JavaScript DOM - Beginner Friendly Notes

# 🌳 What is the DOM?

Think of the DOM as a tree-like structure that represents the entire content and structure of your web page. It allows JavaScript to interact with HTML and CSS dynamically.

Analogy: Imagine your HTML page is like a house blueprint. The DOM is a live model of the house—and JavaScript is the person who can walk through the house, change furniture (text), paint the walls (styles), and even add or remove rooms (elements).

# 🧱 1. Understanding the Structure of the DOM

HTML Example:

<!DOCTYPE html>  
<html>  
 <head>  
 <title>My Page</title>  
 </head>  
 <body>  
 <h1 id="main-title">Welcome!</h1>  
 <p class="description">This is a simple page.</p>  
 <button>Click me</button>  
 </body>  
</html>

DOM Tree (Simplified):

document  
└── html  
 ├── head  
 │ └── title  
 └── body  
 ├── h1#main-title  
 ├── p.description  
 └── button

# 🧠 2. Accessing DOM Elements in JavaScript

JavaScript gives us methods to select elements from the page:

document.getElementById("main-title");  
document.getElementsByClassName("description");  
document.getElementsByTagName("p");  
document.querySelector("p.description");  
document.querySelectorAll("p");

# 🔄 3. Changing Content

Change Text:

let heading = document.getElementById("main-title");  
heading.textContent = "Hello World!";

Change HTML inside an element:

heading.innerHTML = "<em>Hello World!</em>";

🔸 textContent – Sets text only  
🔸 innerHTML – Parses and renders HTML inside

# 🎨 4. Changing Styles

heading.style.color = "blue";  
heading.style.fontSize = "30px";

# 🧩 5. Adding or Removing Elements

Create an element:

let newPara = document.createElement("p");  
newPara.textContent = "I was added with JS!";  
document.body.appendChild(newPara);

Remove an element:

let button = document.querySelector("button");  
button.remove();

# 🎯 6. Event Listeners (Interactivity)

HTML:

<button id="btn">Click me!</button>

JavaScript:

let btn = document.getElementById("btn");  
  
btn.addEventListener("click", function () {  
 alert("Button clicked!");  
});

# 🧭 7. DOM Navigation

let para = document.querySelector("p");  
let parent = para.parentElement;  
let children = document.body.children;  
let next = para.nextElementSibling;  
let prev = para.previousElementSibling;

# 📦 8. Attributes and Classes

Get/Set attributes:

let img = document.querySelector("img");  
img.getAttribute("src");  
img.setAttribute("alt", "Cool image");

Working with classes:

let div = document.querySelector("div");  
div.classList.add("new-style");  
div.classList.remove("old-style");  
div.classList.toggle("highlight");

# 🛠️ 9. Practice Example

HTML:

<h1 id="greet">Hello there!</h1>  
<button id="changeBtn">Change Text & Color</button>

JavaScript:

let btn = document.getElementById("changeBtn");  
let h1 = document.getElementById("greet");  
  
btn.addEventListener("click", function () {  
 h1.textContent = "Text changed!";  
 document.body.style.backgroundColor = "lightblue";  
});

# 📚 10. Real World Use Case

Form Validation:

HTML:

<input type="text" id="nameInput" />  
<button id="submitBtn">Submit</button>  
<p id="message"></p>

JavaScript:

let input = document.getElementById("nameInput");  
let btn = document.getElementById("submitBtn");  
let msg = document.getElementById("message");  
  
btn.addEventListener("click", function () {  
 if (input.value.trim() === "") {  
 msg.textContent = "Name cannot be empty!";  
 msg.style.color = "red";  
 } else {  
 msg.textContent = `Hello, ${input.value}!`;  
 msg.style.color = "green";  
 }  
});

# 🧪 11. DOM Projects to Practice

* To-Do List App
* Color Theme Changer
* Image Slider
* Accordion FAQs
* Live Character Counter

# ✅ Recap Summary

Get element by ID: document.getElementById("id")

Change text: element.textContent = "new text"

Change HTML: element.innerHTML = "<b>bold</b>"

Change style: element.style.color = "red"

Add/remove elements: createElement, appendChild, remove()

Events: addEventListener("click", fn)

Class manipulation: classList.add("new-class")

Attribute manipulation: getAttribute, setAttribute

Navigation: parentElement, nextElementSibling