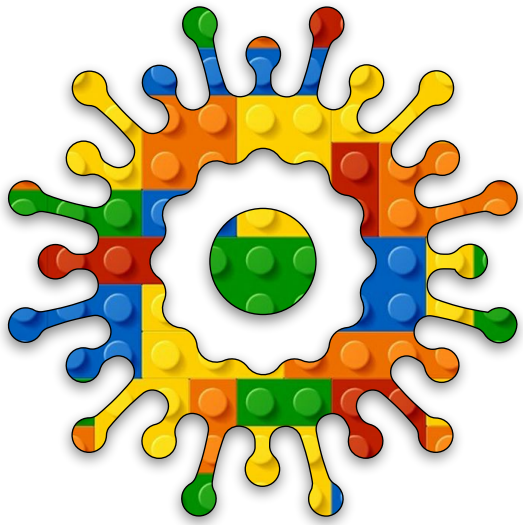
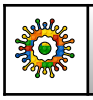
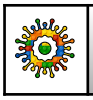
Creative Coding Module A Unit #6 ellipses and triangles





Content

Module A Unit #6 ellipses and triangles



Introduction to ellipses and triangles

As well as introducing two new shapes, we will also explore how to interact with shapes with the mouse. We will be able to move and stretch these shapes, offering more opportunities to create interesting designs that are more than static images.

Key elements

- ellipse()
- triangle()
- mouseX
- mouseY
- move
- stretch

Sketch A6.1 drawing an ellipse

We start with a new sketch with an ellipse. The `ellipse()` function has four arguments: the x and y coordinates, and the width and height of the ellipse.

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

function draw()
{
  background(220)
  ellipse(200, 200, 300, 100)
}
```

🗒️ Notes

If you use only three dimensions, you will draw a circle.

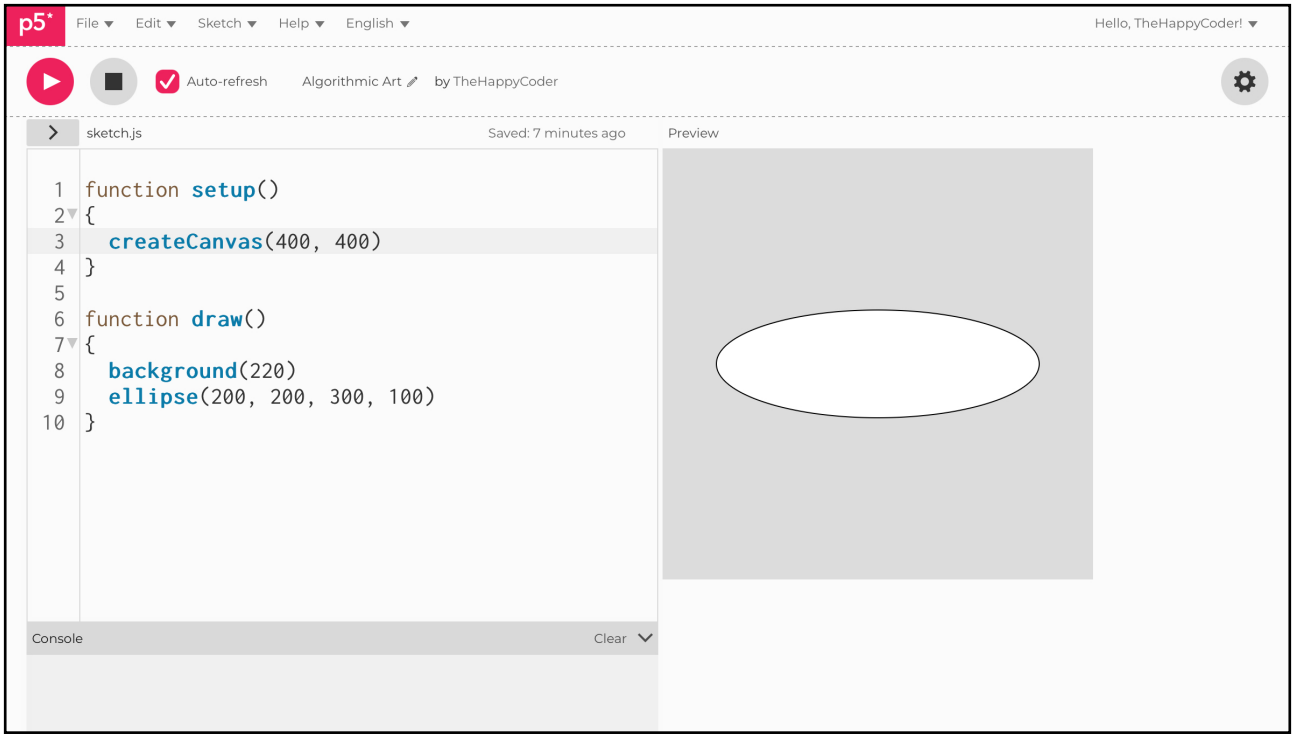
🌻 Challenge

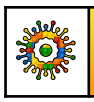
Try other dimensions for the height and width.

🛠️ Code Explanation

`ellipse(200, 200, 300, 100)`	We have an ellipse at the centre of the canvas (200, 200) and a width of 300 and height of 100

Figure A6.1


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: 7 minutes ago
Preview
1 function setup()
2 {
3 createCanvas(400, 400)
4 }
5
6 function draw()
7 {
8 background(220)
9 ellipse(200, 200, 300, 100)
10 }
Console
Clear



Sketch A6.2 mouseX and mouseY

We can use some built-in variables; one set of variables is called mouseX and mouseY. They return the coordinates of the mouse cursor position on the canvas. We can illustrate it with our ellipse.

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

function draw()
{
  background(220)
  ellipse(mouseX, mouseY, 300, 100)
}
```

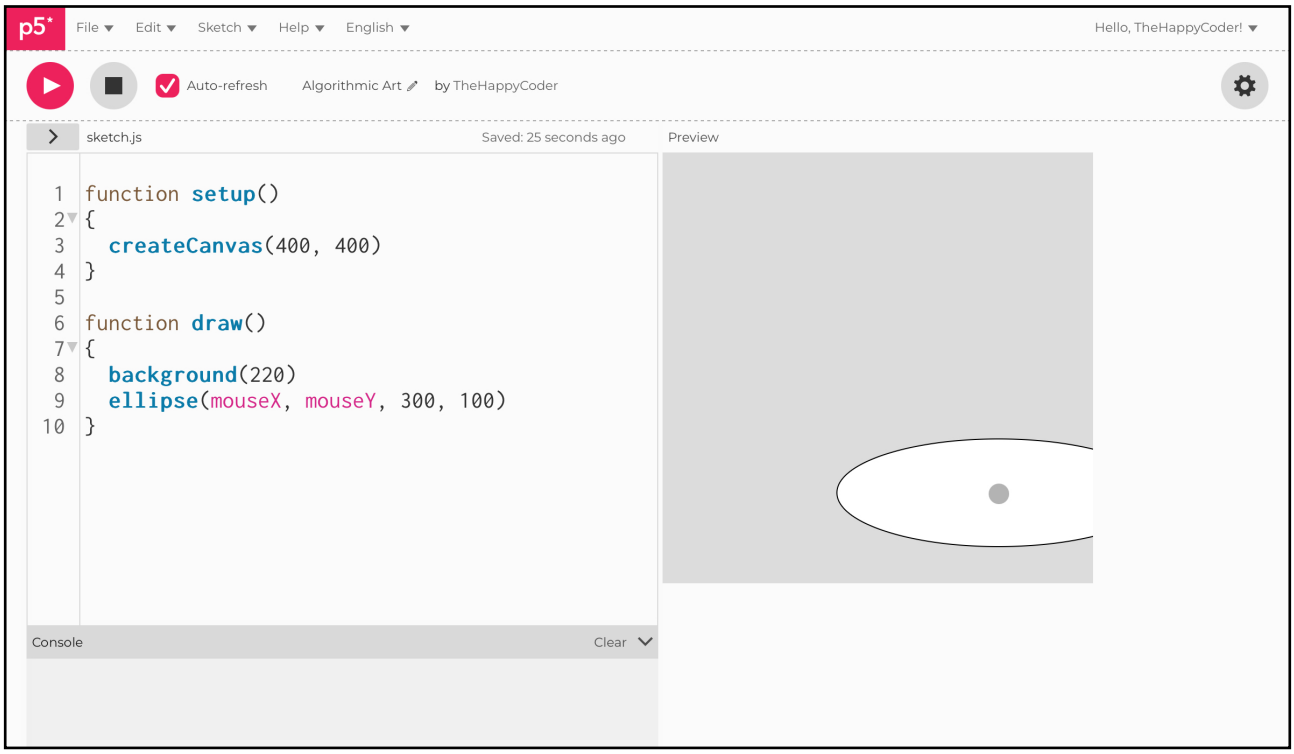
Notes

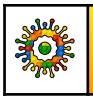
As you move your mouse cursor around the canvas, the ellipse follows it.

Code Explanation

`ellipse(mouseX, mouseY, 300, 100)`	mouseX returns the x coordinate and mouseY returns the y coordinate of the cursor.

Figure A6.2


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 }
5
6 function draw()
7 {
8 background(220)
9 ellipse(mouseX, mouseY, 300, 100)
10 }
Console
Clear



Sketch A6.3 stretchy time

If we move mouseX and mouseY to the dimensions of the ellipse, we can see how we can change the shape of the ellipse through moving our cursor.

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

function draw()
{
  background(220)
  ellipse(200, 200, mouseX, mouseY)
}
```

Notes

You can now stretch the ellipse by moving your mouse.

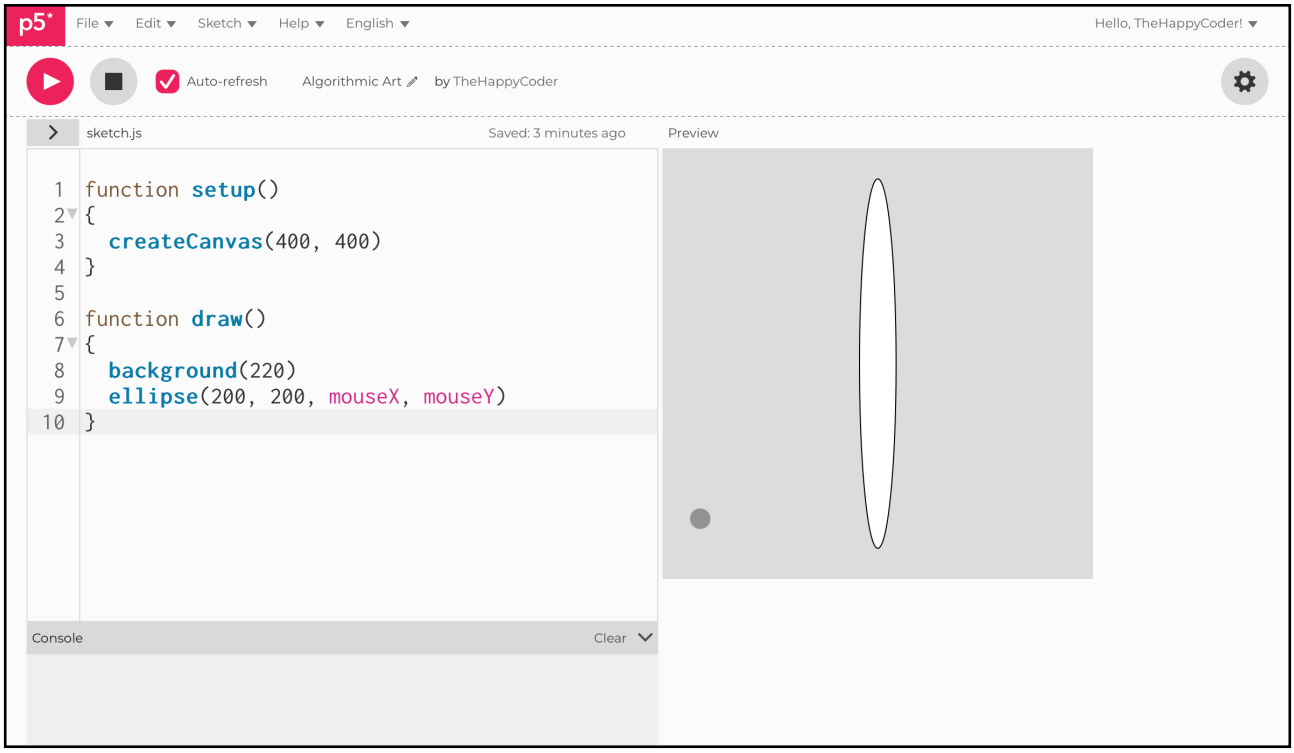
Challenge

Try it with the rectangle.

Code Explanation

`ellipse(200, 200, mouseX, mouseY)`	The mouseX and mouseY are the width and height respectfully.

Figure A6.3


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: 3 minutes ago
Preview
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
ellipse(200, 200, mouseX, mouseY)
}
Console
Clear

Sketch A6.4 triangle

Instead of an ellipse, let's make a triangle. A triangle has coordinates for all three corners, so the function `triangle()` has six arguments. There are a lot of numbers to get your head around; often, I sketch it out on a piece of paper first so I get the right coordinates for the right corner.

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

function draw()
{
  background(220)
  triangle(100, 100, 100, 300, 300, 300)
}
```

🗒️ Notes

We have three sets of coordinates: (100, 100), (100, 300) and (300, 300).

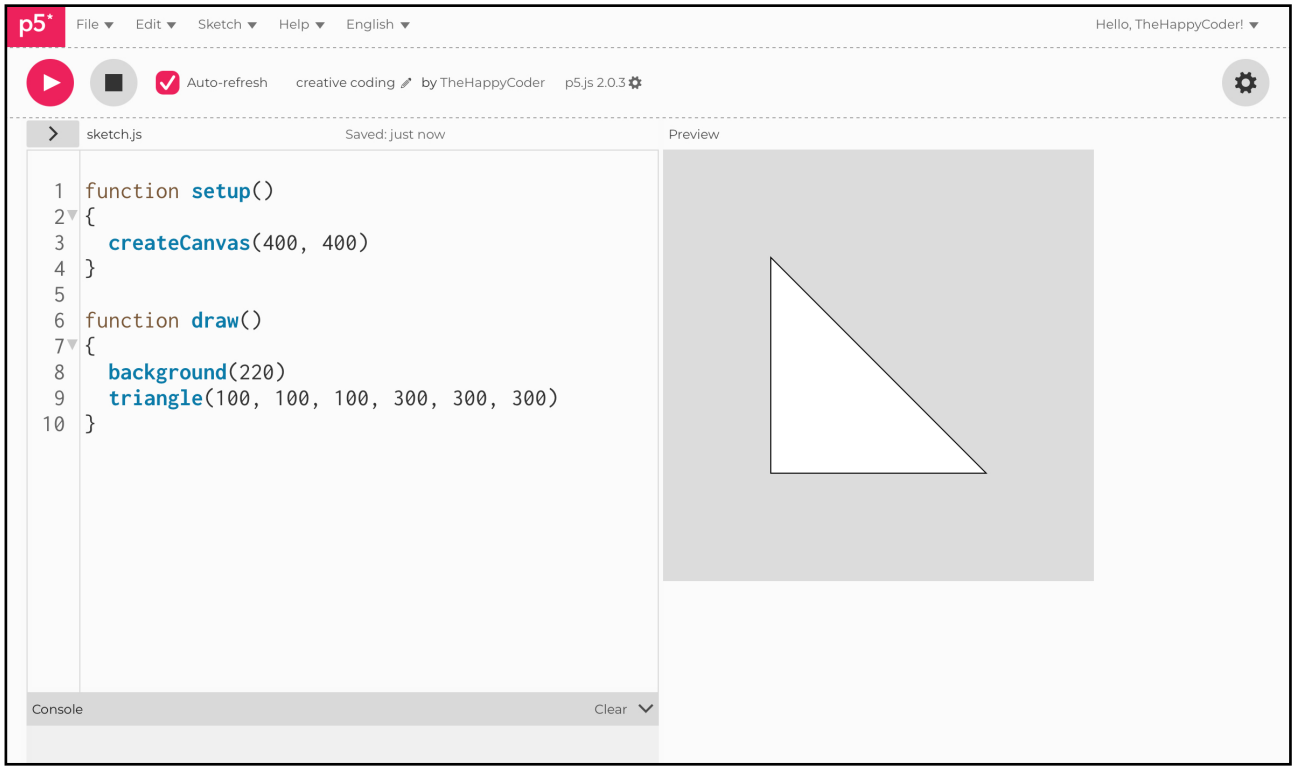
🌻 Challenge

Draw more triangles, make a simple pattern.

🛠️ Code Explanation

`triangle(100, 100, 100, 300, 300, 300)`	A triangle with three sets of coordinates for each corner (vertex)

Figure A6.4


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
1 function setup()
2 {
3 createCanvas(400, 400)
4 }
5
6 function draw()
7 {
8 background(220)
9 triangle(100, 100, 100, 300, 300, 300)
10 }
Console
Clear

Sketch A6.5 width and height

We can make use of another built-in variable, `width` and `height`. These variables return the width and height of the canvas. So in our case, the value of `width` is `400` and the value of `height` is also `400`. If we change the dimensions of the canvas, these values also change. If we divide the `width` and `height` by 2, we always get the centre of the canvas, whatever its dimensions.

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  triangle(100, 100, 100, 300, 300, 300)
}
```

🗒️ Notes

We have translated the origin of the canvas to its centre rather than the top left-hand corner. This has pushed the triangle down and to the left by `200` pixels.

🌻 Challenge

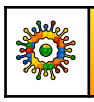
Change the dimensions of the canvas.

🛠️ Code Explanation

`translate(width/2, height/2)`	The width/2 is 200 and the height/2 is also 200

Figure A6.5


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 }
5
6 function draw()
7 {
8 background(220)
9 translate(width/2, height/2)
10 triangle(100, 100, 100, 300, 300, 300)
11 }
Console
Clear



Sketch A6.6 rebuilding the triangle

This is where a physical drawing often helps. We can redraw the triangle with new coordinates. Remember that the origin is now in the centre of the canvas, and we measure the x and y coordinates from the centre; hence, −100 is to the left of the centre for x and upwards for the y coordinates, and vice versa for 100.

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  triangle(-100, -100, -100, 100, 100, 100)
}
```

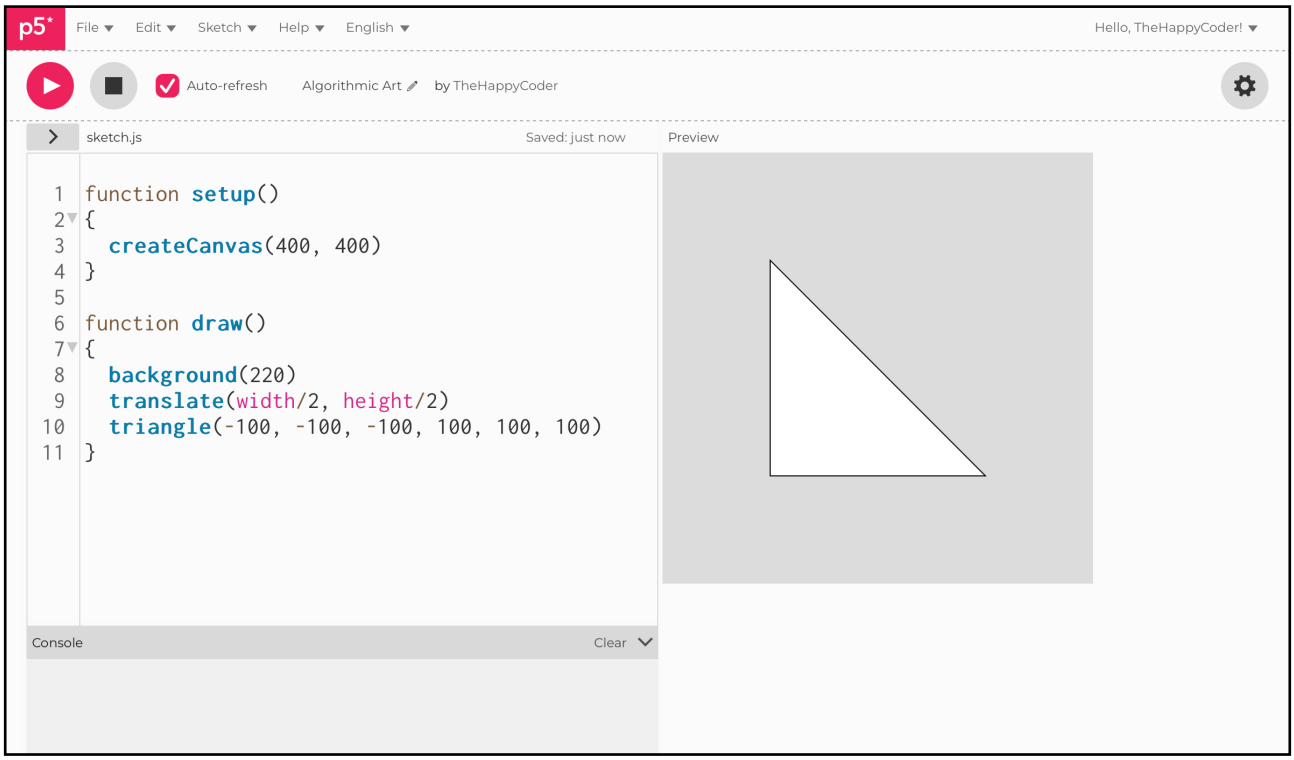
🗒️ Notes

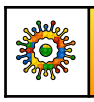
We have the same triangle as previously.

🌻 Challenge

Change the coordinates to draw different types of triangles.

Figure A6.6


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 }
5
6 function draw()
7 {
8 background(220)
9 translate(width/2, height/2)
10 triangle(-100, -100, -100, 100, 100, 100)
11 }
Console
Clear



Sketch A6.7 a length

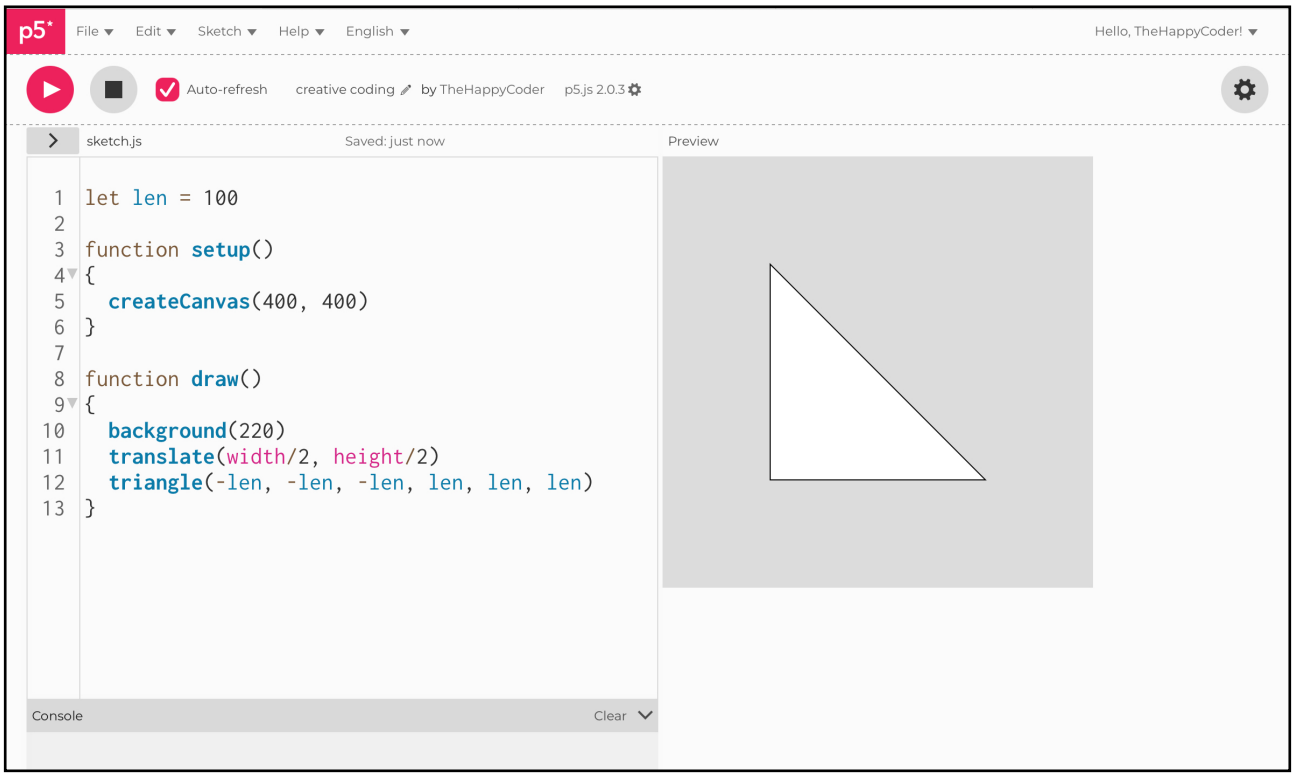
Let's give the dimension a variable called `len` (for length).

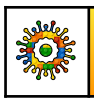
```
let len = 100

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  triangle(-len, -len, -len, len, len, len)
}
```

Notes

This can be handy if we change the dimension and therefore only need to change it once.

Figure A6.7


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 len = 100
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(width/2, height/2)
triangle(-len, -len, -len, len, len, len)
}
Console
Clear



Sketch A6.8 a bit more pointy

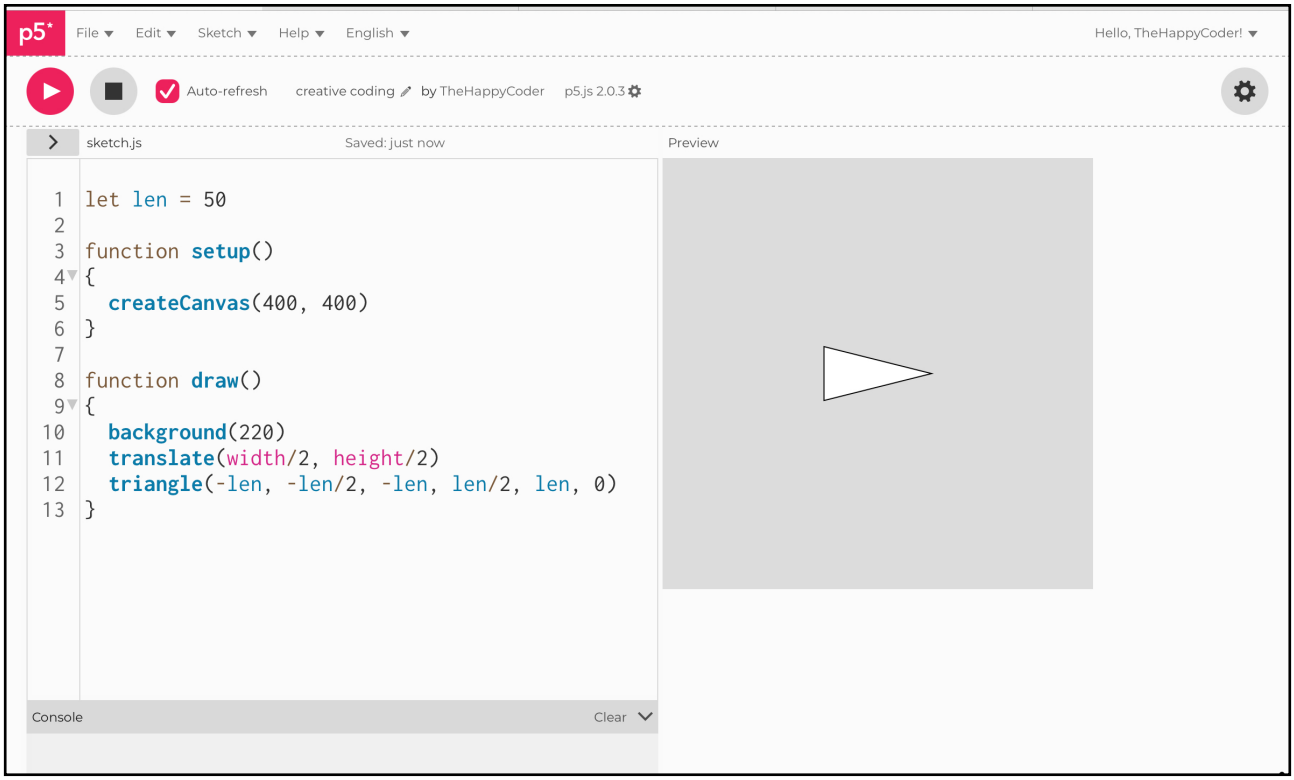
Make it a bit smaller and more pointy so that it isn't an equilateral triangle.

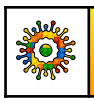
```
let len = 50


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


function draw()
{
  background(220)
  translate(width/2, height/2)
  triangle(-len, -len/2, -len, len/2, len, 0)
}
```

Notes

We can see what direction it is pointing in.

Figure A6.8


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 len = 50
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
translate(width/2, height/2)
triangle(-len, -len/2, -len, len/2, len, 0)
}
Console
Clear



Sketch A6.9 a hundred of them

Now we create a `for()` loop to draw `100` triangles. We put those two lines of code inside the `for()` loop.

```
let len = 50


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


function draw()
{
  background(220)
  for (let i = 0; i < 100; i++)
  {
    translate(width/2, height/2)
    triangle(-len, -len/2, -len, len/2, len, 0)
  }
}
```

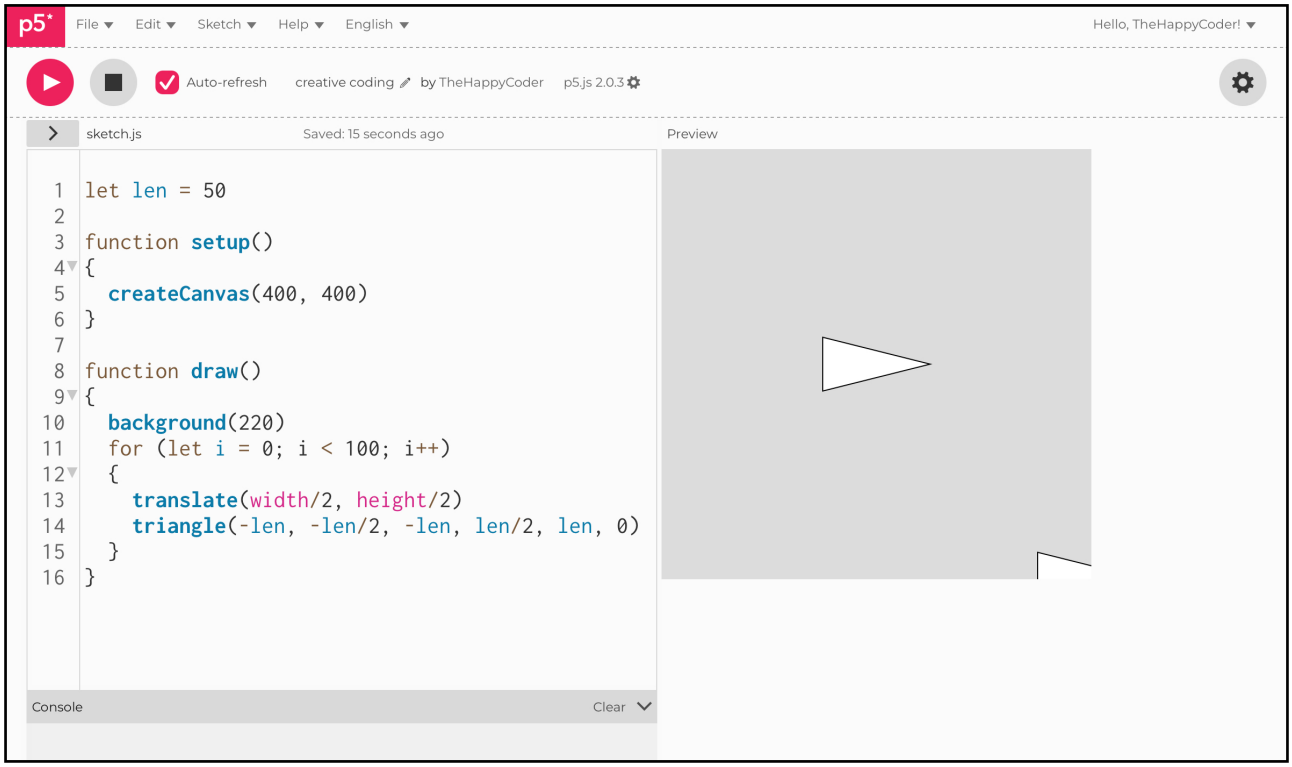
🗒️ Notes

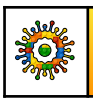
The problem is that the `translate()` function is accumulative, which means it shifts the origin by that amount each iteration in the `for()` loop. We need to use `push()` and `pop()`, and also we need to stop repeating the `for()` in `draw()` with `noLoop()`.

🛠️ Code Explanation

`for (let i = 0; i < 100; i++)`	A for() loop where the variable i which is defined and initialised to 0, then incremented by 1 until it reaches 100

Figure A6.9


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
creative coding
by TheHappyCoder
p5.js 2.0.3
sketch.js
Saved: 15 seconds ago
Preview
let len = 50
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
for (let i = 0; i < 100; i++)
{
translate(width/2, height/2)
triangle(-len, -len/2, -len, len/2, len, 0)
}
}
Console
Clear



Sketch A6.10 push pop stop

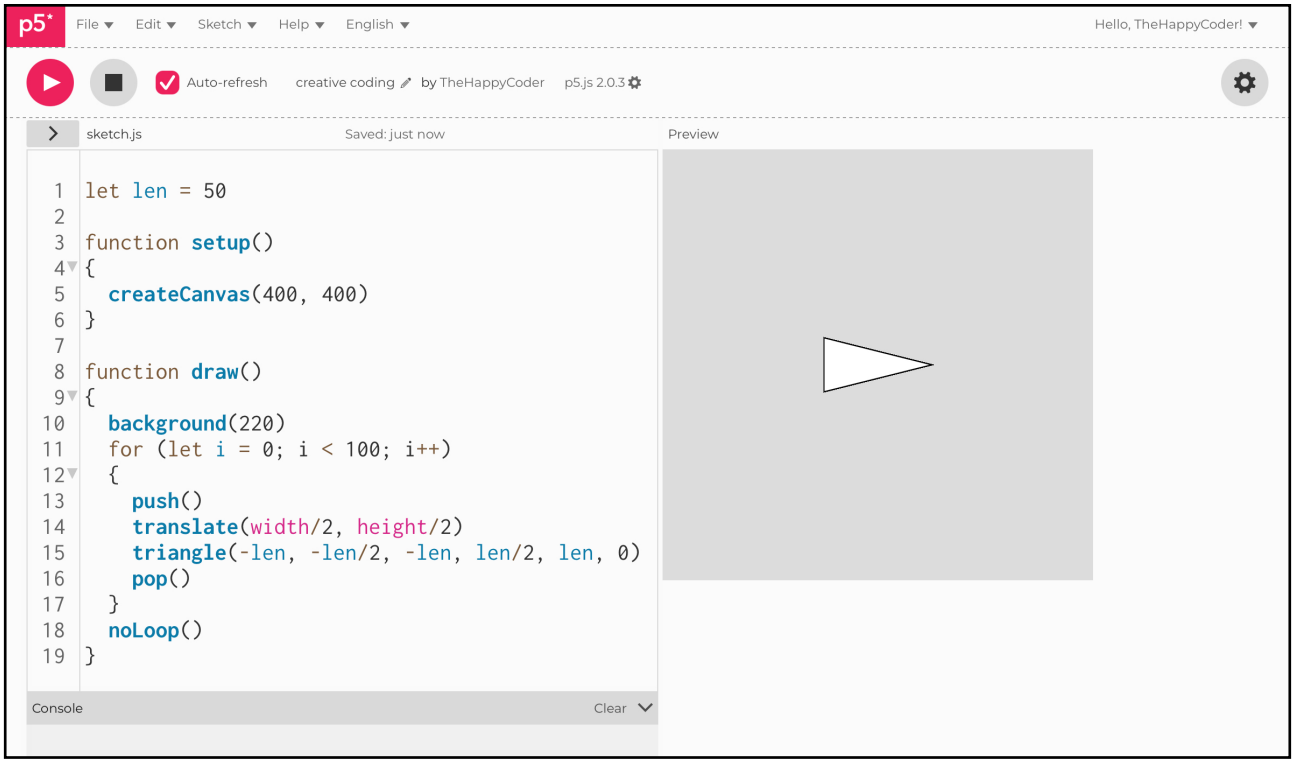
We add `push()` and `pop()` to stop the `translate()` from adding itself 100 times, and also the `noLoop()` to stop the `draw()` function once it has drawn the 100 triangles.

```
let len = 50

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

function draw()
{
  background(220)
  for (let i = 0; i < 100; i++)
  {
    push()
    translate(width/2, height/2)
    triangle(-len, -len/2, -len, len/2, len, 0)
    pop()
  }
  noLoop()
}
```

Notes

We have `100` triangles, but they are all in one place. We could also put all of this code in the `setup()` function and therefore no need to use `noLoop()`.

Figure A6.10


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 len = 50
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
for (let i = 0; i < 100; i++)
{
push()
translate(width/2, height/2)
triangle(-len, -len/2, -len, len/2, len, 0)
pop()
}
noLoop()
}
Console
Clear

Sketch A6.11 rotate the triangle

We are rotating all the triangles by 45° degrees.

```
let len = 50
let angle = 45

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

function draw()
{
  background(220)
  for (let i = 0; i < 100; i++)
  {
    push()
    translate(width/2, height/2)
    rotate(angle)
    triangle(-len, -len/2, -len, len/2, len, 0)
    pop()
  }
  noLoop()
}
```

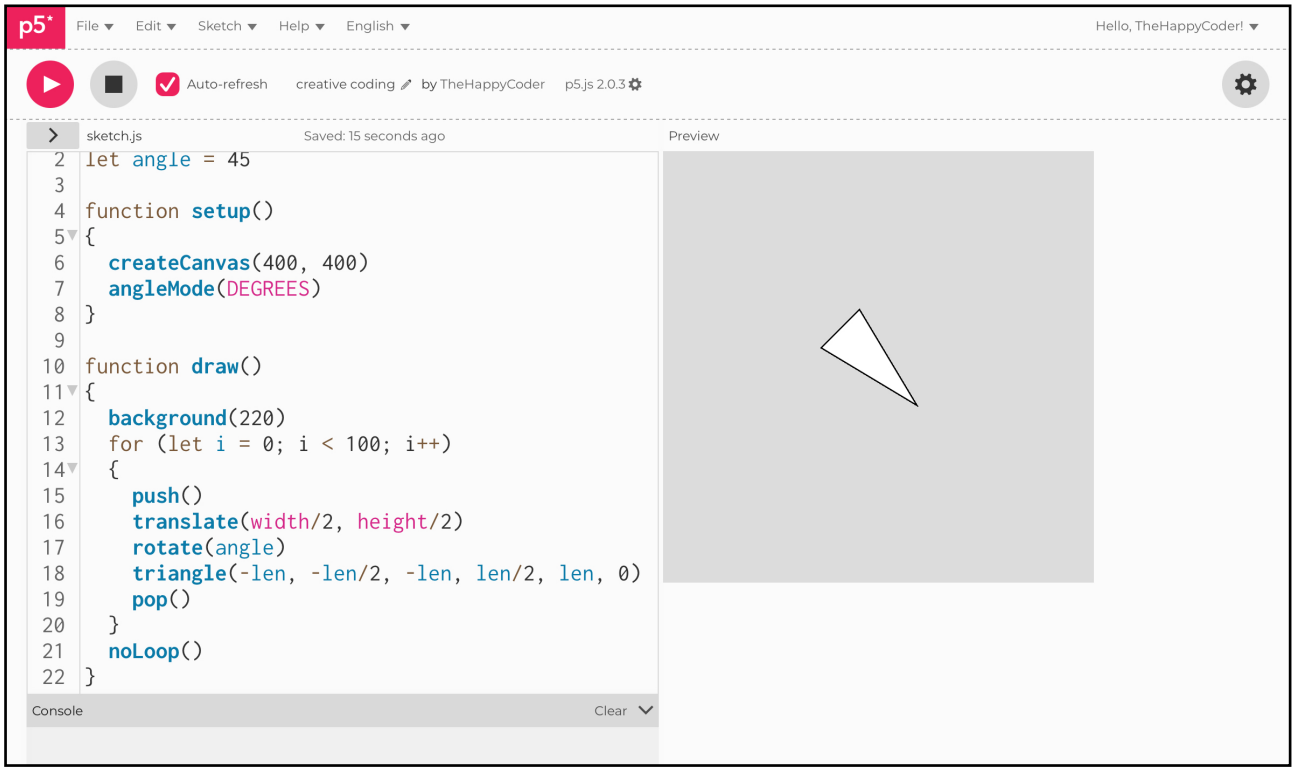
🗒️ Notes

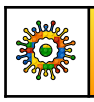
We have 100 triangles, all rotated by 45°.

🌻 Challenge

Try other angles.

Figure A6.11


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
creative coding
by TheHappyCoder
p5.js 2.0.3
sketch.js
Saved: 15 seconds ago
Preview
let angle = 45
function setup()
{
createCanvas(400, 400)
angleMode(DEGREES)
}
function draw()
{
background(220)
for (let i = 0; i < 100; i++)
{
push()
translate(width/2, height/2)
rotate(angle)
triangle(-len, -len/2, -len, len/2, len, 0)
pop()
}
noLoop()
}
Console
Clear



Sketch A6.12 split the triangles up

But we want to see all the triangles; to do that, we can translate them randomly across the whole canvas (`width` and `height`).

```
let len = 50
let angle = 45

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

function draw()
{
  background(220)
  for (let i = 0; i < 100; i++)
  {
    push()
    translate(random(width), random(height))
    rotate(angle)
    triangle(-len, -len/2, -len, len/2, len, 0)
    pop()
  }
  noLoop()
}
```

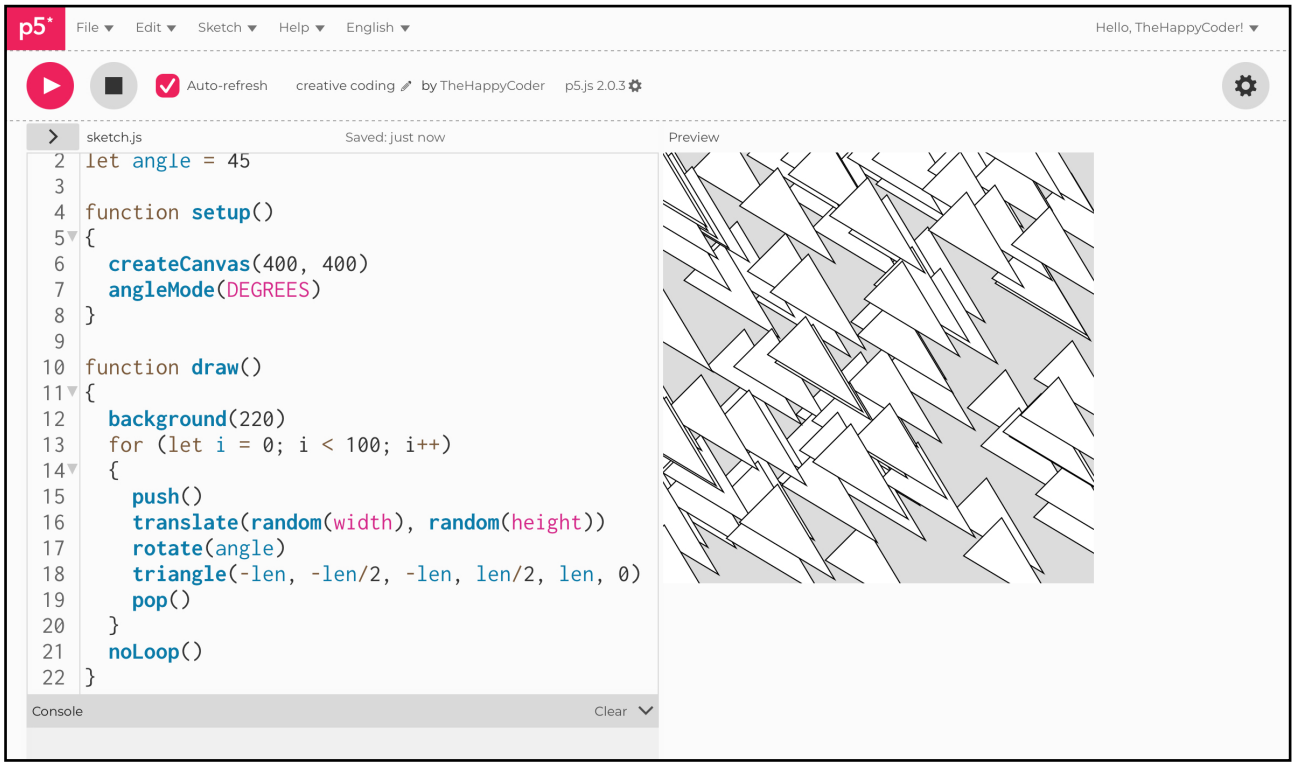
Notes

Now each triangle drawn has its own random position.

🌻 Challenges

1. Try pushing them all into a square in the middle.
2. Make the triangles smaller or random in size.
3. Can you work out how to rotate each one separately?

Figure A6.12


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 angle = 45
function setup()
{
createCanvas(400, 400)
angleMode(DEGREES)
}
function draw()
{
background(220)
for (let i = 0; i < 100; i++)
{
push()
translate(random(width), random(height))
rotate(angle)
triangle(-len, -len/2, -len, len/2, len, 0)
pop()
}
noLoop()
}
Console
Clear

Sketch A6.13 random angles

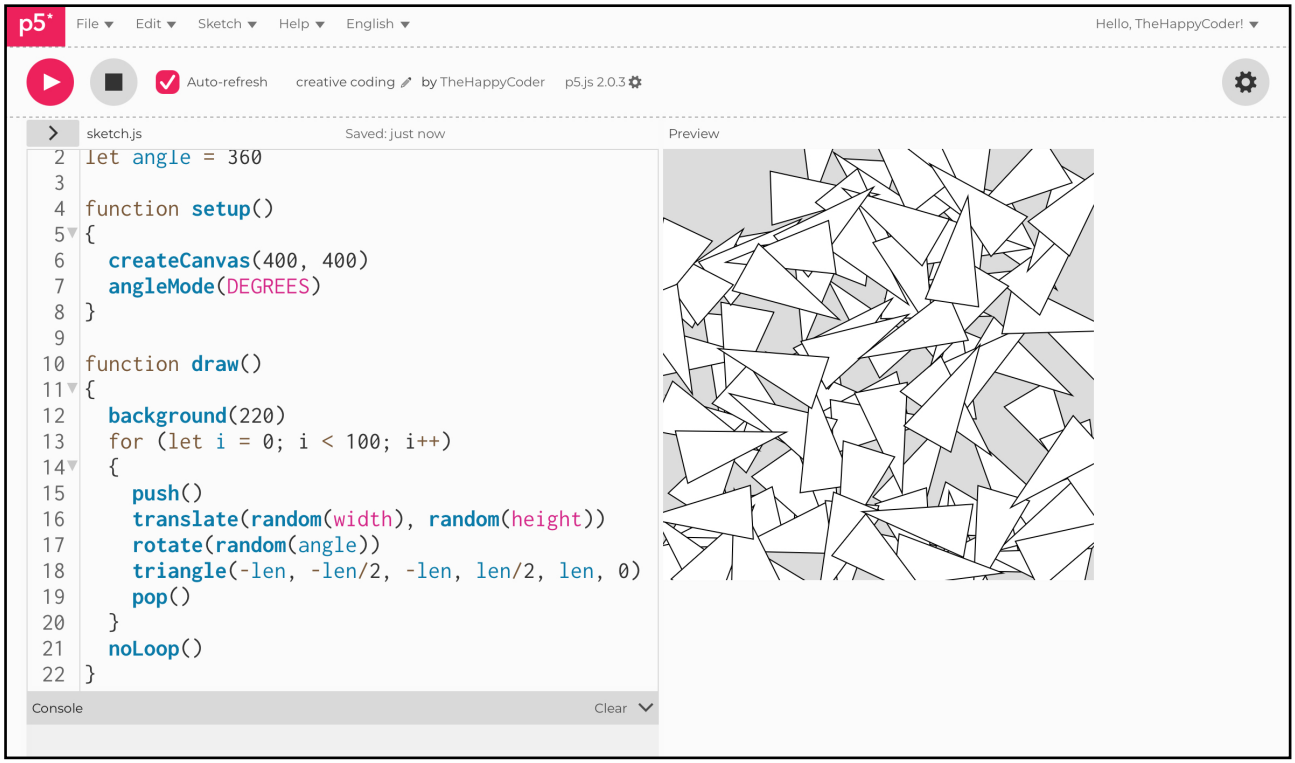
This is how you can randomly set the angles. Change the angle to a full 360° and rotate randomly within that angle.

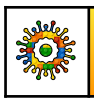
```
let len = 50
let angle = 360

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

function draw()
{
  background(220)
  for (let i = 0; i < 100; i++)
  {
    push()
    translate(random(width), random(height))
    rotate(random(angle))
    triangle(-len, -len/2, -len, len/2, len, 0)
    pop()
  }
  noLoop()
}
```

Notes

Rotating randomly from 0° to 360° degrees

Figure A6.13


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 angle = 360
function setup()
{
createCanvas(400, 400)
angleMode(DEGREES)
}
function draw()
{
background(220)
for (let i = 0; i < 100; i++)
{
push()
translate(random(width), random(height))
rotate(random(angle))
triangle(-len, -len/2, -len, len/2, len, 0)
pop()
}
noLoop()
}
Console
Clear



Sketch A6.14 random length

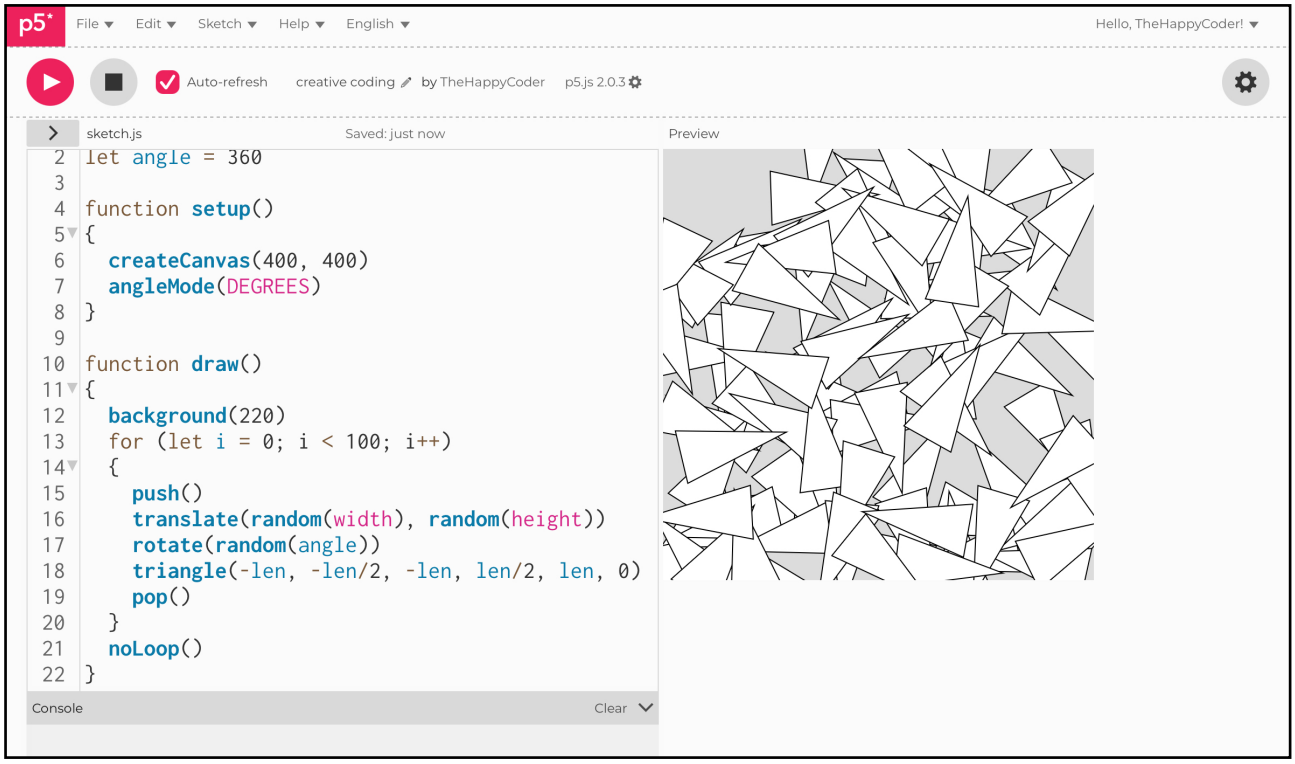
Now for the size of the triangle. This will keep the proportions the same. We will have triangles between 5 and 50 long.

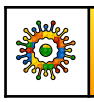
```
let len = 50
let angle = 360

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

function draw()
{
  background(220)
  for (let i = 0; i < 100; i++)
  {
    push()
    translate(random(width), random(height))
    rotate(random(angle))
    len = random(5, 50)
    triangle(-len, -len/2, -len, len/2, len, 0)
    pop()
  }
  noLoop()
}
```

Notes

There is a lot to play with here.

Figure A6.14


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 angle = 360
function setup()
{
createCanvas(400, 400)
angleMode(DEGREES)
}
function draw()
{
background(220)
for (let i = 0; i < 100; i++)
{
push()
translate(random(width), random(height))
rotate(random(angle))
triangle(-len, -len/2, -len, len/2, len, 0)
pop()
}
noLoop()
}
Console
Clear



Sketch A6.15 random colouring

Make the background white and give each triangle a random colour (and alpha).

```
let len = 50
let angle = 360

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

function draw()
{
  background(255)
  for (let i = 0; i < 100; i++)
  {
    push()
    translate(random(width), random(height))
    rotate(random(angle))
    len = random(5, 50)
    fill(random(255), random(255), random(255), random(255))
    triangle(-len, -len/2, -len, len/2, len, 0)
    pop()
  }
  noLoop()
}
```

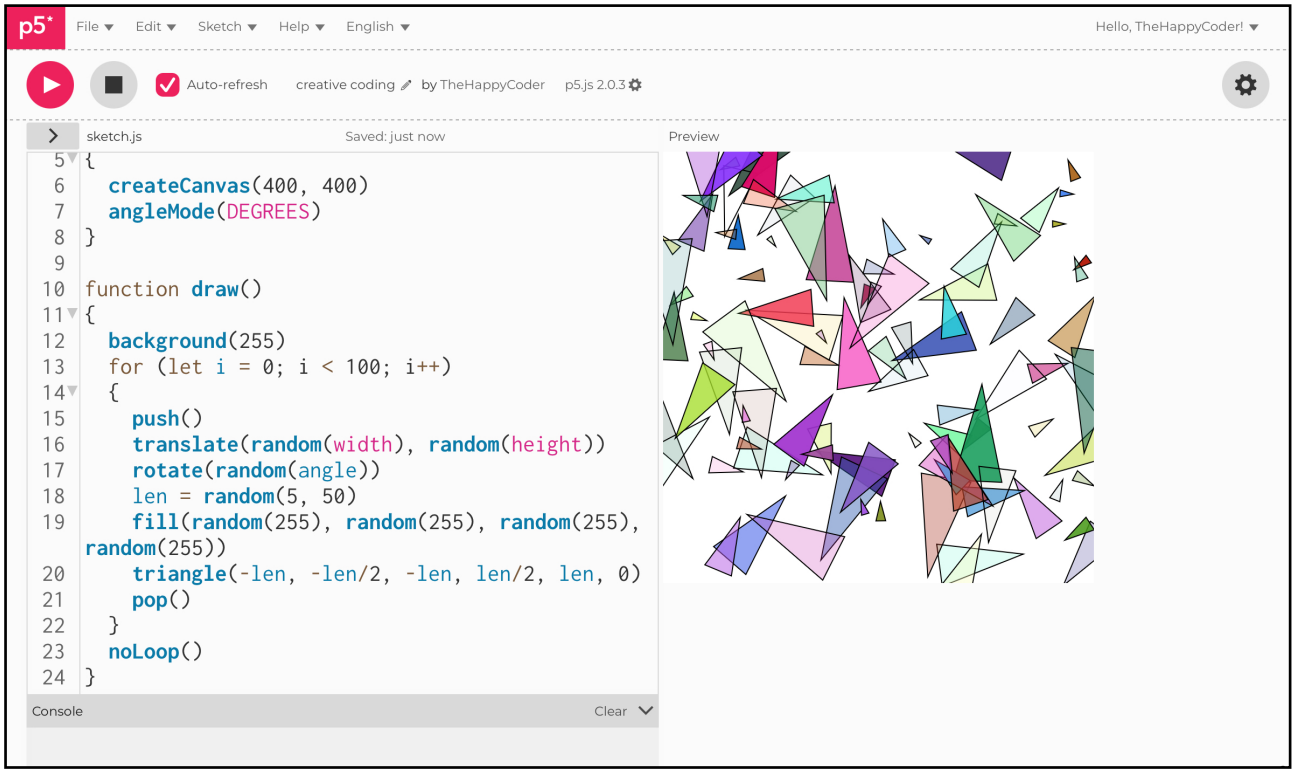
Notes

Starting to look interesting.

🌻 Challenges

1. What else could you develop alongside these ideas?
2. Different shapes, colours, etc.

Figure A6.15


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
{
createCanvas(400, 400)
angleMode(DEGREES)
}
function draw()
{
background(255)
for (let i = 0; i < 100; i++)
{
push()
translate(random(width), random(height))
rotate(random(angle))
len = random(5, 50)
fill(random(255), random(255), random(255), random(255))
triangle(-len, -len/2, -len, len/2, len, 0)
pop()
}
noLoop()
}
Console
Clear

Next. . .

In the next unit (#7), we will go very small and look at pixels; they are the dots on the canvas that make up your screen.