Class No: 07

JavaScript Events & ES6 Features

JavaScript Events

JavaScript events allow interaction with users when they perform actions such as clicking, hovering, or pressing keys. Events make web pages dynamic and interactive.

Common JavaScript Events:

- onclick - Triggers when an element is clicked.
- onmouseover - Triggers when the mouse hovers over an element.
- onkeydown - Triggers when a key is pressed.

Handling Events in JavaScript

1. Using HTML Attribute (Not Recommended):

```
<button onclick="alert('Button Clicked!')">Click Me</button>
```

This method is not recommended because it makes the HTML code less readable and harder to manage.

2. Using JavaScript Event Listener (Recommended):

```
<button id="myButton">Click Me</button><script>
```

Js:

```
let btn = document.getElementById("myButton");
btn.addEventListener("click", function() {
alert("Button Clicked!");    });</script>
```

This method is preferred as it keeps the JavaScript code separate from HTML, making it easier to manage and maintain.

ES6 Features (Modern JavaScript)

1. Let & Const (Block-scoped variables):

- let allows reassignment.
- const does not allow reassignment.

```
let x = 10;
x = 15; // Allowed

const y = 20;
y = 25; // Error! Cannot reassign a constant variable.
```

2. Template Literals (String Interpolation):

Allows embedding variables directly into strings using backticks (`).

```
let name = "John";
console.log(`Hello, ${name}!`); // Output: Hello, John!
```

3. Spread & Rest Operators:

- Spread (...): Expands arrays or objects.
- Rest (...): Collects multiple values into an array.

```
let arr = [1, 2, 3];
let newArr = [...arr, 4, 5];
console.log(newArr); // Output: [1, 2, 3, 4, 5]
```

Rest Operator Example:

```
function showNames(...names) {
  console.log(names);
}
showNames("Alice", "Bob", "Charlie");               // Output: ["Alice", "Bob", "Charlie"]
```

4. Destructuring Assignment:

Extract values from arrays or objects easily.

```
let person = { name: "Alice", age: 25 };
let { name, age } = person;
console.log(name, age);          // Output: Alice 25
```

HTML & JavaScript Example: Interactive Button

```
<body>
  <button id="myButton">Click Me</button>
  <p id="message"></p>

  <script>
    let btn = document.getElementById("myButton");
    let message = document.getElementById("message");

    btn.addEventListener("click", function() {
      message.textContent = "Button was clicked!";
    });
  </script>
</body>
```

2: Display List of Names

```
<body>
  <button id="showNames">Show Names</button>
  <ul id="nameList"></ul>
  <script>
    function displayNames(...names) {
      let list = document.getElementById("nameList");
      list.innerHTML = "";
      names.forEach(name => {
        let li = document.createElement("li");
        li.textContent = name;
        list.appendChild(li);
      });
    }
    document.getElementById("showNames").addEventListener("click", function() {
      displayNames("Alice", "Bob", "Charlie");
    });
  </script>
</body>
```