**1] What is JavaScript?**

=> Java script was designed to add interacity to HTML pages.

Java script is a Scriting Language.

Java script is usually emmeded directly into HTML pages.

**2] What is the use of isNaN function?**

=> The JavaScript isNaN () Function is used to check whether

a given value is an illegal number or not.

It returns true if the value is a NaN else returns false.

It is different from the Number.isNaN () Method.

**3] 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.

JavaScript shows the NEGATIVE\_INFINITY value as -Infinity.

**4] 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.

**5] 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.

If we use ‘typeof’ operator to get the value of an undeclared variable, we will face the runtime error with the return value as “undefined”.

The scope of the undeclared variables is always global.

**6] Write the code for adding new elements dynamically?**

=> 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.

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

**7] What is the difference between ViewState and SessionState?**

=> SessionState : Can be persisted in memory, which makes it a

fast solution.

Which means state cannot be shared in the Web Farm/Web Garden.

Can be persisted in a Database, useful for Web Farms / Web Gardens.

Is Cleared when the session dies - usually after 20min of inactivity.

ViewState : Is sent back and forth between the server and

client,taking up bandwidth.

Has no expiration date.

Is useful in a Web Farm / Web Garden

**8] What is === operator?**

=> An operator is a sign or symbol that specifies the type

of calculation to perform within an expression.

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

=> 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 approaches that allow us to achieve this task.

**10] How to read and write a file using JavaScript?**

=> 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.

Syntax:

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

(2) fs.writeFile( file\_name, data, options, callback )

**11] What are all the looping structures in JavaScript?**

=> Loops offer a quick and easy way to do something repeatedly. This chapter of the JavaScript Guide introduces the different iteration

statements available to JavaScript.

You can think of a loop as a computerized version of the game where you tell someone to take X steps in one direction, then Y steps in another. For example, the idea "Go five steps to the east" could be expressed this way as a loop:

for (let step = 0; step < 5; step++)

{

// Runs 5 times, with values of step 0 through 4.

console.log("Walking east one step");

}

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

=> For performing the above-illustrated task, we would be using a method (or a function) provided by JavaScript called as parseInt().

This is a special method, provided by JavaScript, that takes an integer value (of any base which is either specified or not) and further converts the string into an integer value.

String Integer

"1002" 1002

**13] What is the function of the delete operator?**

=> The delete operator removes a property from an object.

If the property's value is an object and there are no more references to the object, the object held by that property is eventually released automatically.

Syntax: delete object.property

delete object[property]

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

=> Alert Box: window.alert("sometext");

Confirm Box: window.confirm("sometext");

Prompt Box: window.prompt("sometext","defaultText");

Line Breaks: alert("Hello\nHow are you?");

**15] What is the use of Void (0)?**

=> “void (0)”, also seen as “void 0”. It’s a quirky command; one whose purpose many aren’t entirely clear on.

To confuse things even more, there’s also an error message: “javascript:void(0)” that is basically unrelated.

We’ll talk about both to clear up the confusion. Let’s see if we can clear things up a bit. Shall we begin?

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

=> 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.

Syntax: <script>

window.location = <Path / URL>

</script>

**17] What are the disadvantages of using innerHTML in JavaScript?**

=> The use of innerHTML very slow: The process of using innerHTML is much slower as its contents as slowly built, also already parsed contents

and elements are also re-parsed which takes time.

Preserves event handlers attached to any DOM elements: The event handlers do not get attached to the new elements created by setting

innerHTML automatically.

To do so one has to keep track of the event handlers and attach it to new elements manually.

This may cause a memory leak on some browsers.

Content is replaced everywhere: Either you add, append, delete or modify contents on a webpage using innerHTML, all contents is replaced,

also all the DOM nodes inside that element are reparsed and recreated.

Appending to innerHTML is not supported: Usually, += is used for appending in JavaScript.

But on appending to an Html tag using innerHTML, the whole tag is re-parsed.

example:<p id="geek">Geeks</p>

title = document.getElementById('#geek')

// The whole "geek" tag is reparsed

title.innerHTML += '<p> forGeeks </p>'