function setup()
{
  createCanvas(400, 400)
  background(255, 255, 0)
  stroke(100, 0, 0)
}

function draw()
{
  if (random(1) < 0.5)
  {
    line(x, y, x + spacing, y + spacing)
  }
  else
  {
    line(x + spacing, y, x, y + spacing)
  }
  x += spacing
  if (x >= width)
  {
    x = 0
    y += spacing
```

```
  }
  if (y >= height)
  {
    noLoop()
  }
}
```

🗒️ Notes

We have an `if()...else` statement. You could use two `if()` statements, but this is a bit more elegant.

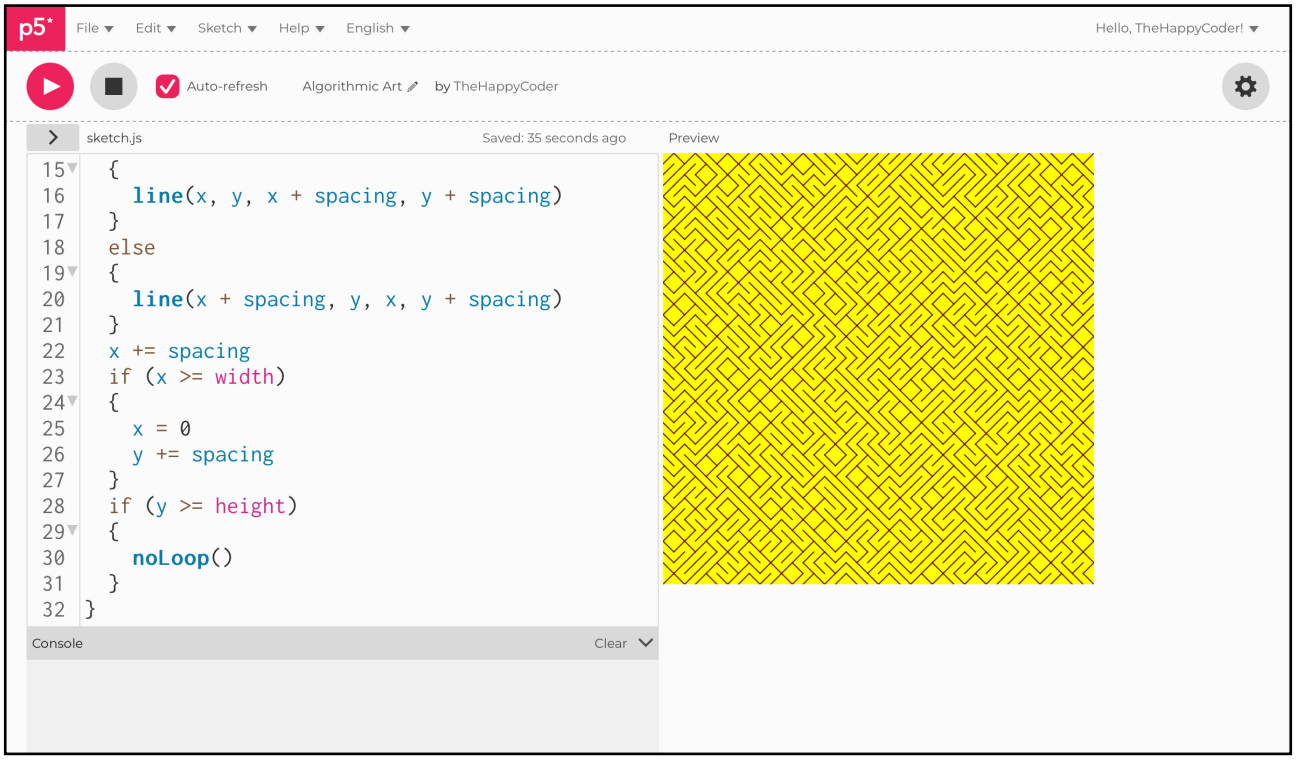
🌻 Challenge

1. What happens when you change the percentage, e.g. `0.5` to `0.8`?
2. Use other colours (random colours).
3. Thicker `strokeWeight()`.
4.

🛠️ Code Explanation

`if (random(1) < 0.5)`	If the random element is less than 0.5, do this, or else do that
`else`	A useful addition to the if() statement

Figure A8.11


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: 35 seconds ago
Preview
{
line(x, y, x + spacing, y + spacing)
}
else
{
line(x + spacing, y, x, y + spacing)
}
x += spacing
if (x >= width)
{
x = 0
y += spacing
}
if (y >= height)
{
noLoop()
}
}
Console
Clear

Sketch A8.12 variations on a theme

❗ Start a completely newish sketch.

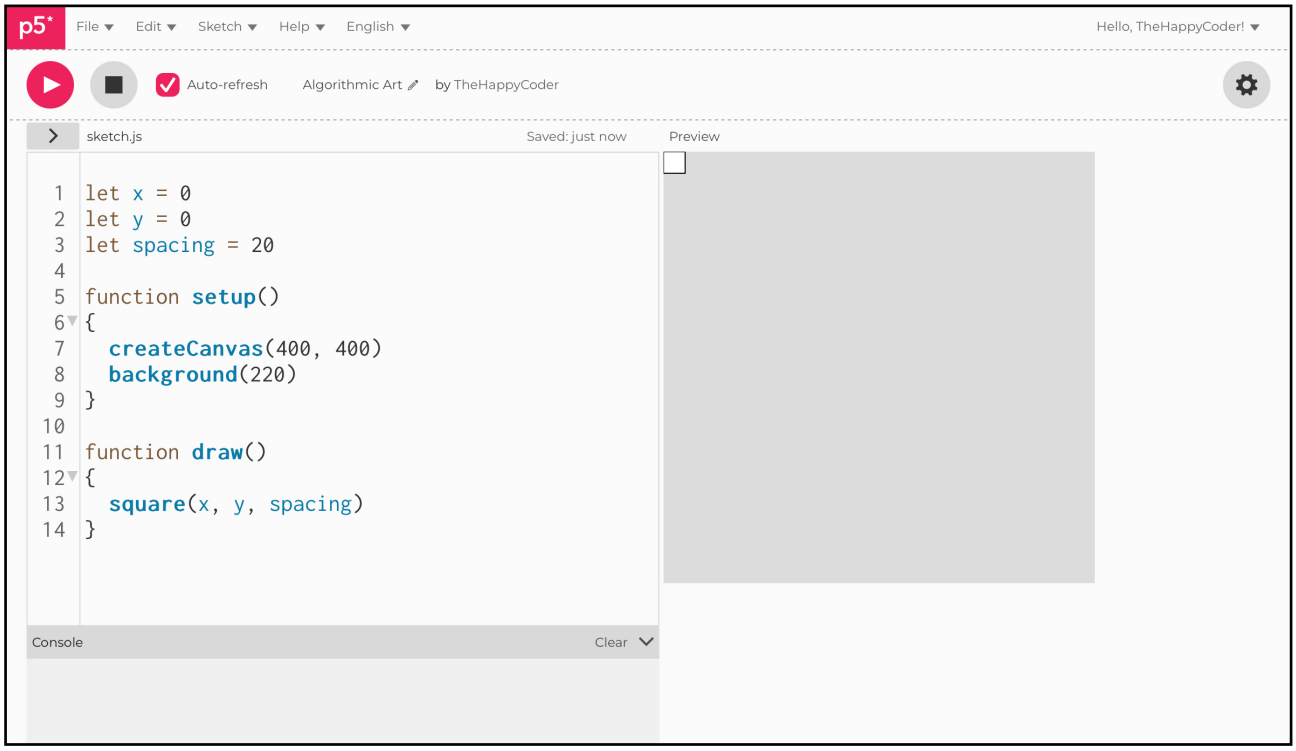
Instead of a line, we are going to draw a `square`. We will draw it at the top corner. The `square` takes three arguments: the first two are the `x` and `y` co-ordinates, and the third is the length of the side (`20`). We will use the variable `spacing` as it will come in handy.

```
let x = 0
let y = 0
let spacing = 20

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

function draw()
{
  square(x, y, spacing)
}
```

🗒️ Notes

We get a single, solitary square.

Figure A8.12


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 0
let y = 0
let spacing = 20
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
square(x, y, spacing)
}
Console
Clear

Sketch A8.13 A row of squares

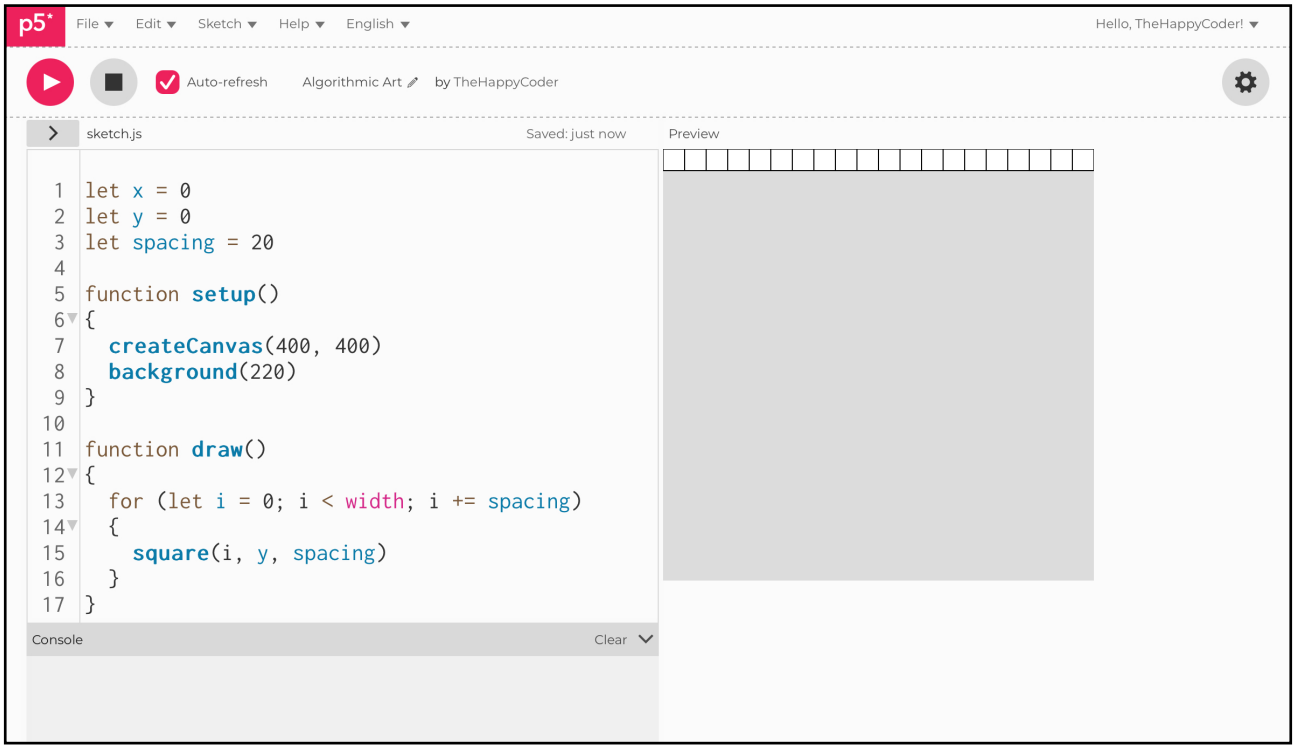
We are going to draw a row of them using a `for()` loop. A quick reminder, the `for()` loop has three parts to it. The first part (`let i = 0;`) creates a variable called `i` and gives it a value of `0`. The second part tells the loop when to stop, in this case when `i` gets to `width` (`i < width;`). The third part is how many steps (jumps) it takes (`i += spacing`). We set the `spacing` to `20`, so we are jumping in `20`s from `0` to `400` (the `width`).

```
let x = 0
let y = 0
let spacing = 20

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

function draw()
{
  for (let i = 0; i < width; i += spacing)
  {
    square(i, y, spacing)
  }
}
```

Notes

Not solitary anymore. We could've used `x` instead of `i`.

Figure A8.13


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 0
let y = 0
let spacing = 20
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
for (let i = 0; i < width; i += spacing)
{
square(i, y, spacing)
}
}
Console
Clear

Sketch A8.14 rows and columns

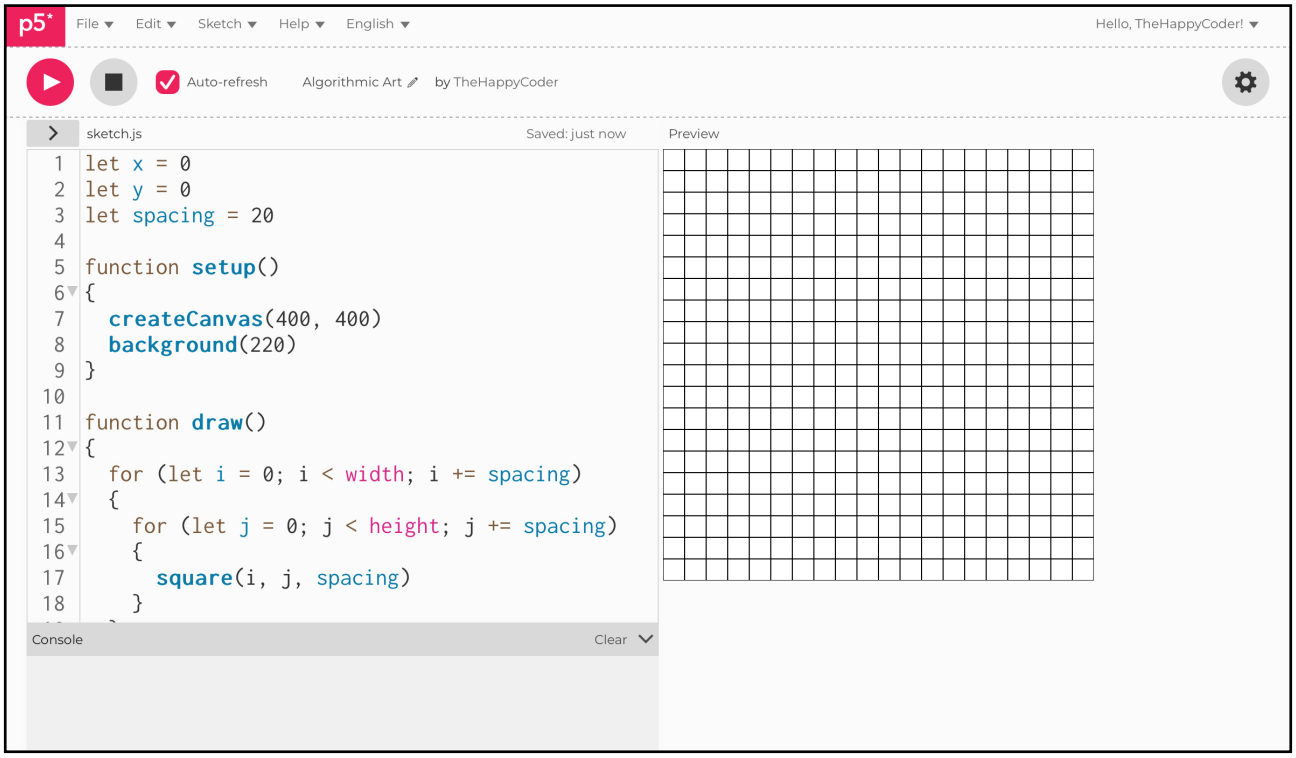
Adding columns as well as rows to fill the canvas, here we have a nested loop for the `y` value.

```
let x = 0
let y = 0
let spacing = 20

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

function draw()
{
  for (let i = 0; i < width; i += spacing)
  {
    for (let j = 0; j < height; j += spacing)
    {
      square(i, j, spacing)
    }
  }
}
```

Notes

A canvas full of squares. We don't need a `noLoop()` because it stops drawing them at the end of the loops. This is also where variables become very useful. The `i` and `j` are just convention; we could've used `x` and `y` as in a previous example. If you need a third variable, we could use `k` (or `z`) for a third loop.

Figure A8.14


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 0
let y = 0
let spacing = 20
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
for (let i = 0; i < width; i += spacing)
{
for (let j = 0; j < height; j += spacing)
{
square(i, j, spacing)
}
Console
Clear

Sketch A8.15 random size squares

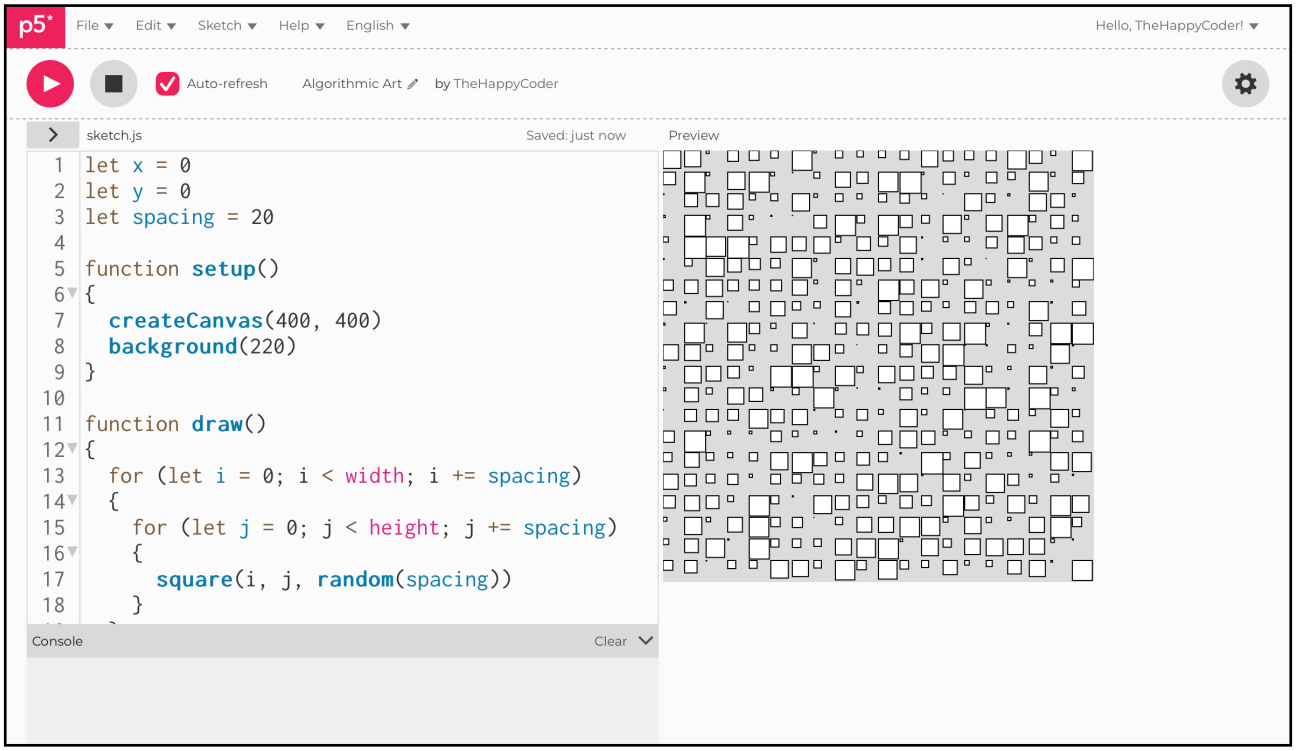
We can have some more fun with random, instead of having the same size squares. What if we change the size every time we draw one? We will need a `noLoop()` function now to stop it waggling all the time.

```
let x = 0
let y = 0
let spacing = 20

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

function draw()
{
  for (let i = 0; i < width; i += spacing)
  {
    for (let j = 0; j < height; j += spacing)
    {
      square(i, j, random(spacing))
    }
  }
  noLoop()
}
```

Notes

Nice, although it still doesn't look quite right.

Figure A8.15


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 0
let y = 0
let spacing = 20
function setup()
{
createCanvas(400, 400)
background(220)
}
function draw()
{
for (let i = 0; i < width; i += spacing)
{
for (let j = 0; j < height; j += spacing)
{
square(i, j, random(spacing))
}
Console
Clear

Sketch A8.16 using rectMode(CENTER)

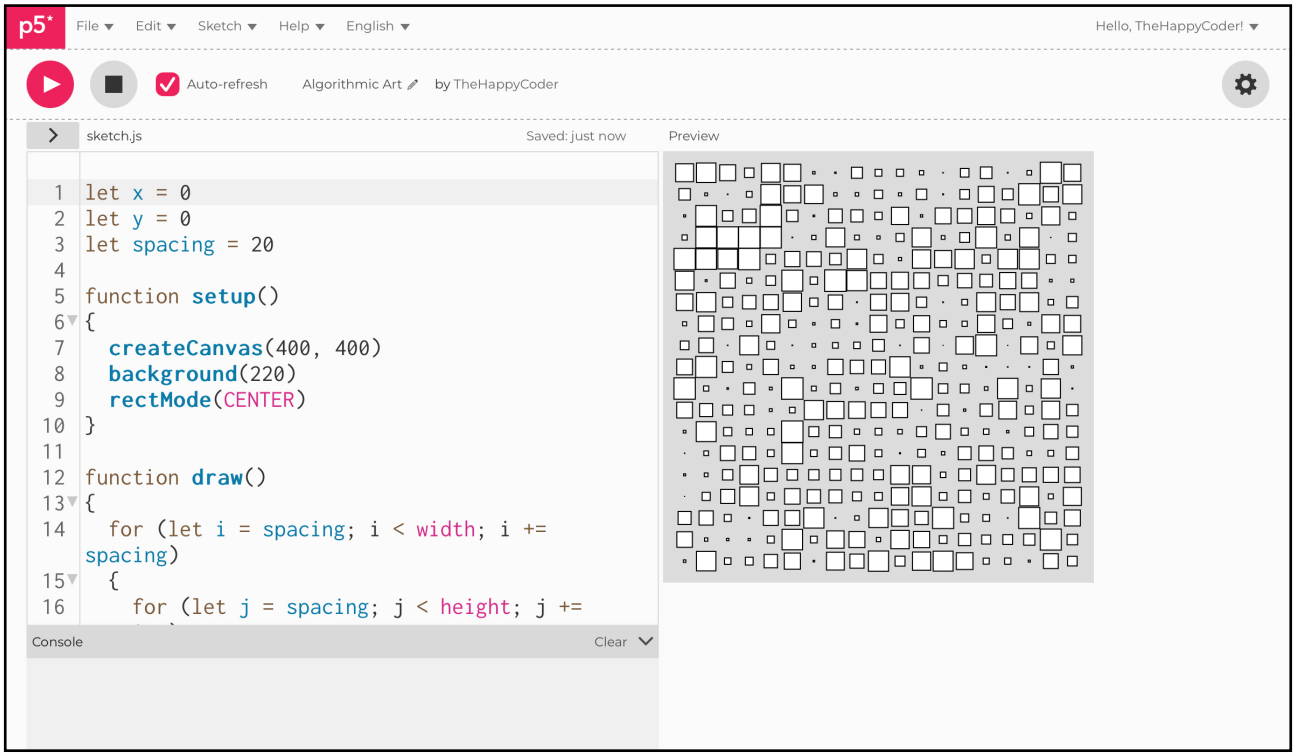
A little bit of refactoring. At the moment, the coordinates of the `square` are at the top-left corner of the `square`. To change that so that the co-ordinates are in the centre of the `square`, we use a function called `rectMode(CENTER)`. Also, to give it a border, we start with `spacing` for `i` and `j` rather than `0`.

```
let x = 0
let y = 0
let spacing = 20

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

function draw()
{
  for (let i = spacing; i < width; i += spacing)
  {
    for (let j = spacing; j < height; j += spacing)
    {
      square(i, j, random(spacing))
    }
  }
  noLoop()
}
```

Notes

Looks a lot better, but we can improve more.

Figure A8.16


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let x = 0
let y = 0
let spacing = 20
function setup()
{
createCanvas(400, 400)
background(220)
rectMode(CENTER)
}
function draw()
{
for (let i = spacing; i < width; i +=
spacing)
{
for (let j = spacing; j < height; j +=
Console
Clear

Sketch A8.17 random colours

A bit more fun will be to colour them each randomly. We will also make the background white to show up the colours, adding a small gap between `random (spacing - 2)`.

```
let x = 0
let y = 0
let spacing = 20

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

function draw()
{
  for (let i = spacing; i < width; i += spacing)
  {
    for (let j = spacing; j < height; j += spacing)
    {
      fill(random(255), random(255), random(255))
      square(i, j, random(spacing - 2))
    }
  }
  noLoop()
}
```

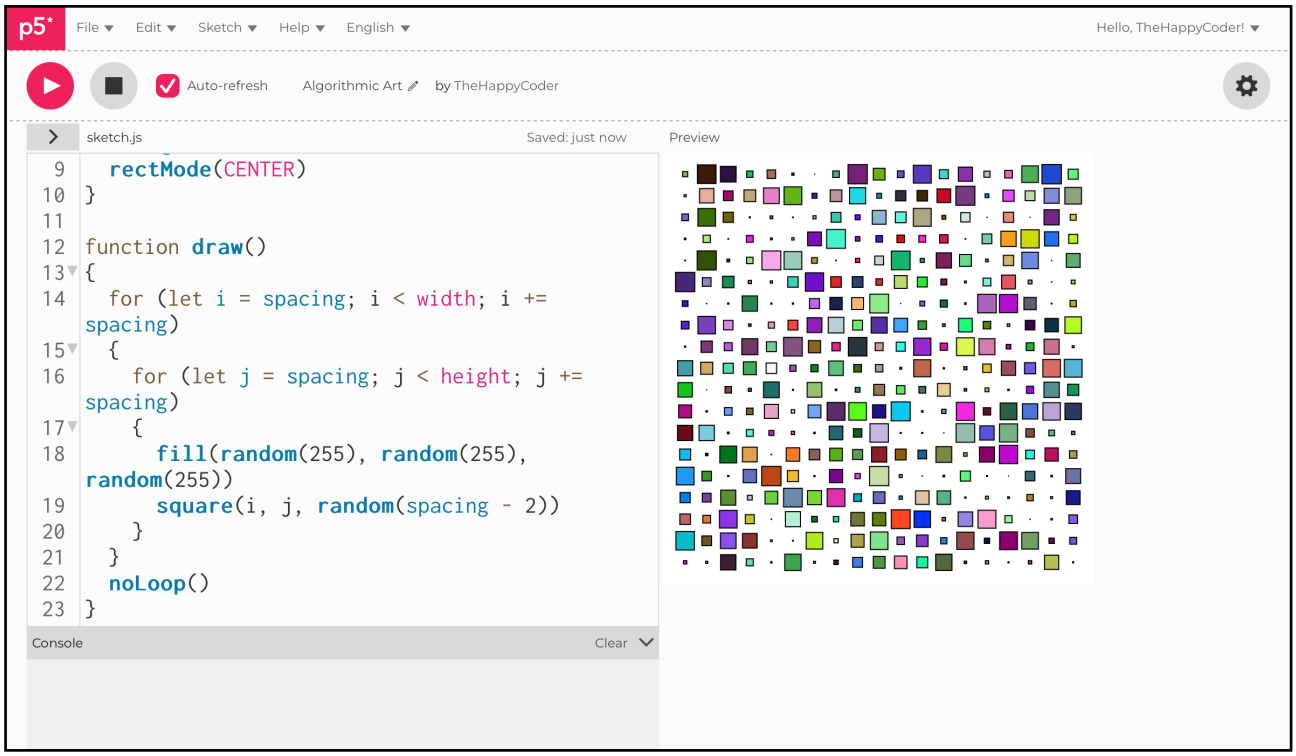
🗒️ Notes

Looking good.

🌻 Challenge

Try some other shapes.

Figure A8.17


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
rectMode(CENTER)
}
function draw()
{
for (let i = spacing; i < width; i += spacing)
{
for (let j = spacing; j < height; j += spacing)
{
fill(random(255), random(255), random(255))
square(i, j, random(spacing - 2))
}
}
noLoop()
}
Console
Clear

Next. . .

Well done for completing this module! Now, move on to the next module, which builds upon module A with increased depth and further exploration of coding concepts and ideas.