**Java script Basic &Amp; Dom**

• What is JavaScript?

ANS-

Java Script is a lightweight, interpreted programming language.

It is complimentary to and integrated with Java.

JavaScript is very easy to implement because it is integrated with HTML.

It is open and cross-platform.

every modern web browser

For example:- Chrome, Mozilla Firefox, Safari supports Java script.

Java script helps you create really beautiful and crazy fast and dynamic websites.

• What is the use of is NaN 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.

• What is negative Infinity?

ANS:-

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

    <button onclick="TopsNegativeInfinity()">

      Generate negative infinite

  </button>

  <!-- button end -->

    <p id="Tops"></p>

<script>

        function TopsNegativeInfinity() {

            //negative value greater than the

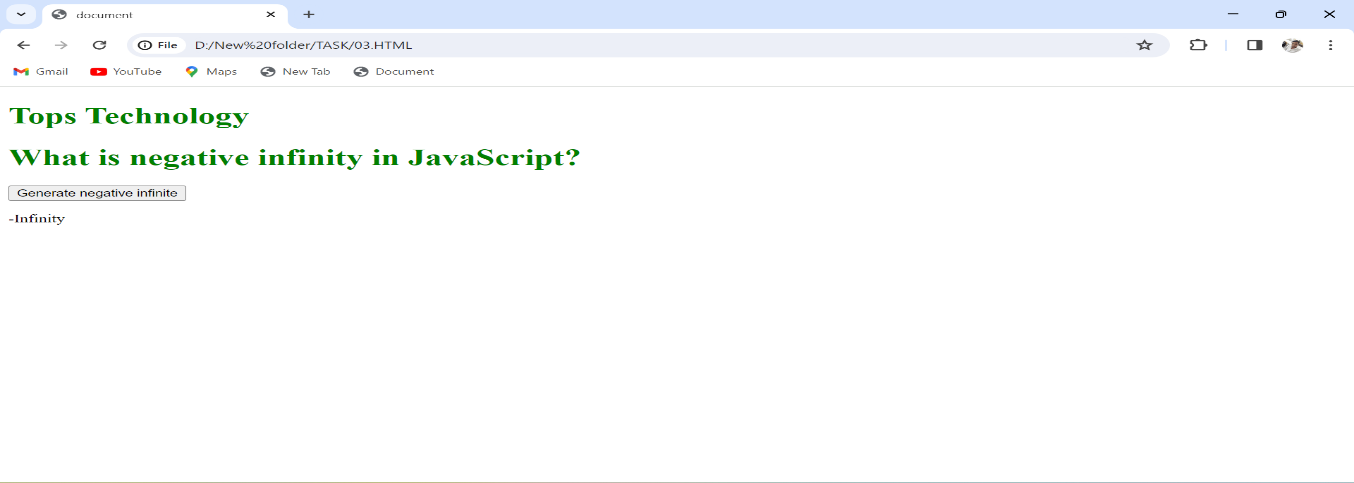
            //largest representable number in JavaScript

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

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

        }

    </script>

OUTPUT:- 

• Which company developed JavaScript?

ANS:- JavaScript is a scripting language developed by Netscape.

• What are undeclared and undefined variables?

**Ans.**

>Undefined :- variable means a variable has been declared but does not have a value.

Undefined Example

var dog;

console.log(dog);

Output:

Undefined

>Undeclared :- variable means that the variable does not exist in the program at all.

>Undeclared Example

console.log(cat);

ReferenceError: cat is not defined

• Write the code for adding new elements dynamically?

Ans:-

<body id="body">

    <h1 style="text-align:center; color:green;">

        Top Technology

    </h1>

    <!-- Form to add item -->

    <form action="#" style="text-align: center;">

        <!-- Type of element -->

        <label for="type">

            Add Element Type

        </label>

        <input type="text" id="type" placeholder="Like: div, h1, li...." value="h3" />

        <br /><br />

        <!-- Text/value for element --->

        <label for="value"> Add Element Value</label>

        <input type="text" id="value" placeholder="Like: Hello Top Technology" value="Hello Top Technology" />

        <br /><br />

        <!-- Add attributes for element --->

        <label for="attr"> Add Attributes</label>

        <form id="attr">

            <label for="attrType">Add Attribute Type</label>

            <input type="text" style="width:240px;" placeholder="forexample: enter

                            'class' without quotes" id="attrType" value="class" />

            <br />

            <br />

            <label for="attrValue">

                Add Attribute Value

            </label>

            <input style="width:240px;" type="text" placeholder="for example: enter

                            'child' without quotes" id="attrValue" value="child" />

        </form>

        <br /><br />

        <!-- Submit the form -->

        <button type="button" onClick="addItem()">

            Add

        </button>

    </form>

    <div>

        <p>CHILD 1</p>

    </div>

    <script>

        // Define the addItem function

        // to be called through onclick()

        function addItem() {

            // Get Elements by id of the form inputs

            let parent =

                document.getElementById("body");

            let attrType =

                document.getElementById("attrType");

            let attrValue =

                document.getElementById("attrValue");

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

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

            if (type.value == "" || value.value == "") {

                window.alert(

                    "There is an error in form input");

                window.reload();

            }

            // createElement() method is used

            // to create a new element

            type = document.createElement(type.value);

            // Append a text node for this example

            type.appendChild(

                document.createTextNode(value.value));

            if (attrValue.value == ""

                    || attrType.value == "") {

                attr = null;

            }

            else {

                // setAttribute() is used to set

                // the attributes of the element

                type.setAttribute(

                    attrType.value, attrValue.value);

            }

            // Append as child to the parent

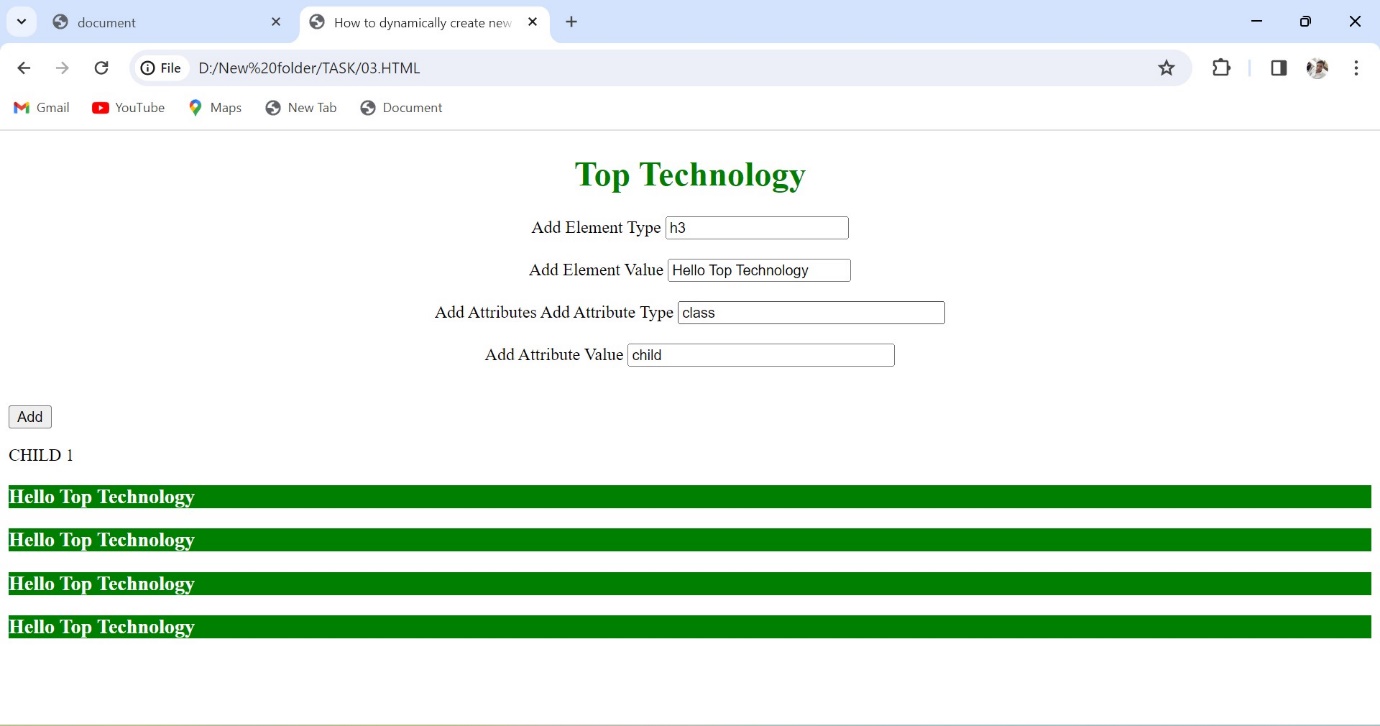
            // i.e. body

            parent.appendChild(type);

        }

    </script>

</body>



• What is the difference between ViewState and SessionState?

Ans:- **Ans.**

|  |  |
| --- | --- |
| **View State** | **Session State** |
| 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 post back 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. |
| 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. |

• What is === operator?

Ans:- Ans. The strict equality operator (===) checks whether its two operands are equal, returning a Boolean result. Unlike the equality operator, the strict equality operator always considers operands of different types to be different.

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

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

**Syntax:**

document. getElementById("id"). style. Property = new style

**Example:** In this example, we have built a PAN number validator. First, we will take the input value and match it with a regex pattern. If it matches then using JavaScript add an inline style on the <p> tag. Otherwise, add a different style on the <p> tag.

• How to read and write a file using JavaScript?

**Ans.** **Reading from the file**

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:

Following example tries to read the contents of the file populate in the previous example and print it −

const fs = require('fs')

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

   if (err) throw err;

      console.log(inputD.toString());

})

Output

Following is the output of the above example −

You are reading the content from Tutorials Point

The text which is displayed in the console is the text which is in the given file.

**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 −

writeFile(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:

Following is an example of the write operation in files in JavaScript.

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")

   }

})

• What are all the looping structures in JavaScript?

Ans:- **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:-loop through write one time and after check condition.

• How can you convert the string of any base to an integer in JavaScript?

Ans:- Given a string containing an integer value and along with that user passes a base value. We need to convert that string of any base value to an integer in JavaScript.

String          Integer

"1002"            1002

For ex:-

<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>

Output:

Initial Type: string

Final Type: number

512000

Initial Type: string

Final Type: number

126410

Initial Type: string

Final Type: number

35189

• What is the function of the delete operator?

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

If it is used to delete an object property that already exists, it returns true and removes the property from the object. However, deleting an object property that doesn’t exist will not affect the object, but will still return true.

The only time false will be returned is when the delete operator is used to delete a variable or a function.

Syntax

The syntax for using the delete operator is as follows:

delete object.property;  
// OR  
delete object["property"];

Parameters

**object**: This is the object whose property we want to delete.

**property**: This is the property to be deleted.

Return value

The delete operator returns true if the specified property is deleted, or false if the property is not deleted.

Code

In the code below, an object is created and the delete operator is used to delete some of its properties:

let human = {

    name: "John Doe",

    age: 15,

    country: "Nigeria"

}

let dog = {

    name: "Buddy",

    age: 2,

    country : "Germany"

}

// log retured values after delete

console.log(delete human["country"]) // same as human.country

console.log(delete dog.country) // same as dog["country"]

// log affected objects

console.log(human)  // "country" property deleted

console.log(dog)    // "country" property deleted

Output

true

true

{ name: 'John Doe', age: 15 }

 { name: 'Buddy', age: 2 }

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

Ans:- There are three types pop up box in java script

**1.Alert Box**,

**2.Confirm Box**

**3.Prompt Box**.

**1.Alert Box**:-

<script>

**only pop up box with ok button**

 function display()

 {

    alert('Login success');

    window.location="confirm.html";

 }

</script>

</head>

<body>

<input type="submit" value="alert box" onclick="display()">

</body>

**2.Confirm Box** :-

<script>

**only pop up box wih ok and cancel button**

    function confirm\_fun()

    {

        var ans=confirm("Are you sure want to pay....");

        if(ans==true)

        {

            alert("payment success");

            window.location='alert.html';  // page redirect

            return true;

        }

        else

        {

            alert("transaction Failed");

            return false;

        }

    }

</script>

</head>

<body>

<input type="submit" value="confirm box"  onclick="confirm\_fun()">

</body>

**3.Prompt Box**.

<script>

**only pop up box with ok and cancel and with textbox**

    function prompt\_fun()

    {

        var value=prompt("enter your code");  // input text value

        if(value=="1234")

        {

            window.location='alert.html';  // page redirect

            return true;

        }

        else

        {

            return true;

        }

    }

</script>

</head>

<body>

<input type="submit" value="Propmt box"  onclick="prompt\_fun()">

</body>

• What is the use of Void (0)?

**Ans. Using “javascript:void(0);” in anchor tag…**

**>** void is an important keyword in JavaScript which can be used as a

unary operator that appears before its single operand,

which may be of any type

> This operator specifies an expression to be evaluated without returning a value.

For ex:-

<script>

    const hello = () => {

        document.write('hello');

    }

</script>

<a href="" onclick="hello()">

    It will do nothing.

</a>

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

    It will do nothing.

</a>

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

Ans:-

**Step 1:** Create a file named ***index.html***. Add a heading and two buttons to it. One button forcefully loads a page with a live URL and the other button loads a local HTML page. In the *<script>* tag we have two functions, one loads gfg home page, and the second loads a local HTML page using ***window.location*** property.

For ex:- index.html

<body>

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

    <br>

    <button onclick="force\_load\_top()">

        Force Load Top Page

    </button>

   <br><br>

    <button onclick="force\_load\_local()">

        Force Load Local HTML page

    </button>

    <script>

        // top technologies link page

        function force\_load\_top() {

            window.location =

                "<https://www.tops-int.com/>"

        }

        function force\_load\_local() {

            window.location =

                "F:/top/PageRedirect/newPage.html"

        }

    </script>

</body>

**Step 2:** Create a file named ***newPage.html***. This is the local HTML page that would be loaded by Java script.

…….Redirect this page…….

<!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> New Page </title>

</head>

<body>

    <!-- new loaded page -->

    <h3>This is the new loaded page</h3>

</body>

</html>

**Output:**

• What are the disadvantages of using innerHTML in JavaScript?

**Ans.**

**Disadvantages of using innerHTML property 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.

For ex:-

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

    <h1 style="color:green">

        Top Technology

    </h1>

    <p id="P">

        A computer science

        portal for tops.

    </p>

    <button onclick="tops()">

        Try it

    </button>

    <p id="p"></p>

    <script>

    // try it function start

        function tops() {

            var x = document.getElementById("P")

                        .innerHTML;

            document.getElementById("p")

                        .innerHTML = x;

            document.getElementById("p")

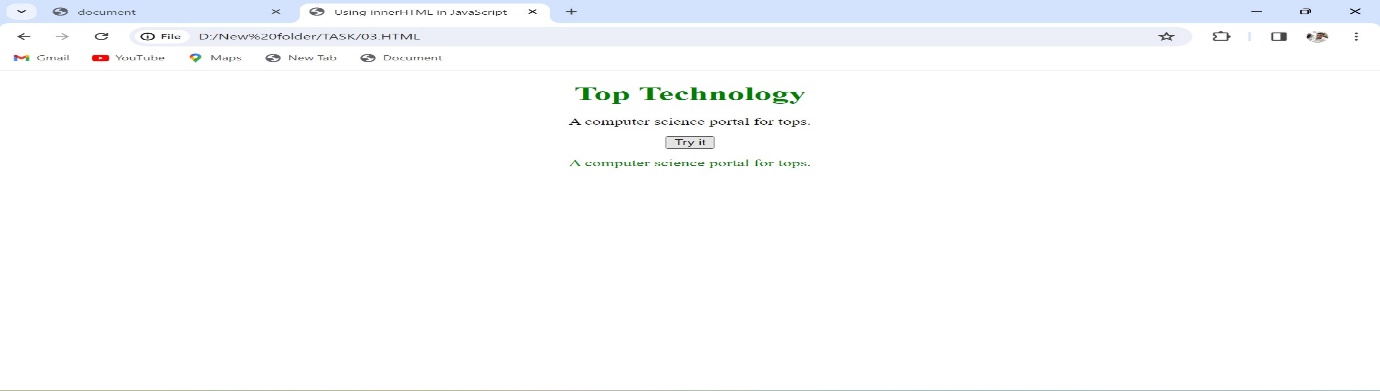
                        .style.color = "green";

        }

    </script>

</body>

**Output:-**



• Create password field with show hide functionalities

Ans:-

<body>

    <input type="password" id="one">

    <input type="checkbox" onclick="main()">

    <script>

        function main(){

            let a = document.getElementById("one");

            if(a.type === "password"){

                a.type = Text

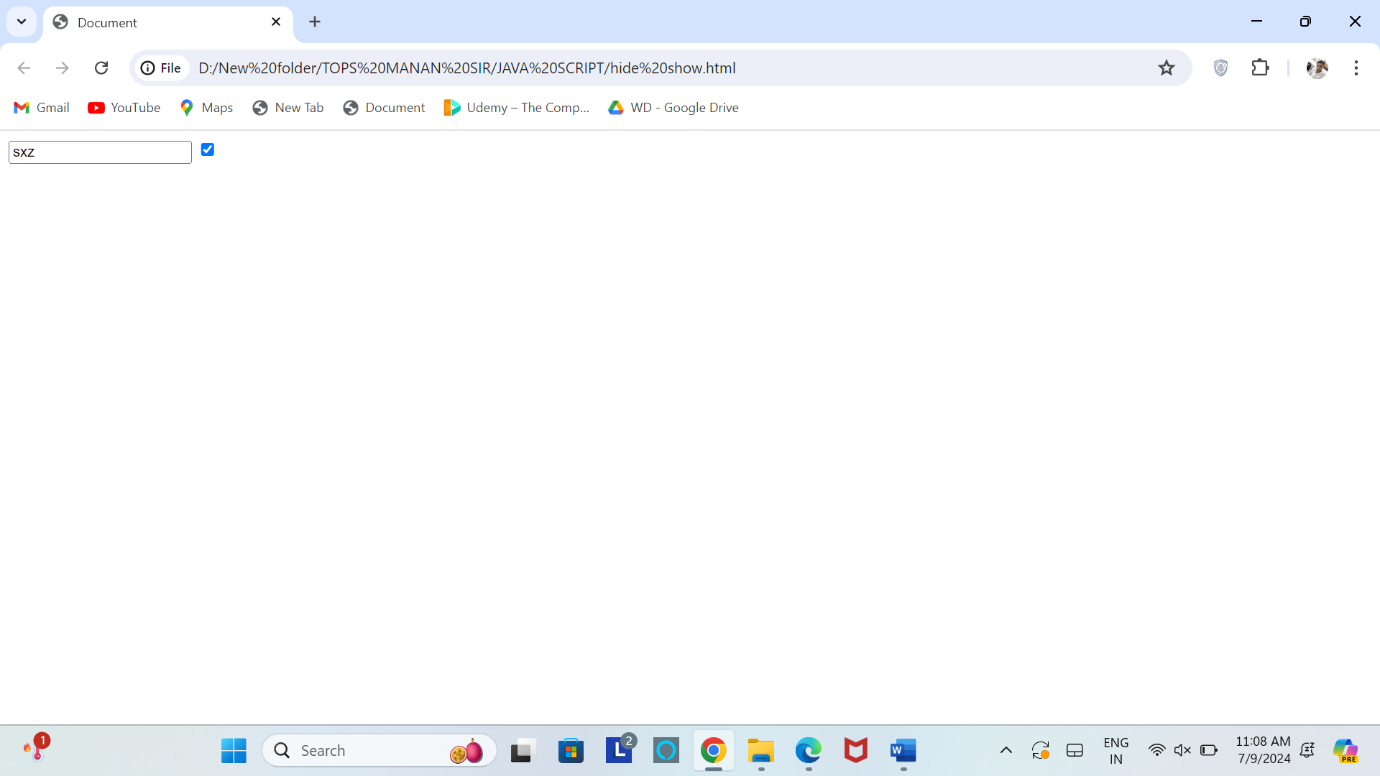
            }

        }

    </script>

</body>

Out put:-



.Create basic math operation in JS

Ans:- <body>

    FIRST VALUE : <input type="number" id="one">

    <br>

    SECOND VALUE : <input type="number" id="two">

    <br>

    <button onclick="main()">+</button>

    <button onclick="sub()">-</button>

    <button onclick="into()">\*</button>

    <button onclick="div()">/</button>

    <h1 id="demo"></h1>

    <script>

        function main() {

            let a = document.getElementById("one").value;

            let b = document.getElementById("two").value;

            let c = Number(a) + Number(b)

            document.getElementById("demo").innerHTML = c;

        }

        function sub() {

            let a = document.getElementById("one").value;

            let b = document.getElementById("two").value;

            let c = Number(a) - Number(b)

            document.getElementById("demo").innerHTML = c;

        }

        function into() {

            let a = document.getElementById("one").value;

            let b = document.getElementById("two").value;

            let c = Number(a) \* Number(b)

            document.getElementById("demo").innerHTML = c;

        }

        function div() {

            let a = document.getElementById("one").value;

            let b = document.getElementById("two").value;

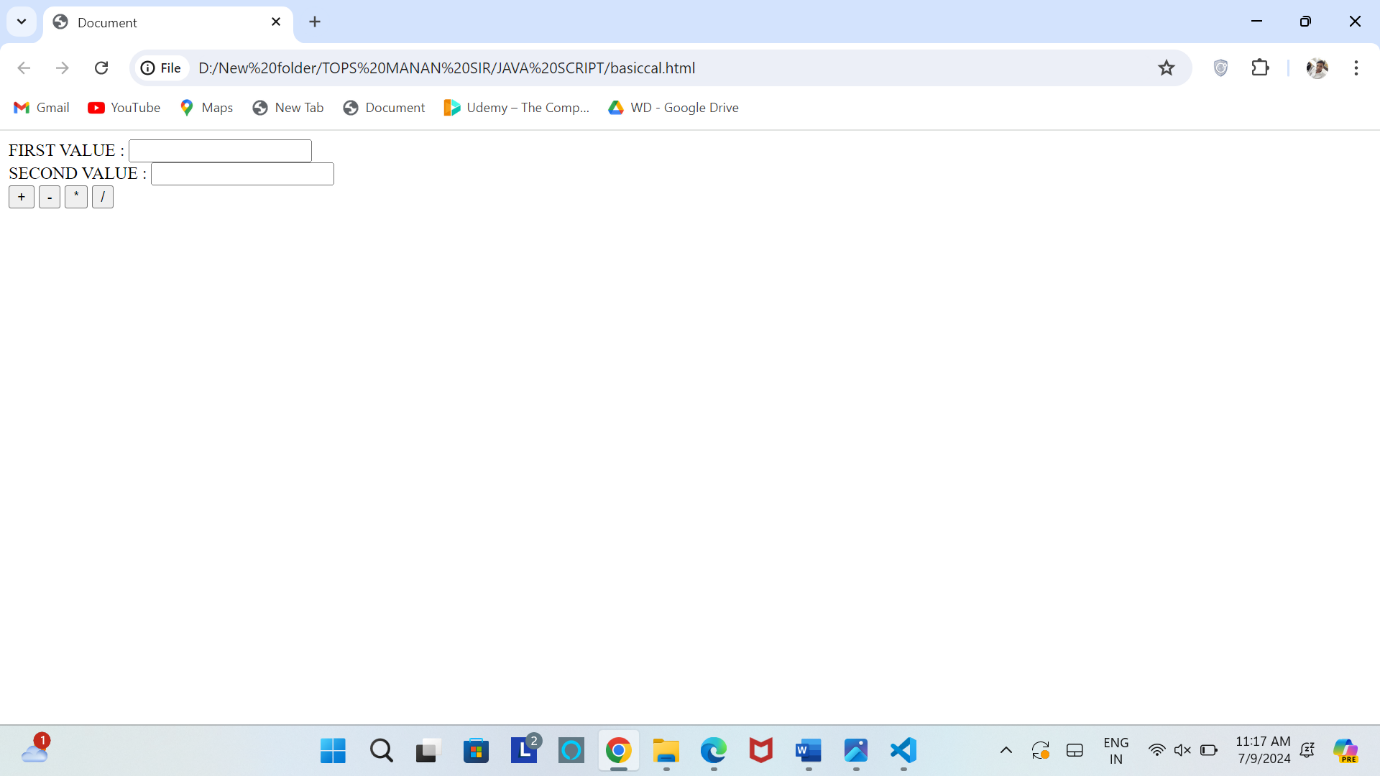
            let c = Number(a) / Number(b)

            document.getElementById("demo").innerHTML = c;

        }

    </script>

</body>

Out put:- 

**• Create result**

**Ans:-**

**<body>**

**<div class="text-center">**

**<h2>RESULT</h2>**

**<BR>**

**<fieldset>**

**<table border="0" align="center">**

**<form onsubmit="return result()">**

**<tr>**

**<td>**

**<label for="">ENGLISH:</label>**

**</td>**

**<td><input type="text" required min="0" class="form-control" max="100" id="english"**

**placeholder="Enter your mark"></td>**

**</tr>**

**<tr>**

**<td><label for="">HINDI:</label></td>**

**<td><input type="text" required min="0" class="form-control" max="100" id="hindi"**

**placeholder="Enter your mark"></td>**

**</tr>**

**<tr>**

**<td><label for="">GUJRATI:</label></td>**

**<td><input type="text" required min="0" class="form-control" max="100" id="gujarati"**

**placeholder="Enter your mark"></td>**

**</tr>**

**<tr>**

**<td><label for="">MATHES:</label></td>**

**<td><input type="text" required min="0" class="form-control" max="100" id="maths"**

**placeholder="Enter your mark"></td>**

**</tr>**

**<tr>**

**<td><label for="">SCIENCE:</label></td>**

**<td><input type="text" required min="0" class="form-control" max="100" id="scienc"**

**placeholder="Enter your mark"></td>**

**</tr>**

**<tr>**

**<TD></TD>**

**<td><button class="btn btn-primary">SUBMIT</button></td>**

**</tr>**

**</form>**

**</table>**

**<h1 id="total"></h1>**

**<h1 id="per"></h1>**

**<h1 id="grade"></h1>**

**<h1 id="suc"></h1>**

**</fieldset>**

**</div>**

**<script>**

**function result() {**

**var english = document.getElementById('english').value;**

**var hindi = document.getElementById('hindi').value;**

**var gujarati = document.getElementById('gujarati').value;**

**var maths = document.getElementById('maths').value;**

**var scienc = document.getElementById('scienc').value**

**var total;**

**var total = +english + +hindi + +gujarati + +maths + +scienc;**

**document.getElementById('total').innerHTML = "total mark =" + total;**

**var per;**

**per = (total / 500) \* 100;**

**document.getElementById('per').innerHTML = "per=" + per + " %";**

**if (per >= 90) {**

**document.getElementById('grade').innerHTML = "grade : A"**

**document.getElementById('grade').style = "color : green"**

**document.getElementById('suc').innerHTML = "congratulation you have pass this exam"**

**document.getElementById('suc').style = "color : green"**

**}**

**else if (per >= 70) {**

**document.getElementById('grade').innerHTML = "grade : B"**

**document.getElementById('grade','suc').style = "color : green"**

**document.getElementById('suc').innerHTML = "congratulation you have pass this exam"**

**document.getElementById('suc').style = "color : green"**

**}**

**else if (per >= 50) {**

**document.getElementById('grade').innerHTML = "grade : C"**

**document.getElementById('grade').style = "color : green"**

**document.getElementById('suc').innerHTML = "congratulation you have pass this exam"**

**document.getElementById('suc').style = "color : green"**

**}**

**else if (per >= 40) {**

**document.getElementById('grade').innerHTML = "grade : D"**

**document.getElementById('grade').style = "color : green"**

**document.getElementById('suc').innerHTML = "congratulation you have pass this exam"**

**document.getElementById('suc').style = "color : green"**

**}**

**else {**

**document.getElementById('grade').innerHTML = "fail"**

**document.getElementById('grade').style = "color : red"**

**document.getElementById('suc').innerHTML = "butter luck next time"**

**document.getElementById('suc').style = "color : red"**

**}**

**return false;**

**}**

**</script>**

**</body>**

**Out put:-**

