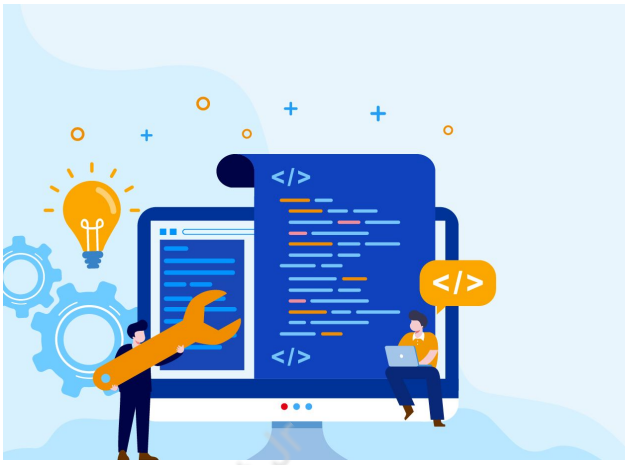
Writing functions which can take arguments



What is our GOAL for this MODULE?

The goal for this module is to extend our knowledge of functions and write functions which can take arguments.

What did we ACHIEVE in the class TODAY?

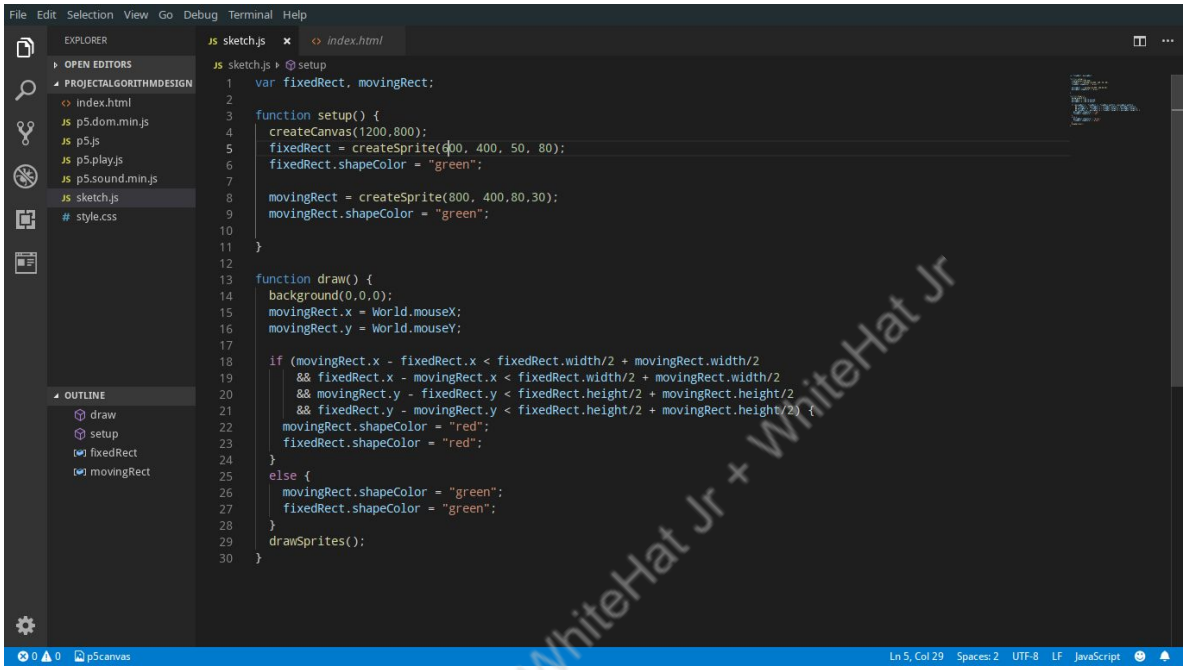
- Learn "true" and "false" as the two boolean values
- Write a function which can accept arguments, return values and can be re-used for the different game objects.
- Create a little code library and use it within the code.

Which CONCEPTS/ CODING BLOCKS did we cover today?

- Creating custom function

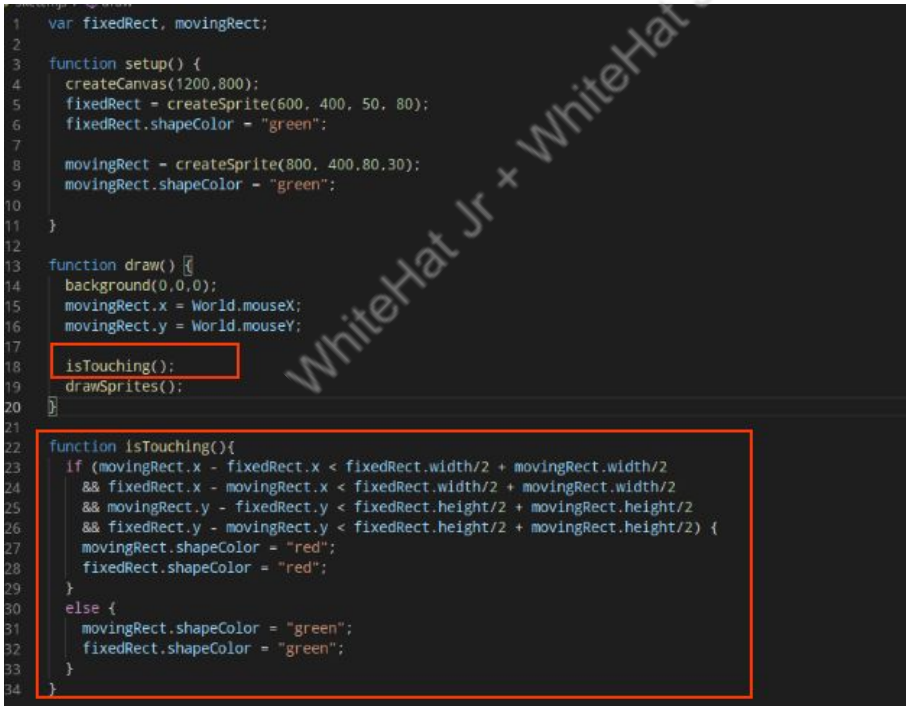
How did we DO the activities?

1. We have to write a big series of if-else conditions inside the code— code can detect collision between fixedRect and movingRect.



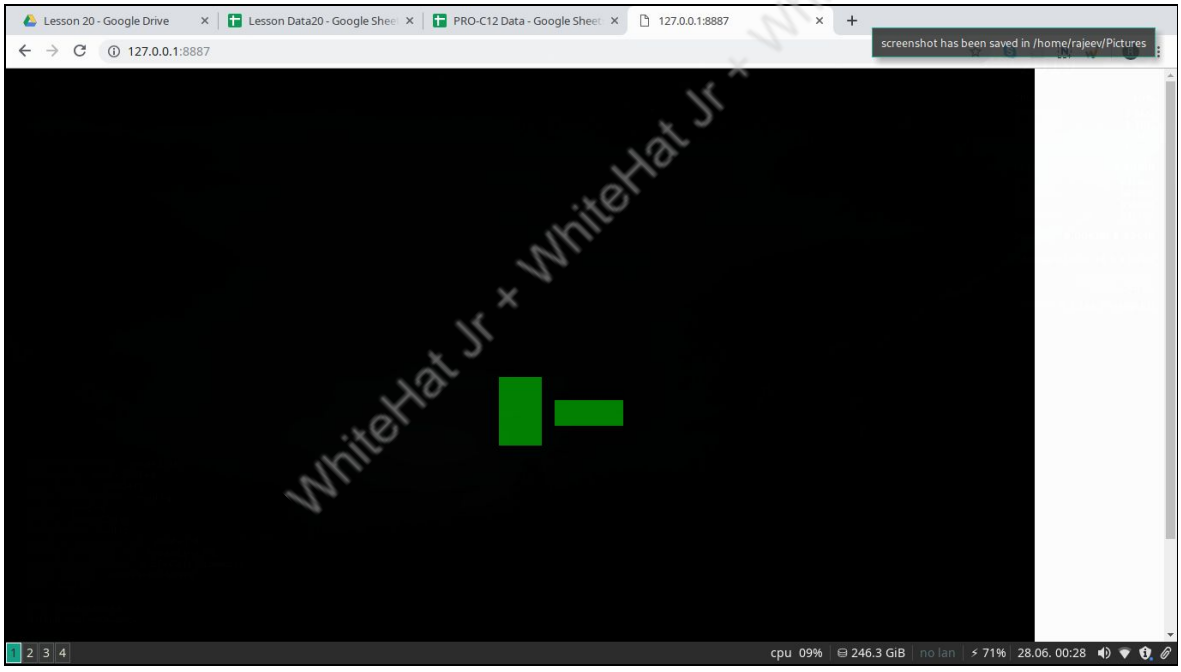
2. Write a function called isTouching. Modify the code to create a function called isTouching()

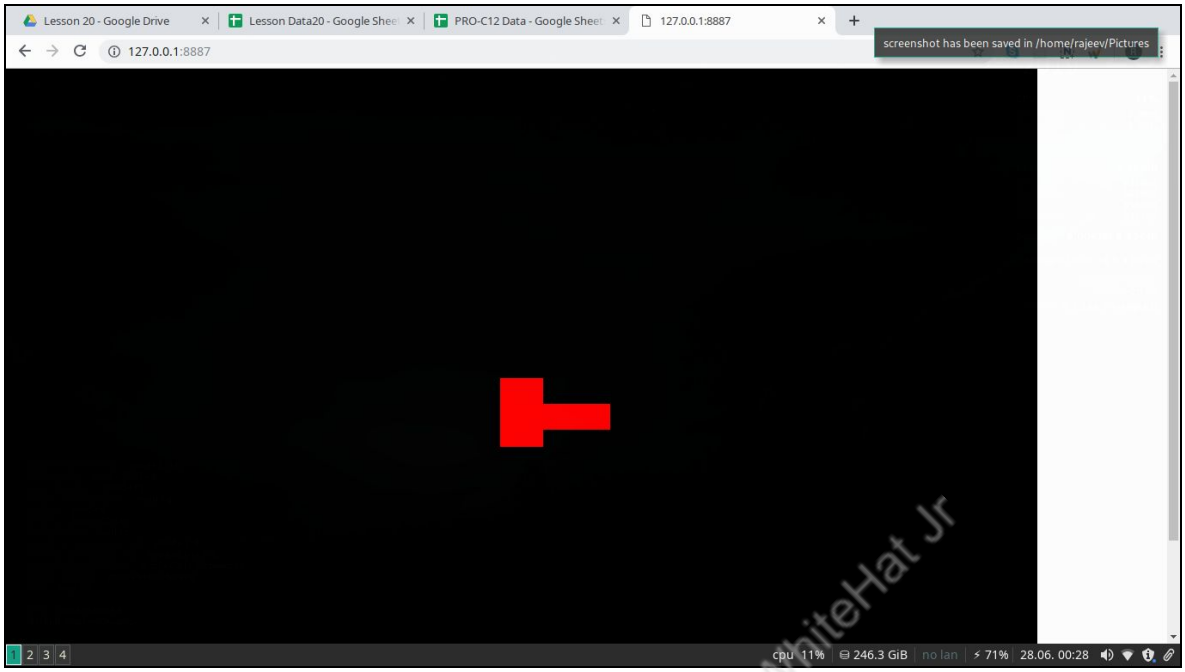
3. Place our code from line 18 to 28 inside isTouching() function— call the isTouching function



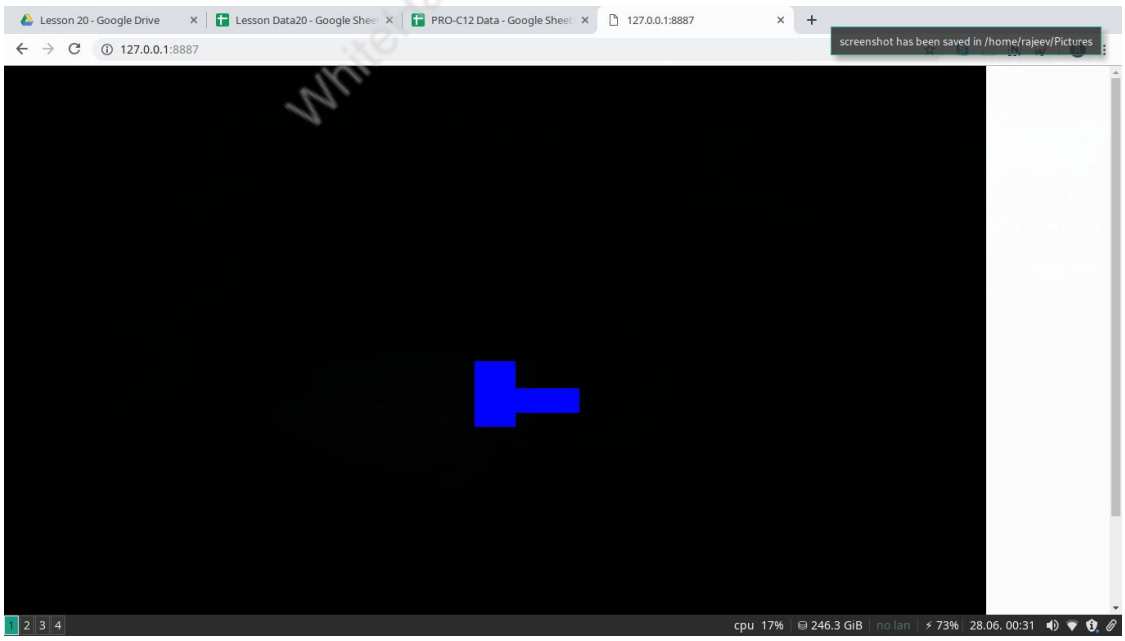
4. Modify our code for the function isTouching() so that it tells "yes" if the two rectangles are touching and "no" if the two rectangles are not touching. In computer language, "yes" and "no" are written as true and false.

```
movingRect = createSprite(800, 400,80,30);
movingRect.shapeColor = "green";

}

function draw() {
  background(0,0,0);
  movingRect.x = World.mouseX;
  movingRect.y = World.mouseY;

  if(isTouching()){
    movingRect.shapeColor = "red";
    movingRect.shapeColor = "red";
  }
  else {
    movingRect.shapeColor = "green";
    movingRect.shapeColor = "green";
  }
  drawSprites();
}

function isTouching(){
  if (movingRect.x - fixedRect.x < fixedRect.width/2 + movingRect.width/2
    && fixedRect.x - movingRect.x < fixedRect.width/2 + movingRect.width/2
    && movingRect.y - fixedRect.y < fixedRect.height/2 + movingRect.height/2
    && fixedRect.y - movingRect.y < fixedRect.height/2 + movingRect.height/2) {

    return true;
  }
  else {
    return false;
  }
}
```

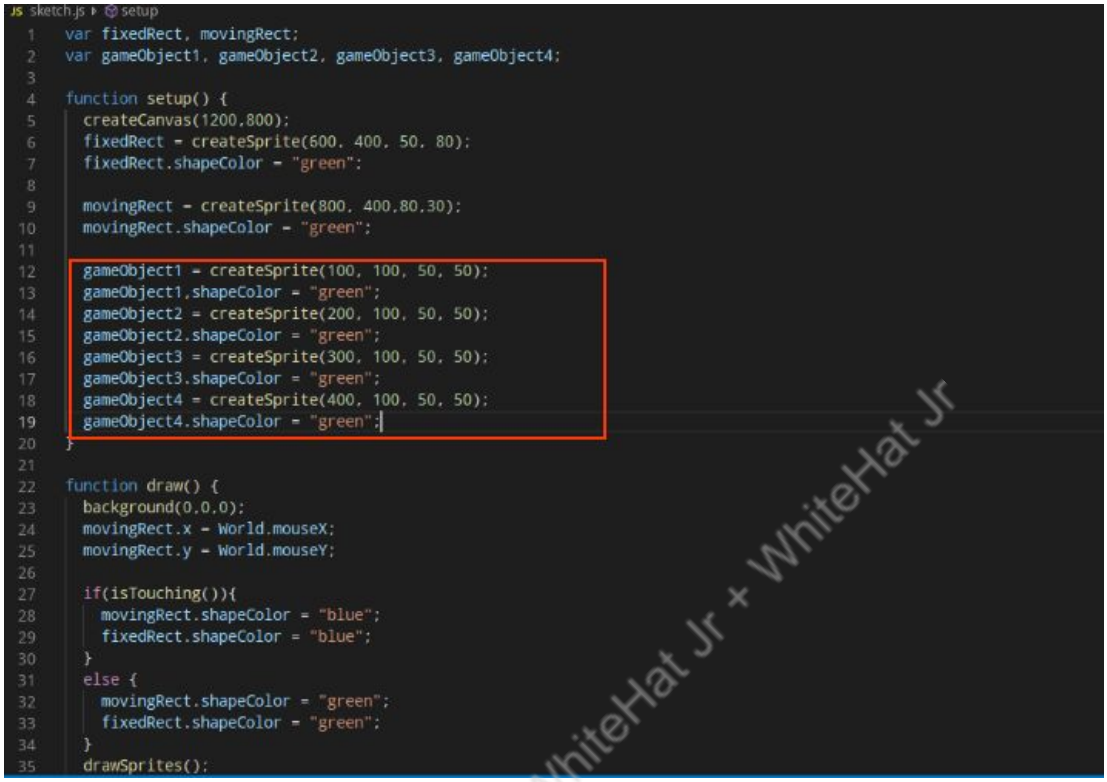




5. Make the rectangles turn to blue instead of red when the two rectangles collide.



6. Create some more game objects (sprites)



```
var fixedRect, movingRect;
var gameObject1, gameObject2, gameObject3, gameObject4;

function setup() {
  createCanvas(1200,800);
  fixedRect = createSprite(600, 400, 50, 80);
  fixedRect.shapeColor = "green";

  movingRect = createSprite(800, 400,80,30);
  movingRect.shapeColor = "green";

  gameObject1 = createSprite(100, 100, 50, 50);
  gameObject1.shapeColor = "green";
  gameObject2 = createSprite(200, 100, 50, 50);
  gameObject2.shapeColor = "green";
  gameObject3 = createSprite(300, 100, 50, 50);
  gameObject3.shapeColor = "green";
  gameObject4 = createSprite(400, 100, 50, 50);
  gameObject4.shapeColor = "green";
}

function draw() {
  background(0,0,0);
  movingRect.x = World.mouseX;
  movingRect.y = World.mouseY;

  if(isTouching()){
    movingRect.shapeColor = "blue";
    fixedRect.shapeColor = "blue";
  }
  else {
    movingRect.shapeColor = "green";
    fixedRect.shapeColor = "green";
  }
  drawSprites();
```

7. Change our function definition to make it accept arguments.

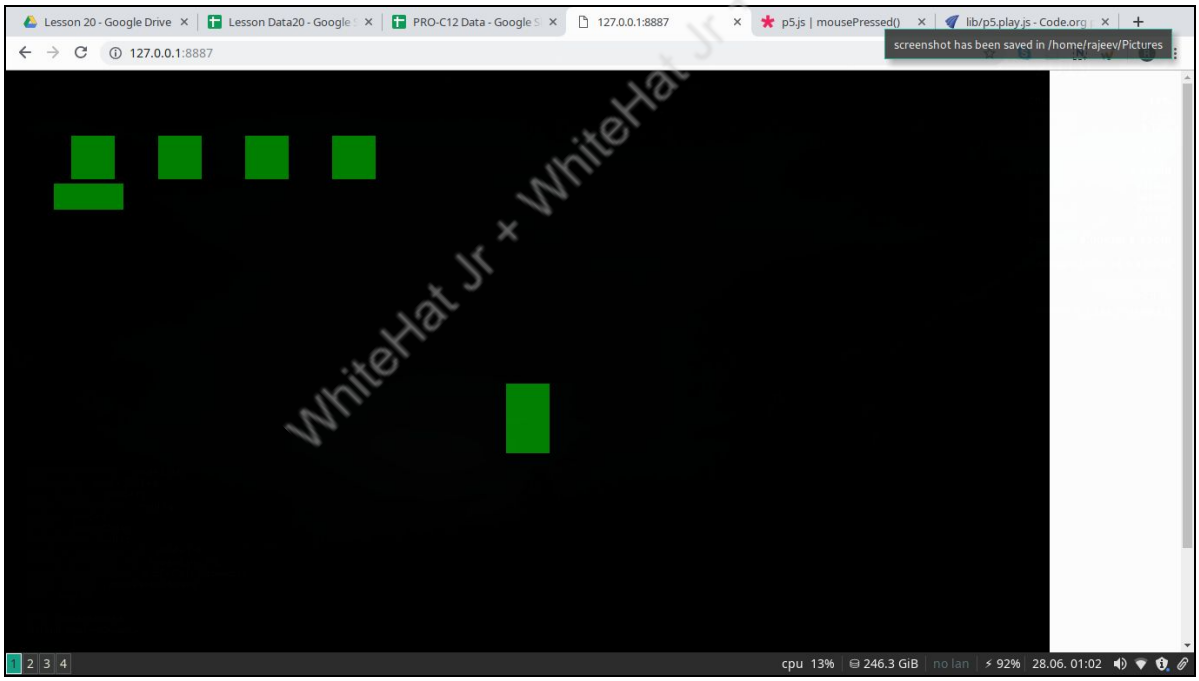
```
gameObject3 = createSprite(300, 100, 50, 50);
gameObject3.shapeColor = "green";
gameObject4 = createSprite(400, 100, 50, 50);
gameObject4.shapeColor = "green";
}

function draw() {
background(0,0,0);
movingRect.x = World.mouseX;
movingRect.y = World.mouseY;

if(isTouching()){
movingRect.shapeColor = "blue";
fixedRect.shapeColor = "blue";
}
else {
movingRect.shapeColor = "green";
fixedRect.shapeColor = "green";
}
drawSprites();
}

function isTouching(object1, object2){
if (movingRect.x - fixedRect.x < fixedRect.width/2 + movingRect.width/2
&& fixedRect.x - movingRect.x < fixedRect.width/2 + movingRect.width/2
&& movingRect.y - fixedRect.y < fixedRect.height/2 + movingRect.height/2
&& fixedRect.y - movingRect.y < fixedRect.height/2 + movingRect.height/2) {

return true;
}
else {
return false;
}
}
```

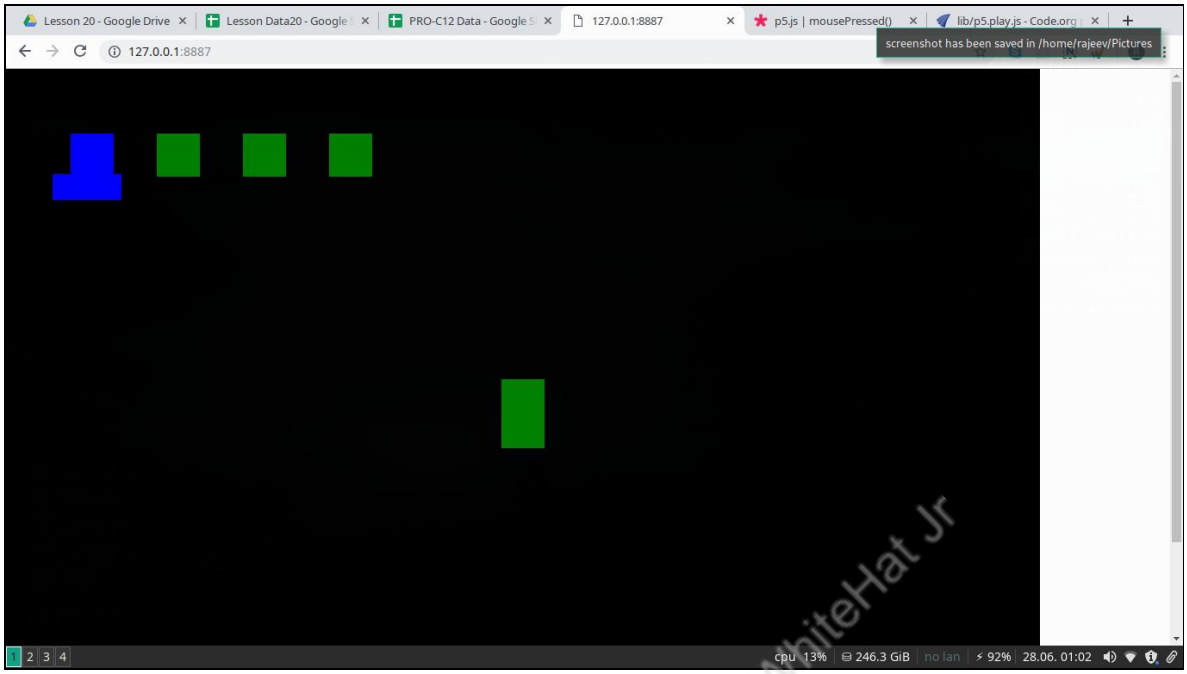
8. Modify the function isTouching() to change movingRect and fixedRect to object1 and object2.

```
gameObject2.shapeColor = "green";
gameObject3 = createSprite(300, 100, 50, 50);
gameObject3.shapeColor = "green";
gameObject4 = createSprite(400, 100, 50, 50);
gameObject4.shapeColor = "green";
}

function draw() {
background(0,0,0);
movingRect.x = World.mouseX;
movingRect.y = World.mouseY;

if(isTouching(movingRect, gameObject1)){
movingRect.shapeColor = "blue";
gameObject1.shapeColor = "blue";
}
else {
movingRect.shapeColor = "green";
gameObject1.shapeColor = "green";
}
drawSprites();
}

function isTouching(object1,object2){
if (object1.x - object2.x < object2.width/2 + object1.width/2
&& object2.x - object1.x < object2.width/2 + object1.width/2
&& object1.y - object2.y < object2.height/2 + object1.height/2
&& object2.y - object2.y < object2.height/2 + object1.height/2) {

return true;
}
else {
return false;
}
}
```

9. Modify the code to check collisions between movingRect and other gameObjects.

```
  gameObject2.shapeColor = "green";
  gameObject3 = createSprite(300, 100, 50, 50);
  gameObject3.shapeColor = "green";
  gameObject4 = createSprite(400, 100, 50, 50);
  gameObject4.shapeColor = "green";
}

function draw() {
  background(0,0,0);
  movingRect.x = World.mouseX;
  movingRect.y = World.mouseY;

  if(isTouching(movingRect, gameObject1)){
    movingRect.shapeColor = "blue";
    gameObject1.shapeColor = "blue";
  }
  else {
    movingRect.shapeColor = "green";
    gameObject1.shapeColor = "green";
  }
  drawSprites();
}

function isTouching(object1,object2){
  if (object1.x - object2.x < object2.width/2 + object1.width/2
    && object2.x - object1.x < object2.width/2 + object1.width/2
    && object1.y - object2.y < object2.height/2 + object1.height/2
    && object2.y - object2.y < object2.height/2 + object1.height/2) {

    return true;
  }
  else {
    return false;
  }
}
```

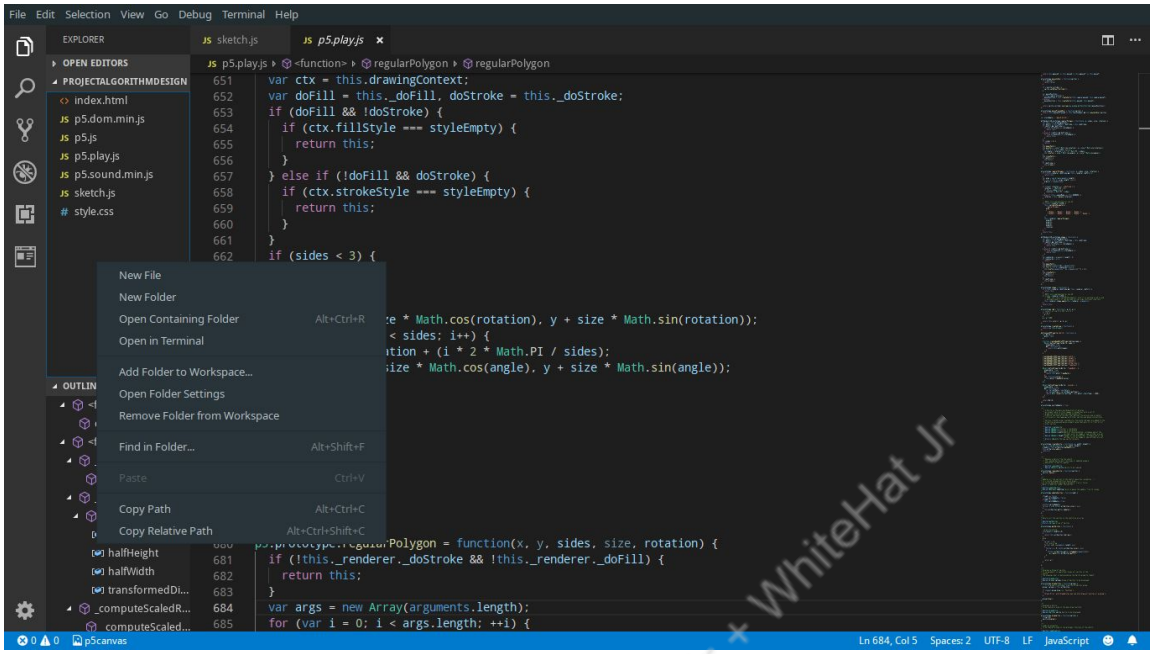




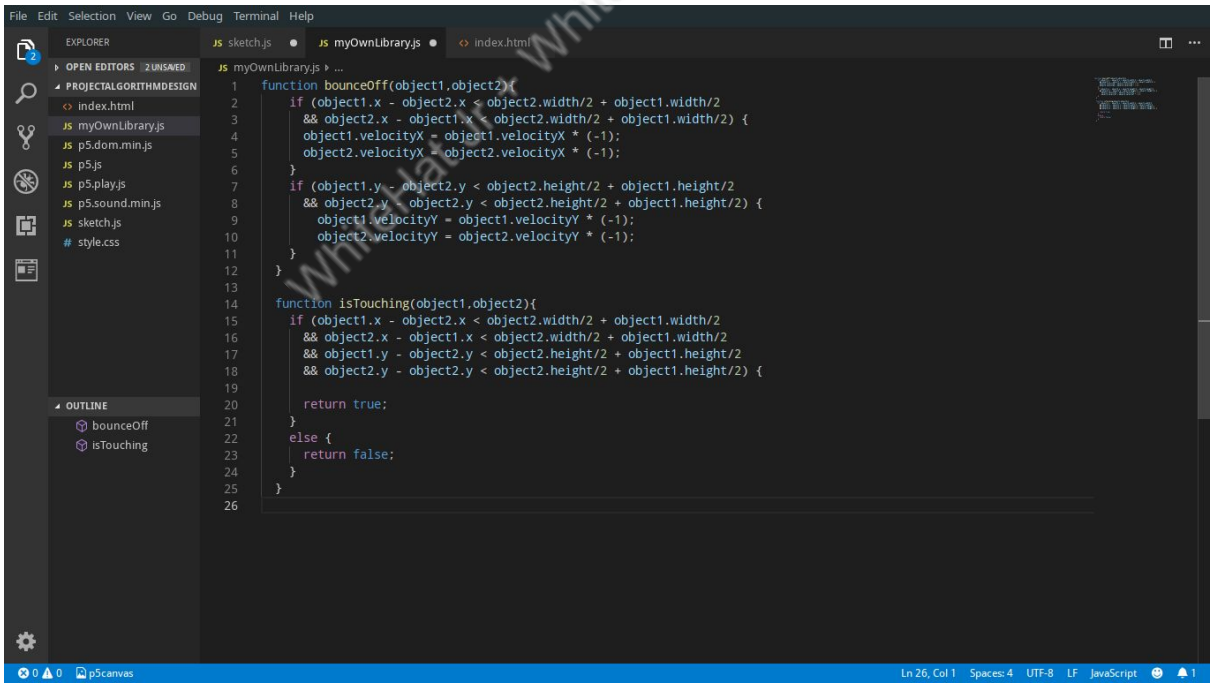
10. Write the bounceOff function and test it.

```
movingRect.debug = true;

movingRect.velocityY = -5;
fixedRect.velocityY = +5;
}

function draw() {
background(0,0,0);

bounceOff(movingRect,fixedRect);
drawSprites();
}

function bounceOff(object1,object2){
if (object1.x - object2.x < object2.width/2 + object1.width/2
&& object2.x - object1.x < object2.width/2 + object1.width/2) {
object1.velocityX = object1.velocityX * (-1);
object2.velocityX = object2.velocityX * (-1);
}
if (object1.y - object2.y < object2.height/2 + object1.height/2
&& object2.y - object2.y < object2.height/2 + object1.height/2) {
object1.velocityY = object1.velocityY * (-1);
object2.velocityY = object2.velocityY * (-1);
}
}
```

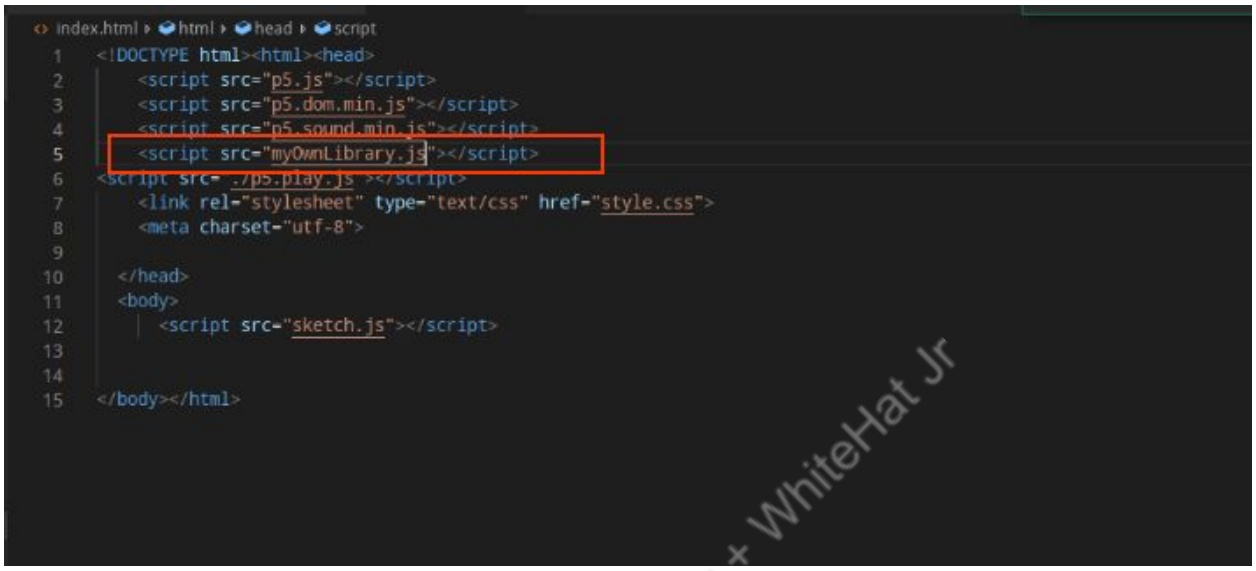
11. Create a file called myOwnLibrary.js



12. Copy the two functions— bounceOff and isTouching created inside myOwnLibrary.js.



13. Include the myOwnLibrary.js in your index.html file. This file can now be included into any project and use the two functions - bounceOff and isTouching - in your code without writing any code!



What's next?

We will get started on creating the Angry Birds game.

Extend your knowledge:

- Go through the following link to Learn more about functions: https://www.w3schools.com/js/js_function_definition.asp