**Assignment**

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

Ans. JavaScript (JS) is a cross-platform, object-oriented programming language used by developers to make web pages interactive.

Add dynamic behaviour to web pages, such as validation of forms, control multimedia, animate images, and almost everything else.

Create web and mobile apps.

Build web servers and develop server applications.

Create browser games.

Q.2 How many type of Variable in JavaScript?

Ans.2 There Are Three Types of variables

Var - This is the oldest and most commonly used type of variable. It is function-scoped, meaning that it can be accessed anywhere within the function in which it is declared.

Let-

- This is a newer type of variable that was introduced in ES6. It is block-scoped, meaning that it can only be accessed within the block of code in which it is declared.

Const-

- This is also a newer type of variable that was introduced in ES6. It is used to declare constant variables, which means that their values cannot be changed once they are assigned.

Q.3 Define a Data Types in js?

Ans.3 Primitive

[String](https://www.shiksha.com/online-courses/articles/javascript-data-types-with-examples/#str)

[Number](https://www.shiksha.com/online-courses/articles/javascript-data-types-with-examples/#num)

[Boolean](https://www.shiksha.com/online-courses/articles/javascript-data-types-with-examples/#boo)

[Undefined](https://www.shiksha.com/online-courses/articles/javascript-data-types-with-examples/#und)

[Null](https://www.shiksha.com/online-courses/articles/javascript-data-types-with-examples/#nul)

[Symbol](https://www.shiksha.com/online-courses/articles/javascript-data-types-with-examples/#sym)

Non- Primitive

[Object](https://www.shiksha.com/online-courses/articles/javascript-data-types-with-examples/#obj)

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

Ans.4 function mul(x) {

return function (y) {

return function (z) {

return x \* y \* z;

};

};

}

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

Ans.5 A declared variable is one that has been created using the var, let, or const keyword.

**var** x;  
console.log(x); // undefined

An undeclared variable is one that has not been created using any of these keywords.

console.log(x); // Reference Error: x is not defined

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:

Ans.6

<!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>

    <div class="main"></div>

    <script>

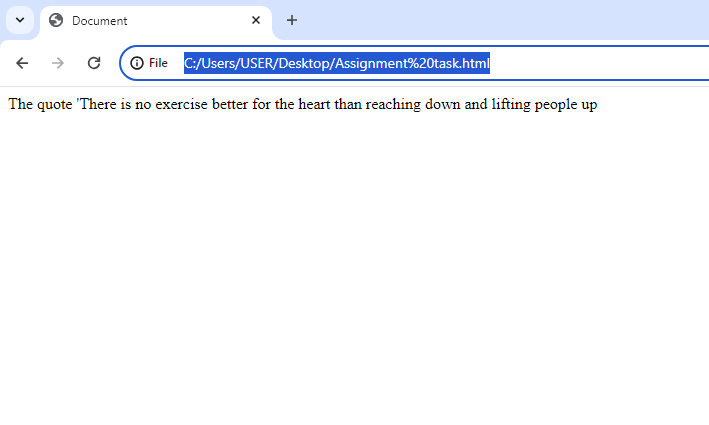
        console.log("The quote 'There is no exercise better for the heart than reaching down and lifting people up");

        document.write("The quote 'There is no exercise better for the heart than reaching down and lifting people up");

    </script>

</body>

</html>



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

Ans.7

<!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>

    <div class="main"></div>

    <script>

        // check if the type of '10' is exactly equal to 10

        if (typeof '10' !== typeof 10) {

            // Convert '10' to a number

            const num = parseInt('10');

            // Print the converted number

            console.log(num);// Output:10

        }

        else {

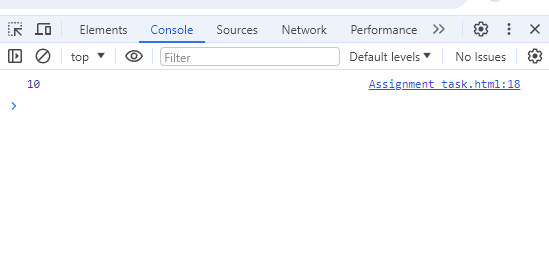
            console.log("10' is already of the same type as 10.");

        }

    </script>

</body>

</html>



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

Ans.8

<!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>

    <div class="main"></div>

    <script>

        // Function to calculate the area of a triangle

        function calculateTriangleArea(base, height) {

            // Formula to calculate area of a triangle: 0.5 \* base \* height

            var area = 0.5 \* base \* height;

            return area;

        }

        // Example usage:

        var base = 5;

        var height = 8;

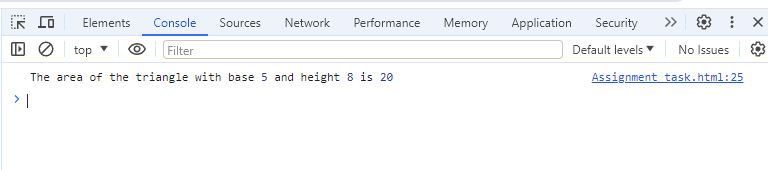
        var area = calculateTriangleArea(base, height);

        console.log("The area of the triangle with base", base, "and height", height, "is", area);

    </script>

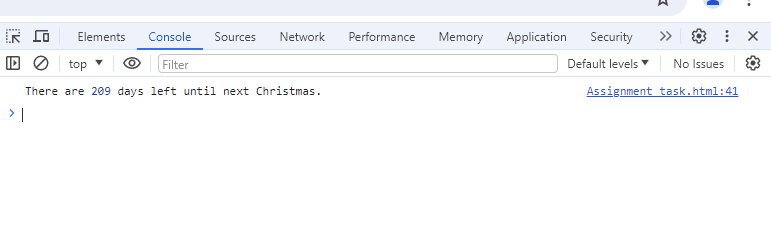
</body>

</html>



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

Ans.9



Q.10 What is Condition Statement?

Ans.10 A condition statement is a statement that controls the flow of a program based on a specified condition. The most common conditional statement is the if statement, which executes a block of code if a specified condition is true. The else statement is used to execute a block of code if the condition is false. The else if statement is used to check additional conditions after the first condition in an if statement is false.

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

Ans.11

<!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>

    <div class="calculateRectanglePerimeter"></div>

    <script>

        // Function to calculate the circumference of a rectangle

        function calculateRectangleCircumference(sideLength) {

            var circumference = 4 \* sideLength;

            return circumference;

        }

        // Example usage:

        var sideLength = 5;

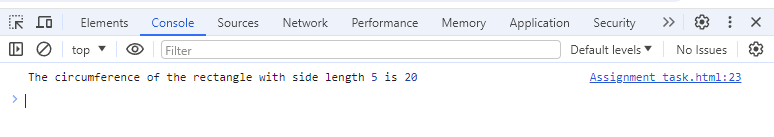
        var circumference = calculateRectangleCircumference(sideLength);

        console.log("The circumference of the rectangle with side length", sideLength, "is", circumference);

    </script>

</body>

</html>



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

Ans.12

<!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>

    <div class="yearsToDays"></div>

    <script>

        // Function to convert years into days

        function yearsToDays(years) {

            var days = years \* 365.25; // Allowing for leap years

            return days;

        }

        // Function to convert days into years

        function daysToYears(days) {

            var years = days / 365.25; // Allowing for leap years

            return years;

        }

        // Example usage:

        var years = 2;

        var days = yearsToDays(years);

        console.log(years + " years is equal to approximately " + days + " days.");

        days = 730; // 2 years worth of days

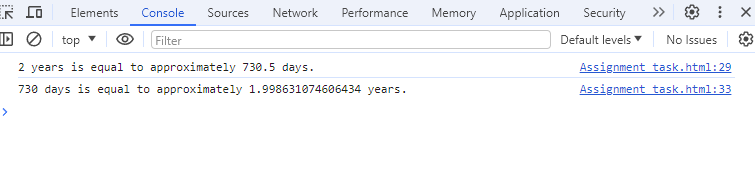
        years = daysToYears(days);

        console.log(days + " days is equal to approximately " + years + " years.");

    </script>

</body>

</html>



Q.13 Convert temperature Fahrenheit to Celsius? (Conditional logic Question)

Ans.13

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Temperature Converter</title>

</head>

<body>

    <h2>Temperature Converter</h2>

    <label for="fahrenheitInput">Enter Temperature in Fahrenheit:</label>

    <input type="number" id="fahrenheitInput" placeholder="Enter temperature...">

    <button onclick="convertToFahrenheit()">Convert to Celsius</button>

    <p id="celsiusResult"></p>

    <script>

        // Function to convert Fahrenheit to Celsius

        function convertToFahrenheit() {

            var fahrenheit = parseFloat(document.getElementById("fahrenheitInput").value);

            if (!isNaN(fahrenheit)) {

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

                document.getElementById("celsiusResult").textContent = fahrenheit + "°F is equal to " + celsius.toFixed(2) + "°C.";

            } else {

                document.getElementById("celsiusResult").textContent = "Please enter a valid temperature in Fahrenheit.";

            }

        }

    </script>

</body>

</html>



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

Ans.14

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Temperature Converter</title>

</head>

<body>

    <div class="getFileExtension"></div>

    <script>

        function getFileExtension(filename) {

            // Find the position of the last dot in the filename

            const lastDotPosition = filename.lastIndexOf('.');

            // If there is no dot or the dot is the first character (hidden files), return an empty string

            if (lastDotPosition === -1 || lastDotPosition === 0 || lastDotPosition === filename.length - 1) {

                return '';

            }

            // Return the substring from the last dot position to the end

            return filename.substring(lastDotPosition + 1);

        }

        // Test cases

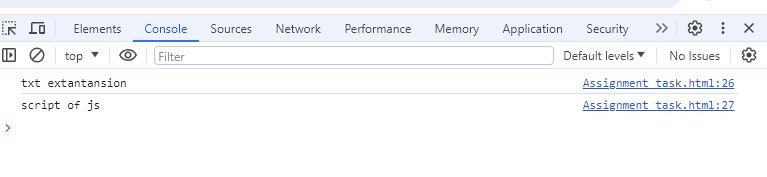
        console.log(getFileExtension("document.txt extantansion"));

        console.log(getFileExtension("archive.script of js"));

    </script>

</body>

</html>



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

Ans.15

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Temperature Converter</title>

</head>

<body>

    <div class=""></div>

    <script>

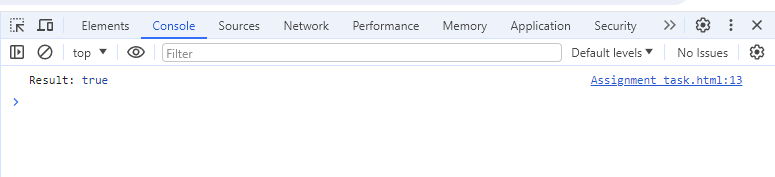
        const result = 5 > 3 && 2 < 4;

        console.log("Result:", result);

    </script>

</body>

</html>



The condition 5 > 3 evaluates to true because 5 is indeed greater than 3.

The condition 2 < 4 also evaluates to true because 2 is less than 4.

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

Ans.16

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Temperature Converter</title>

</head>

<body>

    <div class=""></div>

    <script>

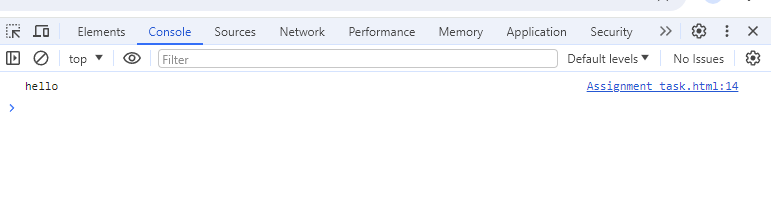
        const result = (true && 1 && "hello");

        console.log(result);

    </script>

</body>

</html>



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

Ans.17 true && false evaluates to false because one of the operands (false) is false.

false && true also evaluates to false for the same reason.

Then, we have false || false, where both operands are false. So, it returns the last operand, which is false.

So, the result of the expression true && false || false && true is false.

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Temperature Converter</title>

</head>

<body>

    <div class=""></div>

    <script>

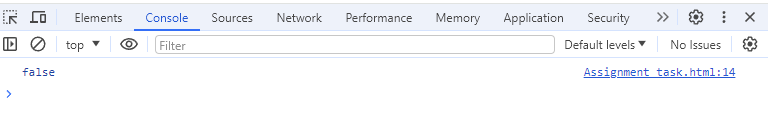
        const result = true && false || false && true;

        console.log(result);

    </script>

</body>

</html>



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

Ans.18 Loops and switch cases are both control flow statements in JavaScript. They are used to control the flow of execution of a program.

Loops:

A loop is a control structure that repeats a block of code until a certain condition is met. There are three types of loops in JavaScript: for loop, while loop, and do-while loop.

Switch cases:

A switch case is a control structure that executes different blocks of code based on the value of an expression. The syntax for a switch case is:

switch (expression) {

case value1:

// code block to be executed if expression is equal to value1

break;

case value2:

// code block to be executed if expression is equal to value2

break;

default:

// code block to be executed if expression is not equal to any of the case values

}

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

Ans.19   In JavaScript NaN is short for "Not-a-Number". The is NaN() method returns true if a value is NaN.

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

Ans.20 || means “or”, it evaluates to true if the expression on the either side is truthly, && “and “it evaluates to false if the expression on either side is false.

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

Ans.21 JavaScript void 0 means returning undefined (void) as a primitive value. You might 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.

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

Ans.22

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Temperature Converter</title>

</head>

<body>

    <div class=""></div>

    <script>

        function checkNumber(num) {

            if (num > 0) {

                console.log(num + " is positive.");

            } else if (num < 0) {

                console.log(num + " is negative.");

            } else {

                console.log(num + " is zero.");

            }

        }

        checkNumber(10);

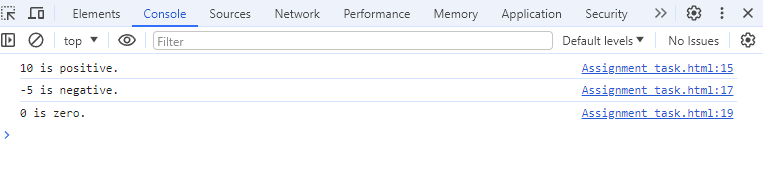
        checkNumber(-5);

        checkNumber(0);

    </script>

</body>

</html>



Q.23 Find the Character Is Vowel or Not?

Ans.23

<!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>

    <div class=""></div>

    <script>

        function isVowel(char) {

            // Convert the character to lowercase to handle both cases

            char = char.toLowerCase();

            // Check if the character is one of the vowels

            if (char === 'a' || char === 'e' || char === 'i' || char === 'o' || char === 'u') {

                return true;

            } else {

                return false;

            }

        }

        console.log(isVowel('a'));  // Output: true

        console.log(isVowel('E'));  // Output: true

        console.log(isVowel('b'));  // Output: false

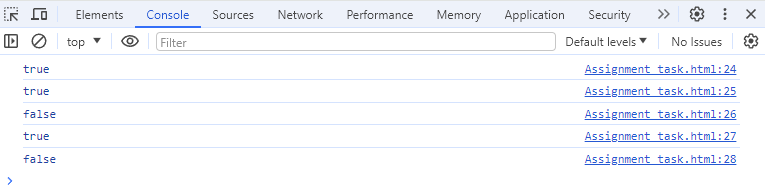
        console.log(isVowel('O'));  // Output: true

        console.log(isVowel('z'));  // Output: false

    </script>

</body>

</html>



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

Ans.24

<!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>

    <div class=""></div>

    <script>

        function checkNumber(num) {

            if (num > 0) {

                return num + " is positive.";

            } else if (num < 0) {

                return num + " is negative.";

            } else {

                return num + " is zero.";

            }

        }

        console.log(checkNumber(10));

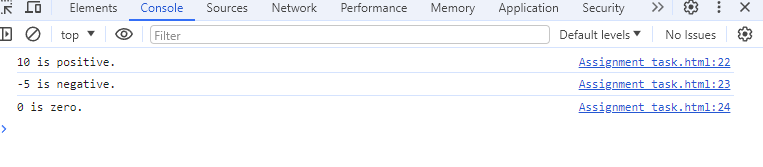
        console.log(checkNumber(-5));

        console.log(checkNumber(0));

    </script>

</body>

</html>



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

Ans.25

<!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>

    <div class=""></div>

    <script>

        function isEvenOrOdd(num) {

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

        }

        let userInput = prompt("Enter a number:");

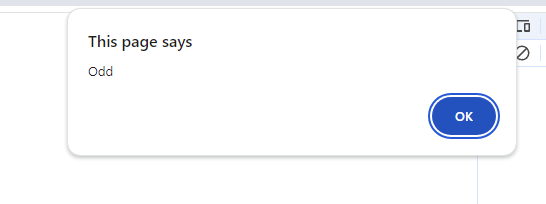
        let number = parseFloat(userInput);

        alert(isEvenOrOdd(number));

    </script>

</body>

</html>



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

Ans.26

<!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>

    <div class=""></div>

    <script>

        function findMax(a, b, c) {

            return (a >= b) ?

                ((a >= b) ? a : c) :

                ((b >= c) ? b : c);

        }

        console.log(findMax(10, 20, 30));

        console.log(findMax(50, 20, 30));

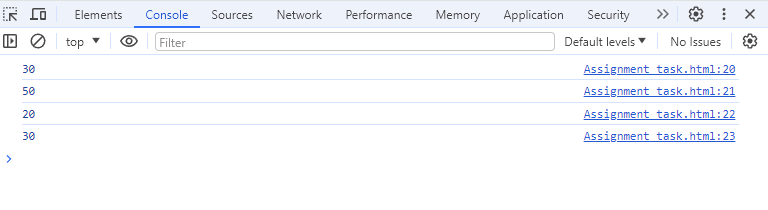
        console.log(findMax(10, 20, 20));

        console.log(findMax(30, 30, 30));

    </script>

</body>

</html>



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

Ans.27

<!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>

    <div class=""></div>

    <script>

        function findmin(a, b, c) {

            return (a <= b) ?

                ((a <= b) ? a : c) :

                ((b <= c) ? b : c);

        }

        console.log(findmin(10, 20, 30));

        console.log(findmin(50, 20, 30));

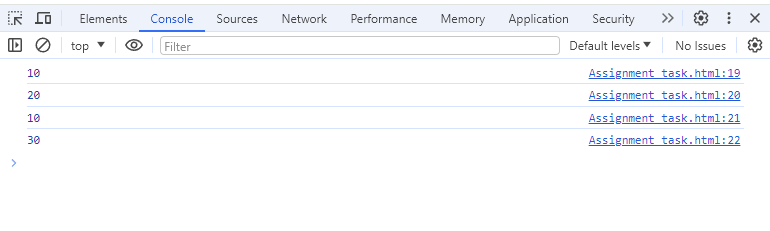
        console.log(findmin(10, 20, 20));

        console.log(findmin(30, 30, 30));

    </script>

</body>

</html>



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

Ans.28

<!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>

    <div class=""></div>

    <script>

        function findLargestNumber(num1, num2, num3) {

            let largest = num1;

            if (num2 > largest) {

                largest = num2;

            }

            if (num3 > largest) {

                largest = num3;

            }

            return largest;

        }

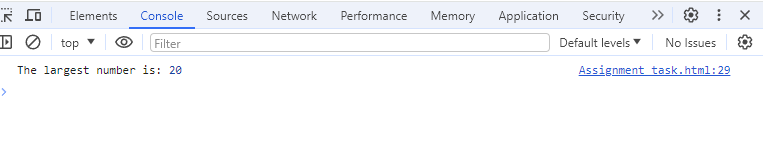
        const largestNumber = findLargestNumber(10, 5, 20);

        console.log("The largest number is:", largestNumber);

    </script>

</body>

</html>



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

Ans.29

<!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>

    <div class=""></div>

    <script>

        function getDayOfWeek(dayNumber) {

            let day;

            switch (dayNumber) {

                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";

                    break;

            }

            return day;

        }

        console.log("Day 1 is:", getDayOfWeek(1));

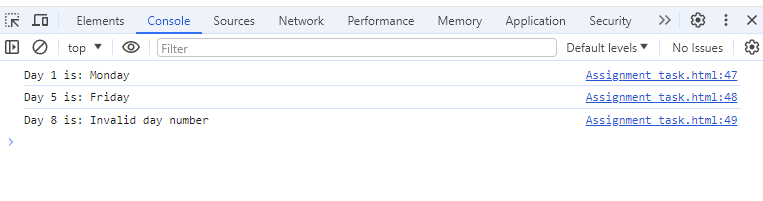
        console.log("Day 5 is:", getDayOfWeek(5));

        console.log("Day 8 is:", getDayOfWeek(8));

    </script>

</body>

</html>



(Conditional looping logic Question)

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

Ans.30

In JavaScript, there are several looping structures that allow you to execute a block of code repeatedly. The main looping structures are:

for loop: Executes a block of code a specified number of times.

while loop: Executes a block of code while a specified condition is true.

do...while loop: Similar to the while loop, but the block of code is executed at least once, even if the condition is false.

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

Ans.31

<!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>

    <div class=""></div>

    <script>

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

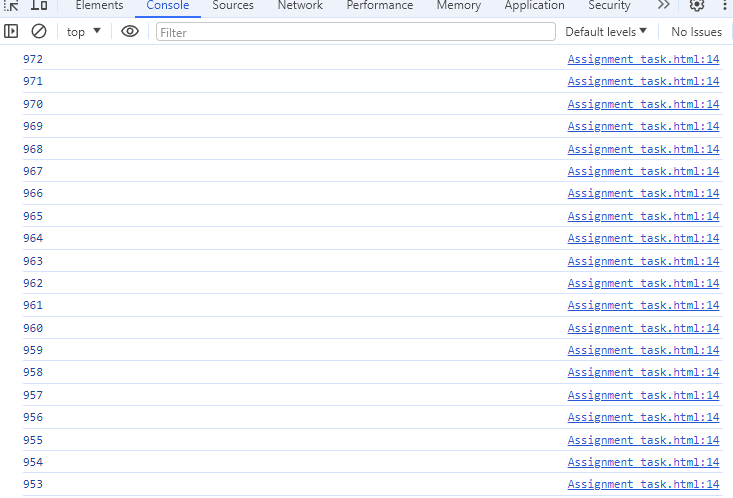
            console.log(i);

        }

    </script>

</body>

</html>



Q.32 Write to print factorial of given number?

Ans.32

<!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>

    <div class=""></div>

    <script>

        function factorial(n) {

            if (n === 0 || n === 1) {

                return 1;

            } else {

                let result = 1;

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

                    result \*= i;

                }

                return result;

            }

        }

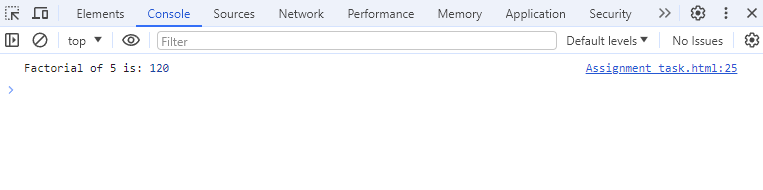
        const number = 5;

        console.log(`Factorial of ${number} is:`, factorial(number));

    </script>

</body>

</html>



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

Ans.33

<!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>

    <div class=""></div>

    <script>

        function fibonacciSeries(limit) {

            let fibSeries = [0, 1];

            let nextTerm = 0;

            while (true) {

                nextTerm = fibSeries[fibSeries.length - 1] + fibSeries[fibSeries.length - 2];

                if (nextTerm <= limit) {

                    fibSeries.push(nextTerm);

                } else {

                    break;

                }

            }

            return fibSeries;

        }

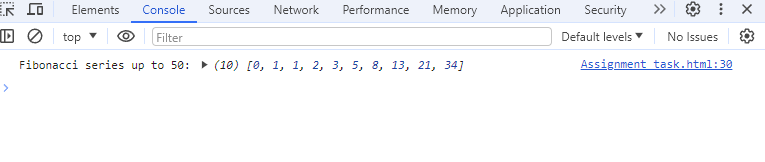
        const limit = 50;

        console.log(`Fibonacci series up to ${limit}:`, fibonacciSeries(limit));

    </script>

</body>

</html>



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

Ans.34

<!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>

    <div class=""></div>

    <script>

        function reverseNumber(number) {

            const reversedNumber = parseInt(number.toString().split('').reverse().join(''));

            return reversedNumber;

        }

        const number = 64728;

        const reversedNumber = reverseNumber(number);

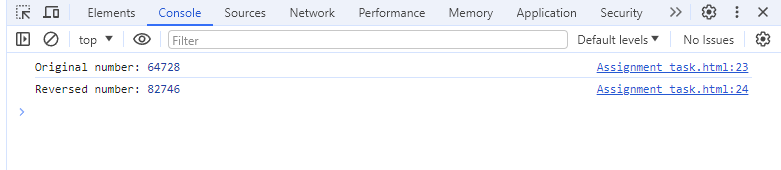
        console.log("Original number:", number);

        console.log("Reversed number:", reversedNumber);

    </script>

</body>

</html>



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

Ans.35

<!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>

    <div class=""></div>

    <script>

        function summationOfNumber(number) {

            let sum = 0;

            const numString = number.toString();

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

                sum += parseInt(numString[i]);

            }

            return sum;

        }

        const number = 1523;

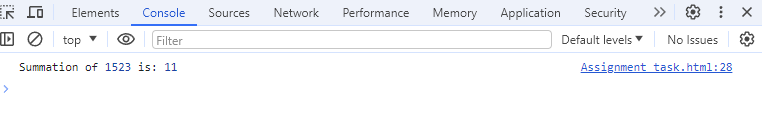
        const sum = summationOfNumber(number);

        console.log("Summation of", number, "is:", sum);

    </script>

</body>

</html>



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

Ans.36

<!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>

    <div class=""></div>

    <script>

        function summationOfFirstAndLastDigit(number) {

            const numString = number.toString();

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

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

            return firstDigit + lastDigit;

        }

        const number = 1234;

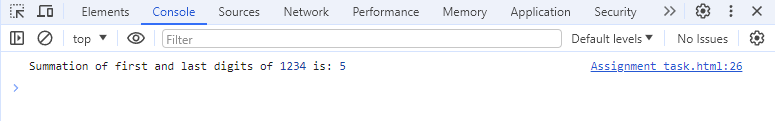
        const sum = summationOfFirstAndLastDigit(number);

        console.log("Summation of first and last digits of", number, "is:", sum);

    </script>

</body>

</html>



Q.37 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

Ans.37

<!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>

    <div class=""></div>

    <script>

        function printPattern(rows) {

            let counter = 1;

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

                let rowOutput = '';

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

                    rowOutput += Math.pow(counter, j) + ' ';

                }

                console.log(rowOutput.trim());

                counter++;

            }

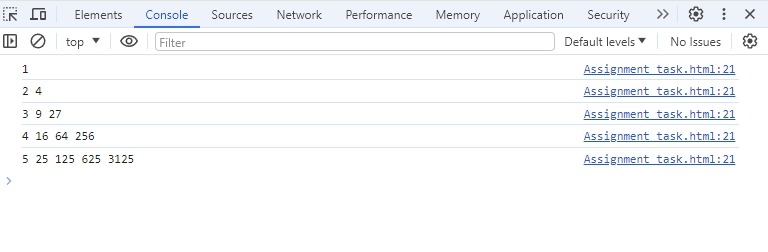
        }

        printPattern(5);

    </script>

</body>

</html>



Q.38 Use pattern in console.log in JS? 1) 1 1 0 1 0 1 1 0 1 0 1 0 1 0 1 2) A B C D E F G H I J K L M N O 3) 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 4) \* \* \* \* \* \* \* \* \* \* \* \* \* \* \*

Ans.38

<!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>

    <div class=""></div>

    <script>

        // Pattern 1

        console.log("Pattern 1:");

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

            let line = '';

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

                if (j % 2 === 0) {

                    line += '0 ';

                } else {

                    line += '1 ';

                }

            }

            console.log(line);

        }

        // Pattern 2

        console.log("\nPattern 2:");

        let charCode = 65; // ASCII code for 'A'

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

            let line = '';

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

                line += String.fromCharCode(charCode++) + ' ';

            }

            console.log(line);

        }

        // Pattern 3

        console.log("\nPattern 3:");

        let num = 1;

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

            let line = '';

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

                line += num++ + ' ';

            }

            console.log(line);

        }

        // Pattern 4

        console.log("\nPattern 4:");

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

            let line = '';

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

                line += '\* ';

            }

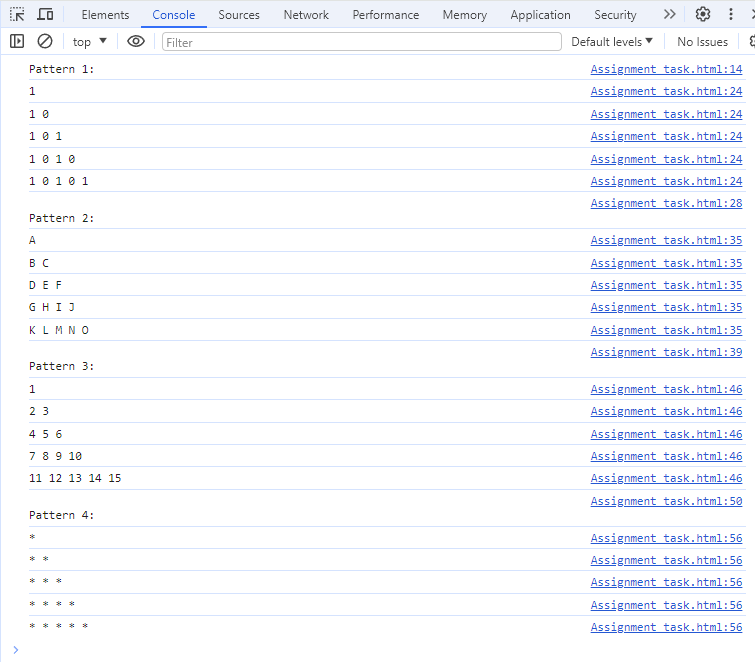
            console.log(line);

        }

    </script>

</body>

</html>



Q.39 Accept 3 numbers from user using while loop and check each numbers palindrome?

Ans.39

<!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>

    <div class=""></div>

    <script>

             const readline = ('readline');

const rl = readline.createInterface({

    input: process.stdin,

    output: process.stdout

});

function isPalindrome(number) {

    const numString = number.toString();

    const reversedString = numString.split('').reverse().join('');

    return numString === reversedString;

}

let count = 1;

function checkPalindrome() {

    rl.question(`Enter number ${count}: `, (number) => {

        if (!isNaN(number)) {

            if (isPalindrome(number)) {

                console.log(`${number} is a palindrome.`);

            } else {

                console.log(`${number} is not a palindrome.`);

            }

            count++;

            if (count <= 3) {

                checkPalindrome();

            } else {

                rl.close();

            }

        } else {

            console.log("Please enter a valid number.");

            checkPalindrome();

        }

    });

}

checkPalindrome();

    </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 ?

Ans.40

<!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>

    <div class=""></div>

    <script>

        function getCurrentDay() {

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

            const now = new Date();

            const day = days[now.getDay()];

            return day;

        }

        function getCurrentTime() {

            const now = new Date();

            let hours = now.getHours();

            let minutes = now.getMinutes();

            let ampm = hours >= 12 ? 'PM' : 'AM';

            hours = hours % 12;

            hours = hours ? hours : 12; // Handle midnight (0 hours)

            minutes = minutes < 10 ? '0' + minutes : minutes;

            return `${hours} ${ampm}: ${now.getSeconds()} : ${minutes} ${now.getTimezoneOffset() / 60} ?`;

        }

        const day = getCurrentDay();

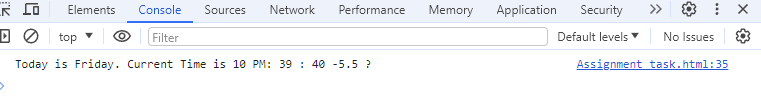
        const time = getCurrentTime();

        console.log(`Today is ${day}. Current Time is ${time}`);

    </script>

</body>

</html>



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

Ans.41

<!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>

    <div class=""></div>

    <script>

        function getCurrentDate() {

            const now = new Date();

            const year = now.getFullYear();

            const month = String(now.getMonth() + 1).padStart(2, '0'); // Adding padding if month is a single digit

            const day = String(now.getDate()).padStart(2, '0'); // Adding padding if day is a single digit

            return `${year}-${month}-${day}`;

        }

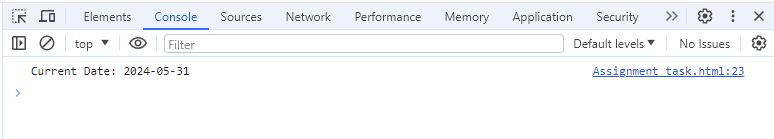
        const currentDate = getCurrentDate();

        console.log("Current Date:", currentDate);

    </script>

</body>

</html>



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

Ans.42

<!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>

    <div class=""></div>

    <script>

        function compareObjects(obj1, obj2) {

            const keys1 = Object.keys(obj1);

            const keys2 = Object.keys(obj2);

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

                return false;

            }

            for (let key of keys1) {

                if (obj1[key] !== obj2[key]) {

                    return false;

                }

            }

            for (let key of keys2) {

                if (obj1[key] !== obj2[key]) {

                    return false;

                }

            }

            return true;

        }

        const obj1 = { a: 1, b: 2 };

        const obj2 = { b: 2, a: 1 };

        const obj3 = { a: 1, b: 2, c: 3 };

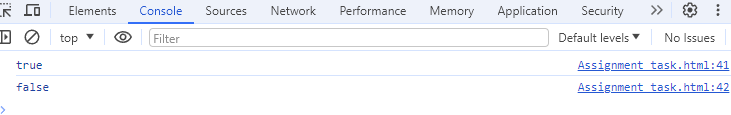
        console.log(compareObjects(obj1, obj2));

        console.log(compareObjects(obj1, obj3));

    </script>

</body>

</html>



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

Ans.43

<!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>

    <div class=""></div>

    <script>

        function convertArrayToCSV(array) {

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

            const rows = [];

            rows.push(headers.join(','));

            array.forEach(obj => {

                const values = headers.map(header => {

                    const value = obj[header];

                    return typeof value === 'undefined' ? '' : value;

                });

                rows.push(values.join(','));

            });

            return rows.join('\n');

        }

        const data = [

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

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

            { name: 'Bob', age: 35, city: 'Chicago' }

        ];

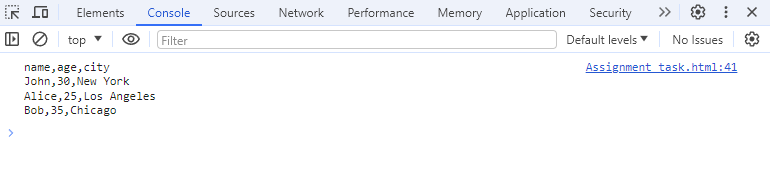
        const csvString = convertArrayToCSV(data);

        console.log(csvString);

    </script>

</body>

</html>



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

Ans.44

<!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>

    <div class=""></div>

    <script>

        function capitalizeFirstLetter(str) {

            if (!str) {

                return str;

            }

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

        }

        const inputString = "hello world";

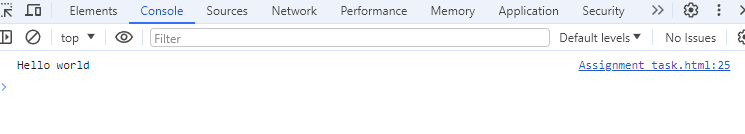
        const capitalizedString = capitalizeFirstLetter(inputString);

        console.log(capitalizedString);

    </script>

</body>

</html>



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

Ans.45.

<!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>

    <div class=""></div>

    <script>

        function isArray(variable) {

            return Array.isArray(variable);

        }

        const arr = [1, 2, 3];

        const notArr = "Hello";

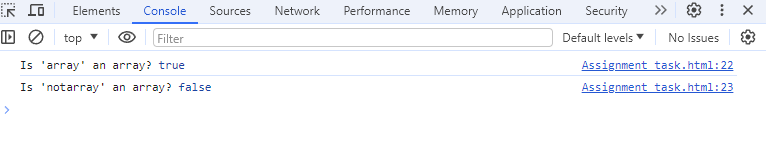
        console.log("Is 'array' an array?",isArray(arr));

        console.log("Is 'notarray' an array?",isArray(notArr));

    </script>

</body>

</html>



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

Ans.46

<!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>

    <div class=""></div>

    <script>

        function cloneArray(arr) {

            return [...arr];

        }

        function cloneArray(arr) {

            return Array.from(arr);

        }

        function cloneArray(arr) {

            return arr.slice();

        }

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

        const clonedArray = cloneArray(originalArray);

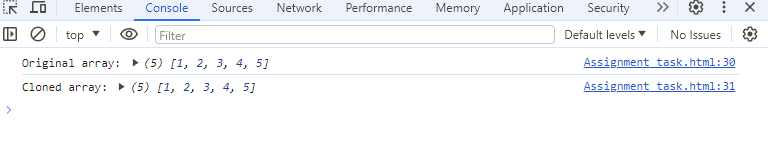
        console.log("Original array:", originalArray);

        console.log("Cloned array:", clonedArray);

    </script>

</body>

</html>



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

Ans.47

There are a few drawbacks to declaring methods directly in JavaScript objects:

Performance: Each object will contain its own copy of the method, which can take up a lot of memory, especially if the object is created many times.

Reusability: Methods declared directly in objects cannot be reused by other objects.

Maintainability: Code can become more difficult to maintain if methods are scattered throughout different objects.

To avoid these drawbacks, it is generally recommended to declare methods on the prototype of the object. This way, each object will share a single copy of the method, which can save memory and improve performance. Methods declared on the prototype can also be reused by other objects and are easier to maintain.

Here is an example of how to declare a method on the prototype of an object:

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

Ans.48

<!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>

    <div class=""></div>

    <script>

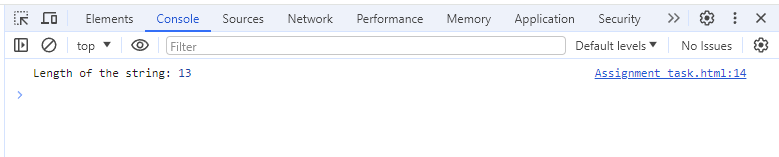
        const myString = "Hello, world!";

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

    </script>

</body>

</html>



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

Ans.49

<!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>

    <div class=""></div>

    <script>

        const myString = "hello, world!";

        const capitalizedString = myString.toUpperCase();

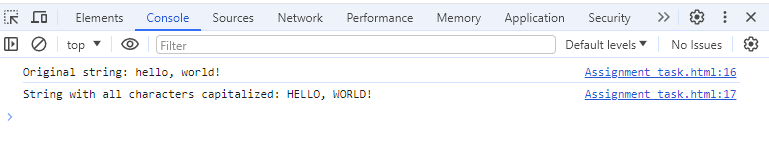
        console.log("Original string:", myString);

        console.log("String with all characters capitalized:", capitalizedString);

    </script>

</body>

</html>



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

Ans.50 There are a few drawbacks to declaring methods directly in JavaScript objects:

Performance: Each object will contain its own copy of the method, which can take up a lot of memory, especially if the object is created many times.

Reusability: Methods declared directly in objects cannot be reused by other objects.

Maintainability: Code can become more difficult to maintain if methods are scattered throughout different objects.

To avoid these drawbacks, it is generally recommended to declare methods on the prototype of the object. This way, each object will share a single copy of the method, which can save memory and improve performance. Methods declared on the prototype can also be reused by other objects and are easier to maintain.

Here is an example of how to declare a method on the prototype of an object:

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?

Ans.51

<!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>

    <div class=""></div>

    <script>

        function getCurrentDate(format) {

            const currentDate = new Date();

            const year = currentDate.getFullYear();

            let month = String(currentDate.getMonth() + 1).padStart(2, '0');

            let day = String(currentDate.getDate()).padStart(2, '0');

            if (format === "mm-dd-yyyy" || format === "mm/dd/yyyy") {

                return `${month}-${day}-${year}`;

            } else if (format === "dd-mm-yyyy" || format === "dd/mm/yyyy") {

                return `${day}-${month}-${year}`;

            } else {

                return "Invalid format";

            }

        }

        console.log("Current date (mm-dd-yyyy):", getCurrentDate("mm-dd-yyyy"));

        console.log("Current date (mm/dd/yyyy):", getCurrentDate("mm/dd/yyyy"));

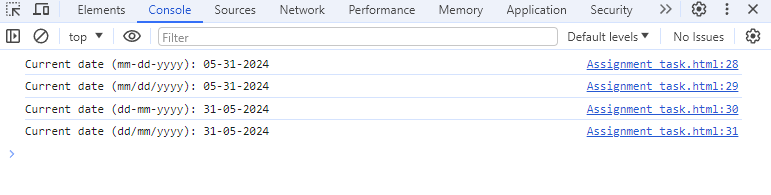
        console.log("Current date (dd-mm-yyyy):", getCurrentDate("dd-mm-yyyy"));

        console.log("Current date (dd/mm/yyyy):", getCurrentDate("dd/mm/yyyy"));

    </script>

</body>

</html>



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

Ans.52

<!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>

    <div class=""></div>

    <script>

        const text = "30 Days Of JavaScript";

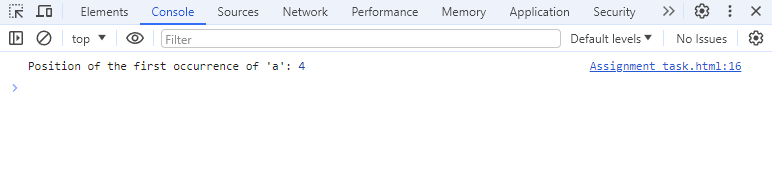
        const position = text.indexOf('a');

        console.log("Position of the first occurrence of 'a':", position);

    </script>

</body>

</html>



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

Ans.53

<!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>

    <div class=""></div>

    <script>

        const text = "30 Days Of JavaScript";

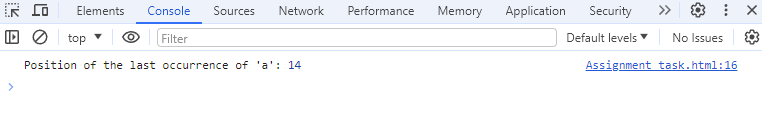
        const position = text.lastIndexOf('a');

        console.log("Position of the last occurrence of 'a':", position);

    </script>

</body>

</html>



Q.54 Form Validation in JS?

Ans.54

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>User Form</title>

</head>

<body>

    <form action="massage.html" onsubmit="return main()">

        <label for="one">FIRST NAME :</label>

        <input type="text" id="one" required>

        <br>

        <label for="two">LAST NAME :</label>

        <input type="text" id="two" required>

        <br>

        <input type="submit" value="Submit">

    </form>

    <h1 id="demo"></h1>

    <script>

        function main() {

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

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

            if (a && b) {

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

                true

            } else {

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

                return false;

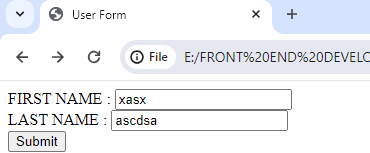
            }

        }

    </script>

</body>

</html>



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

Ans.55

<!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>

    <form id="myForm" onsubmit="return validateForm()">

        <label for="email">Email:</label>

        <input type="email" id="email" name="email"><br><br>

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

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

        <input type="submit" value="Submit">

    </form>

    <div id="errorMessages"></div>

    <script>

        function validateForm() {

            var emailInput = document.getElementById("email").value;

            var passwordInput = document.getElementById("password").value;

            var errorMessages = "";

            if (!emailInput || emailInput.trim() === "") {

                errorMessages += "Email is required.\n";

            } else {

                var emailPattern = /^\w+@[a-zA-Z\_]+?\.[a-zA-Z]{2,3}$/;

                if (!emailPattern.test(emailInput)) {

                    errorMessages += "Please enter a valid email address.\n";

                }

            }

            if (!passwordInput || passwordInput.trim() === "") {

                errorMessages += "Password is required.\n";

            } else if (passwordInput.length < 8) {

                errorMessages += "Password must be at least 8 characters long.\n";

            }

            if (errorMessages !== "") {

                document.getElementById("errorMessages").innerText = errorMessages;

                return false;

            }

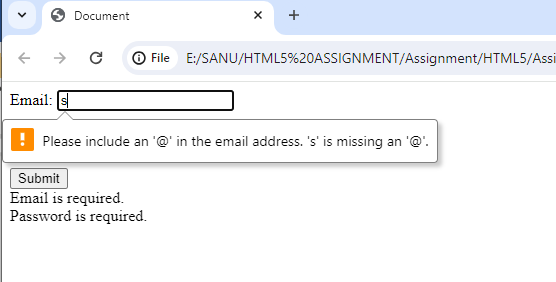
            return true;

        }

    </script>

</body>

</html>



Q.56 Dynamic Form Validation in JS?

Ans.56

<!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>

    <form id="myForm">

        <label for="email">Email:</label>

        <input type="email" id="email" name="email"><br>

        <span id="emailError" class="error"></span><br>

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

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

        <span id="passwordError" class="error"></span><br>

        <input type="submit" value="Submit">

    </form>

    <script>

        document.getElementById("myForm").addEventListener("submit", function (event) {

            var emailInput = document.getElementById("email").value;

            var passwordInput = document.getElementById("password").value;

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

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

            var isValid = true;

            if (!emailInput || emailInput.trim() === "") {

                emailError.textContent = "Email is required.";

                isValid = false;

            } else {

                var emailPattern = /^\w+@[a-zA-Z\_]+?\.[a-zA-Z]{2,3}$/;

                if (!emailPattern.test(emailInput)) {

                    emailError.textContent = "Please enter a valid email address.";

                    isValid = false;

                } else {

                    emailError.textContent = "";

                }

            }

            if (!passwordInput || passwordInput.trim() === "") {

                passwordError.textContent = "Password is required.";

                isValid = false;

            } else if (passwordInput.length < 8) {

                passwordError.textContent = "Password must be at least 8 characters long.";

                isValid = false;

            } else {

                passwordError.textContent = "";

            }

            if (!isValid) {

                event.preventDefault();

            }

        });

        document.getElementById("email").addEventListener("input", function () {

            var emailInput = document.getElementById("email").value;

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

            if (!emailInput || emailInput.trim() === "") {

                emailError.textContent = "Email is required.";

            } else {

                var emailPattern = /^\w+@[a-zA-Z\_]+?\.[a-zA-Z]{2,3}$/;

                if (!emailPattern.test(emailInput)) {

                    emailError.textContent = "Please enter a valid email address.";

                } else {

                    emailError.textContent = "";

                }

            }

        });

        document.getElementById("password").addEventListener("input", function () {

            var passwordInput = document.getElementById("password").value;

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

            if (!passwordInput || passwordInput.trim() === "") {

                passwordError.textContent = "Password is required.";

            } else if (passwordInput.length < 8) {

                passwordError.textContent = "Password must be at least 8 characters long.";

            } else {

                passwordError.textContent = "";

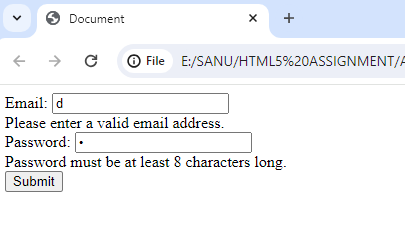
            }

        });

    </script>

</body>

</html>



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

Ans.57 There are many different types of JavaScript events. Some of the most common ones include:

* Onload: When your page loads, it performs accordingly.
* Onclick: When a user clicks on a button or inputs it occurs.
* Onmouseover: When a user mouses over on the button.
* Onfocus: Certain scenarios when a user keeps the cursor in a form field.
* Onblur: If a particular form field leaves within it.

**For example, the following code adds an event listener to a button that will call a function called myFunction() when the button is clicked:**

**button.addEventListener("click", myFunction);**

**Q.59 What is Bom vs Dom in JS?**

Ans.59 The Document Object Model (DOM) and the Browser Object Model (BOM) are two important concepts in JavaScript development. The DOM represents the structure and content of a web document, while the BOM provides access to browser-specific functionality and features.

**DOM**

The Document Object Model (DOM) is a programming interface for HTML and XML documents. It represents the page as a tree of objects, where each object represents a part of the page, such as a paragraph, a heading, or an image. The DOM allows you to access and manipulate the content and structure of a web page.

**BOM**

The Browser Object Model (BOM) is a collection of objects that represent the browser and its environment. This includes objects such as the window, the document, the navigator, and the location. The BOM allows you to interact with the browser, such as opening and closing windows, navigating to different pages, and getting information about the user's browser.

**Q.60 Array vs object deference in JS?**

Ans.60

**Arrays**

* Arrays are used to store a collection of elements of the same data type or structure.
* Arrays are ordered collections, meaning that the elements in an array have a specific order.
* Arrays are indexed, meaning that you can access the elements in an array using their index number.
* Arrays are mutable, meaning that you can change the elements in an array.

**Objects**

* Objects are used to group multiple sets of data that belong together using labels.
* Objects are unordered collections, meaning that the elements in an object do not have a specific order.
* Objects are not indexed, meaning that you cannot access the elements in an object using their index number.
* Objects are mutable, meaning that you can change the elements in an object.

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

**Ans.61**

<!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>

    <div class="main"></div>

    <script>

        var str = "Hello World";

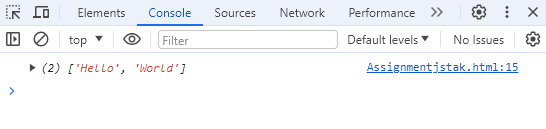
        var arr = str.split(" ");

        console.log(arr);

    </script>

</body>

</html>



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

Ans.62

<!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>

    <div class="main"></div>

    <script>

        var str = "JavaScript is a scripting language";

        var word = "Script";

        if (str.includes(word)) {

            console.log("The string contains the word 'Script'.");

        } else {

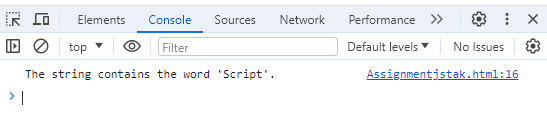
            console.log("The string does not contain the word 'Script'.");

        }

    </script>

</body>

</html>



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

Ans.63

<!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>

    <div class="main"></div>

    <script>

        var str = "Hello World";

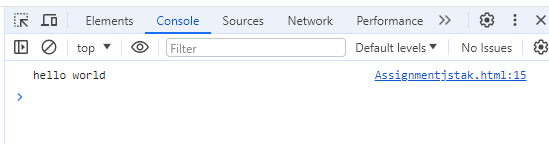
        var lowercaseStr = str.toLowerCase();

        console.log(lowercaseStr);

    </script>

</body>

</html>



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

Ans.64

<!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>

    <div class="main"></div>

    <script>

        var str = '30 Days of JavaScript';

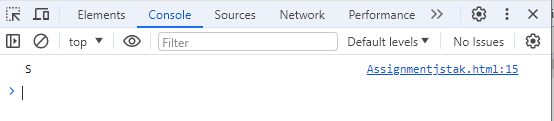
        var charAtIndex15 = str.charAt(15);

        console.log(charAtIndex15);

    </script>

</body>

</html>



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

**Find the length of a string without using libraryFunction?**

Ans.65 <!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>

    <div class="main"></div>

    <script>

        var str1 = "Hello";

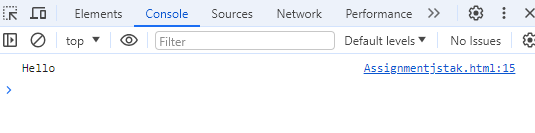
        var str2 = str1;

        console.log(str2);

    </script>

</body>

</html>



**Find the length of a string without using libraryFunction?**

<!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>

    <div class="main"></div>

    <script>

        function findStringLength(str) {

            var length = 0;

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

                length++;

            }

            return length;

        }

        var str = "Hello";

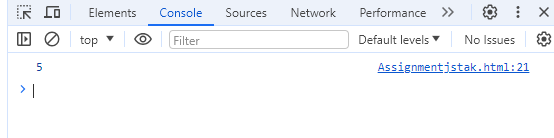
        var length = findStringLength(str);

        console.log(length);

    </script>

</body>

</html>



**Q.66 What is JavaScript?**

**Ans.66** JavaScript (JS) is a high-level, interpreted programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS. JavaScript enables interactive web pages and is used for a wide range of tasks on both the client-side and server-side.

**Q.67 What is the use of isNaN function?**

**Ans.67** The isNaN() function is used to determine whether a value is NaN, which stands for "Not a Number".

**Q.68 What is negative Infinity?**

**Ans.68** 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.

**Q.69 Which company developed JavaScript?**

Ans.69 JavaScript was created at Netscape Communications by Brendan Eich in 1995. Netscape and Eich designed JavaScript as a scripting language for use with the company's flagship web browser, Netscape Navigator.

**Q.70.What are undeclared and undefined variables?**

**Ans.70**

* Undeclared variables

are variables that have not been declared using the var, let, or const keyword. Trying to access an undeclared variable will throw a Reference Error.

* Undefined variables

are variables that have been declared but have not been assigned a value. The value of an undefined variable is undefined.

**Q.71 Write the code for adding new elements dynamically?**

**Ans.71.**

<!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>

    <div id="container">

    </div>

    <button id="addButton">Add Element</button>

    <script>

        function addNewElement() {

            var newElement = document.createElement("div");

            newElement.textContent = "New Element";

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

            container.appendChild(newElement);

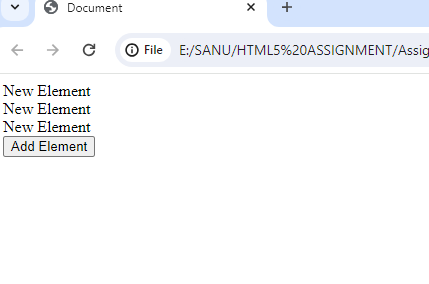
        }

        document.getElementById("addButton").addEventListener("click", addNewElement);

    </script>

</body>

</html>



**Q.72** What is the difference between View State and Session State?

Ans.72 ViewState and SessionState are two different ways to store data in a web application. ViewState is stored on the client side, while SessionState is stored on the server side.

**Q.73 What is === operator?**

**Ans.73** The === operator in JavaScript is a strict equality operator. It checks if two values are equal, and also checks if they are of the same type.

**Q.74 How can the style/class of an element be changed?**

**Ans.74 *document.getElementById("myElement").style.color = "violet";***

**Q.75 How to read and write a file using JavaScript?**

**Ans.75** readFile() and rs. writeFile() methods are used to read and write of a file using javascript. The file is read using the fs. readFile() function, which is an inbuilt method.

**Q.76 What are all the looping structures in JavaScript?**

**Ans.76 The statements for loops provided in JavaScript are:**

* for statement.
* do...while statement.
* while statement.
* break statement.
* continue statement.

**Q.77 How can you convert the string of any base to an integer in JavaScript?**

**Ans.77**

<!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 binaryString = "1010";

        var binaryInteger = parseInt(binaryString, 2);

        console.log(binaryInteger);

        var octalString = "755";

        var octalInteger = parseInt(octalString, 8);

        console.log(octalInteger);

        var hexString = "1A";

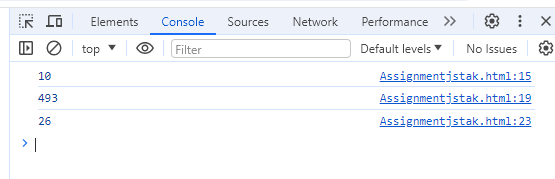
        var hexInteger = parseInt(hexString, 16);

        console.log(hexInteger);

    </script>

</body>

</html>



**Q.78 What is the function of the delete operator?**

**Ans.78** The delete operator in JavaScript is used to delete properties from objects. It is the only way to completely remove a property from an object.

**Q.79 What are all the types of Pop up boxes available in JavaScript?**

**Ans.79** There are three types of popup boxes in JavaScript:

* Alert box:

This is the simplest type of popup box, and it is used to display a message to the user. The user can only click OK to close the box.

* Confirm box:

This type of popup box is used to ask the user a question and get their confirmation. The user can click OK or Cancel to close the box.

* Prompt box:

This type of popup box is used to ask the user for input. The user can enter text into the box and click OK to submit it.

**Q.80 What is the use of Void (0)?**

**Ans.80** The void operator evaluates an expression and returns undefined . By running void(0) in the URL JavaScript code**, nothing is evaluated or returned**.

**Q.81 How can a page be forced to load another page in JavaScript?**

**Ans.81 Navigate to a new page by setting window.location.href to the URL of the new page**

**window.location.href = "https://example.com/newpage";**

**Q.82 What are the disadvantages of using innerHTML in JavaScript?**

**Ans.82** It is very slow because as inner HTML already parses the content even we have to parse the content again so that's why it takes time.

**Q.83 Create password field with show hide functionalities.**

**Ans.83**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Document</title>

</head>

<body>

    <h1>Show/Hide Password</h1>

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

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

    <button id="togglePassword">Show Password</button>

    <script>

        var passwordField = document.getElementById("password");

        var toggleButton = document.getElementById("togglePassword");

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

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

                passwordField.type = "text";

                toggleButton.textContent = "Hide Password";

            } else {

                passwordField.type = "password";

                toggleButton.textContent = "Show Password";

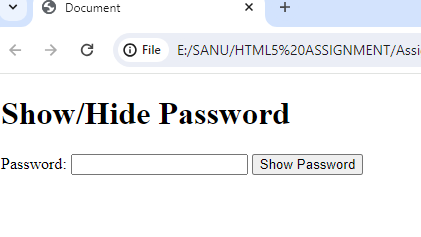
            }

        });

    </script>

</body>

</html>



**Q.84 Create basic math operation in JS.**

**Ans.84**

<!DOCTYPE html>

<html lang="en">

<head>

    <meta charset="UTF-8">

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

    <title>Basic Math Operations</title>

</head>

<body>

    <h2>Basic Math Operations</h2>

    <label for="num1">Number 1:</label>

    <input type="number" id="num1">

        <br>

    <label for="num2">Number 2:</label>

    <input type="number" id="num2">

    <button onclick="add()">Add</button>

    <button onclick="subtract()">Subtract</button>

    <button onclick="multiply()">Multiply</button>

    <button onclick="divide()">Divide</button>

    <p id="result"></p>

    <script>

        function add() {

            var num1 = parseFloat(document.getElementById("num1").value);

            var num2 = parseFloat(document.getElementById("num2").value);

            var result = num1 + num2;

            document.getElementById("result").innerText = "Result: " + result;

        }

        function subtract() {

            var num1 = parseFloat(document.getElementById("num1").value);

            var num2 = parseFloat(document.getElementById("num2").value);

            var result = num1 - num2;

            document.getElementById("result").innerText = "Result: " + result;

        }

        function multiply() {

            var num1 = parseFloat(document.getElementById("num1").value);

            var num2 = parseFloat(document.getElementById("num2").value);

            var result = num1 \* num2;

            document.getElementById("result").innerText = "Result: " + result;

        }

        function divide() {

            var num1 = parseFloat(document.getElementById("num1").value);

            var num2 = parseFloat(document.getElementById("num2").value);

            if (num2 === 0) {

                document.getElementById("result").innerText = "Error: Division by zero";

            } else {

                var result = num1 / num2;

                document.getElementById("result").innerText = "Result: " + result;

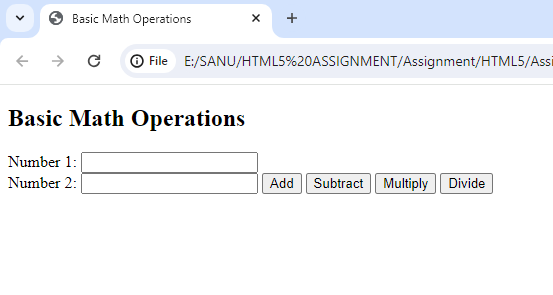
            }

        }

    </script>

</body>

</html>



**Q.85 Create result**

**Ans.85**

<!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>

    <form action="">

        ENTER STUDENT NAME : <input type="text" name="" id="sname" oninput="one()" required pattern="[a-zA-Z]{1,}">

    </form>

    <br>

    <br>

    <br>

    <table border="2" cellspacing="0" cellpadding="0">

        <tr>

            <td colspan="4">STUDENT NAME : <span id="s-name"></span></td>

        </tr>

        <tr>

            <td>SUBJECT</td>

            <td>SESSIONAL (30)</td>

            <td>FINAL (70)</td>

            <td>TOTAL</td>

        </tr>

        <tr>

            <td>

                MATHS

            </td>

            <td>

                <input type="number" oninput="one()" id="sub-1">

            </td>

            <td>

                <input type="number" oninput="one()" id="sub-2">

            </td>

            <th id="sub-total"></th>

        </tr>

        <tr>

            <td>

                SCIENCE

            </td>

            <td>

                <input type="number" oninput="one()" id="sub-3">

            </td>

            <td>

                <input type="number" oninput="one()" id="sub-4">

            </td>

            <th id="sub-total-2"></th>

        </tr>

        <tr>

            <td colspan="3">TOTAL MARKS OBTAIN : </td>

            <th id="gtot" oninput="one()"></th>

        </tr>

        <tr>

            <td colspan="3">TOTAL PERCENTAGE : </td>

            <th id="per" oninput="one()"></th>

        </tr>

        <tr>

            <td colspan="3"> PASS OR FAIL</td>

            <th id="result" oninput="one()"></th>

        </tr>

        <tr>

            <td colspan="3">GRADE</td>

            <th id="grade"></th>

        </tr>

    </table>

    <button onclick="print()">PRINT</button>

    <script>

        function one() {

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

            document.getElementById("s-name").innerHTML = sname;

            let a = document.getElementById("sub-1");

            let b = document.getElementById("sub-2");

            let c = document.getElementById("sub-3");

            let d = document.getElementById("sub-4");

            let tot1 = Number(a.value) + Number(b.value);

            let tot2 = Number(c.value) + Number(d.value);

            document.getElementById("sub-total").innerHTML = tot1;

            document.getElementById("sub-total-2").innerHTML = tot2;

            let gtotal = document.getElementById("gtot").innerHTML = tot1 + tot2;

            let totper = Number(gtotal / 200) \* 100;

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

            if (totper > 100) {

                document.getElementById("result").innerHTML = "invalid";

                document.getElementById("result").style.color = "red"

            }

            else if (totper > 40) {

                document.getElementById("result").innerHTML = "PASS";

                document.getElementById("result").style.color = "green"

            }

            else {

                document.getElementById("result").innerHTML = "FAIL";

                document.getElementById("result").style.color = "red"

            }

            let grade = calculateGrade(totper);

            document.getElementById("grade").innerHTML = grade;

        }

        function calculateGrade(percentage) {

            if (percentage >= 90) {

                return "A+";

            } else if (percentage >= 80) {

                return "A";

            } else if (percentage >= 70) {

                return "B";

            } else if (percentage >= 60) {

                return "C";

            } else if (percentage >= 50) {

                return "D";

            } else {

                return "F";

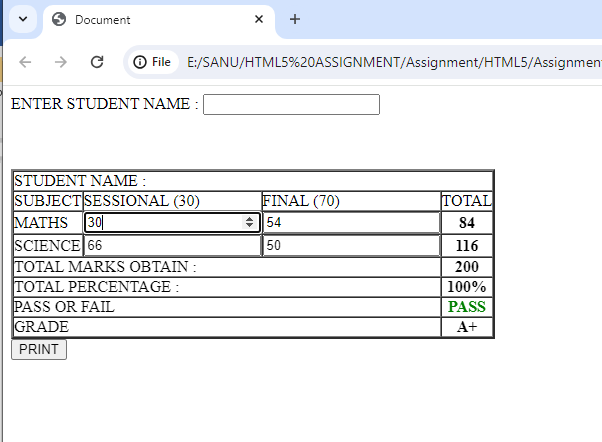
            }

        }

    </script>

</body>

</html>



**Que.86 Create a slider using JavaScript.**

**Ans.86**

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

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

<title>JavaScript Slider</title>

<style>

    .slider {

        width: 400px;

        height: 200px;

        overflow: hidden;

        position: relative;

    }

    .slides {

        display: flex;

        text-align: center;

        transition: transform 0.5s ease-in-out;

    }

    .slide {

        flex: 0 0 auto;

        width: 400px;

        height: 350px;

    }

    button {

        position: absolute;

        top: 50%;

        transform: translateY(-50%);

        padding: 10px;

        cursor: pointer;

        border: none;

        background-color: rgba(0, 0, 0, 0.5);

        color: white;

    }

    #prev {

        left: 0;

    }

    #next {

        right: 0;

    }

</style>

</head>

<body>

<h1>Basic Slider</h1>

<div class="slider">

    <div class="slides">

        <div class="slide" style="background-color: rgb(69, 202, 36);">Slide No.1</div>

        <div class="slide" style="background-color: rgb(160, 33, 111);">Slide No.2</div>

        <div class="slide" style="background-color: rgb(24, 173, 173);">Slide No.3</div>

    </div>

    <button id="prev" onclick="moveSlide(-1)">Prev</button>

    <button id="next" onclick="moveSlide(1)">Next</button>

</div>

<script>

    let slideIndex = 0;

    function moveSlide(n) {

        const slides = document.querySelectorAll('.slide');

        const slidesCount = slides.length;

        slideIndex = (slideIndex + n + slidesCount) % slidesCount;

        const slideWidth = slides[0].offsetWidth;

        document.querySelector('.slides').style.transform = `translateX(-${slideIndex \* slideWidth}px)`;

    }

</script>

</body>

</html>

