

# Assignment : JavaScript DOM

**Q1. Explain the DOM and its role in Web development.**

Ans)

The **Document Object Model (DOM)** is a programming interface for web documents. It represents the web page as a tree-like structure of objects, where each object corresponds to an element or attribute in the web page. The DOM provides a way for JavaScript to interact with and manipulate the elements and content of a web page.

When a web page is loaded in a browser, the browser parses the HTML and creates the DOM tree, which is a live representation of the page's structure. This tree-like structure allows JavaScript to interact with and modify the content, structure, and styles of the web page, making it more interactive and dynamic.

**Q2. Explain the concept of event delegation and provide a scenario where it is beneficial.**

Ans)

Event delegation is a technique where instead of attaching event listeners to multiple individual child elements, you attach a single listener to a common parent and use event bubbling to handle events from its children.

## Working

1. When an event happens on a child element, it bubbles up to the parent.
2. The parent's listener checks `event.target` (the original clicked element) to decide what to do.

## Benefits

1. **Dynamic content** : When child elements are added/removed dynamically, you don't need to reattach listeners.
2. **Performance**: Fewer listeners mean less memory usage and better performance.

**Q3. Explain the concept of Event Bubbling in the DOM.**

Ans)

**Event bubbling** is the process where an event starts from the deepest (most specific) element that triggered it and then propagates upward through its ancestors in the DOM hierarchy until it reaches the **document root**.

## How it works

1. We click on a nested element.
2. The event is first handled by the target element.
3. Then, the same event travels up to its parent, then document, triggering any matching event listeners along the way.

## Why it matters

1. It allows event delegation (one listener for many children).
2. We need to be aware of it to avoid accidental event triggering.
3. We can stop it using **`event.stopPropagation()`**.

**Q4. Explain the purpose of the `addEventListener` method in JavaScript and how it facilitates event handling in the DOM.**

Ans)

**`addEventListener`** is a JavaScript method used to attach event handlers to DOM elements without overwriting existing ones.

#### Purpose

1. Listens for specific events (like click, input, keydown) on a DOM element.
2. Allows multiple event handlers on the same element for the same event type.
3. Supports **event bubbling** and **capturing** phases through an optional parameter.
4. Enables clean separation of **HTML** and **JavaScript** instead of using inline event attributes.

#### How it facilitates event handling

1. **Flexibility:** add/remove listeners dynamically at runtime.
2. **Multiple handlers:** No risk of overwriting existing event code.
3. **Control:** Can specify event phase (capture or bubble) and other options (e.g., once, passive).
4. **Separation of concerns:** Keeps markup clean by moving behavior into JavaScript.

**Q5. Create an HTML page with a button. Use JavaScript to display an alert when the button is clicked.**

Ans) **HTML:** [link](#) **CSS:** [link](#) **JS:** [link](#)

**Q6. Create a simple image carousel using HTML and JavaScript, Design a basic HTML structure with images, and use JavaScript to implement functionality that allows users to navigate through the images.**

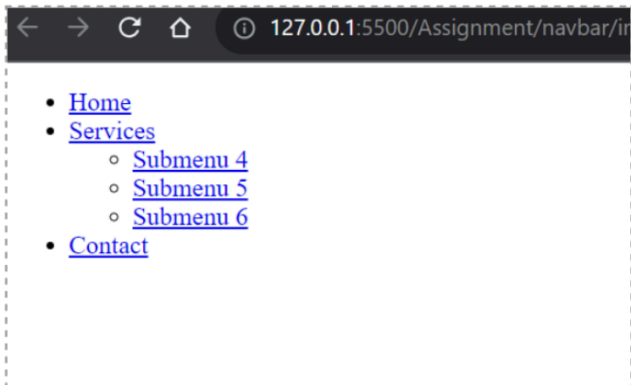
**Expected output –**



Ans) **HTML:** [link](#) **CSS:** [link](#) **JS:** [link](#)

**Q7. Build a dynamic dropdown menu using HTML and JavaScript. Create an HTML structure for a navigation menu with dropdowns. Use JavaScript to toggle the visibility of dropdowns when the user hovers over menu items.**

**Expected output -**

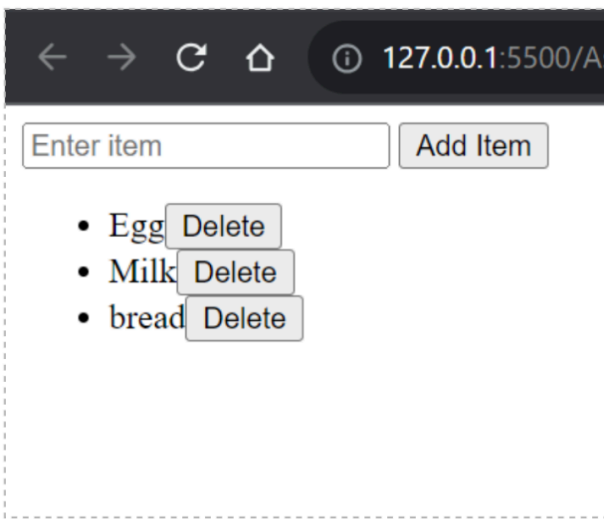


Ans) **HTML:** [link](#) **CSS:** [link](#) **JS:** [link](#)

**Q8. Create a simple dynamic shopping list with the following features**

- The item should appear in the list.
- Each item should be given a button that can be pressed to delete that item off the list.
- The input should be emptied and focused ready for you to enter another item

**Expected output -**



Ans) **HTML:** [link](#) **CSS:** [link](#) **JS:** [link](#)