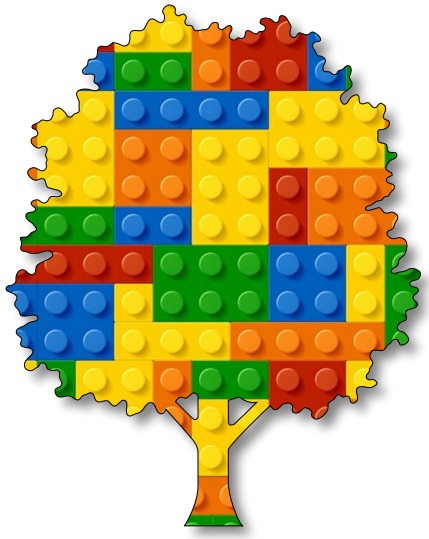
Algorithmic Art

Module B

Unit #9

arcs and mapping



Content

Module B Unit #9 arcs

Introduction to arcs and mapping

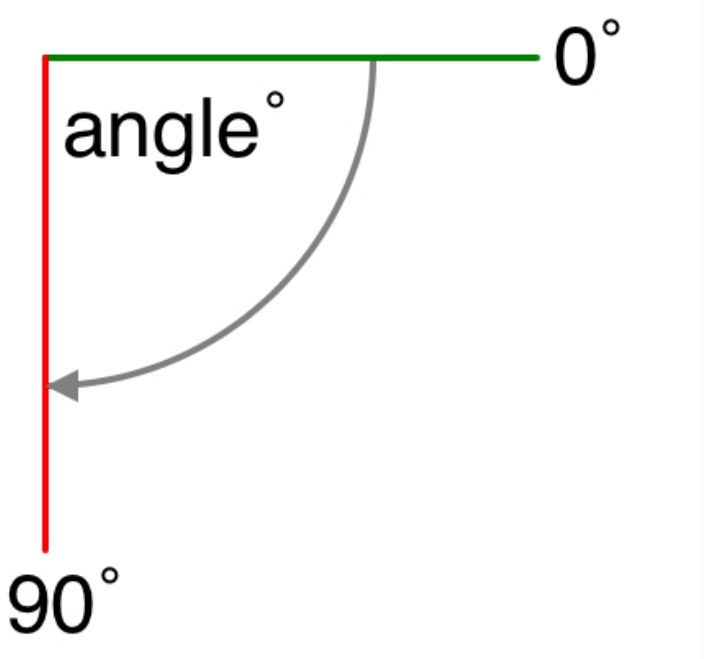
The arc() function has six arguments (three pairs), so if we consider this example:

```
arc(100, 200, 150, 300, 0, 180)
```

The first two arguments are the x and y coordinates (x is 100, y is 200).
The second two are the dimensions of the ellipse (150 wide, 300 tall).
The third pair of arguments are the angles from and to (starts at 0° and stops at 180°).

The angle is described clockwise from the horizontal 0°, to the new angle 90° (see Fig.1). It is measured in radians by default; hence, we will be using the angleMode() to change to degrees to make it more intuitive.

Figure 1: arc angle orientation



Key concepts

- arcs
- ellipseMode()
- mapping

Sketch B9.1 the 90° arc

A simple 90° arc.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  arc(200, 200, 200, 200, 0, 90)
}
```

🗒️ Notes

Because the width and height are the same, it scribes a circle, or a quarter of one in this case.

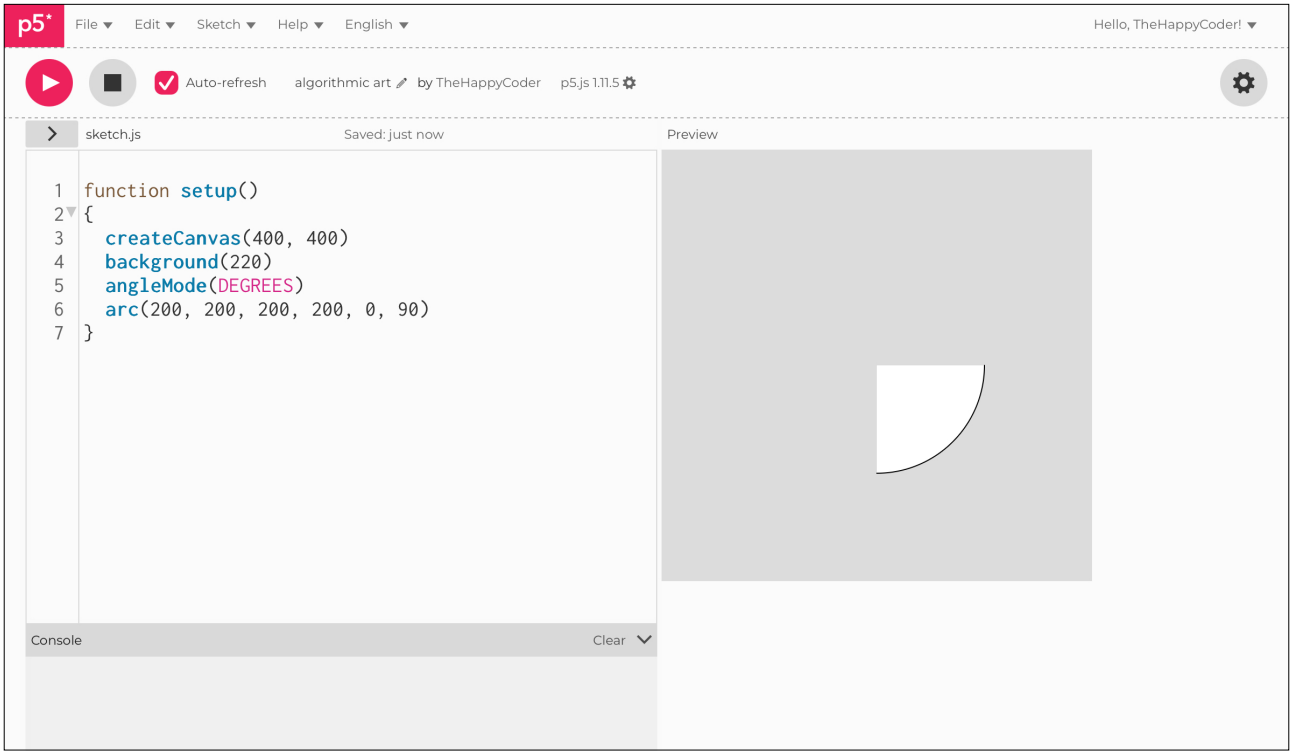
🌻 Challenge

Try other angles, starting and stopping.

🛠️ Code Explanation

`arc(200, 200, 200, 200, 0, 90)`	Arc drawn at position (200, 200) with a width and height of 200, starting at 0° and finishing at 90°

Figure B9.1


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
algorithmic art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400)
background(220)
angleMode(DEGREES)
arc(200, 200, 200, 200, 0, 90)
}
Console
Clear

Sketch B9.2 the negative 90° arc

We can start anywhere; here we start at −90° and arc round to 0°.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  arc(200, 200, 200, 200, -90, 0)
}
```

🗒️ Notes

A different way of using the angles.

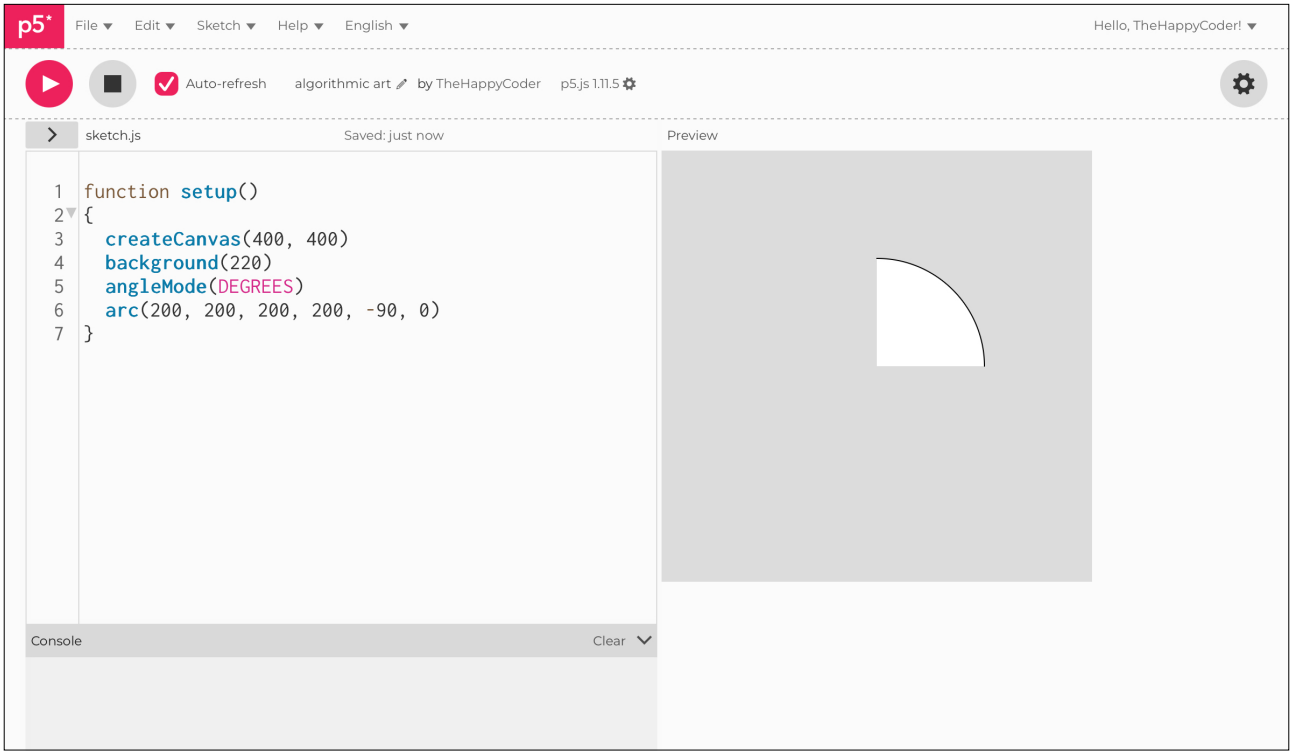
🌻 Challenge

Try other negative values.

🛠️ Code Explanation

`arc(200, 200, 200, 200, -90, 0)`	Starts at -90° and moves round to 0° clockwise

Figure B9.2


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
algorithmic art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400)
background(220)
angleMode(DEGREES)
arc(200, 200, 200, 200, -90, 0)
}
Console
Clear

Sketch B9.3 drawing a complete circle

Although there are a number of easy ways to draw a circle, we can do the same with arcs; here we increase the final angle by one in a `for()` loop.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  for (let i = 0; i < 360; i++)
  {
    arc(200, 200, 200, 200, i - 1, i)
  }
}
```

🗒️ Notes

We start with i - 1 because we want to draw the arc at each increment; otherwise, we get to the end of the loop and just draw the final arc from `0°` to `360°`.

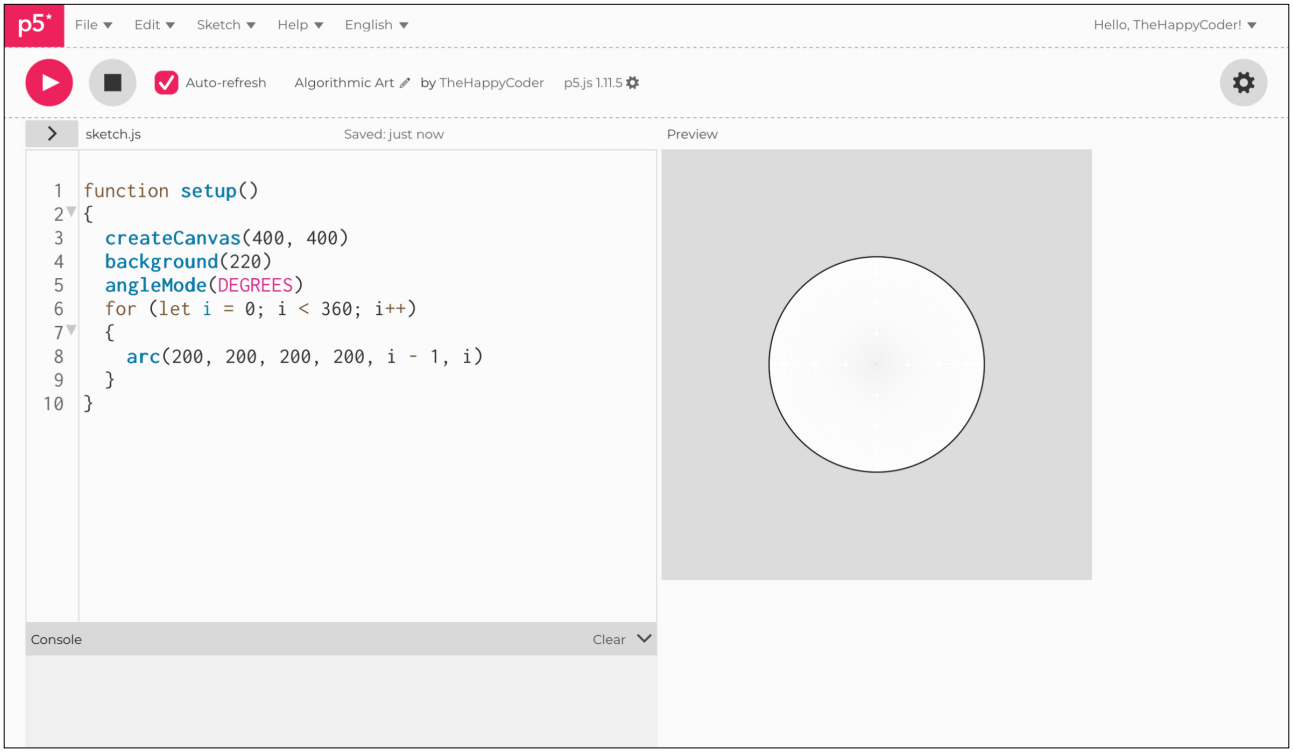
🌻 Challenge

What happens if you start with `0°`?

🛠️ Code Explanation

`for ( let i = 0; i < 360; i++)`	A for() loop starting at 0° and finishing at 360° in increments of 1°

Figure B9.3


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400)
background(220)
angleMode(DEGREES)
for (let i = 0; i < 360; i++)
{
arc(200, 200, 200, 200, i - 1, i)
}
}
Console
Clear

Sketch B9.4 random

We can now add some randomness to the circle so that each arc has a slightly different value.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  for (let i = 0; i < 360; i++)
  {
    arc(200, 200, 200 + random(-10, 10), 200 + random(-10, 10), i - 1, i)
  }
}
```

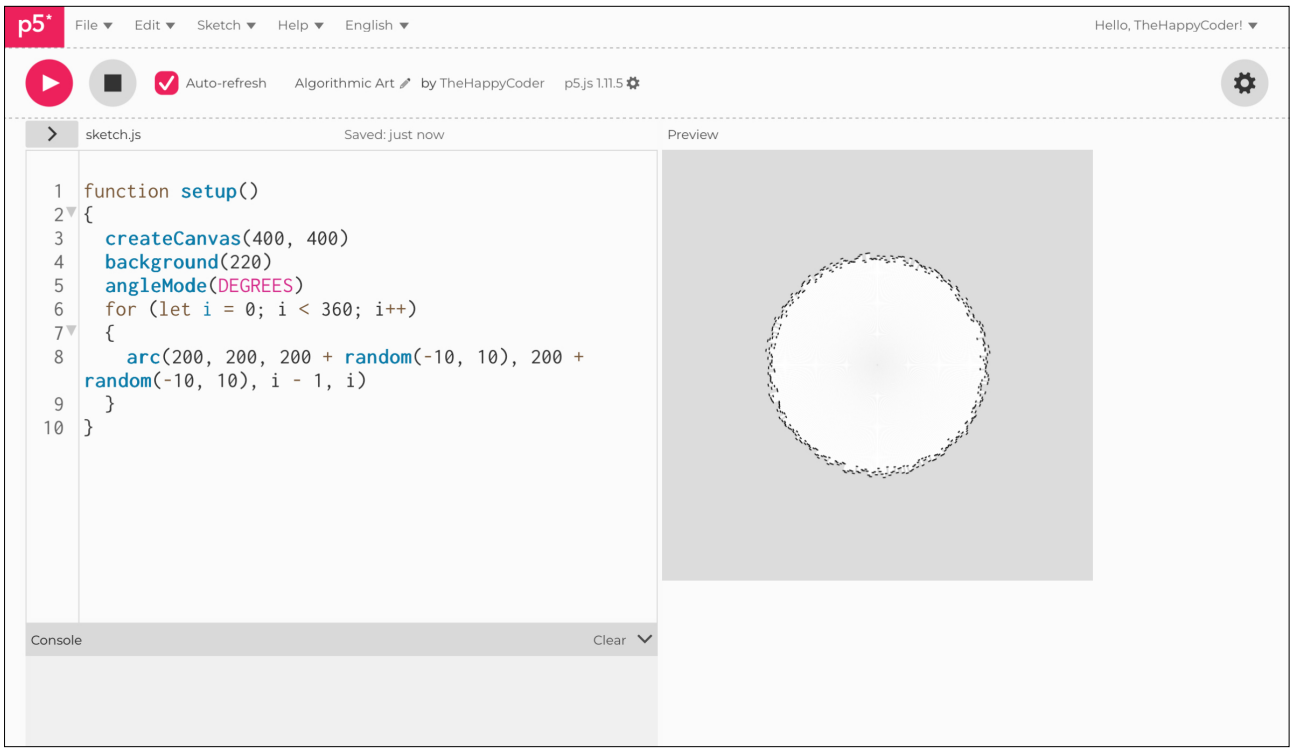
🗒️ Notes

Each diameter (width and height) will vary randomly between `-10` and `+10`.

🌻 Challenge

1. Add in a `noFill()`.
2. Try `i - 5` (or more). Why do you think you get that effect?

Figure B9.4


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400)
background(220)
angleMode(DEGREES)
for (let i = 0; i < 360; i++)
{
arc(200, 200, 200 + random(-10, 10), 200 + random(-10, 10), i - 1, i)
}
}
Console
Clear

Sketch B9.5 accentuating the randomness

This just pushes the boat out.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  noFill()
  for (let i = 0; i < 360; i++)
  {
    strokeWeight(2)
    arc(200, 200, 250 + random(-30, 30), 250 + random(-30, 30), i - 15, i)
  }
}
```

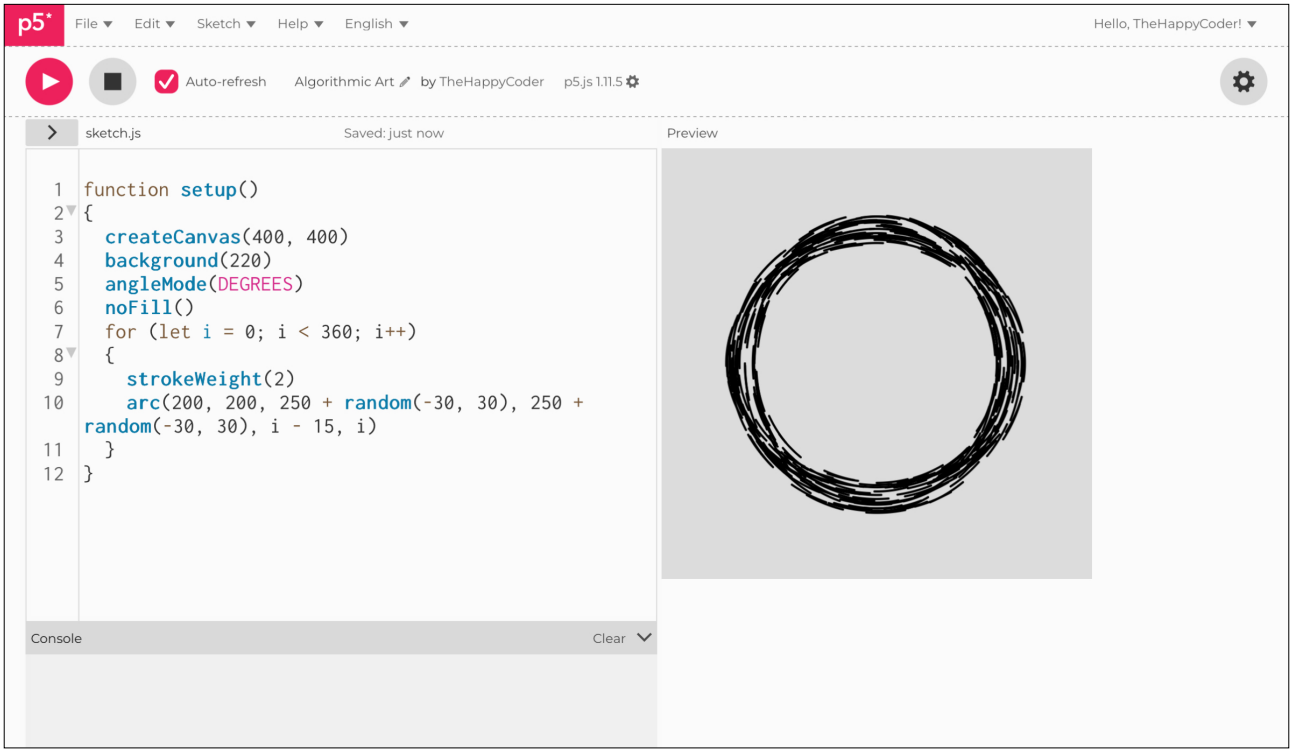
🗒️ Notes

Added `noFill()` and `strokeWeight()`.

🌻 Challenge

Add some colour and play with the values.

Figure B9.5


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400)
background(220)
angleMode(DEGREES)
noFill()
for (let i = 0; i < 360; i++)
{
strokeWeight(2)
arc(200, 200, 250 + random(-30, 30), 250 + random(-30, 30), i - 15, i)
}
}
Console
Clear

Sketch B9.6 spiral

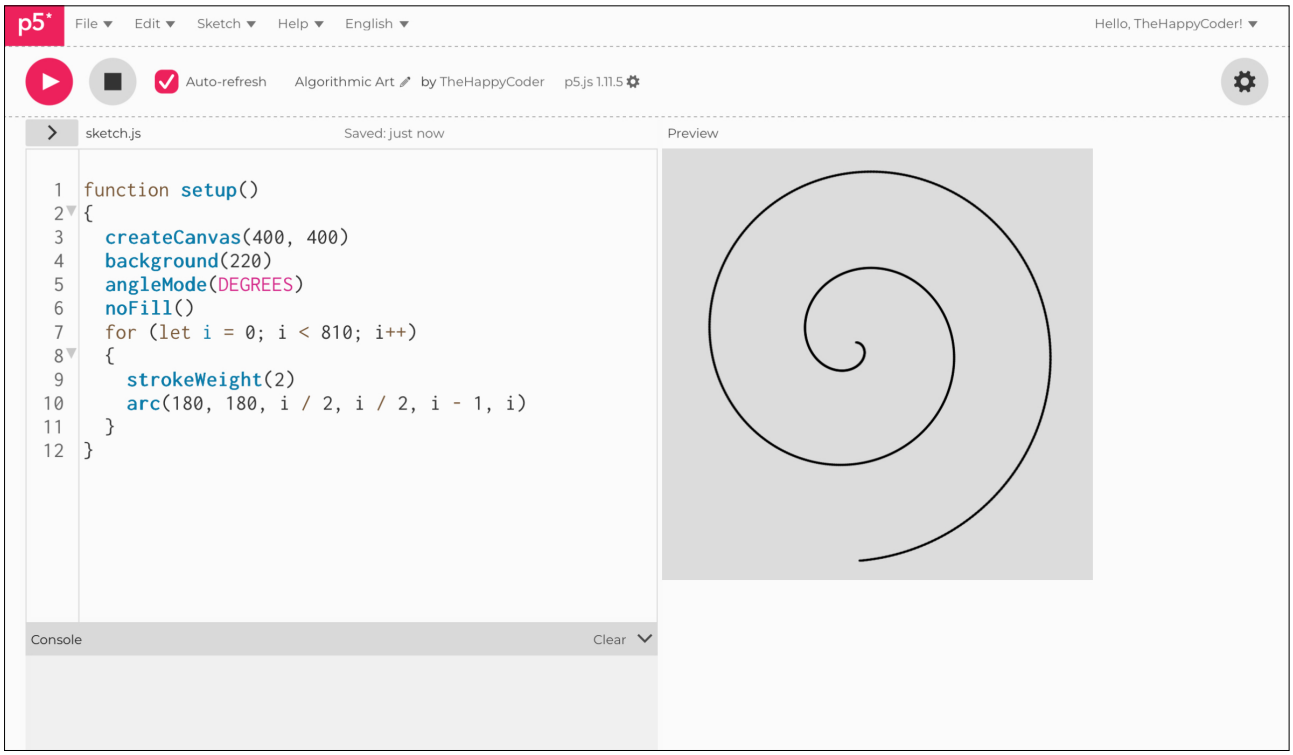
We can make it do a spiral.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  noFill()
  for (let i = 0; i < 810; i++)
  {
    strokeWeight(2)
    arc(180, 180, i / 2, i / 2, i - 1, i)
  }
}
```

Notes

Moved the centre of the spiral so it fits on the canvas.

Figure B9.6



Sketch B9.7 being a bit more OPEN

❗ start with a new sketch

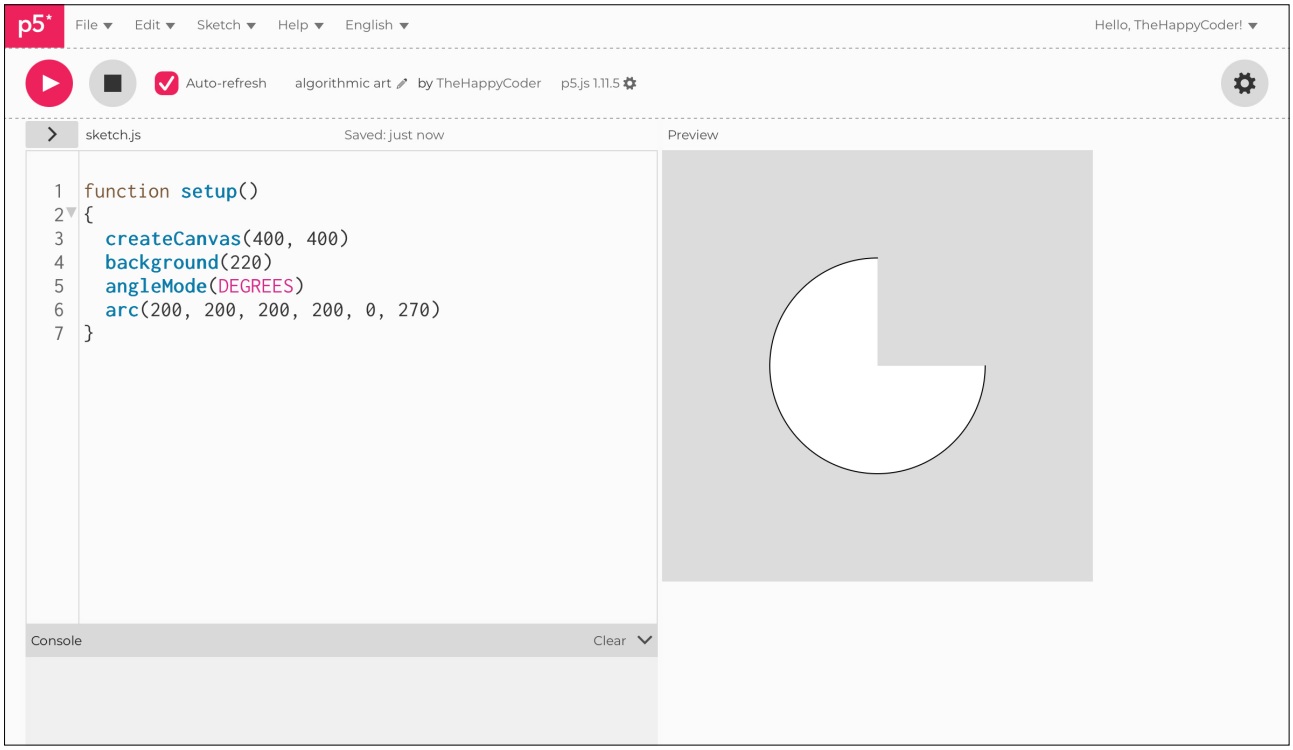
There are a number of attributes that you can use that change the appearance of an arc. The first one we will look at is called OPEN.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  arc(200, 200, 200, 200, 0, 270)
}
```

Notes

This is an arc through to 270°.

Figure B9.7



Sketch B9.8 OPEN sesame

Adding the word OPEN at the end of the function (7th argument).

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  arc(200, 200, 200, 200, 0, 270, OPEN)
}
```

Notes

It has a dramatic effect.

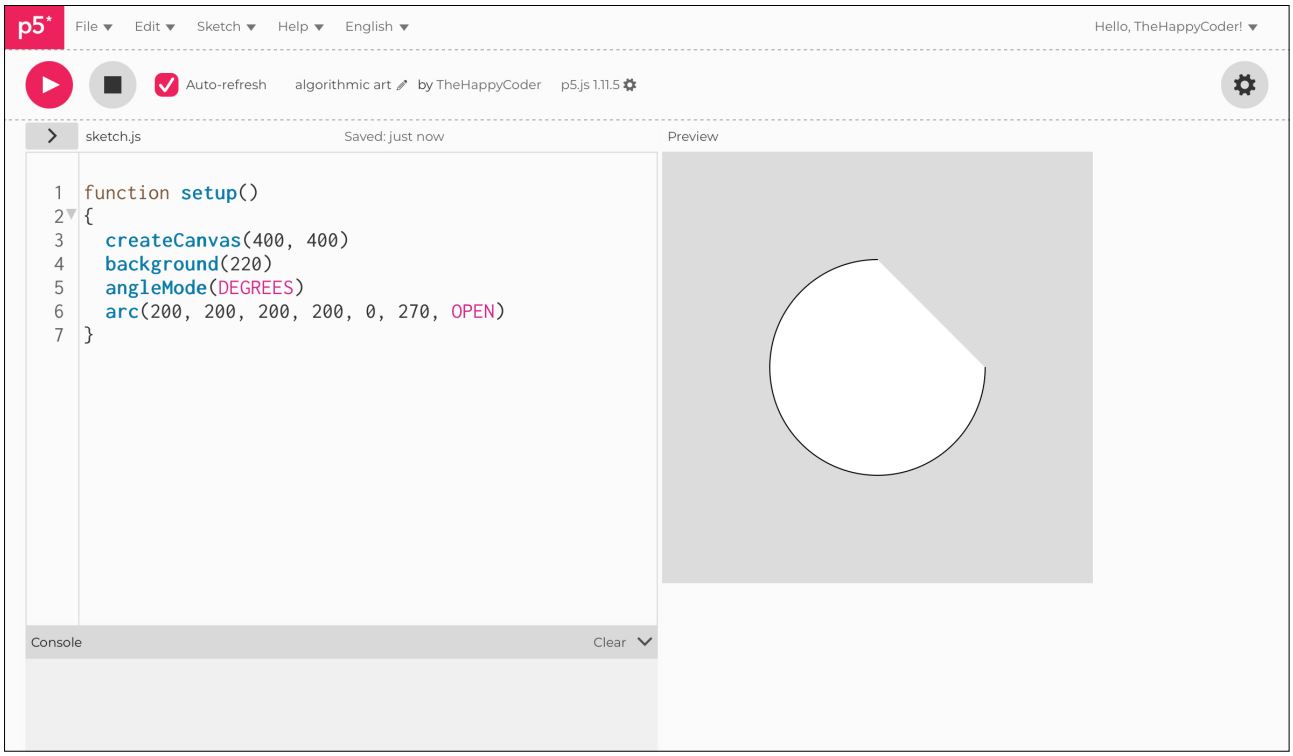
Challenge

Try it with other angles.

Code Explanation

`arc(200, 200, 200, 200, 0, 270, OPEN)`	Adding the attribute OPEN as the seventh argument

Figure B9.8



Sketch B9.9 strikes a CHORD

We will try a different attribute, the CHORD.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  arc(200, 200, 200, 200, 0, 270, CHORD)
}
```

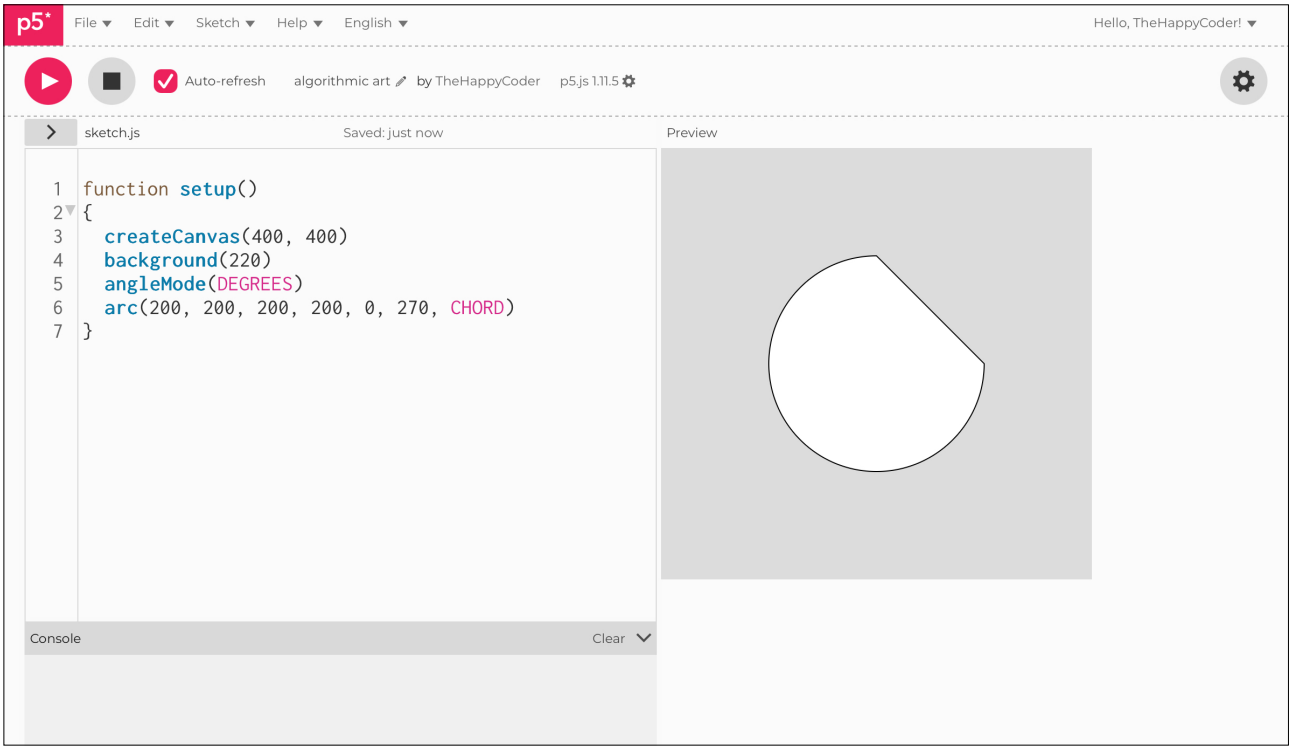
Notes

It has the same effect but also draws the outline.

Code Explanation

`arc(200, 200, 200, 200, 0, 270, CHORD)`	Adding the attribute CHORD as the seventh argument

Figure B9.9



Sketch B9.10 a piece of the PIE

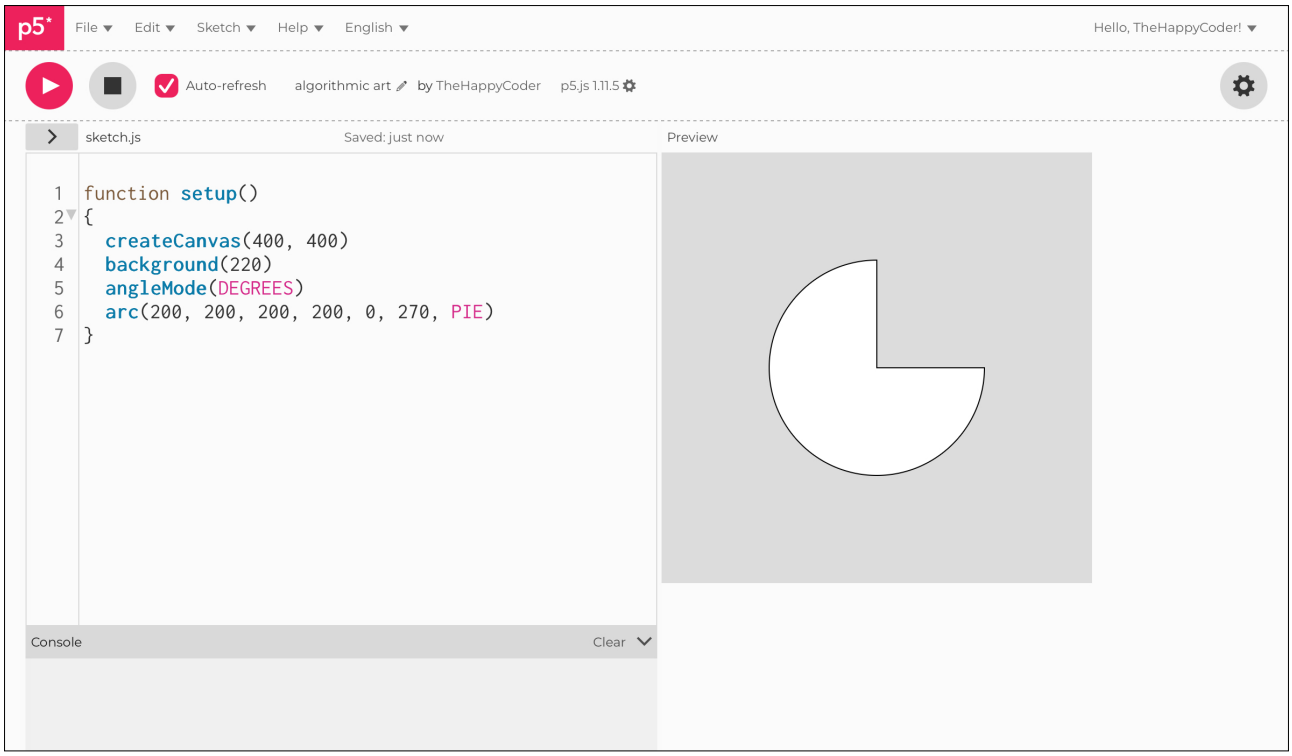
Another attribute, this time PIE.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  angleMode(DEGREES)
  arc(200, 200, 200, 200, 0, 270, PIE)
}
```

Notes

More like the original but with an outline.

Figure B9.10



Sketch B9.11 the making of a PAC-MAN

I will leave you to work out the logic, just a bit of fun; art can be anything.

```
let bite = 10
let angleA
let angleB

function setup()
{
  createCanvas(400, 400)
  angleMode(DEGREES)
}

function draw()
{
  background('darkblue')
  fill('yellow')
  angleA = bite * sin(millis()/2) + bite + 0.01
  angleB = -angleA
  arc(200, 200, 200, 200, angleA, angleB, PIE)
  fill('darkblue')
  circle(225, 150, 25)
}
```

🗒️ Notes

The sine of an angle fluctuates between `-1` and `+1`, that is why we multiply by the bit and add it; it doubles and then cancels itself out. We have a small addition (`0.01`) so that it doesn't close all the way to `zero`.

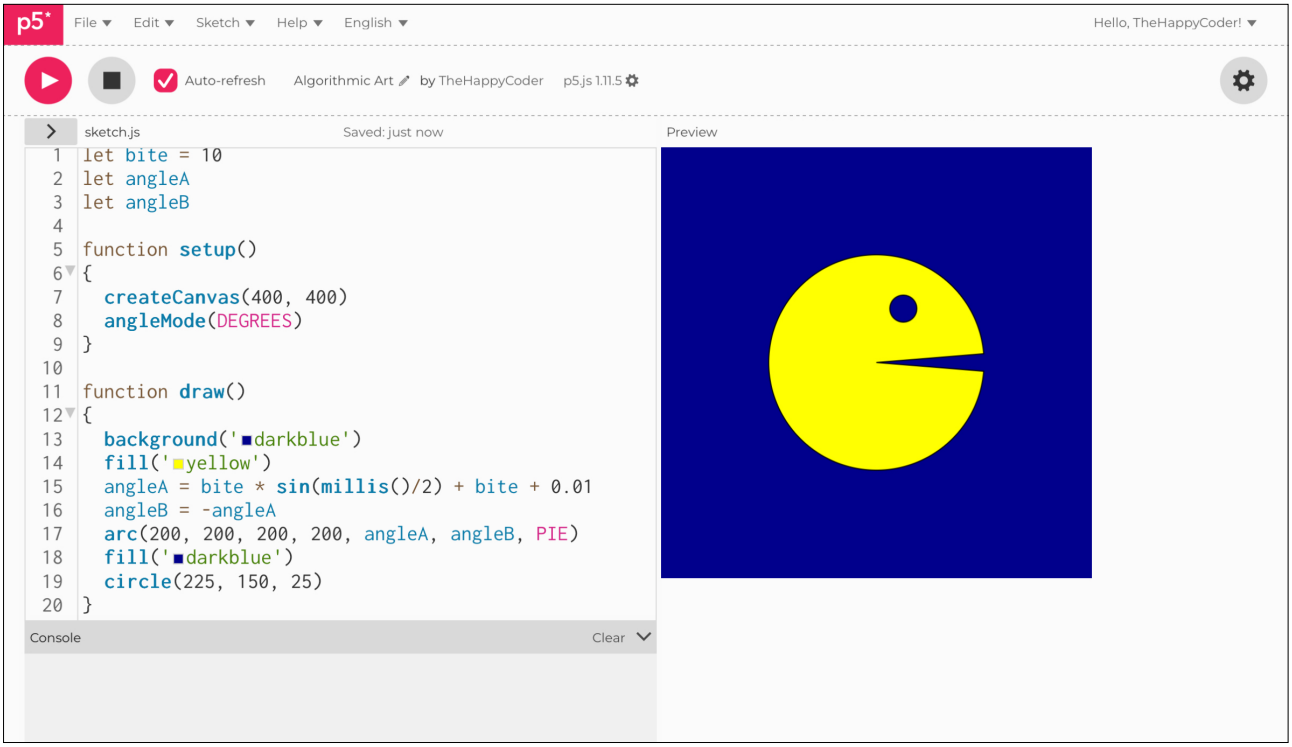
🌻 Challenge

You could use `frameCount` which also adds up very quickly but you may need to multiply by `15` or more/less

🛠️ Code Explanation

`millis()`	Counts the number of milliseconds since the sketch started running

Figure B9.11


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
let bite = 10
let angleA
let angleB
function setup()
{
createCanvas(400, 400)
angleMode(DEGREES)
}
function draw()
{
background('darkblue')
fill('yellow')
angleA = bite * sin(millis()/2) + bite + 0.01
angleB = -angleA
arc(200, 200, 200, 200, angleA, angleB, PIE)
fill('darkblue')
circle(225, 150, 25)
}
Console
Clear

Creating four arcs

These are our four arcs.

Arc 1: `arc(x, y, spacing, spacing, 0, 270)`
Arc 2: `arc(x, y, spacing, spacing, 90, 0)`
Arc 3: `arc(x, y, spacing, spacing, 270, 180)`
Arc 4: `arc(x, y, spacing, spacing, 180, 90)`

They start and finish in different places.

Figure 2: our four arcs

We are going to randomly select them, where a quarter of the time any single one is selected. As a basis, we will use the `10PRINT` code from a previous unit, which I have adapted slightly (rather than going through the whole thing again).

We will create a random number (between `0` and `1`), then divide the probability into four equal parts:

- less than `0.25`
- between `0.25` and `0.5`
- between `0.5` and `0.75`
- more than `0.75`

Sketch B9.12 10PRINT arcs

I have highlighted the main elements that are very different.

```
let x = 0
let y = 0
let spacing = 25

function setup()
{
  createCanvas(400, 400)
  ellipseMode(CORNER)
  angleMode(DEGREES)
  background(220)
  noFill()
  x = spacing
  y = spacing
}

function draw()
{
  if (random(1) <= 0.25)
  {
    arc(x, y, spacing, spacing, 0, 270)
  }
  else if (random(1) >= 0.25 && random(1) <= 0.5)
  {
    arc(x, y, spacing, spacing, 90, 0)
  }
  else if (random(1) >= 0.5 && random(1) <= 0.75)
  {
    arc(x, y, spacing, spacing, 270, 180)
  }
  else
  {
    arc(x, y, spacing, spacing, 180, 90)
  }
  x += spacing
```

```
  if (x >= width - spacing)
  {
    x = spacing
    y += spacing
  }
  if (y >= height - spacing)
  {
    noLoop()
  }
}
```

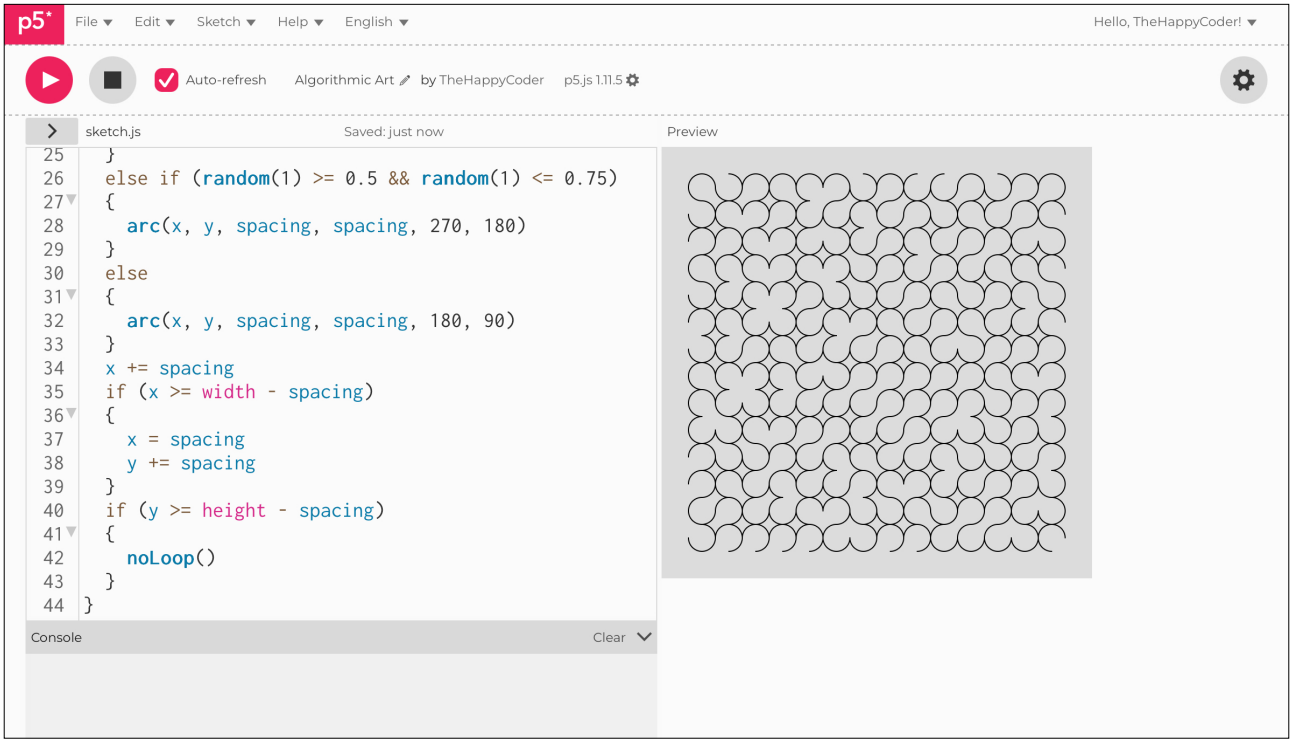
🗒️ Notes

The `ellipseMode()` function allows you to move the centre of the circle (in terms of coordinates).

🛠️ Code Explanation

`ellipseMode(CORNER)`	Puts the coordinates of the circle top left hand rather than in the centre
`if (random(1) <= 0.25)`	If less than 0.25 draw first arc
`else if (random(1) >= 0.25 && random(1) <= 0.5)`	If between than 0.25 and 0.5 draw second arc
`else if (random(1) >= 0.5 && random(1) <= 0.75)`	If between than 0.5 and 0.75 draw third arc
`else`	If none of the above then draw the fourth arc

Figure B9.12


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
25 }
26 else if (random(1) >= 0.5 && random(1) <= 0.75)
27 {
28 arc(x, y, spacing, spacing, 270, 180)
29 }
30 else
31 {
32 arc(x, y, spacing, spacing, 180, 90)
33 }
34 x += spacing
35 if (x >= width - spacing)
36 {
37 x = spacing
38 y += spacing
39 }
40 if (y >= height - spacing)
41 {
42 noLoop()
43 }
44 }
Console
Clear

Sketch B9.13 mapping

We can use a function called `map()` to map the horizontal (`mouseX`) position of the mouse across the canvas to the angle of the arc. The `map()` function takes five arguments.

The first one is the input variable you want to map, in this case it is the `mouseX` value which will range from `0` to `400`. The second two are start and end values for that input value (`0` and `400`), and the final two values are the values you want to map to, in this case the `angle` of the arc which we want in a range of `0` to `360`.

In effect, we are mapping `0` to `400` onto `0` to `360`.

```
let angle = 0

function setup()
{
  createCanvas(400, 400)
  angleMode(DEGREES)
}

function draw()
{
  background(220)
  fill(255)
  angle = map(mouseX, 0, width, 0, 360)
  if (angle > 360)
  {
    angle = 360
  }
  if (angle < 0)
  {
    angle = 0
  }
  arc(200, 200, 200, 200, 0, angle, PIE)
  fill(0)
  textSize(32)
  text(floor(angle) + '°', 100, 100)
}
```

🗒️ Notes

We have also put limitations on how far to the left and right of the canvas we can go; otherwise, you will have values bigger than 360° and negative numbers.

🛠️ Code Explanation

`angle = map(mouseX, 0, width, 0, 360)`	Mapping the mouse position (0-400) onto the angle (0-360)

Figure B9.13