

dom

- The DOM allows programming languages like JavaScript to access, modify, and manipulate the document's structure and content dynamically.

2. DOM Tree Structure

- **Node Types:**
 - **Element nodes:** Represent HTML tags (e.g., `<div>`, `<p>`).
 - **Text nodes:** Contain the actual text content inside elements.
 - **Attribute nodes:** Represent the attributes of HTML elements (e.g., `class`, `id`).
 - **Comment nodes:** Represent comments in HTML (`<!-- Comment -->`).
- Explain the hierarchical structure: Root (`document`), parent, child, sibling, and leaf nodes.

3. Accessing DOM Elements

- **Selecting Elements:**
 - `getElementById()` : Selects an element by its ID.
 - `getElementsByClassName()` : Selects elements by class name.
 - `getElementsByTagName()` : Selects elements by tag name.
 - `querySelector()` : Selects the first matching element using a CSS selector.
 - `querySelectorAll()` : Selects all matching elements using a CSS selector.
- **Examples:**

```
const element = document.getElementById('myId');  
const items = document.querySelectorAll('.item');
```

4. Manipulating DOM Elements

- **Changing Content:**

- `innerHTML` : Changes the HTML content inside an element.
- `textContent` : Changes the text content inside an element.

```
element.innerHTML = '<p>New Content</p>';  
element.textContent = 'Updated Text';
```

- **Modifying Attributes:**

- `setAttribute()` : Sets the value of an attribute.
- `getAttribute()` : Gets the value of an attribute.
- `removeAttribute()` : Removes an attribute.

```
javascript  
Copy code  
element.setAttribute('class', 'new-class');  
const id = element.getAttribute('id');  
element.removeAttribute('title');
```

- **Modifying Styles:**

- `style` : Directly modify the `style` property of an element.

```
element.style.color = 'red';  
element.style.backgroundColor = 'blue';
```

5. Creating and Inserting Elements

- **Creating New Elements:**

- `createElement()` : Creates a new element.
- `appendChild()` : Adds a new child element to a parent node.
- `insertBefore()` : Inserts an element before another element.

```
const newDiv = document.createElement('div');
newDiv.textContent = 'New Div';
document.body.appendChild(newDiv);
```

- **Removing Elements:**

- `removeChild()` : Removes a child element.
- `remove()` : Removes an element from the DOM.

```
const item = document.getElementById('item');
item.remove();
```

6. Event Handling

- **Adding Event Listeners:**

- `addEventListener()` : Attaches an event handler to an element.
- Common events: `click` , `mouseover` , `keydown` , `DOMContentLoaded` .

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

- **Removing Event Listeners:**

- `removeEventListener()` : Removes a previously attached event listener.

```
element.removeEventListener('click', callbackFunction);
```

7. DOM Traversal

- **Navigating between elements:**

- `parentNode` : Accesses the parent of an element.
- `childNodes` : Accesses the children of an element (returns a NodeList).
- `firstChild` , `lastChild` : Accesses the first/last child element.
- `nextSibling` , `previousSibling` : Accesses adjacent sibling nodes.

```
const parent = element.parentNode;  
const firstChild = parent.firstChild;
```