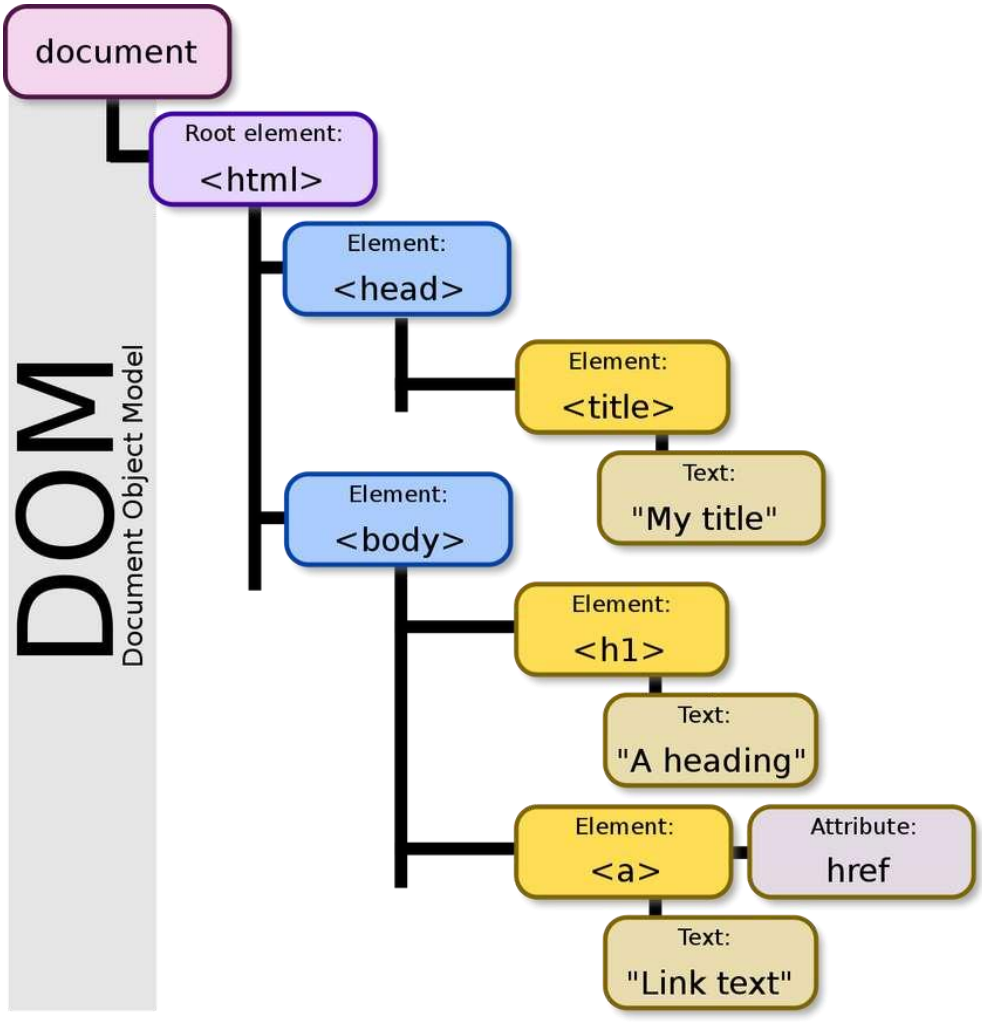
DOM IN JS

DocumentObject Model

The **Document Object Model (DOM)** is a programming interface for web documents. It represents the page so that JavaScript can manipulate it dynamically. The DOM provides a structured representation of the document as a **tree of objects**.

Think of the DOM as a tree-like structure that represents the elements of a web page. It provides a structured way to access, modify, and interact with the content and structure of HTML documents.



Application of DOM

- Imagine you have a web page with various elements like headings, paragraphs, buttons, and images. The DOM represents each of these elements as nodes in a tree structure.

- You can use JavaScript to traverse this tree, find specific elements, and change their content, styles, or even add new elements dynamically.

Method to Select a HTML Elements

Selecting by ID (getElementById)

```
<p id="title">Hello World</p>
<script>
    let element = document.getElementById("title");
    console.log(element); // Prints: <p id="title">Hello World</p>
</script>
```

Selecting by Class (getElementsByClassName)

```
<p class="text">Paragraph 1</p>
<p class="text">Paragraph 2</p>

<script>
    let elements = document.getElementsByClassName("text");
    console.log(elements); // Returns a collection of elements
</script>
```

Selecting by Tag Name (getElementsByTagName)
Selects elements based on the tag (like <p>, , <button>).

```
<p>Paragraph 1</p>
<p>Paragraph 2</p>

<script>
    let paragraphs = document.getElementsByTagName("p");
    console.log(paragraphs); // Returns all <p> elements
</script>
```

Selecting by Query (querySelector and querySelectorAll)

- querySelector selects **the first matching** element.
- querySelectorAll selects **all matching** elements.

```
<p class="example">First Paragraph</p>
<p class="example">Second Paragraph</p>

<script>
    let firstPara = document.querySelector(".example");
    let allParas = document.querySelectorAll(".example");

    console.log(firstPara); // First matching element
    console.log(allParas);  // All elements with class 'example'
</script>
```

Modifying Elements in the DOM

innerHTML:

```
<p id="demo">Old Content</p>
<button onclick="changeContent()">Click Me</button>

<script>
    function changeContent() {
        document.getElementById("demo").innerHTML = "<b>New Content</b>";
    }
</script>
```

Modifying Attributes (setAttribute, removeAttribute)

```
<img id="image" src="old.jpg">
<button onclick="changeImage()">Change Image</button>

<script>
    function changeImage() {
        document.getElementById("image").setAttribute("src", "new.jpg"
    }
</script>
```

Modifying Styles using JavaScript

Changing Individual Style Properties

```
<p id="text">Style Me!</p>
<button onclick="changeStyle()">Change Style</button>

<script>
    function changeStyle() {
        document.getElementById("text").style.color = "red";
    }
</script>
```

Adding Elements to the DOM

- **document.createElement()** → Creates a new element.
- **appendChild()** → Adds an element at the end of a parent.
- **append()** → Similar to appendChild() but allows adding text also.
- **insertBefore()** → Adds an element before another element.
- **innerHTML** → Injects new content inside an element

```
<div id="container">
    <p>Existing Paragraph</p>
</div>
<button onclick="addElement()">Add New Element</button>

<script>
    function addElement() {
        let newPara = document.createElement("p"); // Create <p> element
        newPara.textContent = "I am a new paragraph!"; // Add text

        document.getElementById("container").appendChild(newPara); // App
    }
</script>
```

Removing Elements from the DOM

- **`remove()`** → Removes an element directly.
- **`removeChild()`** → Removes a child element from a parent.
- **`innerHTML = ""`** → Removes all content inside an element.

```
<p id="text">I will be removed</p>
<button onclick="removeElement()">Remove</button>

<script>
    function removeElement() {
        let element = document.getElementById("text");
        element.remove(); // Removes itself
    }
</script>
```

Actions & Event Listeners in JavaScript

Adding Click Events (addEventListener)

```
<button id="myButton">Click Me</button>

<script>
    document.getElementById("myButton").addEventListener("click", function() {
        alert("Button Clicked!");
    });
</script>
```

Handling Forms and Input Fields

Getting Input Field Value

```
<input type="text" id="username" placeholder="Enter Name">
<button onclick="getValue()">Get Value</button>

<script>
    function getValue() {
        let inputValue = document.getElementById("username").value;
        alert("You entered: " + inputValue);
    }
</script>
```

Setting Input Field Value

```
<input type="text" id="userInput">
<button onclick="setValue()">Set Value</button>

<script>
    function setValue() {
        document.getElementById("userInput").value = "Hello World!";
    }
</script>
```