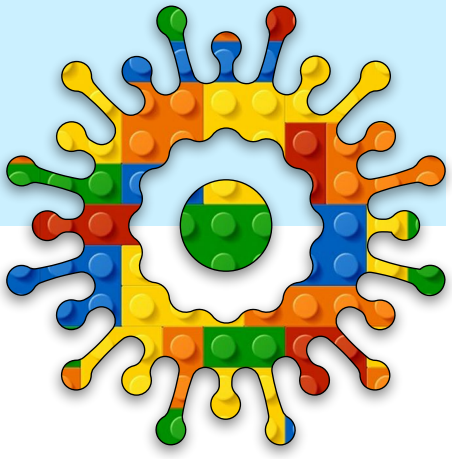
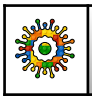
Physics Simulation Module A Unit #1 Oscillations



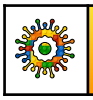


Contents

Module A Unit #1 Oscillation

Introduction to Unit #1 Oscillation

Creating a simulated spring and pendulum through exploring circular motion. It includes the introduction of polar co-ordinates, vectors, forces, etc., on a particle with a mass.



Sketch A1.1 polar co-ordinates

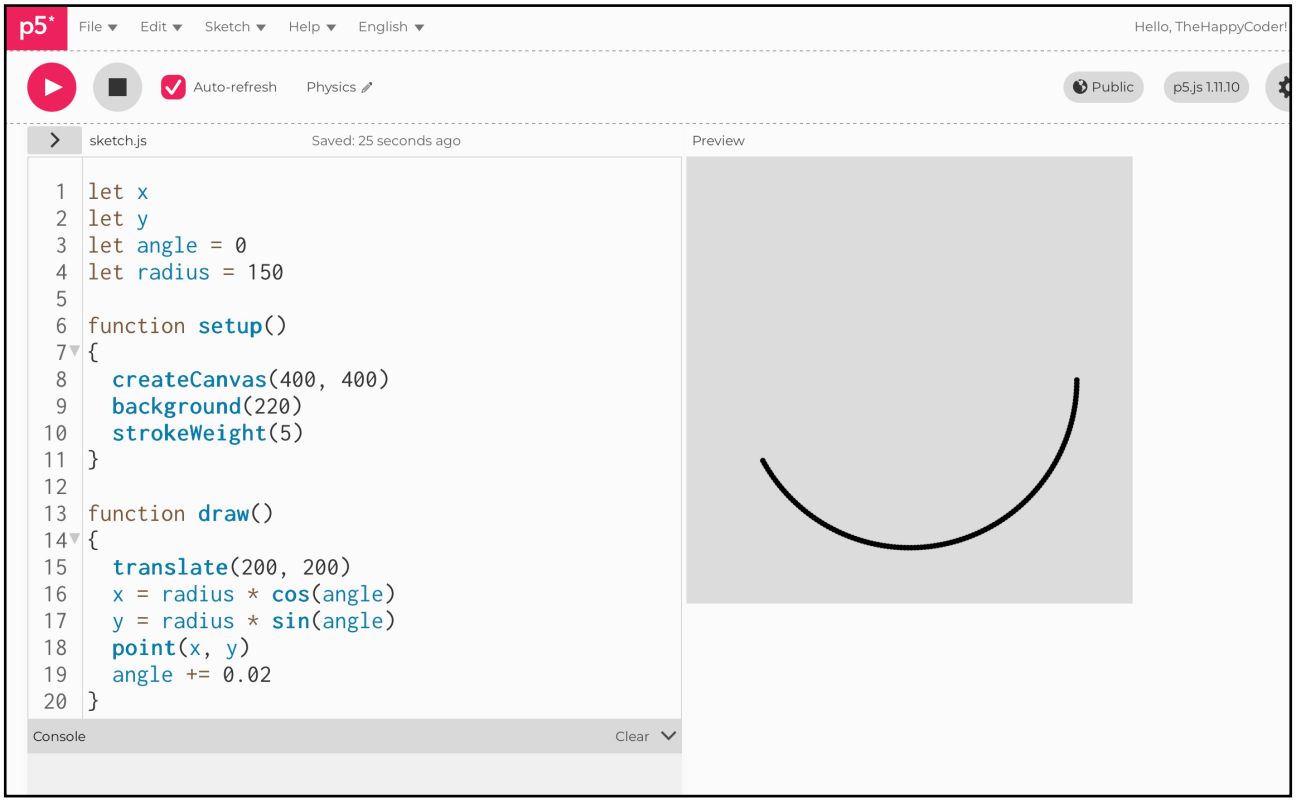
This is our starting sketch to illustrate polar to Cartesian co-ordinates - drawing the circle. We translate the co-ordinates to the centre of the canvas and use the angle (sine and cosine) to rotate round and draw a point at each 0.02 radians.

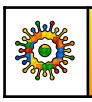
```
let x
let y
let angle = 0
let radius = 150

function setup()
{
  createCanvas(400, 400)
  background(220)
  strokeWeight(5)
}

function draw()
{
  translate(200, 200)
  x = radius * cos(angle)
  y = radius * sin(angle)
  point(x, y)
  angle += 0.02
}
```

Notes

This draws a nice circle one point at a time.

Figure A1.1


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: 25 seconds ago
Preview
let x
let y
let angle = 0
let radius = 150
function setup()
{
createCanvas(400, 400)
background(220)
strokeWeight(5)
}
function draw()
{
translate(200, 200)
x = radius * cos(angle)
y = radius * sin(angle)
point(x, y)
angle += 0.02
}
Console
Clear



Sketch A1.2 spiral

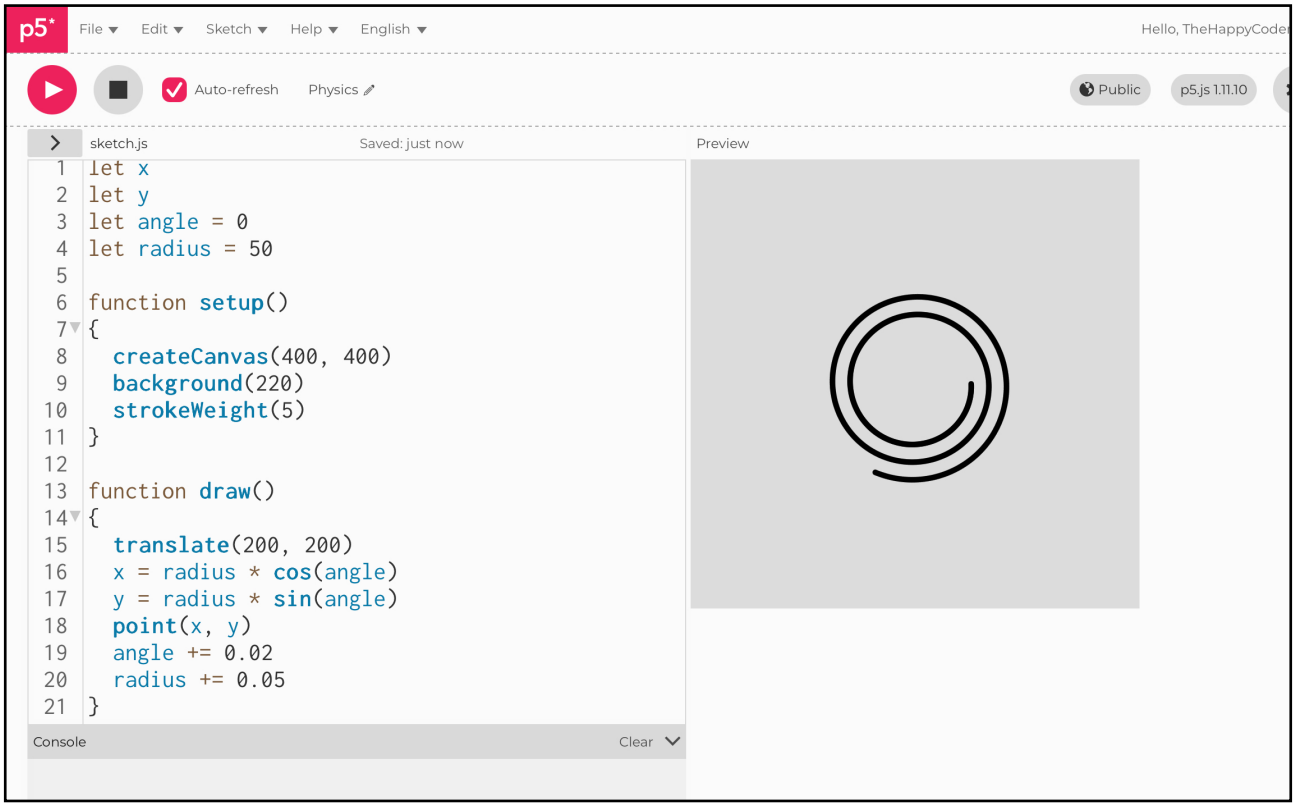
Following the same principle but this time creating a spiral by increasing the radius incrementally, starting with a slightly smaller `radius`.

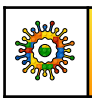
```
let x
let y
let angle = 0
let radius = 50

function setup()
{
  createCanvas(400, 400)
  background(220)
  strokeWeight(5)
}

function draw()
{
  translate(200, 200)
  x = radius * cos(angle)
  y = radius * sin(angle)
  point(x, y)
  angle += 0.02
  radius += 0.05
}
```

Notes

We now have a nice spiral emerging.

Figure A1.2


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
let x
let y
let angle = 0
let radius = 50
function setup()
{
createCanvas(400, 400)
background(220)
strokeWeight(5)
}
function draw()
{
translate(200, 200)
x = radius * cos(angle)
y = radius * sin(angle)
point(x, y)
angle += 0.02
radius += 0.05
}
Console
Clear



Sketch A1.3 vertex

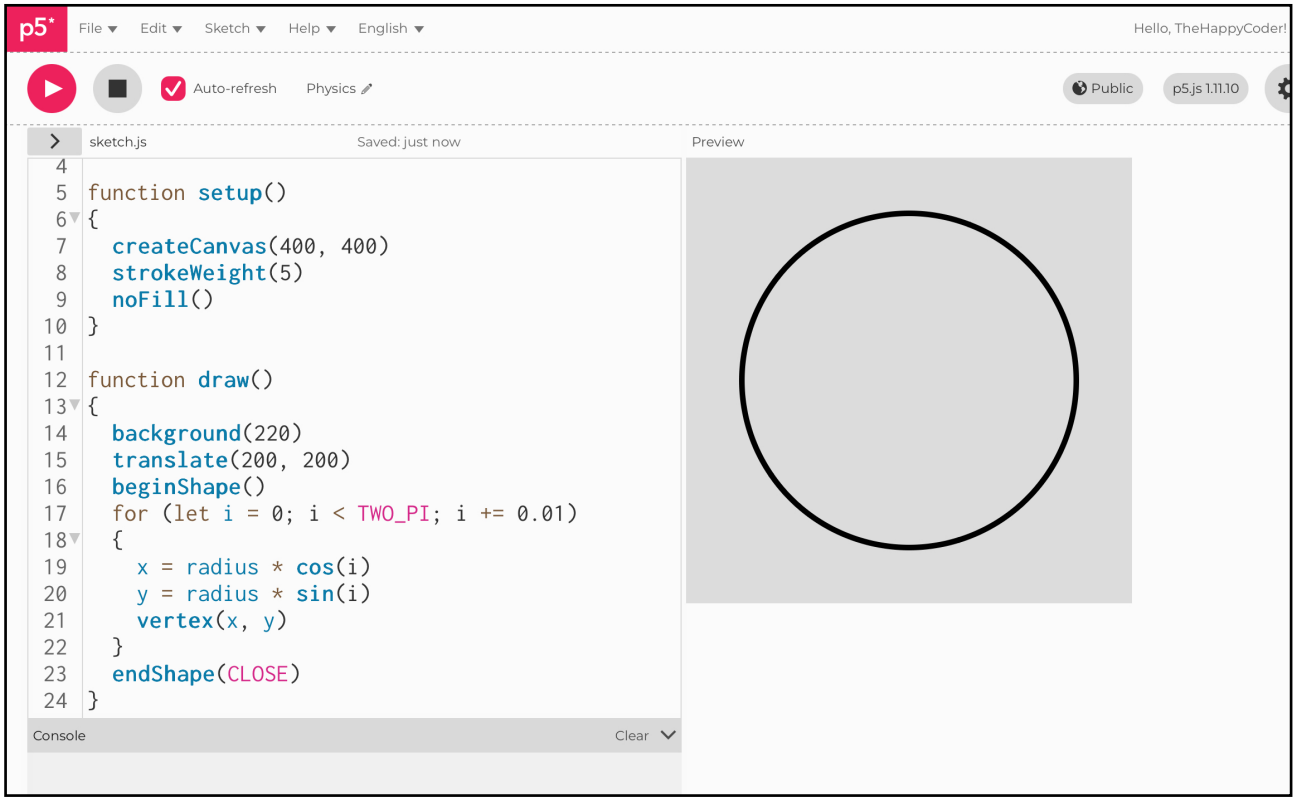
Drawing the circle by joining the points with the `vertex()` co-ordinates (changing the radius back to `150`, removing the angle, and moving the background back into `draw()` function.

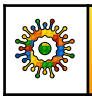
```
let x
let y
let radius = 150

function setup()
{
  createCanvas(400, 400)
  strokeWeight(5)
  noFill()
}

function draw()
{
  background(220)
  translate(200, 200)
  beginShape()
  for (let i = 0; i < TWO_PI; i += 0.01)
  {
    x = radius * cos(i)
    y = radius * sin(i)
    vertex(x, y)
  }
  endShape(CLOSE)
}
```

Notes

We have a circle but the points on the circle are joined by lines.

Figure A1.3


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
function setup()
{
createCanvas(400, 400)
strokeWeight(5)
noFill()
}
function draw()
{
background(220)
translate(200, 200)
beginShape()
for (let i = 0; i < TWO_PI; i += 0.01)
{
x = radius * cos(i)
y = radius * sin(i)
vertex(x, y)
}
endShape(CLOSE)
}
Console
Clear



Sketch A1.4 mapping the mouse

Mapping the `increment` to `mouseX`.

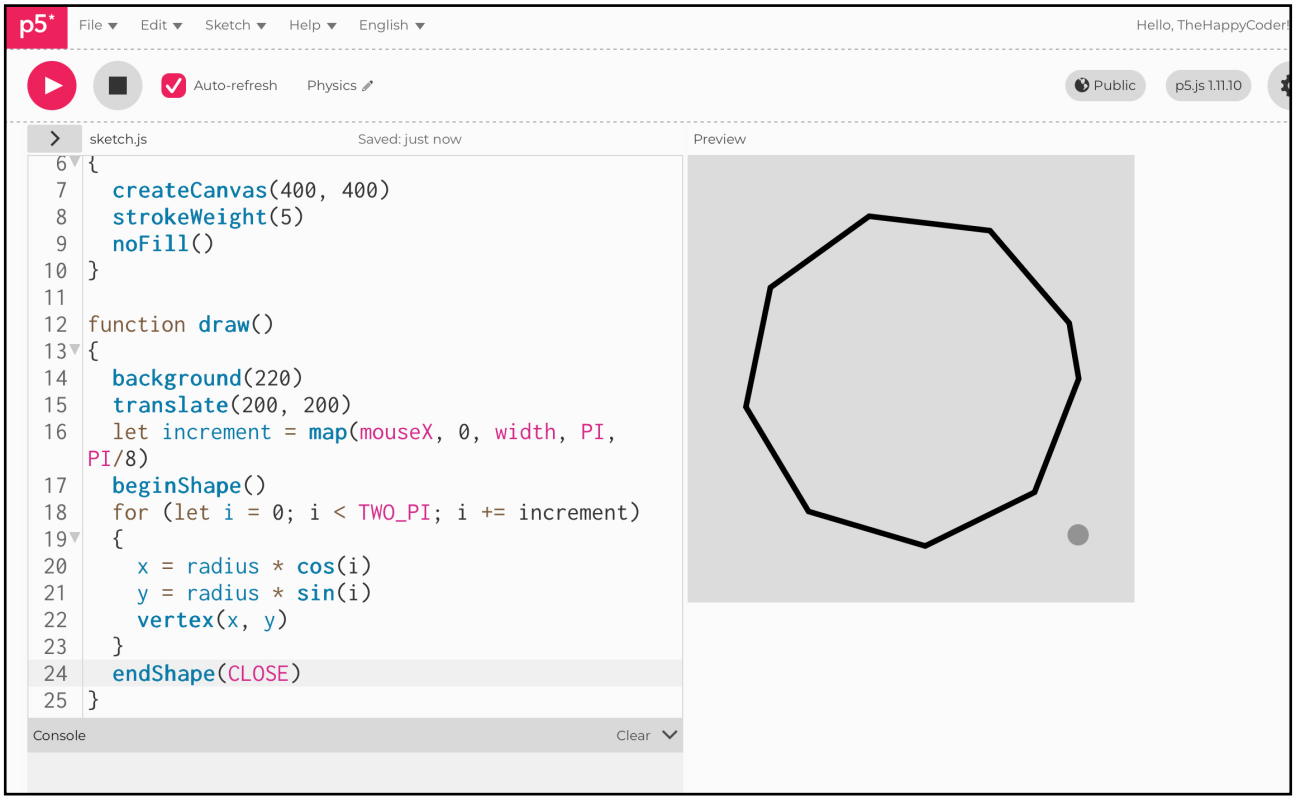
❗ There is a slight risk of an infinite loop if your mouse strays too far. Be warned, the programme will crash, so save it before going any further.

```
let x
let y
let radius = 150

function setup()
{
  createCanvas(400, 400)
  strokeWeight(5)
  noFill()
}

function draw()
{
  background(220)
  translate(200, 200)
  let increment = map(mouseX, 0, width, PI, PI/8)
  beginShape()
  for (let i = 0; i < TWO_PI; i += increment)
  {
    x = radius * cos(i)
    y = radius * sin(i)
    vertex(x, y)
  }
  endShape(CLOSE)
}
```

🗒️ Notes

As you move your mouse across the canvas the number of increments increases (the mapping takes care of this). This increases the number of points (vertices). You will start with a flat line and move through to nearly a circle.

Figure A1.4


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
{
createCanvas(400, 400)
strokeWeight(5)
noFill()
}
function draw()
{
background(220)
translate(200, 200)
let increment = map(mouseX, 0, width, PI, PI/8)
beginShape()
for (let i = 0; i < TWO_PI; i += increment)
{
x = radius * cos(i)
y = radius * sin(i)
vertex(x, y)
}
endShape(CLOSE)
}
Console
Clear

Sketch A1.5 SHM translated

❗ Start a new sketch.

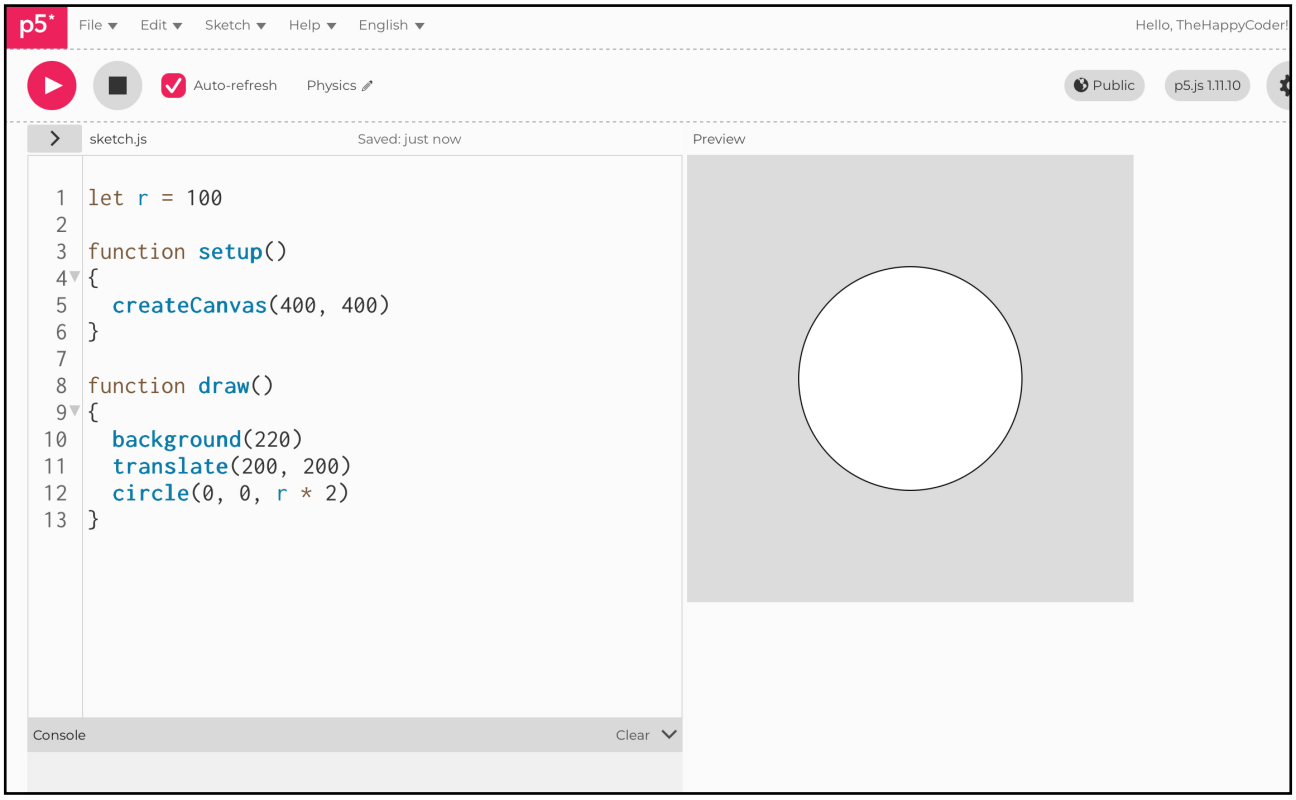
We are going to explore simple harmonic motion (SHM). The basic circle sketch with the centre translated to the middle and with a radius r of 100.

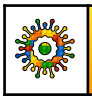
```
let r = 100

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

function draw()
{
  background(220)
  translate(200, 200)
  circle(0, 0, r * 2)
}
```

Notes

It's just a circle!

Figure A1.5


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
let r = 100
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(200, 200)
circle(0, 0, r * 2)
}
Console
Clear



Sketch A1.6 SHM breathe

Now we are going to let the circle `breathe` to the waveform of the sine wave. The sine of an angle returns a value between `-1` and `1`. To get round the problem of having a circle of radius between `-1` and `1`, we use the `map()` function so its radius is between `50` and `150`.

```
let r = 100
let angle = 0

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

function draw()
{
  background(220)
  translate(200, 200)
  r = map(sin(angle), -1, 1, 50, 150)
  circle(0, 0, r * 2)
  angle += 0.05
}
```

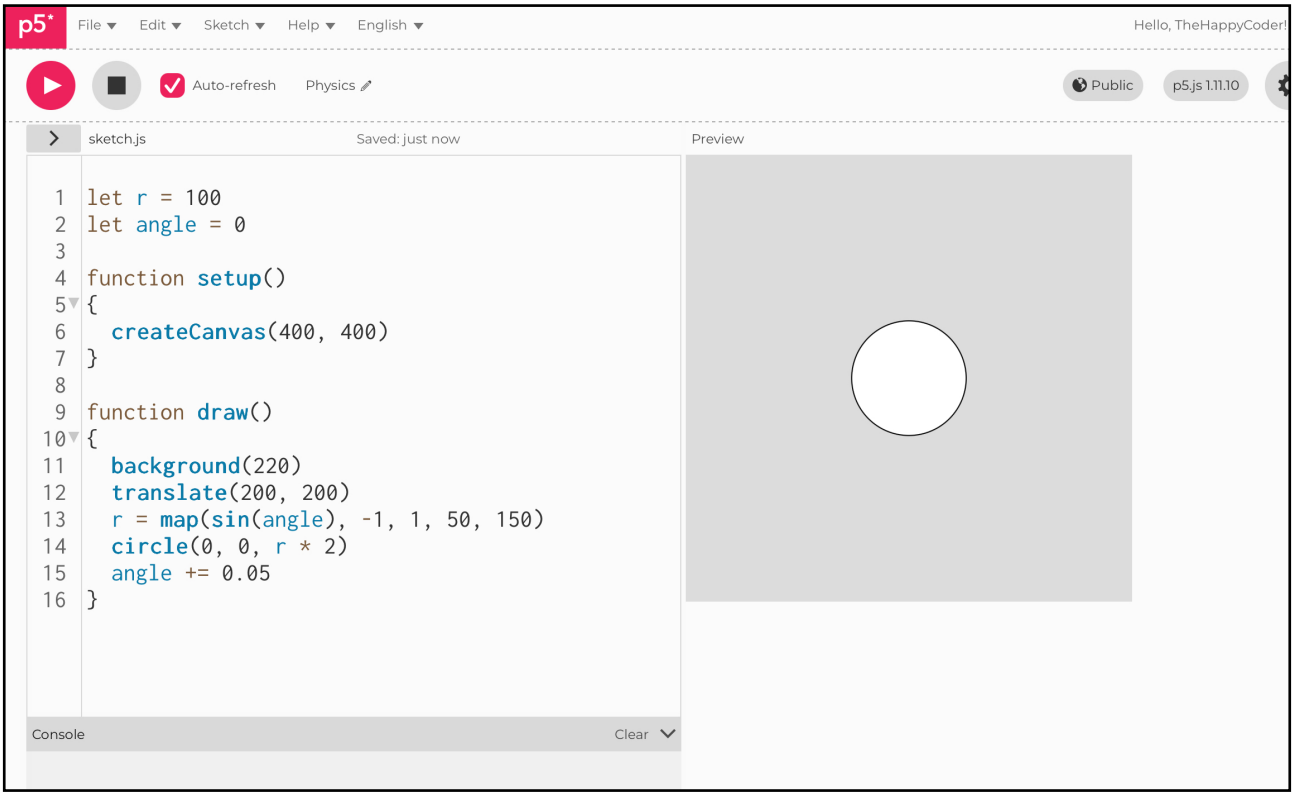
📝 Notes

You should see it inflate and deflate.

🌻 Challenges

1. Try different angle increments.
2. Different mapping values.
3. Colour: `fill(r * 2, 200 - r, 200)`

Figure A1.6


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
let r = 100
let angle = 0
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(200, 200)
r = map(sin(angle), -1, 1, 50, 150)
circle(0, 0, r * 2)
angle += 0.05
}
Console
Clear

Sketch A1.7 SHM period

Introducing a period, how long is the cycle of one sine wave? It is 360° or 2π.
The frame rate of your screen is around 60 frames per second; you can check this by including the line of code `console.log(frameRate())` in the `draw()` function. To do the maths so that we have 1 wave per second, we divide 360/60, which will give us the amount of rotation per frame to make a couple of cycles in 1 second, which comes out at 6°. Phew! In the sketch, we will continue to use radians, hence 2π/60 (TWO_PI/60).

```
let r = 100
let angle = 0

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

function draw()
{
  background(220)
  translate(200, 200)
  r = map(sin(angle), -1, 1, 50, 150)
  circle(0, 0, r * 2)
  let increment = TWO_PI/60
  angle += increment
}
```

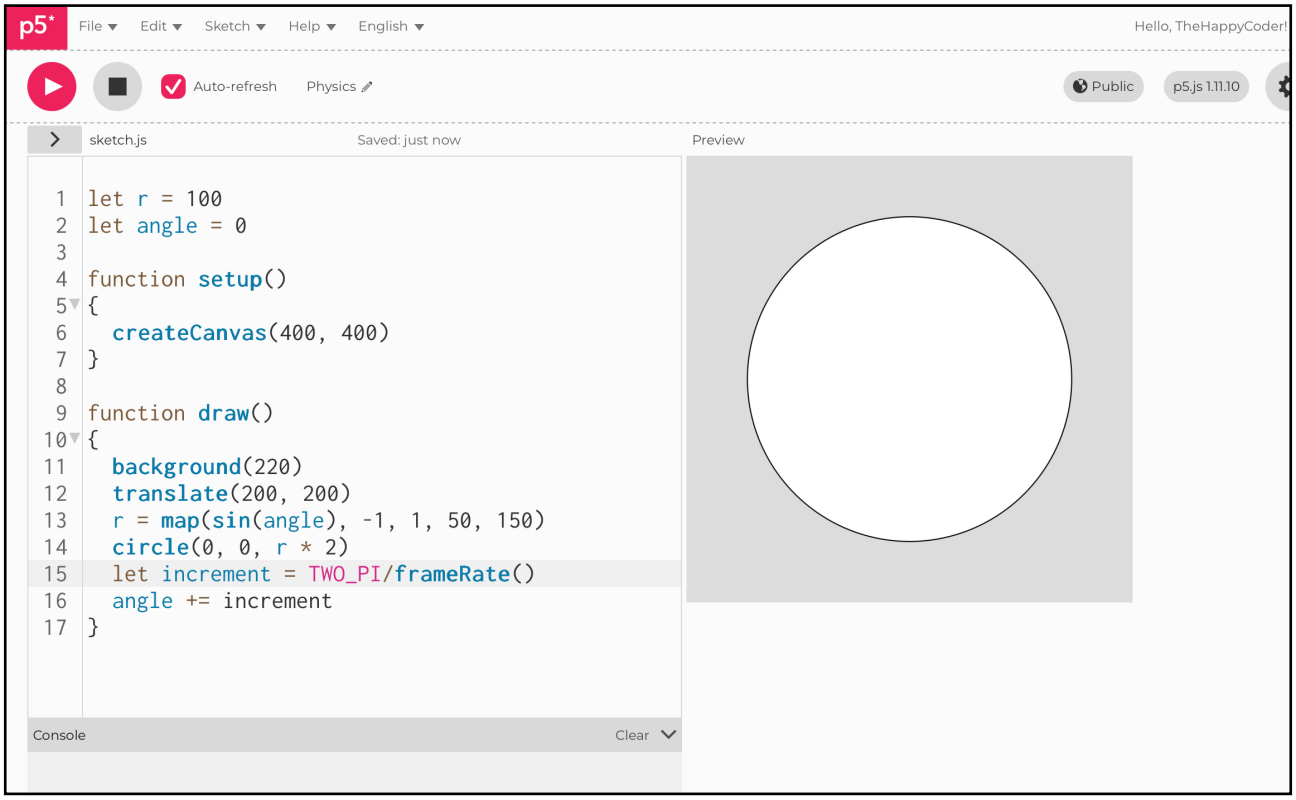
Notes

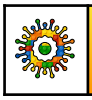
This is just another way of simulating movement, we are simulating this digitally.

🌻 Challenges

1. Change the line of code to: `let increment = TWO_PI/frameRate()`
2. Change the time period using `millis()`.
3. Add an angular velocity so it gets faster over time.
4. Try it in 3D.

Figure A1.7


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
let r = 100
let angle = 0
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(200, 200)
r = map(sin(angle), -1, 1, 50, 150)
circle(0, 0, r * 2)
let increment = TWO_PI/frameRate()
angle += increment
}
Console
Clear



Sketch A1.8 graphing sine wave

❗ Start a new sketch for this bit.

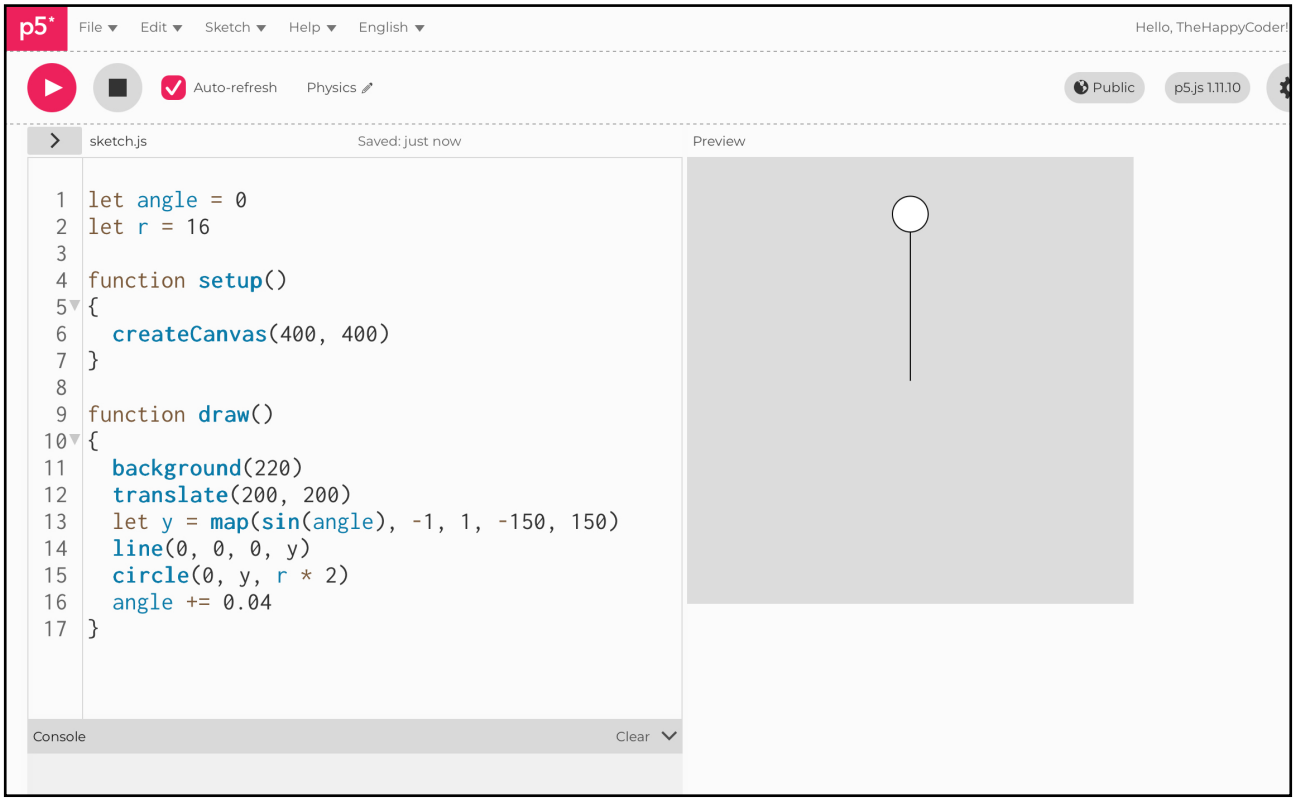
Creating an oscillating sphere. A simple oscillating sphere attached to an elastic string.

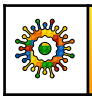
```
let angle = 0
let r = 16

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

function draw()
{
  background(220)
  translate(200, 200)
  let y = map(sin(angle), -1, 1, -150, 150)
  line(0, 0, 0, y)
  circle(0, y, r * 2)
  angle += 0.04
}
```

Notes

A bit like an everlasting yo-yo.

Figure A1.8


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
let angle = 0
let r = 16
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(200, 200)
let y = map(sin(angle), -1, 1, -150, 150)
line(0, 0, 0, y)
circle(0, y, r * 2)
angle += 0.04
}
Console
Clear



Sketch A1.9 an array of spheres

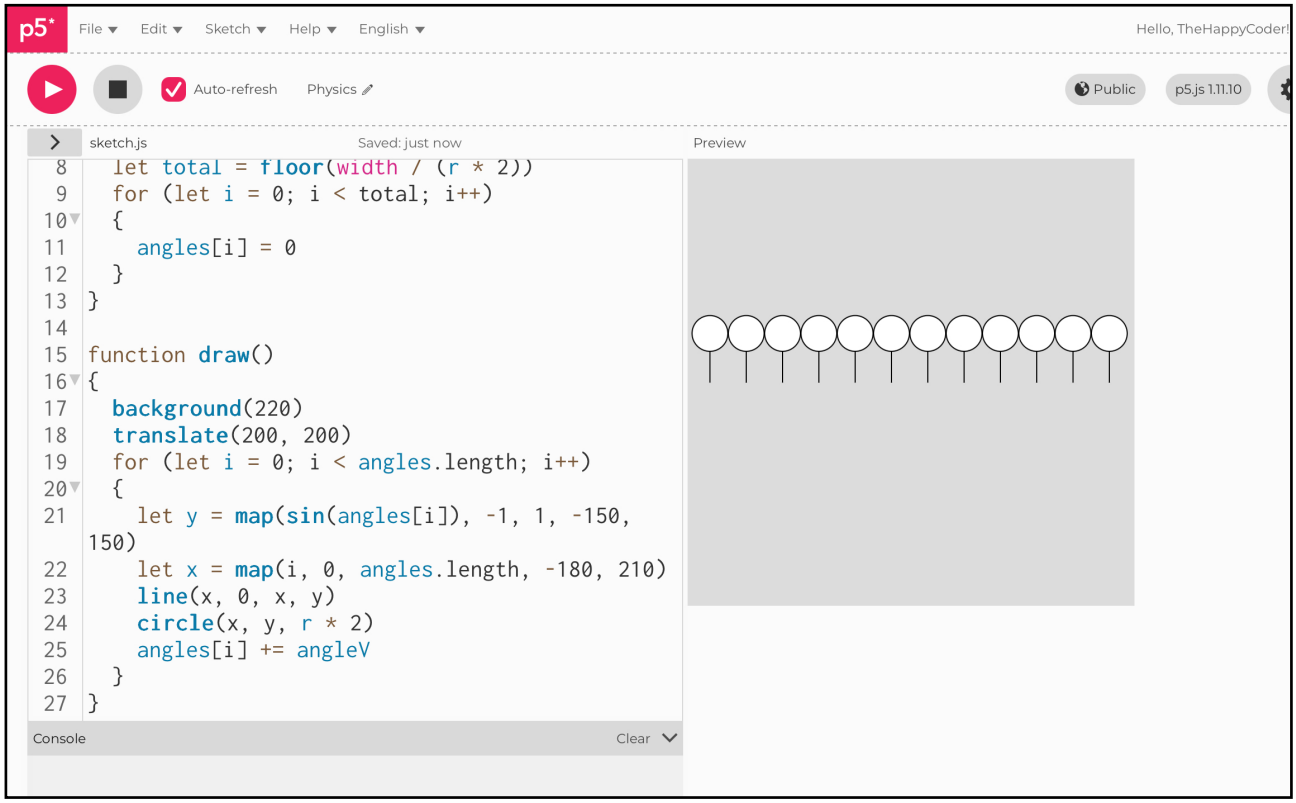
Adding in an array of spheres, adding lines to each one, and adjusting so it fits nicely on the canvas. The `floor()` function returns an integer.

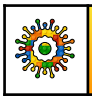
```
let angles = []
let angleV = 0.04
let r = 16

function setup()
{
  createCanvas(400, 400)
  let total = floor(width / (r * 2))
  for (let i = 0; i < total; i++)
  {
    angles[i] = 0
  }
}

function draw()
{
  background(220)
  translate(200, 200)
  for (let i = 0; i < angles.length; i++)
  {
    let y = map(sin(angles[i]), -1, 1, -150, 150)
    let x = map(i, 0, angles.length, -180, 210)
    line(x, 0, x, y)
    circle(x, y, r * 2)
    angles[i] += angleV
  }
}
```

🗒️ Notes

Lots of yo-yos, all moving in unison.

Figure A1.9


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
let total = floor(width / (r * 2))
for (let i = 0; i < total; i++)
{
angles[i] = 0
}
}
function draw()
{
background(220)
translate(200, 200)
for (let i = 0; i < angles.length; i++)
{
let y = map(sin(angles[i]), -1, 1, -150, 150)
let x = map(i, 0, angles.length, -180, 210)
line(x, 0, x, y)
circle(x, y, r * 2)
angles[i] += angleV
}
}
Console
Clear



Sketch A1.10 a random value

Make an array of angles set to a random value between -0.1 and 0.1.

```
let angles = []
let angleV = []
let r = 16

function setup()
{
  createCanvas(400, 400)
  let total = floor(width / (r * 2))
  for (let i = 0; i < total; i++)
  {
    angles[i] = 0
    angleV[i] = random(-0.1, 0.1)
  }
}

function draw()
{
  background(220)
  translate(200, 200)
  for (let i = 0; i < angles.length; i++)
  {
    let y = map(sin(angles[i]), -1, 1, -150, 150)
    let x = map(i, 0, angles.length, -180, 210)
    line(x, 0, x, y)
    circle(x, y, r * 2)
    angles[i] += angleV[i]
  }
}
```

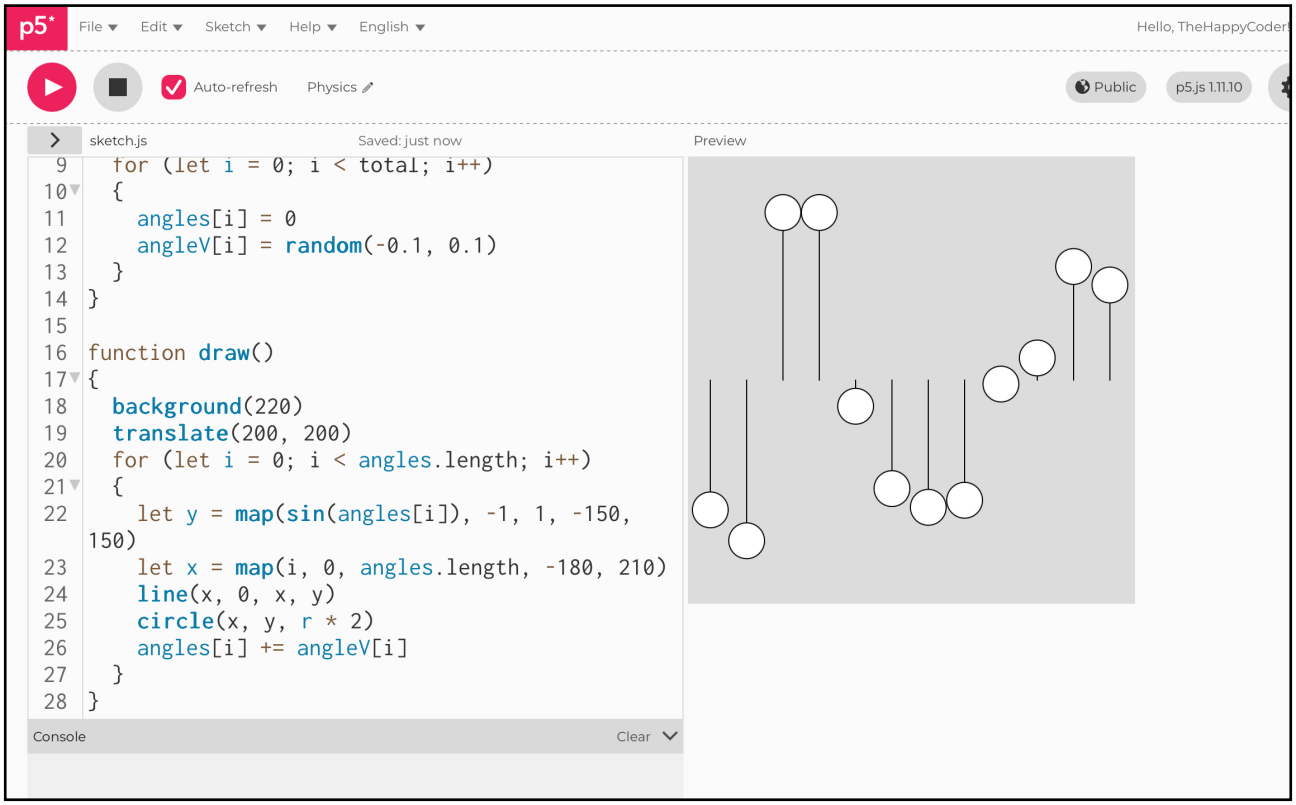
🗒️ Notes

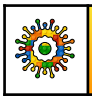
Not so unison, randomly phased

🌻 Challenge

1. Change `angleV[i] = random(-0.1, 0.1)` to `angleV[i] = i / 100` and keep it for the next sketch.
2. Play around with the values.

Figure A1.10


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
for (let i = 0; i < total; i++)
{
angles[i] = 0
angleV[i] = random(-0.1, 0.1)
}
}
function draw()
{
background(220)
translate(200, 200)
for (let i = 0; i < angles.length; i++)
{
let y = map(sin(angles[i]), -1, 1, -150, 150)
let x = map(i, 0, angles.length, -180, 210)
line(x, 0, x, y)
circle(x, y, r * 2)
angles[i] += angleV[i]
}
}
Console
Clear



Sketch A1.11 smaller and symmetrical

Removing the lines and making the spheres smaller `10`, adding vertex lines to join the circles. Remove: `line(x, 0, x, y)`

```
let angles = []
let angleV = []
let r = 10

function setup()
{
  createCanvas(400, 400)
  let total = floor(width / (r * 2))
  for (let i = 0; i < total; i++)
  {
    angles[i] = map(i, 0, total, 0, TWO_PI)
    angleV[i] = 0.04
  }
}

function draw()
{
  background(220)
  translate(200, 200)
  beginShape()
  for (let i = 0; i < angles.length; i++)
  {
    let y = map(sin(angles[i]), -1, 1, -150, 150)
    let x = map(i, 0, angles.length, -300, 300)
    fill(255)
    circle(x, y, r * 2)
    noFill()
```

```
    vertex(x, y)
    angles[i] += angleV[i]
  }
  endShape()
}
```

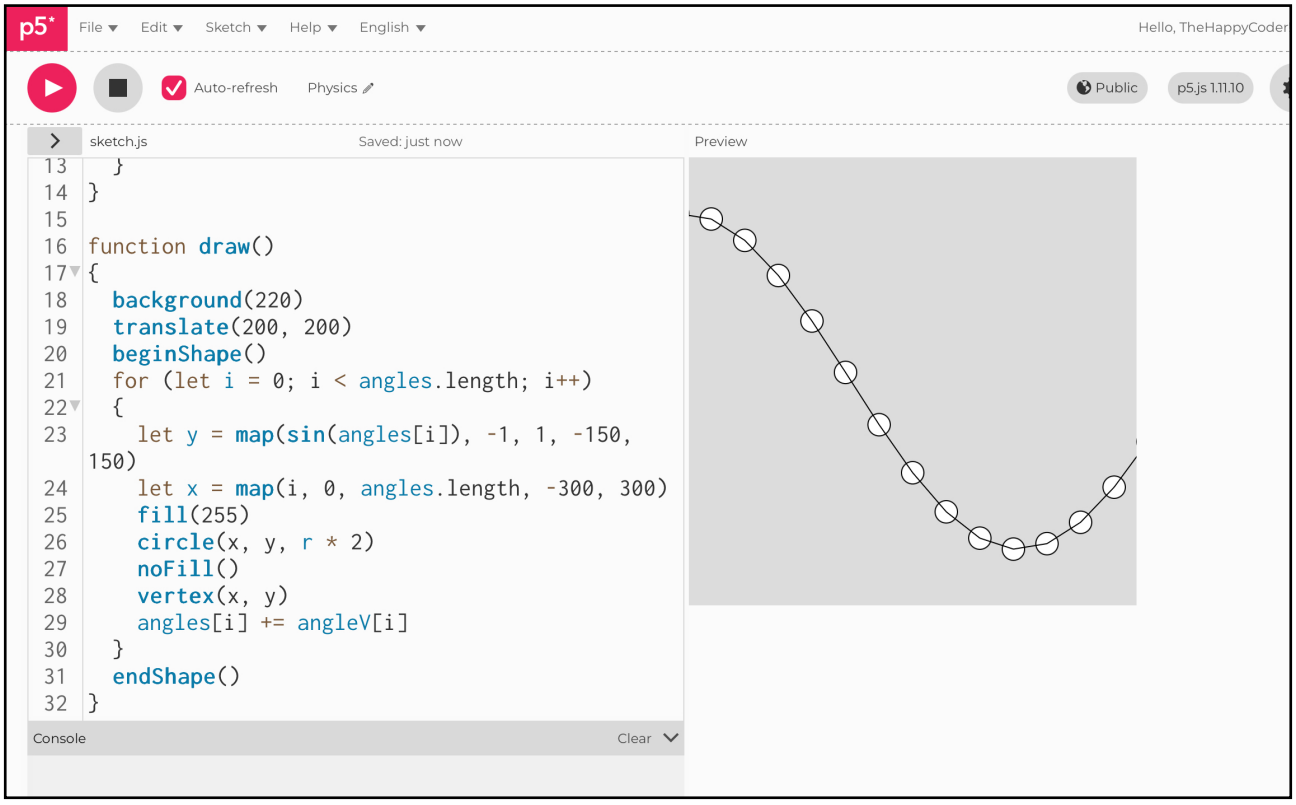
🗒️ Notes

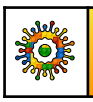
We have a bit of a sine wave

🌻 Challenge

Replace `angleV[i] = 0.04` with `angleV[i] = random(0.04)`.

Figure A1.11


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
13 }
14 }
15
16 function draw()
17 {
18 background(220)
19 translate(200, 200)
20 beginShape()
21 for (let i = 0; i < angles.length; i++)
22 {
23 let y = map(sin(angles[i]), -1, 1, -150, 150)
24 let x = map(i, 0, angles.length, -300, 300)
25 fill(255)
26 circle(x, y, r * 2)
27 noFill()
28 vertex(x, y)
29 angles[i] += angleV[i]
30 }
31 endShape()
32 }
Console
Clear



Sketch A1.12 amplitude, period and phase

❗ Start a new sketch.

There are three parts to a wave: amplitude (height of the wave), period (length of one wave), and phase (offset). In this sketch, we are going to add waves together. First, using the final sketch from the last example, we will make it object-orientated.

```
let wave

function setup()
{
  createCanvas(400, 400)
  wave = new Wave(150, 400, 0)
}

function draw()
{
  background(220)
  for (let x = 0; x < width; x += 10)

  {
    let y = wave.calculate(x)
    circle(x, y + height/2, 10)
  }
}

class Wave
{
  constructor(amp, period, phase)
  {
    this.amplitude = amp
    this.period = period
```

```
    this.phase = phase
  }

  calculate(x)
  {
    return sin(this.phase + TWO_PI*x/this.period)*this.amplitude
  }
}
```

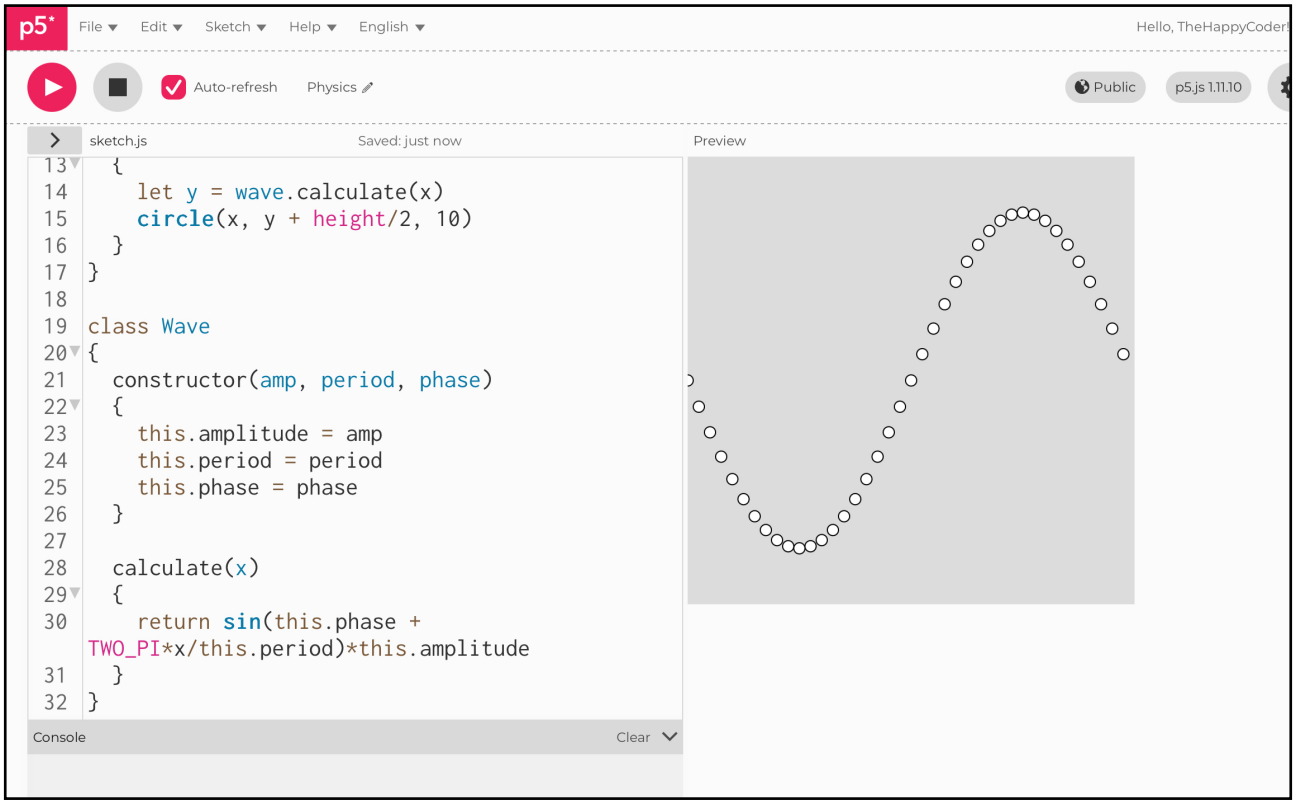
🗒️ Notes

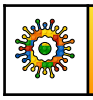
More line a full sine wave

🌻 Challenge

Try different values for amplitude, period, and phase:
e.g. `wave = new Wave(100, 300, PI/3)`.

Figure A1.12


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
{
let y = wave.calculate(x)
circle(x, y + height/2, 10)
}
}
class Wave
{
constructor(amp, period, phase)
{
this.amplitude = amp
this.period = period
this.phase = phase
}
calculate(x)
{
return sin(this.phase +
TWO_PI*x/this.period)*this.amplitude
}
}
Console
Clear



Sketch A1.13 five waves

Here we will introduce five random waves, we create them in the `setup()` function, each with a different `amplitude`, `period`, and `phase`.

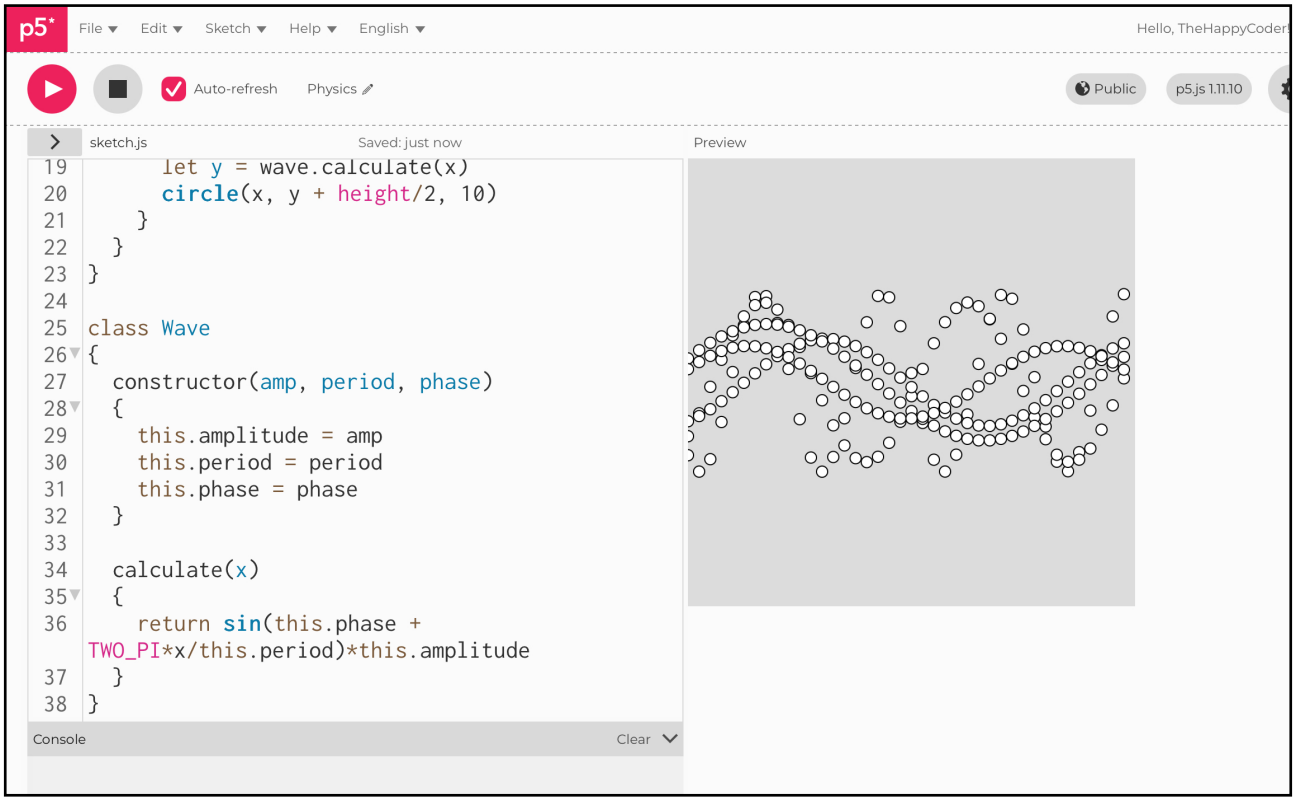
```
let waves = []


function setup()
{
  createCanvas(400, 400)
  for (let i = 0; i < 5; i++)
  {
    waves[i] = new Wave(random(20, 80), random(100, 400),
random(TWO_PI))
  }
}


function draw()
{
  background(220)
  for (let x = 0; x < width; x += 10)
  {
    for (let wave of waves)
    {
      let y = wave.calculate(x)
      circle(x, y + height/2, 10)
    }
  }
}


class Wave
{
  constructor(amp, period, phase)
```

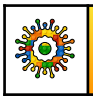
```
  {
    this.amplitude = amp
    this.period = period
    this.phase = phase
  }

  calculate(x)
  {
    return sin(this.phase + TWO_PI*x/this.period)*this.amplitude
  }
}
```

Notes

This looks a bit messy, but wait.

Figure A1.13


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
let y = wave.calculate(x)
circle(x, y + height/2, 10)
}
}
}
class Wave
{
constructor(amp, period, phase)
{
this.amplitude = amp
this.period = period
this.phase = phase
}
calculate(x)
{
return sin(this.phase +
TWO_PI*x/this.period)*this.amplitude
}
}
Console
Clear



Sketch A1.14 adding waves

Now adding them together, moving the drawing of the circle so it is drawn after adding the waves together.

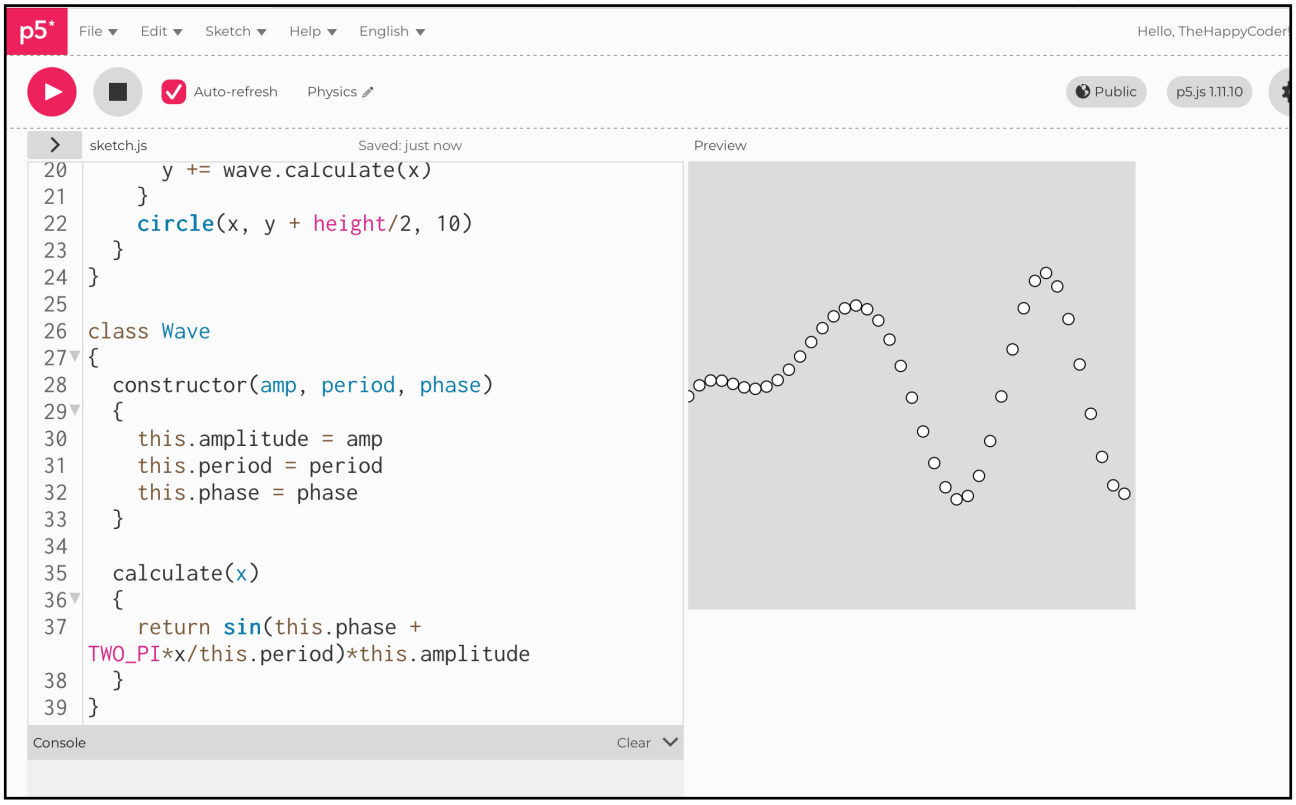
```
let waves = []


function setup()
{
  createCanvas(400, 400)
  for (let i = 0; i < 5; i++)
  {
    waves[i] = new Wave(random(20, 80), random(100, 400),
random(TWO_PI))
  }
}


function draw()
{
  background(220)
  for (let x = 0; x < width; x += 10)
  {
    let y = 0
    for (let wave of waves)
    {
      y += wave.calculate(x)
    }
    circle(x, y + height/2, 10)
  }
}


class Wave
{
```

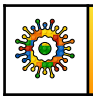
```
  constructor(amp, period, phase)
  {
    this.amplitude = amp
    this.period = period
    this.phase = phase
  }

  calculate(x)
  {
    return sin(this.phase + TWO_PI*x/this.period)*this.amplitude
  }
}
```

🗒️ Notes

Every time you refresh the sketch you will get something very different.

Figure A1.14


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
y += wave.calculate(x)
}
circle(x, y + height/2, 10)
}
}
class Wave
{
constructor(amp, period, phase)
{
this.amplitude = amp
this.period = period
this.phase = phase
}
calculate(x)
{
return sin(this.phase +
TWO_PI*x/this.period)*this.amplitude
}
}
Console
Clear



Sketch A1.15 using phase

Getting it in motion by using the phase.

```
let waves = []


function setup()
{
  createCanvas(400, 400)
  for (let i = 0; i < 5; i++)
  {
    waves[i] = new Wave(random(20, 80), random(100, 400),
random(TWO_PI))
  }
}


function draw()
{
  background(220)
  for (let x = 0; x < width; x += 10)
  {
    let y = 0
    for (let wave of waves)
    {
      y += wave.calculate(x)
    }
    circle(x, y + height/2, 10)
  }
  for (let wave of waves)
  {
    wave.phase += 0.1
  }
}
```

```
class Wave
{
  constructor(amp, period, phase)
  {
    this.amplitude = amp
    this.period = period
    this.phase = phase
  }

  calculate(x)
  {
    return sin(this.phase + TWO_PI*x/this.period)*this.amplitude
  }
}
```

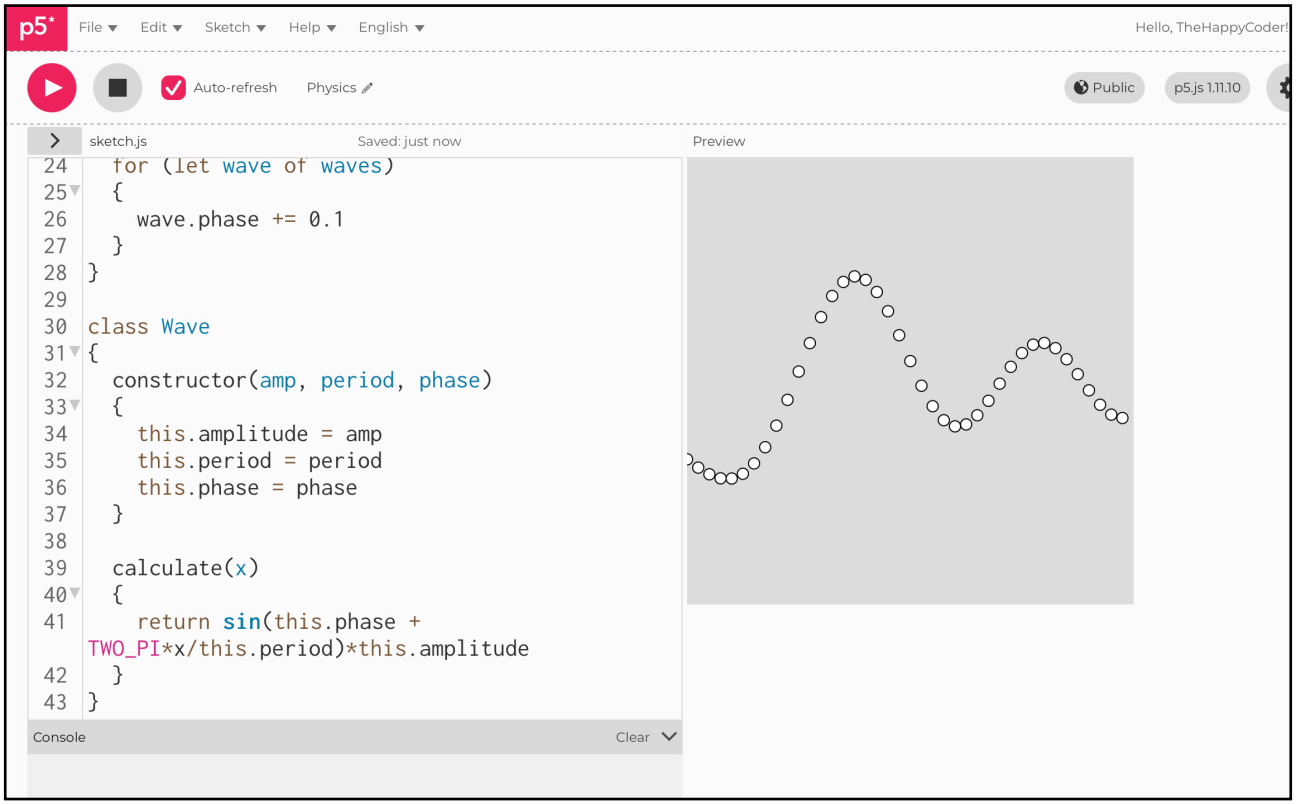
🗒️ Notes

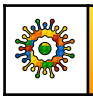
This creates a moving wave

🌻 Challenges

1. Add more waves.
2. Add colour and alpha.
3. Change the values for period, amplitude, and phase.
4. Use vertex to draw lines in between.

Figure A1.15


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: just now
Preview
for (let wave of waves)
{
wave.phase += 0.1
}
}
class Wave
{
constructor(amp, period, phase)
{
this.amplitude = amp
this.period = period
this.phase = phase
}
calculate(x)
{
return sin(this.phase +
TWO_PI*x/this.period)*this.amplitude
}
}
Console
Clear



Sketch A1.16 a few tweaks

For clarity, this is an updated version of the previous sketch. The following changes: add `update()` to the class and use it at the end of `draw()`.

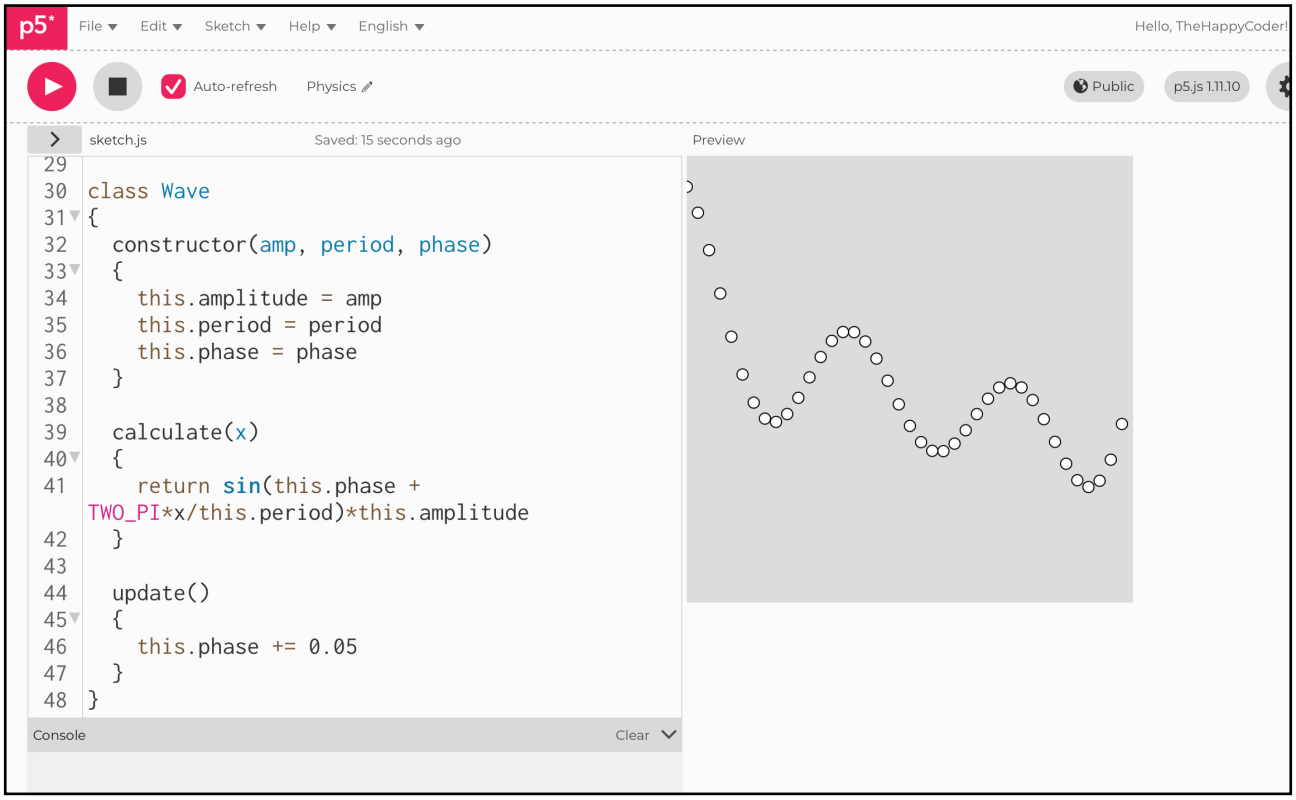
```
let waves = []


function setup()
{
  createCanvas(400, 400)
  for (let i = 0; i < 5; i++)
  {
    waves[i] = new Wave(random(20, 80), random(100, 400),
random(TWO_PI))
  }
}


function draw()
{
  background(220)
  for (let x = 0; x < width; x += 10)
  {
    let y = 0
    for (let wave of waves)
    {
      y += wave.calculate(x)
    }
    circle(x, y + height/2, 10)
  }
  for (let wave of waves)
  {
    wave.update()
  }
```

```
}

class Wave
{
  constructor(amp, period, phase)
  {
    this.amplitude = amp
    this.period = period
    this.phase = phase
  }

  calculate(x)
  {
    return sin(this.phase + TWO_PI*x/this.period)*this.amplitude
  }

  update()
  {
    this.phase += 0.05
  }
}
```

🗒️ Notes

A simple tweak.

Figure A1.16


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Physics
Public
p5.js 1.11.10
sketch.js
Saved: 15 seconds ago
Preview
29
30 class Wave
31 {
32 constructor(amp, period, phase)
33 {
34 this.amplitude = amp
35 this.period = period
36 this.phase = phase
37 }
38
39 calculate(x)
40 {
41 return sin(this.phase +
TWO_PI*x/this.period)*this.amplitude
42 }
43
44 update()
45 {
46 this.phase += 0.05
47 }
48 }
Console
Clear