1. **What is the DOM?**

 The Document Object Model (DOM) is a programming interface that allows developers to interact with and manipulate HTML and XML documents.

1. **What is the difference between the DOM and the BOM?**

The **DOM (Document Object Model)** represents the structure of a web page and allows manipulation of HTML and CSS.

The **BOM (Browser Object Model)** includes everything provided by the browser, such as window, navigator, screen, location, and history.

1. **How is the DOM structured?**

The DOM is structured as a **tree .**The **root node** is the document object, which contains elements (html, head, body, etc.), attributes, and text nodes.

1. **What are nodes in the DOM?**

Nodes are individual objects that make up the DOM tree, such as elements, attributes, comments and text.

1. **What are the different types of nodes in the DOM?**

 **Element nodes** (<div>, <p>, etc.)

 **Attribute nodes** (e.g., id="header")

 **Text nodes** (text inside elements)

 **Comment nodes** (<!-- This is a comment -->)

 **Document nodes** (the entire document)

1. **What is the difference between document and window objects?**

The document object represents the DOM and provides methods to interact with the content of the web page.

The window object represents the browser window and provides methods and properties to interact with the browser e.g window.alert

1. **What is the purpose of the getElementById method?**

The getElementById method is used to select a single element from the DOM by its unique id attribute. It returns the element as an object or null if no element is found.

1. **What is the difference between querySelector and getElementsByClassName?**

The getElementsByClassName method is a built-in JavaScript function that allows you to retrieve all elements in the DOM that have a specific class name. It returns a live HTMLCollection of elements

(querySelector: Returns the first element that matches a CSS selector (e.g., querySelector('.class')).

getElementsByClassName: Returns a live HTMLCollection of all elements with the specified class name.)

1. **How do you add a new element to the DOM?**

document.createElement() and appendChild().

1. **How do you remove an element from the DOM?**

removeChild() or remove()

**Intermediate**

1. **What are the differences between innerHTML, innerText, and textContent?**

innerHTML: Returns or sets the HTML content inside an element, including tags. Gets or sets **HTML content**.

innerText: Returns or sets the visible text content of an element, excluding hidden elements and respecting CSS styling. Gets or sets **only visible text**, excluding hidden elements.

textContent: Returns or sets the text content of an element, including hidden elements, but without HTML tags. Gets or sets **all text**, including hidden elements.

1. **How do you dynamically change the styles of a DOM element?**

Use the style property

1. **What is event delegation?**

Event delegation is a technique where you attach a single event listener to a parent element to handle events for multiple child elements.

1. **Explain the difference between capturing and bubbling phases in event propagation.**

 **Capturing phase**: Events move **from top to bottom** (window → document → element).

 **Bubbling phase**: Events move **from bottom to top** (element → document → window).

 By default, events **bubble** in JavaScript.

1. **How do you prevent event propagation in JavaScript?**

event.stopPropagation()

**Event propagation** is the way events travel through the **DOM tree** when an event occurs on an element. It consists of three phases:

1. **Capturing Phase (Trickling Down)**
   * The event starts at the **root (window) and moves down** to the target element.
2. **Target Phase**
   * The event reaches the **target element** (the element that triggered the event).
3. **Bubbling Phase (Going Up)**
   * The event **bubbles up** from the target element back to the window.
4. **What is the purpose of the createElement method?**

The createElement method is used to create a new HTML element in memory, which can then be added to the DOM using methods like appendChild or insertBefore.

1. **What are the differences between appendChild and insertBefore?**

appendChild: Adds a node as the last child of a parent element.

insertBefore: Inserts a node before a specified reference child node within a parent element.

1. **What is the difference between childNodes and children?**

childNodes: Returns a NodeList of all child nodes, including text nodes, comment nodes, and element nodes. Returns **all** child nodes (including text and comment nodes).

children: Returns an HTMLCollection of only the child element nodes. Returns **only element** nodes.

1. **How do you find an element’s parent, sibling, or child nodes in the DOM?**

Parent: element.parentNode or element.parentElement.

Sibling: element.previousSibling, element.nextSibling, element.previousElementSibling, or element.nextElementSibling.

Child: element.childNodes, element.children, element.firstChild, or element.lastChild.

1. **What is the difference between dataset and getAttribute for accessing custom data attributes?**

dataset: Provides access to custom data attributes (e.g., data-\*) as properties of an object (e.g., element.dataset.key).

getAttribute: Retrieves the value of any attribute, including custom data attributes (e.g., element.getAttribute('data-key')).