JavaScript DOM (Document Object Model) - Detailed Notes

1. Introduction to the DOM

The DOM (Document Object Model) is a programming interface provided by browsers that allows JavaScript to interact with and manipulate HTML and XML documents.

It represents the page so that programs can change the document structure, style, and content.

Key concepts:

- The DOM treats an HTML or XML document as a tree structure where each node is an object representing a part of the document.

- DOM is not part of JavaScript; it's provided by the browser.

2. DOM Tree Structure

The DOM represents an HTML document as a tree of nodes.
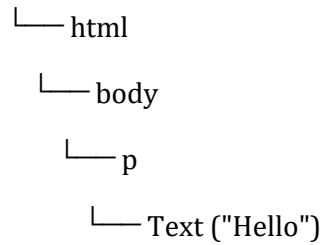
Important types of nodes:

- Document: The root of the document.

- Element: Represents HTML elements (e.g., <p>, <div>).

- Text: Represents the text inside elements.

- Attribute: Represents attributes of HTML elements.

Example:

<html>

 <body>

  <p>Hello</p>

 </body>

</html>

Tree:

```
Document
 └── html
      └── body
           └── p
                └── Text ("Hello")
```

3. Accessing Elements

You can access HTML elements using JavaScript with methods such as:

- getElementById()

- getElementsByClassName()

- getElementsByTagName()

- querySelector()

- querySelectorAll()

Example:

document.getElementById("myId");

document.querySelector(".myClass");

4. DOM Manipulation

Once you have accessed elements, you can change their content or style.

- innerHTML: Changes the HTML inside an element.

- textContent: Changes the text.

- style.property: Changes CSS styles.

Example:

let para = document.getElementById("myPara");

para.innerHTML = "Updated content";

para.style.color = "red";

5. Creating and Removing Elements

You can dynamically add or remove elements from the DOM.

- createElement(): Creates a new element.

- appendChild(): Adds the element to the DOM.

- removeChild(): Removes an element.

Example:

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

newDiv.textContent = "Hello";

document.body.appendChild(newDiv);

6. Event Handling

DOM allows you to make pages interactive using events.

- addEventListener(): Attach an event.

- removeEventListener(): Remove an event.

Example:

button.addEventListener("click", function() {

 alert("Button clicked!");

});

7. DOM Properties and Methods

Common properties and methods:

- innerHTML, textContent, className, id, value

- setAttribute(), getAttribute(), removeAttribute()

- parentNode, childNodes, firstChild, lastChild

Example:

```
let input = document.getElementById("myInput");

input.setAttribute("placeholder", "Enter name");
```

8. DOM Collections and NodeList

When selecting multiple elements, you may get a collection:

- HTMLCollection: Live collection from getElementsBy*

- NodeList: Static or live list from querySelectorAll()

You can loop through these using for or forEach:

Example:

```
let items = document.querySelectorAll("li");

items.forEach(item => console.log(item.textContent));
```