**DOM**

**Document Object Model**

"The W3C Document Object Model (DOM) is a platform and language-neutral interface that allows programs and scripts to dynamically access and update the content, structure, and style of a document.”

**HTML DOM**

The HTML DOM is a standard for how to get, change, add, or delete HTML elements. It can be accessed with JavaScript (and with other programming languages).

In the DOM, all HTML elements are defined as **OBJECTS.**

HTML DOM **methods** are **actions** (like add or deleting an HTML element) you can perform (on HTML Elements). HTML DOM **properties** are **values** (of HTML Elements) that you can set or change (like changing the content of an HTML element).

**How to access the HTML element?**

**By the use of DOM method we can access HTML element [obj].**

**getElementById Method: most common way to access an HTML element is to use the id of the element. innerHTML Property: innerHTML property is useful for getting or replacing the content of HTML elements.**

**<script>**

**let a=document.getElementById("demo").innerHTML;**

**document.getElementById("demo").innerHTML=a;**

**</script>**

**DOM Document:**

**DOM document object is the owner of all other objects in your web page.** **If you want to access any element in an HTML page, you always start with accessing the document object.**

**DOM Elements:**

With JavaScript, you want to manipulate HTML elements. To do so, you have to find the elements first. There are several ways to do this:

* Finding HTML elements by id
* Finding HTML elements by tag name
* Finding HTML elements by class name
* Finding HTML elements by CSS selectors
* Finding HTML elements by HTML object collections

**How to manipulate the HTML content and attributes?**

**For Content .innerHTML: it is the best way to access the content of the HTML.**

**document.getElementById(“demo”.innerHTML=’’hello’’;**

**For Attribute: document.getElementById(“demo”.src=’’home.jpg’’;**

**DOM Events:**

A JavaScript can be executed when an event occurs, like when a user clicks on an HTML element.

* When a user clicks the mouse
* When a web page has loaded
* When an image has been loaded
* When the mouse moves over an element
* onmouseover
* onmouseout
* onmouseup
* onmousedown
* onfocus
* When an input field is changed
* When an HTML form is submitted

**DOM Event listeners:**

**The addEventListener() method attaches an event handler to an element without overwriting existing event handlers. You can add many event handlers of same type and also different type to one element i.e. 2 click event. When using the addEventListener() method, the JavaScript is separated from the HTML markup, for better readability and allows you to add event listeners even when you do not control the HTML markup.**

**You can easily remove an event listener by using the removeEventListener() method.**

**The addEventListener() method makes it easier to control how the event reacts to bubbling and capturing.**