

Lab Exercise – JavaScript

1. Internal JavaScript

- In HTML, JavaScript code is inserted between **<script>** and **</script>** tags.

Example:

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

    <h2>JavaScript in Body</h2>

    <p id="demo"></p>

    <script>
      document.getElementById("demo").innerHTML = "My First JavaScript";
    </script>

  </body>
</html>
```

2. External JavaScript

- Scripts can also be placed in external files.
- JavaScript files have the file extension **.js**.
- Put the name of the script file in the src (source) attribute of a <script> tag.
Example: **<script src="myScript.js"></script>**

Example:

External File: myScript.js

```
function myFunction() {
  document.getElementById("demo").innerHTML = "Paragraph changed.";
}
```

HTML File:

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

    <h2>External JavaScript</h2>

    <p id="demo">A Paragraph.</p>

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

    <p>This example links to "myScript.js".</p>

    <p>(myFunction is stored in "myScript.js")</p>

    <script src="myScript.js"></script>

  </body>
</html>
```

3. Change HTML Content

- One of many JavaScript HTML methods is `getElementById()`.
- The example below "finds" an HTML element (with id="demo") and changes the element content (innerHTML) to "Hello JavaScript".

Example:

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

    <h2>What Can JavaScript Do?</h2>

    <p id="demo">JavaScript can change HTML content.</p>

    <button type="button" onclick="document.getElementById("demo").innerHTML =
      "Hello JavaScript!">Click Me!</button>

  </body>
</html>
```

4. Change HTML Attribute Values

- In this example JavaScript changes the value of the src (source) attribute of an tag.

Example:

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

    <h2>What Can JavaScript Do?</h2>

    <p>JavaScript can change HTML attribute values.</p>

    <p>In this case JavaScript changes the value of the src (source) attribute of an
      image.</p>

    <button onclick="document.getElementById('myImage').src=
      'pic_bulbon.gif'">Turn on the light</button>

    

    <button onclick="document.getElementById('myImage').src=
      'pic_bulboff.gif'">Turn off the light</button>

  </body>
</html>
```

5. Change HTML Styles (CSS)

Example:

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

    <h2>What Can JavaScript Do?</h2>

    <p id="demo">JavaScript can change the style of an HTML element.</p>

    <button type="button"
      onclick="document.getElementById('demo').style.fontSize='35px'">Click
      Me!</button>

  </body>
</html>
```

6. Hide HTML Elements

- Hiding HTML elements can be done by changing the **display** style to “**none**”

Example:

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

    <h2>What Can JavaScript Do?</h2>

    <p id="demo">JavaScript can hide HTML elements.</p>

    <button type="button"
onclick="document.getElementById('demo').style.display='none'">Click
Me!</button>

  </body>
</html>
```

7. Show HTML Elements

- Showing hidden HTML elements can also be done by changing the **display** style to “**block**”

Example:

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

    <h2>What Can JavaScript Do?</h2>

    <p>JavaScript can show hidden HTML elements.</p>

    <p id="demo" style="display:none">Hello JavaScript!</p>

    <button type="button"
onclick="document.getElementById('demo').style.display='block'">Click
Me!</button>

  </body>
</html>
```

8. JavaScript Output

JavaScript can "display" data in different ways:

- Writing into an HTML element, using innerHTML or innerText.
- Writing into the HTML output using document.write().
- Writing into an alert box, using window.alert().
- Writing into the browser console, using console.log().

a) Using innerHTML

- To access an HTML element, you can use the `document.getElementById(id)` method.
- Use the `id` attribute to identify the HTML element.
- Then use the `innerHTML` property to change the HTML content of the HTML element:

Example:

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

    <h1>My Web Page</h1>

    <p id="demo"></p>

    <script>
      document.getElementById("demo").innerHTML = "<h2>Hello World</h2>";
    </script>

  </body>
</html>
```

b) Using document.write()

- For testing purposes, it is convenient to use `document.write()`:
- Using `document.write()` after an HTML document is loaded, will **delete all existing HTML**
- The `document.write()` method should only be used for testing.

Example:

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

    <h2>My First Web Page</h2>
    <p>My first paragraph.</p>

    <p>Never call document.write after the document has finished loading.
    It will overwrite the whole document.</p>

    <script>
      document.write(5 + 6);
    </script>

  </body>
</html>
```

9. JavaScript Variables

- Variables are Containers for Storing Data
- JavaScript Variables can be declared in 4 ways:
 - Automatically
 - Using var
 - Using let
 - Using const

a) Automatically

In this example, x, y, and z are undeclared variables.
They are automatically declared when first used:

Example:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>JavaScript Variables</h1>

    <p>In this example, x, y, and z are undeclared.</p>
    <p>They are automatically declared when first used.</p>

    <p id="demo"></p>

    <script>
      x = 5;
      y = 6;
      z = x + y;
      document.getElementById("demo").innerHTML =
        "The value of z is: " + z;
    </script>

  </body>
</html>
```

b) Using var

Example:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>JavaScript Variables</h1>

    <p>In this example, x, y, and z are variables.</p>

    <p id="demo"></p>

    <script>
      var x = 5;
      var y = 6;
      var z = x + y;
      document.getElementById("demo").innerHTML =
        "The value of z is: " + z;
    </script>
  </body>
</html>
```

c) Using let

Example:

```
<!DOCTYPE html>
<html>
  <body>
    <h1>JavaScript Variables</h1>

    <p>In this example, x, y, and z are variables.</p>

    <p id="demo"></p>

    <script>
      let x = 5;
      let y = 6;
      let z = x + y;
      document.getElementById("demo").innerHTML =
        "The value of z is: " + z;
    </script>

  </body>
</html>
```

d) Using Const

Example:

```
<!DOCTYPE html>

<html>
  <body>
    <h1>JavaScript Variables</h1>

    <p>In this example, x, y, and z are variables.</p>

    <p id="demo"></p>

    <script>
      const x = 5;
      const y = 6;
      const z = x + y;
      document.getElementById("demo").innerHTML =
        "The value of z is: " + z;
    </script>

  </body>
</html>
```

e) Mixed Example

- In this example, the two variables price1 and price2 are declared with the const keyword.
- These are constant values and cannot be changed.
- The variable total is declared with the let keyword.
- The value total can be changed.

Example:

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

    <h1>JavaScript Variables</h1>
    <p>In this example, price1, price2, and total are variables.</p>
    <p id="demo"></p>

    <script>
      const price1 = 5;
      const price2 = 6;
      let total = price1 + price2;
      document.getElementById("demo").innerHTML =
        "The total is: " + total;
    </script>

  </body>
</html>
```


10. JavaScript Operators

Javascript operators are used to perform different types of mathematical and logical computations.

Examples:

- The Assignment Operator = assigns values
- The Addition Operator + adds values
- The Multiplication Operator * multiplies values
- The Comparison Operator > compares values

a) Assignment Operator

The **Assignment Operator** (=) assigns a value to a variable:

Example:

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

    <h1>JavaScript Operators</h1>
    <h2>The Assignment (=) Operator</h2>

    <p id="demo"></p>

    <script>
      // Assign the value 5 to x
      let x = 5;
      // Assign the value 2 to y
      let y = 2;
      // Assign the value x + y to z
      let z = x + y;
      // Display z
      document.getElementById("demo").innerHTML = "The sum of x + y is: " +
      z;
    </script>

  </body>
</html>
```

11. JavaScript Comments

a) Single line comment

- Single line comments start with `//`.
- Any text between `//` and the end of the line will be ignored by JavaScript (will not be executed).

Example:

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

    <h1 id="myH"></h1>
    <p id="myP"></p>

    <script>
      // Change heading:
      document.getElementById("myH").innerHTML = "JavaScript Comments";
      // Change paragraph:
      document.getElementById("myP").innerHTML = "My first paragraph.";
    </script>

  </body>
</html>
```

b) Multiple lines comment

- Multi-line comments start with `/*` and end with `*/`.
- Any text between `/*` and `*/` will be ignored by JavaScript.

Example:

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

    <h1 id="myH"></h1>
    <p id="myP"></p>

    <script>
      /*
      The code below will change
      the heading with id = "myH"
      and the paragraph with id = "myP"
      */
      document.getElementById("myH").innerHTML = "JavaScript Comments";
      document.getElementById("myP").innerHTML = "My first paragraph.";
    </script>

  </body>
</html>
```

12. JavaScript Functions

A JavaScript function is

- a block of code designed to perform a particular task.
- executed when "something" invokes it (calls it).
- defined with the **function** keyword, followed by a **name**, followed by parentheses ().
 - Function names can contain letters, digits, underscores, and dollar signs (same rules as variables).
 - The parentheses may include parameter names separated by commas: **(parameter1, parameter2, ...)**
 - The code to be executed, by the function, is placed inside curly brackets: {}

Example:

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

    <h1>JavaScript Functions</h1>

    <p>Call a function which performs a calculation and returns the
      result:</p>

    <p id="demo"></p>

    <script>
      let x = myFunction(4, 3);
      document.getElementById("demo").innerHTML = x;

      function myFunction(a, b) {
        return a * b;
      }
    </script>

  </body>
</html>
```

13. Conditional Statements

a) The if Statement

Example:

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

    <h2>JavaScript if</h2>

    <p>Display "Good day!" if the hour is less than 18:00:</p>

    <p id="demo">Good Evening!</p>

    <script>
      if (new Date().getHours() < 18) {
        document.getElementById("demo").innerHTML = "Good day!";
      }
    </script>

  </body>
</html>
```

b) The else Statement

Example:

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

    <h2>JavaScript if .. else</h2>

    <p>A time-based greeting:</p>

    <p id="demo"></p>

    <script>
      const hour = new Date().getHours();
      let greeting;

      if (hour < 18) {
        greeting = "Good day";
      } else {
        greeting = "Good evening";
      }

      document.getElementById("demo").innerHTML = greeting;
    </script>

  </body>
</html>
```

c) Switch Statement

Example:

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

    <h2>JavaScript switch</h2>

    <p id="demo"></p>

    <script>
      let day;
      switch (new Date().getDay()) {
        case 0:
          day = "Sunday";
          break;
        case 1:
          day = "Monday";
          break;
        case 2:
          day = "Tuesday";
          break;
        case 3:
          day = "Wednesday";
          break;
        case 4:
          day = "Thursday";
          break;
        case 5:
          day = "Friday";
          break;
        case 6:
          day = "Saturday";
        }
      document.getElementById("demo").innerHTML = "Today is " + day;
    </script>

  </body>
</html>
```

14. JavaScript Loops

a) For Loop

Example:

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

    <h2>JavaScript For Loop</h2>

    <p id="demo"></p>

    <script>
      let text = "";

      for (let i = 0; i < 5; i++) {
        text += "The number is " + i + "<br>";
      }

      document.getElementById("demo").innerHTML = text;
    </script>

  </body>
</html>
```

b) While Loop

Example:

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

    <h2>JavaScript While Loop</h2>

    <p id="demo"></p>

    <script>
      let text = "";
      let i = 0;
      while (i < 10) {
        text += "<br>The number is " + i;
        i++;
      }
      document.getElementById("demo").innerHTML = text;
    </script>

  </body>
</html>
```