DOM Manipulation & Events 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();
```

Events in JavaScript

Events in JavaScript are **actions or occurrences** that happen in the browser — like when a user clicks a button, types in a field, or moves the mouse.
JavaScript can **listen** for these events and **respond** to them (this is called *event handling*).

1. What is an Event?

An **event** is something that happens to an element.
For example:

- Clicking a button
- Pressing a key
- Submitting a form
- Loading a page
- Hovering over text

---

2. Event Handling Methods

There are **three main ways** to handle events in JavaScript:

a. Inline Event Handling

(Directly in HTML)

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

Or with a function:

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

<script>
  function showMessage() {
    alert("Hello Raj!");
  }
</script>
```

---

b. DOM Property Method

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

<script>
  let btn = document.getElementById("btn");
  btn.onclick = function() {
    alert("Button clicked!");
  };
</script>
```

---

c. Using `addEventListener()` (Best Practice)

This is the modern and recommended method.

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

<script>
  let btn = document.getElementById("btn");
  btn.addEventListener("click", function() {
    alert("Hello Raj, you clicked the button!");
  });
</script>
```

You can also define the function separately:

```
function greet() {
  alert("Welcome!");
}
btn.addEventListener("click", greet);
```

---

3. Common Event Types

Event Name	Description	Example
`click`	When an element is clicked	Button, link
`dblclick`	Double click	Image, button
`mouseover`	Mouse pointer moves over element	Menu hover
`mouseout`	Mouse pointer leaves element	Tooltip hide
`mousedown`	Mouse button pressed down	Drawing apps
`mouseup`	Mouse button released	Drawing apps
`keydown`	Key is pressed	Input fields
`keyup`	Key is released	Form validation
`submit`	Form is submitted	Login form
`change`	Input value changes	Dropdown
`focus`	Element is focused	Input box
`blur`	Element loses focus	Validation
`load`	Page finishes loading	Window, image
`scroll`	User scrolls	Animation triggers

---

4. Example: Changing Text on Click

```
<h2 id="title">Hello Raj!</h2>
<button id="btn">Change Text</button>

<script>
  document.getElementById("btn").addEventListener("click", function()
{
    document.getElementById("title").innerText = "You clicked the
button!";
  });
</script>
```

---

5. Example: Mouse Events

```
<p id="para">Hover over this text!</p>

<script>
  let p = document.getElementById("para");

  p.addEventListener("mouseover", () => {
    p.style.color = "red";
  });

  p.addEventListener("mouseout", () => {
    p.style.color = "black";
  });
</script>
```

---

6. Example: Keyboard Event

```
<input type="text" id="name" placeholder="Type your name">
<p id="output"></p>

<script>
  document.getElementById("name").addEventListener("keyup", function()
{
    document.getElementById("output").innerText = this.value;
  });
</script>
```

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)

DOM Manipulation:

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!"**

Events:

1. Create a button that changes the background color when clicked.
2. Create a paragraph that changes text color on hover.
3. Create an input box that displays what the user types in real-time.
4. Create a button that shows an alert with your name on double-click.