Q.1 What is JavaScript. How to use it?

JavaScript is the **Programming Language** for the Web.

JavaScript can update and change both **HTML** and **CSS.**

JavaScript can **calculate**, **manipulate** and **validate** data.

Q.2 How many type of Variable in JavaScript?

Variables are Containers for Storing Data

JavaScript Variables can be declared in 4 ways:

* Automatically
* Using var
* Using let
* Using const

Q.3 Define a Data Types in js?

### JavaScript has 8 Datatypes

1. String  
2. Number  
3. Bigint  
4. Boolean  
5. Undefined  
6. Null  
7. Symbol  
8. Object

Q.4 Write a mul Function Which will Work Properly When invoked With Following Syntax.

Q.5 What the deference between undefined and undeclare in JavaScript?

The variables which are written in the code but haven’t been assigned any value yet are called undefined.

Eg. let x;

console.log(x);

Undeclare:

These are the variables that do not exist in the memory

Q.6 Using console.log() print out the following statement: The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another. Using console.log() print out the following quote by Mother Teresa:

Console.log(“'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another’-mother Teresa”)

Q.7 Check if typeof '10' is exactly equal to 10. If not make it exactly equal?

If(10===10)

Q.8 Write a JavaScript Program to find the area of a triangle?

const baseValue = prompt('Enter the base of a triangle: ');

const heightValue = prompt('Enter the height of a triangle: ');

// calculate the area

const areaValue = (baseValue \* heightValue) / 2;

console.log(

`The area of the triangle is ${areaValue}`

);

Q.9 Write a JavaScript program to calculate days left until next Christmas?

<h1 style="color: green;">

year

</h1>

<h3>

Program to calculate days left until

next Christmas using JavaScript?

</h3>

<script>

// Get the current date

let today = new Date();

// Get the year of the current date

let christmasYear = today.getFullYear();

// Check if the current date is

// already past by checking if the month

// is December and the current day

// is greater than 25

if (today.getMonth() == 11 &&

today.getDate() > 25) {

// Add an year so that the next

// Christmas date could be used

christmasYear = christmasYear + 1;

}

// Get the date of the next Christmas

let christmasDate =

new Date(christmasYear, 11, 25);

// Get the number of milliseconds in 1 day

let dayMilliseconds =

1000 \* 60 \* 60 \* 24;

// Get the remaining amount of days

let remainingDays = Math.ceil(

(christmasDate.getTime() - today.getTime()) /

(dayMilliseconds)

);

// Write it to the page

document.write("There are " + remainingDays +

" days remaining until Christmas.");

</script>

Q.10 What is Condition Statement?

**JavaScript Conditional statements**allow you to execute specific blocks of code based on conditions. If the condition meets then a particular block of action will be executed otherwise it will execute another block of action that satisfies that particular condition.

Q.11 Find circumference of Rectangle formula : C = 4 \* a ?

var rectangleLength = parseInt(prompt("Enter length of rectangle:"));

var rectangleWidth = parseInt(prompt("Enter width of rectangle:"));

var areaOfRectangle = rectangleLength \* rectangleWidth ;

console.log("Area of rectangle is: " + areaOfRectangle);

var circumferenceOfRectangle = 2 \* (rectangleLength) + 2 \* (rectangleWidth);

console.log("Circumference of rectangle is: " + circumferenceOfRectangle);

Q.12 WAP to convert years into days and days into years?

function convertDays(days) {

// Convert days into total years, months, and days

const years = Math.floor(days / 365);

days -= years \* 365;

const months = Math.floor(days / 30);

days -= months \* 30;

return [years, months, days];

}

const [years, months, days] = convertDays(400);

console.log(Years: ${years}, Months: ${months}, Days: ${days});

Q.13 Convert temperature Fahrenheit to Celsius?

function fahrenheitToCelsius(fahrenheit) {

return (fahrenheit - 32) \* 5 / 9;

}

let fahrenheit = 100;

let celsius = fahrenheitToCelsius(fahrenheit);

console.log(`${fahrenheit}°F is ${celsius}°C`);

Q.14 Write a JavaScript exercise to get the extension of a filename.?

.js

Q.15 What is the result of the expression (5 > 3 && 2 < 4)?

Ans:in 5>3 the value is not more than 5 and minimum than 3 and value is not more

Q.16 What is the result of the expression (true && 1 && "hello")?

Ans:hello

Q.17 What is the result of the expression true && false || false && true?

Ans:false

Q.18 What is a Loop and Switch Case in JavaScript define that ?

Ans:

Switch:

* The switch expression is evaluated once.
* The value of the expression is compared with the values of each case.
* If there is a match, the associated block of code is executed.
* If there is no match, the default code block is executed.

Loop:

**JavaScript Loops** are powerful tools for performing repetitive tasks efficiently. Loops in JavaScript execute a block of code again and again while the condition is true.

Q.19 What is the use of is Nan function?

The  global property is a value representing Not-A-Number.

Q.20 What is the difference between && and || in JavaScript?

&&:is define both condition are true then execute

||:is define in both condition either one or two condition true it execute

Q.21 What is the use of Void (0)?

 This expression returns undefined primitive value. This is often used with hyperlinks. Sometimes, you will decide to call some JavaScript from inside a link. Normally, when you click a link, the browser loads a brand new page or refreshes the same page (depending on the URL specified). But you most likely don’t desire this to happen if you have hooked up some JavaScript thereto link. To prevent the page from refreshing, you could use void(0).

Q.22 Check Number Is Positive or Negative in JavaScript?

If(no>0)

{

Document.getelementById(“msg”).innertext=”Positive”

}

Else{

Document.getelementById(“msg”).innertext=”negative”

}

Q.23 Find the Character Is Vowel or Not ?

If(char==a||char==e||char==i||char==o||char==u)

{

Document.getelementById(“msg”).innerhtml=`character is vowel`

}

{

Document.getelementById(“msg”).innerhtml=`character is not vowel`

}

Q.24 Write to check whether a number is negative, positive or zero?

If(no==)

{

Document.getelementById(“msg”).innerhtml=`number is zero`

}

Elseif(no>0)

{

Document.getelementById(“msg”).innerhtml=`number is Positive`

}

Else{

Document.getelementById(“msg”).innerhtml=`number is negative`

}

Q.25 Write to find number is even or odd using ternary operator in JS?

let num = 15;

let result = (num % 2 == 0) ? "Even" : "Odd";

console.log(result);

Q.26 Write find maximum number among 3 numbers using ternary operator in JS?

const num1 = parseFloat(prompt("Enter first number: "));

const num2 = parseFloat(prompt("Enter second number: "));

const num3 = parseFloat(prompt("Enter third number: "));

let largest;

// check the condition

if(num1 >= num2 && num1 >= num3) {

largest = num1;

}

else if (num2 >= num1 && num2 >= num3) {

largest = num2;

}

else {

largest = num3;

}

// display the result

console.log("The largest number is " + largest);

Q.27 Write to find minimum number among 3 numbers using ternary operator in JS?

const num1 = parseFloat(prompt("Enter first number: "));

const num2 = parseFloat(prompt("Enter second number: "));

const num3 = parseFloat(prompt("Enter third number: "));

let largest;

// check the condition

if(num1 <= num2 && num1 <= num3) {

largest = num1;

}

else if (num2 <= num1 && num2 <= num3) {

largest = num2;

}

else {

largest = num3;

}

// display the result

console.log("The minimum number is " + minimum);

Q.28 Write to find the largest of three numbers in JS?

const num1 = parseFloat(prompt("Enter first number: "));

const num2 = parseFloat(prompt("Enter second number: "));

const num3 = parseFloat(prompt("Enter third number: "));

let largest;

// check the condition

if(num1 >= num2 && num1 >= num3) {

largest = num1;

}

else if (num2 >= num1 && num2 >= num3) {

largest = num2;

}

else {

largest = num3;

}

// display the result

console.log("The largest number is " + largest);

Q.29 Write to show i. Monday to Sunday using switch case in JS? ii. Vowel or Consonant using switch case in JS?

Q.30 What are the looping structures in JavaScript? Any one Example?

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

Q.31 Write a print 972 to 897 using for loop in JS?

for (let i = 972; i >= 897; i--) {

        console.log(i);

}

Q.32 Write to print factorial of given number?

const number = parseInt(prompt('Enter a positive integer: '));

// checking if number is negative

if (number < 0) {

console.log('Error! Factorial for negative number does not exist.');

}

// if number is 0

else if (number === 0) {

console.log(`The factorial of ${number} is 1.`);

}

// if number is positive

else {

let fact = 1;

for (i = 1; i <= number; i++) {

fact \*= i;

}

console.log(`The factorial of ${number} is ${fact}.`);

}

Q.33 Write to print Fibonacci series up to given numbers?

const number = parseInt(prompt('Enter the number of terms: '));

let n1 = 0, n2 = 1, nextTerm;

console.log('Fibonacci Series:');

for (let i = 1; i <= number; i++) {

console.log(n1);

nextTerm = n1 + n2;

n1 = n2;

n2 = nextTerm;

}

Q.34 Write to print number in reverse order e.g.: number = 64728 ---> reverse =82746 in JS?

let num1 = 64728;

let result = num1.toString().split('').reverse().join('');

console.log(result);

Q.35 Write a program make a summation of given number (E.g., 1523 Ans: - 11) in JS?

Q.36 Write a program you have to make a summation of first and last Digit. (E.g., 1234 Ans: - 5) in JS?

Q.37 Use console.log() and escape characters to print the following pattern in JS?

1)

<!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>

      var result = '';

for (var i = 0; i < 6; i++) {

    for (var j = 0; j < 6; j++) {

        if(i == 0 && j > 0){

          result += '';

        }

        else if(j == 0 && i>0){

          result += '';

        }

        else if(i>0 && j>0){

        result += (i\*j) + ' ';

        }

    }

    result += '\n'

}

console.log(result);

        </script>

</body>

</html>

2)

<!DOCTYPE html>

<html>

<body>

<script>

var i, j, k;

for(i=1; i<=5; i++)

{

 for(j=1; j<=i; j++)

 {

  if(j % 2 == 0)

   document.write("0");

  else

   document.write("1");

 }

 document.write("<br>");

}

</script>

</body>

</html>

3)

<!DOCTYPE html>

<html>

<body>

<script>

var i, j, k = 1;

for(i=1; i < 5; i++)

{

for(j=1; j<=i; j++)

document.write(k++ + " ");

document.write("<br>");

}

</script>

</body>

</html>

4)

<!DOCTYPE html>

<html>

<body>

<script>

let rows = 5;

// pattern variable carries the final pattern in string format

let pattern = "";

// outer loop runs for `rows` no. of times

for (let n = 1; n <= rows; n++) {

   // print stars for n number of times in each row

   for (let num = 1; num <= n; num++) {

      pattern += "\*";

   }

   pattern += "\n";

}

console.log(pattern);

</script>

</body>

</html>

5)

<!DOCTYPE html>

<html>

<body>

<script>

let n = 5; // you can take input using prompt or change the value

let string = "";

let count = 0;

// External loop

for (let i = 1; i <= n; i++) {

  for (let j = 0; j < i; j++) {

    string += String.fromCharCode(count + 65);

    count++; // increment cause next alphabet

  }

  string += "\n";

}

console.log(string);

</script>

</body>

</html>

(Array and object Question)

Q.40 Write a JavaScript Program to display the current day and time in the following format. Sample Output: Today is Friday. Current Time is 12 PM: 12 : 22 2 ?

const d = new Date("2022-03-25");

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

Q.41 Write a JavaScript program to get the current date?

<script>

const d = new Date();

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

</script>

Q.42 Write a JavaScript program to compare two objects?

let a = 1;

let b = 1;

let a1 = a;

console.log(a === a1);

Q.43 Write a JavaScript program to convert an array of objects into CSV string?

let array = [“abc”,”4”,”ajdcnj”];

let csv = array.toString();

console.log(csv);

Q.44 Write a JavaScript program to capitalize first letter of a string?

function capitalize(s)

{

return s && s[0].toUpperCase() + s.slice(1);

}

Q. 45 Write a JavaScript program to determine if a variable is array?

let colors = ['red','green','blue'];

let isArray = Array.isArray(colors);

console.log(isArray); *// true;*

Q.46 Write a JavaScript program to clone an array?

const originalArray = [1, 2, 3];

const clonedArray = originalArray.slice();

console.log(clonedArray);

Q.47 What is the drawback of declaring methods directly in JavaScript objects?

,calling a function is slightly longer (eg : **my.foo()**) and if we have bunch of extra nested functions, it would be more longer

Q.48 Print the length of the string on the browser console using console.log()?

let str = "GeeksforGeeks";

let i = 0, cnt = 0;

while (str[i] !== undefined) {

i++;

cnt++;

}

console.log(cnt);

Q.49 Change all the string characters to capital letters using toUpperCase() method?

let text = "Hello World!";  
let result = text.toUpperCase();

Q.50 What is the drawback of declaring methods directly in JavaScript objects?

method requires some extra processing by JS engine but this will not affect the speed much as it would be like optimizing our code execution time ~0.1ms , not an achievement).

method, calling a function is slightly longer (eg : **my.foo()**) and if we have bunch of extra nested functions, it would be more longer

Q.51 Write a JavaScript program to get the current date. Expected Output : mm-dd-yyyy, mm/dd/yyyy or dd-mm-yyyy, dd/mm/yyyy?

const date = new Date();

const formatter = new Intl.DateTimeFormat('en-US', { dateStyle: 'short' });

const formattedDate = formatter.format(date);

console.log(formattedDate);

const date = new Date();

const formatter = new Intl.DateTimeFormat('en-US', { day: '2-digit', month: '2-digit', year: 'numeric' });

const formattedDate = formatter.format(date);

console.log(formattedDate);

const date = new Date();

const formattedDate = `${date.getDate()}-${date.getMonth() + 1}-${date.getFullYear()}`;

console.log(formattedDate);

const date = new Date();

const formattedDate = `${date.getMonth() + 1}-${date.getDate()}-${date.getFullYear()}`;

console.log(formattedDate);

Q.52 Use indexOf to determine the position of the first occurrence of a in 30 Days Of JavaScript?

Q,53 Use lastIndexOf to determine the position of the last occurrence of a in 30 Days Of JavaScript?

let text = "Hello world, welcome to the universe.";  
let result = text.indexOf("welcome");

Q.54 Form Validtion in JS?

function validateForm() {  
  let x = document.forms["myForm"]["fname"].value;  
  if (x == "") {  
    alert("Name must be filled out");  
    return false;  
  }  
}

Q.55 Form in Email, number, Password, Validation?

function showPassword(typechge, icon) {

let inptype = document.getElementById(typechge);

if (inptype.type == "password" && icon.classList.contains("fa-eye") == true) {

inptype.type = "text";

icon.classList.remove("fa-eye")

icon.classList.add("fa-eye-slash")

} else {

inptype.type = "password";

icon.classList.remove("fa-eye-slash")

icon.classList.add("fa-eye")

}

}

function userCheck() {

var no = document.getElementById("mobileno");

if (no.value == "") {

document.getElementById("msg").innerHTML = "Enter Number";

no.classList.add("error");

no.focus()

}

return false

}

function numberCheck() {

var no = document.getElementById("mobileno");

var pattern = /^[0-9]\*$/;

var start = /^[6-9]/

if (!(pattern.test(no.value))) {

document.getElementById("msg").innerHTML = "Invalid Number!!!";

no.classList.add("error")

} else if (!(start.test(no.value))) {

document.getElementById("msg").innerHTML = "Number should be start in between 6-9";

no.classList.add("error")

} else if (no.value.length < 10) {

document.getElementById("msg").innerHTML = "Enter 10 digits";

no.classList.add("error")

} else {

document.getElementById("msg").innerHTML = ""

no.classList.remove("error")

}

return false

}

function emailCheck() {

// var emailptn = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{3})+$/;

var emailptn = /[A-Za-z0-9\.\_%$+\-]+@[A-Za-z\.\-]+\.[A-Za-z]{3}/

var mail = document.getElementById("email");

if (emailptn.test(mail.value)) {

document.getElementById("msg").innerHTML = ""

} else {

document.getElementById("msg").innerHTML = "Invalid Email Address!!!"

}

}

Q.56 Dynamic Form Validation in JS?

function validateForm() {

const name =

document.getElementById("name").value;

const address =

document.getElementById("address").value;

const email =

document.getElementById("email").value;

const password =

document.getElementById("password").value;

const subject =

document.getElementById("subject").value;

const agree =

document.getElementById("agree").checked;

const nameError =

document.getElementById("name-error");

const addressError = document.getElementById(

"address-error"

);

const emailError = document.getElementById(

"email-error"

);

const passwordError = document.getElementById(

"password-error"

);

const subjectError = document.getElementById(

"subject-error"

);

const agreeError = document.getElementById(

"agree-error"

);

nameError.textContent = "";

addressError.textContent = "";

emailError.textContent = "";

passwordError.textContent = "";

subjectError.textContent = "";

agreeError.textContent = "";

let isValid = true;

if (name === "" || /\d/.test(name)) {

nameError.textContent =

"Please enter your name properly.";

isValid = false;

}

if (address === "") {

addressError.textContent =

"Please enter your address.";

isValid = false;

}

if (email === "" || !email.includes("@")) {

emailError.textContent =

"Please enter a valid email address.";

isValid = false;

}

if (password === "" || password.length < 6) {

passwordError.textContent =

"Please enter a password with at least 6 characters.";

isValid = false;

}

if (subject === "") {

subjectError.textContent =

"Please select your course.";

isValid = false;

}

if (!agree) {

agreeError.textContent =

"Please agree to the above information.";

isValid = false;

}

return isValid;

}

Q.57 how many type of JS Event? How to use it ?

|  |  |
| --- | --- |
| **Event** | **Description** |
| onchange | An HTML element has been changed |
| onclick | The user clicks an HTML element |
| onmouseover | The user moves the mouse over an HTML element |
| onmouseout | The user moves the mouse away from an HTML element |
| onkeydown | The user pushes a keyboard key |
| onload | The browser has finished loading the page |

Q.59 What is Bom vs Dom in JS?

**Browser Object Model (BOM)** is a browser-specific convention referring to all the objects exposed by the web browser. The BOM allows JavaScript to “interact with” the browser. The [window object](https://www.geeksforgeeks.org/window-object-in-javascript/) represents a browser window and all its corresponding features. A window object is created automatically by the browser itself. Java Script’s window.screen object contains information about the user’s screen. It can also be written without the window prefix. This Object Model supports the following Window properties:

* [**screen.width**](https://www.geeksforgeeks.org/html-screen-width-property/)**:** The screen.width property returns the user’s screen width in pixels.
* [**screen.height**](https://www.geeksforgeeks.org/html-dom-screen-height-property/)**:** The screen.height property returns the user’s screen height in pixels.
* [**screen.availWidth**](https://www.geeksforgeeks.org/html-screen-availwidth-property/)**:** The screen.availWidth property returns the user’s screen width in pixels, excluding the interface features.
* [**screen.availHeight**](https://www.geeksforgeeks.org/html-screen-availheight-property/)**:**The screen.availHeight property returns the user’s screen height in pixels excluding the interface features.
* [**screen.colorDepth**](https://www.geeksforgeeks.org/html-screen-colordepth-property/)**:** The screen.colorDepth property returns the bits (number) to be used to display one color.
* [**screen.pixelDepth**](https://www.geeksforgeeks.org/html-screen-pixeldepth-property/)**:** The screen.pixelDepth property returns the pixel depth of the screen.

[**Document Object Model (DOM)**](https://www.geeksforgeeks.org/dom-document-object-model/) is a programming interface for HTML and XML documents, that allows to create, manipulate, or delete the element from the document. It defines the logical structure of documents and the way a document is accessed and manipulated. With the help of DOM, the webpage can be represented in a structured hierarchy, i.e., we can easily access and manipulate tags, IDs, classes, Attributes, or Elements of HTML using commands or methods provided by the Document object, that will guide the programmers and users to understand the document in an easier manner.

HTML is used to structure the web pages and [Javascript](https://www.geeksforgeeks.org/javascript/) is used to add behavior to our web pages. When an HTML file is loaded into the browser, the javascript can not understand the HTML document directly. So, a corresponding document is created(DOM). DOM is basically the representation of the same HTML document but in a different format with the use of objects. DOM provides several methods to find & manipulate the behavior of the [HTML element](https://www.geeksforgeeks.org/html-elements/):

* [**getElementById() Method**](https://www.geeksforgeeks.org/html-dom-getelementbyid-method/)**:**This method returns the elements that have given an ID which is passed to the function.
* [**getElementsByClassName() Method**](https://www.geeksforgeeks.org/html-dom-getelementsbyclassname-method/)**:** This method in Javascript returns an object containing all the elements with the specified class names in the document as objects: This
* [**getElementsByName() Method**](https://www.geeksforgeeks.org/html-dom-getelementsbyname-method/)**:** This method returns the collection of all elements of a particular document by name.
* [**getElementsByTagName() Method**](https://www.geeksforgeeks.org/html-dom-getelementsbytagname-method/)**:** This method in HTML returns the collection of all the elements in the document with the given tag name.
* [**querySelector() Method**](https://www.geeksforgeeks.org/html-dom-queryselector-method/)**:**This method in HTML is used to return the first element that matches a specified CSS selector(s) in the document.
* [**querySelectorAll() Method**](https://www.geeksforgeeks.org/html-dom-queryselectorall-method/)**:**This method is used to return a collection of an element’s child elements that match a specified CSS selector(s), as a static NodeList object.

Q.60 Array vs object defences in JS?

| **Array** | **Array of objects** |
| --- | --- |
| Arrays are best to use when the elements are **numbers**. | Objects are best to use when the elements’ **strings (text)**. |
| The data inside an array is known as **Elements.** | The data inside objects are known as **Properties**which consists of a **key**and a **value.** |
| The elements can be manipulated using []. | The properties can be manipulated using both**.DOT** notation and []. |
| The elements can be popped out of an array using the **pop()**function. | The keys or properties can be deleted by using the **delete**keyword. |
| Iterating through an array is possible using **For loop, For..in, For..of**, and **ForEach().** | Iterating through an array of objects is possible using **For..in, For..of, and ForEach().** |

Q.61 Split the string into an array using split() Method?

let text = "How are you doing today?";

const myArray = text.split(" ");

document.getElementById("demo").innerHTML = myArray[1];

Q.62 Check if the string contains a word Script using includes() method?

**const** str **=** 'The quick brown fox jumps over the lazy dog.';

console.**log**(str.**indexOf**('fox'));

*// expected output: 16*

Q.63 Change all the string characters to lowercase letters using toLowerCase() Method.

function func() {

let str = 'GEEKSFORGEEKS';

let string = str.toLowerCase();

console.log(string);

}

func();

Q.64 What is Character at index 15 in ’30 Days of JavaScript’ string? Use charAt() method.

let text = "’30 Days of JavaScript";  
let letter = text.charAt(15);

Q.65 copy to one string to another string in JS?

let str = 'Sliced\_String';

let res = str.slice(0, 5);

console.log(res);

Q.66 Find the length of a string without using libraryFunction?

let btnGet = document.querySelector("button");

let output = document.querySelector("h1");

let myString = "Hello world";

btnGet.addEventListener("click", () => {

let result = getStringLength(myString);

output.innerText = result;

});

function getStringLength(str){

let length = Array.from(str).reduce(function(count){return count+1}, 0);

return length

}

• What is JavaScript?

avaScript 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 isNaN function?

In JavaScript NaN is short for "Not-a-Number".

The isNaN() method returns true if a value is NaN.

The isNaN() method converts the value to a number before testing it.

• What is negative Infinity?

Number.NEGATIVE\_INFINITY returns negative infinity.

Number.NEGATIVE\_INFINITY is "a number lower than any other number".

• Which company developed JavaScript? • What are undeclared and undefined variables?

|  |  |
| --- | --- |
| These are the variables that do not exist in the memory heap. | These variables are the ones that do exist in memory but nothing is being assigned to them explicitly by the programmer. |

• Write the code for adding new elements dynamically?

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

• What is the difference between ViewState and SessionState? • What is === operator?

**ViewState:** It is maintained at only one level that is page-level. Changes made on a single page is not visible on other pages. Information that is gathered in view state is stored for the clients only and cannot be transferred to any other place. View state is synonymous with serializable data only

**SessionState:** It is maintained at session-level and data can be accessed across all pages in the web application. The information is stored within the server and can be accessed by any person that has access to the server where the information is stored.

|  |  |
| --- | --- |
| === | equal value and equal type |

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

document.getElementById(*id*).style.*property*=*new style*

document.getElementByClass(*Class*).style.*property*=*new style*

• How to read and write a file using JavaScript?

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

• What are all the looping structures in JavaScript?

* 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?

arseInt(string\_value, base)

• What is the function of the delete operator?

delete object.property

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

Alert box,promptbox,confirmbox

• What is the use of Void (0)?

The reason you’d want to do this with the href of a link is that normally, a javascript: URL will redirect the browser to a plain text version of the result of evaluating that JavaScript. But if the result is undefined, then the browser stays on the same page. void(0) is just a short and simple script that evaluates to undefined.

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

## Using window.location.replace

## : Using **[window.location.assign Property](https://www.geeksforgeeks.org/html-location-assign-method/)**

• 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.

• Create password field with show hide functionalities

<!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>

.userform {

width: 550px;

border: 2px solid black;

}

.row {

display: grid;

grid-template-columns: 0.5fr 2fr 1fr;

padding: 10px;

gap: 20px;

align-items: center;

}

input {

height: 20px;

width: 90%;

border: 1px solid gray;

padding: 3px 10px;

}

input:focus-visible {

outline: none;

}

span {

color: red;

}

.error {

border: 1px solid red;

}

</style>

</head>

<body>

<div class="userform">

<form action="#">

<div class="row">

<div><label for="">Password :</label></div>

<div>

<input type="password" name="" id="pass" onblur="return checkpassword()"

onkeyup="return checkPassPattern()">

</div>

<div>

<span id="msg1"></span>

</div>

</div>

<div class="row">

<div><label for="">Confirm password :</label></div>

<div>

<input type="password" name="" id="cpass" onblur="return checkpassword()"

onkeyup="return Confirmpassword()">

</div>

<div>

<span id="msg2"></span>

</div>

</div>

<div class="row">

<input type="checkbox" onclick="showPassword()"> Show Password

</div>

<div class="row">

<div>

<button onclick="return checkpassword()">Check</button>

</div>

</div>

<p>

Note : password must be in 8 character. which contains atleast lowercase, uppercase, number and special

characters. first letter must be capital

</p>

</form>

</div>

<script>

function checkpassword() {

var pass = document.getElementById("pass");

var cpass = document.getElementById("cpass")

if (pass.value == "") {

document.getElementById("msg1").innerHTML = "Enter Password";

pass.classList.add("error");

pass.focus();

}

if (cpass.value == "") {

document.getElementById("msg2").innerHTML = "Enter Confirm Password";

cpass.classList.add("error");

}

return false

}

function checkPassPattern() {

var pass = document.getElementById("pass");

// var passptn = /^[a-zA-Z0-9!@?%&]{6,16}$/;

var passptn = /^(?=.\*\d)(?=.\*[!@#$%^&\*])(?=.\*[a-z])(?=.\*[A-Z]).{8,}$/;

if (passptn.test(pass.value)) {

document.getElementById("msg1").innerHTML = ""

pass.classList.remove("error")

} else {

document.getElementById("msg1").innerHTML = "Invalid Password";

pass.classList.add("error")

}

return false;

}

function Confirmpassword() {

var pass = document.getElementById("pass");

var cpass = document.getElementById("cpass");

if (cpass.value != pass.value) {

document.getElementById("msg2").innerHTML = "password not match"

cpass.classList.add("error")

} else {

document.getElementById("msg2").innerHTML = ""

cpass.classList.remove("error")

}

return false

}

function showPassword() {

var pass = document.getElementById("pass");

var cpass = document.getElementById("cpass");

if (pass.type == "password" && cpass.type == "password") {

pass.type = "text"

cpass.type = "text"

} else {

pass.type = "password"

cpass.type = "password"

}

}

</script>

</body>

</html>

<!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>

        .row{

            display: grid;

            grid-template-columns: 1fr  1fr 1fr;

            width: 800px;

            margin: auto;

            padding: 10px;

            border: 1px solid black;

        }

    </style>

</head>

<body>

    <form action="#">

        <h5 style="text-align: center;">Marksheet infomation</h5>

        <h3 style="text-align: center;">Enter Marks</h3>

        <div class="row">

            <label for="">1.C language</label>

            <input type="text" id="C">

            <div><span id="msg"></span></div>

        </div>

        <div class="row">

            <label for="">2.C++language</label>

            <input type="text" id="cc">

            <div><span id="msg"></span></div>

        </div>

        <div class="row">

            <label for="">3.Database</label>

            <input type="text" id="db">

            <div><span id="msg"></span></div>

        </div>

        <div class="row">

            <label for="">4.Html</label>

            <input type="text" id="html">

            <div><span id="msg"></span></div>

        </div>

        <div class="row">

            <label for="">5.Css</label>

            <input type="text" id="css">

            <div><span id="msg"></span></div>

        </div>

        <div class="row">

            <label for="">6.Php</label>

            <input type="text" id="php">

            <div><span id="msg"></span></div>

        </div>

        <div class="row">

            <label for="">7.Core Java</label>

            <input type="text" id="java">

            <div><span id="msg"></span></div>

        </div>

        <div class="row">

            <div></div>

            <button id="result" onclick="return displayresult()"

            style="height: 30px; font-size: 20px;">Result</button>

        </div>

        <div class="row">

            <div >Total:

                <span id="total"></span>

            </div>

            <div >Percentage:

                <span id="percentage"></span>

            </div>

            <div >Grade:

                <span id="grade"></span>

            </div>

            <div>You Are

                <span id="pass"></span>

            </div>

        </div>

    </form>

    <script>

        function displayresult() {

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

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

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

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

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

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

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

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

            result=parseFloat(c)+parseFloat(cc)+parseFloat(database)+parseFloat(html)+parseFloat(css)+parseFloat(php)+parseFloat(java);

            document.getElementById("total").innerHTML=` ${result}`

let percentage = (result / 350) \* 100;

if (percentage <= 50 && percentage >= 40) {

    grades = "A";

} else if (percentage <= 40 && percentage >= 30) {

    grades = "B";

} else if (percentage <= 30 && percentage >= 25) {

    grades = "C";

} else {

    grades = "F";

}

document.getElementById("percentage").innerHTML=` ${percentage}`

document.getElementById("grade").innerHTML=` ${grades}`

}

    </script>

</body>

</html>

<!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>

.row1{

    display: grid;

    grid-template-columns: 1fr 1fr;

    width: 400px;

    margin: auto;

    padding: 10px 0px;

    top: 10px;

}

button{

    height: 40px;

    width: 50px;

    margin: 10px;

    align-items: center;

    justify-content: center;

    font-size: 20px;

    }

    span{

        text-align: center;

        font-size: 30px;

    }

    </style>

</head>

<body>

    <form action="#">

        <div  class="row1">

        <label for="">Enter Number 1</label>

         <input type="text" id="n1">

        </div>

        <div  class="row1">

        <label for="">Enter Number 2</label>

        <input type="text" id="n2">

        </div>

        <div  class="row1">

            <button id="plus" onclick="  checkPlus()">+</button>

            <button id="minus" onclick="  checkminus()">-</button>

            <button id="multi" onclick="  checkmulti()">\*</button>

            <button id="divi" onclick="  checkdivi()">/</button>

            <button id="modu" onclick="  checkmodu()">%</button>

            </div>

    <span id="msg" > </span>

</div>

    </form>

    <script>

        function checkPlus() {

            let no1 = Number(document.getElementById("n1").value);

            let no2 = Number(document.getElementById("n2").value);

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

            plus=no1+no2;

            document.getElementById("msg").innerHTML=`Answer is ${plus}`

        }

        function checkminus() {

            let no1 = Number(document.getElementById("n1").value);

            let no2 = Number(document.getElementById("n2").value);

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

            minus=no1-no2;

            document.getElementById("msg").innerHTML=`Answer is ${minus}`

        }

        function checkmulti() {

            let no1 = Number(document.getElementById("n1").value);

            let no2 = Number(document.getElementById("n2").value);

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

            multi=no1\*no2;

            document.getElementById("msg").innerHTML=`Answer is ${multi}`

        }

        function checkdivi() {

            let no1 = Number(document.getElementById("n1").value);

            let no2 = Number(document.getElementById("n2").value);

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

            divi=no1/no2;

            document.getElementById("msg").innerHTML=`Answer is ${divi}`

        }

        function checkmodu() {

            let no1 = Number(document.getElementById("n1").value);

            let no2 = Number(document.getElementById("n2").value);

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

            modu=no1%no2;

            document.getElementById("msg").innerHTML=`Answer is ${modu}`

        }

    </script>

</body>

</html>

<!DOCTYPE html>

<html>

<head>

<meta name="viewport" content="width=device-width, initial-scale=1">

<style>

\* {box-sizing: border-box}

body {font-family: Verdana, sans-serif; margin:0}

.mySlides {display: none}

img {vertical-align: middle;}

/\* Slideshow container \*/

.slideshow-container {

max-width: 1000px;

position: relative;

margin: auto;

}

/\* Next & previous buttons \*/

.prev, .next {

cursor: pointer;

position: absolute;

top: 50%;

width: auto;

padding: 16px;

margin-top: -22px;

color: white;

font-weight: bold;

font-size: 18px;

transition: 0.6s ease;

border-radius: 0 3px 3px 0;

user-select: none;

}

/\* Position the "next button" to the right \*/

.next {

right: 0;

border-radius: 3px 0 0 3px;

}

/\* On hover, add a black background color with a little bit see-through \*/

.prev:hover, .next:hover {

background-color: rgba(0,0,0,0.8);

}

/\* Caption text \*/

.text {

color: #f2f2f2;

font-size: 15px;

padding: 8px 12px;

position: absolute;

bottom: 8px;

width: 100%;

text-align: center;

}

/\* Number text (1/3 etc) \*/

.numbertext {

color: #f2f2f2;

font-size: 12px;

padding: 8px 12px;

position: absolute;

top: 0;

}

/\* The dots/bullets/indicators \*/

.dot {

cursor: pointer;

height: 15px;

width: 15px;

margin: 0 2px;

background-color: #bbb;

border-radius: 50%;

display: inline-block;

transition: background-color 0.6s ease;

}

.active, .dot:hover {

background-color: #717171;

}

/\* Fading animation \*/

.fade {

animation-name: fade;

animation-duration: 1.5s;

}

@keyframes fade {

from {opacity: .4}

to {opacity: 1}

}

/\* On smaller screens, decrease text size \*/

@media only screen and (max-width: 300px) {

.prev, .next,.text {font-size: 11px}

}

</style>

</head>

<body>

<div class="slideshow-container">

<div class="mySlides fade">

<div class="numbertext">1 / 3</div>

<img src="img\_nature\_wide.jpg" style="width:100%">

<div class="text">Caption Text</div>

</div>

<div class="mySlides fade">

<div class="numbertext">2 / 3</div>

<img src="img\_snow\_wide.jpg" style="width:100%">

<div class="text">Caption Two</div>

</div>

<div class="mySlides fade">

<div class="numbertext">3 / 3</div>

<img src="img\_mountains\_wide.jpg" style="width:100%">

<div class="text">Caption Three</div>

</div>

<a class="prev" onclick="plusSlides(-1)">❮</a>

<a class="next" onclick="plusSlides(1)">❯</a>

</div>

<br>

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

<span class="dot" onclick="currentSlide(1)"></span>

<span class="dot" onclick="currentSlide(2)"></span>

<span class="dot" onclick="currentSlide(3)"></span>

</div>

<script>

let slideIndex = 1;

showSlides(slideIndex);

function plusSlides(n) {

showSlides(slideIndex += n);

}

function currentSlide(n) {

showSlides(slideIndex = n);

}

function showSlides(n) {

let i;

let slides = document.getElementsByClassName("mySlides");

let dots = document.getElementsByClassName("dot");

if (n > slides.length) {slideIndex = 1}

if (n < 1) {slideIndex = slides.length}

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

slides[i].style.display = "none";

}

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

dots[i].className = dots[i].className.replace(" active", "");

}

slides[slideIndex-1].style.display = "block";

dots[slideIndex-1].className += " active";

}

</script>

</body>

</html>