Day 12

What is DOM and BOM

DOM stands for Document Object Model.

DOM is a programming interface (API) for representing and interacting with HTML, XHTML and XML documents. It organizes the elements of the document in a tree structure (DOM tree) and in the DOM tree, all elements of the document are defined as objects (tree nodes) which have properties and methods.

DOM tree objects can be accessed and manipulated with the help of any programming language since it is cross-platform and language-independent. Typically, we manipulate the DOM tree with the help of JavaScript and jQuery using multiple ways of accessing elements by their class names, ID, or the name of the element.

BOM stands for Browser Object Model. Unlike DOM, there is no standard defined for BOM, hence different browsers implement it in different ways. Typically, the collection of browser objects is collectively known as the Browser Object Model.

Each HTML page that is loaded into a browser window becomes a Document object and document object is an object in the BOM. You can say BOM is a superset of DOM. BOM has many objects, methods, and properties that are not the part of the DOM structure.

Document Object Model (DOM)	Browser Object Model (BOM)
It mainly focuses on the structure of the displayed document.	It mainly focuses on browser-specific functionality.

Document Object Model (DOM)	Browser Object Model (BOM)
It facilitates a standardized interface to access and modify the elements and content of an HTML or XML document.	It allows JavaScript to interact with browser features beyond the scope of manipulating the document structure.
When an HTML document gets loaded in the browser, then it becomes a document object.	In this case, the window object will be created automatically by the browser.
It facilitates access and manipulation, along with dynamically updating the structure, content, and styling of the web document.	It facilitates the different functionality for governing the browser window, handling the navigation, managing history, and accessing browser-related information.
It provides direct access control to the content of the web document, along with permitting the traversal and modification of its elements and attributes.	It doesn't have any access to the content of the web document directly.

How to use document.getElementsByTagName? The **getElementsByTagName** method of Document interface returns an HTMLCollection of elements with the given tag name.

The complete document is searched, including the root node. The returned HTMLCollection is live, meaning that it updates itself automatically to stay in sync with the DOM tree without having to call document.getElementsByTagName() again

Day 12