1.What is JavaScript?

JavaScript is a programming language that is commonly used to add interactivity and dynamic features to websites. It runs on the user's web browser and allows developers to create interactive elements, validate forms, manipulate web page content, and perform various other tasks to enhance the functionality of websites.

2. What is the use of isNaN function?

isNan function returns true if the argument is not a number; otherwise, it is false.

3. What is negative Infinity?

Negative Infinity is a number in JavaScript which can be derived by dividing negative number by zero.

4. Which company developed JavaScript?

Netscape is the software company that developed JavaScript.

5. What are undeclared and undefined variables?

Undeclared variables are those that do not exist in a program and are not declared. If the program tries to read the value of an undeclared variable, then a runtime error is encountered.

Undefined variables are those that are declared in the program but have not been given any value. If the program tries to read the value of an undefined variable, an undefined value is returned.

6. Write the code for adding new elements dynamically?

<!DOCTYPE html>

<html>

<head>

<title>Add Elements Dynamically</title>

</head>

<body>

<ul id="myList">

<li>Item 1</li>

<li>Item 2</li>

</ul>

<button onclick="addNewElement()">Add New Element</button>

<script>

function addNewElement() {

// Step 1: Select the parent element

const parentElement = document.getElementById("myList");

// Step 2: Create the new element

const newElement = document.createElement("li");

// Step 3: Set attributes or content for the new element

newElement.textContent = "New Item";

// Step 4: Append the new element to the parent element

parentElement.appendChild(newElement);

}

</script>

</body>

</html>

7. What is the difference between ViewState and SessionState?

‘ViewState’ is specific to a page in a session.

‘SessionState’ is specific to user-specific data that can be accessed across all web application pages.

8. What is === operator?

“===” is called a strict equality operator, which returns true when the two operands have the same value without conversion.

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

It can be done in the following way:

document.getElementById("myText"). style. fontSize = "20";

or

document. getElementById ("myText"). className = "anyclass";

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

There are two ways to read and write a file using JavaScript

1. Using JavaScript extensions
2. Using a web page and Active X objects

11. What are all the looping structures in JavaScript?

Following are looping structures in Javascript:

* For 🡺( For in , for of, forEach )
* While
* Do-while loops

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

The parseInt() function is used to convert numbers between different bases. parseInt() takes the string to be converted as its first parameter. The second parameter is the base of the given string.

To convert 4F (or base 16) to integer, the code used will be –

parseInt ("4F", 16);

13. What is the function of the delete operator?

The delete keyword is used to delete the property as well as its value.

Example

var student= {age:20, batch:"ABC"};

Delete student. age;

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

1. Alert
2. Confirm and
3. Prompt

15. What is the use of Void (0)?

Void(0) is used to prevent the page from refreshing, and parameter “zero” is passed while calling.

Void(0) is used to call another method without refreshing the page.

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

The following code has to be inserted to achieve the desired effect:

window.location.href = "https://www.example.com";

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

1. innerHTML can expose your application to security risks like cross-site scripting (XSS) attacks.
2. Manipulating innerHTML can be computationally expensive, impacting performance.
3. Setting innerHTML can result in loss of event listeners and data attached to elements.
4. innerHTML replaces the entire content of an element, lacking control over partial updates.
5. Manipulating innerHTML directly can pose challenges for accessibility and screen reader compatibility.