

JavaScript



JavaScript

What is JavaScript?

JavaScript is the programming language used across the world to create dynamic and interactive web applications and games. Along with Python, JavaScript is one of the most popular programming languages on earth.

Unlike HTML and CSS, JavaScript is a proper **programming language** that can:

1. Take input,
2. Do calculations,
3. Make decisions,
4. Repeat instructions
5. Produce output.

The internet would be very different without JavaScript:

Let's get started!



Exercise#1



Adding Machine:

Enter the code below in to **replit** and see what it does. Look carefully at each line of code to see if you figure out what each line is doing. Make sure to display the page in full and **change the number values** and click the “**Try it**” button.

```
<!DOCTYPE html>
<html>
<body>

Number: <input type="number" id="Num1" value="2">
Number: <input type="number" id="Num2" value="2">

<p>Click the button to display the number of the number field.</p>

<button onclick="myFunction()">Try it</button>

<p id="answer"></p>

<script>
function myFunction()
{
    var x = document.getElementById("Num1").value;
    var y = document.getElementById("Num2").value;
    var z = parseInt(x) + parseInt(y);
    document.getElementById("answer").innerHTML=z;
}
</script>

</body>
</html>
```

Exercise#2

Modify the code above to see create a **Conversion calculator** that can take **kilometers** as *input* and then *output* **miles**. Try to make it as **user friendly** as possible with **instructions** the user can read to help them use your calculator.

Exercise#3



Javascript can be an excellent way to create interactive animations and games. Although learning to do this effectively is *beyond the scope of the course*. We will have a look at some of the basics during this exercise. **Cut and paste** the following code into **Replit**. Look carefully at the **script** to see if you can figure out what's going on. Then make the required modifications listed below:

```
<!DOCTYPE html>
<html>
<style>
#container {
  width: 400px;
  height: 400px;
  position: relative;
  background: yellow;
}
#animate {
  width: 50px;
  height: 50px;
  position: absolute;
  background-color: red;
}
</style>
<body>

<p><button onclick="myMove()">Click Me</button></p>

<div id ="container">
  <div id ="animate"></div>
</div>

<script>
function myMove() {
  let id = null;
  const elem = document.getElementById("animate");
  let pos = 0;
  clearInterval(id);
  id = setInterval(frame, 5);
  function frame() {
    if (pos == 350) {
      clearInterval(id);
    } else {
      pos++;
      elem.style.top = pos + "px";
      elem.style.left = pos + "px";
    }
  }
}
</script>
</body>
</html>
```

Cut and paste the code below into **Replit**, run the code, look at each line of the code to see if you can get an idea of what it's doing, then **make the following modifications (each as separate HTML files)**:

1. Make the red square **only move in the x direction** (move to the right, but not down) Hint: look at **green** code.
2. Make the red square only move in the x direction to about the **midpoint** of the yellow square.
3. Make the red square move in the x-direction to the middle of the yellow square and then move to the bottom of the yellow square and stop...Tough one. **Examine the solution** on the next page if you can't figure it out.

Solution to Exercise#3 part-3

Blue Indicates the important changes

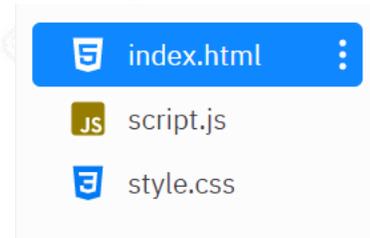
```
function myMove() {
  let id = null;
  const elem = document.getElementById("animate");
  let posx = 0;
  let posy = 0;
  clearInterval(id);
  id = setInterval(frame, 5);
  function frame()
  {
    if (posx == 150)
    {
      posy++;
      elem.style.top = posy + "px";
      if (posy==250)
      {
        clearInterval(id)
      }
    }
    else
    {
      posx++;
      elem.style.left = posx + "px";
    }
  }
}
```

Exercise#4

The code on the next page has some **HTML** some **CSS** and some **JavaScript**.

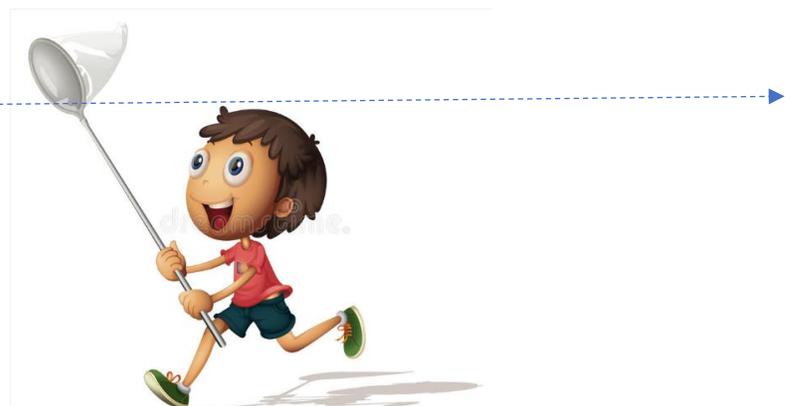
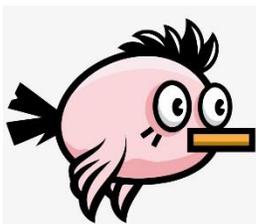
The code is supposed to create an interactive animation that let's you *move a green circle* around the screen with the **arrow keys**. Please do the following:

- Copy and paste the code into Replit. **Make sure you separate the HTML, CSS, and JavaScript into their own sections (pages).**
- Run the code and use the arrow keys to move the green circle around the screen
- Try to modify the code in the following ways (2 marks each):
 - See if you can replace the circle with an image (PNG with no background) of a person or character that you can move around
 - See if you can make the animation smoother by changing the `moveBy` variable or any other aspect of the JavaScript
 - See if you can make your character move to the centre of the screen when you press the 't' key (teleport!).



Bonus:

Add to the **JavaScript** so that if your character comes in contact with a new **bird** character that moves horizontally across the screen, you get a point!



HTML

```
<!DOCTYPE html>
<html>
<head>

  <link rel="stylesheet" href="style.css">
  <title>Document</title>
</head>
<body>

  <div class="circle"></div>
  <script src="script.js"></script>
</body>
</html>
```

CSS

```
.circle{
  height: 100px;
  width: 100px;
  border-radius: 50%;
  background-color: green;
}
```

JavaScript

```
let circle = document.querySelector('.circle');
let moveBy = 10;
window.addEventListener('load', () => {
  circle.style.position = 'absolute';
  circle.style.left = 0;
  circle.style.top = 0;
});
window.addEventListener('keyup', (e) => {
  switch (e.key) {
    case 'ArrowLeft':
      circle.style.left = parseInt(circle.style.left) - moveBy + 'px';
      break;
    case 'ArrowRight':
      circle.style.left = parseInt(circle.style.left) + moveBy + 'px';
      break;
    case 'ArrowUp':
      circle.style.top = parseInt(circle.style.top) - moveBy + 'px';
      break;
    case 'ArrowDown':
      circle.style.top = parseInt(circle.style.top) + moveBy + 'px';
      break;
  }
});
```