Module 6) JAVASCRIPT BASIC & DOM

• What is JavaScript?

Ans: JavaScript is a versatile programming language primarily used for creating interactive effects within web browsers. Originally developed by Netscape, it's now supported by all major web browsers and has become an integral part of web development. JavaScript allows developers to add dynamic content, interactivity, and behavior to web pages. It's often used to create things like interactive forms, animations, games, dynamic styling, and much more. Additionally, with the introduction of Node.js, JavaScript can now be used for server-side development as well.

• What is the use of isNaN function?

Ans: NaN() method returns true if a value is Not-a-Number. Number.isNaN() returns true if a number is Not-a-Number. In other words: isNaN() converts the value to a number before testing it.

" It's used to determine whether a value is NaN (not a number) or can be coerced into a valid number. This function is particularly useful when you're dealing with user inputs or calculations where you need to ensure that a value is indeed a number before performing mathematical operations.

• What is negative Infinity?

Ans: The negative infinity in JavaScript is a constant value that is used to represent a value that is

the lowest available. This means that no other number is lesser than this value. It can begenerated using a self-made function or by an arithmetic operation. Note: JavaScript shows the NEGATIVE_INFINITY value as -Infinity.
Negative infinity is different from mathematical infinity in the following ways:- .Negative infinity results in -0(different from 0) when divided by any other number..When divided by itself or positive infinity, negative infinity return NaN.Negative infinity, when divided by any positive number (apart from positive infinity) is negativeinfinity.

• Which company developed JavaScript?

Ans:JavaScript was developed by Brendan Eich while he was working at Netscape Communications Corporation. It was initially created in 1995 under the name

"Mocha," which was later renamed "LiveScript" before settling on the name "JavaScript." Netscape, a now-defunct internet browser company, introduced JavaScript to complement its Netscape Navigator browser.

• What are undeclared and undefined variables?

Ans: 1.Undeclared Variable:-

.An undeclared variable is a variable that has been used in the code without being declared

using a var, let, or const keyword.

.In some languages, this might lead to an error, while in others, the interpreter or compiler may

create a global variable with that name.

2.Undefined Variable:-

.An undefined variable is a variable that has been declared but has not been assigned a value.

.In many programming languages, variables are initialized with a default value of `undefined`

until a value is explicitly assigned.

• Write the code for adding new elements dynamically?

Ans: -Javascript is a very important language when it comes to learning how the browser works.

Often there are times we would like to add dynamic elements/content to our web pages. This post deals with all of that.

• What is the difference between ViewState and SessionState?

Ans: 1.ViewState:

Scope: ViewState is a client-side state management technique used in ASP.NET web forms. It maintains the state of individual controls and their values across postbacks within the same page.

Storage: ViewState stores data in a hidden field on the web form itself. It keeps track of the state of controls (like textboxes, dropdowns, etc.) so that their values persist across postbacks, allowing the page to remember their states without needing to fetch data again from the server.

Security: ViewState is stored on the client-side, which can be a potential security risk if sensitive data is stored there. Encryption and validation can be applied to secure the ViewState content.

2. SessionState:

Scope: SessionState, on the other hand, maintains state across multiple requests and multiple pages for a specific user session.

Storage: SessionState data is stored on the server-side either in-process (within the application's memory), in a separate process (like State Server), or in an external storage (like SQL Server). It keeps data available for the entire user session.

Usage: SessionState is typically used to store user-specific information that needs to persist as the user navigates through various pages of a website. It can store larger amounts of data compared to ViewState.

• What is === operator?

Ans: -The === operator is a strict equality operator in JavaScript. It is used to compare two values for

equality without performing type coercion. This means that not only the values being compared

must be equal, but they must also be of the same data type.

Here's how it works:

If the operands are of the same type and have the same value, `=== `returns `true`.

If the operands are of different types or have different values, `===` returns `false`.

.Example:-

5 === 5 // true, because both operands are of the same type and have the same value

5 === '5' // false, because the operands are of different types (number and string)

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

Ans:

***Changed in style:-**

1. Direct Style Property Modification:

```
// Get the element by its ID
var myElement = document.getElementById('myElementId');
```

```
// Change the background color directly
myElement.style.backgroundColor = 'red';
```

2.Using `setAttribute`:

```
// Get the element by its ID
var myElement = document.getElementById('myElementId');
// Set the style attribute directly
myElement.setAttribute('style', 'background-color: blue; color: white;');
```

***Changed in Class:-**

1. Direct Class Property Modification:

```
// Get the element by its ID
var myElement = document.getElementById('myElementId');
// Add a class
myElement.className = 'newClass';
// If you want to append a class without removing existing ones
myElement.className += ' anotherClass';
```

2.Using `classList` Propert:

```
// Get the element by its ID
var myElement = document.getElementById('myElementId');
// Add a class
myElement.classList.add('newClass');
// Remove a class
myElement.classList.remove('oldClass');
// Toggle a class (add if not present, remove if present)
myElement.classList.toggle('toggleClass');
```

• How to read and write a file using JavaScript?

Ans: -On the client side, you can't read or write files in JavaScript browsers. The fs module in Node.js
may be used to accomplish this on the server-side. It has methods for reading and writing files
on the file system that are both synchronous and asynchronous. Let's demonstrate some
examples of reading and writing files with the node.js fs module.
The fs.readFile() and rs.writeFile() methods are used to read and write of a file using javascript.
The file is read using the fs.readFile() function, which is an inbuilt method. This technique reads
the full file into memory and stores it in a buffer.

• What are all the looping structures in JavaScript?

Ans: -JavaScript supports several looping structures that allow you to execute a block of code

repeatedly. The main looping structures in JavaScript are:

1. For loop:-

The `for` loop is a common loop that repeats a block of code a specified number of times.

-for (initialization; condition; iteration) {

// code to be repeated}

2. While loop:-

The `while` loop repeats a block of code as long as a specified condition is true.

-while (condition) {

// code to be repeated

}

3. Do-While Loop:-

Similar to the `while` loop, but the block of code is executed at least once, even if the condition

is initially false.

-do {

// code to be repeated

} while (condition);

4. For...in loop:-

Iterates over the enumerable properties of an object, including inherited properties.

-for (variable in object) {

// code to be repeated

}

5. For...of loop:-

Introduced in ECMAScript 2015 (ES6), the `for...of` loop iterates over the values

of an iterable

object (arrays, strings, etc.).

-for (variable of iterable) {

// code to be repeated

}

6. forEach loop:-

Available for arrays, the `forEach` method executes a provided function once for each array element.

-array.forEach(function(element) {

// code to be repeated

});

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

Ans:In JavaScript, you can use the parseInt function to convert a

string representation of a number from any base to an integer.

The parseInt function takes two arguments: the string to be

converted and the base of the numeral system.

Example:

var a = "1101";

var Number =Number. parseInt(a);

• What is the function of the delete operator?

Ans:- Delete is comparatively a lesser-known operator in JavaScript. This operator is more

specifically used to delete JavaScript object properties.

-The JavaScript pop(), shift(), or splice() methods are available to delete an element

from an array. But because of the key-value pair in an object, deleting is more complicated. Note that, the delete operator only works on objects and not on variables or functions

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

Ans:There are three types of pop-up boxes available in JavaScript: Alert Box, Confirm Box, and Prompt Box

1. Alert Box (`alert`):

The alert box is used to display a message to the user.

It has only one button ("OK") to acknowledge the message.

2. Confirm Box (`confirm`):

The `confirm` box is used to ask the user for confirmation.

It has two buttons ("OK" and "Cancel").

3. Prompt Box(`prompt`):

The `prompt` box is used to prompt the user to enter some information.

It has an input field for the user to type in and two buttons ("OK" and "Cancel").

• What is the use of Void (0)?

Ans: JavaScript void 0 means returning undefined (void) as a primitive value. You might come across the term “JavaScript:void(0)” while going through HTML documents. It is used to prevent any side effects caused while inserting an expression in a web page.

Another use case for void in JavaScript is to facilitate returning from a function without explicitly returning a value. This can be beneficial when working with callback functions, where you want to invoke the callback but don't need to return anything.

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

Ans:Approach: We can use window. location property inside the script tag to forcefully load another page in Javascript. It is a reference to a Location object that is it represents the current location of the document. We can change the URL of a window by accessing it.

In JavaScript, you can load another page by changing the URL or by navigating to a new location. One common way to do this is by using the window.location object.

• What are the disadvantages of using innerHTML in JavaScript?

Ans:Using innerHTML in JavaScript to manipulate the DOM (Document Object

Model) has some disadvantages:

Security Vulnerabilities: innerHTML is susceptible to cross-site scripting (XSS) attacks if you're not careful with the content you're injecting. If the content comes from an untrusted source and is not sanitized properly, it can execute malicious scripts.

Performance Impact: Modifying innerHTML can be slower than other DOM manipulation methods, especially when dealing with large chunks of HTML. When you set innerHTML, the browser has to re-parse and recreate the entire

content inside the element, which can impact performance, especially on older browsers or with complex content.

Event Handlers and Data Loss: When using innerHTML to replace content, event handlers and associated data bound to elements might get lost or detached, requiring additional code to rebind them after the innerHTML change.

Overwriting Content and Memory Leaks: Replacing the entire HTML content using innerHTML might inadvertently remove existing DOM elements and their associated event listeners or data, leading to memory leaks or unexpected behavior if not handled properly.