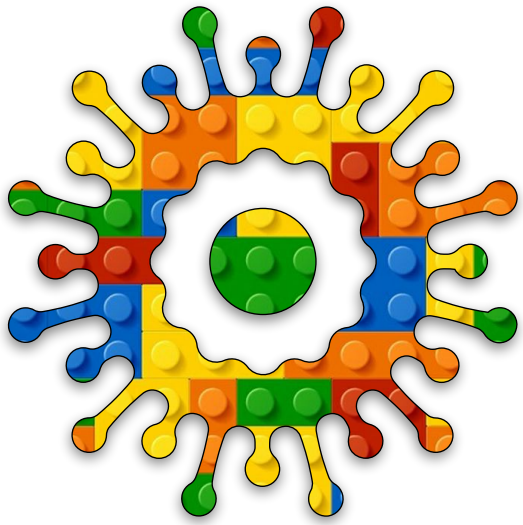
Creative Coding Module B Unit #1 bounce & rotate



Content

Module B Unit #1 move and oscillate

Introduction to bounce and rotate

In this unit, we will get shapes moving and create patterns with some simple algorithms. We will start with moving a circle across the canvas, only to reappear on the other side. Then, we will look at bouncing it off the edge of the canvas. Finally, we will look at rotation, angular velocity, and the acceleration of a spinning baton.

Key Concepts:

- angular velocity
- angular acceleration
- using the OR comparison
- bouncing off an edge

Sketch B1.1 starting sketch

❗ Start a new sketch

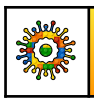
```
function setup()
{
  createCanvas(400, 400)
}

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

🗒️ Notes

Our simple starting sketch

🌻 Challenge

You can add some colour or different size canvas



Sketch B1.2 draw a circle

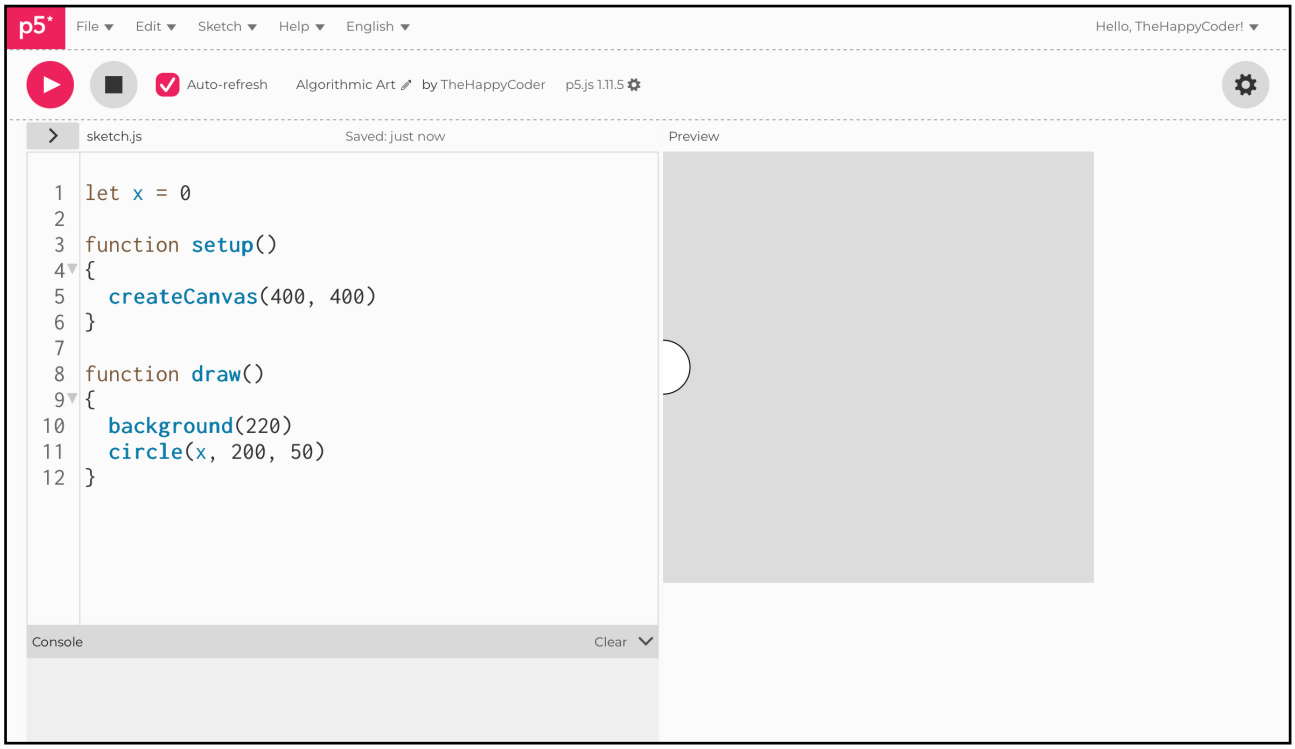
We will create a variable for the X co-ordinate, and draw a circle.

```
let x = 0

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

function draw()
{
  background(220)
  circle(x, 200, 50)
}
```

Notes

Our circle starts on the left hand edge of the canvas

Figure B1.2


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
let x = 0
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
circle(x, 200, 50)
}
Console
Clear

Sketch B1.3 increment

Now we can increment the variable `x` and make it move slowly across the canvas, one pixel at a time.

```
let x = 0

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

function draw()
{
  background(220)
  circle(x, 200, 50)
  x = x + 1
}
```

🗒️ Notes

You will notice that it goes off the canvas (eventually), never to be seen again.

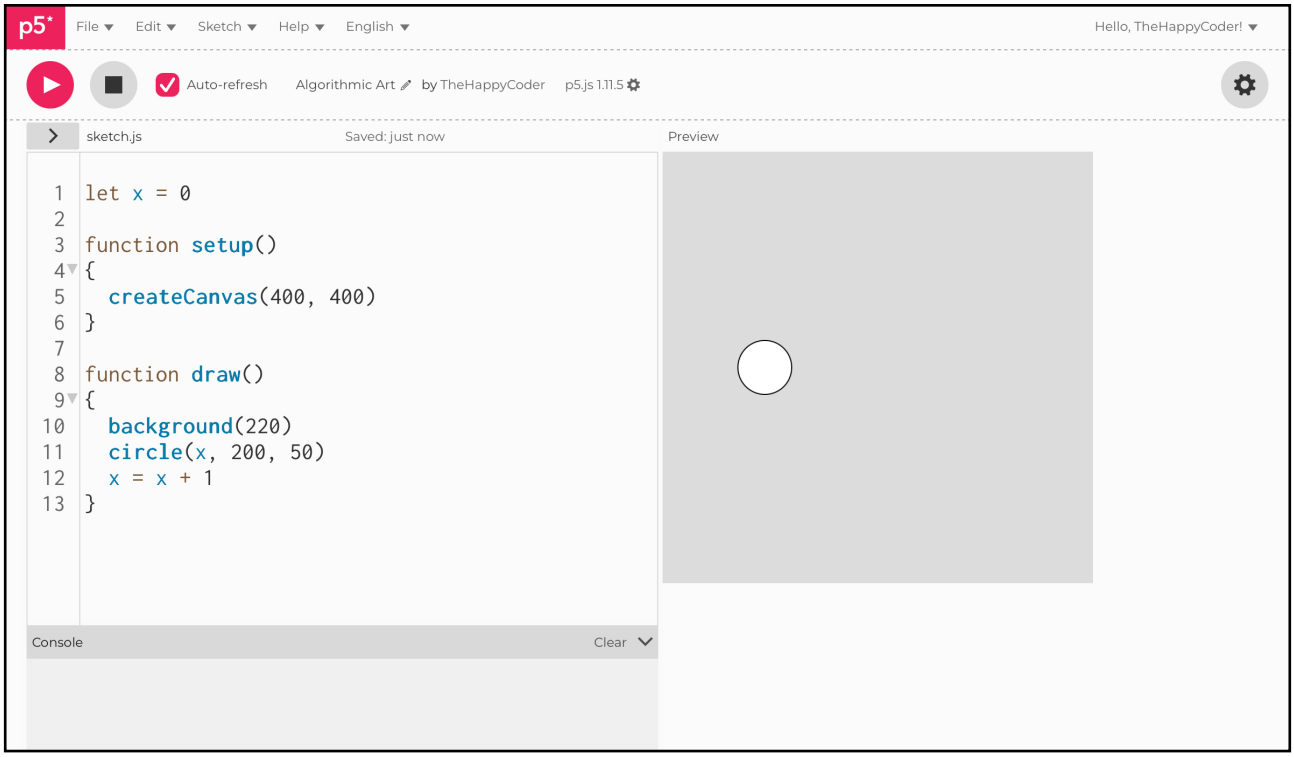
🌻 Challenge

How do you think you could make it move faster?

🛠️ Code Explanation

`x = x + 1`	Adds 1 to the value of x on each iteration

Figure B1.3


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
let x = 0
function setup()
{
createCanvas(400, 400)
}
function draw()
{
background(220)
circle(x, 200, 50)
x = x + 1
}
Console
Clear

Sketch B1.4 a short cut

We can simplify the increment and also increase it from one pixel to five.

```
let x = 0

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

function draw()
{
  background(220)
  circle(x, 200, 50)
  x += 5
}
```

Notes

Moves a lot faster now.

Code Explanation

x += 5	Same as x = x + 5

Sketch B1.5 using an if statement

The problem with the previous sketch is that the circle just wanders off the edge of the canvas. If we want it to start again once it has reached the right-hand edge, then an `if()` statement is what you need. In simple terms: if the circle is at the right-hand edge, start again.

```
let x = 0

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

function draw()
{
  background(220)
  circle(x, 200, 50)
  x += 5
  if (x == 400)
  {
    x = 0
  }
}
```

Notes

This uses an if statement. Notice that when it gets to the right-hand edge, `x` has the value `400` (`x == 400`). It then resets the value of `x` to `0`, which returns it to the start.

Challenge

1. What happens when `x = x + 7`?
2. What could you do about that? Hint: change `x == 400` to `x >= 400`.

🛠️ Code Explanation

`if(x == 400)`	If x is exactly 400 then make it equal to zero (x = 0)

Sketch B1.6 some velocity

We need another variable; we will call it `velocity`. We can use this variable to good effect.

❗ Note, we have also made it so that the `==` (is equal to) symbol is now `>=` (is greater than or equals to).

```
let x = 0
let velocity = 7

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

function draw()
{
  background(220)
  circle(x, 200, 50)
  x += velocity
  if(x >= 400)
  {
    x = 0
  }
}
```

🗒️ Notes

The benefit of having the greater than or equals to (`>=`) symbol is that you can move it as fast as you want. If we didn't change the condition, it would be moving too fast and would effectively jump over the edge of the canvas.

🌻 Challenges

1. What happens if you leave the conditional as ==?
2. Can you make it bounce off the right-hand edge?

🛠️ Code Explanation

`let velocity = 7`	The velocity is initialised to 7
`x += velocity`	Increment the position of the circle by that amount
`if(x >= 400)`	If x is greater than or equal to 400 then set x to zero

Sketch B1.7 bounce off the edge

In this sketch, we go one step better. Instead of starting all over again, we can use an `if()` statement to make the circle bounce off the right-hand edge. To do this, we need another variable called `velocity`, which we can use to change the direction. The variable name `velocity` is one we have just made up.

```
let x = 0
let velocity = 7

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

function draw()
{
  background(220)
  circle(x, 200, 50)
  x += velocity
  if(x >= 400)
  {
    velocity = -velocity
  }
}
```

Notes

This time, rather than adding `1`, we have created a variable called `velocity`, which we start by giving it the value `7`. This has the same effect of adding `7` to `x` each time. When the circle reaches the edge and `x` is greater than or equal to `400` (`x >= 400`), then `7` becomes `-7`, and so it subtracts `7` each time from `x`. At this point, it disappears off the canvas again. The benefit of having the greater than or equals to (`>=`) symbol is that you can move it as fast as you want.

🌻 Challenge

Can you make it bounce off the left-hand edge as well? See the next sketch to find out.

🛠️ Code Explanation

`velocity = -velocity`	This reverse the direction

Sketch B1.8 bounce from every side

Not satisfied with bouncing off one side, how about both edges? To do this, we need a logic OR comparison. The direction will change whether it reaches the right-hand or left-hand edge of the canvas. You will need to locate the | symbols (called pipes ||) on your keyboard; you may find that with some keyboards they are two vertical lines, one on top of the other.

```
let x = 0
let velocity = 7

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

function draw()
{
  background(220)
  circle(x, 200, 50)
  x += velocity
  if (x >= 400 || x <= 0)
  {
    velocity = -velocity
  }
}
```

Notes

You should have the circle bouncing off each side. This `if()` statement has two conditions, the first one is for the right-hand edge `x >= 400`, and the second is the left-hand edge `x <= 0`. The two vertical lines (||) are the symbols for OR, which means if either is true. The symbols are called pipes for some reason.

🌻 Challenge

How could you make the circle get bigger towards the centre and smaller as it approaches the edges? It is doable but requires some mental gymnastics.

🛠️ Code Explanation

`if (x >= 400 || x <= 0)`	If x is greater then or equal to 400 OR x is less than or equal to 0 then reverse the direction

Sketch B1.9 bigger and smaller

Making the circle expand and contract as it moves across the canvas.

```
let x = 0
let velocity = 7
let diameter = 0

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

function draw()
{
  background(220)
  circle(x, 200, diameter)
  x += velocity
  if (x >= 400 || x <= 0)
  {
    velocity = -velocity
  }
  if (x <= 200)
  {
    diameter = x / 2
  }
  if (x >= 200)
  {
    diameter = (400 - x) / 2
  }
}
```

🗒️ Notes

This is a good time to work through the maths here. `x` is the variable that is changing from `0` to `400` as it moves across the canvas and vice versa on the way back.

🌻 Challenges

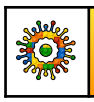
1. Try some other values.
2. Experiment with other shapes.

🛠️ Code Explanation

`diameter = x / 2`	The diameter is dependant on this value in the left hand half of the canvas
`diameter = (400 - x) / 2`	The diameter is dependant on this value in the right hand half of the canvas

Spinning baton

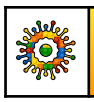
In this section, we are going to create a baton (simply a line with a circle at each end) and try to spin it. We will introduce angular velocity and angular acceleration to the baton.



Sketch B1.10 new sketch

❗ start a new sketch

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

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



Sketch B1.11 translate the line

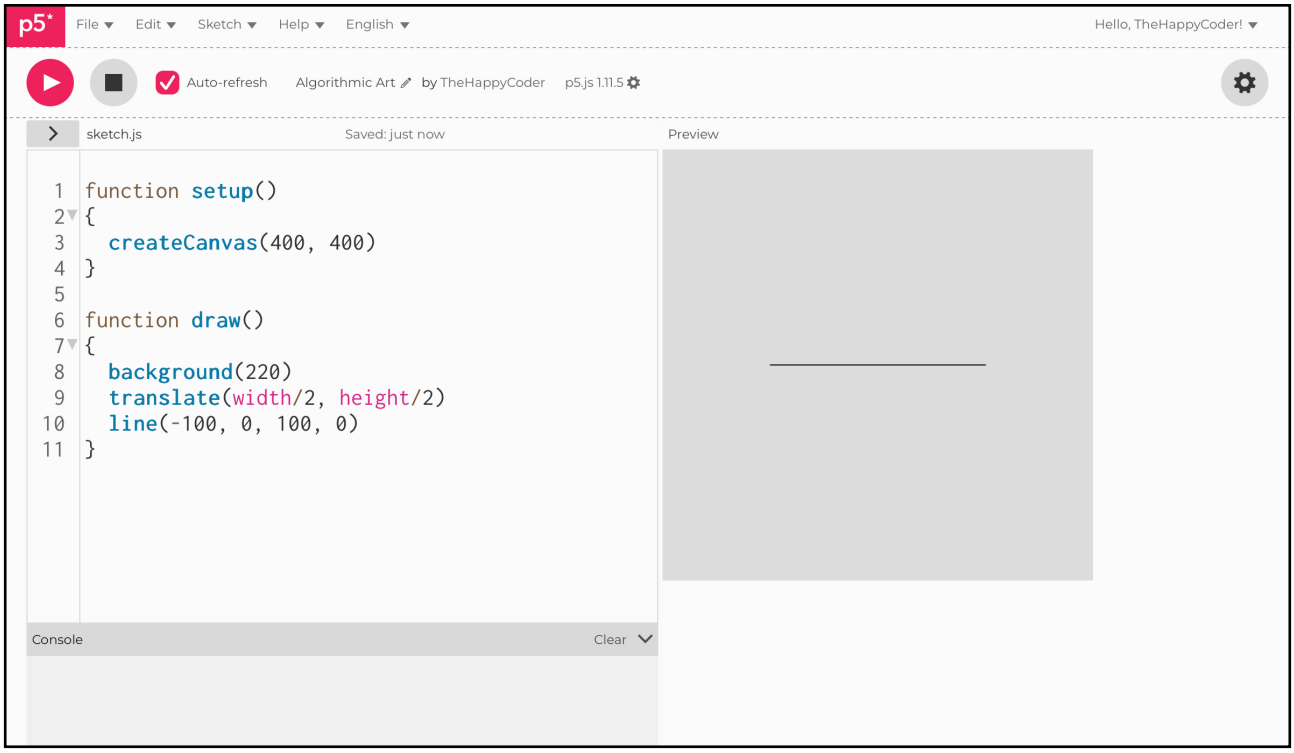
We translate the origin to the centre of the canvas and draw a line so that it straddles the centre.

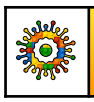
```
function setup()
{
  createCanvas(400, 400)
}

function draw()
{
  background(220)
  translate(width/2, height/2)
  line(-100, 0, 100, 0)
}
```

Notes

This puts the origin in the centre of the canvas, the co-ordinates (the four arguments) of the line reflect this. If you are a little confused, I suggest sketching it out.

Figure B1.11


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
1 function setup()
2 {
3 createCanvas(400, 400)
4 }
5
6 function draw()
7 {
8 background(220)
9 translate(width/2, height/2)
10 line(-100, 0, 100, 0)
11 }
Console
Clear



Sketch B1.12 adding the circles

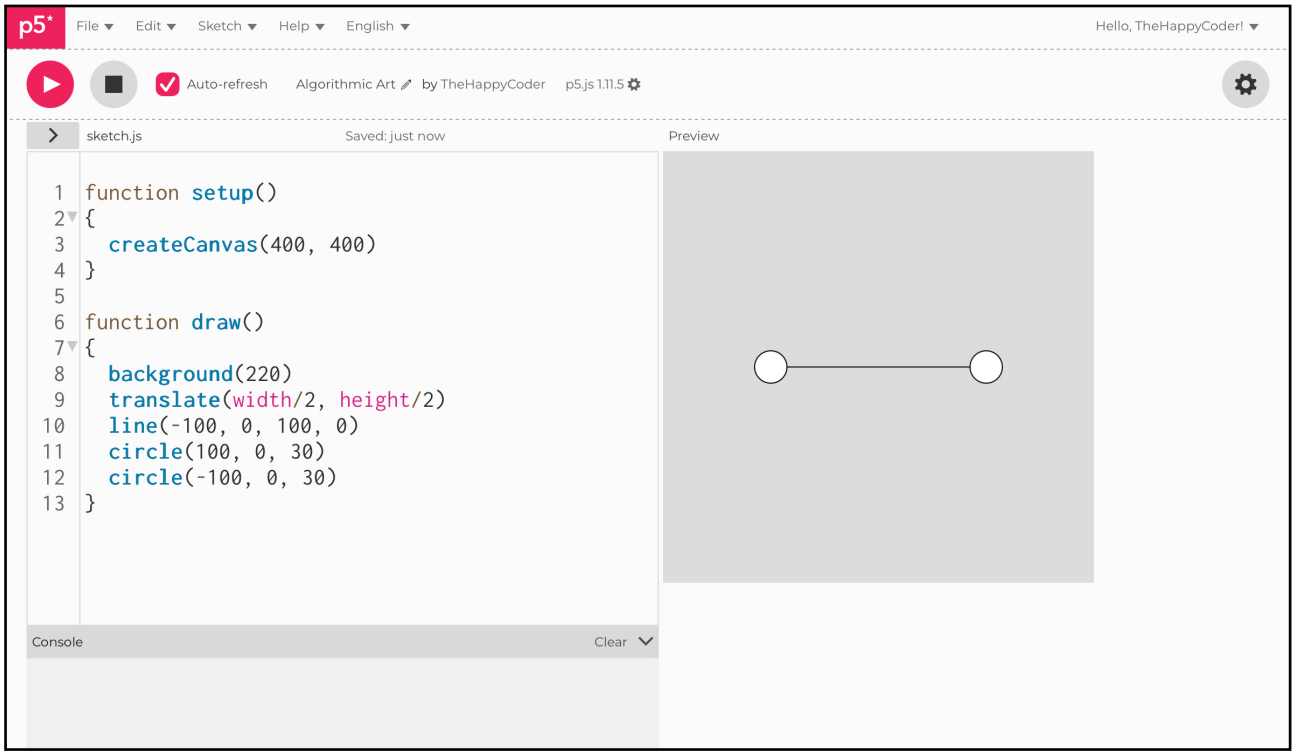
We will add a circle to each end of the line.

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  line(-100, 0, 100, 0)
  circle(100, 0, 30)
  circle(-100, 0, 30)
}
```

Notes

Notices that the co-ordinates take into account the translation; this is so we can rotate about the origin.

Figure B1.12


p5*
File
Edit
Sketch
Help
English
Hello, TheHappyCoder!
Auto-refresh
Algorithmic Art
by TheHappyCoder
p5.js 1.11.5
sketch.js
Saved: just now
Preview
1 function setup()
2 {
3 createCanvas(400, 400)
4 }
5
6 function draw()
7 {
8 background(220)
9 translate(width/2, height/2)
10 line(-100, 0, 100, 0)
11 circle(100, 0, 30)
12 circle(-100, 0, 30)
13 }
Console
Clear

Sketch B1.13 angle of rotation

We add a variable called `angle` and rotate it about that angle. We will keep it in radians for now.

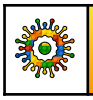
```
let angle = 0


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


function draw()
{
  background(220)
  translate(width/2, height/2)
  rotate(angle)
  line(-100, 0, 100, 0)
  circle(100, 0, 30)
  circle(-100, 0, 30)
}
```

🗒️ Notes

Notice that there is no movement just yet; to do that, we need to increment the angle.

🛠️ Code Explanation

`rotate(angle)`	Rotates the value of variable angle in radians



Sketch B1.14 start it spinning

We can now spin the baton at a nice, steady rate by incrementing the `angle` by `0.05` every iteration.

```
let angle = 0

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  rotate(angle)
  line(-100, 0, 100, 0)
  circle(100, 0, 30)
  circle(-100, 0, 30)
  angle += 0.05
}
```

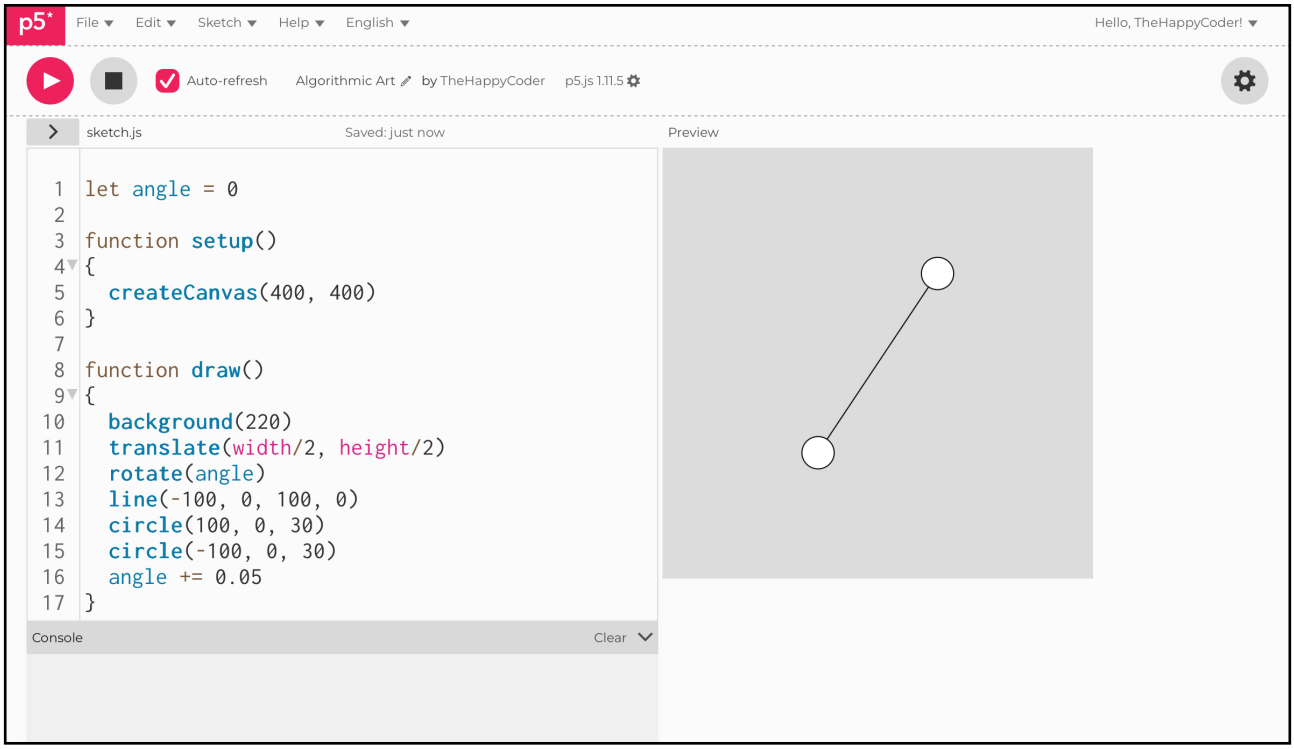
🗒️ Notes

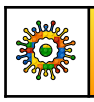
The baton rotates at `0.05` radians per iteration in the `draw()` loop.

🌻 Challenge

Change the value from `0.05`.

Figure B1.14


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)
}
function draw()
{
background(220)
translate(width/2, height/2)
rotate(angle)
line(-100, 0, 100, 0)
circle(100, 0, 30)
circle(-100, 0, 30)
angle += 0.05
}
Console
Clear



Sketch B1.15 velocity

We replace `increment` with a value for `velocity`.

```
let angle = 0
let velocity = 0.05

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  rotate(angle)
  line(-100, 0, 100, 0)
  circle(100, 0, 30)
  circle(-100, 0, 30)
  angle += velocity
}
```

Notes

Nothing has changed.

Sketch B1.16 acceleration

We add an acceleration variable; this means that the velocity is incremented by that amount each iteration (hence the very small number).

```
let angle = 0
let velocity = 0.05
let acceleration = 0.0005

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

function draw()
{
  background(220)
  translate(width/2, height/2)
  rotate(angle)
  line(-100, 0, 100, 0)
  circle(100, 0, 30)
  circle(-100, 0, 30)
  angle += velocity
  velocity += acceleration
}
```

Notes

What you should see (if you give it a minute or two) is the baton spinning faster and faster. Acceleration is the rate of change of velocity. So the velocity is increasing gradually.

Challenge

Try: `let velocity = -0.1`.