Module 6

JAVASCRIPT BASIC & DOM

(Basic logic Question)

Q.50 What is the drawback of declaring methods directly in JavaScript objects?

In JavaScript, declaring methods directly in objects can lead to potential drawbacks, primarily related to memory usage and performance optimization.

Here are some drawbacks :

1. **Memory Usage :** When you declare methods directly in JavaScript objects, each instance of the object will have its own copy of the methods.
2. **Performance impact :** Since each instance of the object has its own copy of methods, it can impact the performance of your application, especially in scenarios where you have many instances of the object or when the methods are complex and consume significant CPU resources.
3. **Inflexibility :** Declaring methods directly in objects can make your code less modular and harder to maintain. It becomes challenging to reuse the methods across different objects or to update them without affecting all instances of the object.

4. **Difficulty in Managing Prototypes :** When methods are directly declared in objects, it becomes more difficult to leverage JavaScript's prototype-based inheritance. To mitigate these drawbacks, it's often recommended to use prototype-based inheritance or to define methods outside of the object and attach them to the object's prototype. Additionally, it makes your code more modular and easier to maintain in the long run.

Q.60 What is Bom vs Dom in JS?

In JavaScript, BOM and DOM are two distinct but related concepts :

1. **BOM (Browser Object Model):**

   The Browser Object Model (BOM) represents everything in the browser environment that is not directly related to the document being displayed. It provides JavaScript interfaces for interacting location, navigator, screen and more.

   Some commonly used objects and properties in the BOM include window, document, location, navigator, history, screen , local storage, session storage etc.

   The BOM allows JavaScript to interact with and control various aspects of the browser window and the browsing environment, such as opening and closing windows, navigating to different URLs, storing data

locally, detecting browser properties, and handling user interactions.

2.DOM (Document Object Model :

The Document Object Model (DOM) represents the hierarchical structure of an HTML or XML document as a tree of objects. It provides a way for scripts to dynamically access, manipulate, and update the content, structure, and style of web pages.

The DOM provides JavaScript interfaces for accessing and manipulating elements and their attributes, properties, and content. Common methods and properties include getElementById(), querySelector(), setAttribute(), appendChild() etc.

The DOM allows JavaScript to dynamically modify the content and structure of web pages in response to user actions, events, or application logic.

In summary, the BOM represents the browser window and its components, while the DOM represents the structure and content of an HTML or XML document. Both BOM and DOM are essential for building interactive and dynamic web applications with javaScript.

Q.61 Array vs object defences in JS?

In JavaScript, both arrays and objects are used for storing and organizing data, but they serve different purposes and

have different characteristics. Here's a comparison of arrays and objects in javaScript.

Arrays :

1. Ordered Collection :

Arrays are ordered collections of values. Each value in an array is identified by an index, starting from 0 for the first element.

2.Use Cases :

Arrays are typically used when you need to store a collection of values that are related and ordered, such as a list of items or a series of values.

3. Accessing Elements :

You can access elements in an array using square brackets (**[]**) notation with the index of the element.

4.Length Property :

Arrays have a length property that indicates the number of elements in the array.

5.Methods :

Arrays come with a variety of built-in methods for manipulating and working with the elements, such as push(), shift(), unshift(), slice(), splice(), forEach(), map(), etc.

Objects :

1.Key-Value Pairs : Objects are collections of key-value pairs, where each key is a unique string (or symbol) that maps to a value.

2.Use Cases: Objects are used when you need to store data in a key-value format or when you need to represent entities with properties and value.

3.Accessing Properties : You can access properties in an object using dot notation (object.property) or bracket notation (object[‘property’]

4.No Ordering: Unlike arrays, objects do not guarantee any particular order for their properties. The order of properties in an object is not guaranteed to be the same as the order in which they were defined.

5.Methods: Objects do not have built-in methods like arrays do. However, you can iterate over the properties of an object using **for…in** loop or use methods like **Object.keys(), Object.values(), and Object.entries()** to work with object properties.

Defenses :

1.Array :

Arrays are suitable for storing homogeneous data or a collection of similar items. They provide easy access to elements using indices.

Arrays are best used when you need ordered data or when you need to perform operations like sorting or filtering based on index.

Use arrays when you need to iterate through items in a specific order or when you need to perform operations like adding, removing, or updating elements in a collaction.

2.Object :

Objects are suitable for storing heterogeneous data or a collection of key-value pairs. They provide easy access to values using keys.

Objects are best used when you need to represent entities with properties or when you need to access data based on specific keys.

Use objects when you need to store data in a structured format or when you need to perform operations like looking up values based on keys or adding/removing properties dynamically.

Choosing between arrays and objects depends on the specific requirements of your application and the type of data you need to store and manipulate. In many cases, you may find that you need to use both arrays and objects together to effectively manage your data.