16 – Document Object Model

The Document Object Model (DOM) is the tree-like representation of the objects that comprise the structure and content of a document on the web, where every element is represented as a “node”. It represents all the HTML in your page as JavaScript objects, allowing read, update, add, and remove operations on parts of the web page without reloading it.

Example:

Document

└── html

├── head

└── body

├── h1#title

└── button#changeTextBtn

The DOM is not a part of JavaScript nor HTML. It is a Web API used to build websites. Without it, JavaScript wouldn’t have any model of the HTML document and its components.

In simple terms, the structure of the page (written in HTML) is represented as the DOM through a Web API, which allows its manipulation through a scripting language such as JavaScript.

**Important DOM Manipulation Concepts**

1) DOM Selection

const title = document.getElementById("title"); // Select by ID

const firstButton = document.querySelector("button"); // Select by CSS selector

const allButtons = document.querySelectorAll("button"); // Select all matching elements

2) Changing Content

element.textContent = "New content";

element.innerHTML = "<em>Emphasized text</em>";

3) Changing Styles

element.style.backgroundColor = "yellow";

4) Class Manipulation

element.classList.add("visible");

element.classList.remove("hidden");

element.classList.toggle("dark-mode");

5) Creating and Inserting Elements

const newDiv = document.createElement("div");

document.body.appendChild(newDiv);

OR

element.insertBefore(newChild, referenceChild);

element.append(); // Modern alternative to appendChild

element.prepend(); // Insert at the beginning

6) Removing Elements

element.remove(); // Modern way

OR

element.parentNode.removeChild(element); // Older way

7) Event Handling (Detailed Discussion in Prev. Article)

button.addEventListener("click", () => {

alert("Button clicked!");

});

8) Attribute Manipulation

element.setAttribute("src", "image.jpg"); // set value

const value = element.getAttribute("href"); // get value

element.removeAttribute("disabled");