

# Training Day 13

Day 13– 9th July 2025

## DOM Manipulation and Dynamic Content

---

### Detailed Description:

On Day 13, we explored the **Document Object Model (DOM)**, which represents a webpage as a **tree of objects**.

The instructor explained that **DOM manipulation** allows JavaScript to dynamically change the content, style, and structure of a webpage **without reloading it**.

---

### ◆ 1. Introduction to DOM

The DOM is a **hierarchical representation** of all HTML elements as objects.

Key points:

- Each HTML element is a **node** in the DOM tree.
- JavaScript can **select, modify, delete, or create elements** dynamically.
- DOM methods allow interaction with HTML elements efficiently.

### Example of DOM structure:

```
<html>
```

```
<body>
```

```
<h1>Heading</h1>
```

```
<p>Paragraph</p>

</body>

</html>
```

- <html> is the root node, <body> is a child, and <h1> and <p> are children of <body>.

---

## ◆ 2. Selecting Elements

We learned different ways to **select elements** in JavaScript:

### 1. By ID

```
let element = document.getElementById("demo");
```

### 2. By Class Name

```
let elements = document.getElementsByClassName("info");
```

### 3. By Tag Name

```
let paragraphs = document.getElementsByTagName("p");
```

### 4. Query Selector (CSS style)

```
let element = document.querySelector(".info"); // first match
```

```
let allElements = document.querySelectorAll(".info"); // all matches
```

Selecting elements is the **first step** before manipulating them dynamically.

---

## ◆ 3. Modifying Content and Styles

By : Malika

URN : 2302600

CRN :2315145

Once elements are selected, we can **change their content or style**:

**Changing text content:**

```
document.getElementById("demo").innerHTML = "New Content Added!";
```

**Changing styles dynamically:**

```
document.getElementById("demo").style.color = "blue";
```

```
document.getElementById("demo").style.fontSize = "20px";
```

These methods allow real-time updates to webpages, enhancing interactivity.

---

◆ **4. Creating and Removing Elements**

We practiced **adding new elements** and **removing existing ones**:

**Creating a new element:**

```
let newPara = document.createElement("p");
```

```
newPara.innerHTML = "This is a dynamically created paragraph.";
```

```
document.body.appendChild(newPara);
```

**Removing an element:**

```
let element = document.getElementById("removeMe");
```

```
element.parentNode.removeChild(element);
```

This taught me how to **modify the structure of a webpage dynamically**.

---

## ◆ 5. Event-Driven DOM Manipulation

We combined **events with DOM manipulation**:

**Example: Change text on button click**

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

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

```
<script>
```

```
function changeText() {
```

```
    document.getElementById("text").innerHTML = "Text Updated Dynamically!";
```

```
}
```

```
</script>
```

This exercise highlighted the **power of JavaScript in creating interactive user experiences**.

---

### **Learning Outcomes:**

- Understood the **DOM structure** and its importance in web development.
- Learned to **select HTML elements** using various methods (ID, class, tag, querySelector).
- Practiced **modifying content and styles dynamically**.
- Learned to **create and remove elements** programmatically.
- Applied **event-driven DOM manipulation** to make pages interactive.

- Developed skills to build **dynamic and responsive web pages**.