* **What is JavaScript?**

Javascript is the programming language for the web. JavaScript can update and change both HTML and CSS. JavaScript can calculate, manipulate and validate data.

* **What is the use of is NaN function?**

The is NaN() function is used to check whether a given value is an illegal number or not. It returns true if value is a NaN else returns false.

* **What is negative Infinity?**

Negative\_Infinity is a property of a JavaScript Number object.

You can only use it as Number.NEGATIVE\_INFINITY.

The negative infinity in JavaScript is a constant value which is used to represent a value which is the lowest available. This means that no other number is lesser than this value.

* **Which company developed Javascript?**

The first ever JavaScript was created by Brendan Eich at Netscape, and has since been updated to conform ECMA-262 Edition 5 and later versions.

* **What are undeclared and undefined variables?**

**Undeclared**: It occurs when we try to access any variable that is not initialized or declared earlier using var or const keyword. If we use ‘typeof’ operator to get the value of an undeclared variable, we will face the runtime error with return value as “undefined”. The scope of the undeclared variables is always global.

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

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

<!DOCTYPE html>

<html>

<body>

<h1>JavaScript Variables</h1>

<p>In this example, a, b, and c are variables.</p>

<p id="demo"></p>

<script>

var a = 5;

var b = 6;

var c = a + b;

document.write("The value of sum is: " + c);

</script>

</body>

</html>

Output:

**JavaScript Variables**

In this example, a, b, and c are variables.

The value of sum is: 11

* **What is the difference 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. |
| View state values are lost/cleared when new page is loaded. | Session state can be cleared by programmer or user or in case of timeouts. |
| Used to store information that you wish to access from same web page. | Used to store information that you wish to access on different web pages. |

* **What is === operator?**

=== (Triple equals) operator is a strict equality comparison operator in Javascript, which returns false for the values which are not of a similar type.

This operator performs type casting for equality.

If we compare 2 with “ 2 ” using ===, then it will return a false value.

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

**Change Style:**

<html>

<head>

<script>

function myfunction()

{

document.getElementById(‘demo’).style.fontSize=”50px”;

document.getElementById(‘demo’).style.color=”red”; document.getElementById(‘demo’).style.backgroundColor= yellow”;

}

</script>

</head>

<body>

<input type=”submit” name=”submit” value=”click here” onclick=”myfunction()”>

<h1 id=”demo” style=”color: blue”> This is style given by Javascript function</h1>

</body>

</html>

**Change Class property:**

<!DOCTYPE html>

<html>

   <head>

     <style>

         .colorBlue

{

           color: blue;

          }

         .colorRed

{

             color: red;

         }

     </style>

</head>

<body>

     <h3>className Example</h3>

      <p class="colorBlue">GeeksforGeeks is a computer science portal

        for geeks.This platform has been designed for every geek wishing to expand the knowledge, share their knowledge and is ready to grab their dream job. GFG have millions of articles, live as well as online courses, thousands of tutorials and much more just for the geek inside you.</p>

     <button id="submit">Change Color</button>

     <script>

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

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

         btn.addEventListener("click", function ()

{

             para.className = "colorRed";       });

     </script>

</body>

</html>

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

The fs.readFile() and 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.

var fs = require("fs");

console.log(" Writing into an file ");

// Sample.txt is an empty file

fs.writeFile(

   "sample.txt",

   "Let's write a few sentences in the file",

  function (err)

{

     if (err)

{

        return console.error(err);

     }

     // If no error the remaining code executes

     console.log(" Finished writing ");

     console.log("Reading the data that's written");

     // Reading the file

     fs.readFile("sample.txt", function (err, data)

{

       if (err)

{

         return console.error(err);

       }

       console.log("Data read : " + data.toString());

    });

  }

);

* **What are all the looping structures in the Javascript?**

Loops can execute a block of code a number of times.

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?**

The parseInt method parses a value as a string and returns the first Integer.

<!DOCTYPE html>

<html>

<body>

<h2>The parseInt() Method</h2>

<p>parseInt() parses a string and returns the first integer:</p>

<p id="demo"></p>

<script>

document.getElementById("demo").innerHTML =

parseInt("10") + "<br>" +

parseInt("10.00") + "<br>" +

parseInt("10.33") + "<br>" +

parseInt("34 45 66") + "<br>" +

parseInt(" 60 ") + "<br>" +

parseInt("40 years") + "<br>" +

parseInt("He was 40");

</script>

</body>

</html>

Output:

## The parseInt() Method

parseInt() parses a string and returns the first integer:

10  
10  
10  
34  
60  
40  
NaN

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

Delete operator deletes both the value of the property and the property itself. The delete operator is designed to be used on object properties.

<!DOCTYPE html>

<html>

<body>

<p id="demo"></p>

<script>

var name ="John";

var age=50;

delete person.age;

document.getElementById("demo").innerHTML =

person.firstname + " is " + person.age + " years old.";

</script>

</body>

</html>

Output:

John is undefined years old.

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

JavaScript has three kind of popup boxes: Alert box, Confirm box, and Prompt box.

**Alert box:**An alert box is often used if you want to make sure information comes through to the user.

When an alert box pops up, the user will have to click "OK" to proceed.

Syntax: window.alert("*sometext*");

**Confirm box**: When a confirm box pops up, the user will have to click either "OK" or "Cancel" to proceed.

If the user clicks "OK", the box returns **true**. If the user clicks "Cancel", the box returns **false**.

Syntax: window.confirm("*sometext*");

**Prompt box:** A prompt box is often used if you want the user to input a value before entering a page.

When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value.

Syntax: window.prompt("*sometext*","*defaultText*");

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

JavaScript void 0 means returning undefined (void) as a primitive value.  It is used to prevent any side effects caused while inserting an expression in a web page.

This will prevent the browser from opening a new or reloading the web page and allowing you to call the JavaScript through it.

* **How can a page 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.

<!DOCTYPE html>

<html lang="en">

<body>

     <h3>This is the original page</h3>

     <br>

     <button onclick="force\_load()">Force Load W3Schools Page</button>

<br>

<script>

         function force\_load()

{

             window.location = <https://www.w3schools.com/>

}

 </script>

</body>

</html>

* **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.

**- 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. 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.