

## JavaScript Assignment

### 1. What is JavaScript?

→ JavaScript is a high-level programming language that is primarily used for web development. It is often referred to as the "language of the web" because it is widely used to add interactivity and dynamic behavior to websites.

Here are some key features of JavaScript:

1. **Client-Side Scripting:** JavaScript is primarily executed on the client-side, meaning it runs on the user's web browser rather than on the server. This allows it to interact with the Document Object Model (DOM) of a web page, enabling dynamic content updates and user interactions without requiring a page reload.
2. **Versatility:** JavaScript can be used for a wide range of tasks, including form validation, creating interactive elements like menus and sliders, handling events, making asynchronous requests to servers (AJAX), and much more. It can also be used on the server-side (e.g., with Node.js) to build scalable web applications.
3. **Syntax:** JavaScript has a syntax similar to other C-style languages, such as Java or C++. However, it also borrows concepts from functional programming, allowing developers to write code in different styles.
4. **Dynamic Typing:** JavaScript is dynamically typed, meaning you don't need to specify the type of a variable explicitly. Variables can hold values of different types, and their types can change during runtime.
5. **Object-Oriented:** JavaScript is prototype-based, which means objects can be created without requiring class definitions. Objects can inherit properties and methods from other objects, allowing for flexible object-oriented programming.
6. **Large Ecosystem:** JavaScript has a vast ecosystem of libraries, frameworks, and tools that make development easier and more efficient. Popular libraries and frameworks include React, Angular, Vue.js, and jQuery, among many others.

JavaScript is supported by all modern web browsers, making it a fundamental technology for building interactive and dynamic web applications.

### 2. What is the use of isNaN function?

→ The **isNaN()** function is a built-in JavaScript function that stands for "is Not a Number." It is used to determine whether a given value is not a valid number.

The **isNaN()** function takes a single argument and performs the following checks:

1. If the argument is not a number (NaN), it returns **true**.
2. If the argument is a valid number, it returns **false**.

The **isNaN()** function can be useful for input validation or checking whether a value is numeric before performing arithmetic operations. However, it's important to note that **isNaN()** has some quirks. For instance, it attempts to convert non-numeric values to numbers before checking for **NaN**, so a non-empty string that cannot be parsed as a number will still return **false**. To overcome this, you can use other functions like **Number.isNaN()** or the **typeof** operator to perform more accurate type checks.

### 3. What is negative Infinity?

→

### 4. Which company developed JavaScript?

→ JavaScript was developed by Netscape Communications Corporation, a software company that was prominent in the early days of the web. The initial development of JavaScript was led by Brendan Eich, who implemented the language in just ten days in May 1995. It was originally named "Mocha" and later "LiveScript" before finally being named "JavaScript."

## 5. What are undeclared and undefined variables?

→ In programming, both "undeclared" and "undefined" are terms used to describe variables in different states.

1. **Undeclared Variables:** An undeclared variable refers to a variable that has not been formally declared or defined before its first use in a program. Declaring a variable means specifying its type and optionally assigning an initial value. If you attempt to use an undeclared variable, it will typically result in a compilation error or a runtime error, depending on the programming language. Some programming languages require variables to be declared before they can be used, while others allow variables to be used without prior declaration, assuming a default type or value.

Ex. `var l; // declaration`

`y=10; // undeclare variable`

2. **Undefined Variables:** An undefined variable refers to a variable that has been declared or defined, but it does not have a valid value assigned to it. This can occur when a variable is declared without initializing it or when it is assigned a value later in the program flow. Using an undefined variable typically leads to unpredictable behavior or runtime errors, as the variable's value can be arbitrary or contain leftover data from memory.

`var x;`

`console.log(x); //undefined variable`

## 6. Write the code for adding new elements dynamically?

→ `document.createElement('div');`

## 7. What is the difference between ViewState and SessionState?

→ ViewState and SessionState are concepts related to web application development, specifically in the context of ASP.NET.

1. **ViewState:** ViewState is a mechanism used in ASP.NET web forms to persist the state of server-side controls across postbacks. It is designed to store the state of individual controls or the entire page and is used to maintain the state between subsequent requests made by the same client. ViewState is stored as a hidden field on the page and is encrypted to ensure data integrity and security.

ViewState is primarily used to maintain the state of controls, such as their property values or user inputs, across postbacks. It allows the server-side code to remember the state of controls and restore them when the page is posted back to the server. ViewState is specific to a single user and is maintained on the client-side.

2. **SessionState:** SessionState, on the other hand, is a mechanism used to store user-specific data on the server-side across multiple requests. It allows storing and retrieving user-specific information during a user session. SessionState data is unique for each user and is typically stored on the server, although different storage options are available, such as in-memory, out-of-process, or database-based storage.

SessionState is useful for storing information that needs to be maintained throughout a user's interaction with a website, such as user preferences, shopping cart items, or authentication-related data. It provides a way to persist data across multiple pages or interactions for a specific user, allowing developers to create personalized experiences.

In summary, ViewState is used to maintain the state of controls within a single page and is stored on the client-side, while SessionState is used to store user-specific data across multiple requests and is stored on the server-side.

## 8. What is === operator?

→ The `===` operator is a comparison operator used in programming languages such as JavaScript. It is often referred to as the "strict equality operator" or the "identity operator." In JavaScript, the `===` operator checks for strict equality between two operands. It compares the values of the operands and their types to determine if they are exactly equal. It returns **true** if the operands are equal and of the same type, and **false** otherwise.

In contrast, the double equals operator (==) in JavaScript performs a loose comparison, where it attempts to convert the operands to a common type before comparison. This can lead to unexpected results in some cases, so it is generally recommended to use the === operator for strict equality checks.

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

→ To apply the style on element to use the varname.style.property = "value";

To apply the class to write the varname.classList.add = "classname";

Second method id varname.className="classname";

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

→

11. What are all the looping structures in JavaScript?

→ JavaScript provides several looping structures that allow you to repeat a block of code multiple times. Here are the main looping structures in JavaScript:

1. **for loop:** The **for** loop is a common looping structure used when you know the number of iterations in advance. It consists of three parts: initialization, condition, and increment/decrement.
2. **while loop:** The while loop repeats a block of code while a specified condition is true. The loop continues until the condition becomes false.
3. **do-while loop:** The do-while loop is similar to the while loop, but it executes the code block at least once before checking the condition.
4. **for...in loop:** The for...in loop iterates over the properties of an object. It allows you to loop through enumerable properties of an object.
5. **for...of loop:** The for...of loop is used to iterate over iterable objects such as arrays, strings, and other collections. It provides an easy way to loop through the elements of an iterable.

12.