

JAVASCRIPT BASIC & DOM

1. What is JavaScript

JavaScript is a widely used programming language primarily designed for creating interactive and dynamic content on websites.

JavaScript is often referred to as a "client-side" programming language because it runs in the web browser of the user, allowing developers to enhance web pages by adding behaviour and interactivity.

2. What is the use of isNaN function?

The isNaN() function in JavaScript is used to determine whether a value is "Not-A-Number" (NaN) or not.

NaN is a special value in JavaScript that represents the result of an undefined or unrepresentable mathematical operation, such as dividing zero by zero.

3. What is negative Infinity?

Negative Infinity (denoted as -Infinity) is a special value in JavaScript and many other programming languages that represents the lowest possible numeric value, which is considered to be infinitely small or approaching negative infinity.

4. Which company developed JavaScript?

JavaScript was developed by Netscape Communications Corporation, a company that played a significant role in the early days of the World Wide Web.

The language was created by Brendan Eich, a programmer working at Netscape at the time. JavaScript was originally introduced under the name "LiveScript" in September 1995 as part of Netscape Navigator 2.0. However, shortly after its initial release, it was renamed to "JavaScript" to align itself with the growing popularity of Java, another

programming language.

5. What are undeclared and undefined variables?

Undeclared Variables: An undeclared variable is a variable that has been used in code but has not been formally declared using a variable declaration statements like var, let, or const.

Undefined Variables: An undefined variable is a variable that has been declared but has not been assigned a value. In most programming languages, when you declare a variable without assigning a value to it, it is automatically assigned the special value undefined,

6. Write the code for adding new elements dynamically?

```
<body>
  <h2>Dynamic List</h2>
  <ul id="itemList">
    <li>Item 1</li>
    <li>Item 2</li>
  </ul>
  <button id="addItemButton">Add Item</button>

  <script> // Get references to the list and the button

    var itemList = document.getElementById("itemList");

    var addItemButton = document.getElementById("addItemButton");

    // Add an event listener to the button

    addItemButton.addEventListener("click", function()
    { // Create a new list item element
      var newItem = document.createElement("li");
      newItem.textContent = "New Item";
      // Append the new item to the list
      itemList.appendChild(newItem); });
  </script>
</body>
```

7. What is the difference between ViewState and SessionState?

ViewState and SessionState are concepts in web development, specifically in the context of ASP.NET, a framework for building web applications. They are both mechanisms used to manage and maintain state information in web applications, but they serve different purposes and have different scopes.

ViewState is used primarily to maintain the state of individual controls within a page. It's useful for scenarios where you want to preserve user input and selections across postbacks. However, since the data is stored on the client side, it can increase the size of the HTML output and potentially impact performance if used excessively.

SessionState is used to store data that needs to be shared between different parts of a web application during a user's session. This could include user authentication status, shopping cart contents, user preferences, and more.

The data stored in SessionState is not tied to a specific page or control but is accessible throughout the entire user session.

8. What is === operator?

The === operator, also known as the "strict equality" or "identity" operator, is used in many programming languages, including JavaScript, to compare two values for both their equality in value and their equality in data type. It checks whether the values being compared are not only equal in content but also of the same type.

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

Changing Inline Styles: You can change the inline styles of an element using the style property. The style property allows you to modify individual style properties of the element, such as colour, font size, width, etc.

Adding Classes: You can also change the appearance of an element by adding or removing CSS classes. This is a more flexible way to change styles because you can define different styles in your CSS and apply them using classes

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

In web browsers, JavaScript doesn't have direct access to the file system for

security reasons. However, you can work with files using the File API, which provides a way to interact with files selected by the user through file input fields. You can read the content of selected files and manipulate them in memory, but you can't directly write to files on the user's machine.

11. What are all the looping structures in JavaScript?

JavaScript provides several looping structures to help you repeat a block of code multiple times. The main looping structures in JavaScript are:

for Loop: The for loop is a widely used looping structure that allows you to execute a block of code a specific number of times. It typically consists of three parts: initialization, condition, and increment.

while Loop: The while loop continues executing a block of code as long as a specified condition is true.

do...while Loop: The do...while loop is similar to the while loop, but it ensures that the code block is executed at least once before checking the condition.

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

In JavaScript, you can convert a string representation of a number in any base (binary, octal, decimal, or hexadecimal) to an integer using the `parseInt()` function. The `parseInt()` function takes two arguments: the string to be converted and the base of the number system that the string represents.

13. What is the function of the delete operator?

The delete operator in JavaScript is used to delete properties from objects, as well as elements from arrays. Its primary function is to remove a specific property or element from an object or array.

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

JavaScript?

JavaScript provides three types of popup boxes or dialogs that you can use to interact with users and display messages or prompts:

Alert Box: An alert box displays a simple message to the user with an "OK" button. It's commonly used to display information or warnings.

Confirm Box: A confirm box is used to prompt the user for a yes-or-no response. It displays a message along with "OK" and "Cancel" buttons.

The function returns true if the user clicks "OK" and false if the user clicks "Cancel".

Prompt Box: A prompt box is used to prompt the user for input. It displays a message, an input field for the user to type into, and "OK" and "Cancel" buttons. The function returns the value entered by the user, or null if the user clicked "Cancel".

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

In the case of using void(0), it's often seen as a way to create an "empty" link or a placeholder link that doesn't perform any action when clicked. This can be useful when you want to create clickable elements that trigger JavaScript actions but don't navigate to a new page.

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

To force a page to load another page using JavaScript, you can modify the window.location object.

The window.location object represents the current URL of the browser and allows you can navigate to a different URL, effectively loading a new page.

There are a few methods you can use to achieve this:

window.location.href: You can set the href property of the window.location object to the URL of the page you want to load. This will cause the browser to navigate to the specified URL.

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

Replacement is done everywhere When innerHTML property is used to modify, all the DOM nodes will have to be parsed and created again.

It is not possible to append innerHTML In JavaScript, '+=' is commonly used for Appending.
However, when using innerHTML to append to an HTML tag, the entire tag is re-parsed.