

JS - Document Object Model (DOM)

1. What is the DOM?

The Document Object Model (DOM) represents the structure of a webpage as a tree of objects. It allows JavaScript to interact with the content, structure, and styles of HTML documents dynamically. Using the DOM, you can manipulate HTML elements, change attributes, and respond to events.

2. Accessing Elements

There are several ways to access HTML elements in the DOM:

```
// 1. Accessing by ID
let elementById = document.getElementById("myElement");

// 2. Accessing by Class Name
let elementsByClass = document.getElementsByClassName("myClass");

// 3. Accessing by Tag Name
let elementsByTag = document.getElementsByTagName("p");

// 4. Accessing by Query Selector (CSS Selector)
let elementBySelector = document.querySelector(".myClass");
let elementsBySelector = document.querySelectorAll("p");
```

Example - Accessing Elements:

Content has been changed!

This is a paragraph with the class "myClass".

Another paragraph with the same class.

3. Manipulating Elements

You can manipulate the content, attributes, and styles of DOM elements using JavaScript:

```
// 1. Change innerHTML
elementById.innerHTML = "Updated content!";

// 2. Change element style
elementById.style.color = "blue";

// 3. Change element attributes
elementById.setAttribute("class", "newClass");
```

4. Creating and Removing Elements

You can dynamically create new elements and remove existing ones from the DOM:

```
// 1. Create a new element
let newElement = document.createElement("div");
newElement.innerHTML = "This is a new dynamically created element.";
```



```
// 2. Append the new element to the body
document.body.appendChild(newElement);

// 3. Remove an element from the DOM
document.body.removeChild(newElement);
```

5. Event Handling

The DOM allows you to handle events such as clicks, mouse movements, or keyboard presses:

```
// Adding an event listener
elementById.addEventListener("click", function() {
    alert("Element clicked!");
});
```

6. DOM Traversal

You can navigate through the DOM using various methods to access parent, child, and sibling elements:

```
// 1. Parent Node
let parentElement = elementById.parentNode;

// 2. Child Node
let firstChild = elementById.firstChild;

// 3. Sibling Node
let nextSibling = elementById.nextSibling;
```

7. Summary of DOM Manipulation

Action	Method/Property
Access element by ID	document.getElementById("id")
Access by class	document.getElementsByClassName("class")
Access by tag	document.getElementsByTagName("tag")
Access by selector	document.querySelector("selector")
Change content	element.innerHTML = "new content"
Change style	element.style.property = "value"
Change attribute	element.setAttribute("attribute", "value")
Create element	document.createElement("tag")
Append element	parentElement.appendChild(newElement)
Remove element	parentElement.removeChild(element)
Event handling	element.addEventListener("event", function() {})
Traversing DOM	parentElement.parentNode, element.firstChild, element.nextSibling

This is a new element!

