JAVASCRIPT BASIC & DOM

Q.1:- What is JavaScript?

Ans:- avaScript is a scripting or programming language that allows you to implement complex features on web pages — every time a web page does more than just sit there and display static information for you to look at — displaying timely content updates, interactive maps, animated 2D/3D graphics, scrolling video jukeboxes, etc. — you can bet that JavaScript is probably involved. It is the third layer of the layer cake of standard web technologies, two of which ([HTML](https://developer.mozilla.org/en-US/docs/Learn/HTML) and [CSS](https://developer.mozilla.org/en-US/docs/Learn/CSS)) we have covered in much more detail in other parts of the Learning Area.

Q.2:- What is the use of is NaN function?

Ans:- In Javascript NaN stands for **"Not-a-Number"**. Hence here, isNaN() is a method in Javascript used to check whether a value is a number or not.

The method isNaN in JavaScript returns true if the given value is Not-a-Number, else it returns false.

**Example:**

console.log(isNaN('51'))

console.log(isNaN(false))

console.log(isNaN('abc'))

console.log(isNaN('17.43'))

console.log(isNaN(new Date()))

false

false

true

false

false

Q.3:- What is negative Infinity?

Ans:-

Negative infinity is a number in java script, which is derived by 'dividing negative number by zero'.  
- A number object needs not to be created to access this static property.  
- The value of negative infinity is the same as the negative value of the infinity property of the global object.

Q.4:- Which company developed JavaScript?

Ans:- avaScript was created at Netscape Communications by Brendan Eich in 1995. Netscape and Eich designed JavaScript as a scripting language for use with the company's flagship web browser, Netscape Navigator.

Q.5:- What are undeclared and undefined variables?

Ans:-

**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*.

**Example:- Undeclared:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        function Dfs() {

 // 'use strict' verifies that no undeclared

 // variable is present in our code

 'use strict';

 x = "GeeksForGeeks";

}

Dfs(); // Accessing the above function

    </script>

</body>

</html>

**Example 2:- undefined:**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

        function checkVar() {

    let string;

    if (typeof variable === "undefined") {

        string = "Variable is undefined";

    } else {

        string = "Variable is defined";

    }

    console.log(string);

}

checkVar();

    </script>

</body>

</html>

Q.6:- Write the code for adding new elements dynamically?

Ans:- New elements can be created in JS by using the **createElement()** method.

Exm.:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        html,

        body {

            height: 100%;

            width: 100%;

        }

        .button {

            display: flex;

            align-items: center;

            justify-content: center;

        }

        .tasks {

            display: flex;

            justify-content: center;

            align-items: center;

            flex-direction: column;

            margin-top: 20px;

        }

    </style>

</head>

<body>

    <div class="button">

        <button id="addTask">Add task</button>

    </div>

    <div class="tasks"></div>

    <script type="text/javascript">

        // Getting the parent element in which

        // the new div will be created

        let task = document.getElementsByClassName("tasks");

        // Getting the addTask button element

        let addTask = document.getElementById("addTask");

        // Adding onclick event to the button

        addTask.addEventListener('click', function () {

            // Traversing through collection of HTML

            // elements (tasks here)

            for (let i = 0; i < task.length; i++) {

                // New div element is created

                let newDiv = document.createElement("div");

                // Setting the attribute of class type to newDiv

                newDiv.setAttribute("class", "list");

                // innerText used to write the text in newDiv

                newDiv.innerText = "New Div created";

                // Finally append the newDiv to the

                // parent i.e. tasks

                task[i].append(newDiv);

            }

        })

    </script>

</body>

</html>

Q.7:-What is the difference between ViewState and SessionState?

Ans:-

* **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.

Q.8:- What is === operator?

Ans:- **strict equality (===)** operator checks whether its two operands are equal, returning a Boolean result. Unlike the [equality](https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Operators/Equality) operator, the strict equality operator always considers operands of different types to be different.

**Exmp.:-**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <Script>

        "hello" === "hello"; // true

"hello" === "hola"; // false

3 === 3; // true

3 === 4; // false

true === true; // true

true === false; // false

null === null; // true

    </Script>

</body>

</html>

**Q.9:-** How can the style/class of an element be changed?

Ans:- Changing CSS with the help of the style property:

Exmp:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1 style="color: green;">

        css/style 123

    </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>

       // document.getElementById("id").style.property = new\_style

       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>

Exmp 2:- **Changing the class itself**

* **he add() method:** It adds one or more classes.
* **The remove() method:**It removes one or more classes.
* **The toggle() method:**If the class does not exist it adds it and returns true. It removes the class and returns false. The second boolean argument forces the class to be added or removed.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        .hide {

            display: none;

        }

        .blueColor {

            color: blue;

        }

    </style>

</head>

<body>

    <h1 style="color: green;">

        GeeksforGeeks

    </h1>

    <h2>

         style/class of

        an element be changed?

    </h2>

    <h3>Hide and Show the Para</h3>

    <p>

        Lorem ipsum dolor sit amet consectetur adipisicing elit. Deleniti ad illum eum hic illo sit fugit porro, dolorum

        provident nisi maxime, saepe, dignissimos quas et doloremque quidem officia quae atque.

    </p>

    <button id="hide">Hide</button>

    <button id="show">Show</button>

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

    <script>

        const btn\_hide = document.getElementById("hide");

        const btn\_show = document.getElementById("show");

        const btn\_color = document.getElementById("color");

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

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

            para.classList.add("hide");

        });

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

            para.classList.remove("hide");

        });

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

            para.classList.toggle("blueColor");

        });

    </script>

</body>

</html>

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

Ans:- 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.

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

* **filename:** It contains the filename to be read, or the whole path if the file is saved elsewhere.
* **encoding:** It stores the file’s encoding. ‘utf8’ is the default setting.
* **callback function:**This is a function that is invoked after the file has been read. It requires two inputs:
* **err:** If there was an error.
* **data:**The file’s content.
* **Return Value:**It returns the contents contained in the file, as well as any errors that may have occurred.

Q.11:- What are all the looping structures in JavaScript?

Ans:- The **JavaScript loops** are used *to iterate the piece of code* using for, while, do while or for-in loops. It makes the code compact. It is mostly used in array.

There are four types of loops in JavaScript.

1. for loop
2. while loop
3. do-while loop
4. for-in loop

Q.12:- How can you convert the string of any base to an integer in JavaScript?

Ans:- we would be passing the string value in a method (which is explicitly declared for ease purpose) and further that string value is passed inside the parseInt() method which then further converts that string value in the corresponding integer value.

Exmp.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

    let stringConversion = (string\_value) => {

        console.log("Initial Type: " + typeof string\_value);

        let integer\_value = parseInt(string\_value);

        console.log("Final Type: " + typeof integer\_value);

        console.log(integer\_value);

      };

      stringConversion("512000");

      stringConversion("126410");

      stringConversion("0x8975");

    </script>

</body>

</html>

Q.13:- What is the function of the delete operator?

Ans:- The delete operator in JavaScript removes a property from an object. It can delete both own properties and inherited properties. Using delete on an array item leaves a hole.

Exmp.

.:- <!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <script>

    const Employee = {

  firstname: 'viaz',

  lastname: 'jain',

};

console.log(Employee.firstname);

// Expected output: "Maria"

delete Employee.firstname;

console.log(Employee.firstname);

// Expected output: undefined

//EXmp.2

let emp = {

    firstName: "Raj",

    lastName: "Kumar",

    salary: 40000

}

console.log(delete emp.salary);

console.log(emp);

    </script>

</body>

Q.14:- What are all the types of Pop up boxes available in JavaScript?

Ans:-

In Javascript, popup boxes are used to display the message or notification to the user.

There are three types of [pop-up boxes in JavaScript](https://www.geeksforgeeks.org/javascript-dialogue-boxes/):

* [Alert Box](https://www.geeksforgeeks.org/what-are-the-types-of-popup-box-available-in-javascript/#alert-box)
* [Prompt Box](https://www.geeksforgeeks.org/what-are-the-types-of-popup-box-available-in-javascript/#prompt-box)
* [Confirm Box](https://www.geeksforgeeks.org/what-are-the-types-of-popup-box-available-in-javascript/#confirm-box)

Exmp.:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

    <style>

        h1{

            color:green;

        }

    </style>

</head>

<body>

        <h1>hello js</h1>

        <h3>Alert Box</h3>

        <button onclick="hello()">

            Click here for alert box

        </button>

        <!-- Alert box function -->

        <script>

            function hello() {

                alert("An Online Computer Science"

                            + "Portal for Geeks");

            }

        </script>

</body>

</html>

Q.15:- What is the use of Void (0)?

Ans:- **Using “javascript:void(0);” in anchor tag:**Writing “javascript:void(0);” in anchor tag can prevent the page to reload and JavaScript functions can be called on single or double clicks easily.

Exmp.:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <h1 style="color:green">Hello js</h1>

    <h3>JavaScript:void(0)</h3>

    <a href="javascript:void(0);" ondblclick="js()">

        Double click on me </a>

    <p id="Hello">

    </p>

        <script>

             function js() {

        document.getElementById("Hello").innerHTML = "Welcome to the js";

    }

        </script>

</body>

</html>

Q.16:- How can a page be forced to load another page in JavaScript?

Ans:- The **replace** function is used to navigate to a new URL without adding a new record to the history.

Exmp.:-

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <title>Document</title>

</head>

<body>

    <button onclick="replaceLocation()">

        Replace current webpage

    </button>

    <script>

        function replaceLocation() {

            // Replace the current location

            // with new location

            let newloc = "https://www.amazon.in/?&tag=googhydrabk1-21&ref=pd\_sl\_7hz2t19t5c\_e&adgrpid=155259815513&hvpone=&hvptwo=&hvadid=676742245123&hvpos=&hvnetw=g&hvrand=13828803214090887706&hvqmt=e&hvdev=c&hvdvcmdl=&hvlocint=&hvlocphy=9061731&hvtargid=kwd-10573980&hydadcr=14453\_2367553&gad\_source=1";

            window.location.replace(newloc);

        }

    </script>

</body>

</html>

Q.17:- What are the disadvantages of using innerHTML in JavaScript?

Ans:-  [innerHTML property](https://www.geeksforgeeks.org/html-dom-innerhtml-property/" \t "_blank) is a part of the Document Object Model (DOM) that is used to set or return the HTML content of an element.

* **The use of innerHTML very slow:**
* **Preserves event handlers attached to any DOM elements:**
* **Content is replaced everywhere:**
* **Can break the document:**

**EXmp.:-**

<!DOCTYPE html>

<html>

<head>

    <title>

        Using innerHTML in JavaScript

    </title>

</head>

<body style="text-align: center">

    <h1 style="color:green">

        hello JavaScript

    </h1>

    <p id="P">

        A computer science

         i js.

    </p>

    <button onclick="js()">

        Try it

    </button>

    <p id="p"></p>

    <script>

        function js() {

            var x = document.getElementById("P")

                        .innerHTML;

            document.getElementById("p")

                        .innerHTML = x;

            document.getElementById("p")

                        .style.color = "green";

        }

    </script>

</body>

</html>