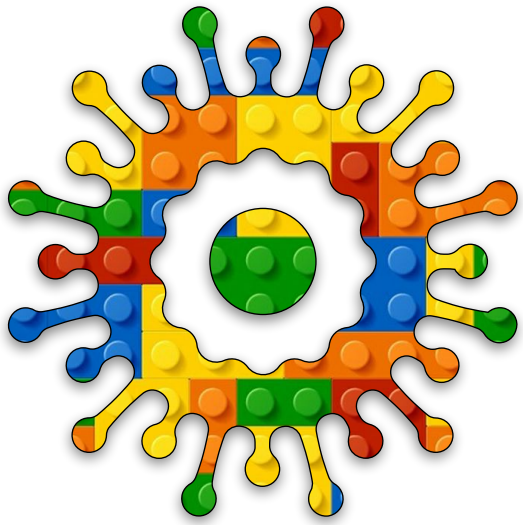
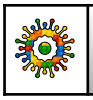
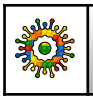
Creative Coding Module C Unit #1 3D shapes





Content

Module C Unit #1 Primitive 3D shapes



Introduction to primitive 3D shapes

In p5.js, WEBGL is one of the two available rendering modes, allowing you to create 3D graphics and interactive experiences alongside its default 2D mode.
Key features of WEBGL in p5.js:

🀄 3D shapes: Draw basic shapes like boxes (cuboids), spheres, cones, cylinders, and more using functions like `box()`, `sphere()`, etc.

🀄 Custom geometry: Create complex models from code or load them from 3D file formats like `OBJ` and `STL`.

🀄 Materials and lighting: Define materials like `basicMaterial()` or `specularMaterial()` and use lights like `ambientLight()` to affect object appearance.

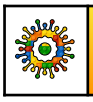
🀄 Camera control: Change the viewpoint using functions like `perspective()`, `ortho()`, and rotate the camera with `rotateX()`, `rotateY()`, `rotateZ()`.

🀄 Textures: Add textures to surfaces for enhanced realism and detail.
Shaders: For advanced users, write custom shaders to achieve unique visual effects.

We will draw some of the basic shapes and learn about the co-ordinates, translation, and rotation. There are a number of very key differences between the default 2D and the 3D environment.

One of the main changes is how we use the coordinates. All coordinates are based around the centre of the 3D space. There is an `x`, `y`, and `z` component to any coordinates; if only two are specified, then it takes the `x` and `y` and relegates the `z` to zero.

The `x` component is left and right, `y` is top and bottom, and the `z` is front and back.



Sketch C1.1 standard sketch

Starting with our standard sketch.

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

function draw()
{
  background(220)
}
```

Sketch C1.2 adding the WEBGL

The first thing to notice is that we have the third argument in the `createCanvas()` function: `WEBGL`.

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

function draw()
{
  background(220)
}
```

Notes

The first thing you notice is that nothing happens. What you now have is effectively a 3D canvas or space. One where you can draw in the `x`, `y`, and `z` directions.

Code Explanation

`createCanvas(400, 400, WEBGL)`	WEGL allows to render an image in 3D

Sketch C1.3 drawing a box

What we need is a 3D shape to put into this space. We will start by drawing a cube. To draw a cube, we use the box() function; it will have dimensions of 100 pixels by 100 pixels by 100 pixels. We only need to give the single dimension, as it assumes the other two are the same. We will add the other two when we draw a cuboid later.

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

function draw()
{
  background(220)
  box(100)
}
```

🗒️ Notes

One thing you will notice is that I have not specified where to draw it. I have given it no coordinates, yet it has drawn it in the centre of the canvas. Also, it doesn't look very 3D...ish. To the first point, the centre of the canvas is in the centre in all three dimensions, the x, y, and z axes. Secondly, we need to rotate it to see it.

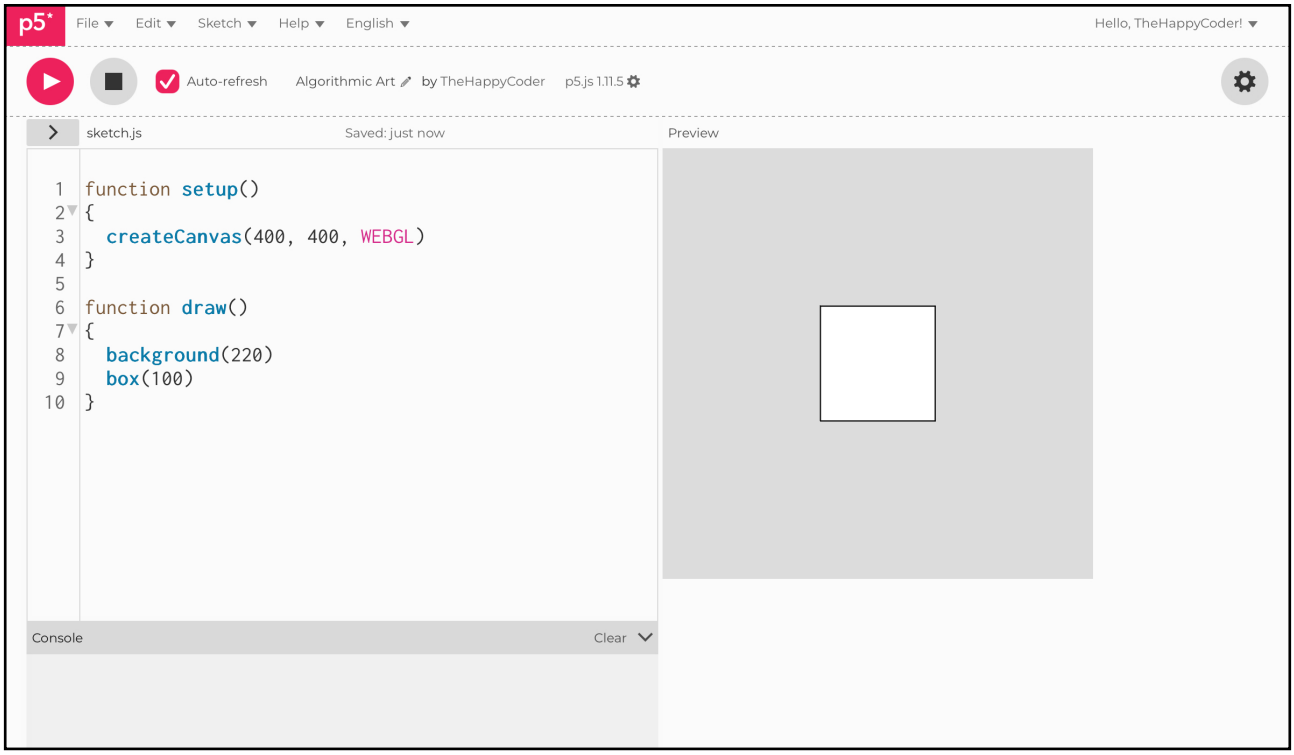
🌻 Challenge

Add the other two dimensions.

🛠️ Code Explanation

box(100)	A rectangle with each side of equal length (100)

Figure C1.3


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, WEBGL)
}
function draw()
{
background(220)
box(100)
}
Console
Clear

Sketch C1.4 rotating

We cannot simply rotate it; we have to specify which axis we want to rotate it on, whether it is the `x`, `y`, or `z` axis, and by some angle (measured in `radians`, for now). The functions we use are: `rotateX()`, `rotateY()`, and `rotateZ()`, which I hope are self-explanatory. We will rotate the box along the `x` axis by `2` radians.

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

function draw()
{
  background(220)
  rotateX(2)
  box(100)
}
```

🗒️ Notes

We can now see that it is a cube, at least from one angle.

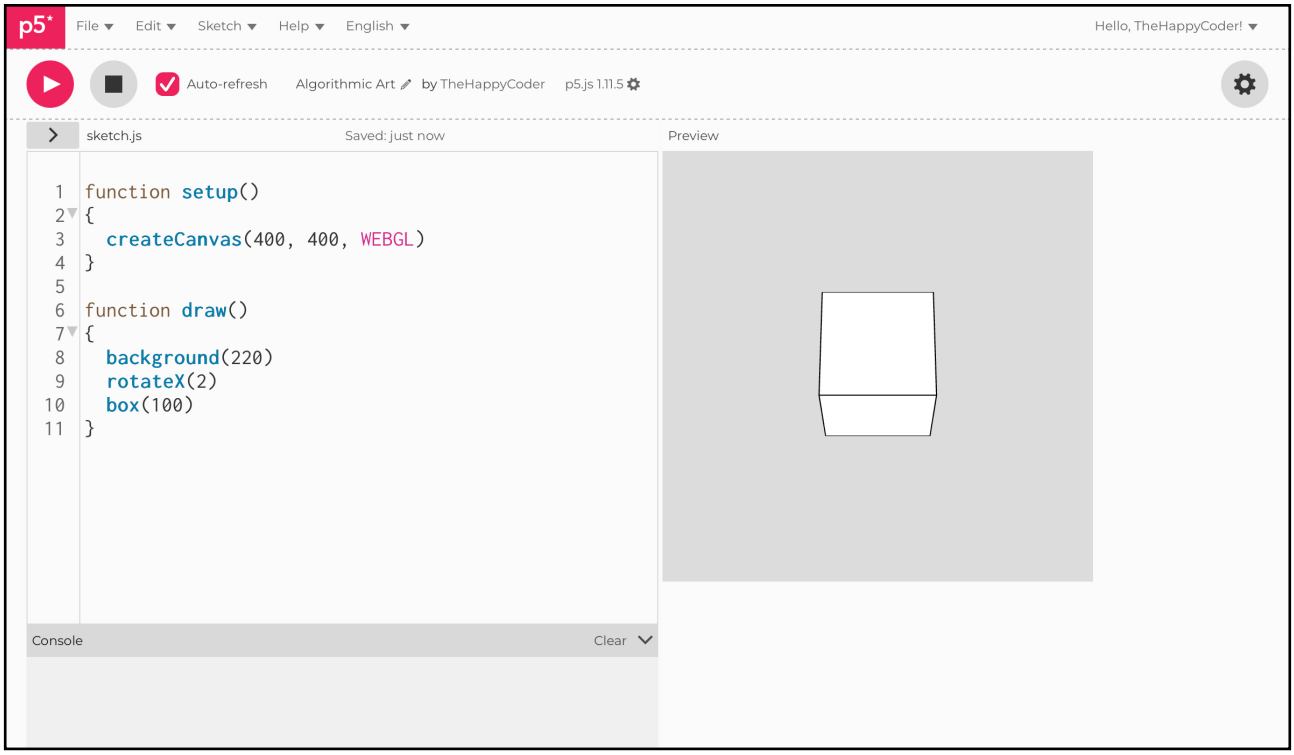
🌻 Challenge

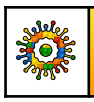
Try other angles.

🛠️ Code Explanation

`rotateX(2)`	Rotate along the x axis by 2 radians

Figure C1.4


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, WEBGL)
}
function draw()
{
background(220)
rotateX(2)
box(100)
}
Console
Clear



Sketch C1.5 45° rotation

To make life a little easier to understand, I will change the angle mode to degrees, which is a bit more intuitive.

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

function draw()
{
  background(220)
  rotateX(45)
  box(100)
}
```

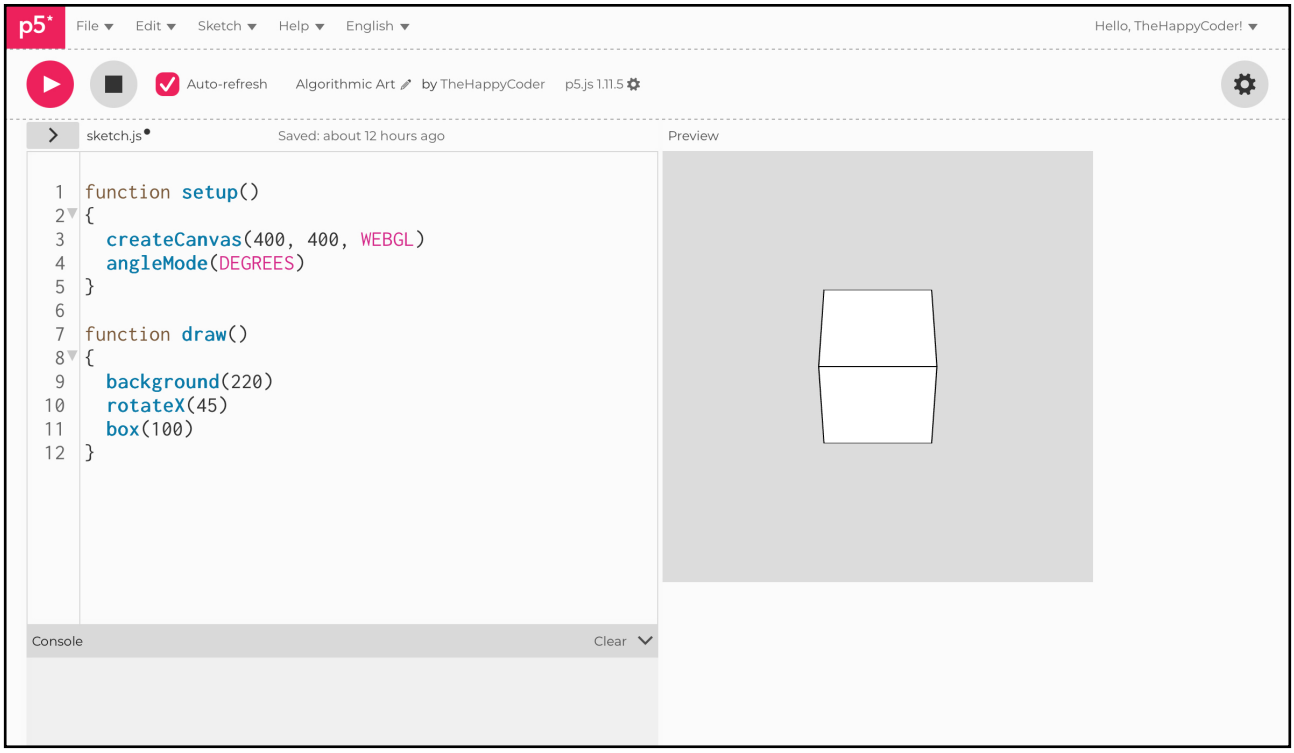
🗒️ Notes

Almost the same

🛠️ Code Explanation

`rotateX(45)`	Rotating through 45° along the x axis

Figure C1.5


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: about 12 hours ago
Preview
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(45)
box(100)
}
Console
Clear

Sketch C1.6 the x and y axis

To show how 3D it is, we are going to rotate along all three axes.

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

function draw()
{
  background(220)
  rotateX(45)
  rotateY(45)
  rotateZ(45)
  box(100)
}
```

🗒️ Notes

That is definitely more 3D...ish.

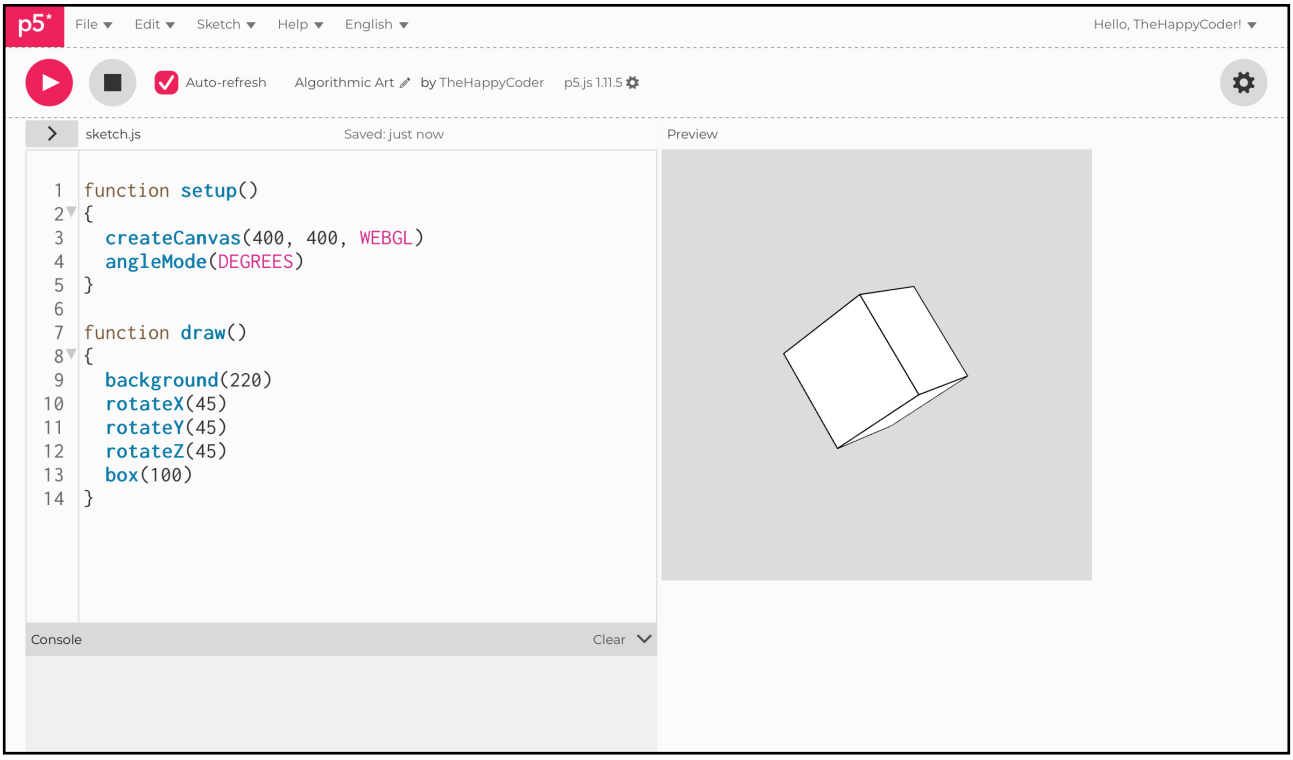
🌻 Challenge

Try different angles.

🛠️ Code Explanation

`rotateY(45)`	Rotating through 45° along the y axis
`rotateZ(45)`	Rotating through 45° along the z axis

Figure C1.6


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, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(45)
rotateY(45)
rotateZ(45)
box(100)
}
Console
Clear

Sketch C1.7 incrementing the rotation

Even better still, we can animate the rotation by introducing a variable (`angle`) and incrementing it.

```
let angle = 0

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

function draw()
{
  background(220)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100)
  angle++
}
```

Notes

You should see it rotating in all directions. We have incremented the angle of rotation by `1°` on all three axes.

Challenges

1. Try `-angle` for one of them.
2. Move it faster: `angle += 3`.

Sketch C1.8 drawing a cuboid

That was just a cube; we can add other dimensions to make it a cuboid.

```
let angle = 0

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

function draw()
{
  background(220)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100, 150, 50)
  angle++
}
```

🗒️ Notes

Nicely rotating cuboid.

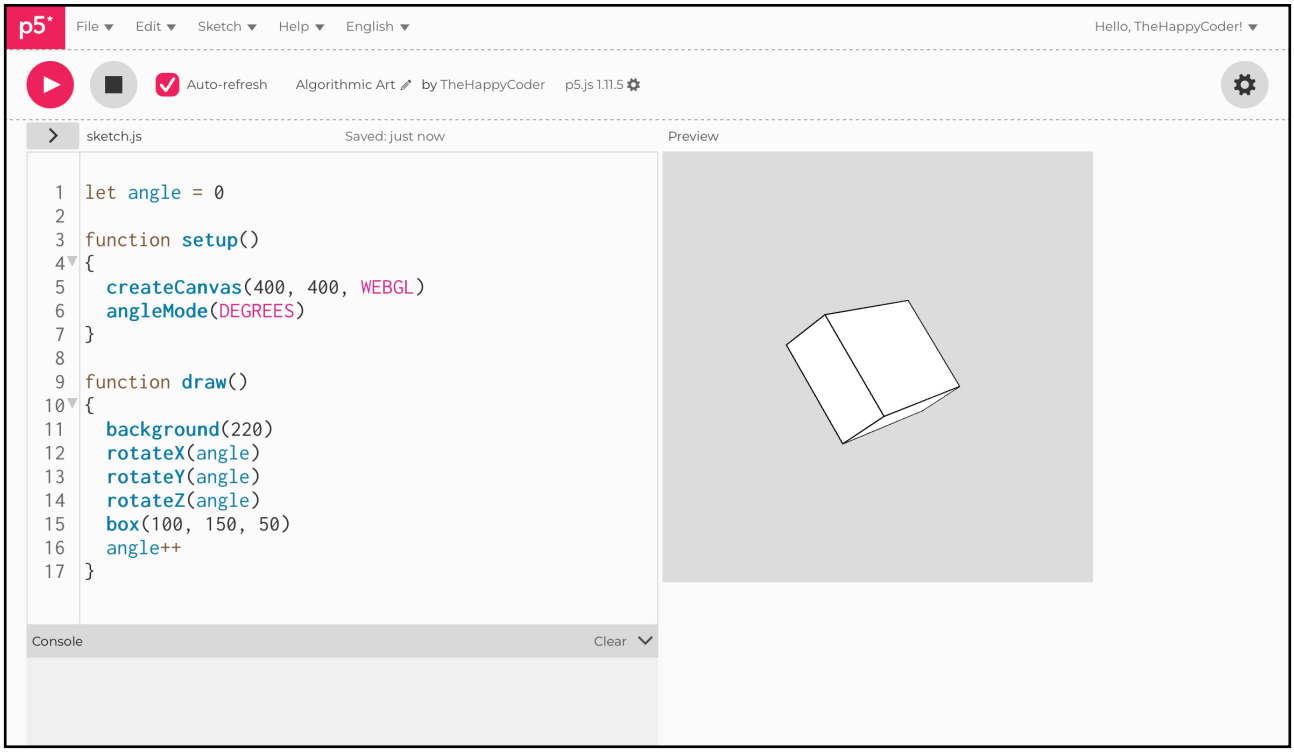
🌻 Challenge

Try other dimensions.

🛠️ Code Explanation

`box(100, 150, 50)`	Draws a cuboid 100 wide, 150 long and 50 deep

Figure C1.8


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
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
box(100, 150, 50)
angle++
}
Console
Clear

Sketch C1.9 drawing a plane

We will draw another shape, a simple plane, where it has two dimensions. Do you notice that there is a line going across it? This is because all the shapes that are created are made up of triangles.

```
let angle = 0

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

function draw()
{
  background(220)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  plane(100, 150)
  angle++
}
```

🗒️ Notes

A floating plane.

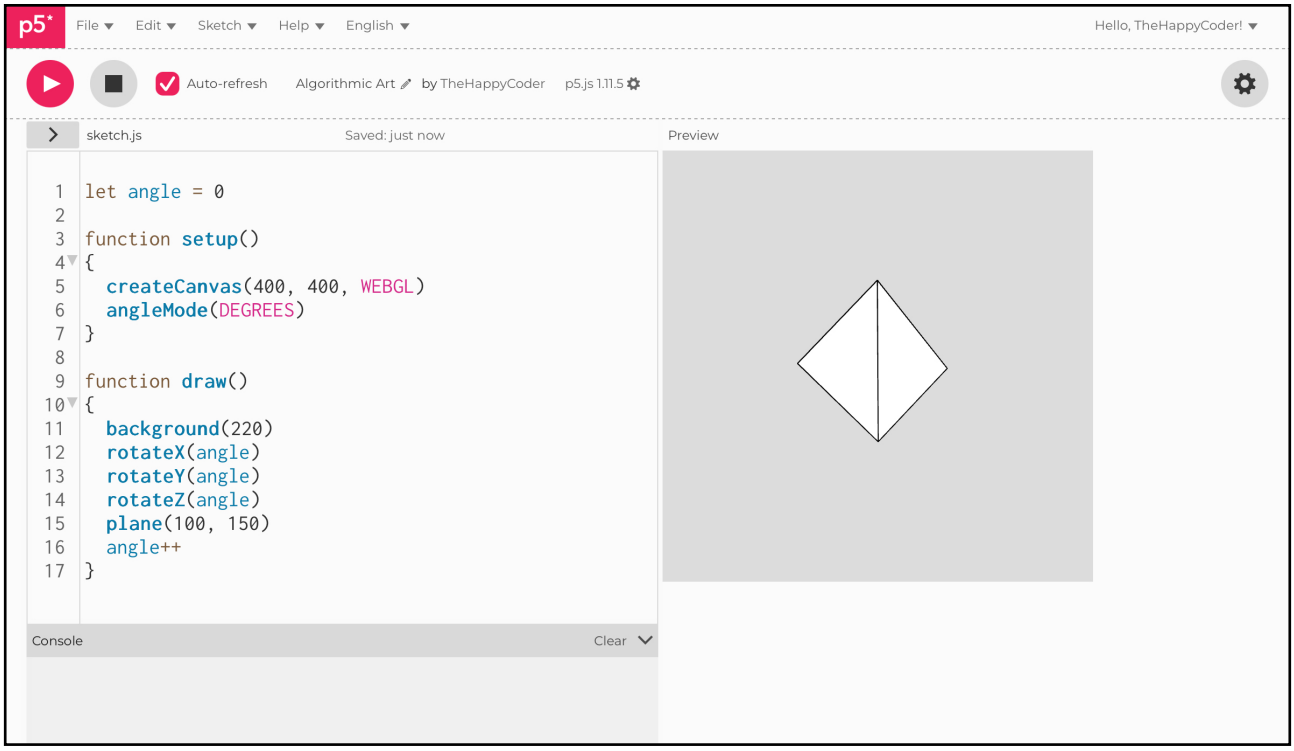
🌻 Challenge

Add noStroke().

🛠️ Code Explanation

plane(100, 150)	Creates a flat plane 100 by 150

Figure C1.9


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
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
plane(100, 150)
angle++
}
Console
Clear

Sketch C1.10 drawing a cylinder

A `cylinder` has two dimensions, one for the radius and the other for the length.

```
let angle = 0

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

function draw()
{
  background(220)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  cylinder(100, 150)
  angle++
}
```

🗒️ Notes

The first is the diameter and the second is the length. If you use `noStroke()` you lose a lot of definition; we will use lights to address that issue (in another unit).

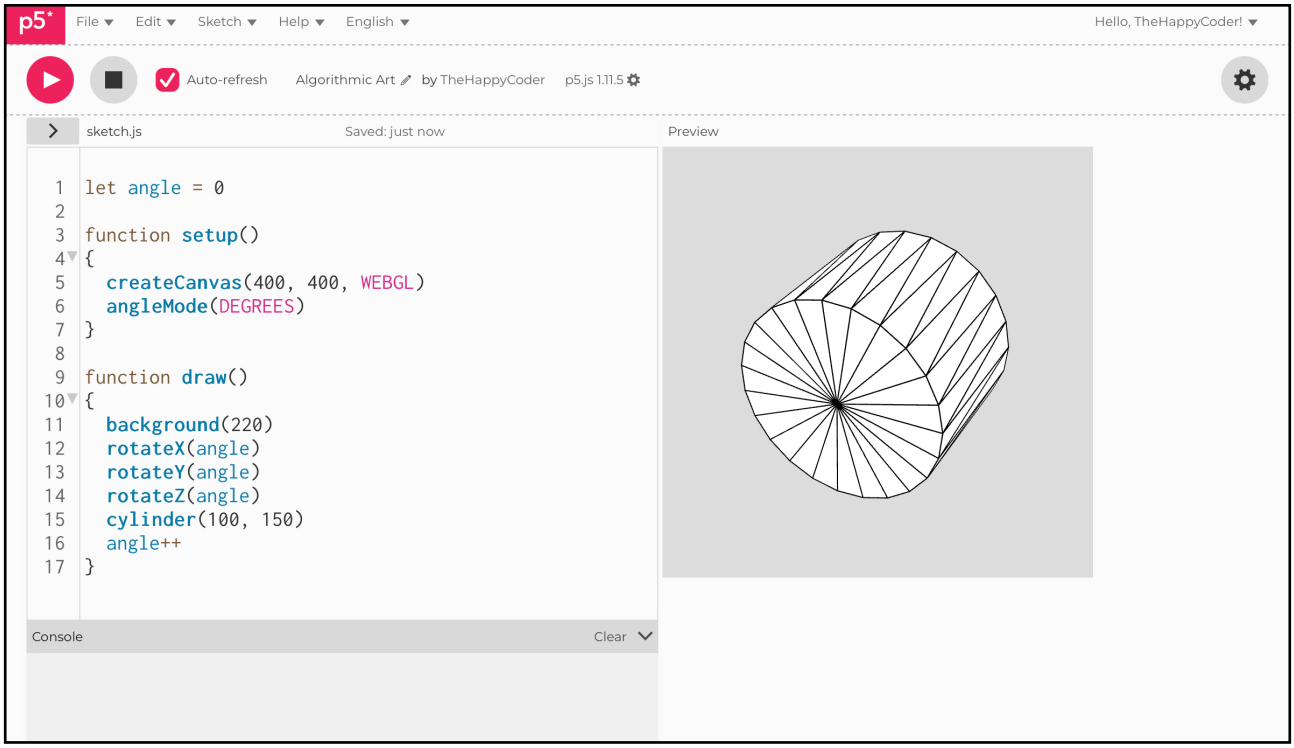
🌻 Challenge

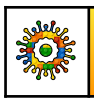
Alter the dimensions.

🛠️ Code Explanation

`cylinder(100, 150)`	A cylinder with a radius 100 and length 150

Figure C1.10


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 angle = 0
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
cylinder(100, 150)
angle++
}
Console
Clear



Sketch C1.11 drawing a sphere

Here is a sphere with a single dimension, the radius.

```
let angle = 0

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

function draw()
{
  background(220)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  sphere(100)
  angle++
}
```

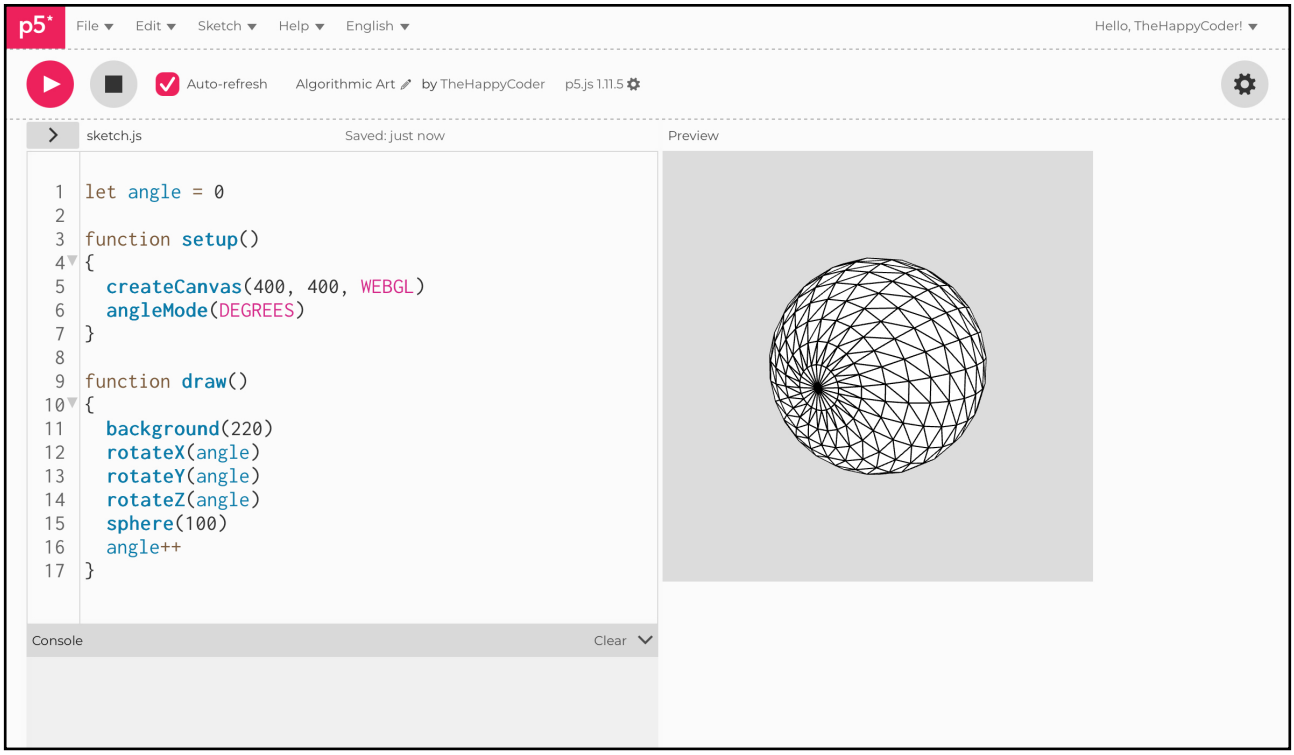
🗒️ Notes

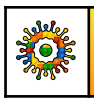
We will explore how to make lots of shapes in different positions at a later date.

🛠️ Code Explanation

`sphere(100)`	Sphere with a radius of 100

Figure C1.11


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 angle = 0
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
sphere(100)
angle++
}
Console
Clear



Sketch C1.12 drawing an ellipsoid

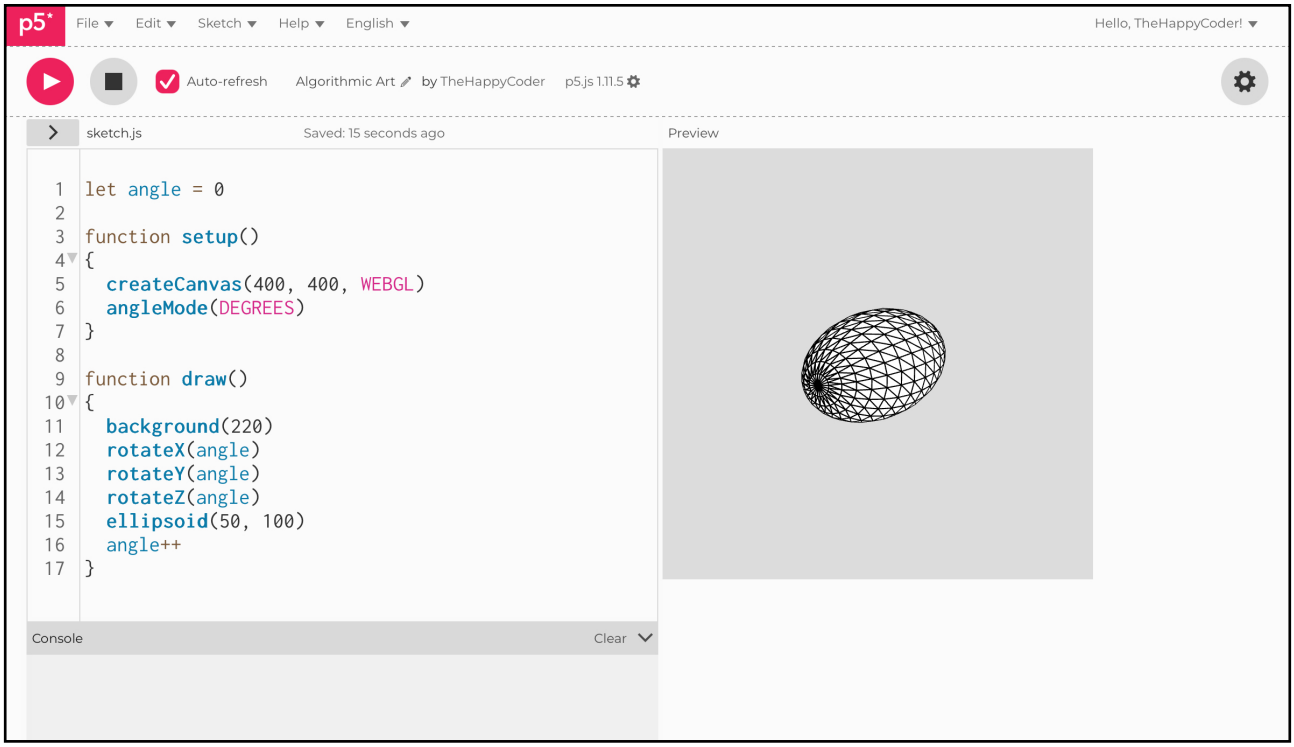
An ellipsoid is a sphere but with two dimension. A radius in one direction and a perpendicular radius.

```
let angle = 0

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

function draw()
{
  background(220)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  ellipsoid(50, 100)
  angle++
}
```

🛠️ Code Explanation

`ellipsoid(50, 100)`	Drawing an ellipsoid 50 wide, 100 long

Figure C1.12


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: 15 seconds ago
Preview
let angle = 0
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
ellipsoid(50, 100)
angle++
}
Console
Clear

Sketch C1.13 drawing a torus

Now for a `torus` (doughnut/donut), which has two dimensions: the torus radius and tube radius (thickness of the doughnut).

```
let angle = 0

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

function draw()
{
  background(220)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  torus(100, 50)
  angle++
}
```

🗒️ Notes

The radius of the torus (taken at the centre of the tube) is the first argument, and the radius of the actual tube itself is the second.

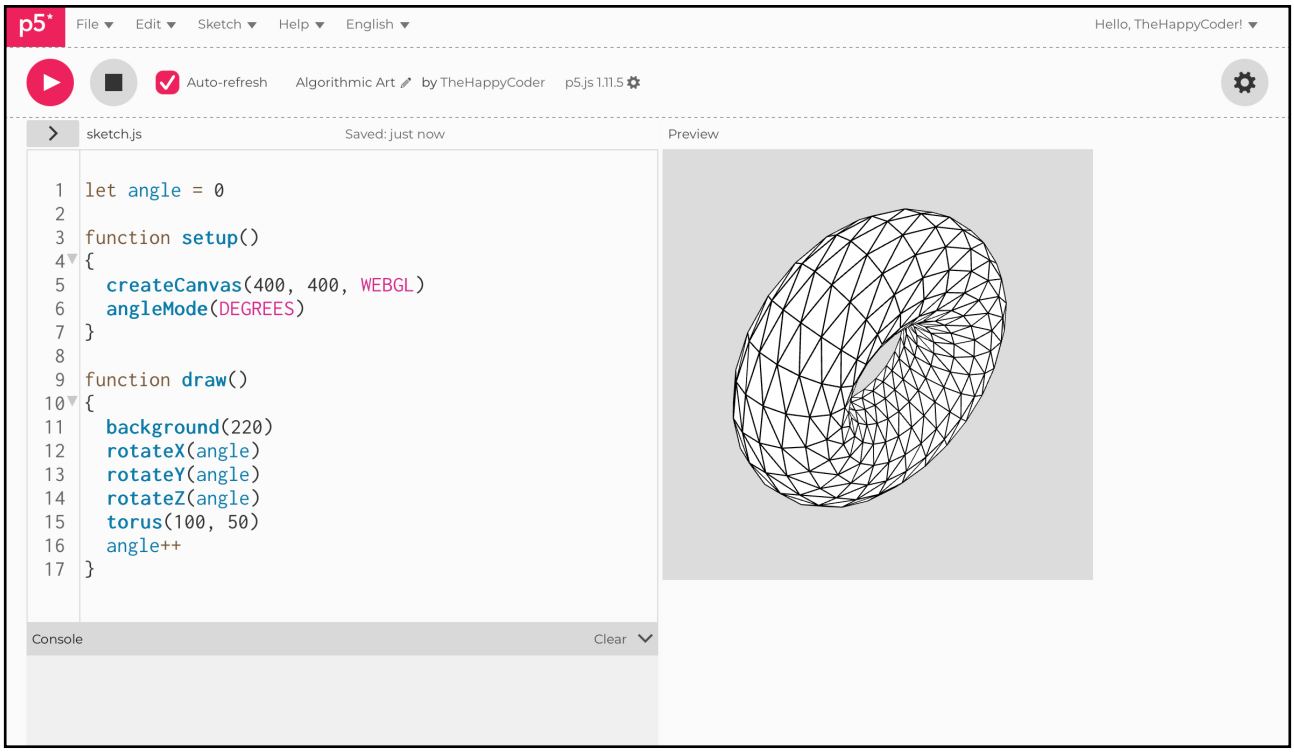
🌻 Challenge

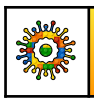
Play with the values to get a feel for the shape.

🛠️ Code Explanation

`torus(100, 50)`	A torus with a radius of 100 and a tube dimension of 50

Figure C1.13


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
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
torus(100, 50)
angle++
}
Console
Clear



Sketch C1.14 drawing a cone

A cone which has two dimensions, also, the first is the radius of the base and the second is the length (or height) of the cone.

```
let angle = 0

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

function draw()
{
  background(220)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  cone(100, 200)
  angle++
}
```

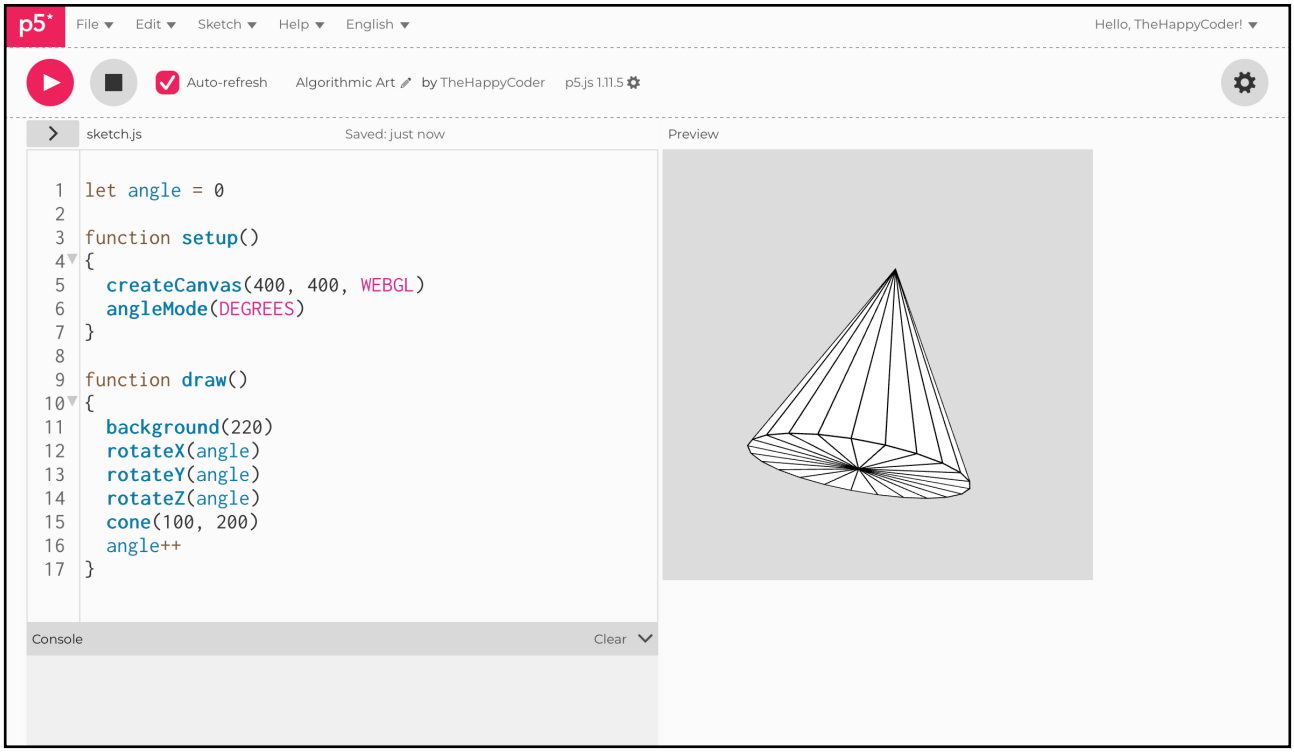
🌻 Challenges

1. Have more than one shape at the same time.
2. Use noFill().

🛠️ Code Explanation

cone(100, 200)	Draws a cone with a radius 100 and length 200

Figure C.14


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
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background(220)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
cone(100, 200)
angle++
}
Console
Clear

Sketch C1.15 adding a bit of colour

❗ remove cone()

Let's go back to our simple cube. We can add colour with `fill()` and give the `background()` some colour for good measure.

```
let angle = 0

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

function draw()
{
  background('darkred')
  fill('yellow')
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100)
  angle++
}
```

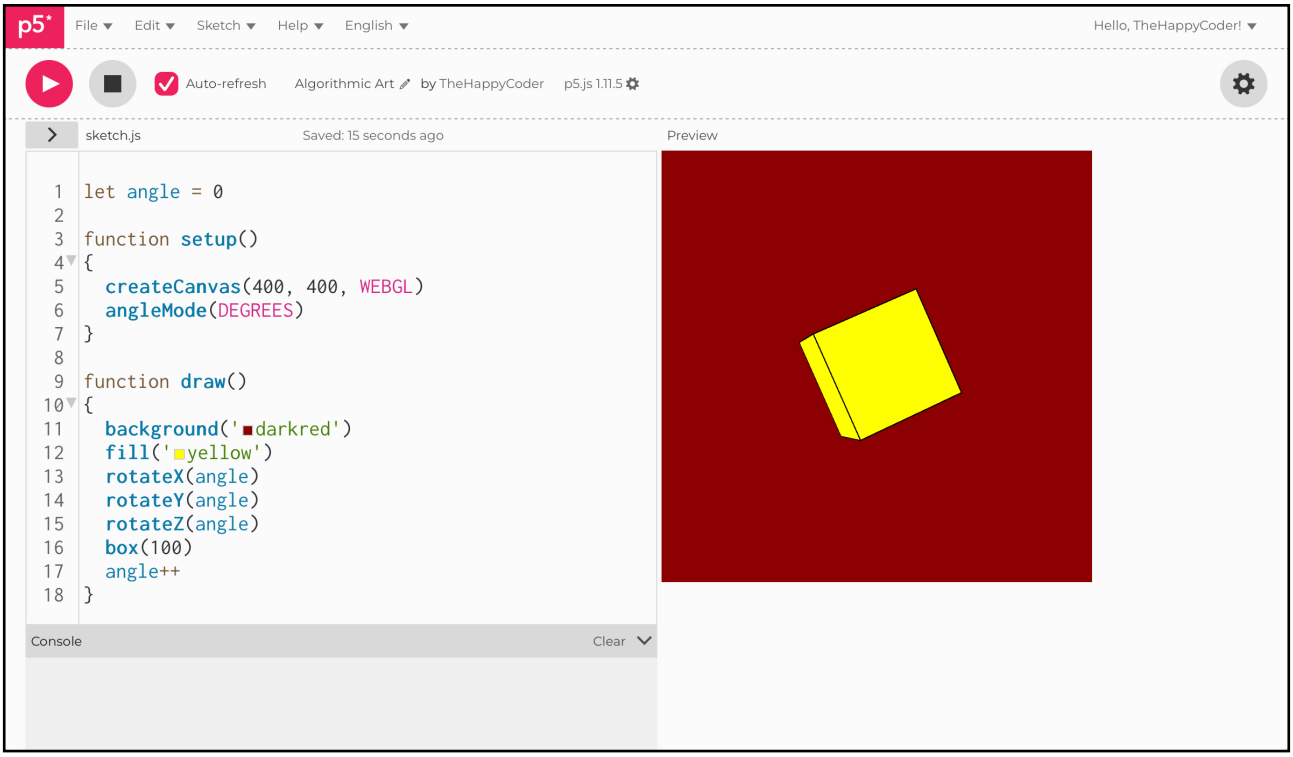
🗒️ Notes

This gives us a nice dark red background and a yellow cube. We will look at materials later.

🌻 Challenge

Explore other colours.

Figure C1.15


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: 15 seconds ago
Preview
let angle = 0
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background('darkred')
fill('yellow')
rotateX(angle)
rotateY(angle)
rotateZ(angle)
box(100)
angle++
}
Console
Clear

Sketch C1.16 translate

To move a shape in 3D (WEBGL), we translate the origin of the space. It takes a bit of getting used to as we are translating relative to the centre of the 3D space rather than from the top left as in the 2D canvas.

```
let angle = 0

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

function draw()
{
  background('darkred')
  fill('yellow')
  translate(100, 100)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100)
  angle++
}
```

Notes

This moves it down and to the right; notice that it still rotates about the new origin.

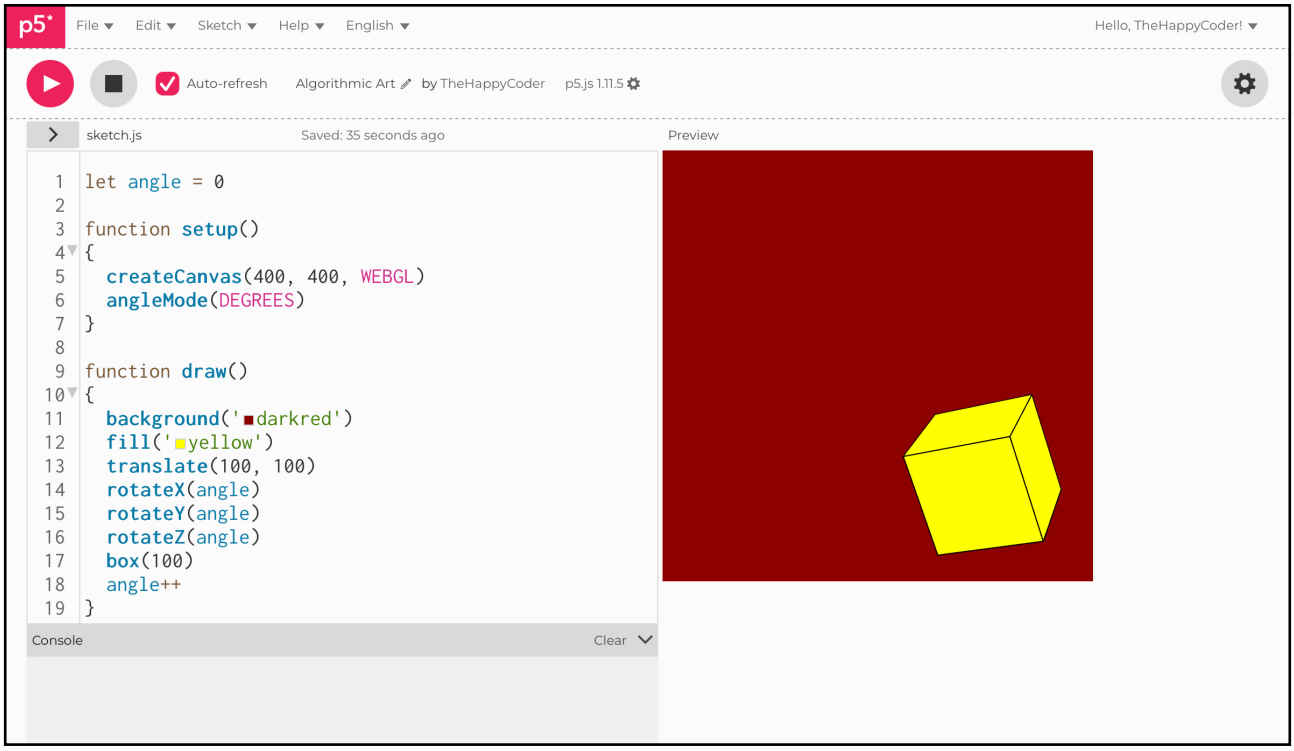
Challenges

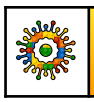
1. Try other translations.
2. Do you know how to translate it to the left?

🛠️ Code Explanation

`translate(100, 100)`	Translating relative to the original origin

Figure C1.16


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
let angle = 0
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background('darkred')
fill('yellow')
translate(100, 100)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
box(100)
angle++
}
Console
Clear



Sketch C1.17 translate `tother way

Translating it to the top left-hand corner

```
let angle = 0

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

function draw()
{
  background('darkred')
  fill('yellow')
  translate(-100, -100)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100)
  angle++
}
```

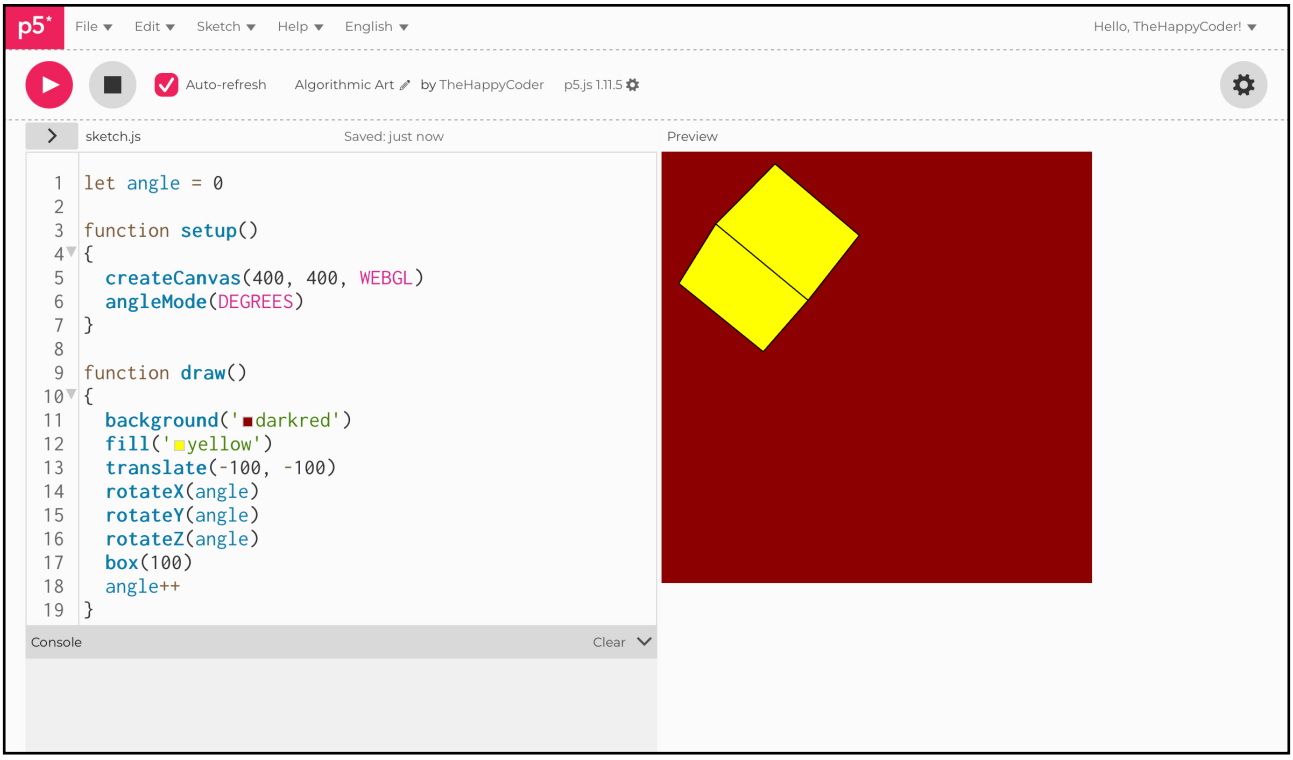
Notes

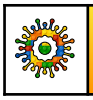
We need to translate it negatively relative to the old origin.

Code Explanation

`translate(-100, -100)`	Translating relative to the original origin

Figure C1.17


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
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background('darkred')
fill('yellow')
translate(-100, -100)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
box(100)
angle++
}
Console
Clear



Sketch C1.18 translate along the z axis

But what about the `z` axis? We will now translate along the `z` axis only; we still need all three arguments for translate.

```
let angle = 0

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

function draw()
{
  background('darkred')
  fill('yellow')
  translate(0, 0, 500)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100)
  angle++
}
```

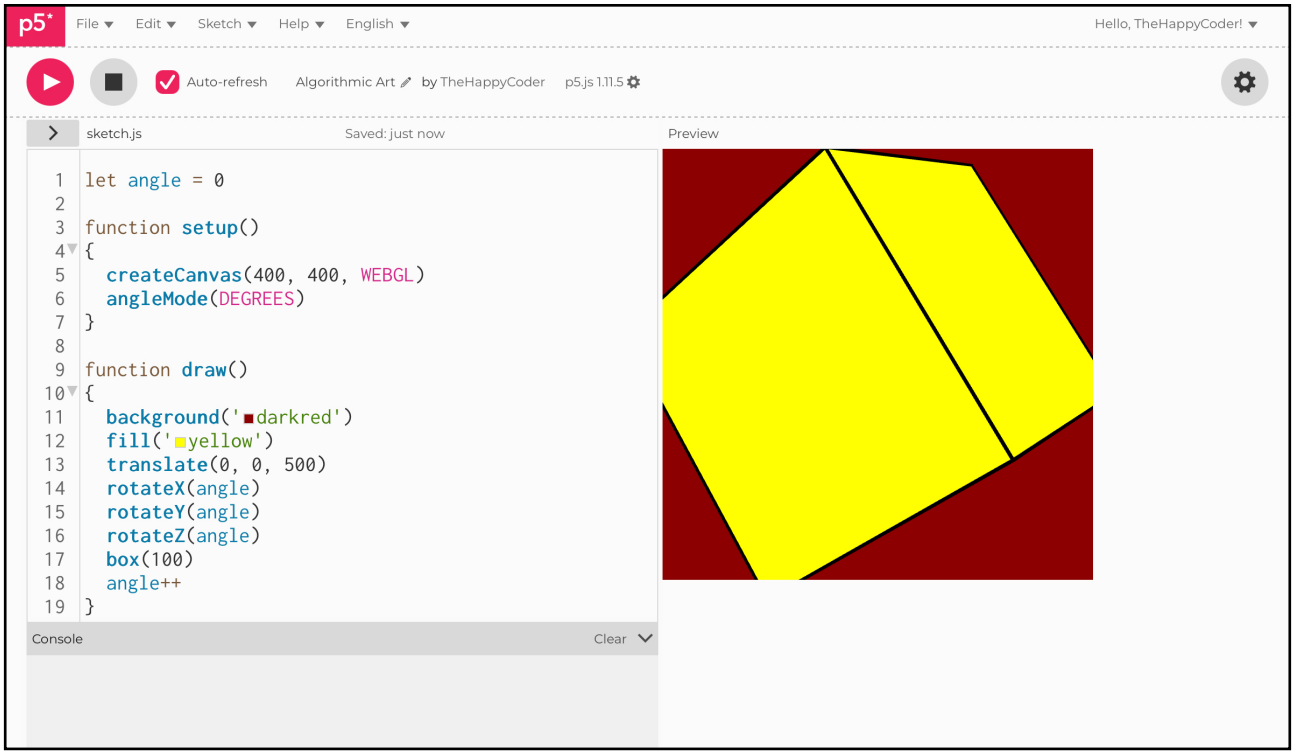
🗒️ Notes

This has moved it a lot closer; a positive value moves it towards you.

🛠️ Code Explanation

`translate(0, 0, 500)`	Translated only in the z direction, positive value brings it forwards

Figure C1.18


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
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background('darkred')
fill('yellow')
translate(0, 0, 500)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
box(100)
angle++
}
Console
Clear

Sketch C1.19 translate the other way

Giving it a negative value moves it further away; you are translating the space, not the shape, remember. So if you add other shapes, they too will be affected by the same amount.

```
let angle = 0

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

function draw()
{
  background('darkred')
  fill('yellow')
  translate(0, 0, -500)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100)
  angle++
}
```

🗒️ Notes

You can see how you can position the shape relative to the origin in all three planes (axes).

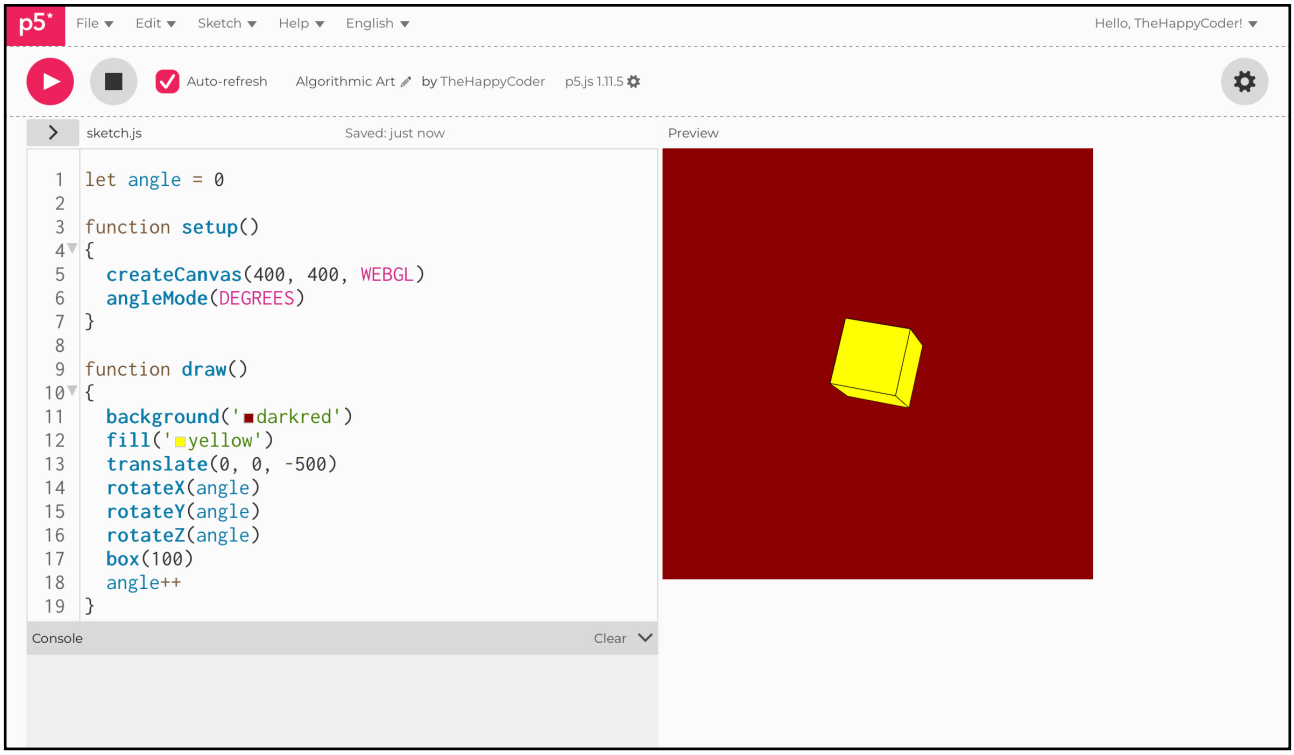
🌻 Challenge

Play with all three values.

🛠️ Code Explanation

`translate(0, 0, -500)`	Moves it further away by 500

Figure C1.19


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
function setup()
{
createCanvas(400, 400, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background('darkred')
fill('yellow')
translate(0, 0, -500)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
box(100)
angle++
}
Console
Clear

Sketch C1.20 more than one shape

But what happens if you have more than one shape occupying the same coordinates? Let's find out. We will add a cone and a torus to the box.

❗ Remove `translate()`

```
let angle = 0

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

function draw()
{
  background('darkred')
  fill('yellow')
  // translate(0, 0, -500)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100)
  torus(100, 25)
  cone(100, 200)
  angle++
}
```

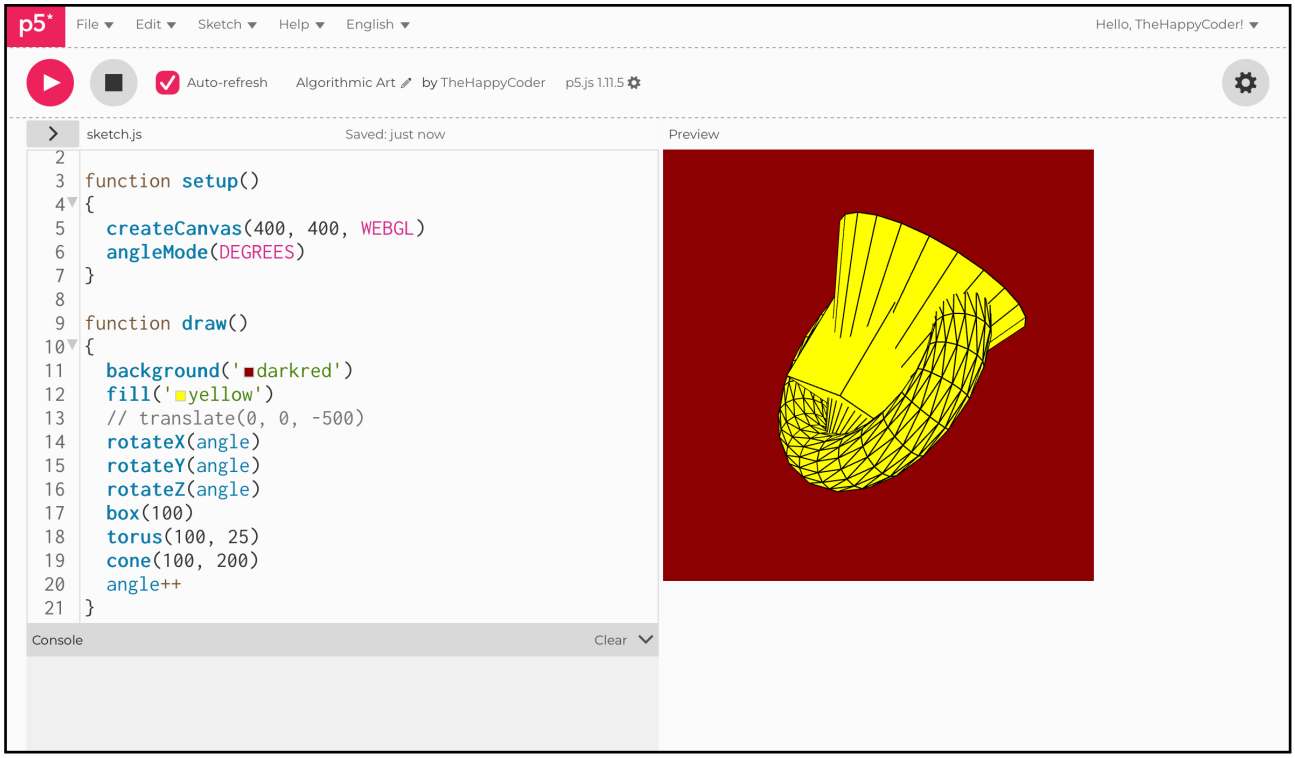
🗒️ Notes

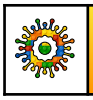
All three are meshed together.

🌻 Challenge

Try `noFill()` and `stroke('white')`.

Figure C1.20


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, WEBGL)
angleMode(DEGREES)
}
function draw()
{
background('darkred')
fill('yellow')
// translate(0, 0, -500)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
box(100)
torus(100, 25)
cone(100, 200)
angle++
}
Console
Clear



Sketch C1.21 pushing and popping

Using `push()` and `pop()` we can rotate and translate individual shapes (or groups of shapes).

```
let angle = 0

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

function draw()
{
  background('darkred')
  fill('yellow')
  // translate(0, 0, -500)
  rotateX(angle)
  rotateY(angle)
  rotateZ(angle)
  box(100)
  push()
  translate(0, 0, 100)
  rotateX(angle * 2)
  torus(100, 25)
  pop()
  cone(100, 200)
  angle++
}
```

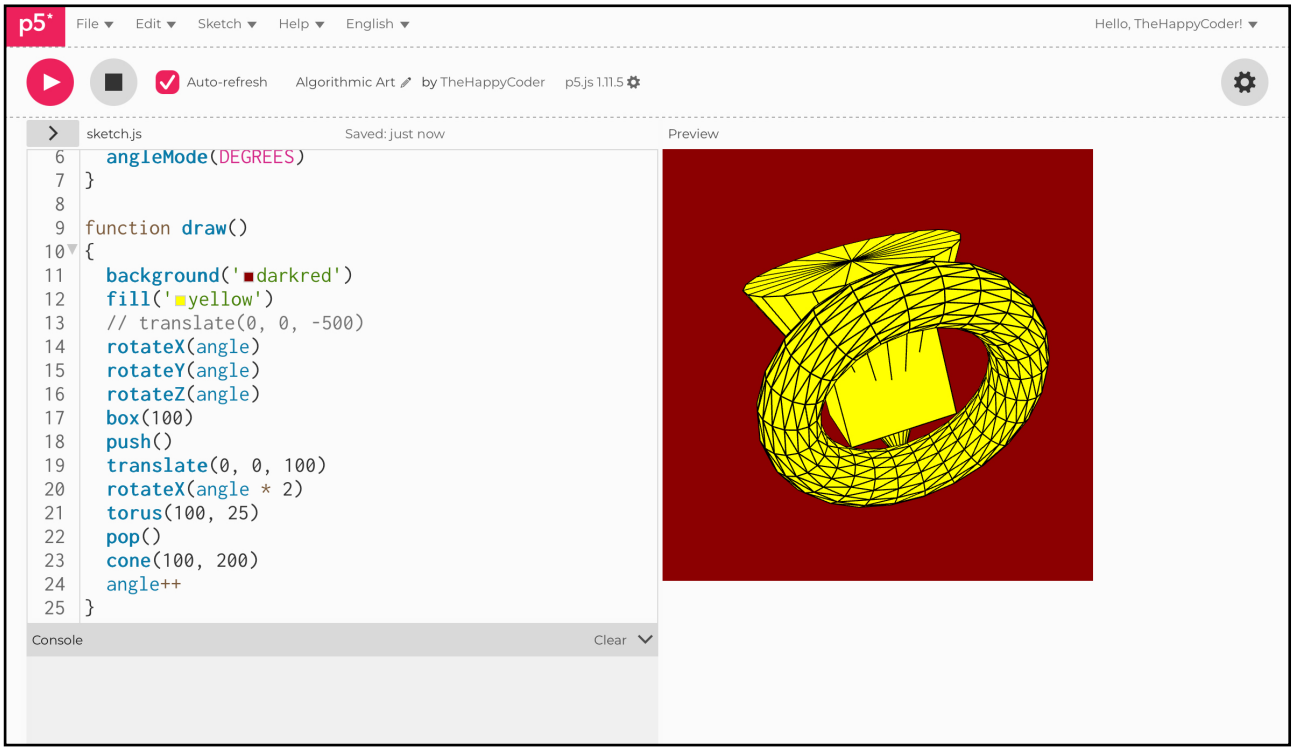
🗒️ Notes

The torus moves independently of the other two shapes, and yet all three are still moving together. There is a lot of scope to do much more.

🌻 Challenge

Just play and see what you can create.

Figure C1.21


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
angleMode(DEGREES)
}
function draw()
{
background('darkred')
fill('yellow')
// translate(0, 0, -500)
rotateX(angle)
rotateY(angle)
rotateZ(angle)
box(100)
push()
translate(0, 0, 100)
rotateX(angle * 2)
torus(100, 25)
pop()
cone(100, 200)
angle++
}
Console
Clear