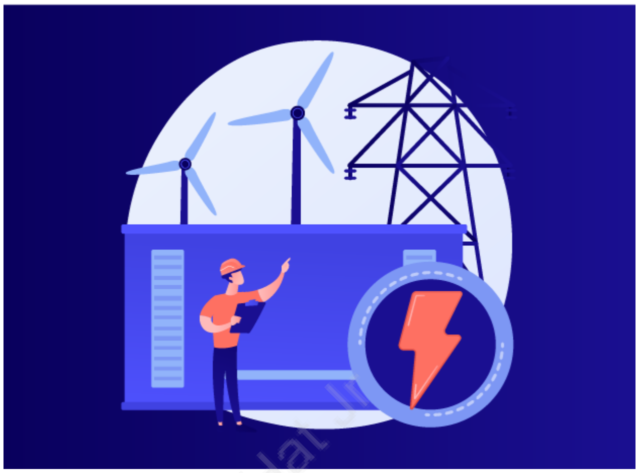
BASICS OF THE PHYSICS ENGINE



What is our GOAL for this MODULE?

In this class, we learned to create various shapes of objects. We also learned to add the physics properties to these objects and observed their behavior on changing properties.

What did we ACHIEVE in the class TODAY?

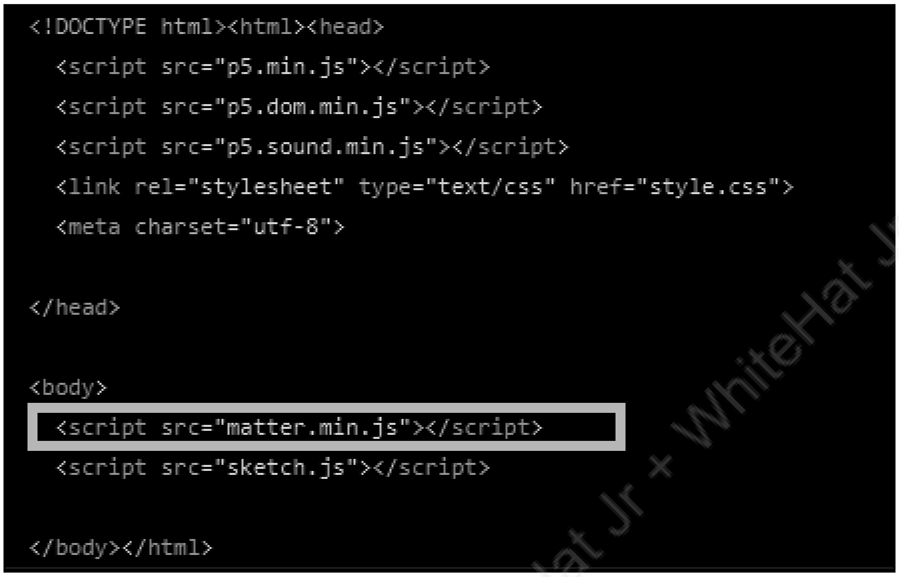
- Installed the physics engine library called **matter.min.js.**
- Created a ball body using a **Bodies.circle()** function.
  Drew the ball body using an **ellipse()** and continuously updated the physics engine.
- Created the ground using the **Bodies.rectangle()**.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Getting to know about physics engines.
- How to create various objects using a physics engine.
- Installation of the physics engine (library called **matter.min.js**).

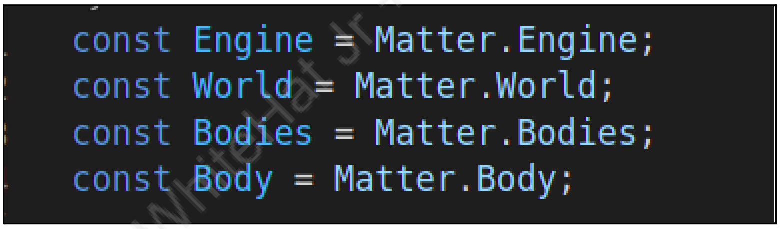
How did we DO the activities?

1. Install the physics engine library called **matter.min.js** by adding it in **index.html.** This library provides a 2D physics engine for the web built in JavaScript.



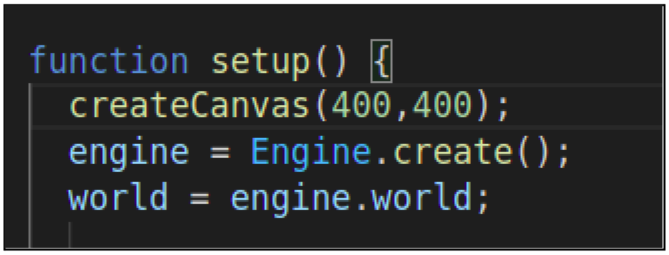
```
<!DOCTYPE html><html><head>
  <script src="p5.min.js"></script>
  <script src="p5.dom.min.js"></script>
  <script src="p5.sound.min.js"></script>
  <link rel="stylesheet" type="text/css" href="style.css">
  <meta charset="utf-8">

</head>

<body>
  <script src="matter.min.js"></script>
  <script src="sketch.js"></script>

</body></html>
```

2. Import modules from the **matter.min.js** library. To import these libraries, write the name of the library and the module which we want to import.



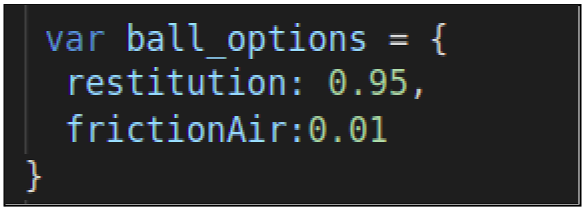
```
const Engine = Matter.Engine;
const World = Matter.World;
const Bodies = Matter.Bodies;
const Body = Matter.Body;
```

3. Create a canvas of **400**,**400.** Next, use the **Engine** module and create the **world.**



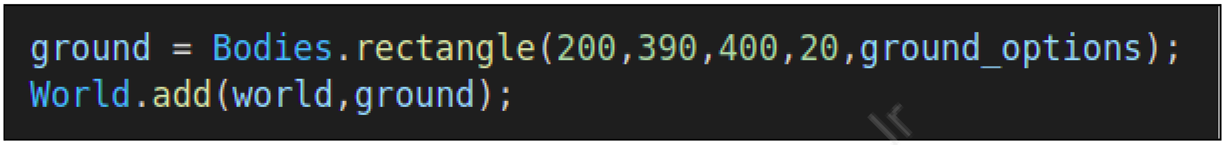
```
function setup() {
  createCanvas(400,400);
  engine = Engine.create();
  world = engine.world;
```

4. Create a ball body using a **Bodies.circle()** function which will help us to create the circular body. Apply properties of **restitution** and **frictionAir** to the ball.



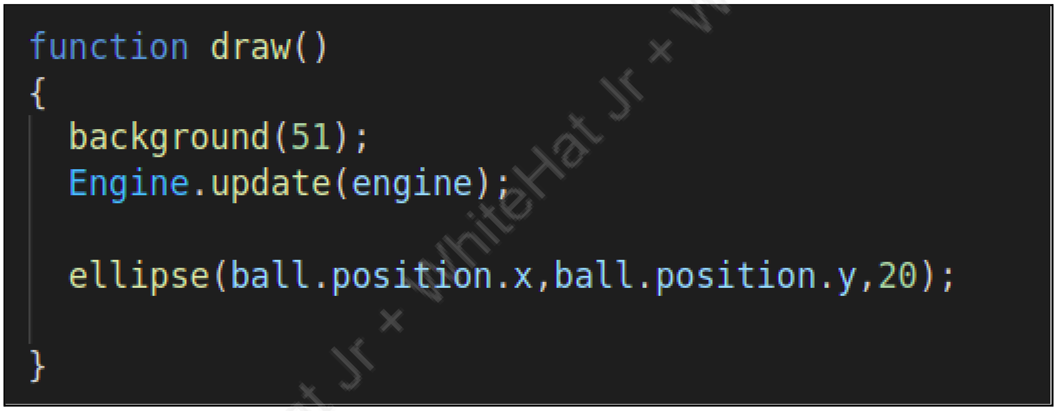
```
var ball_options = {
  restitution: 0.95,
  frictionAir:0.01
}
```

Next, add the body to the world by using the **World.add()** and pass the world and ball to it.



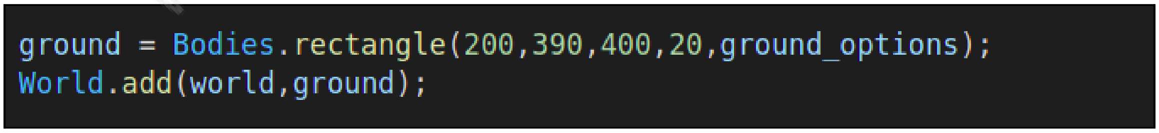
```
ground = Bodies.rectangle(200,390,400,20,ground_options);
World.add(world,ground);
```

5. Use an **ellipse()** object and draw the ball body and update the physics engine with the new changes.



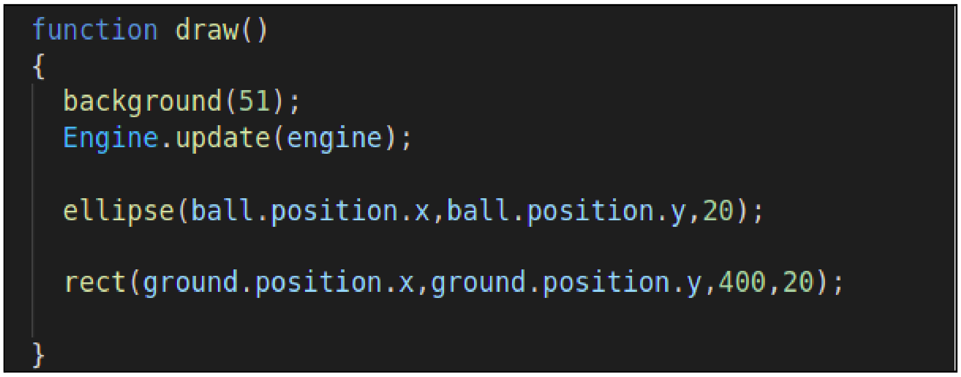
```
function draw()
{
  background(51);
  Engine.update(engine);

  ellipse(ball.position.x,ball.position.y,20);

}
```

6. Create the ground using the **Bodies.rectangle()** function and add the ground to the world.



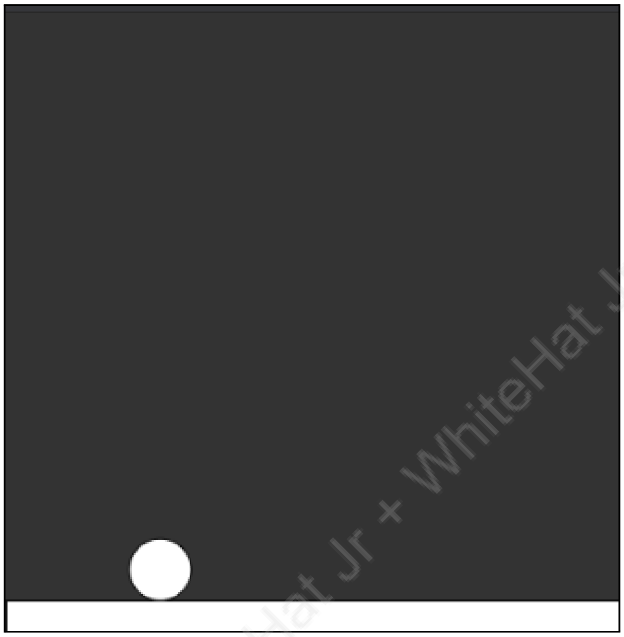
```
ground = Bodies.rectangle(200,390,400,20,ground_options);
World.add(world,ground);
```

Use the **rect()** object to display the ground in the **draw()** function.



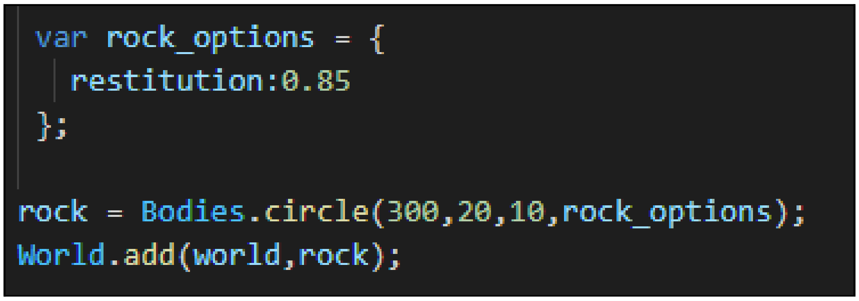
```
function draw()
{
  background(51);
  Engine.update(engine);

  ellipse(ball.position.x,ball.position.y,20);

  rect(ground.position.x,ground.position.y,400,20);

}
```

OUTPUT



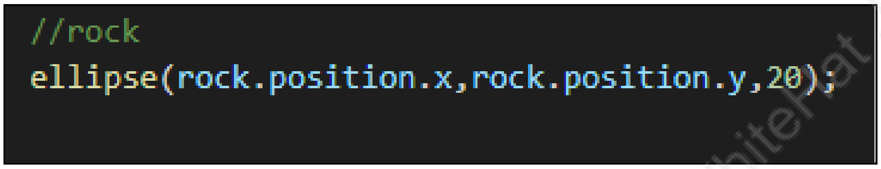
7. Now create two physics bodies on the canvas, a rock and a wall at the center of the canvas, the ball, and the rock will have different properties and positions.

8. Create a rock body using **Bodies.circle()** and define its physics properties in the **rock_options** variable, where you will only set the restitution as **0.85** for this rock body.



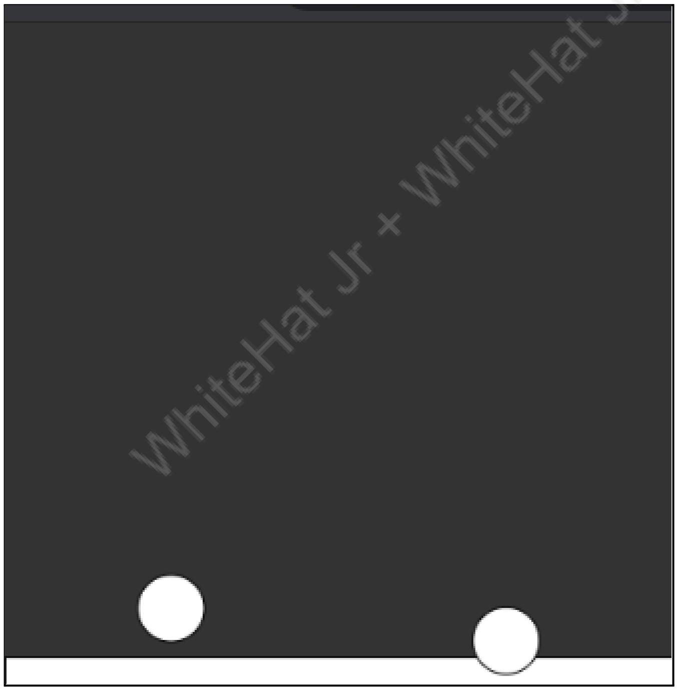
```
 var rock_options = {
   restitution:0.85
 };

rock = Bodies.circle(300,20,10,rock_options);
World.add(world,rock);
```

9. To display the ball on the canvas use the **ellipse()** function which will take the **x, y positions** and the **radius** as **20** of the rock body.

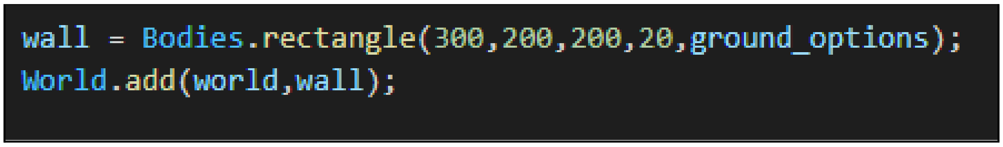


```
//rock
ellipse(rock.position.x,rock.position.y,20);
```

OUTPUT:



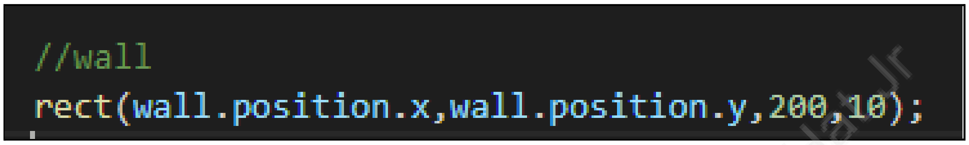
10. Now, create a stationary wall in the center of the canvas, in the **setup()** function using **Bodies.rectangle()** which will take the physics properties of the ground. After that, add a **wall** body to the physics world.



```
wall = Bodies.rectangle(300,200,200,20,ground_options);
World.add(world,wall);
```

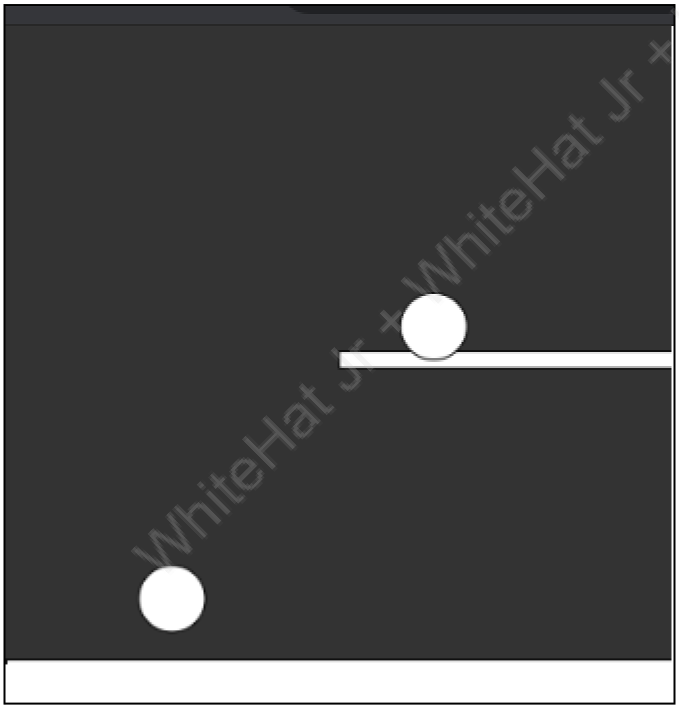
Here, set the values as - **300,200,200,20.**

11. Now finally show the center wall by creating a rectangle in the **draw()** function using the **rect()** function in which you pass the **x**, **y** positions of the wall body and set the width as **200**, and height as **10**.



```
//wall
rect(wall.position.x,wall.position.y,200,10);
```

OUTPUT:



What's next?

In the next class, we will create a class for the ground body.

Expand your knowledge:

Bookmark this link to know more about 2D Physics Engine (**Matter.min.js)**
https://brm.io/matter-js/docs/