**Module (JAVASCRIPT BASIC & DOM)-4**

**(1)What is javascript?**

* Ans.:- JavaScript is a scripting language that enables you to create dynamically updating content, control multimedia, animate images, and pretty much everything else.

Javascript is used to make any website dynamic and interactive. It is the most used programming language in the world

**(2) What is the use of isNaN function?**

* **Ans.:-**
* In javaScript Nan is short for “Not-a-Number”.
* The is Nan() method returns true if a value is NaN.
* The is NaN() method converts the value to a number before testing it

**(3) What is negative Infinity?**

* **Ans.:-** The Negative infinity is a constant value that is used to represent a value lowest of all. This means there is no other value lesser than this value. Negative infinity is a special numeric value that is returned when arithmetic operation or mathematical function generates a negative value greater than the largest representable number in javascript.
* **Example:-**
* **Input:-**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<title>Document</title>

</head>

<body>

<!-- CSS STYLE START -->

<style>

h1 {

color: green;

}

</style>

<!-- CSS STYLE END -->

<!-- HTML START -->

<h1>web designing</h1>

<h1>

What is negative infinity in JavaScript?

</h1>

<button onclick="TopsNegativeInfinity()">

Generate negative infinite

</button>

<p id="Tops"></p>

<!-- HTML END -->

<!-- JAVASCRIPT START -->

<script>

function TopsNegativeInfinity() {

//negative value greater than the

//largest representable number in JavaScript

var n = (-Number.MAX\_VALUE) \* 2;

document.getElementById("Tops").innerHTML = n;

}

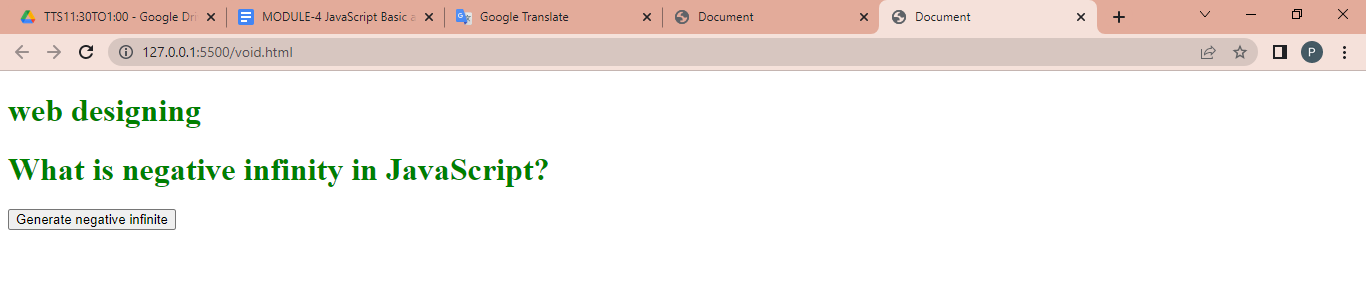
</script>

<!-- JAVASCRIPT END -->

</body>

</html>

* **Output:-**

****

**(4)** **Which company developed JavaScript?**

* **Ans.:-** Javascript was initially known as LiveScript. It was developed by Netscape in 1995. Netscape was the company which developed world’s full fledged web browser and the creator was brendon Eich.

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?**

* **Ans.:-**
* **Undeclared:** A variable is undeclared if it has not been declared with an appropriate keyword (i.e. var, let or const). Accessing an undeclared variable will throw a ReferenceError.
* **Example:-**
* <script>
* console.log(A);
* // ReferenceError: A is not defined
* </script>
* **Undefined:**  A variable is undefined if it hasn't been assigned a value. undefined is a primitive data type in JavaScript and represents the absence of a value, intentional or otherwise.
* **Example:-**
* <script>
* let A;
* console.log(A);
* // undefined
* </script

**(6)** **Write the code for adding new elements dynamically?**

* **Ans.:-**

<!DOCTYPE html>

<html>

<head>

<meta charset=”UTF-8”>

<meta Http-equiv=”X-UA-Compatible” content=”IE=edge”>

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

<title>Adding New Elements</title>

</head>

<script type="text/javascript">

function addNode() { var newP = document.createElement("p");

var textNode = document.createTextNode(" This is a new text node");

newP.appendChild(textNode); document.getElementById("firstP").appendChild (newP); }

</script>

</head>

<body> <p id="firstP">firstP<p> </body>

</html>

**(7)** **What is the difference between ViewState and SessionState?**

* **Ans.:-**

|  |  |
| --- | --- |
| * **ViewState** | * **SessionState** |
| * Maintained at page level only | * Maintained at session level. |
| * View state can only be visible from a single page and not multiple page. | * Session state value availability is in all pages available in a user session. |
| * Information stored on the client’s end only. | * Information in session state stored in the server. |
| * View state will retain values in the event of a postback operation occurring. | * In session state, user data remains in the server. The availability of the data is guaranteed until either the user closes the session or the browser is closed. |
| * View state is used to allow the persistence of pageinstance –spacific data. | * Session state is used for the persistence of user-specific data on the server’s end. |

**(8)** **What is === operator?**

* **Ans.:-** In Javascript === is strict equality operator. Used to compare two variable and check both value and datatype. If both datatypes and value matches of two variables it will return Boolean result (True or False).
* **Example:-**
* **Input:**

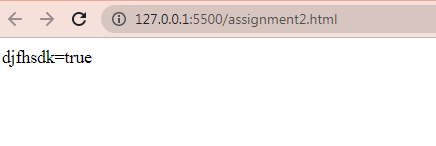
<script>

Let a = 10

Document.write (“dfhkfh=”, a===10)

</script>

* Output:-



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

**Ans.:-**

* **Style:-** JavaScript can change Css styles such as color, font size etc. of elements using some methods such as getElementById( ), getElementByClassName( ) etc.
* **Class:-** Use the classList Property

In modern browsers you can use the DOM element's classList property to add, remove or toggle CSS classes to the HTML elements dynamically with JavaScript.

* **Example:-**
* **Input:-**

<!-- CSS style start -->

<style>

.one{

color: red;

}

.colorblack{

color: chartreuse;

}

</style>

<!-- CSS style start -->

</head>

<body>

<p class="one">Lorem, ipsum dolor sit amet consectetur adipisicing elit. Distinctio, vero?</p>

<button id="click">click</button>

<!-- javascript start -->

<script>

const button=document.getElementById('click');

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

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

para.className="colorblack"

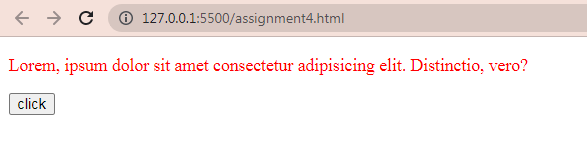
})

</script>

<!-- javascript start -->

</body>

* **Output:-**

****

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

* **Ans.:-**

After the File System module is imported, the reading of the file in JavaScript can be done by using the readFile() function.

Syntax

The syntax to read from a file is as follows −

readFile(path, format, callBackFunc)

The readFile() function accepts three parameters including one optional parameter.

* **Path** − The first parameter is the path of the test file from which the contents are to read. If the current location or directory is the same directory where the file which is to be opened and read is located then, only the file name has to be given.
* **Format** − The second parameter is the optional parameter which is the format of the text file. The format can be ASCII, utf-8 etc.
* **CallBackFunc** − The third parameter is the call back function which takes the error as the parameter and displays the fault is any raised due to the error.

Example:- const fs = require('fs')

fs.readFile('tp.txt', (err, inputD) => {

   if (err) throw err;

      console.log(inputD.toString());

})

* **Write operation on a file**

After the File System file is imported then, the writeFile() operation is called. The writeFile() method is used to write into the file in JavaScript. The syntax of this method is as follows −

riteFile(path,inputData,callBackFunction)

The writeFile() function accepts three parameters –

* **Path** − The first parameter is the path of the file or the name of the file into which the input data is to be written.

If there is a file already, then the contents in the file are deleted and the input which is given by the user will get updated or if the file is not present, then the file with that will be created in the given path and the input information is written into it.

* **inputData** − The second parameter is the input data which contains the data to be written in the file that is opened.
* **callBackFuntion** − The third parameter is the function which is the call back function which takes the error as the parameter and shows the fault if the write operation fails.

Example:- const fs = require('fs')

let fInput = "You are reading the content from Tutorials Point"

fs.writeFile('tp.txt', fInput, (err) => {

   if (err) throw err;

   else{

      console.log("The file is updated with the given data")

   }

})

**(11)** **What are all the looping structures in JavaScript?**

* **Ans.:-**  javascript supports different kinds of loops:

1. For :-loops through a block of code a number of times.
2. For/in :- loops through the properties of an object.
3. For/of :- loops though the values of an iterable object.
4. While :- loops though a block of code while a specified condition is true.
5. Also loops through a block of code while a specified condition is true**.**

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

* **Ans.:-**

Users can use functions in JavaScript to convert a string into an integer. There are many ways to convert a string into an integer value. One is by using JavaScript functions like Number( ), parseFloat( ), parseInt( ).

**(13)** **What is the function of the delete operator?**

* **Ans.:-** The delete operator in JavaScript is used to delete an object’s property.

In JavaScript, the delete operator is the only way to remove properties from an object. When you use delete, it’ll return true when it removes a property and false otherwise. The delete operator shouldn’t be used on predefined JavaScript object properties.

* **Example:-**
* **Input:-**

<body>

<script>

let user = {

firstName: "paresh",

lastName: "solanki",

salary :30000

}

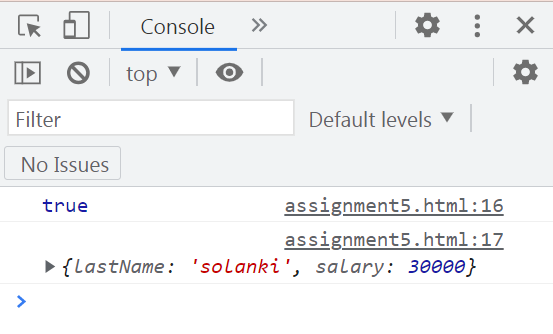
console.log (delete user.firstName);

console.log(user);

</script>

</body>

* **Output:-**

****

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

* **Ans.:-**
* **Alert Box:** The alert( ) function displays a message to the user to display some information to users. This alert box will have the OK button to close the alert box.
* **Example:-**
* **Input:-**

<body>

<button onclick="a()">paresh</button>

<p id="one"></p>

<script>

// alert box start

function a() {

alert("fhfjhjh")

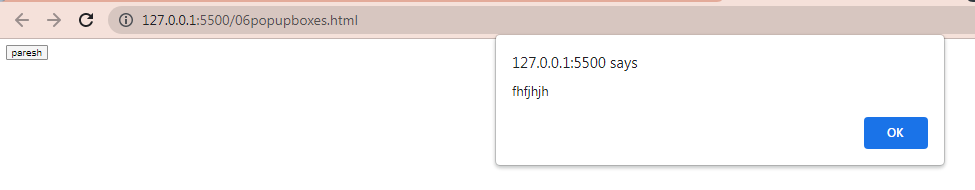
}

// alert box End

</script>

</body>

* **Output:-**

****

* **Confirm Box:** Use the confirm( ) function to take the user's confirmation before starting some task. For example, you want to take the user's confirmation before saving, updating or deleting data.
* **Example:-**
* **Input:-**

<body>

<button onclick="a()">paresh</button>

<p id="one"></p>

<script>

// confirm box start

function a() {

var txt;

if(confirm("sdjvksjk")){

txt = "knmd"

}

else{

txt = "dkfn"

}

document.getElementById("one").innerHTML=txt

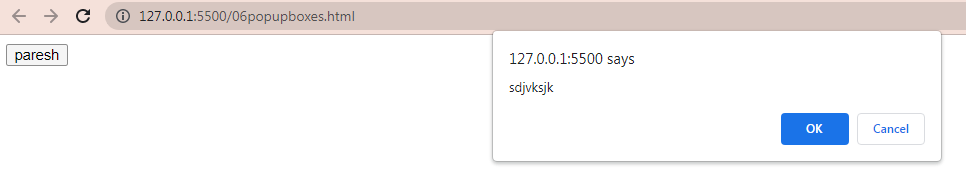
}

// confirm box End

</script>

</body>

* **Output:-**

****

* **Prompt Box:** Use the prompt() function to take the user's input to do further actions. For example, use the prompt() function in the scenario where you want to calculate EMI based on the user's preferred loan tenure.
* **Example:-**
* **Input:**

<body>

<button onclick="a()">paresh</button>

<p id="one"></p>

<script>

// prompt box start

function a() {

var txt;

var person = prompt("Enter your name","Tops")

if(person == null || person == ""){

txt ="plz enter your name"

}

else{

txt="hii" +person+ "how are you "

}

document.getElementById("one").innerHTML=txt

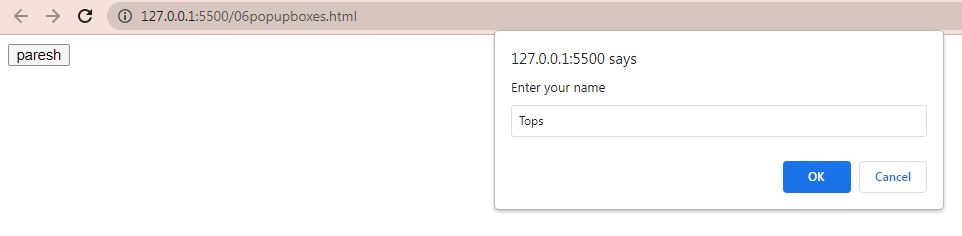
}

// prompt box End

</script>

</body>

* **Output:**

****

**(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
* **Example:-**
* **Input:-**

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

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

<title>Document</title>

</head>

<body>

<center>

<h1 style="color:green">d,dj,vn,jvn </h1>

<h3>JavaScript:void(0)</h3>

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

ondblclick="alert('Welcome to the my web page')">

Double click

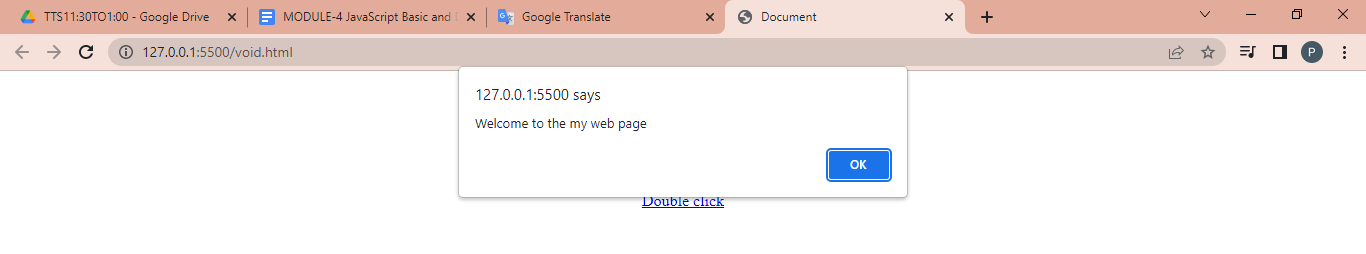
</a>

</center>

</body>

</html>

* **Output:-**

****

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

* **Ans.:-** 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.

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

* **Ans.:-**
* If we use innerHTML then the entire innerHTML content is re-parsed and build into elements. Therefore it's much slower
* The process of using inner HTML is much slower than the rest of the variables as its content is slowly built into different elects and takes time to get re-parsed.
* While using the inner HTML, the content gets replaced in JavaScript.
* Appending to inner HTML can't be used properly.