# ASSIGNMENT - 4 (JAVASCRIPT BASIC & DOM)

1. What is JavaScript?

Ans: JavaScript is an essential language for web development, providing the tools necessary to create rich, interactive user experiences. Its versatility and widespread support make it a fundamental skill for developers in the tech industry.

2. What is the use of isNaN function?

Ans: The isNaN function in JavaScript is used to determine whether a value is NaN (Not-a-Number). NaN is a special value that represents an invalid or unrepresentable number, typically resulting from mathematical operations that do not yield a valid number.

3. What is negative Infinity?

Ans: In JavaScript, Negative Infinity is a special numeric value that represents a value that is less than any finite number. It is one of the two "Infinity" values in JavaScript, the other being Positive Infinity.

4. Which company developed JavaScript?

Ans: JavaScript was developed by Netscape Communications Corporation. The language was created by Brendan Eich in 1995 while he was working at Netscape. Originally named "Mocha," it was later renamed to "LiveScript" and finally to "JavaScript."

5. What are undeclared and undefined variables?

Ans: 1. Undeclared Variables

An undeclared variable is a variable that has not been declared using var, let, or const. If you try to use such a variable, JavaScript will throw a ReferenceError.

2. Undefined Variables

An undefined variable is a variable that has been declared but has not been assigned a value. In JavaScript, if you declare a variable without initializing it, its value is automatically set to undefined.

6. Write the code for adding new elements dynamically?

Ans: <body>

<h1>Dynamic List</h1>

<ul id="myList">

<!-- New items will be added here -->

</ul>

<input type="text" id="itemInput" placeholder="Enter new item" />

<button id="addButton">Add Item</button>

<script>

// Get references to the input field, button, and the list

const inputField = document.getElementById('itemInput');

const addButton = document.getElementById('addButton');

const myList = document.getElementById('myList');

// Function to add a new item to the list

function addItem() {

const newItemText = inputField.value.trim(); // Get the input value and trim whitespace

if (newItemText) { // Check if the input is not empty

const newListItem = document.createElement('li'); // Create a new list item

newListItem.textContent = newItemText; // Set the text content of the new item

myList.appendChild(newListItem); // Append the new item to the list

inputField.value = ''; // Clear the input field

} else {

alert('Please enter an item.'); // Alert if the input is empty

}

}

</script>

7. What is the difference between ViewState and SessionState?

Ans: In ASP.NET, ViewState and SessionState are both mechanisms used to maintain state across postbacks in web applications, but they serve different purposes and operate in distinct ways. Here’s a detailed comparison of the two:

### Summary of Key Differences

| **Feature** | **ViewState** | **SessionState** |
| --- | --- | --- |
| Purpose | Maintain state of a single page | Maintain state across multiple pages |
| Storage Location | Client-side (hidden field) | Server-side |
| Scope | Single page | User session |
| Lifetime | Page lifecycle | User session lifetime |
| Size Limitation | Limited by page size | Larger, server-dependent |
| Security | Visible in page source | More secure, stored on the server |
| Use Case | Control values between postbacks | User-specific data across pages |

8. What is === operator?

Ans:- The === operator in JavaScript is known as the strict equality operator. It is used to compare two values for equality, but with a key distinction from the regular equality operator (==)

9. How can the style/class of an element be changed?

Ans:- Summary of Methods

1.Changing Style:

Use element.style.property to change individual CSS properties.

Example: element.style.backgroundColor = 'red';

2.Changing Class:

Use element.className to set the entire class attribute.

Use element.classList to manipulate classes (add, remove, toggle).

Example: element.classList.add('new-class'); or element.classList.remove('old-class');

Conclusion :

By using JavaScript to manipulate the style and class properties of elements, you can create dynamic and interactive web applications that respond to user actions and events. This allows for a wide range of visual effects and user interface changes on the fly.

10. How to read and write a file using JavaScript?

Ans:- Reading and writing files using JavaScript can vary depending on the environment in which you are running your JavaScript code. Below, I'll outline how to read and write files in both the browser and Node.js environments.

11. What are all the looping structures in JavaScript?

Ans:- In JavaScript, there are several looping structures that allow you to execute a block of code repeatedly based on certain conditions. Here’s a comprehensive overview of the different types of loops available in JavaScript:

**Summary of Looping Structures**

* **for loop**: Ideal for a known number of iterations.
* **while loop**: Executes as long as a condition is true; may not execute at all if the condition is false.
* **do...while loop**: Similar to **while**, but guarantees at least one execution.
* **for...of loop**: Iterates over iterable objects (like arrays) to get values.
* **for...in loop**: Iterates over the properties of an object; not recommended for arrays.
* **Array.prototype.forEach()**: A method to iterate over array elements with a callback function.

12. How can you convert the string of any base to an integer in JavaScript?

Ans:- In JavaScript, you can convert a string representing a number in any base (from base 2 to base 36) to an integer using the parseInt() function. The parseInt() function takes two arguments: the string to be converted and the base (radix) of the number system you are converting from

13. What is the function of the delete operator?

Ans:- The delete operator in JavaScript is used to remove a property from an object or an element from an array. It can also be used to delete variables that are declared with var, let, or const, but this is not common practice and generally not recommended.

14. What are all the types of Pop up boxes available in JavaScript?

Ans:- **Summary of Popup Boxes**

| **Popup Type** | **Purpose** | **Buttons Available** |
| --- | --- | --- |
| **alert** | Display a message to the user | OK |
| **confirm** | Get confirmation from the user | OK, Cancel |
| **prompt** | Collect input from the user | OK, Cancel |

15. What is the use of Void (0)

Ans:- **Key Points about void(0)**

1. **Returns undefined**: The primary purpose of using **void(0)** is to explicitly return **undefined**. This can be useful in certain situations where you want to ensure that a function or link does not return any value.
2. **Preventing Default Behavior**: When used in an **href** attribute of an anchor (**<a>**) tag, **void(0)** effectively creates a link that does nothing when clicked. This is often used in conjunction with JavaScript event handlers.

16. How can a page be forced to load another page in JavaScript?

Ans:- In JavaScript, you can force a web page to load another page using several methods. The most common methods include using the **window.location** object, the **window.open()** method, and the **document.location** object. Here’s a detailed explanation of each method:

17. What are the disadvantages of using innerHTML in JavaScript?

Ans:- While innerHTML can be a quick and easy way to manipulate the DOM, it comes with several disadvantages, particularly regarding security, performance, and maintainability. It's important to be aware of these issues and consider safer and more efficient alternatives when working with dynamic content in your web applications.