MODULE 1

IT ELEC 4

SYSTEMS INTEGRATION & ARCHITECTURE II

BSIT 4

```
<script>
if(age > 19){
    alert("Adult");
} else{
    alert("Teenager");
}
</script>
```

INTRODUCTION

As the development of web technology, and by far, this course will be the continuation of the previous topics we discussed.

In this module, we will learn the JavaScript fundamentals you'll need for front-end or back-end development.

OBJECTIVES

By the end of this module the students will be able to:

- Learn the concepts and basic of JavaScript
- Identify the front-end and back-end use of JavaScript

KEYWORDS

Scripting, JavaScript, Web Development, Programming

JAVASCRIPT

- JavaScript is the world's most popular programming language.
- JavaScript is the programming language of the Web.
- JavaScript is easy to learn.

WHY JAVASCRIPT?

JavaScript is one of the 3 languages all web developers must learn:

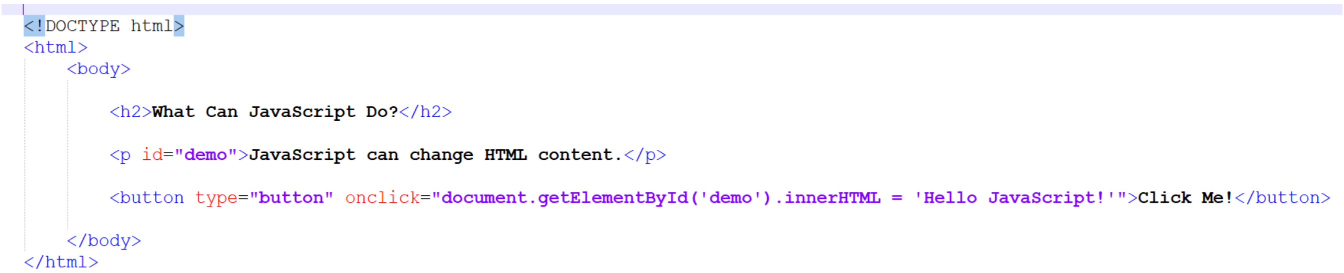
1. HTML to define the content of web pages
2. CSS to specify the layout of web pages
3. JavaScript to program the behavior of web pages

JavaScript Can Change HTML Content

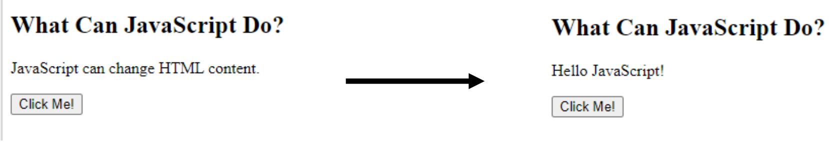
One of many JavaScript HTML methods is getElementById().

The example below "finds" an HTML element (with id="demo"), and changes the element content (innerHTML) to "Hello JavaScript":

```
document.getElementById("demo").innerHTML = "Hello JavaScript";
```



```
<!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>
```



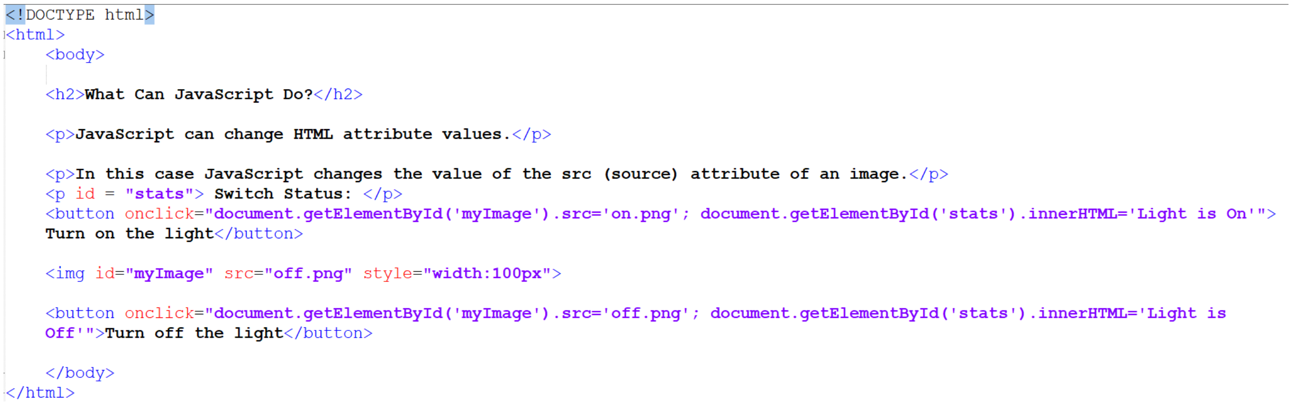
What Can JavaScript Do?

JavaScript can change HTML content.

What Can JavaScript Do?

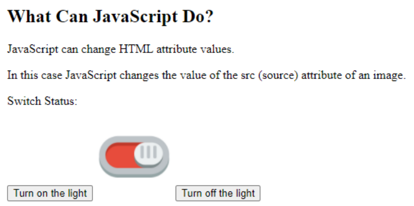
Hello JavaScript!

Where on the page of your HTML, once you click “Click me”, it will show you the output above



```
<!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>
    <p id = "stats"> Switch Status: </p>
    <button onclick="document.getElementById('myImage').src='on.png'; document.getElementById('stats').innerHTML='Light is On'">
    Turn on the light</button>

    <img id="myImage" src="off.png" style="width:100px">

    <button onclick="document.getElementById('myImage').src='off.png'; document.getElementById('stats').innerHTML='Light is
    Off'">Turn off the light</button>

    </body>
</html>
```

Another example on the previous page will have the output below.

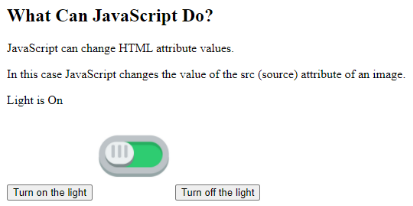


What Can JavaScript Do?

JavaScript can change HTML attribute values.

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

Switch Status:

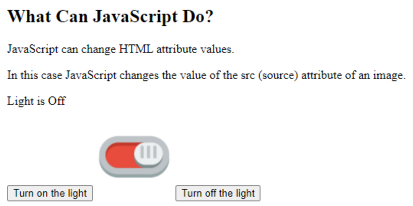


What Can JavaScript Do?

JavaScript can change HTML attribute values.

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

Light is On



What Can JavaScript Do?

JavaScript can change HTML attribute values.

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

Light is Off

Once you clicked the button you create, you will see the output above.

Doing JavaScript is the same with having an inline coding like CSS.

You can create multi-inline coding with the help of “;” to separate statements.

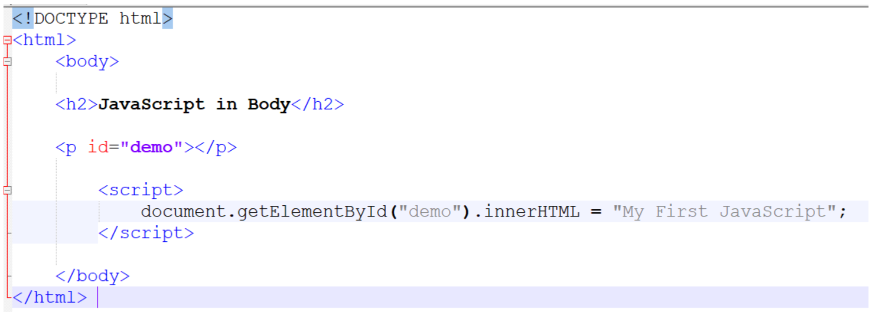
Note: You will find the example in our LMS named lightbulb.zip.

On your first activity modify (edit the images given or manipulate the text) the given example and submit it in our LMS named pre-Activity

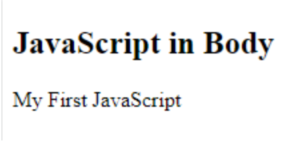
WHERE TO PLACE JS

The <script> Tag

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

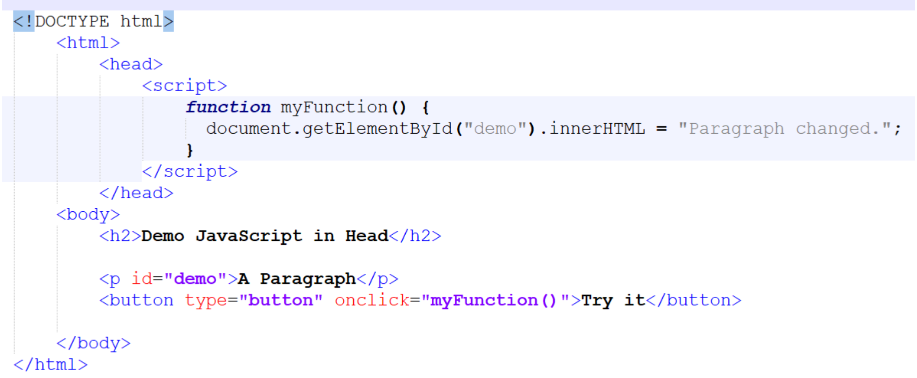


```
<!DOCTYPE html>
<html>
    <body>

    <h2>JavaScript in Body</h2>

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

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

    </body>
</html>
```

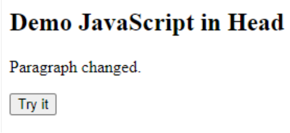


JavaScript in Body

My First JavaScript



```
<!DOCTYPE html>
    <html>
        <head>
            <script>
                function myFunction() {
                  document.getElementById("demo").innerHTML = "Paragraph changed.";
                }
            </script>
        </head>
    <body>
        <h2>Demo JavaScript in Head</h2>

        <p id="demo">A Paragraph</p>
        <button type="button" onclick="myFunction()">Try it</button>

    </body>
</html>
```



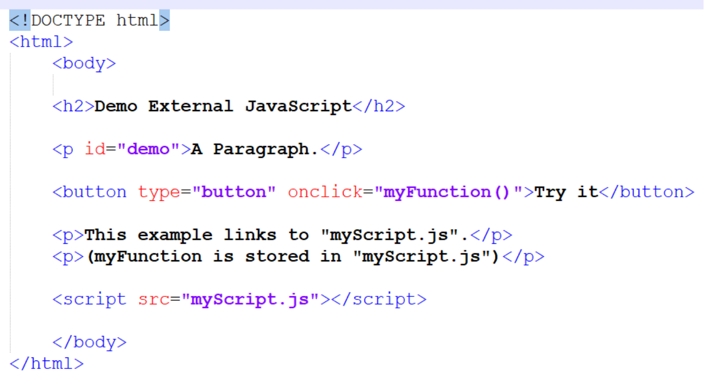
Demo JavaScript in Head

Paragraph changed.

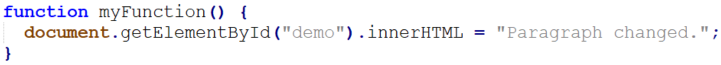
It will have an output the same as the code you’ve written before in the inline code.

The above example is written in the <head> or <body> tags. On next page we will discuss the external JavaScript

Example below is the external JavaScript



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

    <h2>Demo 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>
```



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

The example has two files named index.html and myScript.js.

This example is also uploaded in your LMS named external.zip

External scripts are practical when the same code is used in many different web pages.

JavaScript files have the file extension .js.

To use an external script, put the name of the script file in the src (source) attribute of a <script> tag.

You can place an external script reference in <head> or <body> as you like.

The script will behave as if it was located exactly where the <script> tag is located.

External JavaScript Advantages

Placing scripts in external files has some advantages:

- It separates HTML and code
- It makes HTML and JavaScript easier to read and maintain
- Cached JavaScript files can speed up page loads

External References

An external script can be referenced in 3 different ways:

- With a full URL (a full web address)
- With a file path (like /js/)
- Without any path

```
<script src="https://www.w3schools.com/js/myScript.js"></script>  // full URL
<script src="/js/myScript.js"></script>  // file path
```

JAVASCRIPT OUTPUT

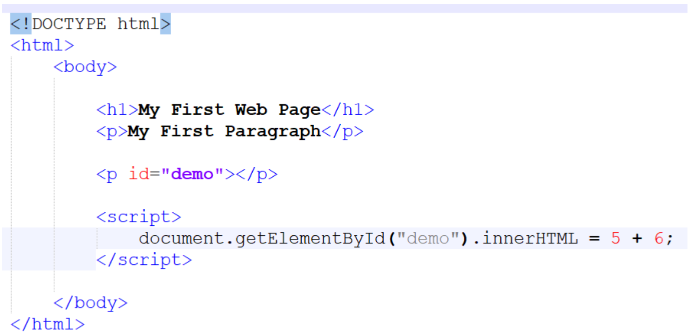
JavaScript can "display" data in different ways:

- Writing into an HTML element, using innerHTML.
- Writing into the HTML output using document.write().
- Writing into an alert box, using window.alert().
- Writing into the browser console, using console.log().
- Getting value from the html tags getElementById(id).value.

Using innerHTML

To access an HTML element, JavaScript can use the document.getElementById(id) method.

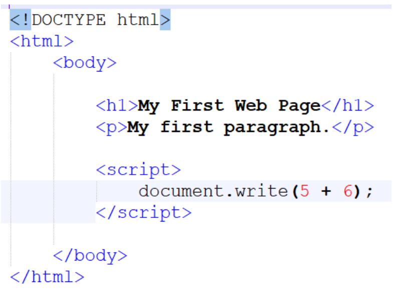
The id attribute defines the HTML element. The innerHTML property defines the HTML content:



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

        <h1>My First Web Page</h1>
        <p>My First Paragraph</p>

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

        <script>
            document.getElementById("demo").innerHTML = 5 + 6;
        </script>

    </body>
</html>
```

Using document.write()

For testing purposes, it is convenient to use document.write():



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

        <h1>My First Web Page</h1>
        <p>My first paragraph.</p>

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

    </body>
</html>
```

Using getElementById(id).value

For testing purposes, it is convenient to use getElementById(id).value:

```
var myname = document.getElementById
("myname").value;
```

//it will get value inside your html tag with the id of myname

Using windows.alert()

It will use an alert box to display data

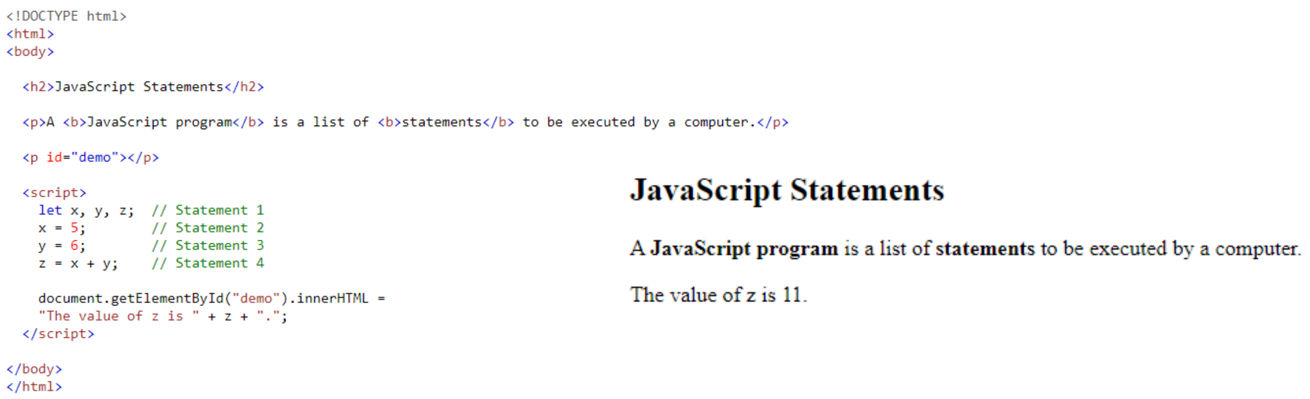
```
<script>
        window.alert(5 + 6);
</script>
```

Using console.log()

This is for debugging purposes, which you will use in the browser to display data

```
<script>
        console.log(5 + 6);
</script>
```

JAVASCRIPT STATEMENTS



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

  <h2>JavaScript Statements</h2>

  <p>A <b>JavaScript program</b> is a list of <b>statements</b> to be executed by a computer.</p>

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

  <script>
    let x, y, z;  // Statement 1
    x = 5;        // Statement 2
    y = 6;        // Statement 3
    z = x + y;    // Statement 4

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

</body>
</html>
```

JavaScript Statements

A **JavaScript program** is a list of **statements** to be executed by a computer.

The value of z is 11.

JavaScript Programs

A computer program is a list of "instructions" to be "executed" by a computer.

In a programming language, these programming instructions are called statements.

A JavaScript program is a list of programming statements.

In HTML, JavaScript programs are executed by the web browser.

JavaScript Statements

JavaScript statements are composed of:

Values, Operators, Expressions, Keywords, and Comments.

This statement tells the browser to write "Hello Dolly." inside an HTML element with id="demo": document.getElementById("demo").innerHTML = "Hello Dolly.";

JavaScript programs (and JavaScript statements) are often called JavaScript code.

JAVASCRIPT SYNTAX

JavaScript syntax is the set of rules, how JavaScript programs are constructed:

```
//How to create variables:

      var x;
      let y;

// How to use variables:

      x=5;
      y=6;
      let z = x + y;
```

JavaScript Values

The JavaScript syntax defines two types of values:

- Fixed values
- Variable values

Fixed values are called **Literals.**

Variable values are called **Variables.**

JavaScript Operators

JavaScript uses arithmetic operators (+ - * /) to compute values:

```
(5 + 6) * 10
let x, y;
X = 5;
y = 6;
```

JavaScript Expressions

An expression is a combination of values, variables, and operators, which computes to a value.

The computation is called an evaluation.

For example, 5 * 10 evaluates to 50:

```
5 * 10
X * 10
```

The values can be of various types, such as numbers and strings.

For example, "John" + " " + "Doe", evaluates to "John Doe":

```
"John" + " " + "Doe"
```

SOME NOTE:

JavaScript is Case Sensitive

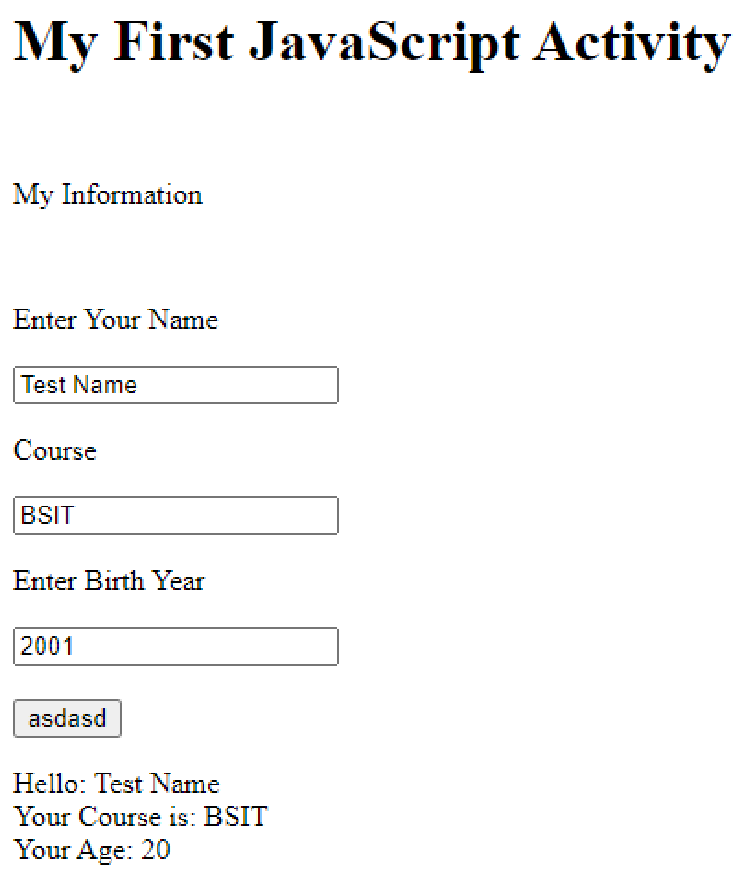
All JavaScript identifiers are case sensitive.

The variables lastName and lastname, are two different variables:

```
let lastname, lastName;
lastName = "Doe";
lastname = "Peterson";
```

ACTIVITY

Create a Program that will execute the image below



My First JavaScript Activity

My Information

Enter Your Name

Test Name

Course

BSIT

Enter Birth Year

2001

Hello: Test Name
Your Course is: BSIT
Your Age: 20

Note:

The program is very simple, based on the given modules and example, execute the given image above using JavaScript.

Submit it in image format in our LMS.

No need for the code, just the image.

MILAN E. BAUSA, MIT
Instructor I
Sorsogon State College – Bulan Campus
Bulan, Sorsogon

✉ great_lhan@yahoo.com / testlhan@gmail.com
☎ +63 909 6267 670 / +63 966 1466 481
f https://www.facebook.com/Ferbilicious

NEXT LESSON: JQUERY