1. What is JavaScript. How to use it ?

Ans. JavaScript is a scripting language that enables you to create dynamically updating content, Control multimedia, Animate images, and pretty much everything else.

JavaScript provides user to interact with web pages as per the requirement.

To use JavaScript we use <script></script> tag to write code.

1. How many type of Variable in JavaScript ?

Ans. Ther are three types of variable in JavaScript.

1. var: variable can be re-declare and can be updated in the same scope.
2. let: It can be updated but can not be re-declare in the same scope.
3. Const: It can not be re-decclare or can not be updated in any scope.
4. Define a Data Types in js ?

Ans. There are mainly two types of datatype

1. Primitive DataType:

* String
* Number
* Boolean
* BigInt
* Null
* Undefined
* Symbol

1. Non-Primitive DataType

* Object : Represents a collection of Key-Value pair where keys are string or symbols and values cab be any data type.
* Array : represents an ordered list of values. Array can contain elements of any datatype.

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

Ans. Function mul()

{

If (argument.length == 0)

{

return 0;

}

elseif(argument.length == 1)

{

return argument[0];

}

else

{

let result=1;

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

{

result = result\*argument[i];  
}

}

}

Test case:

console.log(mul()); => 0

console.log(mul(2)); 🡺 2

consol.log(mul(3,5,9)); => 135

1. What the deference between undefined and undeclare in JavaScript ?

Ans. **Undefined** : Undefined is a primitive type value that represents a variable that has been declared but not assigned a value, or non-existent property of an object.

**Undeclared**: Undeclare refers to a situation where you try to access a variable that has been declared using keyword like var, let or const.

1. 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:

Ans. Console.log(“'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes”);

Consol.log(“help one another – Mother Teresa”);

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

Ans.

if(typeof ‘10’ === typeof 10) checks if the type of ‘10’ is exactly equal to the type of 10.

{

Console.log(“Equal to 10”)

}

Else

{

Let num = parseInt(‘10’); converted to parseInt

If(num === 10)

{

Console.log(“Now it is equal to 10”);

}

Else

{

Console.log(“It can not be converted”)

}

}

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

Ans. <script>

        function area(base, height) {

            // Calculate the area using the formula: area = 0.5 \* base \* height

            var area = ( base \* height);

            var ans = area/2;

            return ans;

        }

        var triangleArea = area(5, 7);

        document.write("The area of the triangle is: " + triangleArea);

    </script>

Output : 17.5

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

Ans. <script>

        function Christmas() {

            // Get the current date

            var today = new Date();

            // Get the year of the current date

            var currentYear = today.getFullYear();

            // Set the target date to be Christmas of the current year

            var targetDate = new Date(currentYear, 11, 25); // Month is 0-based (0 = January, 11 = December)

            // If Christmas has already passed this year, set the target date to be Christmas of the next year

            if (today.getMonth() === 11 && today.getDate() > 25) {

                targetDate.setFullYear(currentYear + 1);

            }

            // Calculate the difference in milliseconds between the target date and the current date

            var timeDifference = targetDate.getTime() - today.getTime();

            // Convert the time difference from milliseconds to days

            var daysLeft = Math.ceil(timeDifference / (1000 \* 60 \* 60 \* 24));

            return daysLeft;

        }

        // Example usage:

        var daysLeft = Christmas();

        document.write("There are " + daysLeft + " days left until Christmas.");

    </script>

Output : There are 233 days left until Christmas.

1. What is Condition Statement ?

Ans. Conditional statement is programming are used to control the flow of a program based on certain conditions. These statements allow the execution of different code blocks depending on whether a specified condition evaluates to true or false.

1. If condition.
2. If-else condition.
3. If-else-if condition.
4. Switch condition.
5. Ternary conditional statement.
6. Find circumference of Rectangle formula : C = 4 \* a ?

Ans. The given formula is not for rectangle this formula if for square.

The formula of rectangle is (2\*(length + width))

<script>

        function Circumference(l , w)

    {

        return (2 \* (l + w));

    }

        var c = Circumference(5, 3);

        document.write("Circumference of " + "a rectangle is " + c);

    </script>

Output : Circumference of a rectangle is 16

1. WAP to convert years into days and days into years ?

Ans. <body>

    <input type="number" id="data"> &nbsp;&nbsp;

    <input type="button" value="Find Years" onclick="findyears()">  &nbsp;&nbsp;

    <input type="button" value="Find days" onclick="findDays()">  &nbsp;&nbsp;

    <p id="ans"></p>

    <script>

        function findyears(){

           var days = document.getElementById("data").value;

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

            var month = (days%365)/30;

            // var month = reminder/30;

            var day = (days%365)%30;

            document.getElementById("ans").innerHTML = "Total Years is = " + years + " and Total Month is = "+month + " and Total Day is = " + day;

        }

        function findDays(){

            var years = document.getElementById("data").value;

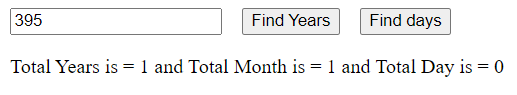
            var day = years\*365;

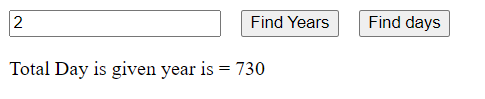
            document.getElementById("ans").innerHTML = "Total Day is given year is  = " + day;

        }

    </script>

Output:





1. Convert temperature Fahrenheit to Celsius? (Conditional logic Question)

Ans. <script>

        function fahrenheitToCelsius(fahrenheit) {

            if (fahrenheit < -459.67) {

                return "Temperature is below absolute zero. Invalid input.";

            } else {

                var celsius = (fahrenheit - 32) \* 5 / 9;

                return celsius.toFixed(2); // Round to 2 decimal places

            }

        }

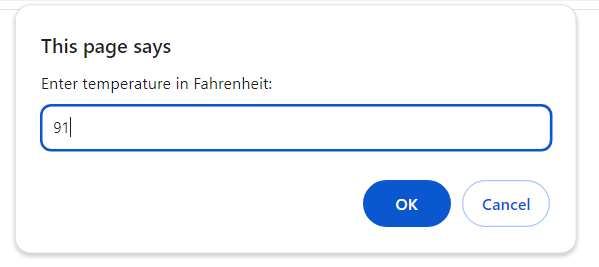
        // Example usage

        var fahrenheitTemperature = parseFloat(prompt("Enter temperature in Fahrenheit: "));

        var celsiusTemperature = fahrenheitToCelsius(fahrenheitTemperature);

        document.write("Temperature in Celsius:", celsiusTemperature);

    </script>

Output : 



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

Ans. <script>

        function getFileExtension(filename) {

            // Split the filename by periods (.)

            var parts = filename.split(".");

            // Return the last part of the array, which is the extension

            return parts[parts.length - 1];

        }

        // Example usage

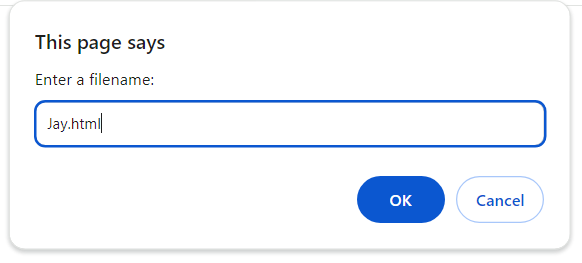
        var filename = prompt("Enter a filename:");

        var extension = getFileExtension(filename);

        document.write("The extension of the filename is: ", extension);

    </script>

Output :





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

Ans. The result of given expression is true. Because ‘&&’ operator represents the logical AND. It returns true when both of its operand are true. In this case ‘5>3’ and ‘2<4’ are true. So overall of the expression is ‘True’.

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

Ans. The result of given expression is true. Because ‘&&’ operator represents the logical AND. It returns true when both of its operand are true. In this case

True evaluates true

1 is number so it evaluates true.

“hello” is string so it evaluates true.

So overall expression is true.

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

Ans. Here two operator is present first && means logical AND and second is || means logical OR. First of all break down the expression

(true && false) || (false && true). Now solve one by one

(true && false) evaluates to false.

(false && true) evaluates to false.

Finally ‘false || false’ evaluates to ‘false’.

So overall given expression is false.

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

Ans. Loop : loop is a programming construct that allows you to execute a block of code repeatedly until a certain condition is satisfied.

1. For loop
2. While loop
3. Do-while loop etc.

Switch case: A switch case statement in JavaScript provides a way to execute different blocks of code based on the value of an expression. It's often used as an alternative to a series of if…else if…else statements when you have multiple conditions to check.

The ‘break’ statement is used to exit the switch block. If it is satisfied with the certain condition it will execute the block of code.

The ‘default’ case is execute when on any case will execute. Default block code will execute.

1. What is the use of is Nan function ?

Ans. The isNaN() function determines whether a value is NaN when converted to a number. Because coercion inside the isNaN() function can be surprising, you may alternatively want to use Number.isNaN() .

isNaN() method returns true if a value is Not-a-Number. Number.isNaN() returns true if a number is Not-a-Number. In other words: isNaN() converts the value to a number before testing it.

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

Ans. && stands for Logical AND.

The ‘&&’ operator turns ‘true’ if both operands are ‘true’, otherwise it returns ‘false’.

Ture && True = True

True && False = False

False && True = False

False && False = False.

|| stands for Logical or.

The || Operator turns ‘true’ if at lease one of operands is ‘true’ otherwise it will returns ‘false’

True || True = True

True || False = True

False || True = True

False || False = False.

1. What is the use of Void (0) ?

Ans. JavaScript void 0 means returning undefined void as a primitive value. You might going come across the term “JavaScript:void(0)” while going through HTML documents. It is used to prevent any side effects caused while inserting an expression in a web page.

1. Check Number Is Positive or Negative in JavaScript ?

Ans. <script>

        function checkNumber(number) {

    if (number > 0) {

        document.writeln("The number is positive.");

    } else if (number < 0) {

        document.write("The number is negative.");

    } else {

        document.write("The number is zero.");

    }

}

// Example usage:

checkNumber(9);    // Output: The number is positive.

// checkNumber(-4);   // Output: The number is negative.

// checkNumber(0);    // Output: The number is zero.

    </script>

Output : 

1. Find the Character Is Vowel or Not ?

Ans. <script>

        function isVowel(character) {

            // Check if the character is one of the vowels

            if (character === 'a' || character === 'e' || character === 'i' || character === 'o' || character === 'u' || character === 'U' || character === 'E' || character === 'I' || character === 'O' || character === 'U') {

                return true;

            } else {

                return false;

            }

        }

        document.write(isVowel('a'));  //true

        document.write(isVowel('E'));  //true

        document.write(isVowel('b'));  // false

    </script>

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

Ans. <script>

        function checkNumber(number) {

    if (number > 0) {

        document.writeln("The number is positive.");

    } else if (number < 0) {

        document.write("The number is negative.");

    } else {

        document.write("The number is zero.");

    }

}

// Example usage:

checkNumber(9);    // Output: The number is positive.

// checkNumber(-4);   // Output: The number is negative.

// checkNumber(0);    // Output: The number is zero.

    </script>

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

Ans. <script>

        function checkEvenOrOdd(number) {

            // Use the ternary operator to check if the number is even or odd

            let result = number % 2 === 0 ? "even" : "odd";

            return result;

        }

        document.write(checkEvenOrOdd(8)); // Output: even

        document.write(checkEvenOrOdd(3)); // Output: odd

    </script>

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

Ans.  <script>

        const num1 = 10;

        const num2 = 20;

        const num3 = 15;

        const max = num1 > num2 ? (num1 > num3 ? num1 : num3) : (num2 > num3 ? num2 : num3);

        document.write("The maximum number is: " + max); //max 20

    </script>

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

Ans. <script>

        const num1 = 10;

        const num2 = 20;

        const num3 = 15;

        const min = num1 < num2 ? (num1 < num3 ? num1 : num3) : (num2 < num3 ? num2 : num3);

        document.write("The minimum number is: " + min);  //min 10

    </script>

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

Ans. <script>

        const num1 = 10;

        const num2 = 20;

        const num3 = 15;

        const max = num1 > num2 ? (num1 > num3 ? num1 : num3) : (num2 > num3 ? num2 : num3);

        document.write("The maximum number is: " + max); //max 20

    </script>

1. Write to show Monday to Sunday using switch case in JS ?

Ans.  <script>

        const daynum = 5; // Change this number to get the corresponding day

        let day;

        switch (daynum) {

            case 1:

                day = "Monday";

                break;

            case 2:

                day = "Tuesday";

                break;

            case 3:

                day = "Wednesday";

                break;

            case 4:

                day = "Thursday";

                break;

            case 5:

                day = "Friday";

                break;

            case 6:

                day = "Saturday";

                break;

            case 7:

                day = "Sunday";

                break;

            default:

                day = "Invalid day number";

        }

        document.write("The day corresponding to " + daynum + " is " + day);  // output Friday

    </script>

1. Write to show Vowel or Consonant using switch case in JS?

Ans. <script>

        const letter = 'C'; // Change this letter to check whether it's a vowel or consonant

        let result;

        switch (letter.toLowerCase()) {

            case 'a':

            case 'e':

            case 'i':

            case 'o':

            case 'u':

                result = "Vowel";

                break;

            default:

                result = "Consonant";

        }

        document.write("'" + letter + "' is a " + result); //C is Consonant

    </script>

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

Ans: looping is a programming construct that allows you to execute a block of code repeatedly until a certain condition is satisfied.

1. For loop
2. While loop
3. Do-while loop etc.

Ex: <script>

        for (let i = 0; i < 5; i++) {

            console.log(i);

        }

    </script>

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

Ans: <script>

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

            document.write(i);

            document.write('<br>');

        }

    </script>

1. Write to print factorial of given number ?

Ans: <script>

        function factorial(num) {

            if (num === 0 || num === 1) {

                return 1;

            } else {

                let result = 1;

                for (let i = 2; i <= num; i++) {

                    result \*= i;

                }

                return result;

            }

        }

        const number = 5;

        document.write(`The factorial of ${number} is: ${factorial(number)}`);

    </script>

Output : The factorial of 5 is: 120

1. Write to print Fibonacci series up to given numbers ?

Ans: <script>

        function fibonacci(upTo) {

            let series = [];

            let a = 0, b = 1, temp;

            while (a <= upTo) {

                series.push(a);

                temp = a + b;

                a = b;

                b = temp;

            }

            return series;

        }

        document.write(" Fibonacci series : ");

        document.write(fibonacci(100));

    </script>

Output :



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

Ans: <script>

        function reverse(num, reversed = 0) {

            if (num === 0) {

                return reversed;

            }

            return reverse(Math.floor(num / 10), reversed \* 10 + num % 10);

        }

        const result = reverse(64728);   // result 82746

        document.write(result);

    </script>

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

Ans: <script>

        function digit(num) {

            let sum = 0;

            // Convert number to string to iterate over each digit

            const digits = num.toString();

            // Iterate over each digit and sum them up

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

                sum += parseInt(digits[i]);

            }

            return sum;

        }

        console.log(`Summation of digits of is ${digit(1523)}`);  //Answer 11

    </script>

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

Ans: <script>

        function sum(num) {

            // Convert number to string to access individual digits

            const numString = num.toString();

            // Extract the first and last digits

            const first = parseInt(numString.charAt(0));

            const last = parseInt(numString.charAt(numString.length - 1));

            // Calculate the sum

            const sum = first + last;

            return sum;

        }

        // Example usage:

        const num = 1234;

        document.write(`Sum of first and last digit of ${num}: ${sum(num)}`);  //output 5

    </script>

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

1 1 1 1 1

2 1 2 4 8

3 1 3 9 27

4 1 4 16 64

5 1 5 25 125

<script>

        // Number of rows

        const numRows = 5;

        // Outer loop for rows

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

            let row = ' ';

            // Inner loop for columns

            for (let j = 1; j <= 3; j++) {

                if (j === 1) {

                    row += i + ' 1 ';

                    row += i + ' ';

                } else {

                    row += Math.pow(i, j) + ' ';

                }

            }

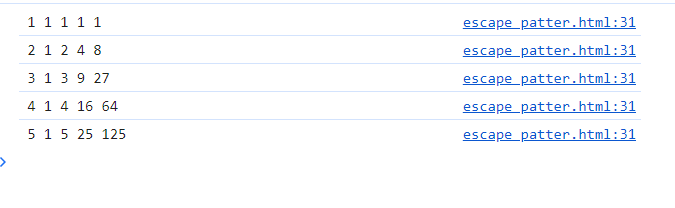
            // Print the row

            console.log(row);

        }

    </script>

Output :



1. Use pattern in console.log in JS?

* 2nd pattern

Ans: <script>

    for(var a=0; a<=4; a++)   //first loop for increment value by \*

    {

       var i=1;    //Intialize variable i to 1

        for(var b=0; b<=a; b++)  // second loop is for starting loop with fisrt \*

        {

            if(i%2==0)     //check condition

            {

                document.write("0")   // if true print 0

            }

            else

            {

                document.write("1")    // if false print 1

            }

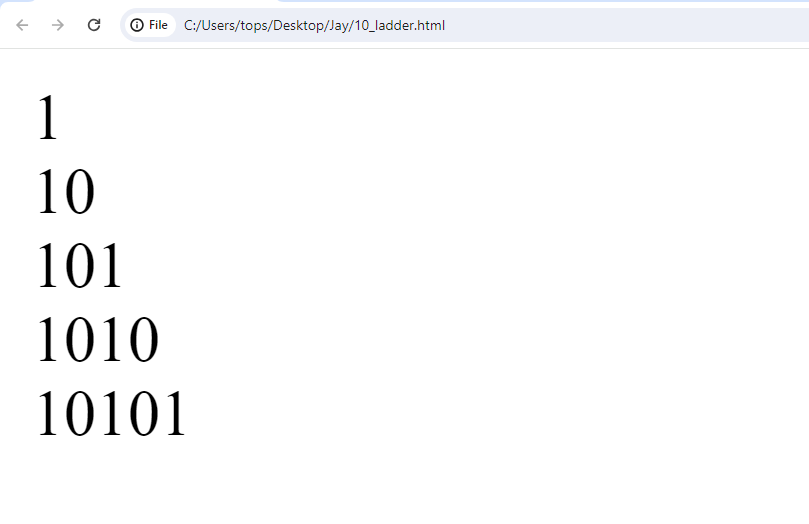
            i++;     // Increment value i by 1

        }

        document.writeln(`<br>`)    // to break the line and start new line

    }

</script>



* 3rd Pattern

<script>

    var AS=65;   //Initialize ASCII value for A

    var str=""   //Create variable to store Character

    for(var a=0; a<=5; a++)   //first loop for increment value by

    {

        for(var b=0; b<=a; b++)  // second loop is for starting loop with fisrt \*

        {

            str = String.fromCharCode(AS)    //Use fromCharCode Function to convert ASCII value to Character.

            AS++;      // Increment ASCII value from A to ....

            document.writeln(str)   // Print Character

        }

        document.writeln(`<br>`)    // to break the line and start new line

    }

</script>

Output:



* 4th Pattern

<script>

    var digi=1;   //Initialize digit to start from

    for(var a=0; a<=4; a++)   //first loop for increment value by

    {

        for(var b=0; b<=a; b++)  // second loop is for starting loop with fisrt \*

        {

            document.writeln(digi)   // Print Digits

            digi++;     // Increment Digits

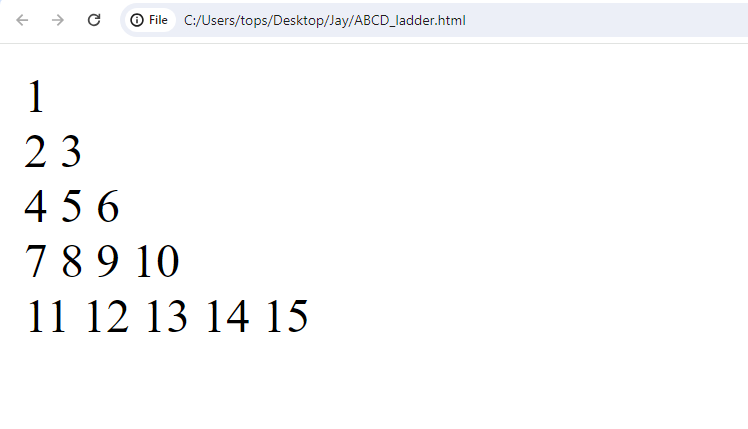
        }

        document.writeln(`<br>`)    // to break the line and start new line

    }

</script>

Output:



* 5th Pattern

<script>

    for(var a=0; a<=4; a++)   //first loop for increment value by \*

    {

        // document.writeln("\*")

        for(var b=0; b<=a; b++)  // second loop is for starting loop with fisrt \*

        {

            document.writeln("\*");  // printing \*

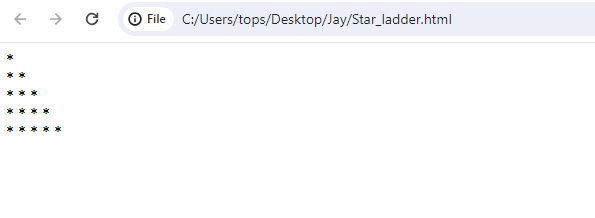
        }

        document.writeln(`<br>`)    // to break the line and start new line

    }

</script>

Output :



1. Accept 3 numbers from user using while loop and check each numbers palindrome ?

Ans: <body>

    <input type="text" id="val">

    <button id="count" onclick="palindrome(), counter()">Click Me</button>

    <p id="ans"></p>

    <script>

        var c = 0;

        function palindrome() {

            var rev = 0;

            var data = document.getElementById("val").value;

            document.getElementById("ans").innerText = data;

                var r, rev = 0, n, copy;

                n = data;

                copy = n;

                while (n != 0) {

                    r = n % 10;

                    rev = rev \* 10 + r;

                    n = parseInt(n / 10);

                }

                if (copy == rev) {

                    document.getElementById("ans").innerText = data +" is a Palindrome number.";

                }

                else {

                    document.getElementById("ans").innerText = data +" is a Not a Palindrome number.";

                }

            }

            function counter() {

                if (c == 3) {

                    document.getElementById("count").disabled = true;

                    document.write("You Reached at Your limit.......");

                }

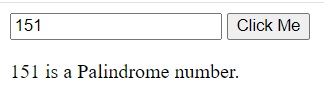
                c++;

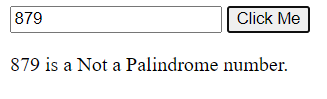
            }

    </script>

</body>

Output:







1. 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 ?

Ans: <script>

        function getCurrentDateTime() {

            // Array of days

            const days = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"];

            // Get current date and time

            const now = new Date();

            // Get day of the week

            const dayIndex = now.getDay();

            const day = days[dayIndex];

            // Get current hour

            let hour = now.getHours();

            const ampm = hour >= 12 ? 'PM' : 'AM';    //Ternary Operator

            // Convert hour to 12-hour format

            if (hour > 12) {

                hour -= 12;

            } else if (hour === 0) {

                hour = 12;

            }

            // Get current minutes and seconds

            const minutes = now.getMinutes();

            const seconds = now.getSeconds();

            // Display the result

            document.write(`Today is ${day}. Current Time is ${hour} ${ampm}: ${minutes} : ${seconds}`);

        }

        // Call the function to display current day and time

        getCurrentDateTime();

    </script>

Output :



1. Write a JavaScript program to get the current date ?

Ans: <script>

        function getCurrentDate() {

            // Get current date

            const now = new Date();

            // Extract year, month, and day

            const year = now.getFullYear();

            const month = now.getMonth() + 1; // Month starts from 0

            const day = now.getDate();

            const currentDate = `${day}-${month}-${year}`;

            // Return the current date

            return currentDate;

        }

        // Call the function to get the current date and display it

        const currentDate = getCurrentDate();

        document.write("Current Date is:", currentDate);

    </script>

Output :



1. Write a JavaScript program to compare two objects ?

Ans: <script>

        function compareObjects(obj1, obj2) {

            // Get the keys of both objects

            const keys1 = Object.keys(obj1);

            const keys2 = Object.keys(obj2);

            // Check if the number of keys is the same

            if (keys1.length !== keys2.length) {

                return false;

            }

            // Check if all keys and their corresponding values are the same

            for (let key of keys1) {

                // If the key is not present in obj2 or the values are not equal, return false

                if (!obj2.hasOwnProperty(key) || obj1[key] !== obj2[key]) {

                    return false;

                }

            }

            // If all checks pass, return true

            return true;

        }

        // Example objects for testing

        const obj1 = { a: 1, b: 2, c: 3 };

        const obj2 = { a: 1, b: 2, c: 3 };

        const obj3 = { a: 1, b: 2, c: 4 };

        // Compare obj1 and obj2

       document.write(compareObjects(obj1, obj2)); // Output: true

       document.write("<br>");

        // Compare obj1 and obj3

        document.write(compareObjects(obj1, obj3)); // Output: false

    </script>

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

Ans: <script>

        function convert(data) {

            // Extract column headers from the keys of the first object

            const headers = Object.keys(data[0]);

            // Construct CSV header row

            const headerRow = headers.join(',');

            // Construct CSV data rows

            const rows = data.map(obj => {

                return headers.map(key => {

                    // Handle values that might contain commas by enclosing them in double quotes

                    const value = obj[key];

                    return typeof value === 'string' && value.includes(',') ? `"${value}"` : value;

                }).join(',');

            });

            // Combine header row and data rows

            const csv = [headerRow, ...rows].join('\n');

            return csv;

        }

        // Example array of objects

        const data = [

            { name: 'John', age: 30, city: 'New York' },

            { name: 'Jane', age: 25, city: 'Los Angeles' },

            { name: 'Doe', age: 40, city: 'Chicago' }

        ];

        // Convert array of objects to CSV string

        const csvString = convert(data);

        // Display CSV string

        document.write(csvString);

    </script>

Output :



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

Ans: <script>

        function Letter(str) {

            // Check if the input string is not empty

            if (str.length === 0) {

                return str; // Return the string unchanged if it's empty

            }

            // Capitalize the first letter and concatenate it with the rest of the string

            return str.charAt(0).toUpperCase() + str.slice(1);

        }

        // Example string

        const inputString = "hello world";

        // Capitalize the first letter of the string

        const capitalString = Letter(inputString);

        // Display the result

        document.write(capitalString); // Output: "Hello world"

    </script>

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

Ans: <script>

        function isArray(variable) {

            // Use the Array.isArray() method to check if the variable is an array

            return Array.isArray(variable);

        }

        const arr = [1, 2, 3];

        const str = "hello";

        const num = 42;

        // Check if the variables are arrays

        document.write(isArray(arr)); // Output: true

        document.write("<br>");

        document.write(isArray(str)); // Output: false

        document.write("<br>");

        document.write(isArray(num)); // Output: false

    </script>

1. Write a JavaScript program to clone an array ?

Ans: <script>

            // Use the spread operator (...) to create a new array with the same elements

            return [...arr];

        }

        // Example array

        const originalArray = [1, 2, 3, 4, 5];

        // Clone the array

        const clonedArray = cloneArray(originalArray);

        // Display original and cloned arrays

        document.write("Original Array:", originalArray);

        document.write("<br>")

        document.write("Cloned Array:", clonedArray);

    </script>

Output :



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

Ans: Declaring methods directly in JavaScript objects can lead to a drawback known as method duplication or method redundancy. This occurs when multiple instances of the same object are created, each containing its own copy of the methods. This can consume more memory than necessary, especially if the methods are large or complex.

Another potential issue is difficulty in maintaining and updating the methods. If a method needs to be modified or fixed, it must be changed in each instance of the object, which can be complex and error-prone, particularly in large program or project.

Additionally, declaring methods directly within objects can make it harder to achieve certain design patterns such as inheritance and polymorphism. Objects created using the same constructor function cannot share methods unless those methods are attached to the prototype of the constructor function.

1. Print the length of the string on the browser console using console.log() ?\

Ans: <script>

        const myString = "Hello, world!";

        // Get the length of the string

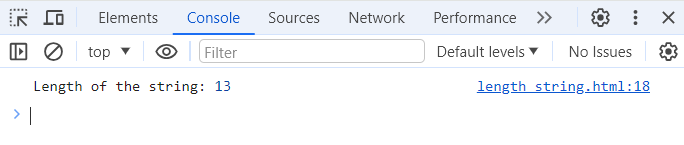
        const lengthOfString = myString.length;

        // Print the length on the browser console

        console.log("Length of the string:", lengthOfString);

    </script>

Output:



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

Ans:

<script>

        let myString = "Hello, world!";

        // Convert all characters to capital letters

        const capitalizedString = myString.toUpperCase();

        // Print the capitalized string on the browser console

        document.write("Capitalized string: ", capitalizedString);

    </script>

Output:   


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

Ans: Declaring methods directly in JavaScript objects can lead to a drawback known as method duplication or method redundancy. This occurs when multiple instances of the same object are created, each containing its own copy of the methods. This can consume more memory than necessary, especially if the methods are large or complex.

Another potential issue is difficulty in maintaining and updating the methods. If a method needs to be modified or fixed, it must be changed in each instance of the object, which can be complex and error-prone, particularly in large program or project.

Additionally, declaring methods directly within objects can make it harder to achieve certain design patterns such as inheritance and polymorphism. Objects created using the same constructor function cannot share methods unless those methods are attached to the prototype of the constructor function.

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

Ans: <script>

        function getCurrentDate(format) {

            const currentDate = new Date();

            let day = currentDate.getDate();

            let month = currentDate.getMonth() + 1; // Months are zero based

            const year = currentDate.getFullYear();

            // Add leading zero if day or month is less than 10

            if (day < 10) {

                day = '0' + day;

            }

            if (month < 10) {

                month = '0' + month;

            }

            // Return the date based on the format

            switch (format) {

                case 'mm-dd-yyyy':

                    return month + '-' + day + '-' + year;

                case 'mm/dd/yyyy':

                    return month + '/' + day + '/' + year;

                case 'dd-mm-yyyy':

                    return day + '-' + month + '-' + year;

                case 'dd/mm/yyyy':

                    return day + '/' + month + '/' + year;

                default:

                    return 'Invalid format';

            }

        }

        // Usage examples

        document.write("Current date in mm-dd-yyyy format:", getCurrentDate('mm-dd-yyyy'));

        document.write("<br>");

        document.write("Current date in mm/dd/yyyy format:", getCurrentDate('mm/dd/yyyy'));

        document.write("<br>");

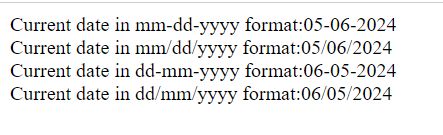
        document.write("Current date in dd-mm-yyyy format:", getCurrentDate('dd-mm-yyyy'));

        document.write("<br>");

        document.write("Current date in dd/mm/yyyy format:", getCurrentDate('dd/mm/yyyy'));

    </script>

Output:



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

Ans:

<script>

        const text = "30 Days Of JavaScript";

        const position = text.indexOf('a');

        document.write("Position of the first occurrence of 'a':", position);

    </script>

Output:



1. Use lastIndexOf to determine the position of the last occurrence of a in 30 Days Of JavaScript ?

Ans: <script>

        const text = "30 Days Of JavaScript";

        const position = text.lastIndexOf('a');

        document.write("Position of the first occurrence of 'a':", position);

    </script>

Output:



1. Form Validtion in JS?
2. Form in Email, number, Password, Validation?
3. Dynamic Form Validation in JS?

Ans: No. 55, No. 56, and No.57 have same code and Answer.

Ans:

<body>

    <form name="myform" action="" method="post" onsubmit="return validate(), success()">

        <table border="2" align="center">

            <tr>

                <th> Registration Page </th>

            </tr>

            <tr>

                <th>User Fist Name:-</th>

                <td>

                    <input type="text" name="ufn">

                    <span style="color:red" id="msg1"></span>

                </td>

            </tr>

            <tr>

                <th>User Name:-</th>

                <td><input type="text" name="un"></td>

            </tr>

            <tr>

                <th>Password:-</th>

                <td><input type="password" name="pass"></td>

            </tr>

            <tr>

                <th>Confirm Password:-</th>

                <td><input type="password" name="cpass"></td>

            </tr>

            <tr>

                <th>Email Address:-</th>

                <td><input type="text" name="email"></td>

            </tr>

            <tr>

                <th>Phone No:-</th>

                <td><input type="text" name="pno"></td>

            </tr>

            <tr>

                <td><input type="submit" name="submit" /></td>

            </tr>

        </table>

    </form>

</body>

<script>

        // first Make one function

        function validate()  // function name

        {

            var ufn = document.forms["myform"]["ufn"].value;

            if (ufn == "" || ufn == null)  // for null condition

            {

                //alert('Please fill out the User First Name');  // alert msg

                document.getElementById('msg1').innerHTML = "Please fill out the User First Name";

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            else {

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

            }

            // /^ $/

            //alpha=/^[A-Za-z]+$/

            //var phone=/^[0-9]{10,11}$/;

            //var mail=/^[a-zA-Z0-9\_]+@[a-zA-Z]+\.[a-zA-Z]{2,4}$/;

            var alpha = /^[A-Za-z]+$/;  // /^[A-Za-z]{3,8}$/

            if (!alpha.test(ufn)) {

                alert('Please fill only alpha User First Name');  // alert msg

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            var un = document.forms["myform"]["un"].value;

            if (un == "" || un == null)  // for null condition

            {

                alert('Please fill out the User Name');  // alert msg

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            var pass = document.forms["myform"]["pass"].value;

            if (pass == "" || pass == null)  // for null condition

            {

                alert('Please fill out the pass');  // alert msg

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            if (!(pass.length <= 8 && pass.length >= 3)) {

                alert('Please,provide min 3 & max 8 char in pass');

                return false;

            }

            var cpass = document.forms["myform"]["cpass"].value;

            if (cpass == "" || cpass == null)  // for null condition

            {

                alert('Please fill out the cpass');  // alert msg

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            if (pass == cpass) {

            }

            else {

                alert('Please Enter Same values');  // alert msg

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            var email = document.forms["myform"]["email"].value;

            if (email == "" || email == null)  // for null condition

            {

                alert('Please fill out the email');  // alert msg

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            var mail = /^[a-zA-Z0-9\_]+@[a-zA-Z]+\.[a-zA-Z]{2,4}$/;

            if (!mail.test(email)) {

                alert('Please fill proper email id');  // alert msg

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            var pno = document.forms["myform"]["pno"].value;

            if (pno == "" || pno == null)  // for null condition

            {

                alert('Please fill out the pno');  // alert msg

                return false;   //return false means msg show and again on same page with value not refresh page

            }

            var phone = /^[0-9]{10,11}$/;

            if (!phone.test(pno)) {

                alert('Please fill only digits in PNO');  // alert msg

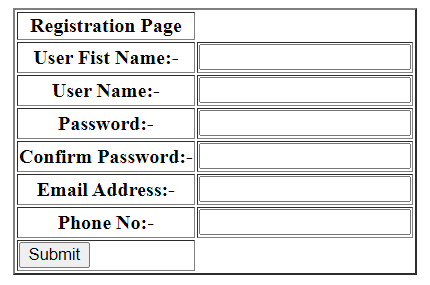
                return false;   //return false means msg show and again on same page with value not refresh page

            }

        }

    </script>

Output:



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

Ans: There are many types of event in Javascript such as..

1. Mouse Event:
2. Click event
3. Mouseover event
4. Mouseout event
5. Mousemove event
6. Keyboard Event:
7. Keydown Event
8. Keyup Event
9. Keypress event
10. Form Event
11. Submit Event
12. Change Event
13. Focus Event
14. Blur event etc…..

Example:

<body>

    <button id="myButton">Click me</button>

    <script>

        // Get the button element

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

        // Attach an event listener for the 'click' event

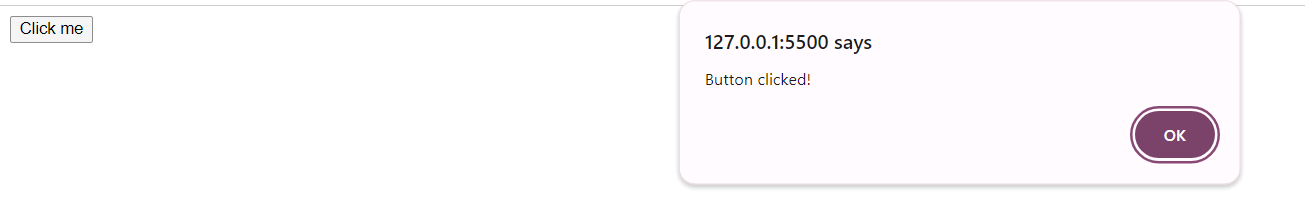
        button.addEventListener('click', function () {

            alert('Button clicked!');

        });

    </script>

</body>



1. What is Bom vs Dom in JS ?

Ans: BOM : BOM stands for Browser Object Model.

The BOM represents additional objects and interfaces provided by the web browser, which are not related to the document structure but are essential for browser interaction and control. It includes object such as window, navigator, screen, location, history, document etc..

The BOM provides functionality for tasks such as controlling browser behavior (e.g., opening new windows or tabs, resizing windows), managing browser history, detecting browser and operating system information, and accessing client-side resources.

Unlike the DOM, the BOM is not standardized by any official specification and may vary between different browsers.

DOM : DOM stands for Document Object Model.

The DOM represents the structure of an HTML or XML document as a tree-like structure, where each node represents an element, attribute, or piece of text within the document.

It provides a way for JavaScript to interact with and manipulate the content, structure, and style of a web page dynamically. W ith the DOM, you can access, create, modify, and delete elements and their attributes, as well as handle events triggered by user interactions or other actions.

T he DOM is standardized by the World Wide Web Consortium (W3C) and is implemented consistently across different browsers.

1. Array vs object defences in JS ?

Ans: Array : Arrays are best when the elements are number.

The data insides an array is known as Element

The Element can be meupulate using [].

The element can be deleted using pop()function.

Object: Objects are best when the element are string(text).

The data insides object known as properties which consists of a key.

The properties can be manipulated using both .DOT notation and [].

The key or properties can be deleted by using the delete keyword.

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

Ans:

<script>

        const myString = "Hello, world! This is a sample string.";

        const myArray = myString.split(' ');

        document.write(myArray);

    </script>

Output:



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

<script>

        const myString = "Hello, this is a JavaScript example.";

        if (myString.includes("Script")) {

            document.write("The string contains the word 'Script'.");

        } else {

            document.write("The string does not contain the word 'Script'.");

        }

    </script>

Output :



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

Ans:

 <script>

        const myString = "Hello, THIS IS a Sample STRING.";

        const lowerCaseString = myString.toLowerCase();

        document.write(lowerCaseString);  //O/P : hello, this is a sample string.

    </script>

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

Ans:

<script>

        const myString = '30 Days of JavaScript';

        const character = myString.charAt(15);

        document.write("Character at index 15:", character);  //Output := Character at index 15:S

    </script>

1. copy to one string to another string in JS ?

Ans: <script>

        const originalString = "This is the original string.";

        // Copying the content to another string

        const copiedString = originalString;

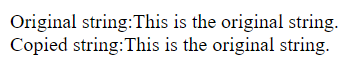
        document.write("Original string:", originalString);

        document.write("<br>")

        document.write("Copied string:", copiedString);

    </script>

Output :



1. Find the length of a string without using libraryFunction ?

Ans: <script>

        const myString = "This is a sample string.";

        let length = 0;

        // Iterate over each character in the string

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

            // Increment the length for each character

            length++;

        }

        document.write("Length of the string:", length);   // Output : Length of the string:24

    </script>

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 provides user to interact with web pages as per the requirement.

1. What is the use of isNaN function?

Ans: The isNaN() function determines whether a value is NaN when converted to a number. Because coercion inside the isNaN() function can be surprising, you may alternatively want to use Number.isNaN() .

isNaN() method returns true if a value is Not-a-Number. Number.isNaN() returns true if a number is Not-a-Number. In other words: isNaN() converts the value to a number before testing it.

1. What is negative Infinity?

Ans: NEGATIVE\_INFINITY is a special numeric value that is returned when an arithmetic operation or mathematical function generates a negative value greater than the largest representable number in JavaScript (i.e., more negative than -Number. MAX\_VALUE) . JavaScript displays the NEGATIVE\_INFINITY value as -Infinity.

When divided by itself or positive infinity, negative infinity return NaN.

Syntax : Number.NEGATIVE\_INFINITY

1. Which company developed JavaScript?

Ans: JavaScript was invented by Brendan Eich in 1995. It was developed for Netscape 2, and become the ECMA-262 standard in 1997. After Netscape handled JavaScript over to ECMA, the Mozilla Foundation continued to develop JavaScript for the Firefox Browser.

1. What are undeclared and undefined variables?

Ans: When a variable is “undeclared,” it means that it has not been declared or defined in the current scope. In other words, it hasn’t been given a value or a type. This can happen if you try to use a variable without declaring it first, or if you misspell the variable name.

For example, consider the following code:

console.log(x);

* Unlike “undeclared” variables, “undefined” variables have been declared in the current scope, but have not been given a value. In other words, they have been initialized, but their value is not defined.

For example:

let x;  
console.log(x);

1. Write the code for adding new elements dynamically?

Ans: New Element can be dynamically created in JavaScript with the help of createElement() Method. The attributes of the created element can be set using the setAttribute() method.

Example: function addItem(){

let type = document.getElementById(“type”).value;

let value = document.getElementById(“value”).value;

type = document.createElement(type);

}

1. What is the difference between ViewState and SessionState?

Ans: ViewState

* Maintained at page level only.
* It will remain value in the event of a postback operation occouring.
* Information is stored on the client’s end only.
* Used to allow the persistence page-instance-specific data

SessionState

* Maintained at session level
* User data remains in the server. Data available until browser is closed.
* Information is stored on the server.
* Used for the persistence user-specific data on the server’s end.

1. What is === operator?

Ans: === is known as strict equality. This operator checks whether its two operands are equal, returning a Boolean result. It does not loose its quality.

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

Ans: There are two common approaches that allow us to change style/class of an element.

1. Style.property :- document.getElementById(“Id”).style.property = new\_style
2. Changing the class itself. :- we can use two properties that can be used to manipulate the classes. There are three method use. Add() , remove(), toggle() Method
3. The ClassList Property : The classList property is a read-only property that returns the class name of an element as a DOMTokenList object.

Document.getElementById(“Id”).classList

1. The className Property: This property is used to set the current class of the element to the specified class.

Document.getElementById(“Id”).className = class.

Example:

Ans: <body>

        <p id="myParagraph">This is a paragraph.</p>

        <script>

            // Get the element by its ID

            const paragraph = document.getElementById('myParagraph');

            // Change inline styles

            paragraph.style.color = 'blue'; // Change text color

            paragraph.style.fontSize = '20px'; // Change font size

            // Add or remove classes

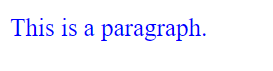
            paragraph.classList.add('red-text'); // Add class

            paragraph.classList.remove('red-text'); // Remove class

        </script>

    </body>

Output:



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

Ans: In JavaScript(). Function or method is used to convert the passed-in string parameter or value to an integer value itself. This function returns an integer of base which is specified in the second argument of the parseInt() Function.

* parseInt() : this is used to accept the string and radix parameter and convert it into an integer. parseInt(Value, radix)
* Number() : This is used to convert any primitive data type to a number, If it is not convertible it returns NAN.
* Unary Operator: This is used to convert a string, Boolean, And Non-string to a number.

1. What are all the looping structures in JavaScript?

Ans: 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 while a specified condition is true.

1. How to read and write a file using JavaScript?

Ans: The fs.readFile() and rs.writeFile() method are used to read and write a file.

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

* File\_name : It’s a string, a Buffer, a URL, or a file description integer that specifies the location of the file to be read.
* Encoding : It stores the file’s encoding. ‘UTF8’ is the default setting.
* Callback Function that is invoked after the file has been read.
* Fs.writeFile(file\_name, data, options, callback)
* File\_name : It’s a string, a Buffer, a URL, or a file description integer that specifies the location of the file to be written.
* Data : The data that will be sent to the file is string, Buffer, TypedArray, Or DataView.
* Options : It’s a string that may be used to include optional output options.
* Encoding : It stores the file’s encoding. ‘UTF8’ is the default setting.
* Callback Function that is invoked after the file has been read.

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

Ans: In JavaScript(). Function or method is used to convert the passed-in string parameter or value to an integer value itself. This function returns an integer of base which is specified in the second argument of the parseInt() Function.

* parseInt() : this is used to accept the string and radix parameter and convert it into an integer. parseInt(Value, radix)
* Number() : This is used to convert any primitive data type to a number, If it is not convertible it returns NAN.
* Unary Operator: This is used to convert a string, Boolean, And Non-string to a number.

1. What is the function of the delete operator?

* The delete operator removes a property from an object. If the property’s value is an object and there are no more references to the object, The object held by the property is released automatically.

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

Ans: Alert Box - An Alert box is often used if you want to make sure information comes through to the user.

Syntax: windows.alert(“something….”);

* Confirm Box – A confirm box is often used if you want the user to verify or accept something. When a confirm box pops up, The user will have to click either “Ok” or “Cancel” OR “True” or “False”.

Syntax: windows.confirm(“something….”);

* Prompt Box – A prompt box is often used if you want the user to input a value before entering a page. When a prompt confirm box pops up, The user will have to click either “Ok” or “Cancel” OR “True” or “False” to proceed after entering an input value.

Syntax: windows.prompt(“something….”,”Default Text”);

1. What is the use of Void (0)?

Ans: JavaScript void 0 means returning undefined void as a primitive value. You might going come across the term “JavaScript:void(0)” while going through HTML documents. It is used to prevent any side effects caused while inserting an expression in a web page.

1. 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 represent the current location of the document. We can change the URL of a window by accessing it.

1. What are the disadvantages of using innerHTML in JavaScript?

Ans: Event handlers attached to any DOM element are preserved.

* Replacement is done everywhere.
* It is not possible to append innerHTML.
* Breaks the documents.
* Used for cross-site scripting.

1. Create password field with show hide functionalities.

Ans:

<body>

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

    <input type="password" id="password" name="password">

    <input type="checkbox" id="showPasswordCheckbox">

    <label for="showPasswordCheckbox">Show password</label>

    <script>

        const passwordInput = document.getElementById('password');

        const showPasswordCheckbox = document.getElementById('showPasswordCheckbox');

        // Add event listener to the checkbox

        showPasswordCheckbox.addEventListener('change', function () {

            // If the checkbox is checked, show the password

            if (this.checked) {

                passwordInput.type = 'text';

            } else {

                // If the checkbox is not checked, hide the password

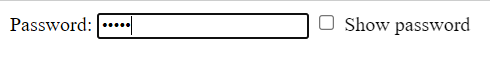
                passwordInput.type = 'password';

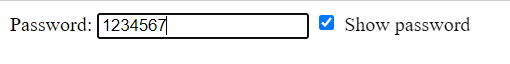
            }

        });

    </script>

Output :





1. Create basic math operation in JS Create

Ans:

 <style>

        .calculator {

            width: 250px;

            margin: 50px auto;

            border: 1px solid #ccc;

            padding: 10px;

            text-align: center;

        }

        .button {

            width: 50px;

            height: 50px;

            margin: 5px;

            font-size: 20px;

            cursor: pointer;

        }

    </style>

</head>

<body>

    <div class="calculator">

        <input type="text" id="display" readonly style="height: 30px; width: 170px;">

        <br>

        <br>

        <br>

        <button class="button" onclick="appendToDisplay('7')">7</button>

        <button class="button" onclick="appendToDisplay('8')">8</button>

        <button class="button" onclick="appendToDisplay('9')">9</button>

        <br>

        <button class="button" onclick="appendToDisplay('4')">4</button>

        <button class="button" onclick="appendToDisplay('5')">5</button>

        <button class="button" onclick="appendToDisplay('6')">6</button>

        <br>

        <button class="button" onclick="appendToDisplay('1')">1</button>

        <button class="button" onclick="appendToDisplay('2')">2</button>

        <button class="button" onclick="appendToDisplay('3')">3</button>

        <button class="button" onclick="appendToDisplay('0')">0</button>

        <button class="button" onclick="appendToDisplay('.')">.</button>

        <button class="button" onclick="clearDisplay()">C</button>

        <button class="button" onclick="operation('+')">+</button>

        <button class="button" onclick="operation('-')">-</button>

        <button class="button" onclick="operation('\*')">\*</button>

        <button class="button" onclick="calculate()">=</button>

        <button class="button" onclick="operation('/')">/</button>

        <button class="button" onclick="operation('%')">%</button>

    </div>

    <script>

        let displayValue = '';

        function appendToDisplay(value) {

            displayValue += value;

            document.getElementById('display').value = displayValue;

        }

        function clearDisplay() {

            displayValue = '';

            document.getElementById('display').value = displayValue;

        }

        function operation(operator) {

            if (displayValue !== '') {

                displayValue += operator;

                document.getElementById('display').value = displayValue;

            }

        }

        function calculate() {

            if (displayValue !== '') {

                const result = eval(displayValue);

                document.getElementById('display').value = result;

                displayValue = result.toString();

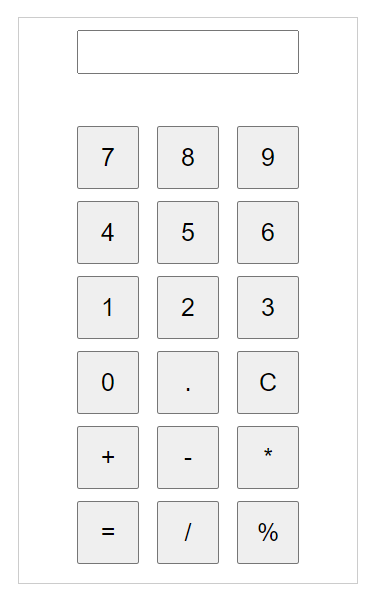
            }

        }

    </script>

</body>

Output :



1. Create Result

Ans:     <style>

        input{

            height: 30px;

            margin-top: 15px;

        }

        button{

            height: 50px;

            width: 100px;

            text-align: center;

        }

        table{

            border: 2px solid black;

        }

    </style>

</head>

<body>

    <table>

        <h3>Marksheet for Information Technology</h3>

        <tr>

            <Span>Result</Span>

            <td><Span>1. C Language </Span></td>

            <td><input type="number" id="c"><br><br></td>

        </tr>

        <tr>

            <td><Span>2. C++ Language </Span></td>

            <td><input type="number" id="cpp"><br><br></td>

        </tr>

        <tr>

            <td><Span>3. Database </Span></td>

            <td><input type="number" id="db"><br><br></td>

        </tr>

        <tr>

            <td><Span>4. HTML </Span></td>

            <td><input type="number" id="html"><br><br></td>

        </tr>

        <tr>

            <td><Span>5. CSS </Span></td>

            <td><input type="number" id="css"><br><br></td>

        </tr>

        <tr>

            <td><Span>6. PHP </Span></td>

            <td><input type="number" id="php"><br><br></td>

        </tr>

        <tr>

            <td><Span>7. Core Java </Span></td>

            <td><input type="number" id="java"><br><br></td>

        </tr>

        <tr>

            <td colspan="2"><Button onclick="total()">Result</Button><br><br></td>

        </tr>

        <tr>

            <td><p id="total">Total is : </p></td>

            <td><p id="percent" >Percentage is : </p></td>

        </tr>

    </table>

    <script>

        function total() {

            var c = parseInt(document.getElementById("c").value);

            var cpp = parseInt(document.getElementById("cpp").value);

            var database = parseInt(document.getElementById("db").value);

            var html = parseInt(document.getElementById("html").value);

            var css = parseInt(document.getElementById("css").value);

            var php = parseInt(document.getElementById("php").value);

            var java = parseInt(document.getElementById("java").value);

            var sum = (c+cpp+database+html+css+php+java);

            var per = Math.ceil((sum \* 100)/350);

            document.getElementById("total").innerHTML ="Total is : "+ sum +"/350";

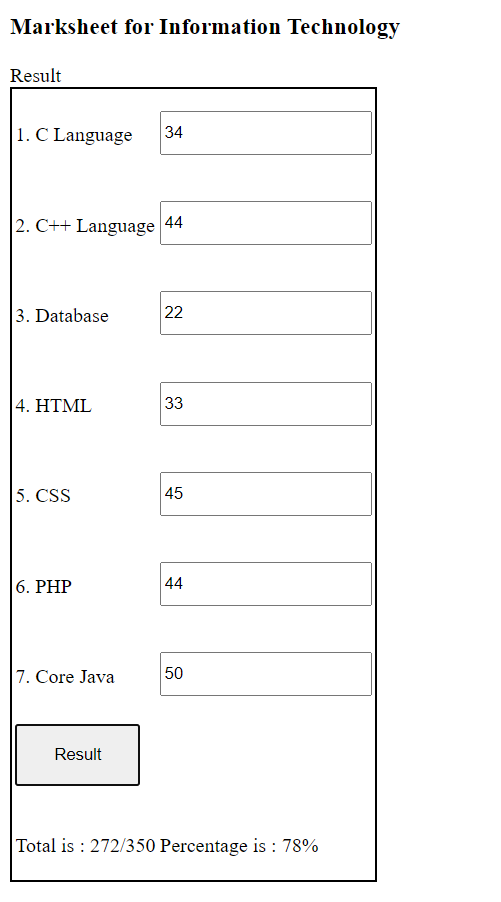
            document.getElementById("percent").innerHTML = "Percentage is : " + per +"%";

        }

    </script>

</body>

Output :



1. Create Slider using JavaScript

Ans:

<style>

        .slider{

            position: relative;

        }

        .box-container {

            display: flex;

            overflow: hidden;

        }

        .box {

            width: 50%;

            height: 90vh;

            flex:   0 0 auto;

            border: 1px solid black;

            margin: 10px;

            transition: transform 1.5 ease-out;

        }

        button {

            position: absolute;

            top: 50%;

            background: none;

            border: none;

            font-size: 24px;

            cursor: pointer;

        }

        .prev {

            left: 0px;

        }

        .next {

            right: 0px;

        }

    </style>

</body>

<div id="slider">

<div class="box-container">

    <div class="box" style="background-color: orange;"></div>

    <div class="box" style="background-color: red;"></div>

    <div class="box" style="background-color: blue;"></div>

    <div class="box" style="background-color: pink;"></div>

</div>

<button class="prev" onclick="prevSlide()">❮</button>

<button class="next" onclick="nextSlide()">❯</button>

</div>

<script>

    let slideIndex = 0;

    let slides = document.querySelectorAll('.box');

    let prev = document.querySelector('.prev');

    let next = document.querySelector('.next');

    function showSlide(index) {

        if (index < 0) {

            index = slides.length - 1;

        } else if (index >= slides.length) {

            index = 0;

        }

        slides.forEach(slide => {

            slide.style.transform = `translateX(-${index \* 50}%)`;

        });

        slideIndex = index;

    }

    function nextSlide() {

        showSlide(slideIndex + 1);

    }

    function prevSlide() {

        showSlide(slideIndex - 1);

    }

</script>

Outout :

