# Comprehensive DOM & JavaScript Event Q&A

## 1. What is the DOM?

The Document Object Model (DOM) is a tree-like representation of an HTML or XML document. Browsers parse HTML into this DOM, allowing JavaScript to interact with the page structure, content, and styles via a standardized API.

## 2. Difference Between innerHTML, innerText, textContent

innerHTML: Gets/sets HTML markup inside an element (parses HTML, can introduce XSS).

textContent: Gets/sets raw text, ignores HTML tags, includes hidden elements (fast).

innerText: Gets/sets visible text, respects CSS styling and layout (slower, reflow).

outerHTML: Includes the element's own tags plus its inner HTML.

Example:

<div id='d'>Hello <span style='display:none'>world</span></div>

document.getElementById('d').innerHTML → 'Hello <span style="display:none">world</span>'

document.getElementById('d').textContent → 'Hello world'

document.getElementById('d').innerText → 'Hello'

## 3. Difference Between NodeList and HTMLCollection

NodeList: Contains any Node objects; static if from querySelectorAll; supports forEach.

HTMLCollection: Contains only Element nodes; always live-updating as DOM changes.

Example:

const live = document.getElementsByTagName('li'); // live HTMLCollection

const staticList = document.querySelectorAll('li'); // static NodeList

## 4. Selectors in JavaScript

getElementById(): Returns single element by ID, fastest.

getElementsByClassName(): Returns live HTMLCollection of elements by class.

getElementsByTagName(): Returns live HTMLCollection by tag name.

querySelector(): Returns first matching element using CSS selector.

querySelectorAll(): Returns static NodeList of all matching elements.

## 5. Traversing the DOM

Access relatives: parentNode/parentElement, children/childNodes, firstElementChild/firstChild, nextElementSibling/nextSibling.

## 6. Creating & Removing Elements

Creating: document.createElement, document.createTextNode.

Inserting: append, prepend, insertBefore, insertAdjacentHTML.

Removing: element.remove(), parent.removeChild(child).

Example:

const li = document.createElement('li');

li.textContent = 'Item';

document.getElementById('list').append(li);

li.remove(); // or parent.removeChild(li);

## 7. Attributes (get/set)

Use getAttribute, setAttribute, removeAttribute for any attribute.

Dot notation for standard properties (e.g., element.src, element.id).

dataset for data-\* attributes.

Example:

const img = document.querySelector('img');

console.log(img.getAttribute('src'));

img.setAttribute('alt', 'New alt');

console.log(img.dataset.userId); // for data-user-id

## 8. Applying Styles via JavaScript

Methods:

- Inline: element.style.property = value.

- cssText: element.style.cssText.

- Toggle classes: element.classList.add/remove/toggle.

- CSS variables: style.setProperty('--var', value).

- Manipulating stylesheets via document.styleSheets.

## 9. Event Object & Listeners

addEventListener(type, handler, options) adds a listener.

Event object properties: type, target, currentTarget, bubbles, cancelable, defaultPrevented, timeStamp, isTrusted.

Event methods: preventDefault(), stopPropagation(), stopImmediatePropagation().

## 10. Event Capturing vs Bubbling

Capturing: from window → target, use {capture: true}.

At target: listeners fire regardless of phase.

Bubbling: from target → window, default phase.

stopPropagation stops flow; stopImmediatePropagation stops other listeners on same element.

## 11. Common Event Types

Mouse: click, dblclick, mousedown, mouseup, mousemove, mouseenter, mouseleave

Keyboard: keydown, keyup

Pointer: pointerdown, pointerup, pointermove, pointerenter, pointerleave

Form: submit, reset, input, change, focus, blur

Window: load, DOMContentLoaded, resize, scroll, unload, beforeunload

Touch: touchstart, touchmove, touchend, touchcancel

Clipboard: copy, cut, paste

Drag & Drop: dragstart, drag, dragenter, dragover, drop, dragend

## 12. Event Delegation & Simulation

Delegation: Attach listener to parent and use event.target to handle child events.

Simulation: element.click(), dispatchEvent(new Event('type')), MouseEvent, KeyboardEvent, CustomEvent.

## 13. Pointer Events

Unified API for mouse, touch, and stylus: pointerdown, pointerup, pointermove, pointerenter, pointerleave, pointerover, pointerout, pointercancel.

Properties: pointerId, pointerType, pressure, width, height, tiltX, tiltY.

## 14. Form Events & FormData

Form events: submit, reset, input, change, focus, blur.

FormData: new FormData(form) or manual, methods append, get, getAll, set, delete, entries.

AJAX submit: fetch(url, { method: 'POST', body: formData }).