**DOM Manipulation in JavaScript**

The **DOM (Document Object Model)** allows JavaScript to **interact with and modify HTML and CSS** dynamically.  
It’s how JavaScript makes web pages **interactive** — for example, showing/hiding content, changing styles, or reacting to user actions (clicks, input, etc.).

**1. What is the DOM?**

When a web page loads, the browser creates a **Document Object Model** — a tree-like structure representing all HTML elements.

JavaScript can use the DOM to **access, modify, add, or delete HTML elements** dynamically.

#### Example HTML:

<h1 id="title">Hello</h1>

<p class="msg">Welcome to JavaScript</p>

The DOM looks like this:

Document

└── html

└── body

├── h1#title

└── p.msg

**2. Accessing Elements**

JavaScript can **select elements** using various methods:

#### By ID:

let heading = document.getElementById("title");

console.log(heading.innerText); // Output: Hello

#### By Class:

let messages = document.getElementsByClassName("msg");

console.log(messages[0].innerText);

#### By Tag Name:

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

**Using Modern Query Selectors:**

document.querySelector("#title"); // Select by ID

document.querySelector(".msg"); // Select by class

document.querySelectorAll("p"); // Select all <p> elements

**3. Changing Content**

You can change the text or HTML inside an element.

|  |  |  |
| --- | --- | --- |
| **Property** | **Description** | **Example** |
| textContent | Changes text only | heading.textContent = "Hi Raj!"; |
| innerHTML | Changes HTML inside | para.innerHTML = "<b>Welcome!</b>"; |
| innerText | Changes visible text | para.innerText = "Visible only text"; |

#### Example:

document.getElementById("title").textContent = "Welcome to JavaScript!";

document.getElementById("title").innerText = "Welcome Raj!";

document.querySelector(".msg").innerHTML = "<b>JavaScript is fun!</b>";

**4. Changing CSS Styles**

You can modify CSS properties directly using .style.

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Example** |
| .style.property | Change CSS directly | heading.style.color = "blue"; |
| .classList.add() | Add a CSS class | heading.classList.add("highlight"); |
| .classList.remove() | Remove a class | heading.classList.remove("highlight"); |
| .classList.toggle() | Add/remove class dynamically | heading.classList.toggle("active"); |

#### Example:

let title = document.getElementById("title");

title.style.color = "blue";

title.style.fontSize = "30px";

title.style.textAlign = "center";

**5. Creating & Adding Elements**

You can create new elements and insert them into the DOM.

|  |  |  |
| --- | --- | --- |
| **Method** | **Description** | **Example** |
| createElement() | Create new element | let div = document.createElement("div"); |
| appendChild() | Add as last child | document.body.appendChild(div); |
| prepend() | Add as first child | document.body.prepend(div); |
| remove() | Delete element | element.remove(); |

#### Example:

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

newPara.innerText = "This is a new paragraph.";

document.body.appendChild(newPara); // Adds to end of body

Or add before/after a specific element:

let div = document.querySelector("div");

div.appendChild(newPara); // inside div

**6. Removing Elements**

let para = document.querySelector(".msg");

para.remove();

**7. Event Handling**

Make elements **react** when users interact (click, hover, type, etc.).

#### Example:

<button id="btn">Click Me</button>

<p id="output"></p>

document.getElementById("btn").addEventListener("click", function() {

document.getElementById("output").innerText = "Button was clicked!";

});

**8. Common Events**

|  |  |
| --- | --- |
| **Event** | **Description** |
| click | When element is clicked |
| mouseover | When mouse hovers |
| mouseout | When mouse leaves |
| keydown | When key is pressed |
| submit | When form is submitted |

**Summary:**

|  |  |
| --- | --- |
| **Action** | **Method / Property** |
| Select element | getElementById, querySelector |
| Change text | .innerText, .innerHTML |
| Change style | .style.propertyName |
| Add element | createElement, appendChild |
| Remove element | .remove() |
| Handle events | addEventListener() |

### 🧩 ****Mini Assignment (Practice)****

Create a small webpage that:

1. Displays a heading and a paragraph.
2. Has a button labeled **“Change Content”**.
3. When you click the button:
   * The heading text changes to **“Welcome to JavaScript DOM!”**
   * The paragraph text changes to **“You just updated the content using JavaScript.”**
   * The text color of the paragraph changes to **blue**.
   * A new paragraph is added at the bottom saying **“New element added!”**

### 💡 ****Hint:****

You’ll need to use:

* document.getElementById()
* .textContent
* .style.color
* createElement()
* appendChild()