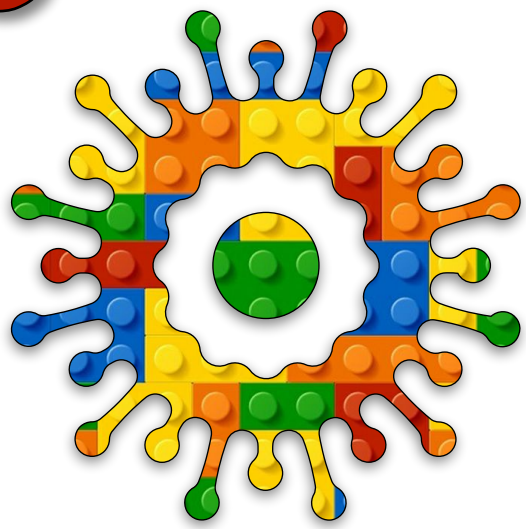
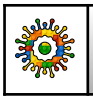
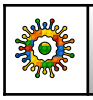
# Creative Coding Module A Unit #5 squares & rectangles





# Content

## Module A Unit #5 squares and rectangles



# Introduction to squares and rectangles

There is a lot to pack into this unit. Here we introduce the square and rectangle and show how we can manipulate these shapes, rotating and translating them. I will be introducing a lot of new concepts to you, but they will be used a lot in the units and examples to come. Just plough your way through the sketches and play with the code; this approach will help you understand what it is doing. I will give you the framework, but it is up to you to explore.

## Key concepts

- square()
- rotate()
- angleMode()
- translate()
- push()
- pop()
- rectMode()
- rectangle()
- comments

# Sketch A5.1 a simple square

❗ our starting sketch

We can draw a simple square in the middle of the canvas. The `square()` function has three arguments: the x co-ordinate, the y co-ordinate and the length of the sides.

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

function draw()
{
  square(200, 200, 100)
}
```

## 🗒️ Notes

The only problem is, it isn't in the centre of the canvas. This is because the co-ordinates of the square are taken from the top left-hand corner, not the centre of the square, but we can fix that quite easily.

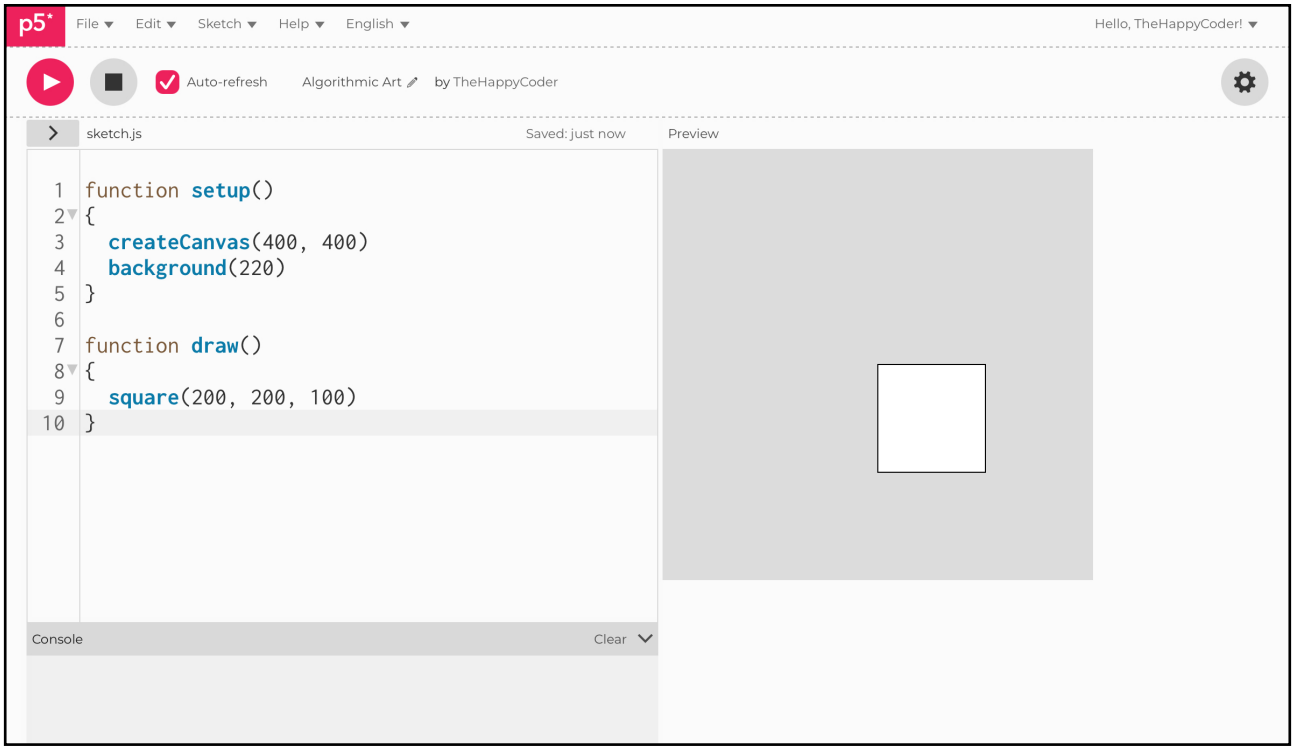
## 🌻 Challenges

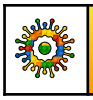
1. What would be the co-ordinates to put the square in the centre of the canvas?
2. Draw more squares.

## 🛠️ Code Explanation

| | |
|---|---|
| `square(200, 200, 100)` | Draws a square at (200, 200) with a length of side 100 pixels |

Figure A5.1


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 background(220)
5 }
6
7 function draw()
8 {
9 square(200, 200, 100)
10 }
Console
Clear



# Sketch A5.2 from the centre

Using a function called `rectMode()`, we can move the co-ordinates to the centre.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  rectMode(CENTER)
}

function draw()
{
  square(200, 200, 100)
}
```

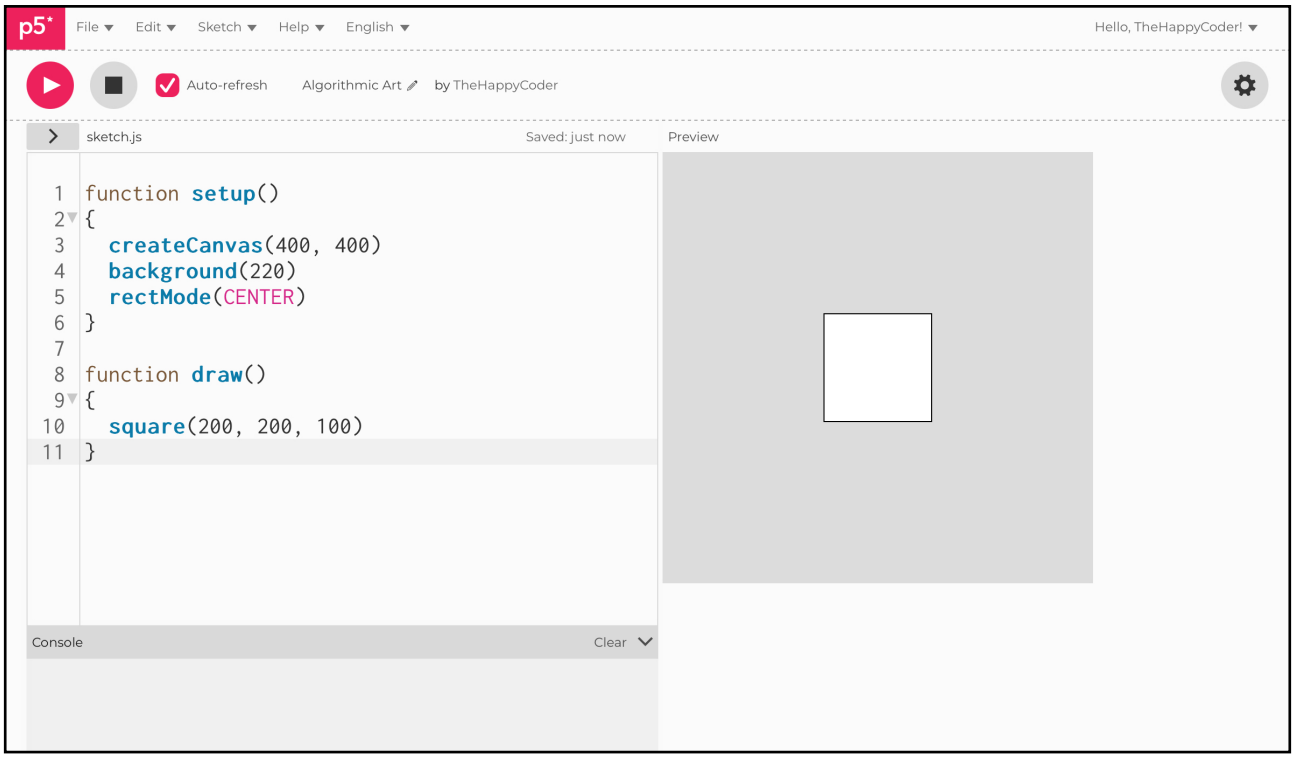
## Notes

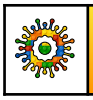
Now it is nicely in the centre!

## Code Explanation

| | |
|---|---|
| `rectMode(CENTER)` | This function needs the information in the brackets (which needs to be uppercase and US spelling) |

Figure A5.2


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 background(220)
5 rectMode(CENTER)
6 }
7
8 function draw()
9 {
10 square(200, 200, 100)
11 }
Console
Clear



# Sketch A5.3 rotate

We can rotate the square now because we have the centre in the centre, if you get my meaning; otherwise, it would rotate around the corner. By default, the angle of rotation is measured in `radians`; however, we can change that to `degrees` with `angleMode()`.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  rectMode(CENTER)
  angleMode(DEGREES)
}

function draw()
{
  rotate(45)
  square(200, 200, 100)
}
```

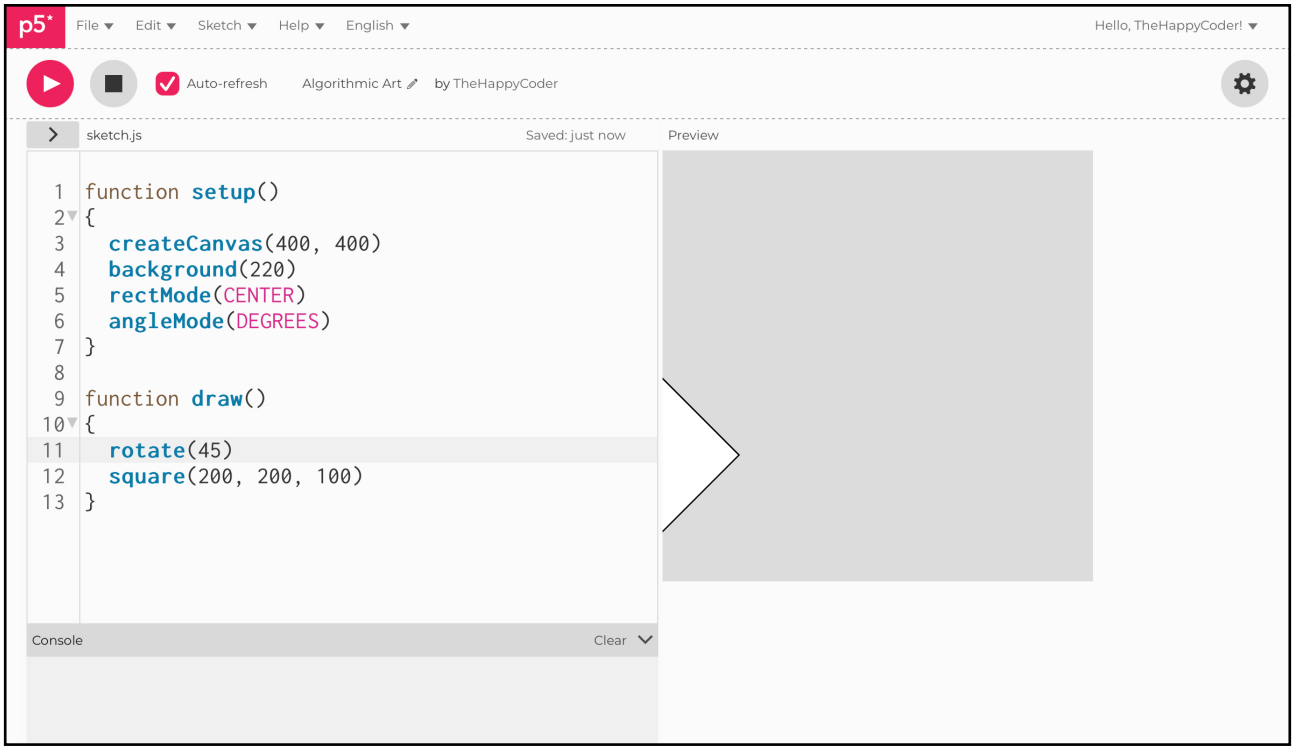
## 🗒️ Notes

As you will see, there is a problem. It has rotated about the corner of the canvas, not the centre of the square. The good news is we have a solution!

## 🛠️ Code Explanation

| | |
|---|---|
| `angleMode(DEGREES)` | Converts from radians to degrees |
| `rotate(45)` | Rotate by 45° |

Figure A5.3


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 background(220)
5 rectMode(CENTER)
6 angleMode(DEGREES)
7 }
8
9 function draw()
10 {
11 rotate(45)
12 square(200, 200, 100)
13 }
Console
Clear

# Sketch A5.4 translate

In order to solve this problem, we have to move the origin of the canvas from the top-left-hand corner to the centre of the canvas. To do this, we use the `translate()` function to move it. We then have to change the co-ordinates of the square to `(0, 0)`.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  rectMode(CENTER)
  angleMode(DEGREES)
}

function draw()
{
  translate(200, 200)
  rotate(45)
  square(0, 0, 100)
}
```

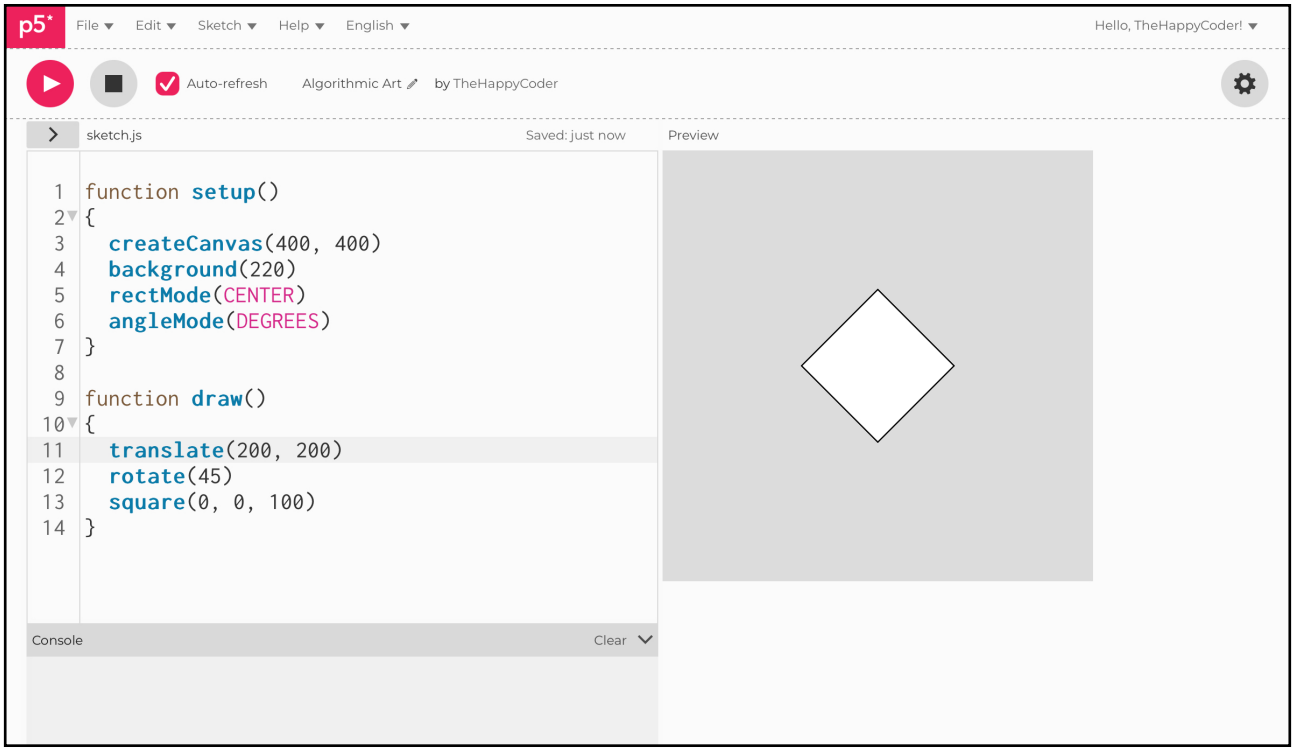
## Notes

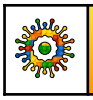
It is now rotating around the centre of the square.

## Code Explanation

| | |
|---|---|
| `translate(200, 200)` | Moves the origin (0, 0) from the top left corner to the centre of the canvas |
| `square(0, 0, 100)` | The coordinates now reflect the changes we have made through translating the origin |

Figure A5.4


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 background(220)
5 rectMode(CENTER)
6 angleMode(DEGREES)
7 }
8
9 function draw()
10 {
11 translate(200, 200)
12 rotate(45)
13 square(0, 0, 100)
14 }
Console
Clear



# Sketch A5.5 no fill and an angle

We will use a variable for the angle called, strangely enough, `angle`. We can add another function to clear any fill colour (default is white) from the square called `noFill()`.

```
let angle = 45

function setup()
{
  createCanvas(400, 400)
  background(220)
  rectMode(CENTER)
  angleMode(DEGREES)
  noFill()
}

function draw()
{
  translate(200, 200)
  rotate(angle)
  square(0, 0, 100)
}
```

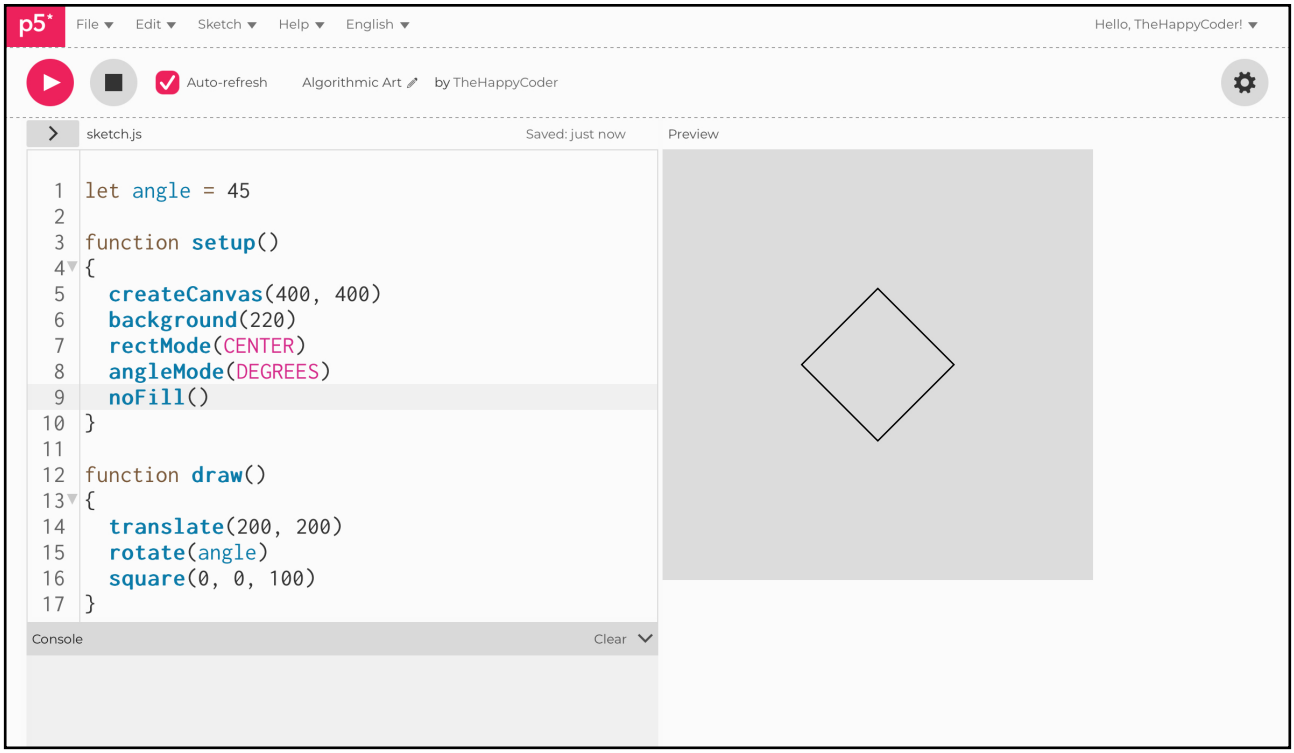
## 🗒️ Notes

We get the same effect of a square at 45° but this time without the default fill colour.

## 🛠️ Code Explanation

| | |
|---|---|
| `noFill()` | Simply means there is no colour at all, making it completely transparent. |

Figure A5.5


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let angle = 45
function setup()
{
createCanvas(400, 400)
background(220)
rectMode(CENTER)
angleMode(DEGREES)
noFill()
}
function draw()
{
translate(200, 200)
rotate(angle)
square(0, 0, 100)
}
Console
Clear

# Sketch A5.6 rotating it slowly

We move the `background()` into the `draw()` function. We can use the `//` symbol to effectively remove the line of code. We add `1` to the angle on each iteration of the `draw()` loop. The effect is that the square slowly rotates.

❗ comment out the `background()` in the `setup()` function

```
let angle = 45

function setup()
{
  createCanvas(400, 400)
  // background(220)
  rectMode(CENTER)
  angleMode(DEGREES)
  noFill()
}

function draw()
{
  background(220)
  translate(200, 200)
  rotate(angle)
  square(0, 0, 100)
  angle++
}
```

## Notes

Commenting out (//) is a very useful way of leaving code in but where you don't want to use it, or use it at a later date. The computer will ignore any lines that start with //. You can leave notes for yourself or someone else to help explain what your code is doing.

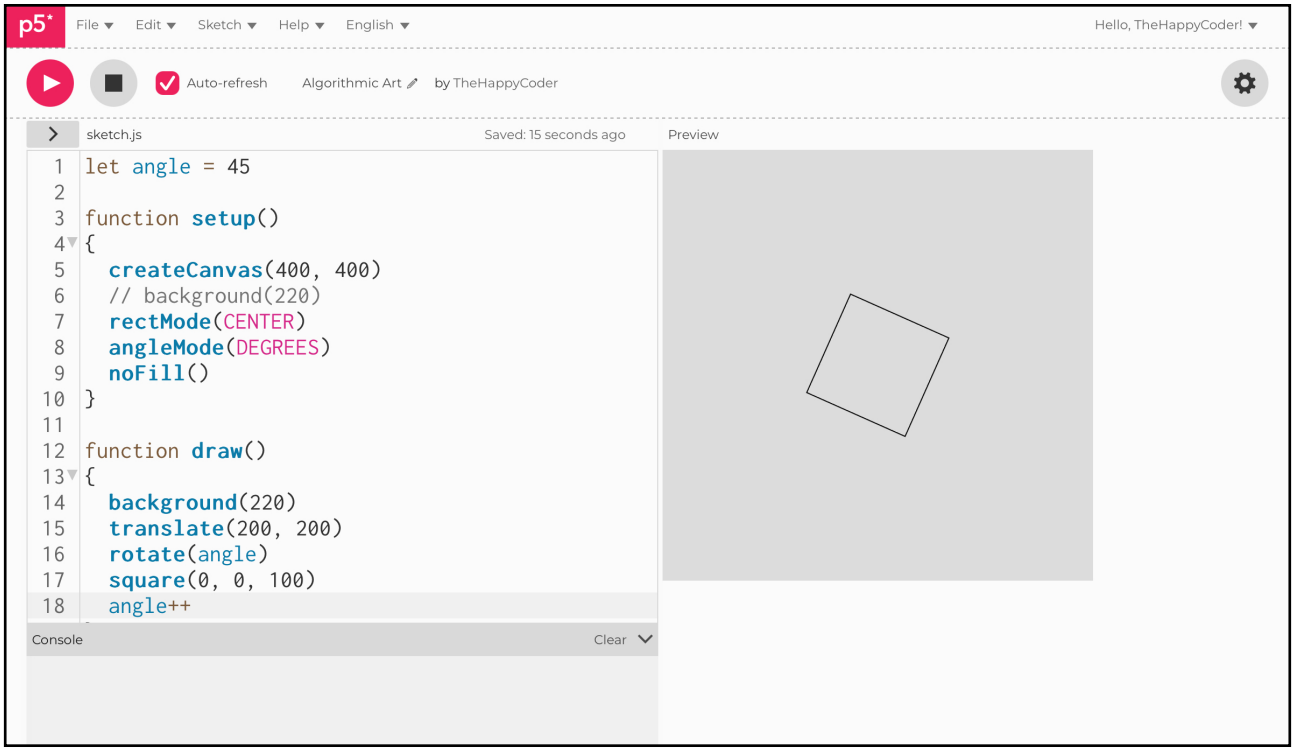
## 🌻 Challenge

Try commenting out other lines of code.

## 🛠️ Code Explanation

| | |
|---|---|
| // | The // symbols mean that the computer does not see anything on that line as code (as if invisible) |

Figure A5.6 slowly rotates



# Sketch A5.7 adding a second square

❗ remove the background() in the setup() function.

If we add another square, the rotation is applied to it as well. Here, there are two squares, but they are both rotating together; hence, it looks as if there is just one square.

```
let angle = 45

function setup()
{
  createCanvas(400, 400)
  rectMode(CENTER)
  angleMode(DEGREES)
  noFill()
}

function draw()
{
  background(220)
  translate(200, 200)
  rotate(angle)
  square(0, 0, 100)
  square(0, 0, 100)
  angle++
}
```

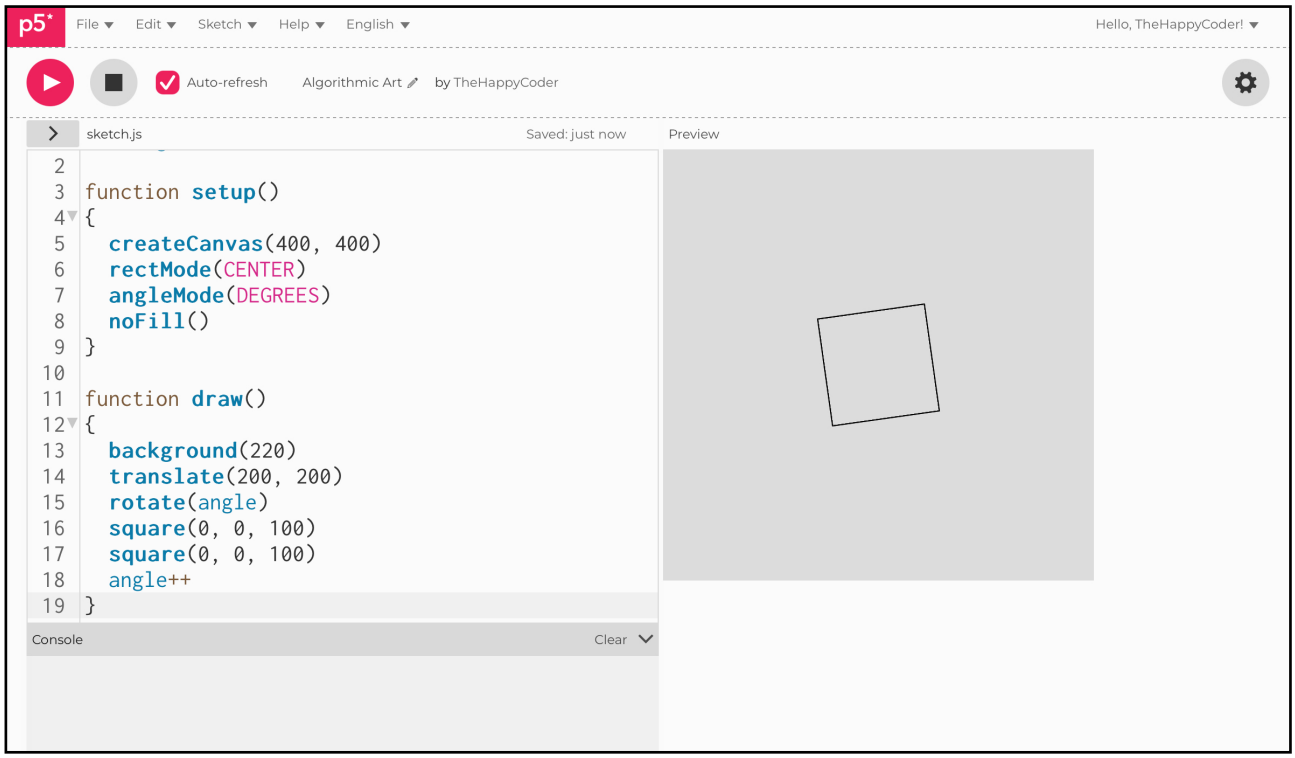
## 🗒️ Notes

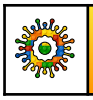
All you get is what looks like a square slowly rotating as before. What you have is both rotating at the same time; there are two squares. We want to stop one and let the other rotate.

## 🌻 Challenges

1. Change the speed of the rotation (`angle += 2`)
2. Change the direction

Figure A5.7


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
2
3 function setup()
4 {
5 createCanvas(400, 400)
6 rectMode(CENTER)
7 angleMode(DEGREES)
8 noFill()
9 }
10
11 function draw()
12 {
13 background(220)
14 translate(200, 200)
15 rotate(angle)
16 square(0, 0, 100)
17 square(0, 0, 100)
18 angle++
19 }
Console
Clear



# Sketch A5.8 push and pop

If we want to separate the two squares and have one rotate and the other not, we use two functions (as a pair) called `push()` and `pop()`. They are like bookends where you take the current state, change it, (rotate, for instance), and then return the code back to its previous state. What happens between `push()` and `pop()` stays between `push()` and `pop()`.

```
let angle = 45

function setup()
{
  createCanvas(400, 400)
  rectMode(CENTER)
  angleMode(DEGREES)
  noFill()
}

function draw()
{
  background(220)
  translate(200, 200)
  push()
  rotate(angle)
  square(0, 0, 100)
  pop()
  square(0, 0, 100)
  angle++
```

## Notes

We now have one static square and a rotating square.

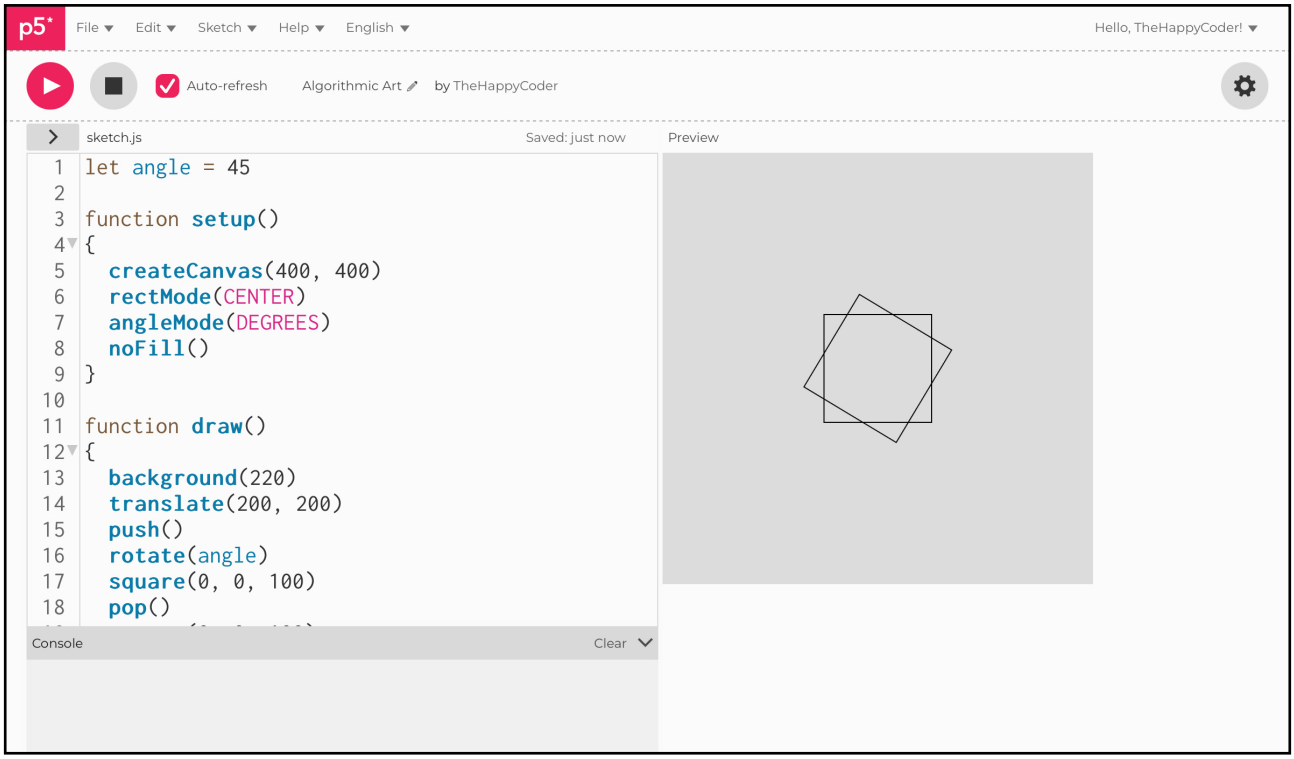
## 🌻 Challenge

Add a third square and have it rotate backwards (`-angle`), don't forget `push()` and `pop()`.

## 🛠️ Code Explanation

| | |
|---|---|
| `push()` | The push() function collects the state of everything at that moment in the code |
| `pop()` | After you have added other functionality, the pop() function returns the original state when you introduced the push() function |

Figure A5.8


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
sketch.js
Saved: just now
Preview
let angle = 45
function setup()
{
createCanvas(400, 400)
rectMode(CENTER)
angleMode(DEGREES)
noFill()
}
function draw()
{
background(220)
translate(200, 200)
push()
rotate(angle)
square(0, 0, 100)
pop()
Console
Clear

# Sketch A5.9 a rectangle

❗ start a new sketch

Instead of a square, we have a rectangle. To achieve this, we add another argument and call the `rect()` function. The first two arguments are the coordinates, the third is the width, and the fourth is the height of the rectangle.

```
function setup()
{
  createCanvas(400, 400)
  background(220)
  rectMode(CENTER)
}

function draw()
{
  rect(200, 200, 300, 50)
}
```

## 🗒️ Notes

If you only give `rect()` three arguments, it will draw a square.

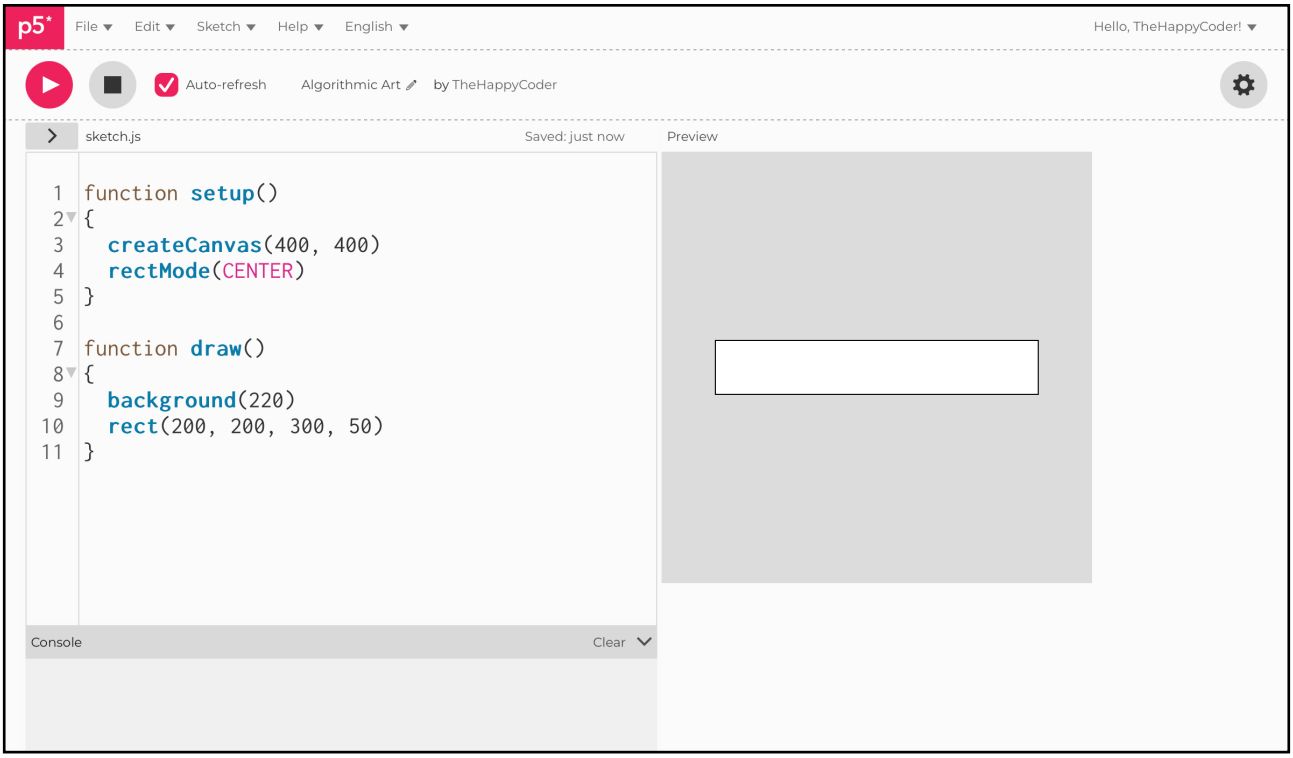
## 🌻 Challenge

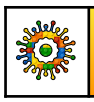
Try other dimensions for sizes.

## 🛠️ Code Explanation

| | |
|---|---|
| `rect(200, 200, 300, 50)` | The rectangle has four arguments, the x coordinate, the y coordinate, the width and the height of the rectangle |

Figure A5.9


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 rectMode(CENTER)
5 }
6
7 function draw()
8 {
9 background(220)
10 rect(200, 200, 300, 50)
11 }
Console
Clear



# Sketch A5.10 random rectangles

Let's go a little mad with random and create a nice pattern with the rectangles. We randomise the coordinates between 100 and 300, and give the width and height a random dimension of 50. We make the background white.

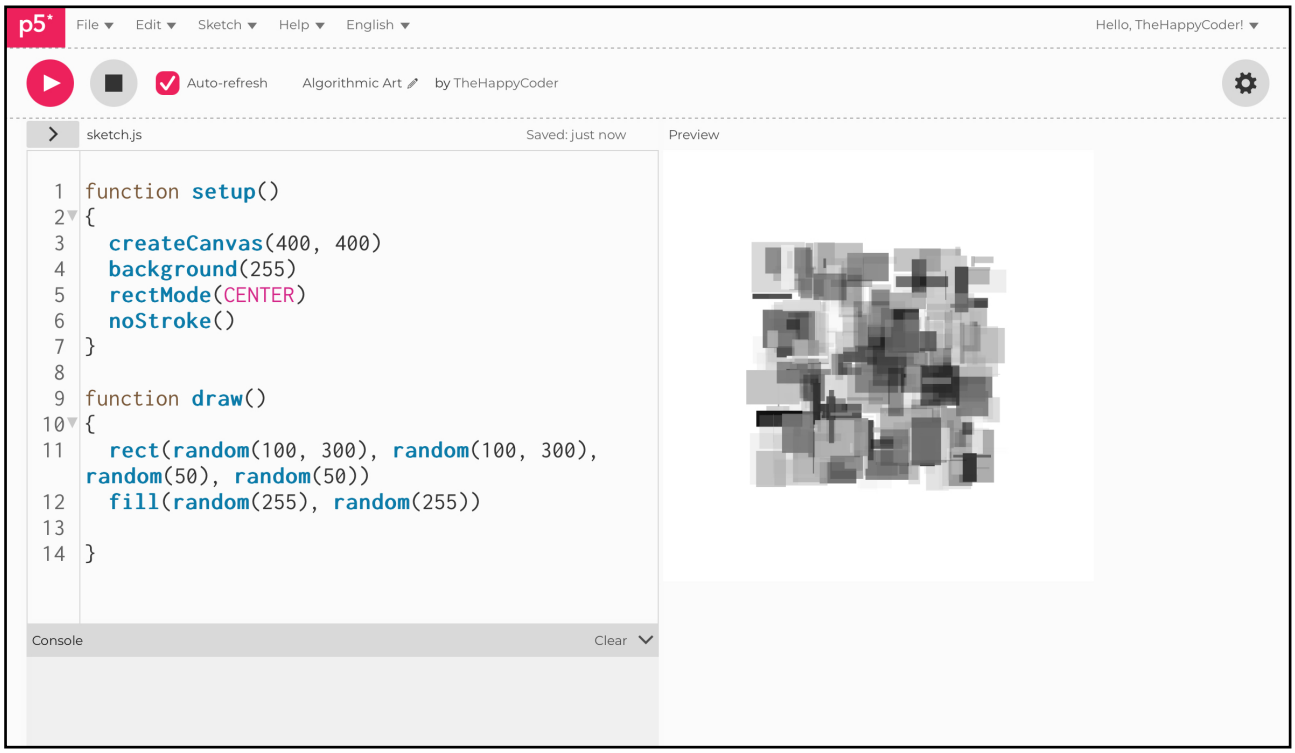
```
function setup()
{
  createCanvas(400, 400)
  background(255)
  rectMode(CENTER)
  noStroke()
}


function draw()
{
  rect(random(100, 300), random(100, 300), random(50),
random(50))
  fill(random(255), random(255))
}
```

## Challenges

1. Change the random dimensions.
2. Have it draw just 100 rectangles.
3. Add colour.

Figure A5.10


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 background(255)
5 rectMode(CENTER)
6 noStroke()
7 }
8
9 function draw()
10 {
11 rect(random(100, 300), random(100, 300),
random(50), random(50))
12 fill(random(255), random(255))
13
14 }
Console
Clear

## Next. . .

Although we will draw some more shapes, we will also make things a bit more interactive in unit #6.