**Experiment 04**

**Aim:** Develop a web page illustrating JavaScript functions and events.

**Learning Objective:** Students should be able to develop a web page using basic JavaScript - HTML DOM Methods and handle events.

**Tools:** IDE(VS Code / Sublime Text3 / Notepad++), Web Browser.

**Theory:**

**JAVASCRIPT**

JavaScript is a lightweight, interpreted programming language. It is designed for creating network-centric applications. It is complementary to and integrated with Java. JavaScript is very easy to implement because it is integrated with HTML. It is open and cross-platform.

First JavaScript Code:

<!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 price 1 = 5;

const price2 = 6;

let total = price1 + price2;

document.getElementById("demo").innerHTML ="The total is: " + total; </script>

</body>

</html>

**HTML DOM (Document Object Model)**

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. **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 innerHTML property can be used to get or change any HTML element, including <html> and <body>.

**The getElementById Method**

The most common way to access an HTML element is to use the id of the element. In the example below the getElementById method used id="demo" to find the element. <!DOCTYPE html>

<html>

<body>

<h2>My First Page</h2>

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

<script>

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

</body>

</html>

**JavaScript Functions**

A JavaScript function is a block of code designed to perform a particular task. A JavaScript function is executed when "something" invokes it (calls it).

● When an event occurs (when a user clicks a button)

● When it is invoked (called) from JavaScript code

● Automatically (self invoked)

**Example :**

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Functions</h2>

<p>This example calls a function which performs a calculation, and returns the result:</p>

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

**<script>**

**function myFunction(p1, p2) {**

**return p1 \* p2;**

**}**

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

</body>

</html>

**Example :**

<!DOCTYPE html>

<html>

<body>

<h2>JavaScript Functions</h2>

<p>This example calls a function which performs a calculation and returns the result:</p>

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

**<script>**

**var x = myFunction(4, 3);**

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

**function myFunction(a, b) {**

**return a \* b;**

**}**

**</script>**

</body>

</html>

**JAVASCRIPT EVENT HANDLERS**

When an event, such as clicking an element or pressing a keyboard key, occurs on an HTML or DOM element, we can invoke certain functions based on these events. So, how do the HTML element knows when to execute the mentioned JavaScript function or JavaScript code? The event handlers handle this.

The event handlers are the properties of the HTML or DOM elements, which manages how the element should react to a specific event.

**Event Description**

onchange An HTML element has been changed

onclick The user clicks an HTML element

onmouseover The user moves the mouse over an HTML element onmouseout The user moves the mouse away from an HTML element onkeydown The user pushes a keyboard key

onload The browser has finished loading the page

Syntax :

<element event='some JavaScript'>

**1. OnClick Event Handler**

The onclick event invokes the Javascript code on clicking a button, link, or tag. Its syntax looks like below:

Syntax:

<htmlTag onclick="JavaScript Code need to be executed"></htmlTag>

Example:

<!DOCTYPE html>

<html>

<body>

<button

onclick="document.getElementById('demo').innerHTML=Date()">The time is?</button>

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

</body>

</html>

**2. OnLoad Event Handler**

The onLoad event executes when a page or a frame or an image is loaded. It initializes some variables when a page is loaded or sets some properties when an image is loaded. Its syntax looks like below:

Syntax when a page loads:

<body onload = "JavaScript code need to be executed"> When a frame loads:

<frameset onload="JavaScript code need to be executed"> Syntax when an image loads:

<img src="image need to be loaded" onload = "Javascript code that need to be executed">

Example:

<html>

<script>

function logInfo(info){

console.log(info+" onload event is triggered")

}

</script>

<body onload="logInfo('Page')"> Demonstrating onload based on DOM object</br>

<img src="../MyLogo.jpg" onload="logInfo('image')"/> </body>

</html>

**3. OnMouseOver Event Handler**

Just like the onclick event, the onMouseOver event triggers when the user moves a mouse over the particular web element. Its syntax looks like below:

Syntax:

<tag onmouseover=" JavaScript needs to be executed">

**4. OnMouseOut event handler**

Just like the onclick event, the onMouseOver event triggers when a user moves a mouse out of the particular web element. Its syntax looks like below:

Syntax:

<tag onmouseout=" JavaScript needs to be executed">

**5. OnKeyPress Event Handler**

This event triggers when a user presses any keyboard key. Its syntax looks like below: Syntax:

<tag onkeypress = "Javascript that needs to be executed"> **6. Onkeyup Event Handler**

You can use this Javascript event in a scenario where you press a keyboard event and it performs as per your code logic.

**7. Onchange Event Handler**

This event identifies the variance in the value of any element listing to this event. The best example of this is when text and dropdown list change events.

**Programs :**

**Program 1:** Demonstrate use of following Event Handlers in JavaScript

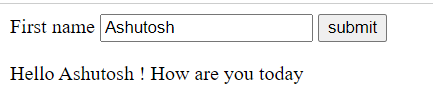
1. OnMouseOver and OnMouseOut

2. OnKeyPress and Onkeyup

**Program 2:**

Write a JavaScript program using functions to take First Name from the user and on click of submit button display the following message.





**Program 3:**

Write a JavaScript program using functions to take two numbers from the user and perform Addition and Subtraction.



A picture containing chart

Description automatically generated

A picture containing waterfall chart

Description automatically generated

**Result and discussion:** We successfully illustrated use of functions and events in JavaScript. **Learning Outcomes:** Students should have the ability to

LO1: To use basic JavaScript - HTML DOM Methods.

LO2: To handle different types of events using Javascript Event Handlers.

**Course Outcomes:** Upon completion of the course students will be able to understand how to use functions and handle event in JavaScript

**Conclusion: In this**

For Faculty Use

|  |  |  |
| --- | --- | --- |
| **Correction Parameters** | **Formative Assessment** | **Timely completion**  **Attendance /**  **of Practical**  **Learning**  **Attitude** |
| **Marks**  **Obtained** |  |