**MODULE 4**

**JAVASCRIPT BASIC & DOM**

**Q1. What is javascript ?**

Ans. JavaScript is a high-level, interpreted programming language that is primarily used to add interactive elements to web pages. It is a versatile language that can be used to create dynamic content, respond to user interactions, and manipulate the Document Object Model (DOM) of a web page.

**Q2. What is the use of isNaN function?**

Ans. The isNaN() function in JavaScript is used to determine whether a given value is not a number. It returns a Boolean value: true if the value is not a number, and false if the value is a number.

**Q3. What is negative Infinity?**

Ans. **Negative Infinity** in JavaScript is a special value that represents a number less than any other finite number. It's often used to indicate a value that is extremely large or negative.

**Q4. Which company developed JavaScript?**

Ans. **Netscape** developed JavaScript. It was initially called LiveScript, but the name was later changed to JavaScript in a marketing agreement with Sun Microsystems (now Oracle).

**Q5. What are undeclared and undefined variables?**

Ans. **1. Undeclared variables:**

* These variables are used without being declared using the var, let, or const keywords.
* They are typically created when you try to assign a value to a variable that doesn't exist yet.
* Accessing an undeclared variable will result in a **ReferenceError**.

**2. Undefined variables:**

* These variables are declared but have not been assigned a value.
* They are typically initialized with the undefined value.
* Accessing an undefined variable will result in the undefined value.

**Q6. Write the code for adding new elements dynamically?**

Ans. function addNewElement(tagName, textContent) { const newElement = document.createElement(tagName);

newElement.textContent = textContent; document.body.appendChild(newElement); }

addNewElement("p", "This is a dynamically added paragraph.");

**Q7. What is the difference between ViewState and** **SessionState?**

Ans. **ViewState:**

* **Scope:** Stores information specific to a single page within a web application.
* **Persistence:** Data is stored in a hidden field on the page, making it visible to the user.
* **Performance:** Can impact page performance, especially with large amounts of data.
* **Security:** Data can be tampered with by malicious users.
* **Use cases:** Storing temporary data for a single page, such as form values or control properties.

**SessionState:**

* **Scope:** Stores information specific to a single user's session across multiple pages.
* **Persistence:** Data is stored on the server, making it invisible to the user.
* **Performance:** Generally has less impact on performance than ViewState.
* **Security:** Data is more secure than ViewState, as it is not exposed to the user.
* **Use cases:** Storing user-specific data, such as login information, shopping cart items, or personalization settings.

**Q8. What is === operator?**

Ans. The === operator in JavaScript is a strict equality operator. It compares both the value and the type of two operands. If both the value and the type are equal, it returns true. Otherwise, it returns false.

**Q9. How can the style/class of an element be changed?**

Ans. **1. Using the style property**

Ex. const element = document.getElementById("myElement"); element.style.color = "red"; element.style.fontSize = "20px";

**2. Using the classList property**

Ex. const element = document.getElementById("myElement"); element.classList.add("active"); element.classList.remove("inactive"); element.classList.toggle("hidden");

**3. Using the className property**

Ex. const element = document.getElementById("myElement"); element.className = "new-class";

**Q10. How to read and write a file using JavaScript?**

Ans. The[fs.readFile()](https://www.geeksforgeeks.org/node-js-fs-readfile-method) and [rs.writeFile()](https://www.geeksforgeeks.org/node-js-fs-writefile-method) methods are used to read and write of a file using javascript.

Ex. fs.readFile( file\_name, encoding, callback\_function )

fs.writeFile( file\_name, data, options, callback )

**Q11. What are all the looping structures in JavaScript?**

Ans. **1. for Loop**

Ex. for (let i = 0; i < 5; i++) { console.log(i); }

**2. for...in Loop**

Ex. const person = { name: "Vraj", age: 20 }; for (let key in person) { console.log(key + ": " + person[key]); }

**3. for...of Loop**

Ex. const persons = ["Het", "Vraj", "Shubham"]; for (let Person of Persons) { console.log(Person); }

**4. forEach Method**

Ex. const numbers = [1, 2, 3, 4, 5]; numbers.forEach(function(number) { console.log(number); });

**5. while Loop**

Ex. let count = 0; while (count < 5) { console.log(count); count++; }

**6. do...while Loop**

Ex. let i = 0; do { console.log(i); i++; }

while (i < 5);

**Q12. How can you convert the string of any base to an integer in JavaScript?**

Ans. **1. Using parseInt() method**

Ex. **let** str = "100";

console.log(**typeof** str);

console.log(**typeof** parseInt(str));

**2. Using number() method**

Ex. let str1 = "23";

let str2 = "100.45";

console.log("Before Conversion: ",

typeof str1, typeof str2);

console.log("Before Conversion: ",

typeof Number(str1), typeof Number(str2));

**3. Using unary operator**

Ex. let str1 = "20";

let str2 = "Vraj";

const str3 = '100';

console.log(

"Before Conversion: ", typeof str1,

typeof str2, typeof str3

);

console.log(

"After Conversion: ", typeof +str1,

typeof +str2, typeof +str3

);

**Q13. What is the function of the delete operator?**

Ans. The delete operator in JavaScript is used to remove a property from an object. It returns true if the deletion was successful and false if the property does not exist or cannot be deleted.

Ex. const person = { name: "Vraj", age: 20, city: "Ahmedabad" };

delete person.age;

console.log(person);

**Q14. What are all the types of Pop up boxes available in JavaScript?**

Ans. **1. Alert Box**

Ex. alert("This is an alert box!");

**2. Confirm Box**

Ex. const confirmed = confirm("Are you sure you want to continue?");

if (confirmed) { "OK" }

else { "Cancel" }

**3. Prompt Box**

Ex. const name = prompt("Enter your name:", "John Doe"); if (name !== null) {

console.log("Hello, " + name + "!"); }

**4. Custom Pop-up Boxes**

Ex. const popup = document.getElementById("myPopup"); const closeButton = document.querySelector(".close-button"); function openPopup() { popup.style.display = "block"; } function closePopup() { popup.style.display = "none"; } closeButton.addEventListener("click", closePopup);

**Q15. What is the use of Void (0)?**

Ans. Void(0) is a JavaScript expression that evaluates to undefined. It's often used in situations where you need to execute a function or perform an action but don't want to return a specific value or assign a value to a variable.

Ex. function doNothing() {

return void(0); }

**Q16. How can a page be forced to load another page in JavaScript?**

Ans.

* **Using the location object:**

Ex. location.href = "https://www.Bookmyshow.com";

location.replace("https://www.indianrail.com");

* **Using the window.open() method:**

Ex. window.open("https://www.nike.com");

* **Using the window.location.reload() method:**

Ex. Window.location.reload ();

**Q17. What are the disadvantages of using innerHTML in JavaScript?**

Ans. **Disadvantages of using innerHTML:**

* **Security vulnerabilities:** Can lead to XSS attacks if not sanitized.
* **Performance issues:** Can be less performant for large amounts of content.
* **Loss of event handlers:** Can remove event handlers attached to elements.
* **Reduced readability:** Can make code less readable and maintainable.
* **Limited control over attributes:** Difficult to modify attributes compared to DOM API.

< THE END >