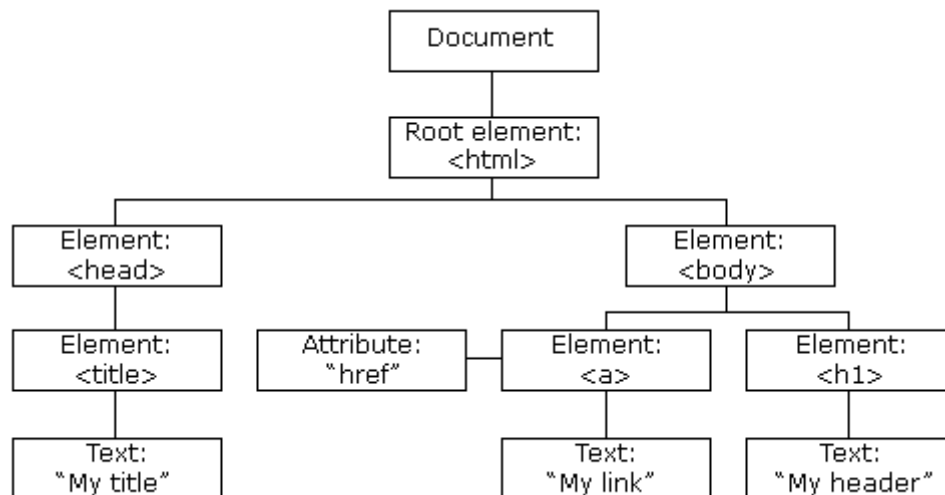


JavaScript DOM (Document Object Model) and DOM Methods

What is the DOM?

The **Document Object Model (DOM)** is a programming interface for web documents. It represents the structure of a document (such as an HTML or XML document) as a tree of objects, where each object corresponds to a part of the document. This allows programming languages like JavaScript to manipulate the structure, style, and content of web pages dynamically.



Structure of the DOM

- **Document:** The root of the DOM tree, representing the entire HTML or XML document.
- **Element:** Nodes representing HTML elements (e.g., <div>, <p>, <a>).
- **Attribute:** Nodes representing attributes of elements (e.g., class, id, href).
- **Text:** Nodes representing the actual text content within elements.

Common DOM Methods

1. Selecting Elements

- **document.getElementById(id):** Selects a single element by its id.

```
var element = document.getElementById("myId");
```
- **document.getElementsByClassName(className):** Selects all elements with a given class name.

```
var elements = document.getElementsByClassName("myClass");
```
- **document.getElementsByTagName(tagName):** Selects all elements with a given tag name.

```
var elements = document.getElementsByTagName("div");
```

- **document.querySelector(selector)**: Selects the first element that matches a CSS selector.

```
var element = document.querySelector(".myClass");
```

- **document.querySelectorAll(selector)**: Selects all elements that match a CSS selector.

```
var elements = document.querySelectorAll(".myClass");
```

2. Manipulating Elements

- **element.innerHTML**: Gets or sets the HTML content of an element.

```
element.innerHTML = "<p>New Content</p>";
```

- **element.textContent**: Gets or sets the text content of an element.

```
element.textContent = "New Text";
```

- **element.setAttribute(name, value)**: Sets the value of an attribute on an element.

```
element.setAttribute("class", "newClass");
```

- **element.getAttribute(name)**: Gets the value of an attribute on an element.

```
var className = element.getAttribute("class");
```

- **element.removeAttribute(name)**: Removes an attribute from an element.

```
element.removeAttribute("class");
```

- **element.classList.add(className)**: Adds a class to an element.

```
element.classList.add("newClass");
```

- **element.classList.remove(className)**: Removes a class from an element.

```
element.classList.remove("oldClass");
```

- **element.classList.toggle(className)**: Toggles a class on an element.

```
element.classList.toggle("active");
```

3. Creating and Inserting Elements

- **document.createElement(tagName)**: Creates a new element with the specified tag name.

```
var newDiv = document.createElement("div");
```

- **element.appendChild(child)**: Adds a child element to the end of a parent element.

```
var parent = document.getElementById("parent");
var child = document.createElement("div");
parent.appendChild(child);
```

- **element.insertBefore(newElement, referenceElement)**: Inserts a new element before a reference element.

```
parent.insertBefore(newDiv, referenceChild);
```

4. Removing Elements

- **element.removeChild(child)**: Removes a child element from a parent element.

```
parent.removeChild(child);
```

- **element.remove()**: Removes the element from the DOM.

```
element.remove();
```

5. Event Handling

- **element.addEventListener(event, function)**: Attaches an event handler to an element.

```
element.addEventListener("click", function() {
    alert("Element clicked!");
});
```

- **element.removeEventListener(event, function)**: Removes an event handler from an element.

```
element.removeEventListener("click", myFunction);
```

Example: Modifying the DOM

```
html
Copy code
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>DOM Example</title>
</head>
```

```
<body>
  <div id="container">
    <h1 id="title">Hello World</h1>
  </div>

  <script>
    // Select the title element
    var title = document.getElementById("title");

    // Change the text content of the title
    title.textContent = "Welcome to the DOM!";

    // Create a new paragraph element
    var newParagraph = document.createElement("p");

    // Add text to the new paragraph
    newParagraph.textContent = "This is a new paragraph.";

    // Append the new paragraph to the container
    var container = document.getElementById("container");
    container.appendChild(newParagraph);
  </script>
</body>
</html>
```

In this example:

- The `getElementById()` method is used to select an element by its ID.
- The `textContent` property changes the content of an element.
- The `createElement()` method creates a new element.
- The `appendChild()` method adds a new element to the DOM.