Module (JAVASCRIPT BASIC & DOM) – 4

* What is JavaScript?

ans:- JavaScript is a dynamic programming language that's used for web development, in web applications, for game development, and lots more. It allows you to implement dynamic features on web pages that cannot be done with only HTML and CSS.

* What is the use of isNaN function?

The isNaN() function determines whether a value is NaN when converted to a number. Because coercion inside the isNaN() function can be surprising, you may alternatively want to use Number.isNaN() .

In JavaScript NaN is short for "Not-a-Number".

The isNaN() method returns true if a value is NaN.

The isNaN() method converts the value to a number before testing it.

* What is negative Infinity?

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 be generated using a self-made function or by an arithmetic operation.

**Note:**JavaScript shows the NEGATIVE\_INFINITY value as -Infinity.

* Which company developed JavaScript?

JavaScript was invented by Brendan Eich in 1995. It was developed for Netscape 2, and became the ECMA-262 standard in 1997. After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser.

* What are undeclared and undefined variables?

**Undefined:**It occurs when a variable has been declared but has not been assigned any value. Undefined is not a keyword.

**Undeclared:**It occurs when we try to access any variable that is not initialized or declared earlier using the *var* or *const keyword*

**Undefined:**

let geek;

undefined

console.log(geek)

**Undeclared:**

// ReferenceError: myVariable is not defined

console.log(myVariable)

* Write the code for adding new elements dynamically?

**Creation of new element:** New elements can be created in JS by using the **createElement()** method.

**Syntax:**

document.createElement("<*tagName*>");

// Where <*tagName*> can be any HTML

// tagName like div, ul, button, etc.

// newDiv element has been created

For Eg: **let newDiv = document.createElement("div");**

* What is the difference between ViewState and SessionState?

**Differences between ViewState and SessionState:**

|  |  |
| --- | --- |
| ViewState | SessionState |
| Maintained at page level only. | Maintained at session level. |
| View state can only be visible from a single page and not multiple pages. | Session state value availability is across all pages available in a user session. |
| It will retain values in the event of a postback operation occurring. | In session state, user data remains in the server. Data is available to user until the browser is closed or there is session expiration. |
| Information is stored on the client’s end only. | Information is stored on the server. |
| used to allow the persistence of page-instance-specific data. | used for the persistence of user-specific data on the server’s end. |
| ViewState values are lost/cleared when new page is loaded. | SessionState can be cleared by programmer or user or in case of timeouts. |
|  |  |

**Usage:**

**SessionState:** It can be used to store information that you wish to access on different web pages.

**ViewState** It can be used to store information that you wish to access from same web page.

* What is === operator?

The strict equality ( === ) operator checks whether its two operands are equal, returning a Boolean result.

console.log(1 === 1);

// Expected output: true

console.log('hello' === 'hello');

// Expected output: true

console.log('1' === 1);

// Expected output: false

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

In this article, we will learn how we can change the style/class of an element. If you want to build a cool website or app then UI plays an important role. We can change, add or remove any CSS property from an HTML element on the occurrence of any event with the help of JavaScript. There are two common approaches that allow us to achieve this task.

<!DOCTYPE html>

<html lang="en">

<body>

<h1 style="color: green;">

Dipak vaghasiya

</h1>

<h2>

How can the style/class of

an element be changed?

</h2>

<b>Validate Pan Number</b>

<input type="text" id="pan" />

<p></p>

<button id="submit">Validate</button>

<script>

const btn = document.getElementById("submit");

btn.addEventListener("click", function () {

const pan = document.getElementById("pan").value;

const para = document.querySelector("p");

let regex = /([A-Z]){5}([0-9]){4}([A-Z]){1}$/;

if (regex.test(pan.toUpperCase())) {

para.innerHTML = "Hurrey It's correct";

// Inline style

para.style.color = "green";

} else {

para.innerHTML = "OOps It's wrong!";

// Inline style

para.style.color = "red";

}

});

</script>

</body>

</html>

* How to read and write a file using JavaScript?

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. 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.

**Syntax:**

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

* What are all the looping structures in JavaScript?

JavaScript supports different kinds of loops:

* for - loops through a block of code a number of times
* for/in - loops through the properties of an object
* for/of - loops through the values of an iterable object
* while - loops through a block of code while a specified condition is true
* do/while - also loops through a block of code while a specified condition is true
* How can you convert the string of any base to an integer in JavaScript?

To convert a string to an integer parseInt(), Number(), and Unary operator(+) function is used in javascript. parseInt() function returns Nan( not a number) when the string doesn't contain number. If a string with a number is sent, then only that number will be returned as the output.

parseInt() use to string to integer value in javascript

* What is the function of the delete operator?

The JavaScript [**pop()**](https://www.geeksforgeeks.org/javascript-array-pop-method/)**,**[**shift()**](https://www.geeksforgeeks.org/javascript-array-shift-method/)**,** or [**splice()**](https://www.geeksforgeeks.org/javascript-array-splice-method/) 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.

**Syntax:**

delete object

// or

delete object.property

// or

delete object['property']

ex- let emp = {

     firstName: "Raj",

     lastName: "Kumar",

     salary: 40000

}

console.log(delete emp.salary);

console.log(emp);

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

Javascript has three type of pop up boxes

1-alert

2-confirm

3-prompt

* What is the use of Void (0)?

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.

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

We can use ***[window.location](https://www.geeksforgeeks.org/javascript-window-location-and-document-location-objects/)*** 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.

**Syntax:**

<script>

window.location = <Path / URL>

</script>

* What are the disadvantages of using innerHTML in JavaScript?

Disadvantages of innerHTML

* Event handlers attached to any DOM element are preserved.
* Replacement is done everywhere.
* It is not possible to append innerHTML.
* Breaks the document.
* Used for Cross-site Scripting.