Informatics Summer Camp

Coding and Web Development

Lesson Resources

GitHub

https://github.com/stephensheridan/SummerCamp2024

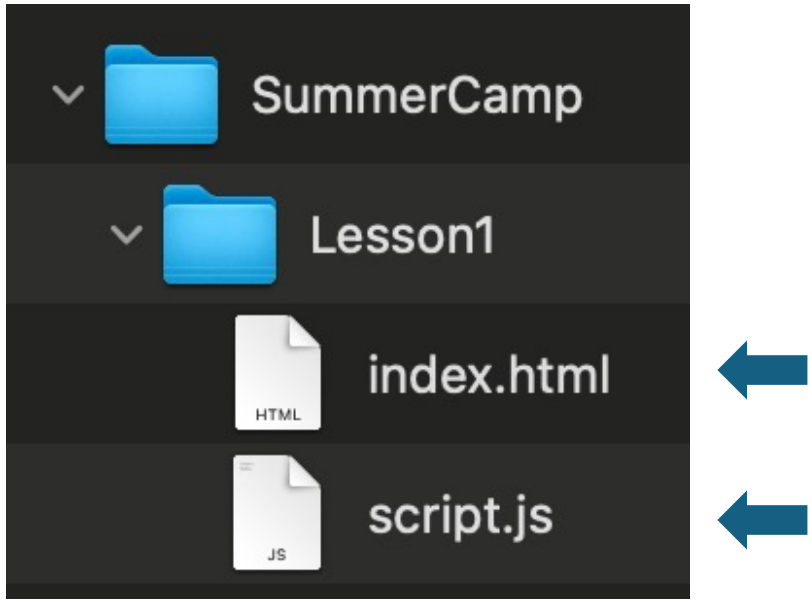
p5play

https://p5play.org/learn/sprite.html

Open these as tabs in your Browser Window

Lesson 1: Setup and Draw

- Create a folder called **SummerCamp**
- Create a folder called **Lesson1** inside the **SummerCamp** folder.
- Using a text editor, create and save the following files in the **Lesson1** folder.



Lesson 1: Setup and Draw

- Copy the Lesson1 ***index.html*** and ***script.js*** code from GitHub

https://github.com/stephensheridan/SummerCamp2024/tree/main/Lesson1

```html
<html>
  <head>
    <!-- p5 and p5play Javascript libraries -->
    <script src="https://cdn.jsdelivr.net/npm/p5@1/lib/p5.min.js"></script>
    <script src="https://p5play.org/v3/planck.min.js"></script>
    <script src="https://p5play.org/v3/p5play.js"></script>
    <!-- Your script -->
    <script src="script.js"></script>
  </head>
  <body>
    <main>
    </main>
  </body>
</html>
```

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

function draw() {
  background('red');
}
```

Lesson 1: Setup and Draw

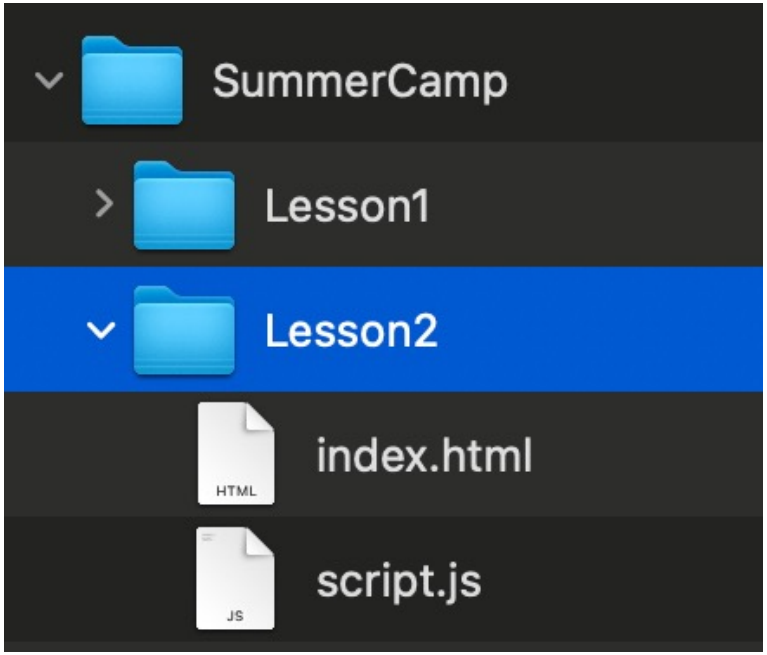
- Open the **Lesson1 index.html** file in a Browser. You should see a blank red canvas. **NOTE:** when you make a change to your code, refresh the browser window to see the results.

Challenge 1: Setup and Draw

- Change the canvas background colour to grey.

Lesson 2: Sprites

- Create a folder called **Lesson2** and copy the **Lesson1** ***index.html*** and ***script.js*** files into it.



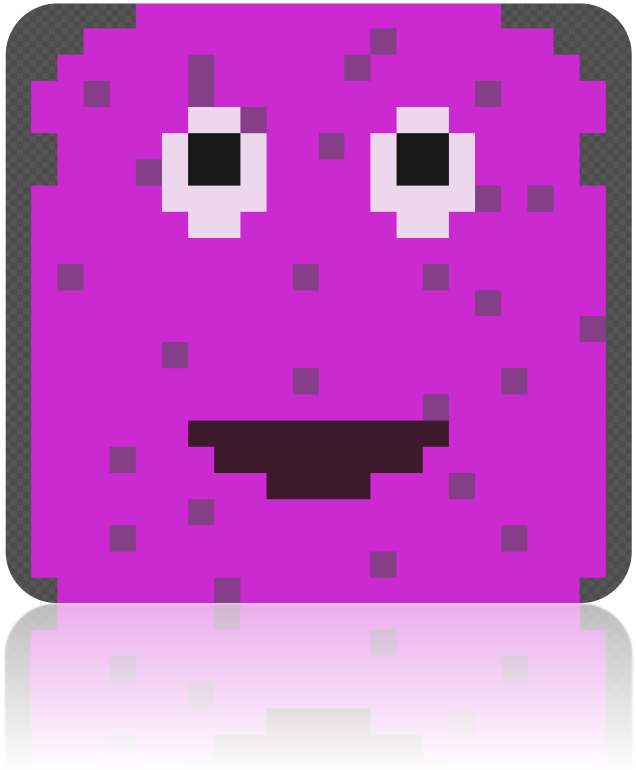
Lesson 2: Sprites

What is a sprite?

A sprite is a ghost!

Video game developers use the word "sprite" to refer to characters, items, or anything else that moves above a background.

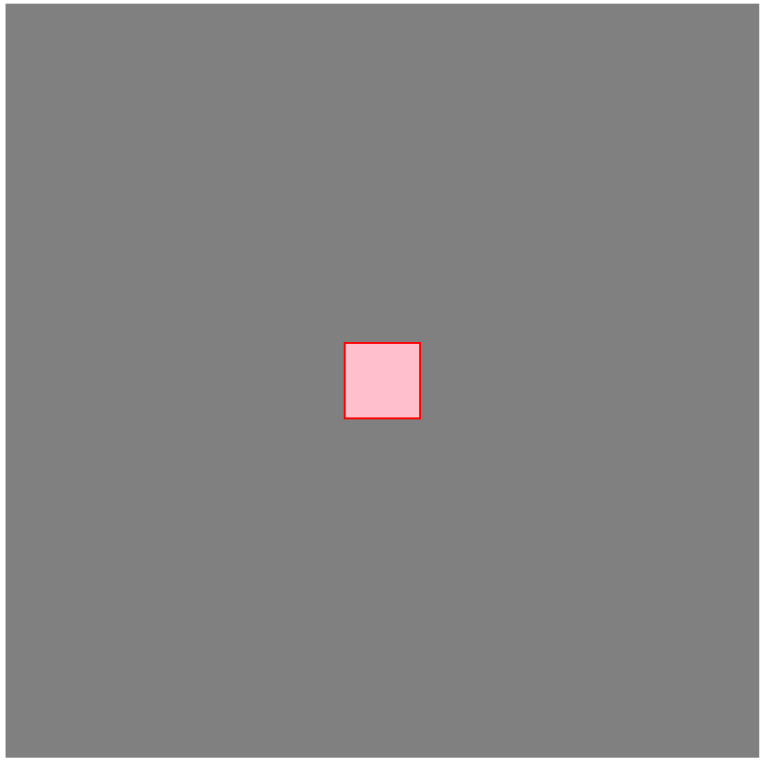
The `new Sprite()` constructor creates a sprite object, which contains variables that define a sprite's position, size, and appearance.



Lesson 2: Sprites

- Modify your **Lesson2 *script.js*** code as follows to see what happens.

```
let sprite;

function setup() {
  createCanvas(400, 400);
  sprite = new Sprite()
  sprite.x = 200;
  sprite.y = 200;
  sprite.w = 40;
  sprite.h = 40;
  sprite.color = 'pink';
  sprite.stroke = 'red';
}

function draw() {
  background('grey');
}
```



Lesson 2: Sprites

- We can set a random location for our sprite by using the **random** function. Examples of how to use the random function can be seen below.

```
// Random number between 0 and 1
var num = random(0,100);
```

```
// Simulate a dice roll
var roll = random(1,6);
```

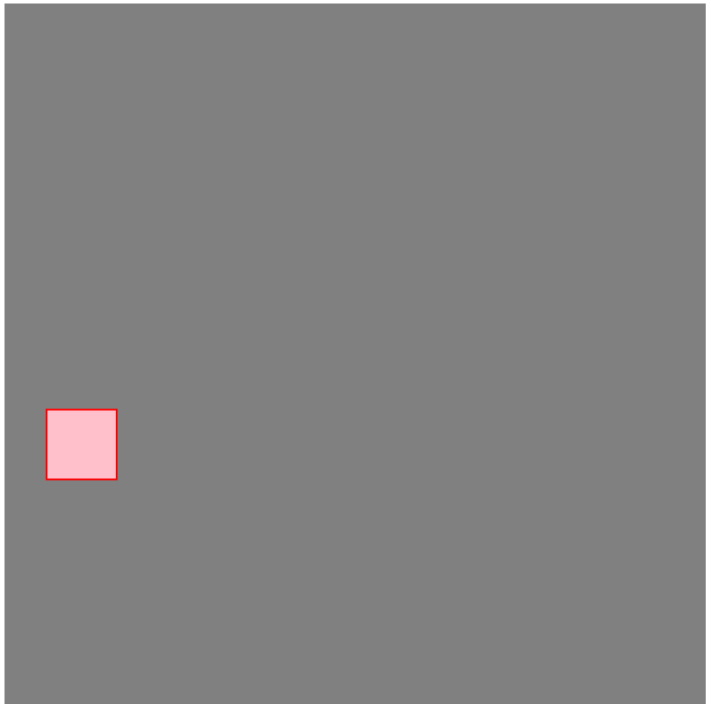
```
// Random x position
var x = random(width);
```

```
// Random y position
var y = random(height);
```

Lesson 2: Sprites

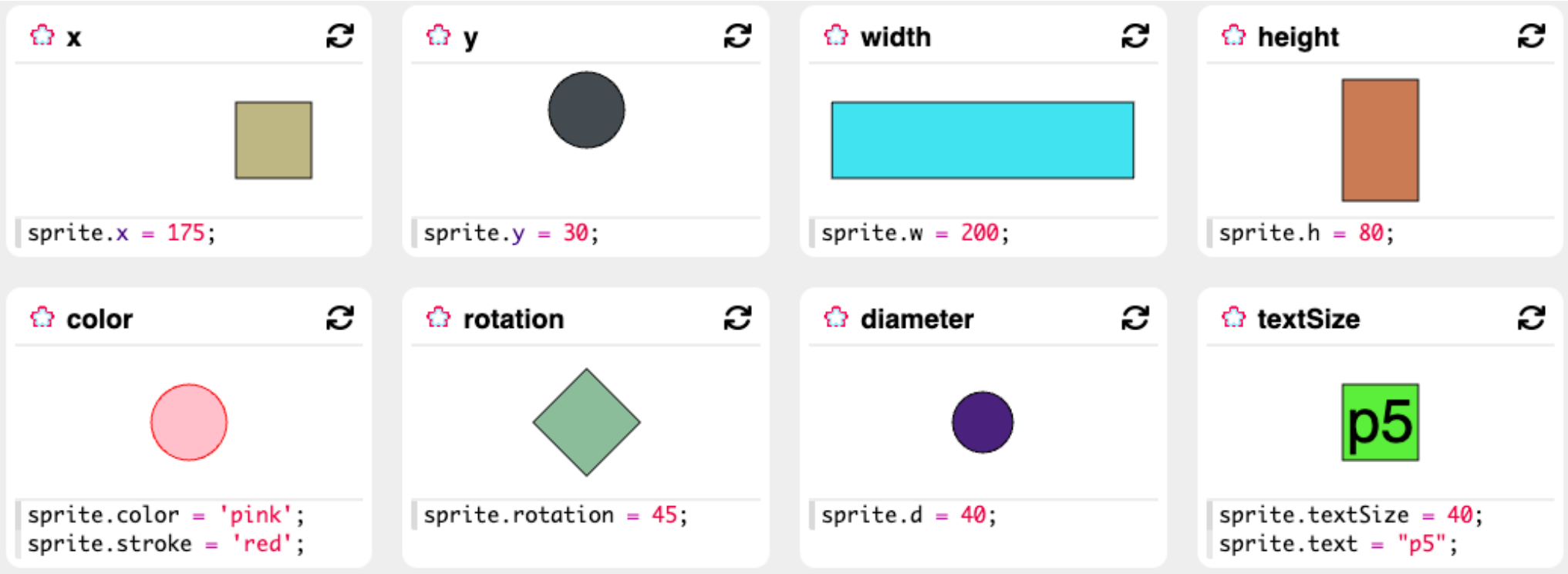
- Modify your **Lesson2** ***script.js*** code as follows to see what happens. Hit **refresh** in the Browser to see what happens.

```
let sprite, ball;

function setup() {
  createCanvas(400, 400);
  sprite = new Sprite()
  sprite.x = random(width);
  sprite.y = random(height);
  sprite.w = 40;
  sprite.h = 40;
  sprite.color = 'pink';
  sprite.stroke = 'red';
}

function draw() {
  background('grey');
}
```



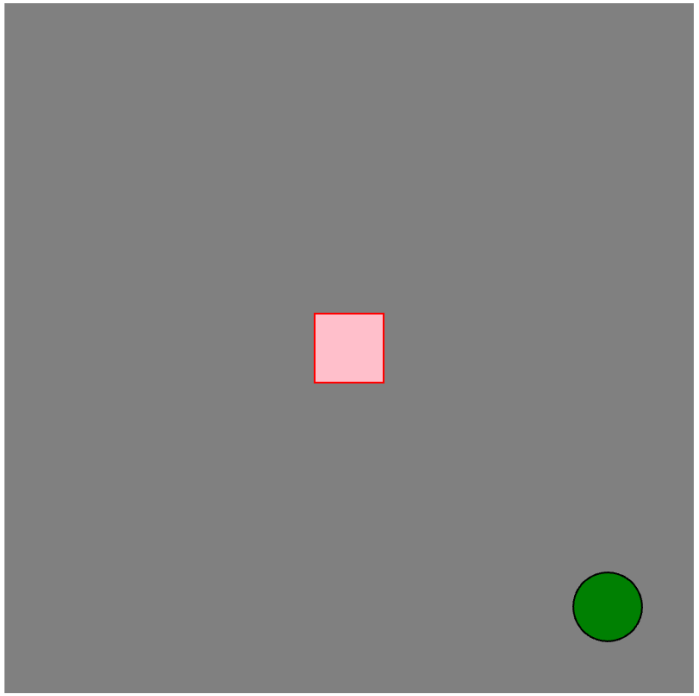
Lesson 2: Sprites

- **Sprites** have many different properties. Experiment with the properties show below.



Challenge 2: Sprites

- Create a second **Sprite** called **ball**. This sprite should be circular in shape with a diameter of **20**, positioned in the bottom right corner of the canvas, filled with green and with a black border.



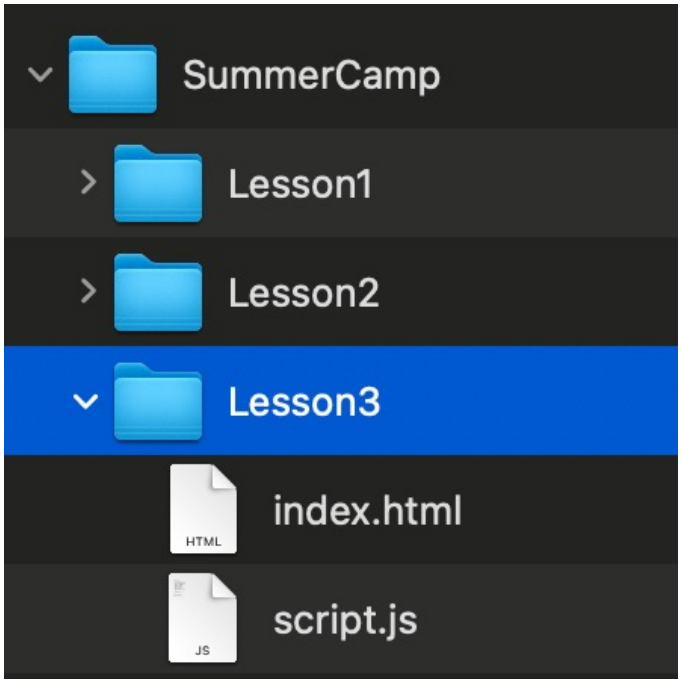
QUESTION

Name the property that controls the colour of a Sprite’s border?

Raise your hand if you know the answer.

Lesson 3: Sprite Movement

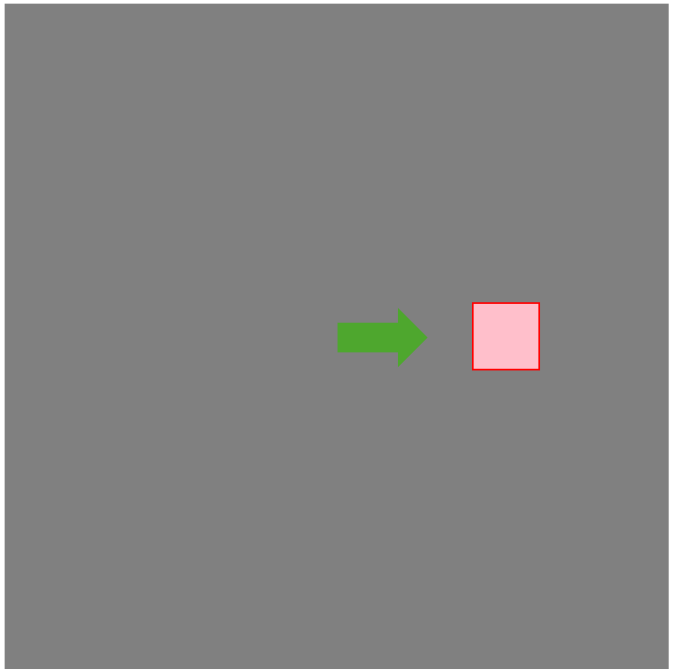
- Create a folder called **Lesson3** and copy the **Lesson2** ***index.html*** and ***script.js*** files into it.



Lesson 3: Sprite Movement

- We can make Sprites move by giving them an **x** or **y velocity**.

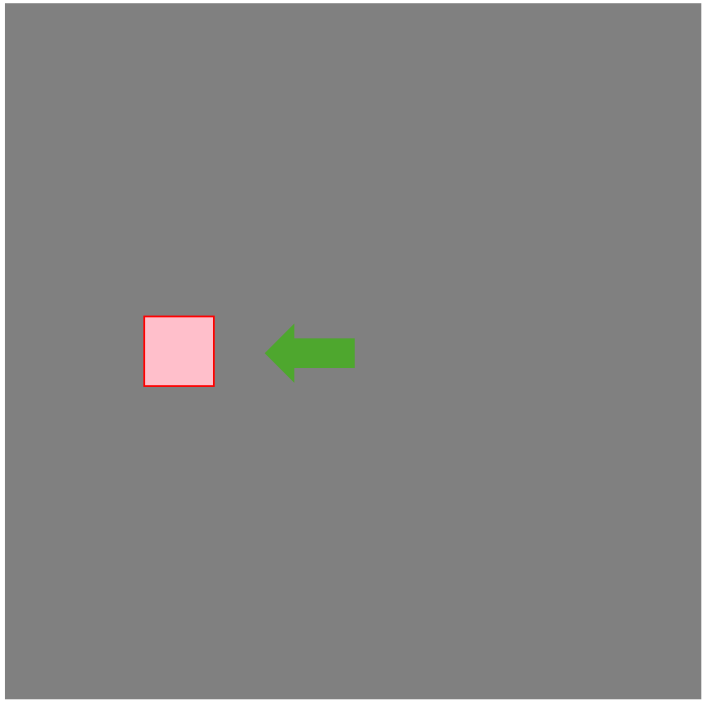
```
let sprite;

function setup() {
  createCanvas(400, 400);
  sprite = new Sprite()
  sprite.x = 200;
  sprite.y = 200;
  sprite.w = 40;
  sprite.h = 40;
  sprite.color = 'pink';
  sprite.stroke = 'red';

  // Make the sprite move right
  sprite.vel.x = 1;
}

function draw() {
  background('grey');
}
```



Lesson 3: Sprite Movement

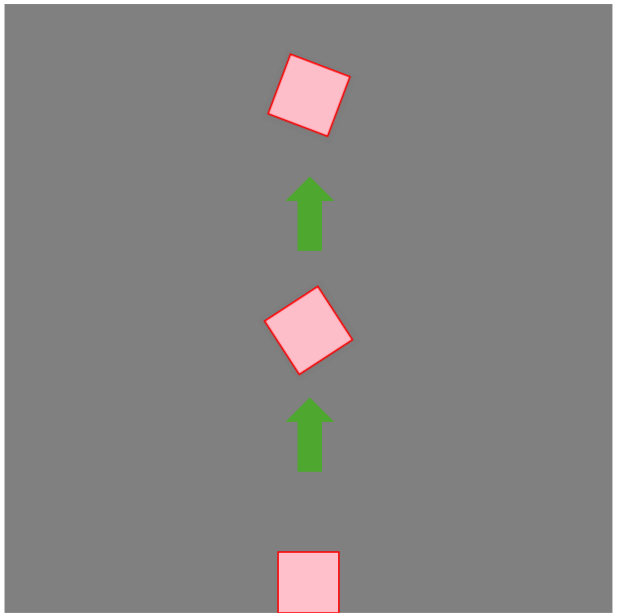
- We can make Sprites move by giving them an **x** or **y velocity**.

```
let sprite;

function setup() {
  createCanvas(400, 400);
  sprite = new Sprite()
  sprite.x = 400;
  sprite.y = 200;
  sprite.w = 40;
  sprite.h = 40;
  sprite.color = 'pink';
  sprite.stroke = 'red';

  // Make the sprite move left
  sprite.vel.x = -1;
}

function draw() {
  background('grey');
}
```



Challenge 3: Sprite Movement

- Make the Sprite move upwards starting at the bottom centre of the canvas. Also, try to give the Sprite some rotation movement using the **rotationSpeed** property.



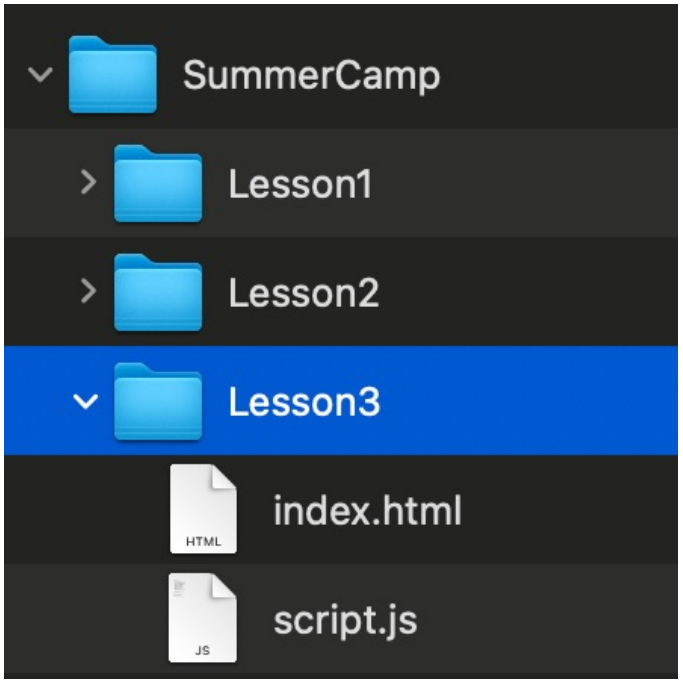
QUESTION

Name the property that controls the colour of a Sprite’s border?

Raise your hand if you know the answer.

Lesson 4: Where's my Sprite?

- Create a folder called **Lesson4** and copy the **Lesson3** ***index.html*** and ***script.js*** files into it.



Lesson 4: Where's my Sprite?

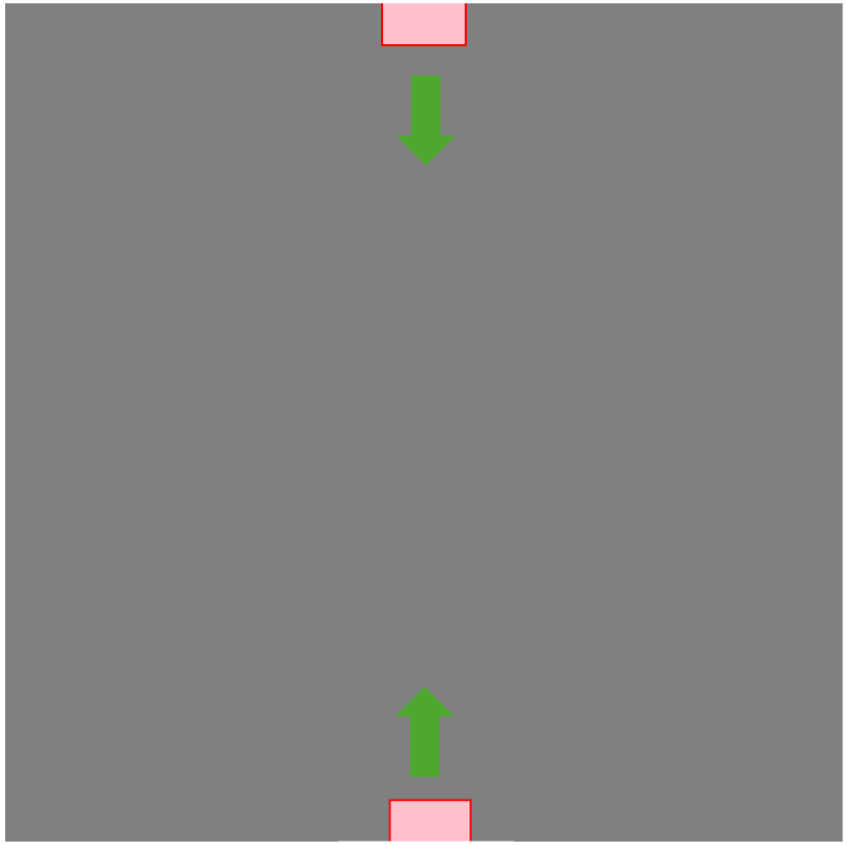
- We can no longer see a Sprite when its **x** and **y** position goes outside the canvas area. We can make the sprite bounce right and left by using a **condition** in the **draw** function.

```
function draw() {
  background('grey');

  if (sprite.x > 400)
    sprite.vel.x = -1;

  if (sprite.x < 0)
    sprite.vel.x = 1;
}
```

Change x direction to -1

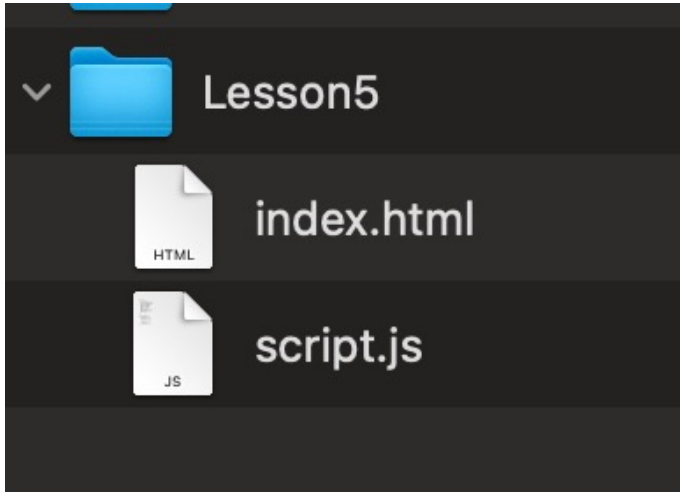
Challenge 4: Where's my Sprite?

- Make the Sprite bounce up and down instead of right to left.



Lesson 5: Sprite Images

- Create a folder called **Lesson5** and copy the **Lesson4** ***index.html*** and ***script.js*** files into it.



Lesson 5: Sprite Images

- We are not limited to basic shapes for Sprites. We can load images from the web into our Sprites to make things more interesting. We can use the **preload** function to ensure all images that we need are loaded and ready to use.

- In these lessons we will use the following images located at the following URL: **https://stephensheridan.github.io/assets/**

tudublin.png

monster.png

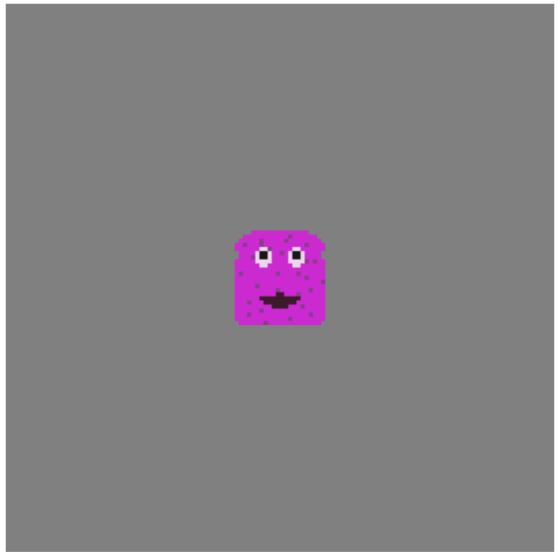
moon.png

earth.png

ship.png

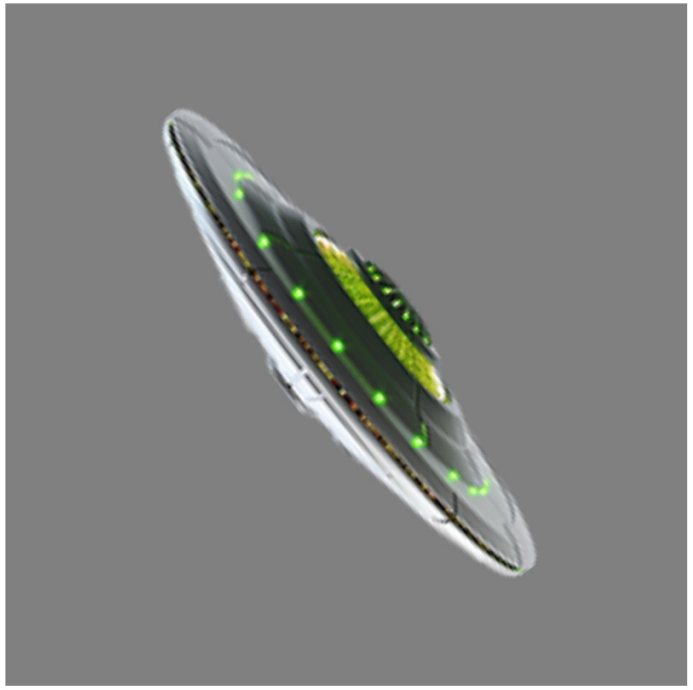
Lesson 5: Sprite Images

```
let monster, monsterImage;

function preload() {
  monsterImage = loadImage('https://stephensheridan.github.io/assets/monster.png');
}

function setup() {
  createCanvas(400, 400);
  monster = new Sprite()
  monster.x = 200;
  monster.y = 200;
  monster.w = 72;
  monster.h = 69;
  monster.addImage(monsterImage);
}

function draw() {
  background('grey');
}
```



Challenge 5: Sprite Images

- Change the monster sprite to a spaceship and resize it to fit the canvas area using the Sprite **scale** property. You can also add a little **rotationSpeed** if you like.



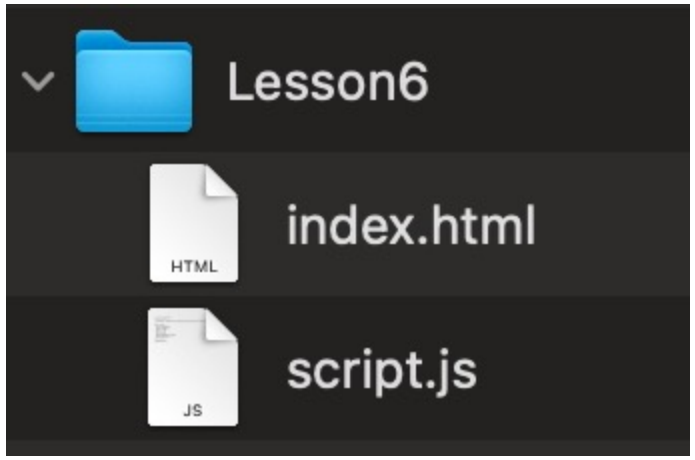
QUESTION

These lessons use PNG images. Can you name another type of image format?

Raise your hand if you know the answer.

Lesson 6: Monster Mover

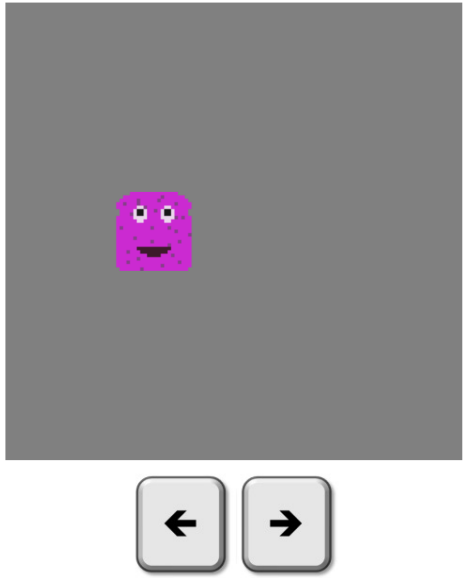
- Create a folder called **Lesson6** and copy the **Lesson5** ***index.html*** and ***script.js*** files into it.



Lesson 6: Monster Mover

- Our monster wants to move. Let's make it move left and right when we press the left and right arrow keys on the keyboard. We can capture **keypresses** in the draw code and change the monster's **x** and **y velocity** to do this.

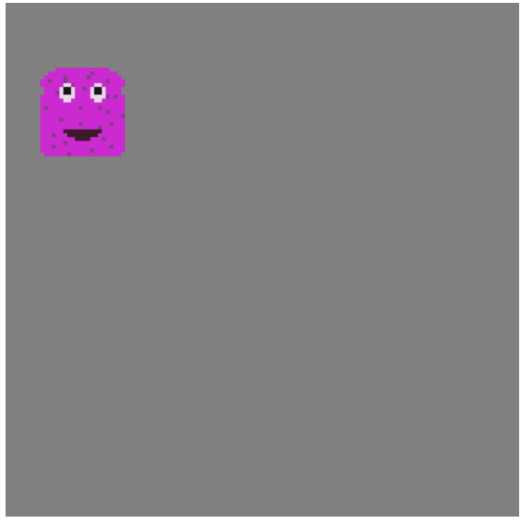
```
function draw() {
  background('grey');

  if (kb.pressing('left'))
    monster.vel.x = -5;
  else if (kb.pressing('right'))
    monster.vel.x = 5;
  else
    monster.vel.x = 0;

}
```



Challenge 6: Monster Mover

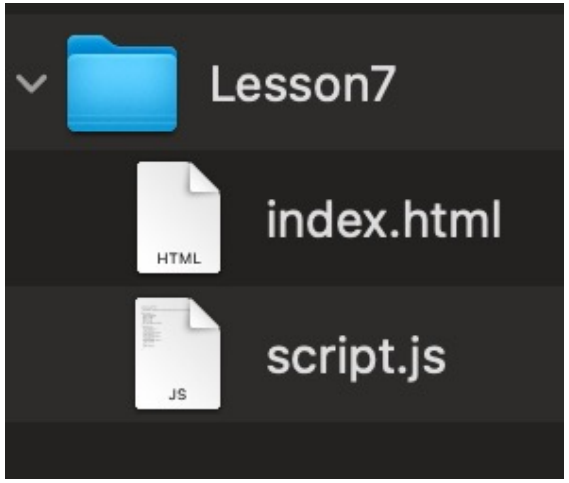
- Add two more **else if** branches to allow the monster to move up and down as well as left and right. Think about the **??** Below.

```
function draw() {
  background('grey');

  if (kb.pressing('left'))
    monster.vel.x = -5;
  else if (kb.pressing('right'))
    monster.vel.x = 5;
  else if (kb.pressing(??))
    monster.vel.y = ??;
  else if (kb.pressing(??))
    monster.vel.y = ??;
  else{
    monster.vel.x = 0;
    monster.vel.y = 0;
  }
}
```



Lesson 7: Seeing Stars

- Create a folder called **Lesson7** and copy the **Lesson6** ***index.html*** and ***script.js*** files into it.



Lesson 7: Seeing Stars

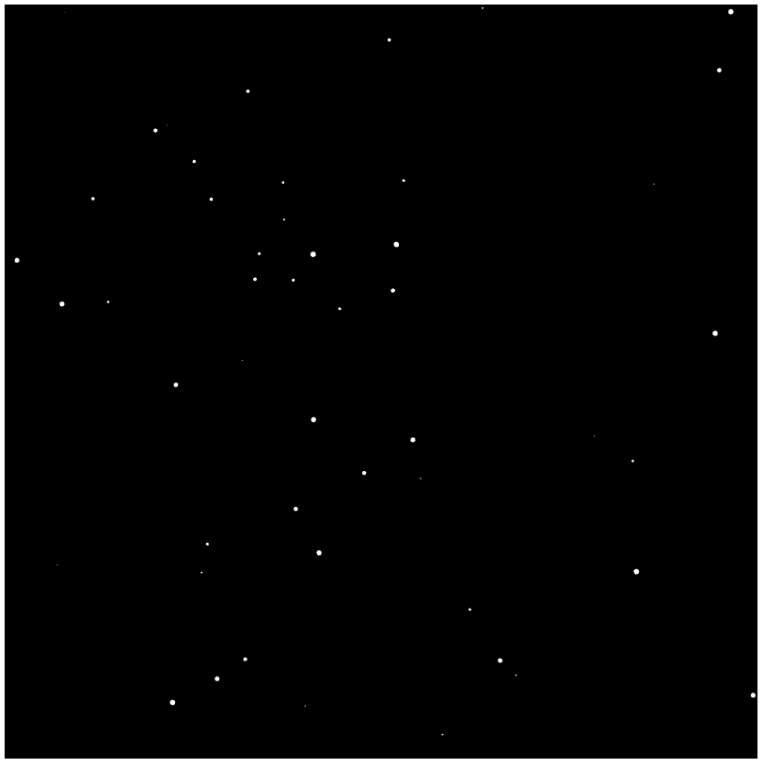
- Let's create some stars. But there are so many, how can we do it?
- Loops can be used to repeat a block of instructions as many times as we like.

```
for (var count = 0; count < 50; count++) {
  // Instructions to repeat
  // in here!!
  // They will be repeated 50
  // times.
  // The counter will start at 0
  // and count up to 50.
}
```

Lesson 7: Seeing Stars

- Add the **drawStars** function to your script code to see what happens.

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

function drawStars(){
  for (var count = 0; count < 50; count++) {
    var x = random(width);
    var y = random(height);
    var size = random(1, 4);
    fill(255);
    ellipse(x, y, size, size);
  }
}

function draw() {
  background('black');
  drawStars()
}
```



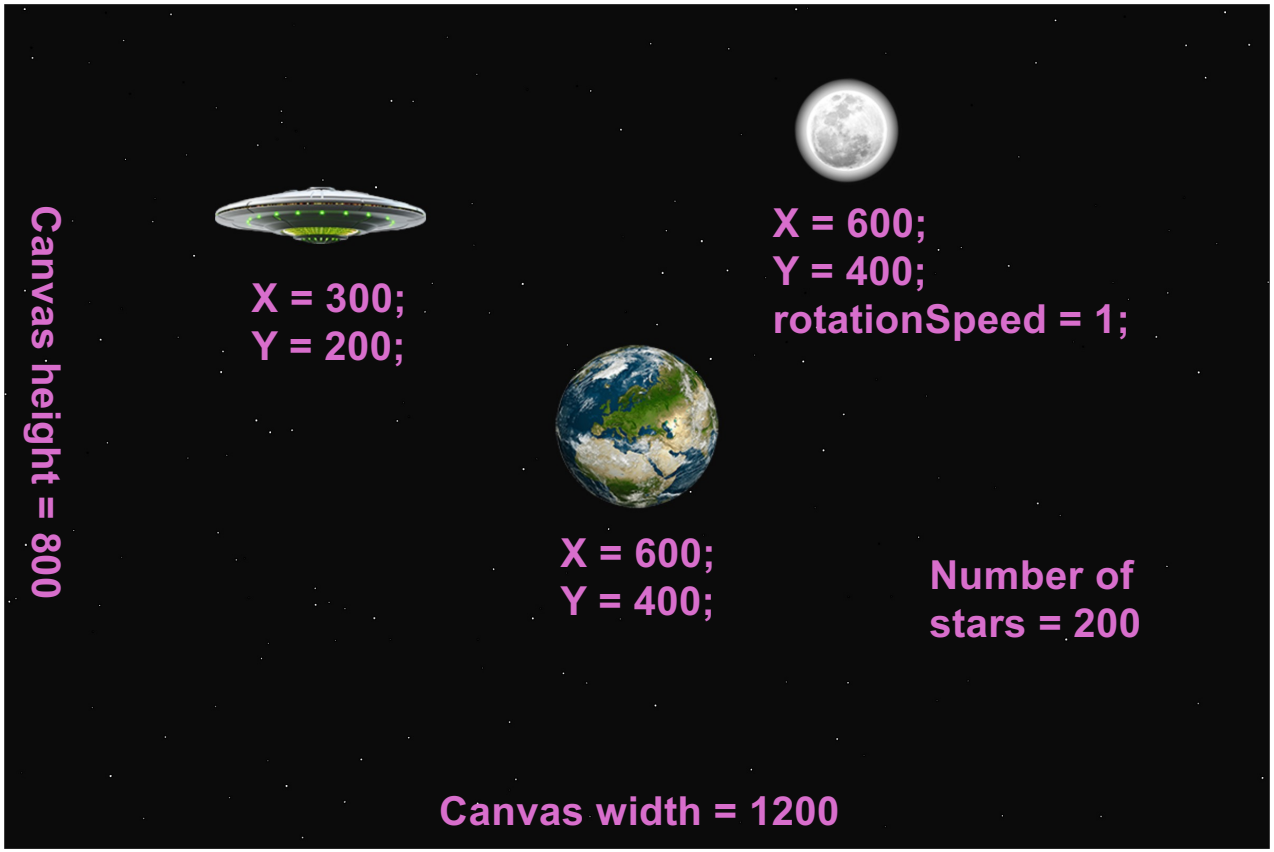
Lesson 7: Seeing Stars

- We can fix the stars in position by adding a random number seed. This will force our code to generate the same random numbers in each draw.

```
function drawStars(){
  randomSeed(99);
  for (var count = 0; count < 50; count++) {
    var x = random(width);
    var y = random(height);
    var size = random(1, 4);
    fill(255);
    ellipse(x, y, size, size);
  }
}
```

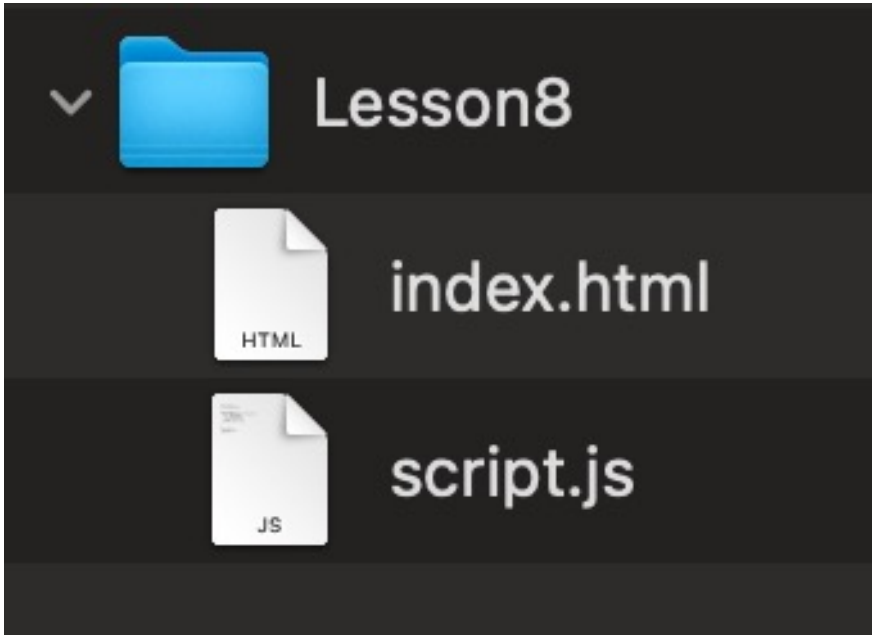
Challenge 7: Seeing Stars

- Can you replicate the following scene? Look at Lesson5 for tips.



Lesson 8: Colliders & Gravity

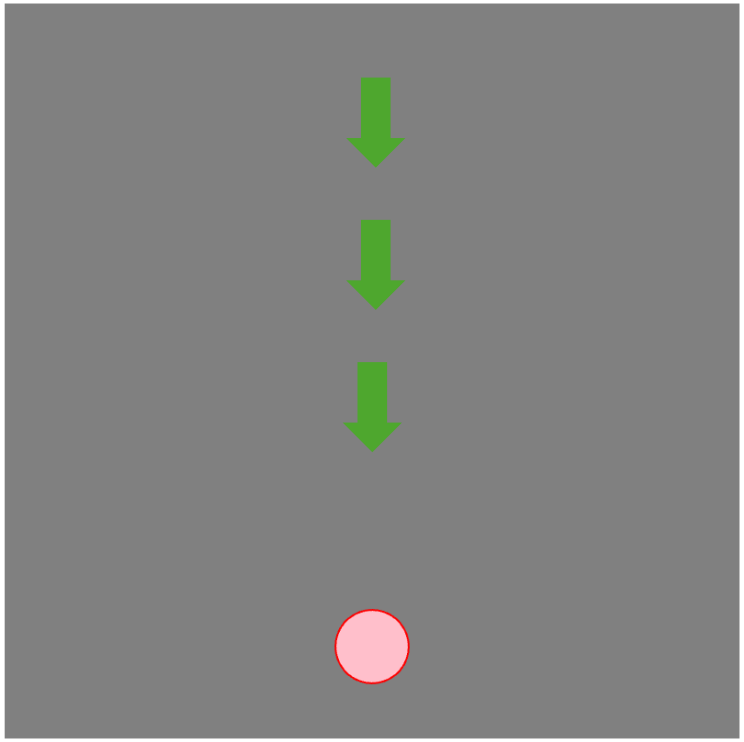
- Create a folder called **Lesson8** and copy the **Lesson7** ***index.html*** and ***script.js*** files into it.



Lesson 8: Gravity & Colliders

- By default, our world does not have any gravity. Let's add some gravity to see how it effects a sprite.

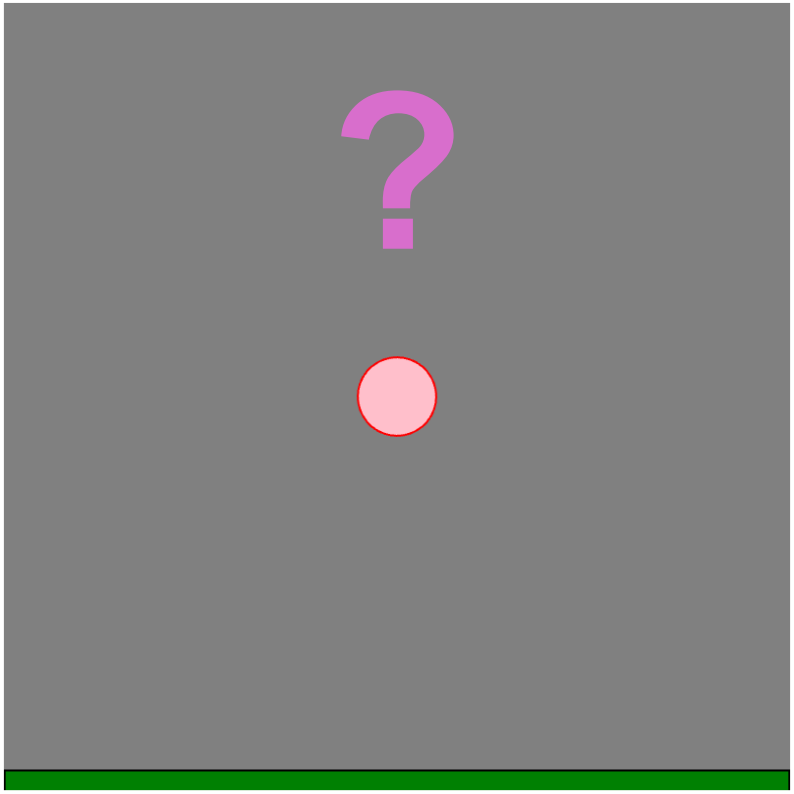
```
let sprite;

function setup() {
  createCanvas(400, 400);
  sprite = new Sprite();
  sprite.x = 200;
  sprite.y = 100;
  sprite.d = 40;
  sprite.colour = 'pink';
  sprite.stroke = 'red';

  world.gravity.y = 10;
}

function draw() {
  background('grey');
}
```



Lesson 8: Gravity & Colliders

- As you can see, the force of gravity causes our sprite to fall downwards and off the canvas. What if we added a floor?

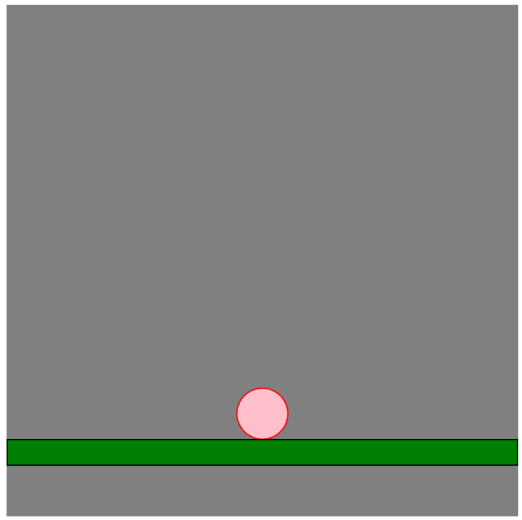
```
let sprite, floor;

function setup() {
  createCanvas(400, 400);
  sprite = new Sprite();
  sprite.x = 200;
  sprite.y = 10;
  sprite.d = 40;
  sprite.colour = 'pink';
  sprite.stroke = 'red';

  floor = new Sprite();
  floor.x = 200;
  floor.y = 350;
  floor.w = 400;
  floor.h = 20;
  floor.colour = 'green';
  floor.stroke = 'black';

  world.gravity.y = 10;
}
```



Lesson 8: Gravity & Colliders

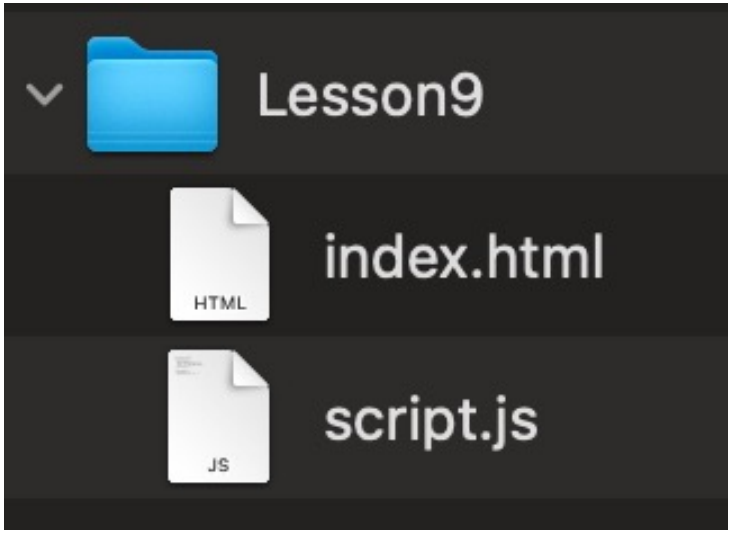
- As you can see, the floor is also affected by gravity and falls of the canvas. This is because all sprites have dynamic colliders by default and are affected by gravity. We can set the floor's **collider** to **static** to make it stationary and unaffected by gravity.

```
floor = new Sprite();
floor.x = 200;
floor.y = 350;
floor.w = 400;
floor.h = 20;
floor.colour = 'green';
floor.stroke = 'black';
floor.collider = 'static';
```



Lesson 9: Attractors

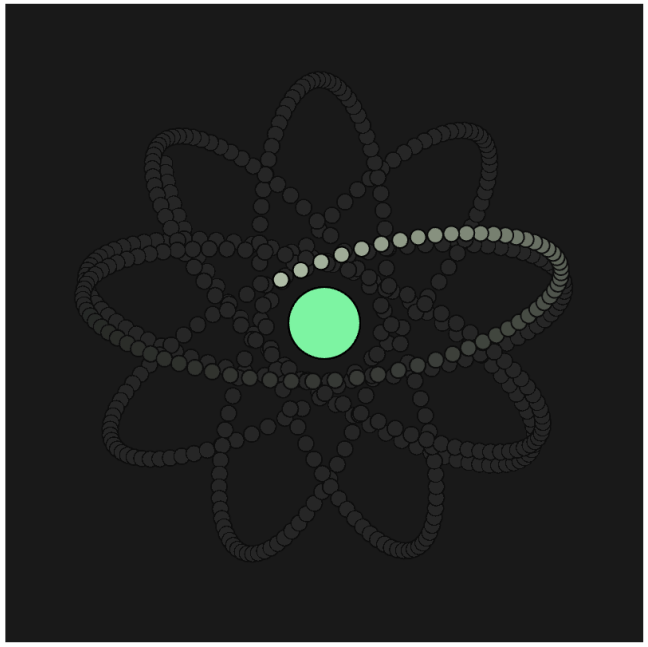
- Create a folder called **Lesson9** and copy the **Lesson8** ***index.html*** and ***script.js*** files into it.



Lesson 9: Attractors

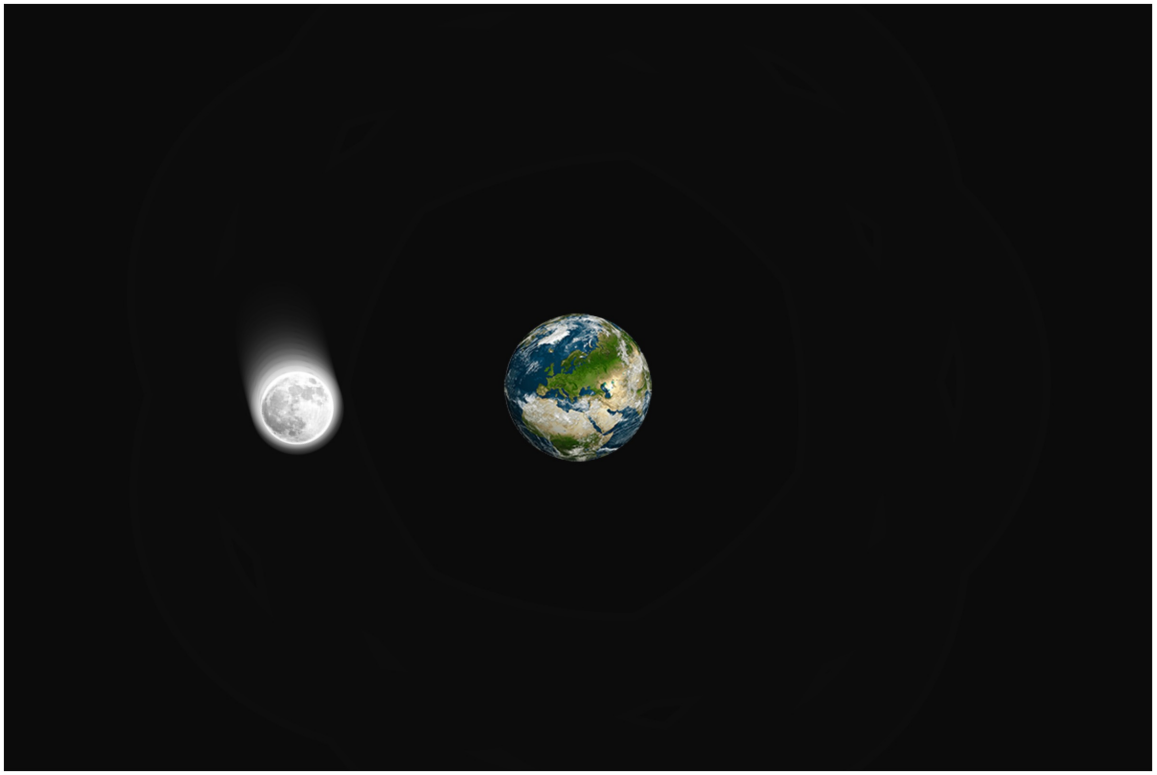
- We can use the **attractTo** function to attract a sprite to a position by applying a force. The position can be given as another sprite with x and y properties or as separate x and y values.

```
let nucleus, electron;

function setup() {
  new Canvas(400, 400);
  nucleus = new Sprite(200, 200, 45);
  electron = new Sprite(100, 100, 10);
  electron.vel.y = 5;
}

function draw() {
  background(16, 10);
  electron.attractTo(nucleus, 5);
}
```



Challenge 9: Attractors

- Make a moon sprite orbit around an earth sprite.



QUESTION

Name the following web browsers?

Raise your hand if you know the answer.

Lesson 10: Putting it all together

- Create a folder called **Lesson10** and copy the **Lesson9** ***index.html*** and ***script.js*** files into it.

- Copy and paste the code from Lesson10 on GitHub and spend some time investigating how it works.

Lesson 10: Putting it all together