Background and Overview:

JavaScript:

JavaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which (HTML and CSS) we have covered in much more detail in other parts of the Learning Area.

What is JavaScript doing on your page?

Here we'll actually start looking at some code, and while doing so, explore what actually happens when you run some JavaScript in your page.

Let's briefly recap the story of what happens when you load a web page in a browser (first talked about in our How CSS works article). When you load a web page in your browser, you are running your code (the HTML, CSS, and JavaScript) inside an execution environment (the browser tab). This is like a factory that takes in raw materials (the code) and outputs a product (the web page).

A very common use of JavaScript is to dynamically modify HTML and CSS to update a user interface, via the Document Object Model API (as mentioned above). Note that the code in your web documents is generally loaded and executed in the order it appears on the page. Errors may occur if JavaScript is loaded and run before the HTML and CSS that it is intended to modify. You will learn ways around this later in the article, in the Script loading strategies section.

JavaScript HTML DOM:

With the HTML DOM, JavaScript can access and change all the elements of an HTML document.

The HTML DOM (Document Object Model), When a web page is loaded, the browser creates a Document Object Model of the page.

With the object model, JavaScript gets all the power it needs to create dynamic HTML:

- JavaScript can change all the HTML elements in the page
- JavaScript can change all the HTML attributes in the page
- JavaScript can change all the CSS styles in the page
- JavaScript can remove existing HTML elements and attributes
- JavaScript can add new HTML elements and attributes
- JavaScript can react to all existing HTML events in the page
- JavaScript can create new HTML events in the page

What You Will Learn

- How to change the content of HTML elements
- How to change the style (CSS) of HTML elements
- How to react to HTML DOM events
- How to add and delete HTML elements

What is the HTML DOM?

The HTML DOM is a standard object model and programming interface for HTML. It defines:

- The HTML elements as objects
- The properties of all HTML elements
- The methods to access all HTML elements
- The events for all HTML elements

In other words: The HTML DOM is a standard for how to get, change, add, or delete HTML elements.

JavaScript - HTML DOM Methods:

HTML DOM methods are actions you can perform (on HTML Elements).

HTML DOM properties are values (of HTML Elements) that you can set or change.

A **property** is a value that you can get or set (like changing the content of an HTML element).

A **method** is an action you can do (like add or deleting an HTML element).

Including JavaScript in Your Page:

Including JavaScript in your page is a fairly simple process.

You can include JavaScript in your HTML in two ways:

- Writing the code in your HTML
- Including it as a link to an external file

For the most part, you will include the JavaScript as an external file.

The Script Tag

The <script> tag is what we use to includes our JavaScript. It's a lot like the <link> tag you've already been using to include your CSS files.

Here's a very basic snippet of JavaScript using the script tag. This JavaScript is written directly into our HTML page. It will call and alert box as soon as the page loads.

```
<script type="text/javascript">
        alert("This alert box was called with the onload event");
</script>
```

When using the script tag, we must always use the attribute name and value of type="text/javascript".

Using the script tag to include an external JavaScript file

To include an external JavaScript file, we can use the script tag with the attribute src. You've already used the src attribute when using images. The value for the src attribute should be the path to your JavaScript file.

```
<script type="text/javascript" src="path-to-javascript-file.js"></script>
```

This script tag should be included between the <head> tags in your HTML document.

Example Code:

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

<h2>My First Page</h2>

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

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

</body>
</html>
```

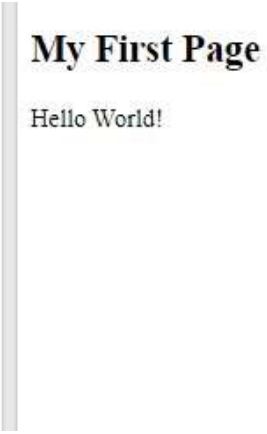

Fig. 1 (Script Tag)

In the example above, **getElementById** is a method, while **innerHTML** is a property.

The getElementById Method

The most common way to access an HTML element is to use the id of the element. In the example above the getElementById method used id="demo" to find the element.

The innerHTML Property:

The easiest way to get the content of an element is by using the innerHTML property. The innerHTML property is useful for getting or replacing the content of HTML elements.

The HTML DOM Document Object

The document object represents your web page. If you want to access any element in an HTML page, you always start with accessing the document object.

Below are some examples of how you can use the document object to access and manipulate HTML.

Finding HTML Elements

Method	Description
document.getElementById(*id*)	Find an element by element id
document.getElementsByTagName(*name*)	Find elements by tag name
document.getElementsByClassName(*name*)	Find elements by class name

Changing HTML Elements:

Property	Description

element.innerHTML = *new html content*	Change the inner HTML of an element
element.attribute = new value	Change the attribute value of an HTML element
element.style.*property = new style*	Change the style of an HTML element
Method	**Description**
element.setAttribute*(attribute, value)*	Change the attribute value of an HTML element

Adding Events Handlers:

Method	**Description**
document.getElementById(*id*).onclick = function(){*code*}	Adding event handler code to an onclick event

JavaScript HTML DOM - Changing HTML:

The HTML DOM allows JavaScript to change the content of HTML elements.

Changing HTML Content

The easiest way to modify the content of an HTML element is by using the innerHTML property.

To change the content of an HTML element, use this syntax: *document.getElementById(id).innerHTML = new HTML* This example changes the content of a <p> element:

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

<h2>JavaScript can Change HTML</h2>

<p id="p1">Hello World!</p>

<script>
document.getElementById("p1").innerHTML = "New text!";
</script>

<p>The paragraph above was changed by a script.</p>

</body>
</html>
```

Fig. 2 (HTML Element)

Example explained:

- The HTML document above contains a <p> element with id="p1"
- We use the HTML DOM to get the element with id="p1"

- A JavaScript changes the content (innerHTML) of that element to "New text!"

This example changes the content of an <h1> element:

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

<h1 id="id01">Old Heading</h1>

<script>
const element = document.getElementById("id01");
element.innerHTML = "New Heading";
</script>

<p>JavaScript changed "Old Heading" to "New Heading".</p>

</body>
</html>
```

Fig. 3 (HTML DOM)

Example explained:

- The HTML document above contains an <h1> element with id="id01"
- We use the HTML DOM to get the element with id="id01"
- A JavaScript changes the content (innerHTML) of that element to "New Heading"

Changing the Value of an Attribute:

To change the value of an HTML attribute, use this syntax:

document.getElementById(id).attribute = new value

This example changes the value of the src attribute of an <img> element:

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

<img id="myImage" src="smiley.gif">

<script>
document.getElementById("myImage").src = "landscape.jpg";
</script>

</body>
</html>
```

Example explained:

- The HTML document above contains an <img> element with id="myImage"
- We use the HTML DOM to get the element with id="myImage"
- A JavaScript changes the src attribute of that element from "smiley.gif" to "landscape.jpg"

JavaScript HTML DOM - Changing CSS:

The HTML DOM allows JavaScript to change the style of HTML elements.

Changing HTML Style

To change the style of an HTML element, use this syntax:

document.getElementById(id).style.property = new style

The following example changes the style of a <p> element:

```
<html>
<body>

<p id="p2">Hello World!</p>

<script>
document.getElementById("p2").style.color = "blue";
</script>

</body>
</html>
```

Using Events

The HTML DOM allows you to execute code when an event occurs.

Events are generated by the browser when "things happen" to HTML elements:

- An element is clicked on
- The page has loaded
- Input fields are changed

Before Clicked:

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

<h1 id="id1">My Heading 1</h1>

<button type="button"
onclick="document.getElementById('id1').style.color = 'red'">
Click Me!</button>

</body>
</html>
```

Fig. 4 (On Click)

After Clicked:

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

<h1 id="id1">My Heading 1</h1>

<button type="button"
onclick="document.getElementById('id1').style.color = 'red'">
Click Me!</button>

</body>
</html>
```

Fig. 5 (On Click)