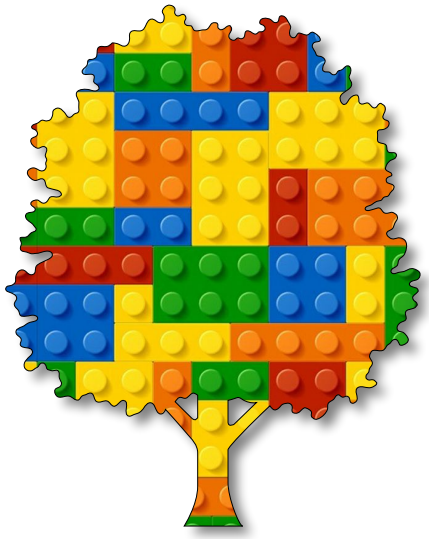
Algorithmic Art

Module B
Unit #2
orbit & oscillate



Content

Module B Unit #2 orbit and oscillate

Introduction to orbit and oscillate

Using sine and cosine, we will move in a circular motion, followed by an introduction to simple harmonic motion (SHM). With these simple algorithms, we can create some interesting patterns.

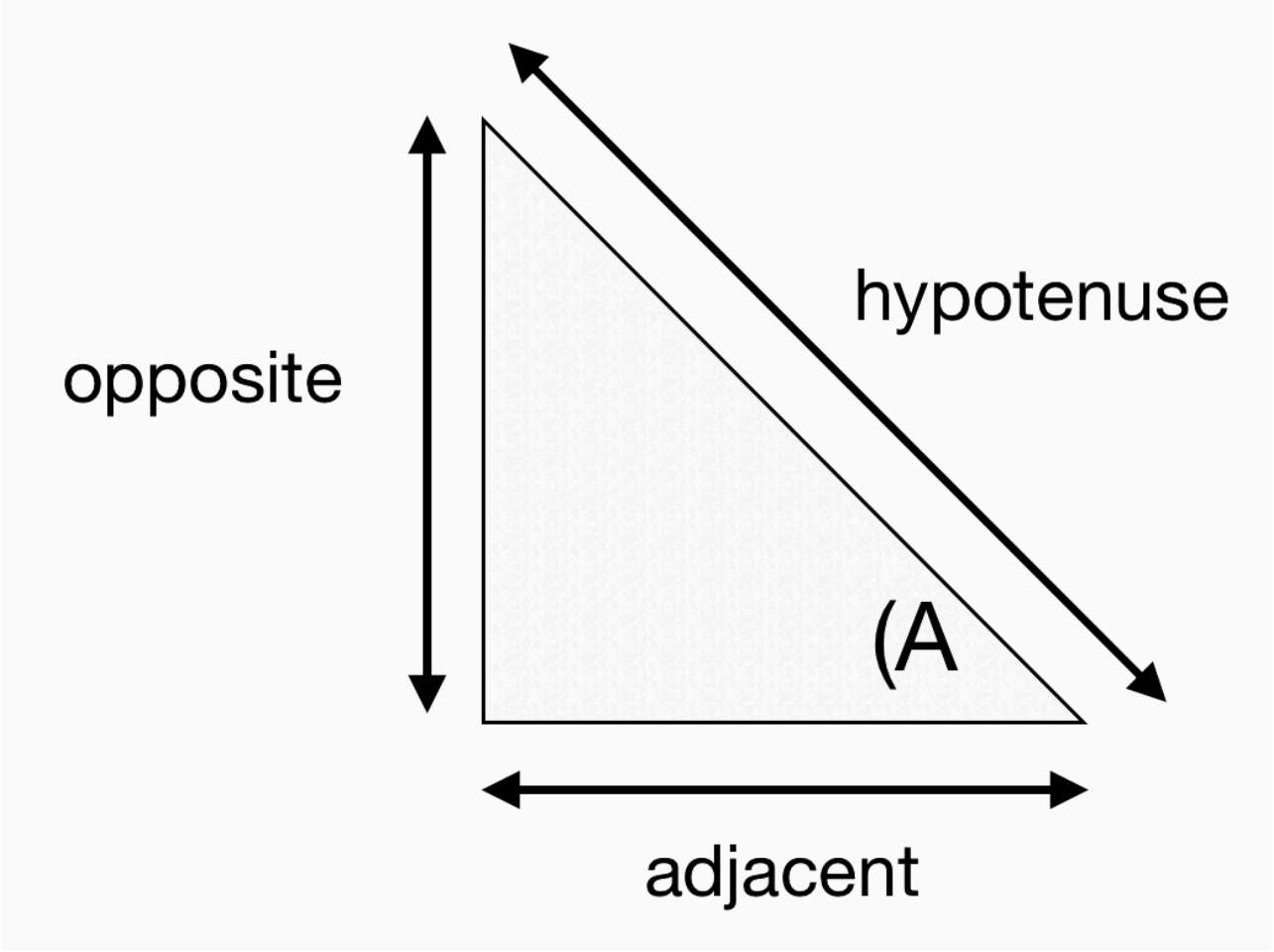
Key concepts:

- polar co-ordinates
- sine and cosine
- oscillation
- simple harmonic motion

In this section, we will look at angles with cosine (cos) and sine (sin). This is a little bit of maths. The sine of an angle in a right-handed triangle is the length of the opposite side divided by the hypotenuse. See fig. 1 below.

```
sine(angle)   = opposite / hypotenuse
cosine(angle) = adjacent / hypotenuse
```

Figure 1: A triangle labelled relative to angle (A)

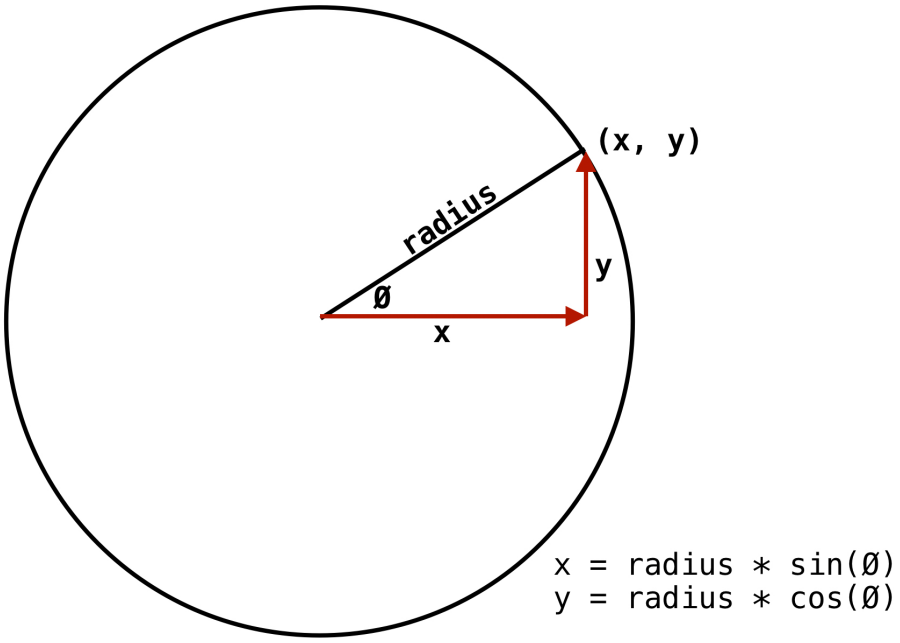


We are going to move a circle in a large circular motion. This may seem easy, but we will need the above maths to work out the co-ordinates of the circle. Without going into lots of maths, the equation for the co-ordinates from the centre point (0, 0) is:

```
x = radius * sin(angle)
y = radius * cos(angle)
```

See fig.2 below where Ø is the angle.

Figure 2: coordinates of a point on a circle



Sketch B2.1 a simple circle

❗ Start a new sketch

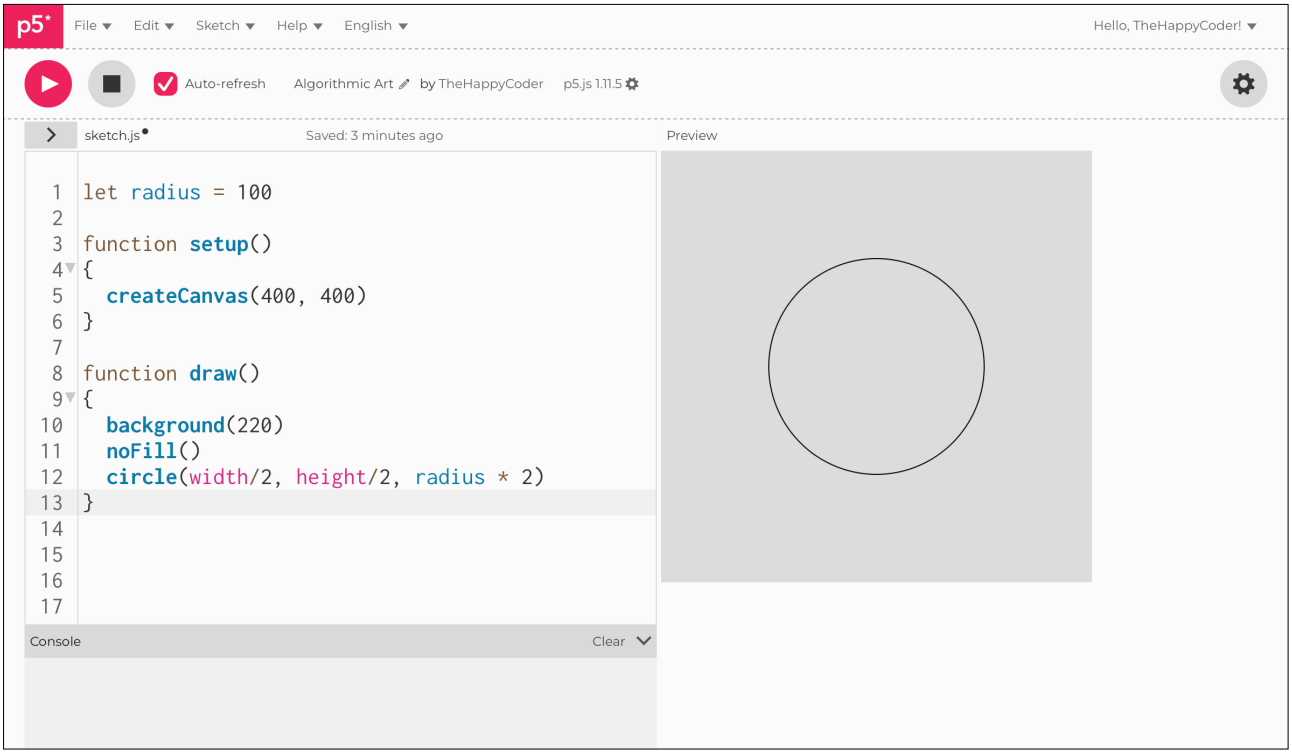
I have drawn a circle in the centre of the canvas with a diameter of `200` (`radius * 2`).

```
let radius = 100


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

function draw()
{
  background(220)
  noFill()
  circle(width/2, height/2, radius * 2)
}
```

🗒️ Notes

What we are going to do is have a small circle follow the path of the large circle.

Figure B2.1


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: 3 minutes ago
Preview
let radius = 100
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
noFill()
circle(width/2, height/2, radius * 2)
}
Console
Clear

Sketch B2.2 creating the variables

We want three variables: an `angle` and the `x` and `y` co-ordinates to draw our circle.

```
let radius = 100
let angle = 0
let x
let y

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

function draw()
{
  background(220)
  noFill()
  circle(width/2, height/2, radius * 2)
}
```

Notes

Nothing yet to show for it, though.

Sketch B2.3 the co-ordinates

We put in the co-ordinates of a point on the circle.

```
let radius = 100
let angle = 0
let x
let y

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

function draw()
{
  background(220)
  noFill()
  circle(width/2, height/2, radius * 2)
  x = radius * sin(angle)
  y = radius * cos(angle)
}
```

🗒️ Notes

This is not correct, as you will see in the next sketch.

🛠️ Code Explanation

`x = radius * sin(angle)`	The x coordinate on a circle at that angle with a radius
`y = radius * cos(angle)`	The y coordinate on a circle at that angle with a radius

Sketch B2.4 draw the circle

Here we can add the small (yellow) circle that we want to follow the path of the large circle.

```
let radius = 100
let angle = 0
let x
let y

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

function draw()
{
  background(220)
  noFill()
  circle(width/2, height/2, radius * 2)
  x = radius * sin(angle)
  y = radius * cos(angle)
  fill('yellow')
  circle(x, y, 20)
}
```

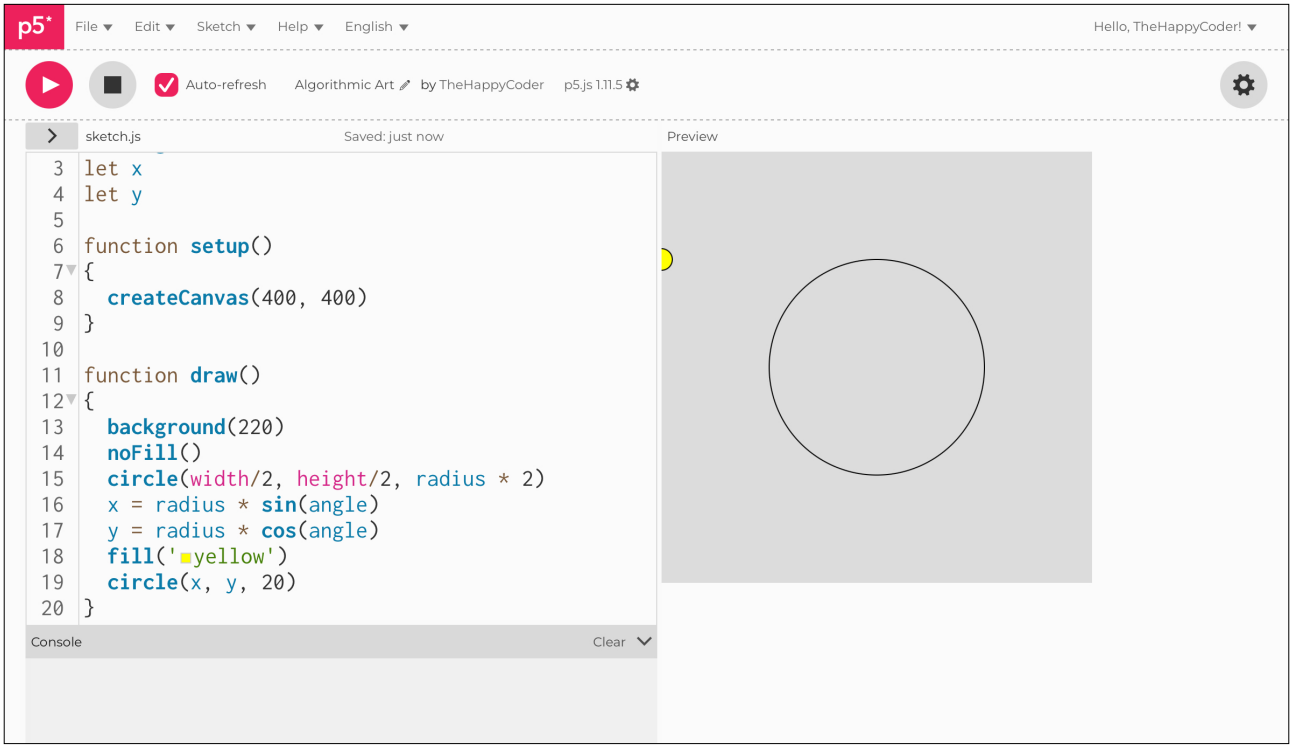
🗒️ Notes

We have a problem, the yellow circle is not on the path of the larger circle. This is because we need to translate the origin of the coordinates to the centre of the canvas (the large circle). Remember the diagram (fig.2).

🌻 Challenge

Can you work out how to do that?

Figure B2.4


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
let x
let y
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
noFill()
circle(width/2, height/2, radius * 2)
x = radius * sin(angle)
y = radius * cos(angle)
fill('yellow')
circle(x, y, 20)
}
Console
Clear

Sketch B2.5 translate

When we translate the co-ordinates for the origin, we shift everything to where we need it, similar to rotating the baton.

```
let radius = 100
let angle = 0
let x
let y

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

function draw()
{
  background(220)
  noFill()
  circle(width/2, height/2, radius * 2)
  translate(width/2, height/2)
  x = radius * sin(angle)
  y = radius * cos(angle)
  fill('yellow')
  circle(x, y, 20)
}
```

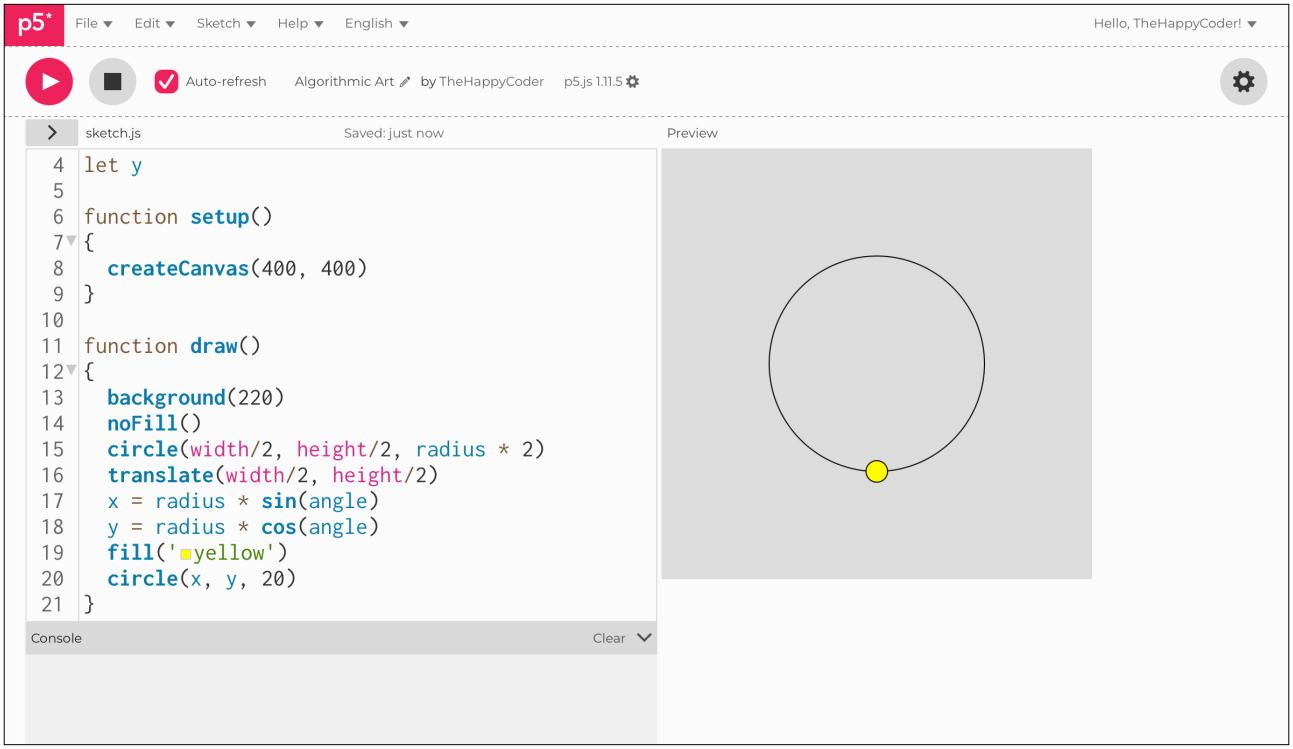
Notes

It should sit nicely on the large circle.

Challenges

1. Change the value of the angle.
2. What happens if you translate before drawing the large circle?

Figure B2.5


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
4 let y
5
6 function setup()
7 {
8 createCanvas(400, 400)
9 }
10
11 function draw()
12 {
13 background(220)
14 noFill()
15 circle(width/2, height/2, radius * 2)
16 translate(width/2, height/2)
17 x = radius * sin(angle)
18 y = radius * cos(angle)
19 fill('yellow')
20 circle(x, y, 20)
21 }
Console
Clear

Sketch B2.6 moving it

Our final step is to get the yellow circle moving. We add another variable called `velocity` (angular) and increment the `angle` according to that variable.

```
let radius = 100
let angle = 0
let x
let y
let velocity = 0.05

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

function draw()
{
  background(220)
  noFill()
  circle(width/2, height/2, radius * 2)
  translate(width/2, height/2)
  x = radius * sin(angle)
  y = radius * cos(angle)
  fill('yellow')
  circle(x, y, 20)
  angle += velocity
}
```

Notes

You should see the yellow circle orbiting the big circle. We give it an initial angular velocity of `0.05`.

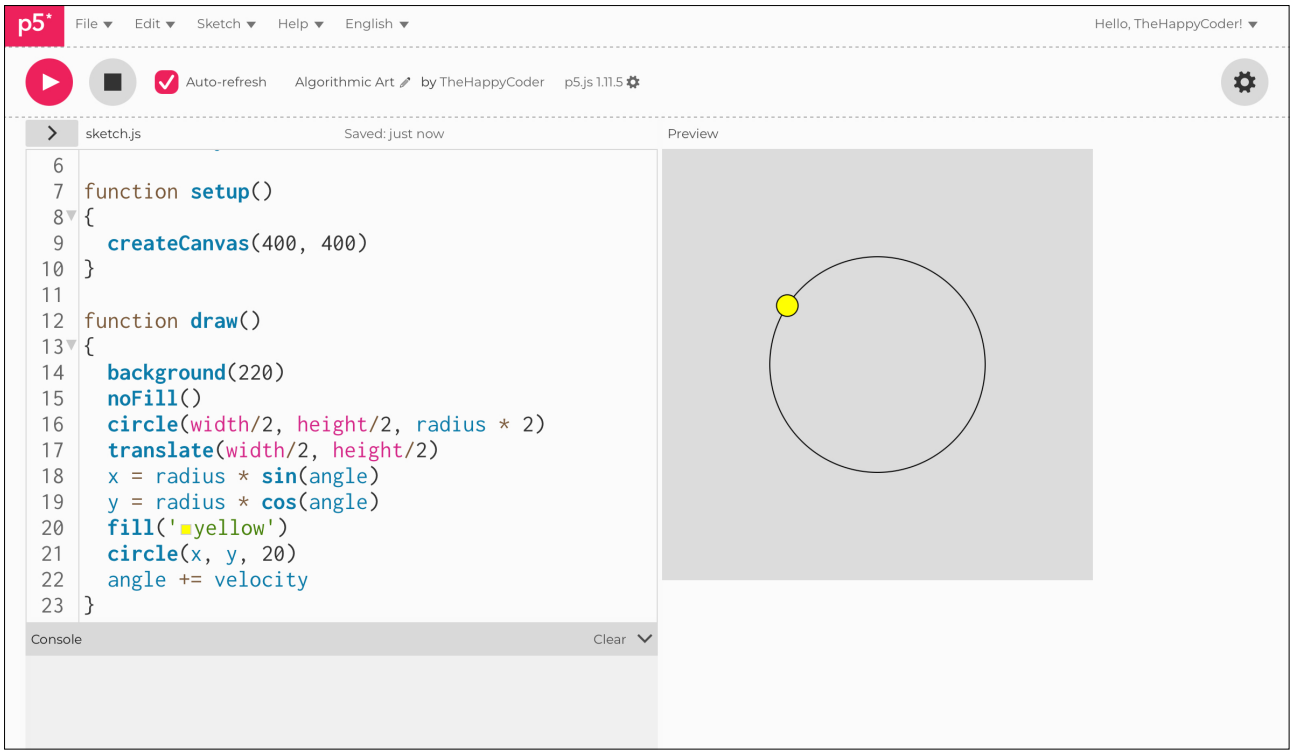
🌻 Challenges

1. Try other values for the velocity.
2. What happens if you give it a negative velocity?
3. How would you have more than one circle?

🛠️ Code Explanation

`angle += velocity`	Changing the angle by that (velocity) amount

Figure B2.6



Sketch B2.7 accelerating

Increasing the angular velocity.

```
let radius = 100
let angle = 0
let x
let y
let velocity = 0.05
let acceleration = 0.0005

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

function draw()
{
  background(220)
  noFill()
  circle(width/2, height/2, radius * 2)
  translate(width/2, height/2)
  x = radius * sin(angle)
  y = radius * cos(angle)
  fill('yellow')
  circle(x, y, 20)
  angle += velocity
  velocity += acceleration
}
```

🗒️ Notes

In this, we simply accelerate the orbiting circle by increasing the velocity incrementally.

🌻 Challenges

1. What happens if you start with a negative velocity?
2. How would you go about changing the direction of rotation every time it reaches the top position?

We have moved an object (yellow circle) scribing a circle using the co-ordinates of all the points on a circle. Using sine and cosine, we can describe other motions such as linear. This motion is called simple harmonic motion, where the oscillation is about a central point.

We can also use this to describe an ellipse (oval) as well as other effects that can be useful in creative coding.

Sketch B2.8 a line and a circle

❗ Start a new sketch.
We are drawing a red circle at the end of the line.

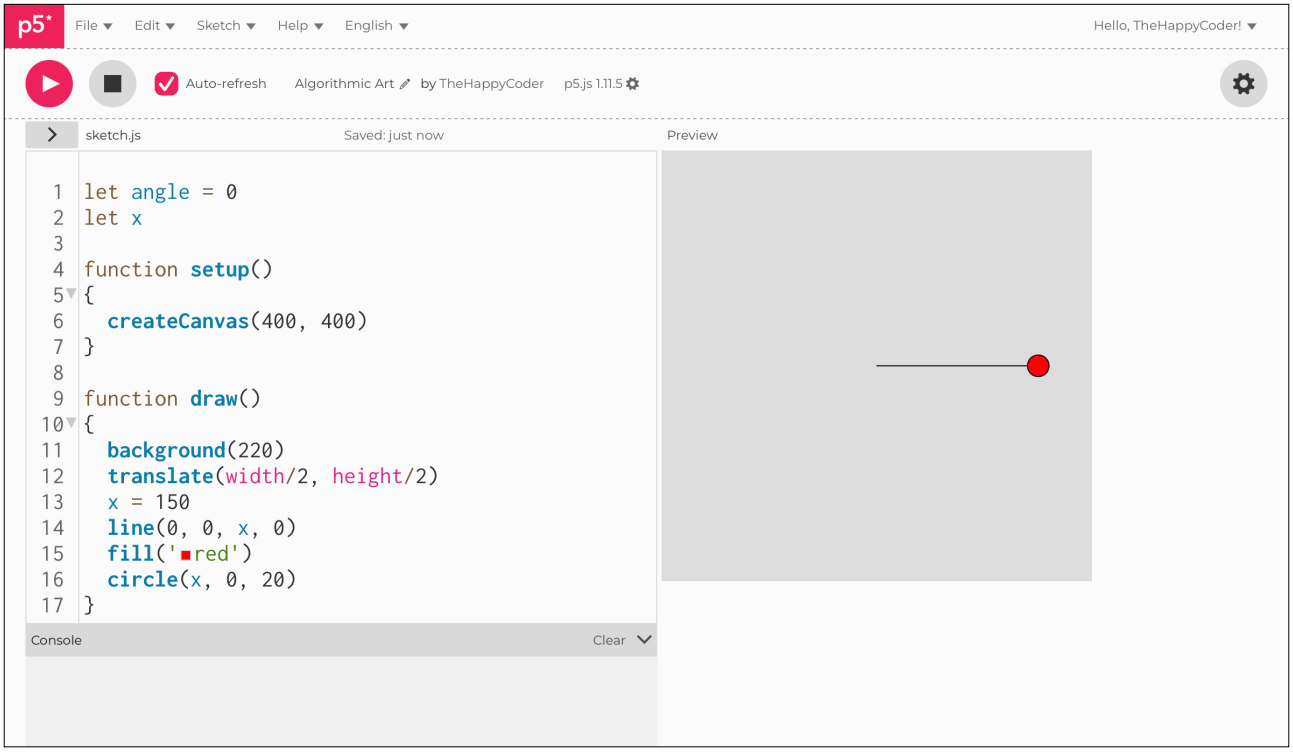
```
let angle = 0
let x

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  x = 150
  line(0, 0, x, 0)
  fill('red')
  circle(x, 0, 20)
}
```

🗒️ Notes
All very static to start with.

🛠️ Code Explanation

`translate(width/2, height/2)`	Translate all the coordinates to the centre of the canvas
`x = 150`	The x coordinate is initialised to 150
`line(0, 0, x, 0)`	Draw a line from the centre of the canvas to the x coordinate
`circle(x, 0, 20)`	Draw a red circle at the end of the line where y is kept at 0 (central plane of the canvas

Figure B2.8


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 angle = 0
let x
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(width/2, height/2)
x = 150
line(0, 0, x, 0)
fill('red')
circle(x, 0, 20)
}
Console
Clear

Sketch B2.9 amplitude

Instead of radius, we will use the variable name amplitude. We are only doing this for the x value for now.

```
let angle = 0
let x
let amplitude = 150

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  x = amplitude * sin(angle)
  line(0, 0, x, 0)
  fill('red')
  circle(x, 0, 20)
}
```

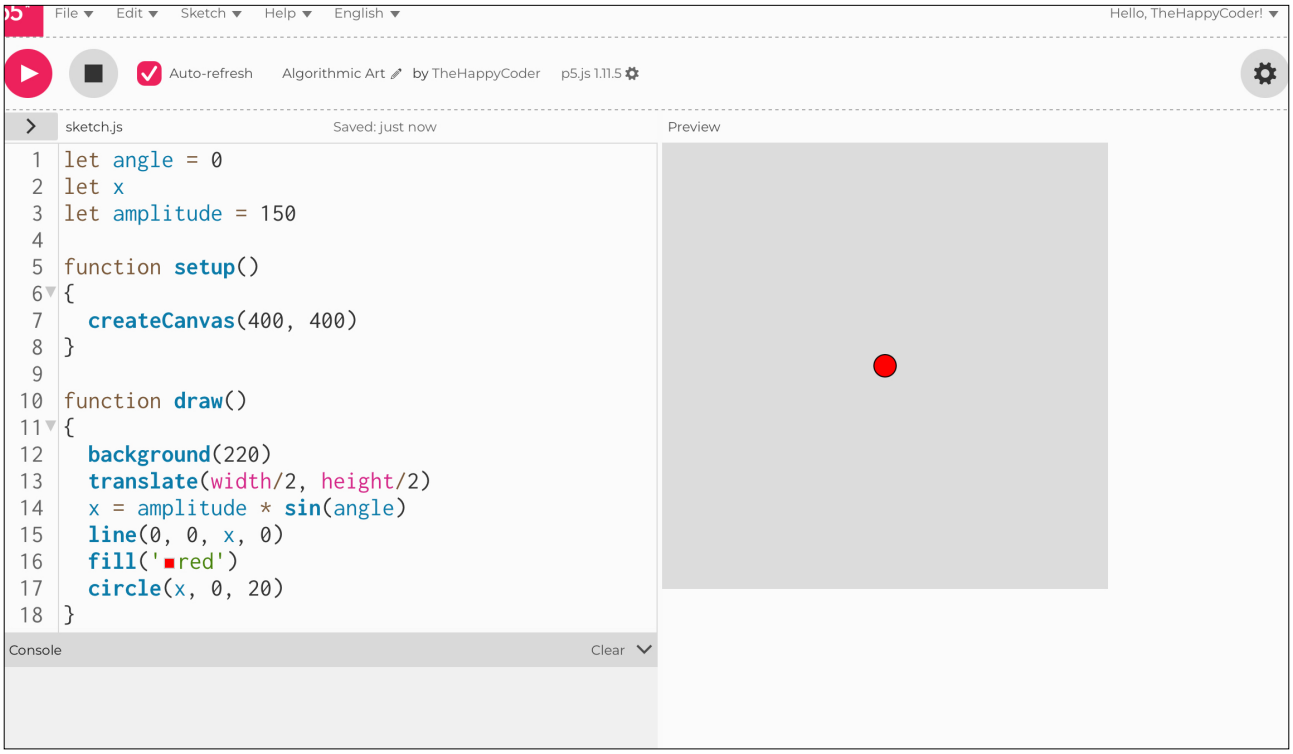
🗒️ Notes

It moves the red circle to the centre; the line is still there!

🛠️ Code Explanation

`x = amplitude * sin(angle)`	Defining the position of the x component of the coordinates

Figure B2.9


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 angle = 0
let x
let amplitude = 150
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(width/2, height/2)
x = amplitude * sin(angle)
line(0, 0, x, 0)
fill('red')
circle(x, 0, 20)
}
Console
Clear

Sketch B2.10 moving it

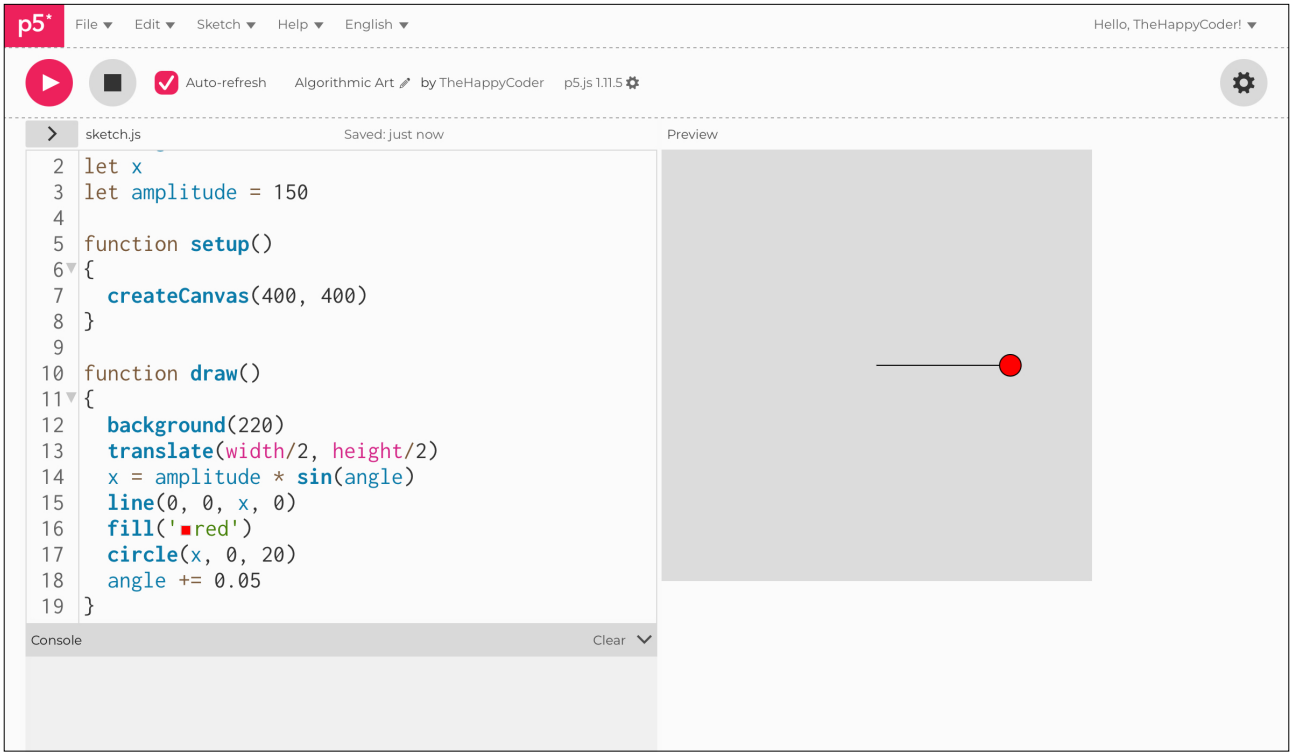
Although we are using sine wave motion, we now have a smooth, simple harmonic motion (SHM). This is similar to the motion of the circle but in a linear direction only.

```
let angle = 0
let x
let amplitude = 150

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  x = amplitude * sin(angle)
  line(0, 0, x, 0)
  fill('red')
  circle(x, 0, 20)
  angle += 0.05
}
```

🗒️ Notes

Now it oscillates nicely about the origin.

Figure B2.10


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 x
let amplitude = 150
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(width/2, height/2)
x = amplitude * sin(angle)
line(0, 0, x, 0)
fill('red')
circle(x, 0, 20)
angle += 0.05
}
Console
Clear

Sketch B2.11 an ellipse

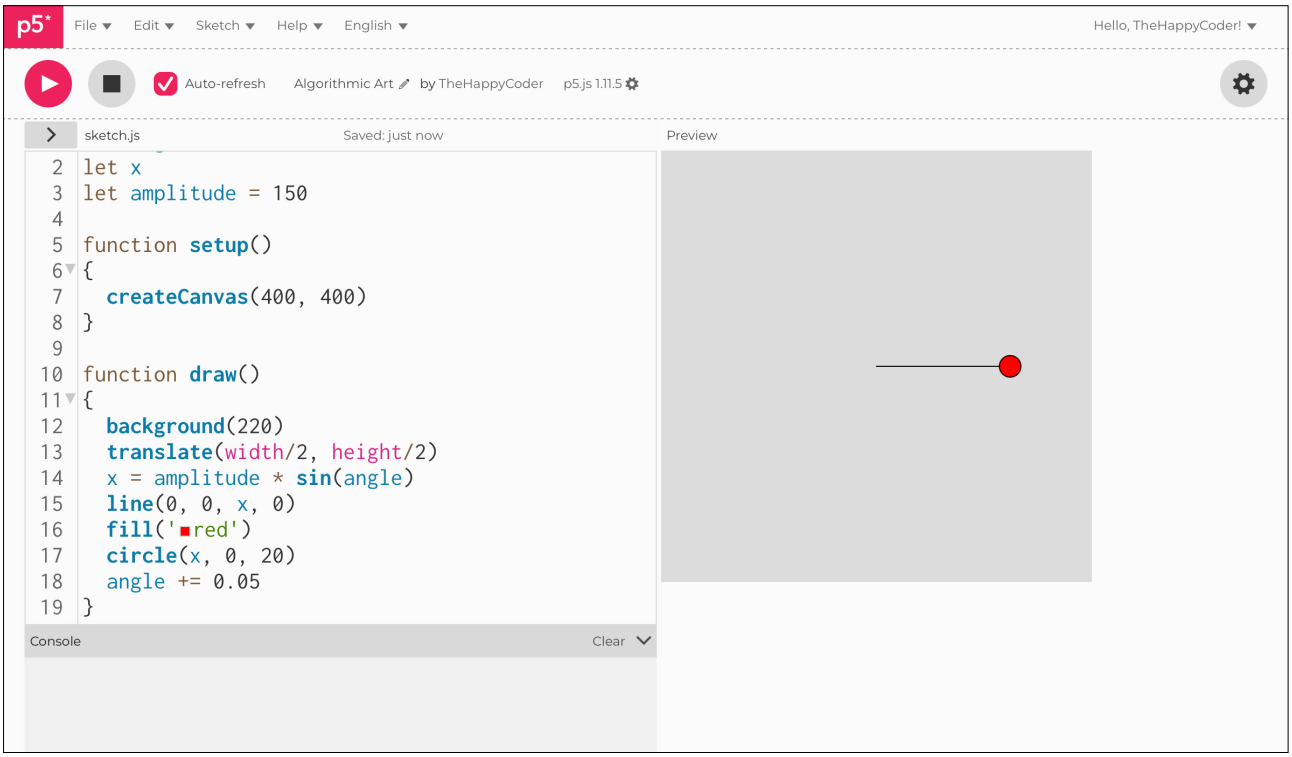
A circle or linear movement is all very well, but we can also do an ellipse. Before we make too many changes at once, let's take it one step at a time. We will change the name of the `amplitude` to `amplitudeX`.

```
let angle = 0
let x
let amplitudeX = 150

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  x = amplitudeX * sin(angle)
  line(0, 0, x, 0)
  fill('red')
  circle(x, 0, 20)
  angle += 0.05
}
```

🗒️ Notes

Nothing drastically changing, just renaming.

Figure B2.12


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
let x
let amplitude = 150
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(width/2, height/2)
x = amplitude * sin(angle)
line(0, 0, x, 0)
fill('red')
circle(x, 0, 20)
angle += 0.05
}
Console
Clear

Sketch B2.12 the y component

To draw an ellipse, we need the y component, as we did with the circle. They should make sense. The difference being that we have a different amplitude (radius) for y than for x, and it is this that gives it the ellipse shape.

```
let angle = 0
let x
let y
let amplitudeX = 150
let amplitudeY = 50

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  x = amplitudeX * sin(angle)
  y = amplitudeY * cos(angle)
  line(0, 0, x, y)
  fill('red')
  circle(x, y, 20)
  angle += 0.05
}
```

Notes

Adding in the y component to the circle and the line, you now have it scribing an elliptical shape.

Making patterns

Here we can play with many different variables. I will just show you a few and highly recommend that you play with them to see what patterns you can create. Later on, we will create a pattern called phyllotaxis, which is very similar to what has been covered here.

Sketch B2.13 moving the background

If we move the `background()` into the `setup()` function, it draws the background only once. We can use `//` at the beginning of a line of code for four reasons:

1. We can remove unwanted code for the moment but may want it later.
2. There is a problem with our code and removing a line of code means it can help us debug (find the problem) the code.
3. We can leave comments or information that is helpful for yourself or someone else looking at the code.
4. The `//` has the effect of making it invisible to the computer and will ignore anything on that line.

Here we are commenting out the `background()` and the `line()` in the `draw()` function. I have highlighted them.

```
let angle = 0
let x
let y
let amplitudeX = 150
let amplitudeY = 50

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

function draw()
{
  // background(220)
  translate(width/2, height/2)
  x = amplitudeX * sin(angle)
  y = amplitudeY * cos(angle)
  // line(0, 0, x, y)
  fill('red')
  circle(x, y, 20)
  angle += 0.05
}
```

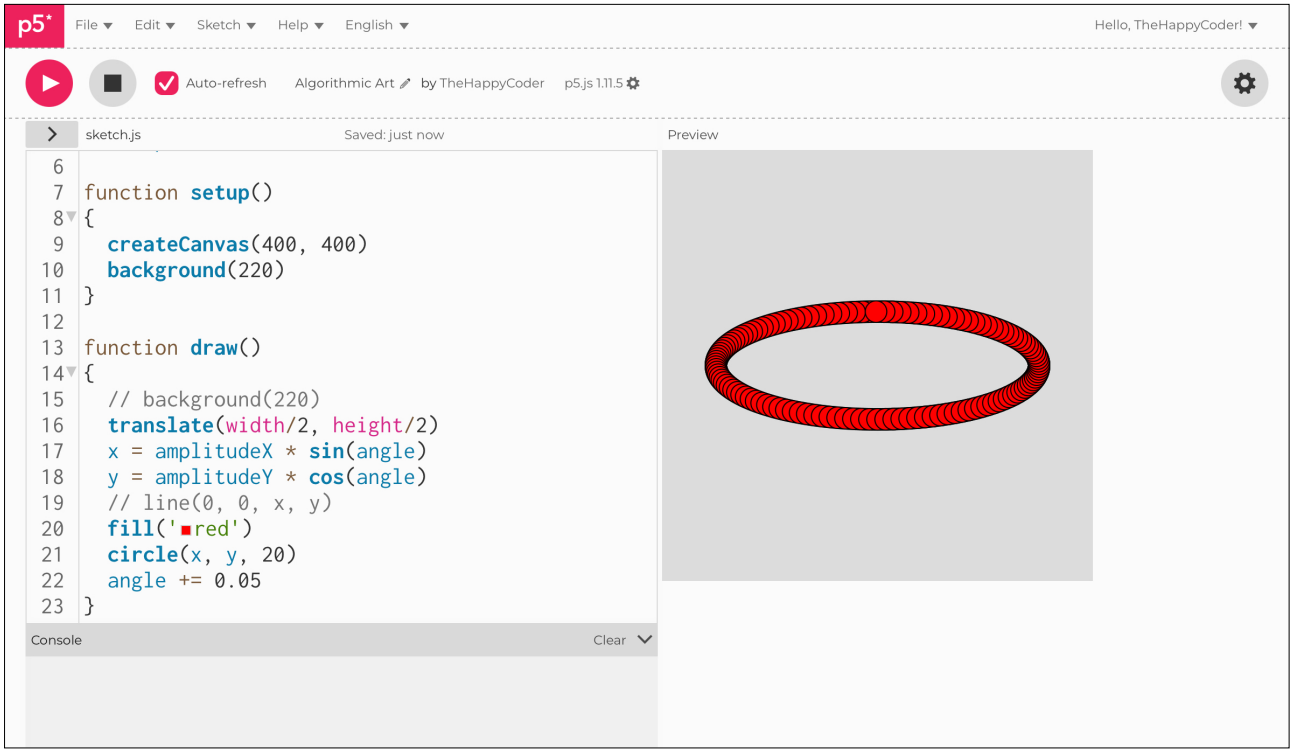
🗒️ Notes

This gives an animated response as if drawing on the canvas.

🛠️ Code Explanation

`// background(220)`	Ignores this line of code
`// line(0, 0, x, y)`	Ignores this line of code as well

Figure B2.13



Sketch B2.14 pattern (1)

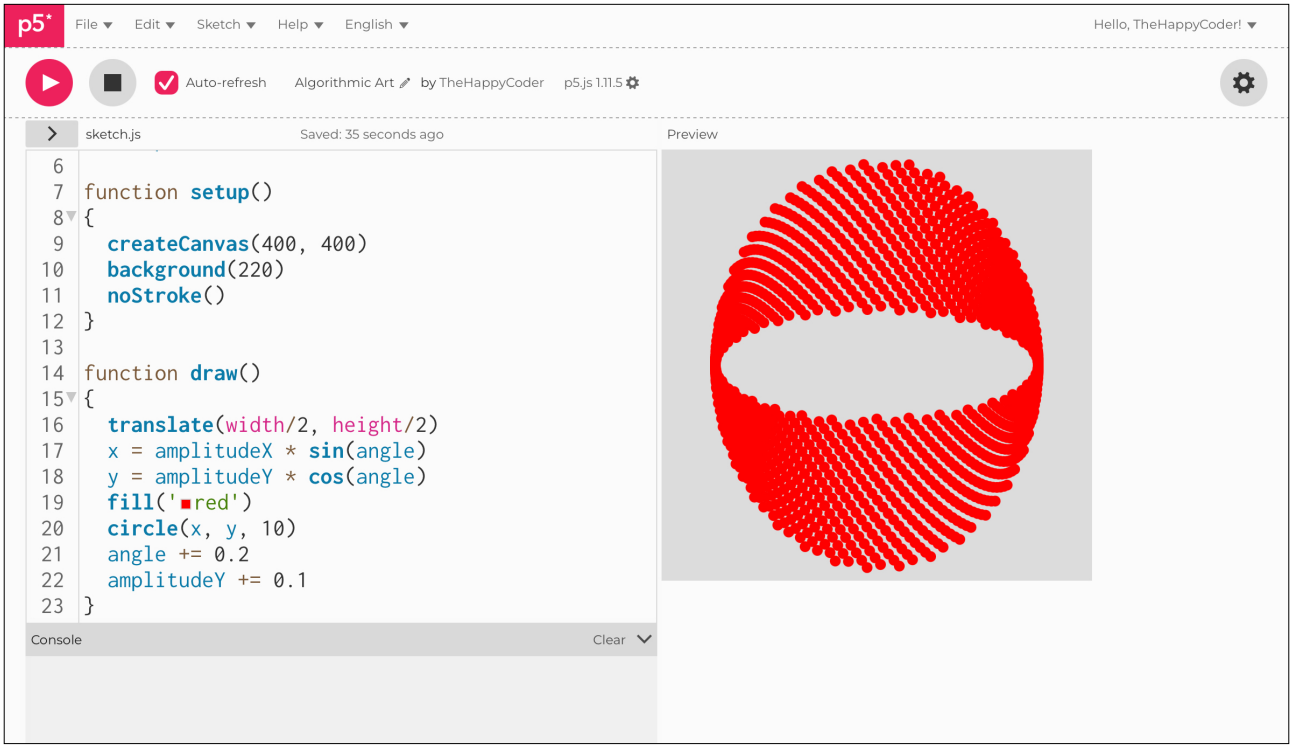
❗ I have removed the commented-out (//) lines of code. I just wanted to demonstrate the use of //. We have added a noStroke(), made the circle a bit smaller, changed the rate of angle, and most importantly, incremented the amplitude of the y component, amplitudeY.

```
let angle = 0
let x
let y
let amplitudeX = 150
let amplitudeY = 50

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

function draw()
{
  translate(width/2, height/2)
  x = amplitudeX * sin(angle)
  y = amplitudeY * cos(angle)
  fill('red')
  circle(x, y, 10)
  angle += 0.2
  amplitudeY += 0.1
}
```

🗒️ Notes

A different sort of pattern emerges.

Figure B2.14


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: 35 seconds ago
Preview
6
7 function setup()
8 {
9 createCanvas(400, 400)
10 background(220)
11 noStroke()
12 }
13
14 function draw()
15 {
16 translate(width/2, height/2)
17 x = amplitudeX * sin(angle)
18 y = amplitudeY * cos(angle)
19 fill('red')
20 circle(x, y, 10)
21 angle += 0.2
22 amplitudeY += 0.1
23 }
Console
Clear

Sketch B2.15 pattern (2)

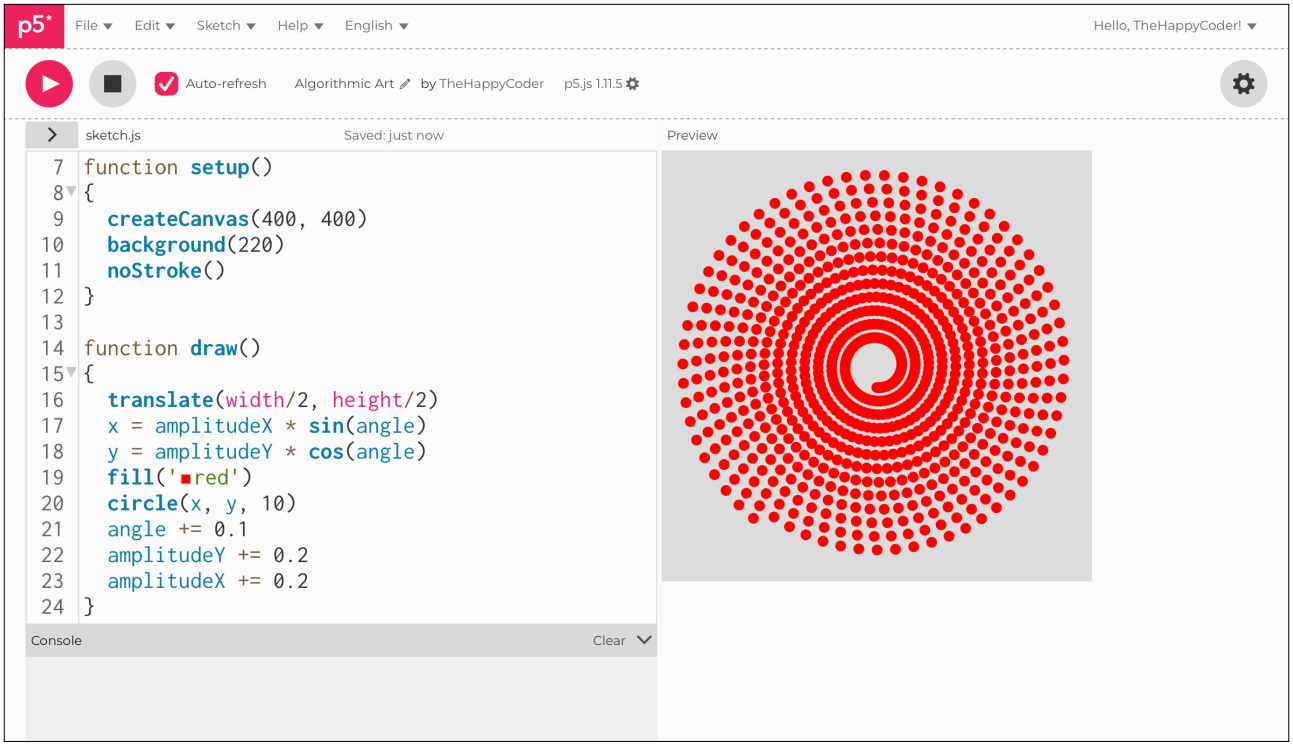
Changing the amplitude for x and y. Also, alter the starting amplitude for both of them and the angle increment.

```
let angle = 0
let x
let y
let amplitudeX = 20
let amplitudeY = 20

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

function draw()
{
  translate(width/2, height/2)
  x = amplitudeX * sin(angle)
  y = amplitudeY * cos(angle)
  fill('red')
  circle(x, y, 10)
  angle += 0.1
  amplitudeY += 0.2
  amplitudeX += 0.2
}
```

Notes

A nice spiral effect.

Figure B2.15


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
7 function setup()
8 {
9 createCanvas(400, 400)
10 background(220)
11 noStroke()
12 }
13
14 function draw()
15 {
16 translate(width/2, height/2)
17 x = amplitudeX * sin(angle)
18 y = amplitudeY * cos(angle)
19 fill('red')
20 circle(x, y, 10)
21 angle += 0.1
22 amplitudeY += 0.2
23 amplitudeX += 0.2
24 }
Console
Clear

Sketch B2.16 pattern (3)

Adding in the `diameter` variable and changing some of the other values.

```
let angle = 0
let x
let y
let amplitudeX = 0
let amplitudeY = 0
let diameter = 1

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

function draw()
{
  translate(width/2, height/2)
  x = amplitudeX * sin(angle)
  y = amplitudeY * cos(angle)
  fill('red')
  circle(x, y, diameter)
  angle += 0.5
  amplitudeY += 0.2
  amplitudeX += 0.2
  diameter += 0.01
}
```

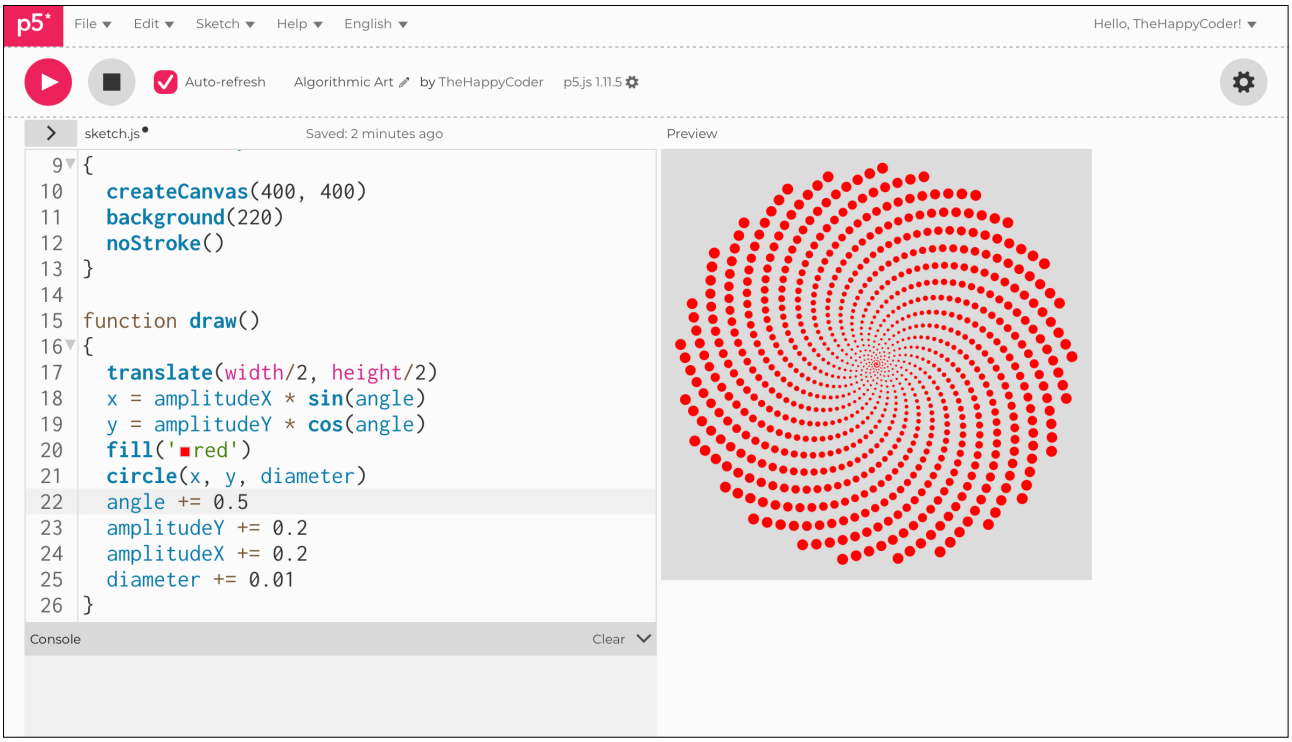
🗒️ Notes

Just shows what can be achieved with a few variables.

🌻 Challenge

The idea is to be creative and make your own designs and patterns through playing. Adding colour, other shapes and levels of alpha, etc.

Figure B2.16


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: 2 minutes ago
Preview
{
createCanvas(400, 400)
background(220)
noStroke()
}
function draw()
{
translate(width/2, height/2)
x = amplitudeX * sin(angle)
y = amplitudeY * cos(angle)
fill('red')
circle(x, y, diameter)
angle += 0.5
amplitudeY += 0.2
amplitudeX += 0.2
diameter += 0.01
}
Console
Clear