Training Day 12

Day 12– 8th July 2025

JavaScript Functions and Events

Detailed Description:

On Day 12, we explored **JavaScript functions and event handling**, which are crucial for writing organized, reusable, and interactive code.
The instructor explained that functions allow you to **group code into blocks** that can be called multiple times, while events enable webpages to **respond to user actions**.

🔹 1. Introduction to Functions

A **function** is a block of code designed to perform a particular task.
Functions make the code **modular and reusable**.

Syntax:

```
function functionName(parameters) {

  // code to be executed

}
```

Example practiced:

```
function greetUser(name) {

  alert("Hello, " + name + "!");
```

```
}

greetUser("Malika");
```

I learned that **functions can take parameters** (inputs) and **return values** if needed.

Example with return value:

```
function addNumbers(a, b) {

  return a + b;

}

let sum = addNumbers(5, 10);

console.log("Sum is: " + sum);
```

Functions help in reducing repetitive code and make programs easier to **maintain and debug**.

🔹 2. Introduction to Events

Events are actions or occurrences that happen in the browser, which JavaScript can respond to.
Common events include:

- onclick – User clicks an element.
- onmouseover – User hovers over an element.
- onmouseout – User moves the cursor away.
- onload – Runs code when the page loads.
- onchange – Triggers when the value of a form element changes.

Example practiced:

```
<button onclick="showMessage()">Click Me</button>



<script>

function showMessage() {

  alert("Button clicked!");

}

</script>
```

Here, clicking the button triggers the showMessage() function, demonstrating **interaction between HTML and JavaScript**.

- **3. Event Listeners**

The instructor introduced **event listeners**, which are a more flexible way to handle events:

```
let button = document.getElementById("myBtn");

button.addEventListener("click", function() {

  alert("Button clicked using Event Listener!");

});
```

I learned that event listeners allow **multiple functions to respond to the same event** and separate JavaScript from HTML for cleaner code.

🔹 4. Practical Exercises

We practiced:

- Creating **buttons that change background color** on click.
- Displaying **alerts or messages** when the user hovers over an element.
- Using **functions with parameters** to update content dynamically.
- Implementing **form validation** using events and functions.

Example: Changing text dynamically

```
<p id="demo">Original Text</p>

<button onclick="changeText()">Change Text</button>



<script>

function changeText() {

  document.getElementById("demo").innerHTML = "Text Updated!";

}

</script>
```

Through these exercises, I saw how functions and events **bring interactivity and logic together** on a webpage.

Learning Outcomes:

- Learned the concept of **functions** and their importance in modular coding.
- Practiced creating functions with **parameters and return values**.
- Understood **event handling** and how webpages can respond to user actions.
- Learned to use **event listeners** for cleaner and more flexible JavaScript.
- Applied functions and events in small practical projects like dynamic text updates, button actions, and form validations.