Lesson : JavaScript DOM

DOM Structure:

- Represents the document structure as a tree of nodes.
- Each node represents a part of the document (e.g., element, text).
- The root node is the document itself.
- Nodes are connected in a parent-child hierarchy.
- Allows dynamic access and manipulation of the document.

Methods of DOM :-

getElementById():

- Retrieves an element by its unique ID.
- Returns a single element.
- Commonly used for accessing specific elements.

getElementsByClassName():

- Retrieves all elements with a specified class name.
- Returns a live HTMLCollection.
- Can be accessed by index.

getElementsByTagName():

- Retrieves all elements with a specified tag name.
- Returns a live HTMLCollection.
- Can be used to access any tag (e.g., div, p).

querySelector():

- Retrieves the first element that matches a specified CSS selector.
- Returns a single element.
- Can use complex CSS selectors.

querySelectorAll():

- Retrieves all elements that match a specified CSS selector.
- Returns a static NodeList.
- Useful for selecting multiple elements.

innerText:

- Retrieves or sets the text content of an element.
- Preserves HTML structure within the element.
- Reflects visible content only.

innerHTML:

- Retrieves or sets the HTML content of an element.
- Can inject HTML markup.
- Useful for dynamic content updates.

textContent:

- Retrieves or sets the text content of an element.
- Includes all text, even if hidden.
- More performant than innerText.

setAttribute():

- Sets a specified attribute on an element.
- Can add new attributes or change existing ones.
- Useful for dynamically modifying element attributes.

getAttribute():

- Retrieves the value of a specified attribute.
- Can be used to read custom data attributes.
- Useful for accessing element properties.

style property:

- Accesses or sets inline styles of an element.
- Can manipulate individual CSS properties.
- Direct way to style elements dynamically.

style.setProperty():

- Sets a CSS property on an element.
- Allows setting custom property names.
- Provides control over CSS variables.

cssText:

- Sets or retrieves the entire inline style as a text string.
- Can apply multiple styles at once.
- Useful for quick style changes.

classList.add():

- Adds one or more class names to an element.
- Can add multiple classes in one call.
- Ensures classes are not duplicated.

classList.remove():

- Removes one or more class names from an element.
- Can remove multiple classes in one call.
- Useful for dynamic class management.

classList.toggle():

- Toggles a class name on or off.
- Adds the class if it's not present, removes it if it is.
- Useful for switch-like behavior.

classList.contains():

- Checks if an element has a specified class.
- Returns a boolean value.
- Useful for conditionally applying styles.

Navigation & relationships:

- Accesses element siblings, parents, and children.
- Methods like parentNode, childNodes, firstChild, etc.
- Navigates the DOM tree.

createElement():

- Creates a new element node.
- Can specify the type of element (e.g., div, p).
- Useful for dynamically adding content.

appendChild():

- Appends a node as the last child of a parent node.
- Moves the node if it's already in the document.
- Essential for DOM manipulation.

append():

- Inserts content at the end of an element.
- Can append multiple nodes or strings.
- More flexible than appendChild.

removeChild():

- Removes a child node from a parent node.
- Requires reference to the child node.
- Essential for removing elements.

replaceChild():

- Replaces a child node with a new node.
- Requires references to both nodes.
- Useful for updating content.

Events in JS

What are events?:

- Actions or occurrences that happen in the browser.
- Triggered by user interactions or other actions.
- Examples include clicks, key presses, and page loads.

Types of events:

- User interface events (e.g., click, focus).
- Keyboard events (e.g., keydown, keyup).
- Mouse events (e.g., mouseover, mouseout).
- Form events (e.g., submit, change).

JavaScript event listeners & ways to add them:

- Listeners are functions that respond to events.
- Can be added via HTML attributes, properties, or methods.
- Examples: onclick attribute, element.onclick property, addEventListener method.

addEventListener & removeEventListener:

- addEventListener: Attaches an event handler to an element.
- removeEventListener: Removes an event handler from an element.
- Supports multiple handlers for the same event.

Event propagation: Bubbling and Capturing:

- Bubbling: Event starts from the target element and bubbles up.
- Capturing: Event starts from the root and captures down to the target.
- Can control the phase at which an event listener is triggered.

JavaScript event method:

- Methods associated with event objects.
- Examples: preventDefault, stopPropagation.
- Used to manage event behavior.

Event delegation:

- Technique to handle events at a higher level.
- Uses a single event listener for multiple elements.
- Improves performance and simplifies code.

stopPropagation:

- Stops the event from bubbling up the DOM tree.
- Prevents other handlers from being executed.

stopImmediatePropagation:

- Stops the event from bubbling and prevents other handlers on the same element.
- Useful for ensuring no other handlers run.

preventDefault:

- Prevents the default action associated with the event.
- Commonly used with form submissions and links.

NodeList vs. HTMLCollection

What is NodeList:

- A collection of DOM nodes.
- Can be static or live.
- Accessed via methods like querySelectorAll.

What is HTML Collection:

- A collection of HTML elements.
- Always live.
- Accessed via methods like getElementsByClassName.

NodeList vs. HTMLCollection:

- NodeList can contain any node type; HTML Collection contains only elements.
- HTML Collection is always live; NodeList can be static.
- NodeList supports forEach method; HTML Collection does not.

Convert NodeList to Array:

- Use Array.from() or spread operator.
- Enables use of array methods on NodeList.

Convert HTMLCollection to Array:

- Use Array.from() or spread operator.
- Allows manipulation with array methods.