**WD - JavaScript Essentials & Advanced**

Q.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 Mothe r Teresa:

Ans.

    <html lang="en">

    <head>

        <meta charset="UTF-8" />

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

        <title>Console.log Example</title>

        </script>

    </head>

    <body>

        <script>

            console.log("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.");

            console.log("The quote by Mother Teresa: 'If you judge people, you have no time to

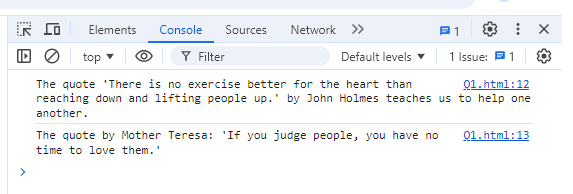
love them.'");

        </script>

    </body>

    </html>

Output :



Q.2 Check if typeof '10' is exactly equal to 10. If not make it exactly equal?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Equal or not Example</title>

    </script>

</head>

<body>

    <script>

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

            // Convert the string '10' to a number

            const numberValue = Number('10');

            console.log(numberValue === 10); // This will now print true

        } else {

            console.log('Already equal');

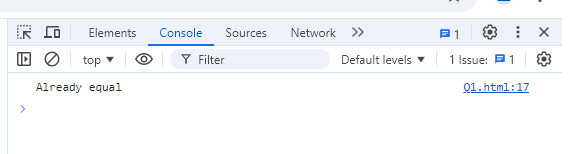
        }

</script>

</body>

</html>

**Output :**

****

Q.3 Write a JavaScript Program to find the area of a triangle?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Area of Triangle Example</title>

    </script>

</head>

<body>

    <script>

        // Function to calculate the area of a triangle

        function calculateTriangleArea(base, height) {

            // Formula to calculate area

            const area = 0.5 \* base \* height;

            return area;

        }

        // Example usage:

        const base = 10;    // Set the base of the triangle

        const height = 5;   // Set the height of the triangle

        // Calculate and print the area of the triangle

        const area = calculateTriangleArea(base, height);

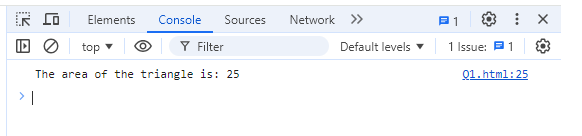
        console.log("The area of the triangle is: " + area);

    </script>

</body>

</html>

Output :



Q.4 Write a JavaScript program to calculate days left until next Christmas?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Days left Example</title>

    </script>

</head>

<body>

    <script>

        // Function to calculate days left until next Christmas

        function daysNextChristmas() {

            const today = new Date(); // Get today's date

            const currentYear = today.getFullYear(); // Get the current year

            // Set Christmas date for this year

            let nextChristmas = new Date(currentYear, 11, 25);

            // If Christmas has already passed this year, calculate for next year

            if (today > nextChristmas) {

                nextChristmas = new Date(currentYear + 1, 11, 25); // Christmas for next year

            }

            // Calculate the difference in time (milliseconds)

            const timeDifference = nextChristmas - today;

            // Convert time difference from milliseconds to days

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

            return daysLeft;

        }

        // Example usage:

        const daysLeft = daysNextChristmas();

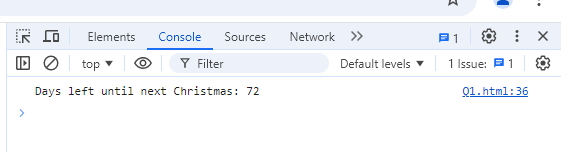
        console.log("Days left until next Christmas: " + daysLeft);

    </script>

</body>

</html>

Output :



Q.5 What is Condition Statement?

Ans.

A **condition statement** is used to declare the result on the base of criteria. If criteria match the condition, it gives positive value or it gives negative value.

In any programming languages, the most common condition statements are:

1. **if statement**
2. **else statement**
3. **else if statement**
4. **switch statement**

**1. if Statement**

The if statement is used to execute a block of code if a condition is true.

let salary = 50000;

if ( salary >= 50000) {

console.log("10% Bonus");

}

**2. else Statement**

The else statement is used in conjunction with if. It specifies a block of code to be executed if the condition is false.

let salary = 50000;

if ( salary >= 50000 ) {

console.log("10% Bonus");

} else {

console.log("20% Bonus");

}

**3. else if Statement**

The else if statement allows you to check multiple conditions in sequence.

let salary = 50000;

if ( salary >= 50000 ) {

console.log("10% Bonus");

} else if ( salary >=25000 {

console.log("20% Bonus");

} else {

console.log("30% Bonus");

}

**4. switch Statement**

The switch statement is another way to handle multiple conditions based on the value of a single expression.

let month = "August";

switch (month) {

case "January":

console.log("It's the start of the month.");

break;

case "August":

console.log("It's the 8th of the year.");

break;

default:

console.log("It's some other month.");

}

Q.6 Find circumference of Rectangle formula : C = 4 \* a ?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Circumference Example</title>

    </script>

</head>

<body>

    <script>

        // Function to calculate the circumference of a square

        function CircumferenceOfSquare(sideLength) {

            // Formula: C = 4 \* a

            const circumference = 4 \* sideLength;

            return circumference;

        }

        // Example usage:

        const sideLength = 5; // Set the length of one side of the square

        const circumference = CircumferenceOfSquare(sideLength);

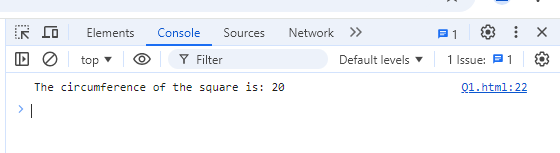
        console.log("The circumference of the square is: " + circumference);

    </script>

</body>

</html>

Output :



Q.7 WAP to convert years into days and days into years?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Converter Example</title>

    </script>

</head>

<body>

    <script>

        // Function to convert years to days

        function convertYearsToDays(years) {

            const days = years \* 365;

            return days;

        }

        // Function to convert days to years

        function convertDaysToYears(days) {

            const years = days / 365;

            return years;

        }

        // Convert years into days

        const years = 3;

        const daysFromYears = convertYearsToDays(years);

        console.log(`${years} years is approximately ${daysFromYears} days.`);

        // Convert days into years

        const days = 365;

        const yearsFromDays = convertDaysToYears(days);

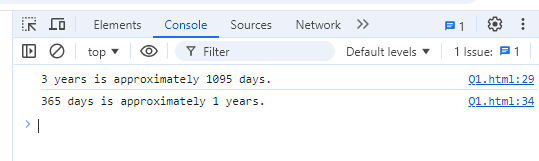
        console.log(`${days} days is approximately ${yearsFromDays} years.`);

    </script>

</body>

</html>

Output :



Q.8 Convert temperature Fahrenheit to Celsius? (Conditional logic Question)

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Temperature Example</title>

    </script>

</head>

<body>

    <script>

        // Function to convert Fahrenheit to Celsius

        function fahrenheitToCelsius(fahrenheit) {

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

            return celsius;

        }

        // Example usage:

        const fahrenheit = 98;  // Set the Fahrenheit temperature

        const celsius = fahrenheitToCelsius(fahrenheit);

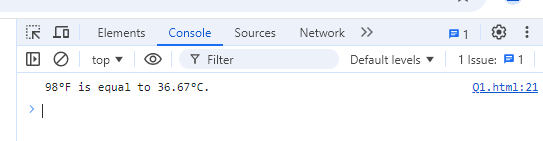
        console.log(`${fahrenheit}°F is equal to ${celsius.toFixed(2)}°C.`);

    </script>

</body>

</html>

Output :



Q.9 Write a JavaScript exercise to get the extension of a filename.?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Filename Extension Example</title>

    </script>

</head>

<body>

    <script>

        // Function to get the extension of a filename

        function getFileExtension(filename) {

            // Use the lastIndexOf method to find the last dot and slice from that position

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

            // Check if there's a dot and if it's not the first character

            if (dotIndex > 0 && dotIndex < filename.length - 1) {

                return filename.slice(dotIndex + 1); // Extract the extension

            } else {

                return "No extension found"; // Handle cases where there's no extension

            }

        }

        // Example usage:

        const filename1 = "document.pdf";

        const filename2 = "my.image.jpeg";

        const filename3 = "noExtensionFile";

        // Print the results

        console.log(getFileExtension(filename1));  // Output: pdf

        console.log(getFileExtension(filename2));  // Output: jpeg

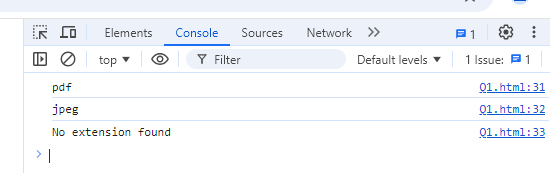
        console.log(getFileExtension(filename3));  // Output: No extension found

    </script>

</body>

</html>

Output :



Q.10 What is the result of the expression (5 > 3 && 2 < 4)?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Filename Extension Example</title>

    </script>

</head>

<body>

    <script>

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

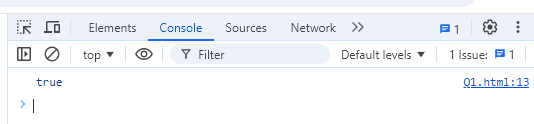
        console.log(result); // Output: true

    </script>

</body>

</html>

Output :



Q.11 What is the result of the expression (true && 1 && "hello")?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Output Example</title>

    </script>

</head>

<body>

    <script>

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

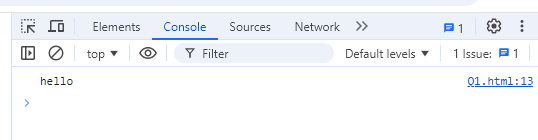
        console.log(result); // Output: true

    </script>

</body>

</html>

Output :



Q.12 What is the result of the expression true && false || false && true?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>True / False Example</title>

    </script>

</head>

<body>

    <script>

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

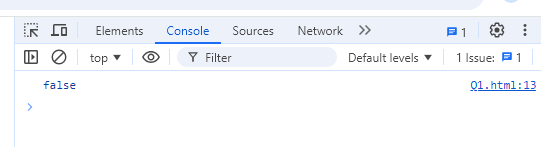
        console.log(result); // Output: true

    </script>

</body>

</html>

Output :



Q.13 Check Number Is Positive or Negative in JavaScript?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Output Example</title>

    </script>

</head>

<body>

    <script>

        function checkNumber(num) {

            if (num > 0) {

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

            } else (num < 0) {

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

            }

        }

        // Example usage:

        checkNumber(18);   // Output: This is a positive number.

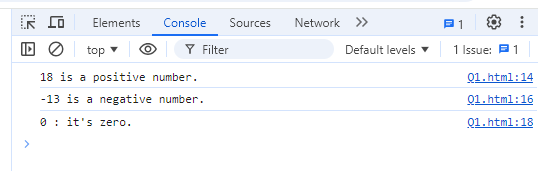
        checkNumber(-13);   // Output: This is a negative number.

    </script>

</body>

</html>

Output :



Q.14 Find the Character Is Vowel or Not ?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Vowel Example</title>

    </script>

</head>

<body>

    <script>

        // Function to check if a character is a vowel

        function isVowel(char) {

            // Convert the character to lowercase to handle both cases

            const lowerChar = char.toLowerCase();

            // Check if the character is one of the vowels

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

                console.log(char + " is a vowel.");

            } else {

                console.log(char + " is not a vowel.");

            }

        }

        // Example usage:

        isVowel('A');  // Output: A is a vowel.

        isVowel('k');  // Output: b is not a vowel.

        isVowel('i');  // Output: e is a vowel.

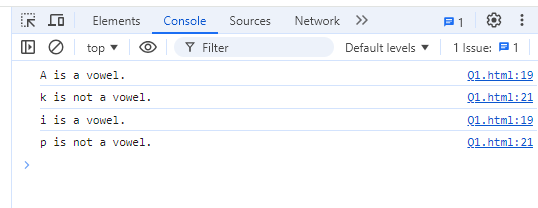
        isVowel('p');  // Output: z is not a vowel.

    </script>

</body>

</html>

Output :



Q.15 Write to check whether a number is negative, positive or zero?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Output Example</title>

    </script>

</head>

<body>

    <script>

        function checkNumber(num) {

            if (num > 0) {

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

            } else if (num < 0) {

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

            } else {

                console.log(num + " : it's zero.");

            }

        }

        // Example usage:

        checkNumber(18);   // Output: This is a positive number.

        checkNumber(-13);   // Output: This is a negative number.

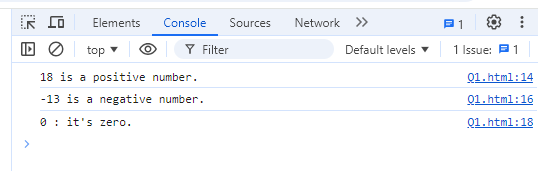
        checkNumber(0);    // Output: This is neither positive nor negative (it's zero).

    </script>

</body>

</html>

Output :



Q.16 Write to find number is even or odd using ternary operator in JS?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Odd/Even Example</title>

    </script>

</head>

<body>

    <script>

        function checkNumber(num) {

            if ( (num % 2) === 0 ){

                console.log(num + " is even number.");

            } else {

                console.log(num + " is odd number.");

            }

        }

        // Example usage:

        checkNumber(18);   // Output: This is a positive number.

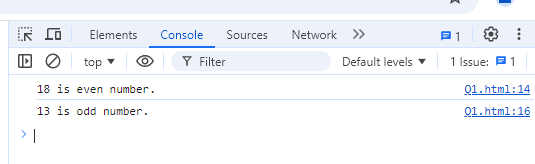
        checkNumber(13);   // Output: This is a negative number.

    </script>

</body>

</html>

Output :



Q.17 Write find maximum number among 3 numbers using ternary operator in JS?

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Maximum Example</title>

    </script>

</head>

<body>

    <script>

        // Function to find the maximum number among three numbers using the ternary operator

        function Maximum(a, b, c) {

            const max = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);

            return max;

        }

        // Example usage:

        const num1 = 10;

        const num2 = 25;

        const num3 = 15;

        const maximum = Maximum(num1, num2, num3);

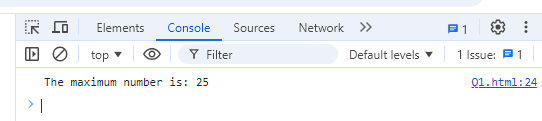
        console.log("The maximum number is: " + maximum);

</script>

</body>

</html>

Output :



Q.18 Write to find minimum number among 3 numbers using ternary operator in JS?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Output Example</title>

    </script>

</head>

<body>

    <script>

        // Function to find the minimum number among three numbers using the ternary operator

        function findMin(a, b, c) {

            const min = (a < b) ? ((a < c) ? a : c) : ((b < c) ? b : c);

            return min;

        }

        // Example usage:

        const num1 = 10;

        const num2 = 25;

        const num3 = 15;

        const minimum = findMin(num1, num2, num3);

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

    </script>

</body>

</html>

Output :



Q.19 Write to find the largest of three numbers in JS?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Largest Example</title>

    </script>

</head>

<body>

    <script>

        // Function to find the largest of three numbers using conditional statements

        function findLargest(a, b, c) {

            let largest;

            if (a >= b && a >= c) {

                largest = a;

            } else if (b >= a && b >= c) {

                largest = b;

            } else {

                largest = c;

            }

            return largest;

        }

        // Example usage:

        const num1 = 10;

        const num2 = 25;

        const num3 = 15;

        const largest = findLargest(num1, num2, num3);

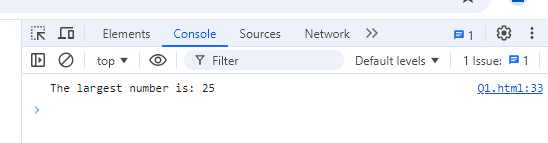
        console.log("The largest number is: " + largest);

    </script>

</body>

</html>

Output :



Q.20 Write to show

i. Monday to Sunday using switch case in JS?

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Weekdays Example</title>

    </script>

</head>

<body>

    <script>

        // Function to get the day of the week based on a number

        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! Please enter a number between 1 and

7.";

            }

            return day;

        }

        // Example usage:

        const dayNumber = 3; // Change this number to test with different days

        const dayOfWeek = getDayOfWeek(dayNumber);

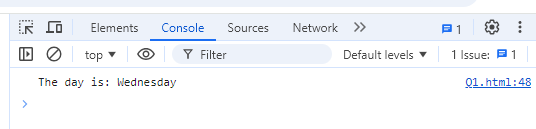
        console.log("The day is: " + dayOfWeek);

    </script>

</body>

</html>

**Output :**

****

ii. Vowel or Consonant using switch case in JS?

Ans.

<html lang="en">

<head>

    <meta charset="UTF-8" />

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

    <title>Weekdays Example</title>

    </script>

</head>

<body>

    <script>

        // Function to check if a character is a vowel or consonant

        function checkVowelOrConsonant(char) {

            // Convert the character to lowercase for case-insensitivity

            const lowerChar = char.toLowerCase();

            switch (lowerChar) {

                case 'a':

                case 'e':

                case 'i':

                case 'o':

                case 'u':

                    console.log(char + " is a vowel.");

                    break;

                default:

                    if (lowerChar >= 'a' && lowerChar <= 'z') {

                        console.log(char + " is a consonant.");

                    } else {

                        console.log(char + " is not a valid letter.");

                    }

            }

        }

        // Example usage:

        checkVowelOrConsonant('A');  // Output: A is a vowel.

        checkVowelOrConsonant('K');  // Output: b is a consonant.

        checkVowelOrConsonant('E');  // Output: E is a vowel.

        checkVowelOrConsonant('P');  // Output: z is a consonant.

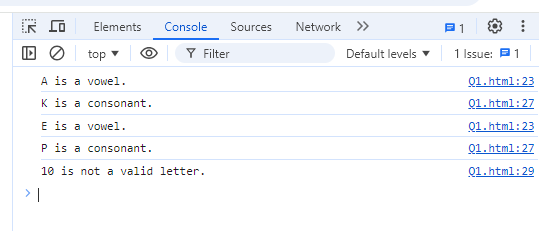
        checkVowelOrConsonant('10');  // Output: 1 is not a valid letter.

    </script>

</body>

</html>

Output :

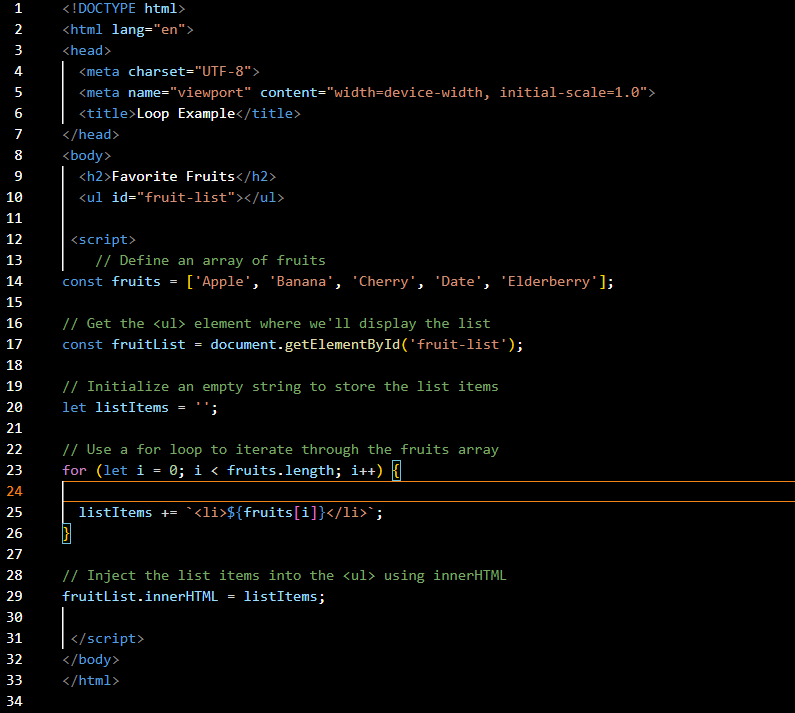


**(Conditional looping logic Question)**

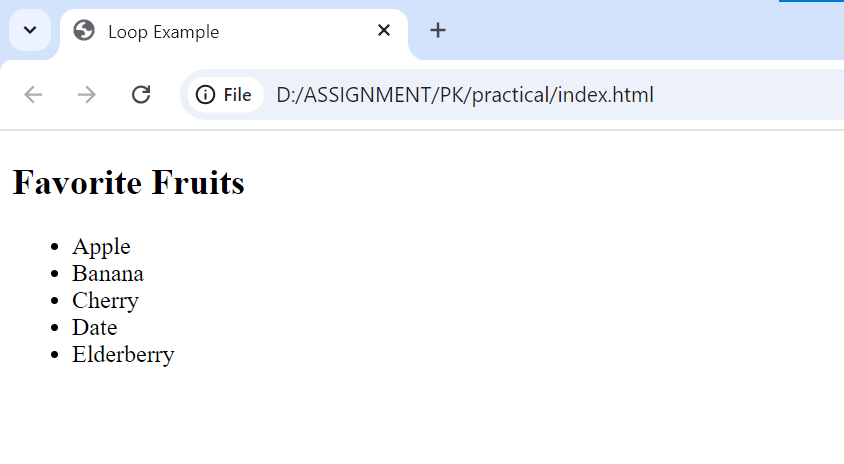
Q.21 What are the looping structures in JavaScript? Any one Example?

Ans - JavaScript offers several looping structures that allow you to iterate over collections, arrays, or execute a block of code multiple times. Here are the main looping structures:

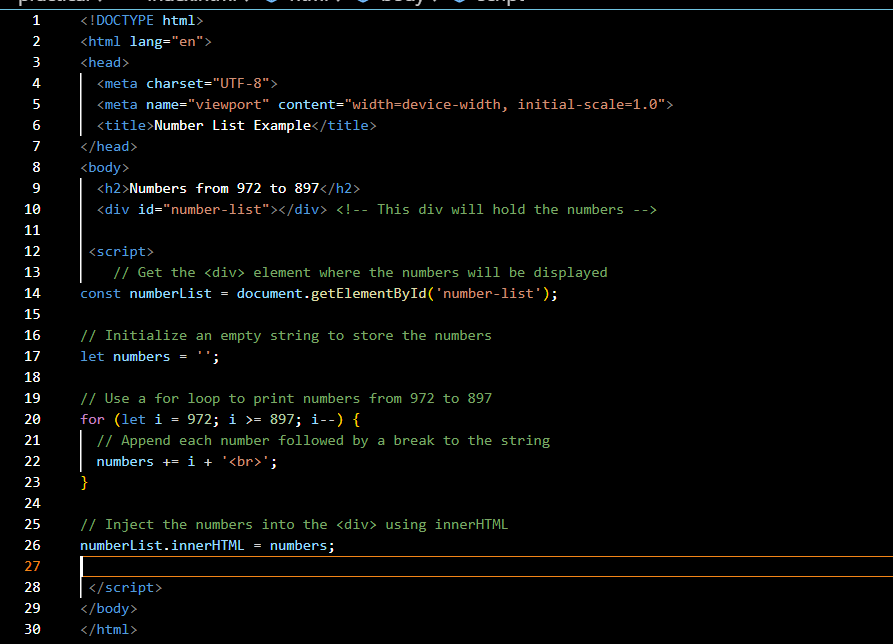
1. **for loop**
2. **while loop**
3. **do...while loop**
4. **for...in loop** (used for iterating over properties of an object)
5. **for...of loop** (used for iterating over iterable objects like arrays)

example of for loop  


output :



Q.22 Write a print 972 to 897 using for loop in JS?

Ans - 

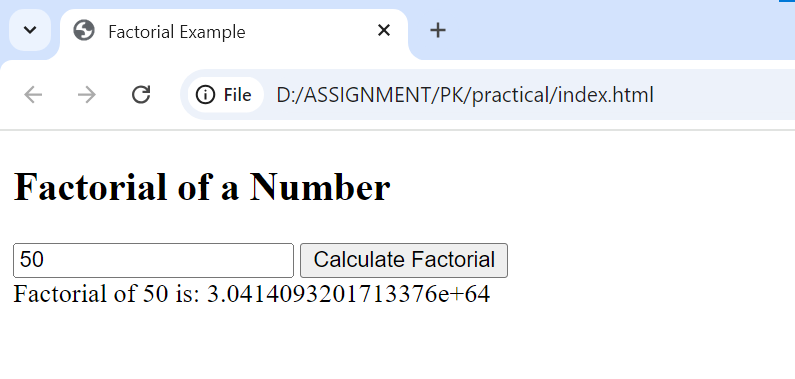
Output: 



Q.23 Write to print factorial of given number?



Output –



Q.24 Write to print Fibonacci series up to given numbers?\

Ans –



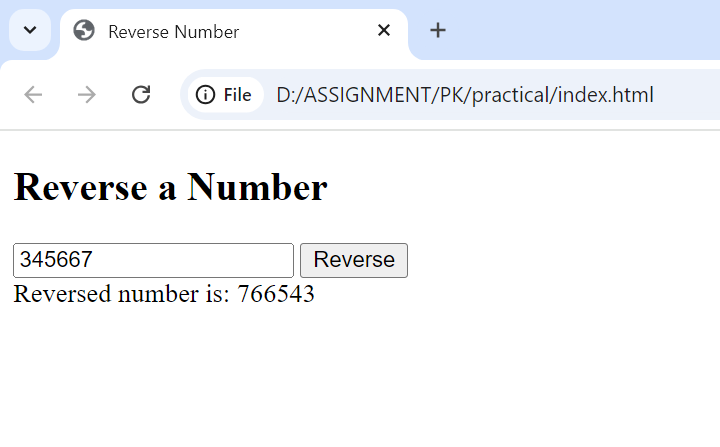
Output :



Q.25 Write to print number in reverse order e.g.: number = 64728 ---> reverse =82746 in JS?

Ans - 

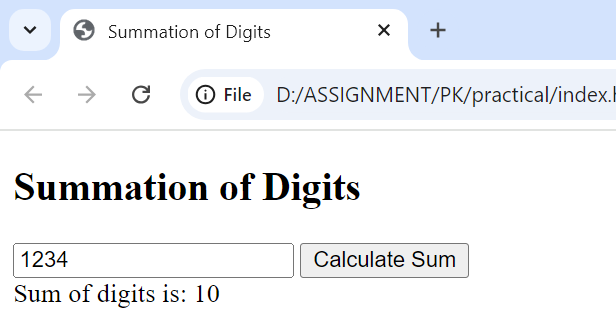
Ans –



Q.26 Write a program make a summation of given number (E.g., 1523 Ans: - 11) in JS?

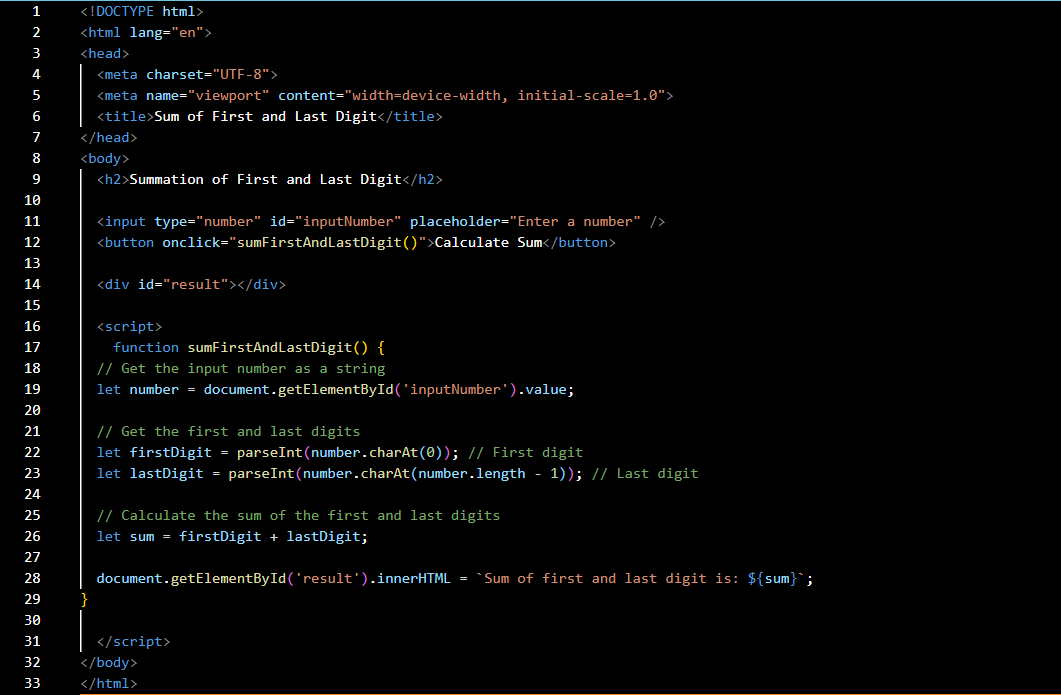
Ans - 

Output :

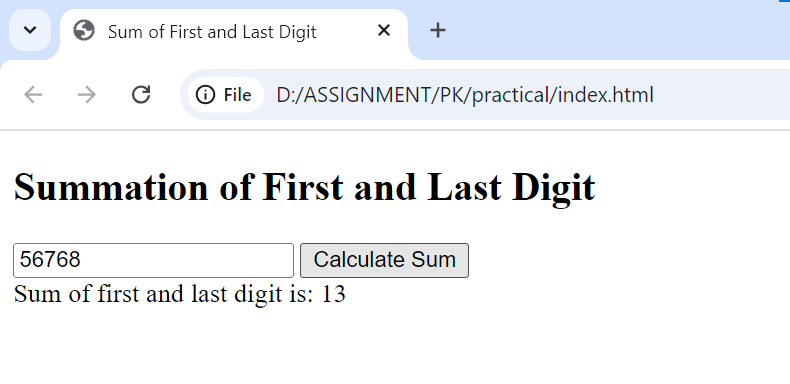


Q.27 Write a program you have to make a summation of first and last Digit. (E.g., 1234 Ans: -

5) in JS?



Output :



Q.28 Use console.log() and escape characters to print the following pattern in JS?

1 1 1 1

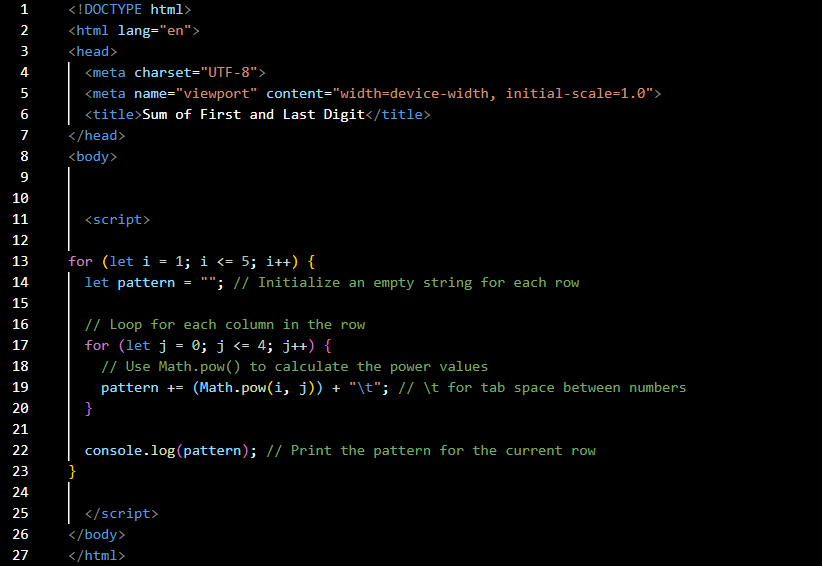
2 1 2 4 8

3 1 3 9 27

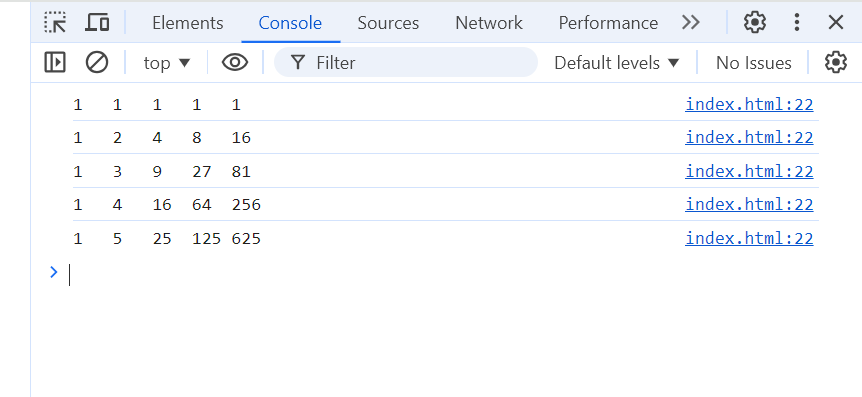
4 1 4 16 64

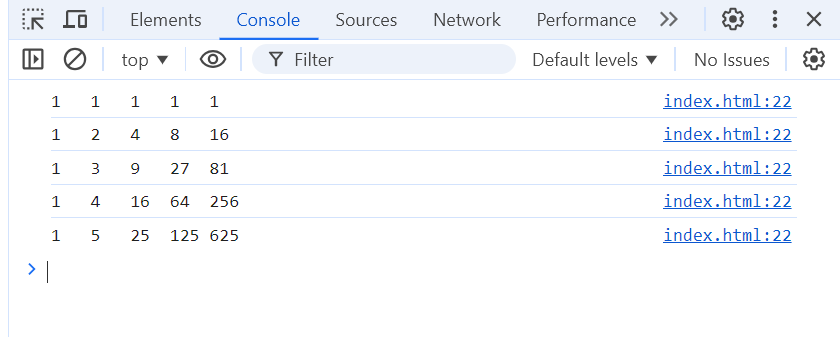
5 1 5 25 125

Ans –



Output :





Q.29 Use pattern in console.log in JS?

1) 1

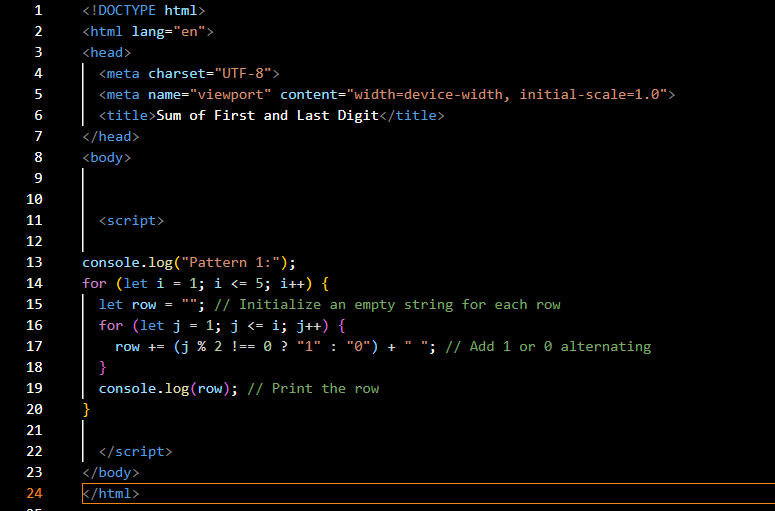
1 0

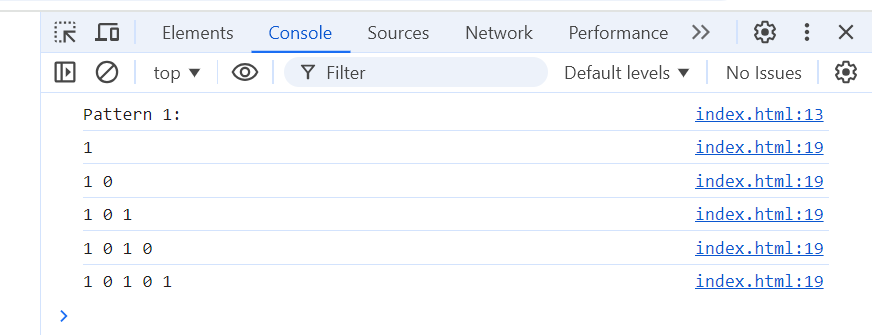
1 0 1

1 0 1 0

1 0 1 0 1

Ans –





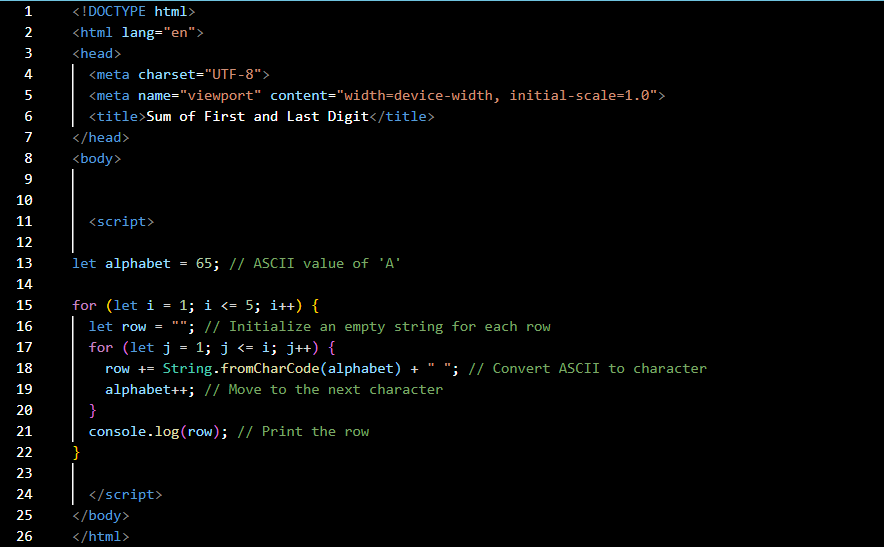
2) A

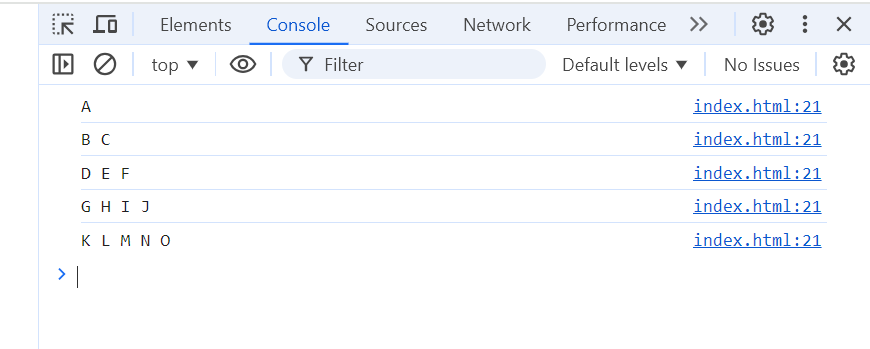
B C

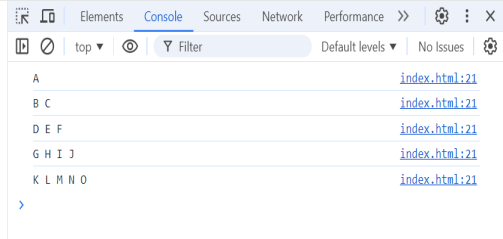
D E F

G H I J

K L M N O



Output – 



3) 1

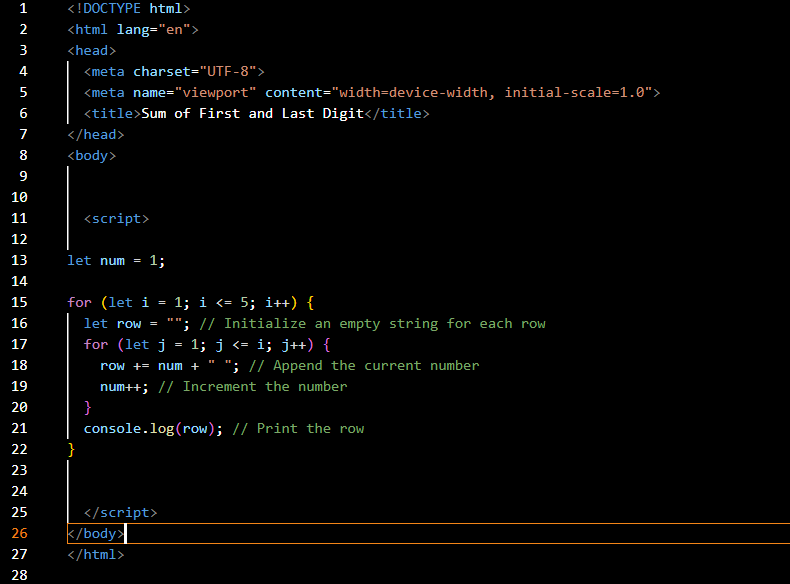
2 3

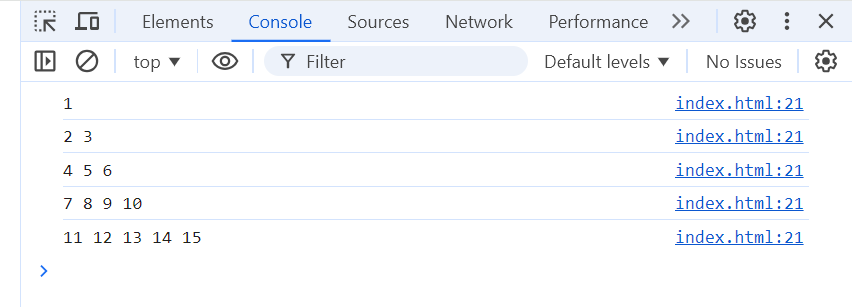
4 5 6

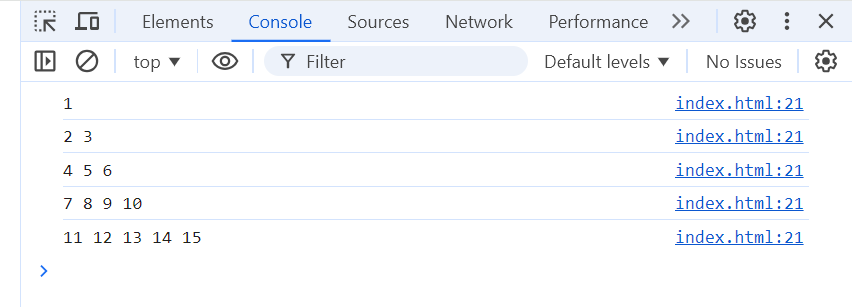
7 8 9 10

11 12 13 14 15

Ans –







4) \*

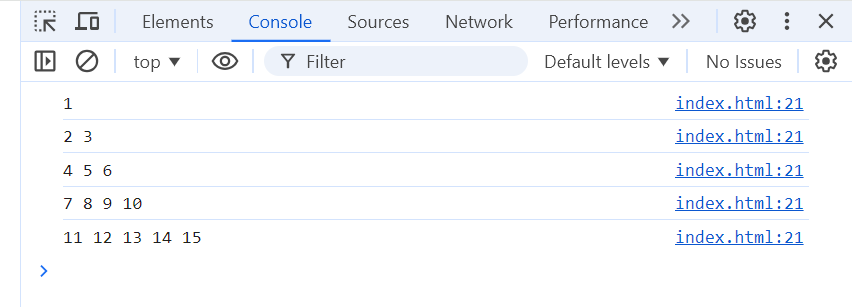
\* \*

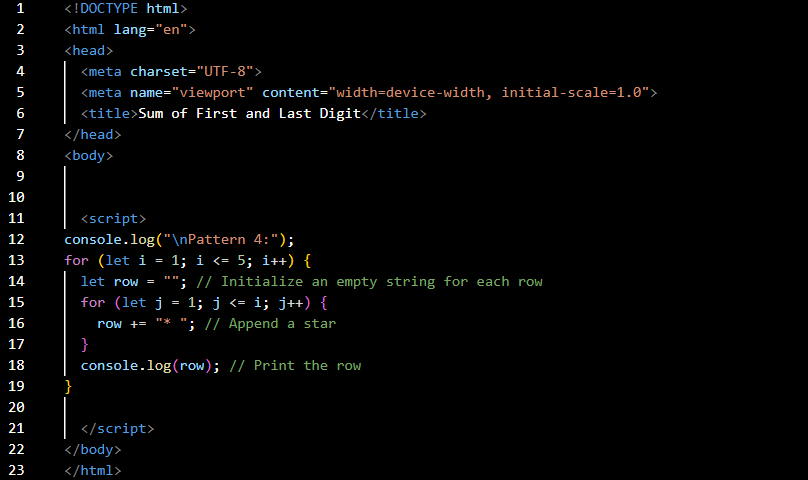
\* \* \*

\* \* \* \*

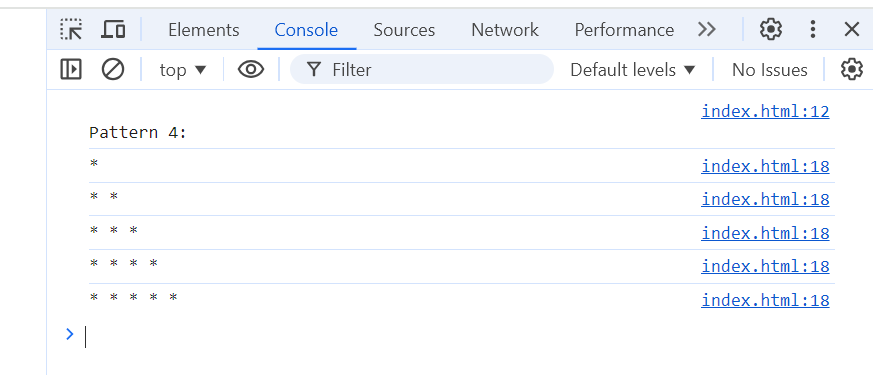
\* \* \* \* \*

Ans -





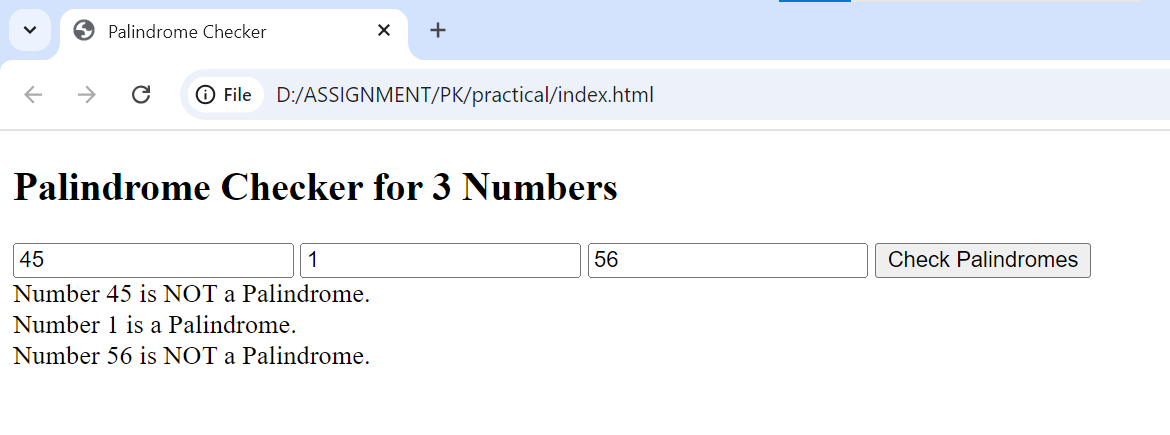
Out put –



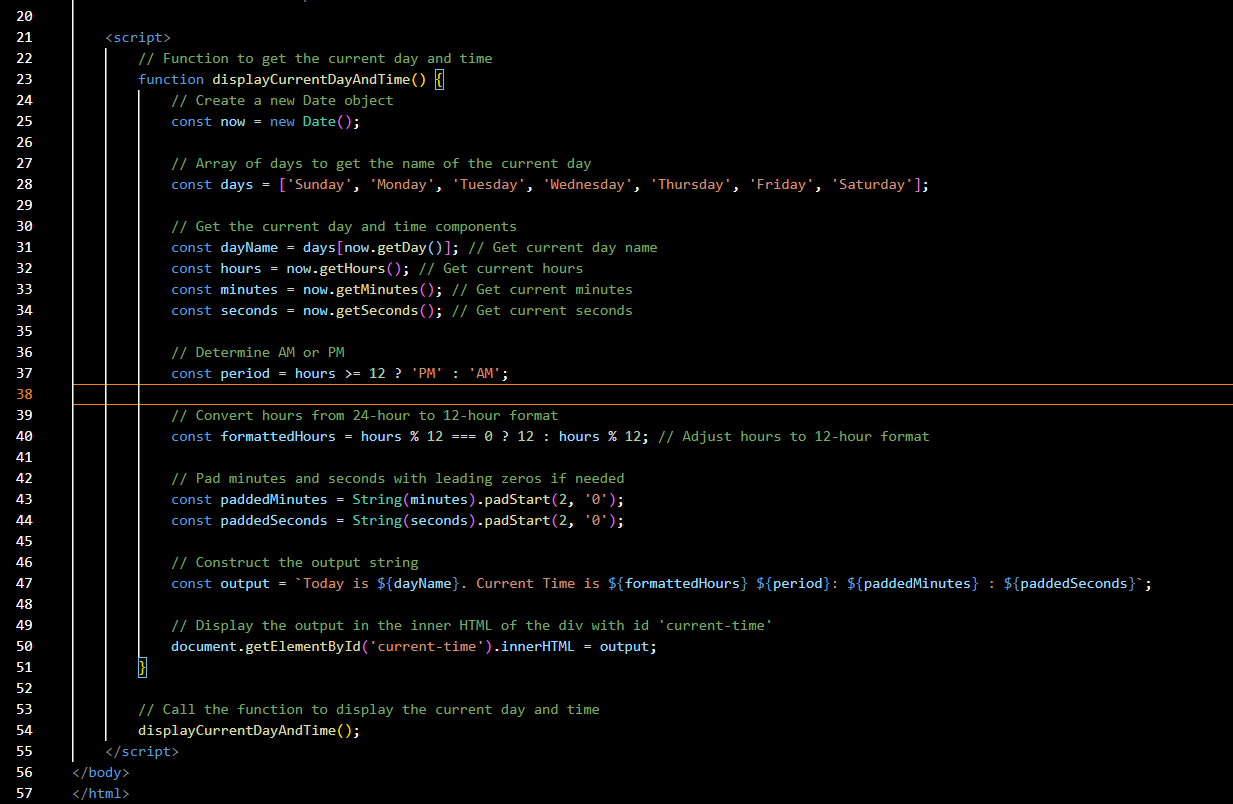
**Q.30** Accept 3 numbers from user using while loop and check each numbers palindrome



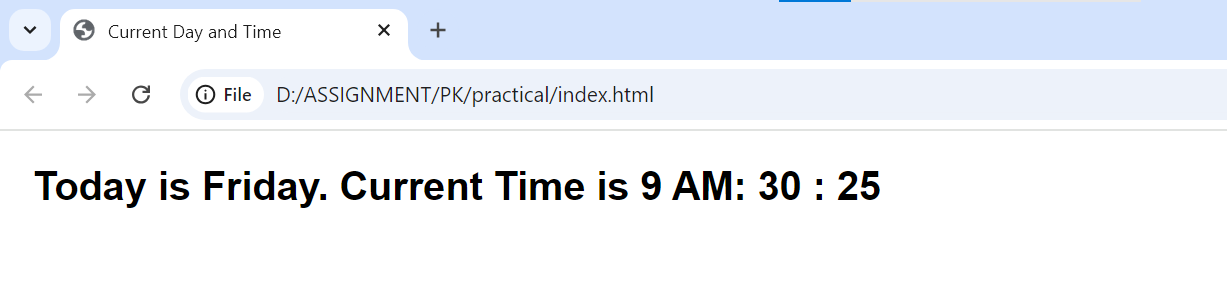
Output –



Q.31 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 ?

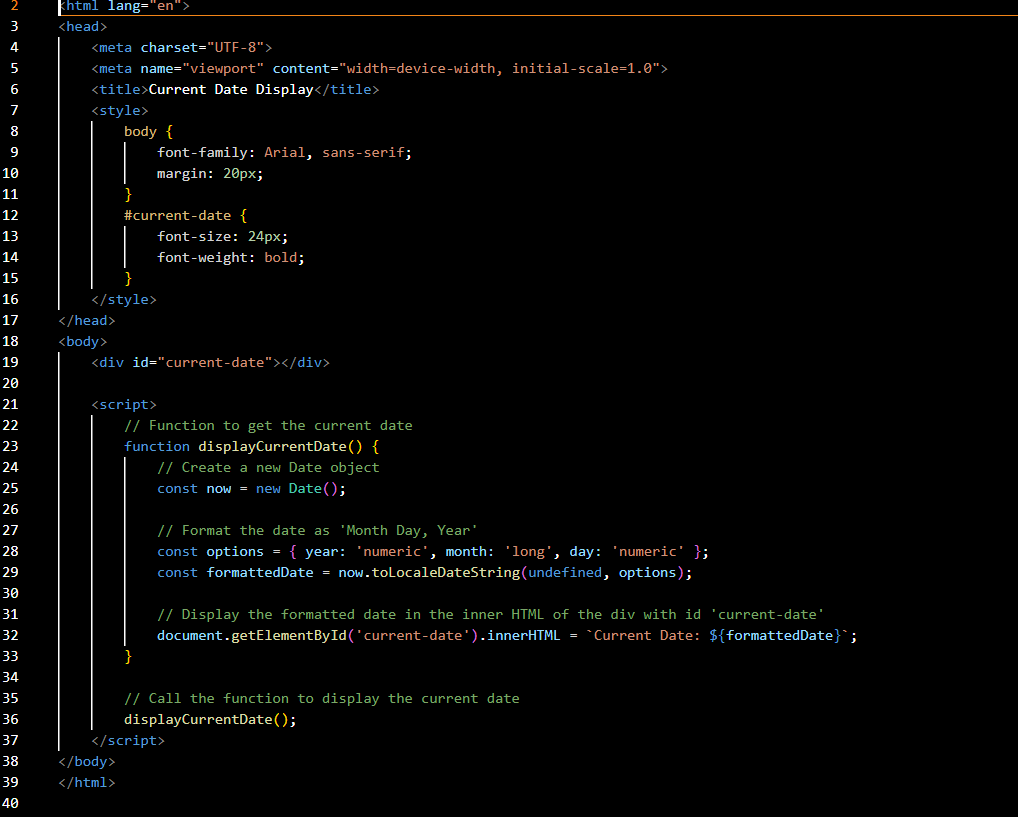


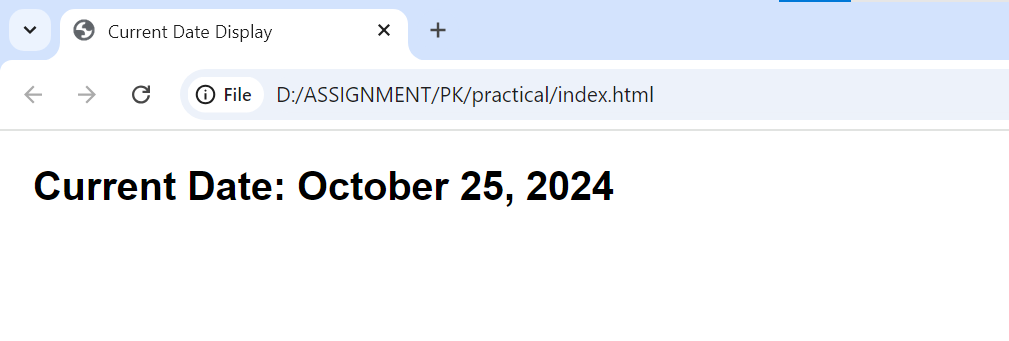
Output :



Q.32 Write a JavaScript program to get the current date?

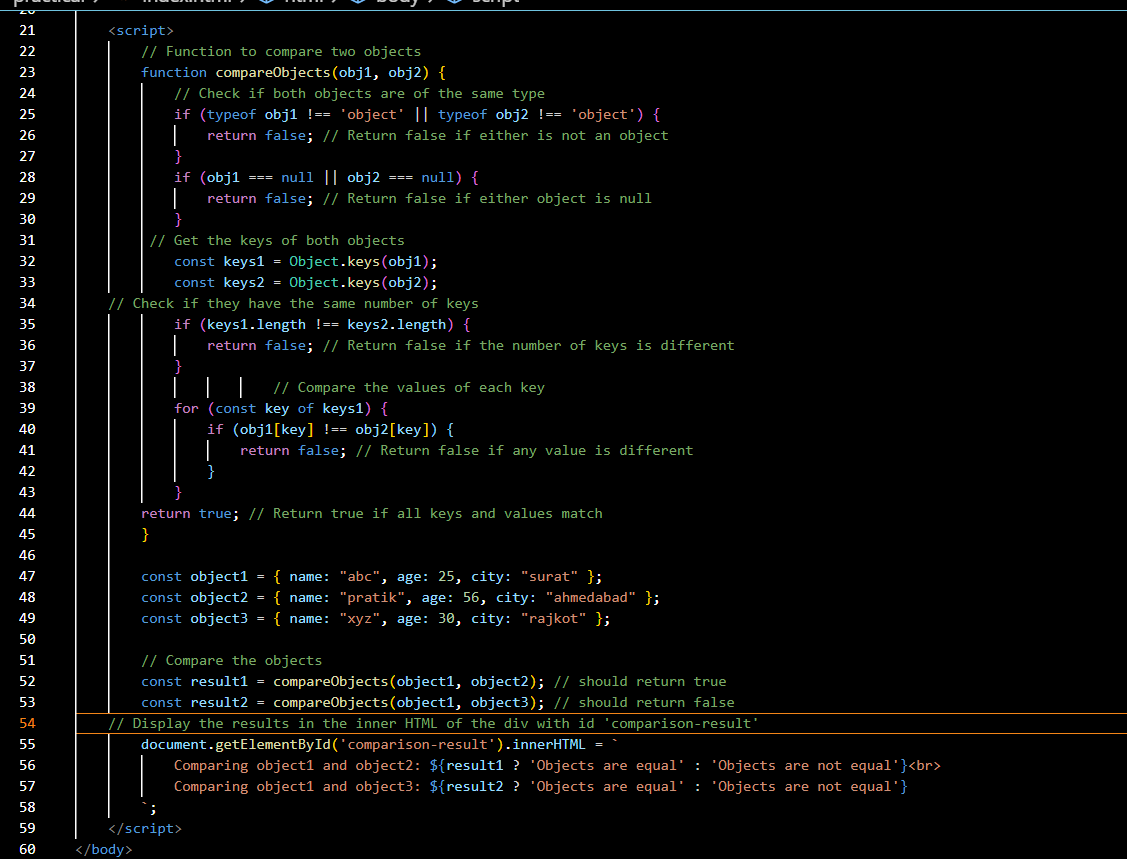
Ans –



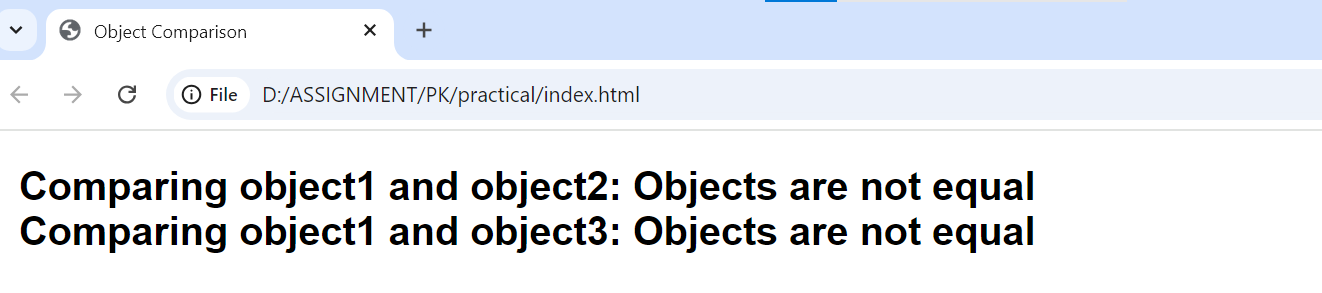
Output : 

Q.33 Write a JavaScript program to compare two objects?

Ans –

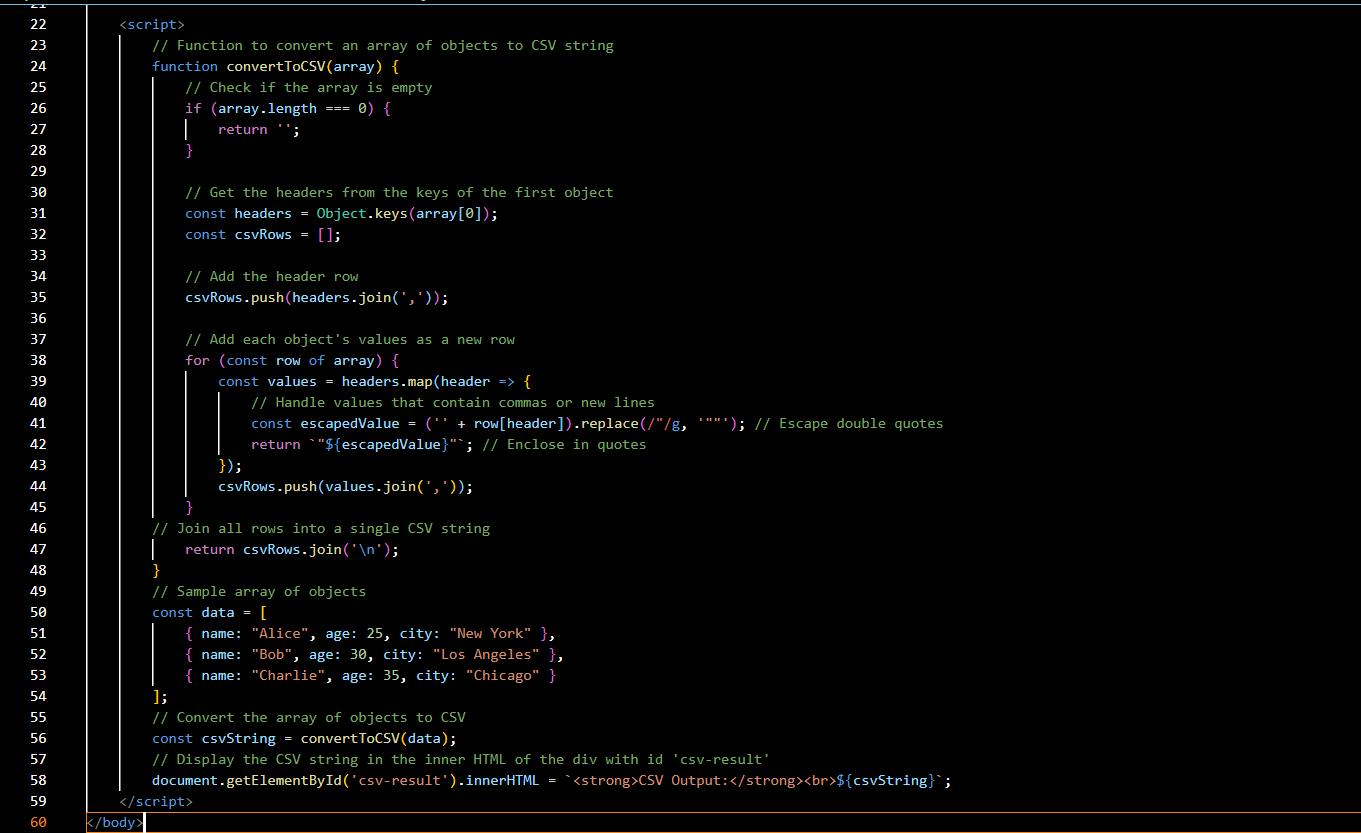


Output :

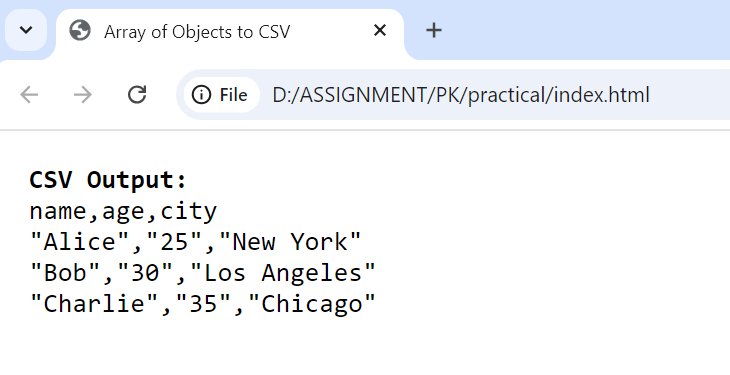


Q.34 Write a JavaScript program to convert an array of objects into CSV string?

Ans –



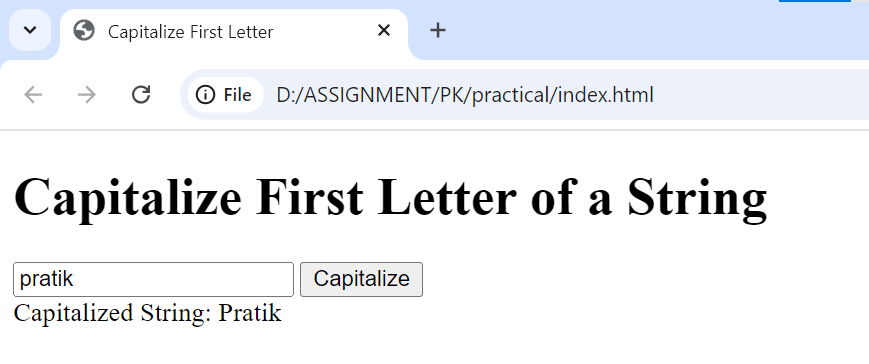
Output :



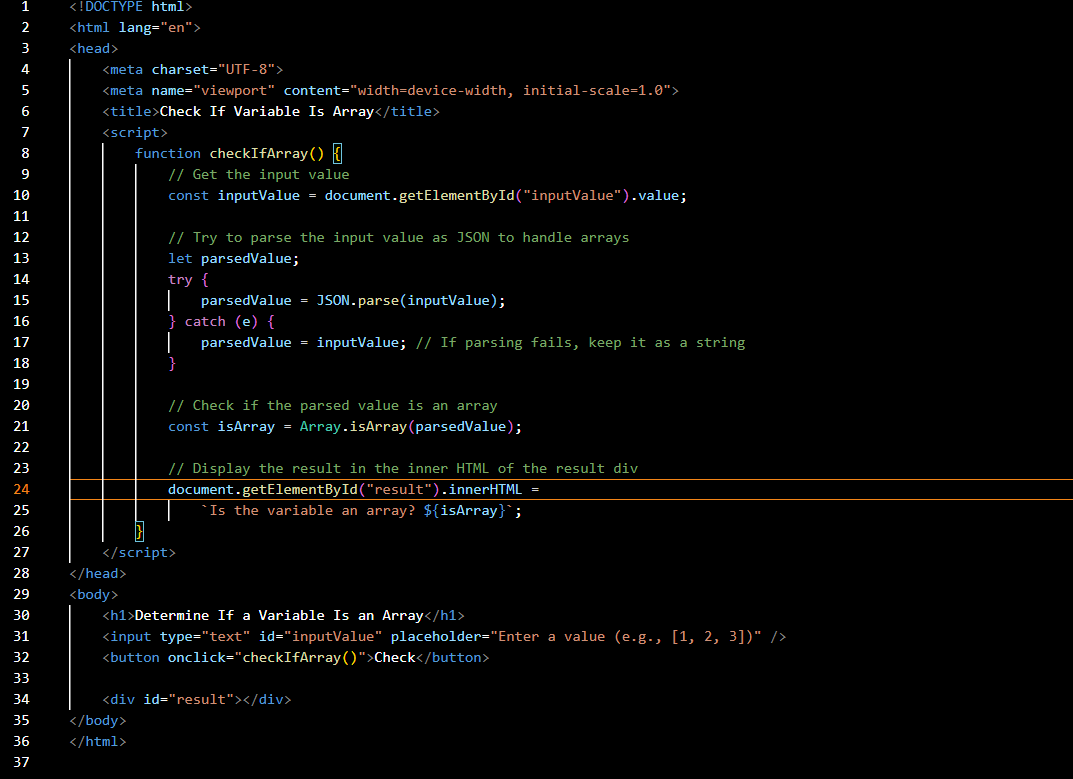
Q.35 Write a JavaScript program to capitalize first letter of a string?



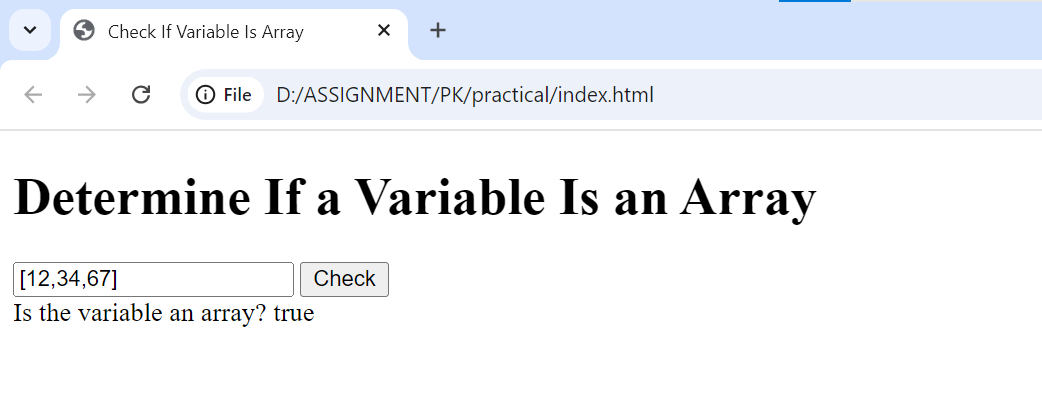
Output



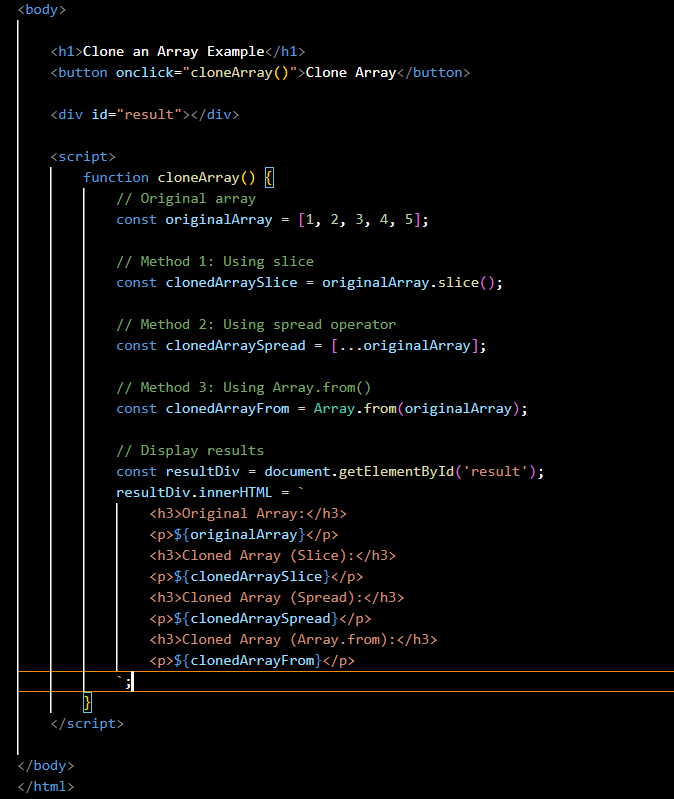
Q. 45 Write a JavaScript program to determine if a variable is array?

Ans - 

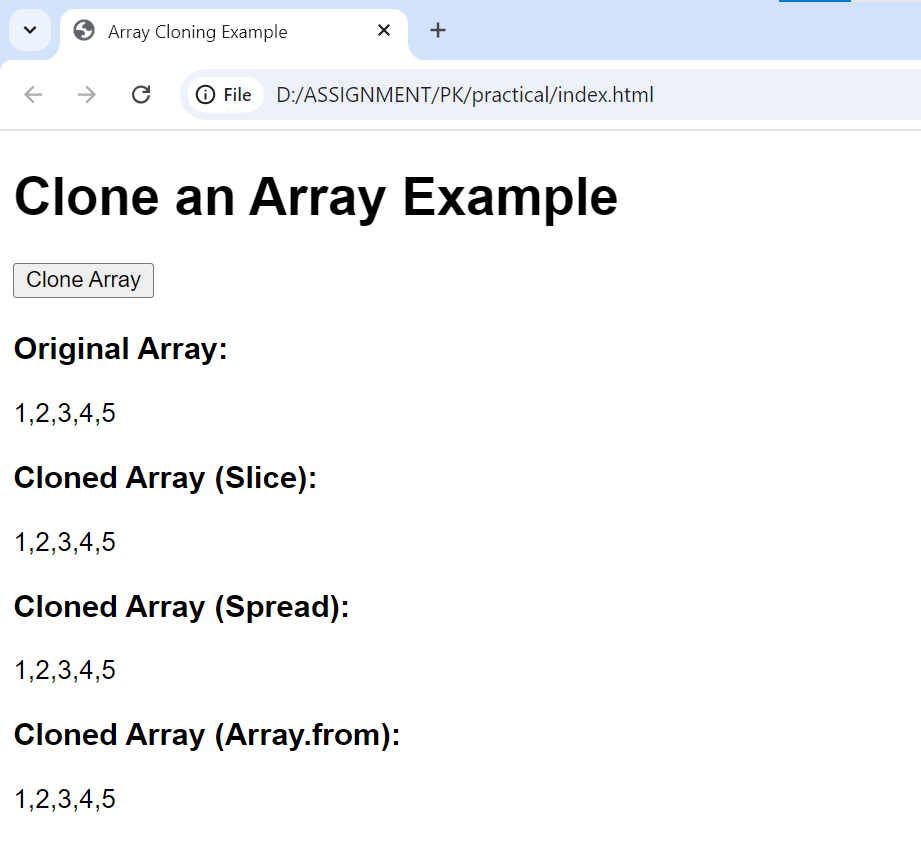
Output :



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

Ans - 

Output :



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

Ans - Declaring methods directly in JavaScript objects has several drawbacks. Here are some key points to consider:

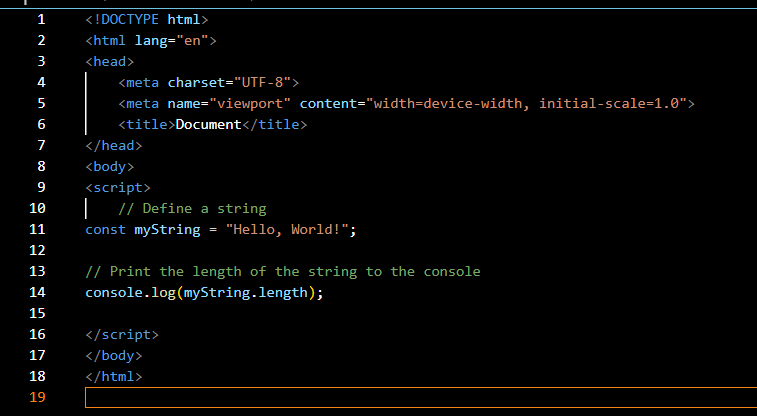
1. **Prototype Chain Limitation**:
   * When methods are defined directly within an object, they are not shared across instances. Each instance gets its own copy of the method, which can increase memory usage.
   * If you need multiple objects with the same methods, you'll have to redefine those methods for each object, leading to redundancy.
2. **Performance Overhead**:

Since each instance has its own copy of the method, it can lead to higher memory consumption, especially if you create many instances of the object.

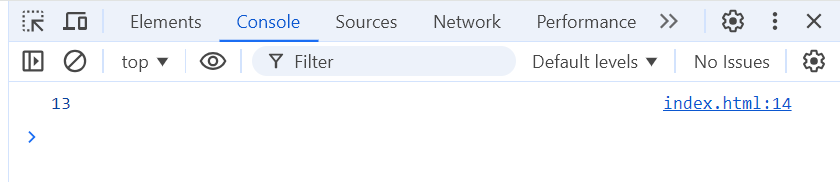
If the method is complex or frequently used, the repeated declaration may negatively impact performance.

1. **Inflexibility**:
   * Directly declared methods can be less flexible when it comes to inheritance. If you need to extend an object or create subclasses, modifying methods defined directly in objects can be cumbersome.
   * Using the prototype allows for easier extension and modification of methods for all instances of the object.
2. **Maintenance Challenges**:
   * When methods are duplicated across multiple object instances, making changes to those methods requires updating each instance. This can lead to errors or inconsistencies if one instance is missed.
   * Maintenance becomes more complex as the codebase grows, particularly when the same logic is needed in multiple places.
3. **Limited to Object Instances**:
   * Methods declared directly in an object can only be called on that specific instance. If you want to use them in a different context, you have to rebind the context or copy the function, which can be error-prone.

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

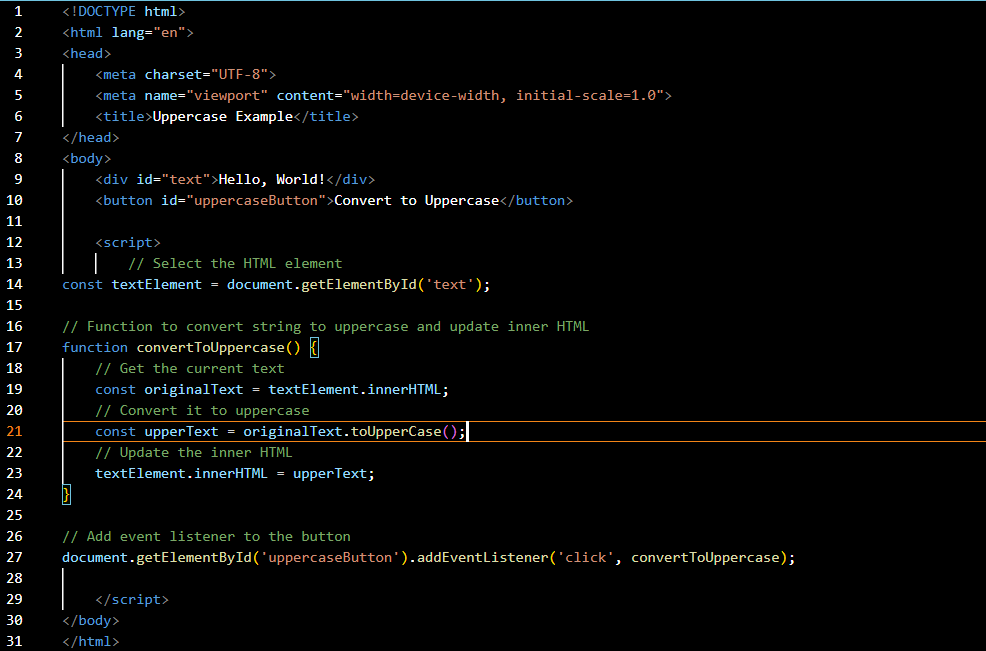
Ans - 

Output :

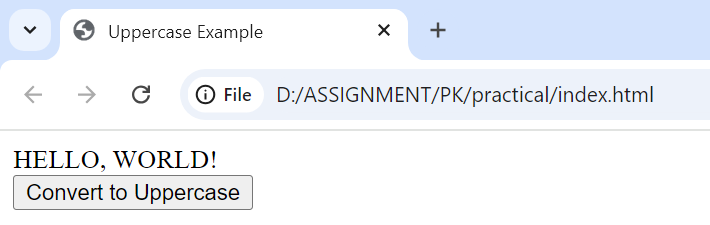


Q.49 Change all the string characters to capital letters using toUpperCase() method?

Ans –



Output –



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

Declaring methods directly in JavaScript objects has several drawbacks. Here are some key points to consider:

1. **Prototype Chain Limitation**:
   * When methods are defined directly within an object, they are not shared across instances. Each instance gets its own copy of the method, which can increase memory usage.
   * If you need multiple objects with the same methods, you'll have to redefine those methods for each object, leading to redundancy.
2. **Performance Overhead**:

Since each instance has its own copy of the method, it can lead to higher memory consumption, especially if you create many instances of the object.

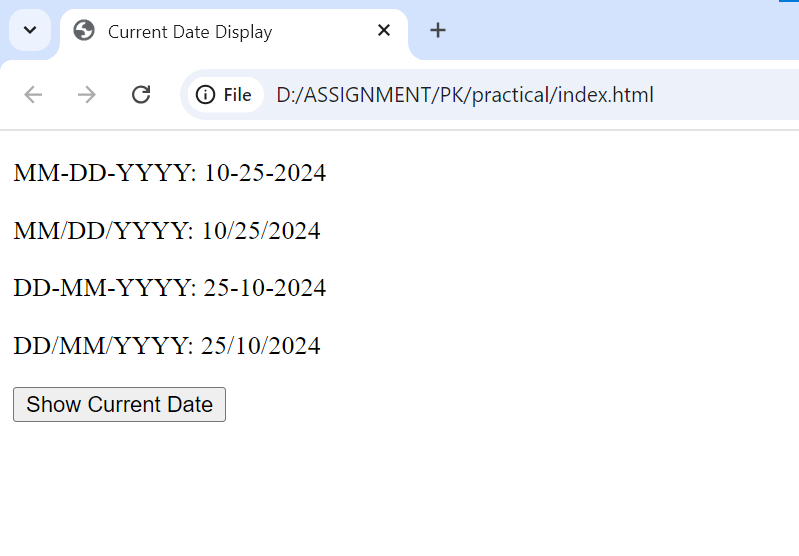
If the method is complex or frequently used, the repeated declaration may negatively impact performance.

1. **Inflexibility**:
   * Directly declared methods can be less flexible when it comes to inheritance. If you need to extend an object or create subclasses, modifying methods defined directly in objects can be cumbersome.
   * Using the prototype allows for easier extension and modification of methods for all instances of the object.
2. **Maintenance Challenges**:
   * When methods are duplicated across multiple object instances, making changes to those methods requires updating each instance. This can lead to errors or inconsistencies if one instance is missed.
   * Maintenance becomes more complex as the codebase grows, particularly when the same logic is needed in multiple places.
3. **Limited to Object Instances**:
   * Methods declared directly in an object can only be called on that specific instance. If you want to use them in a different context, you have to rebind the context or copy the function, which can be error-prone.

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?

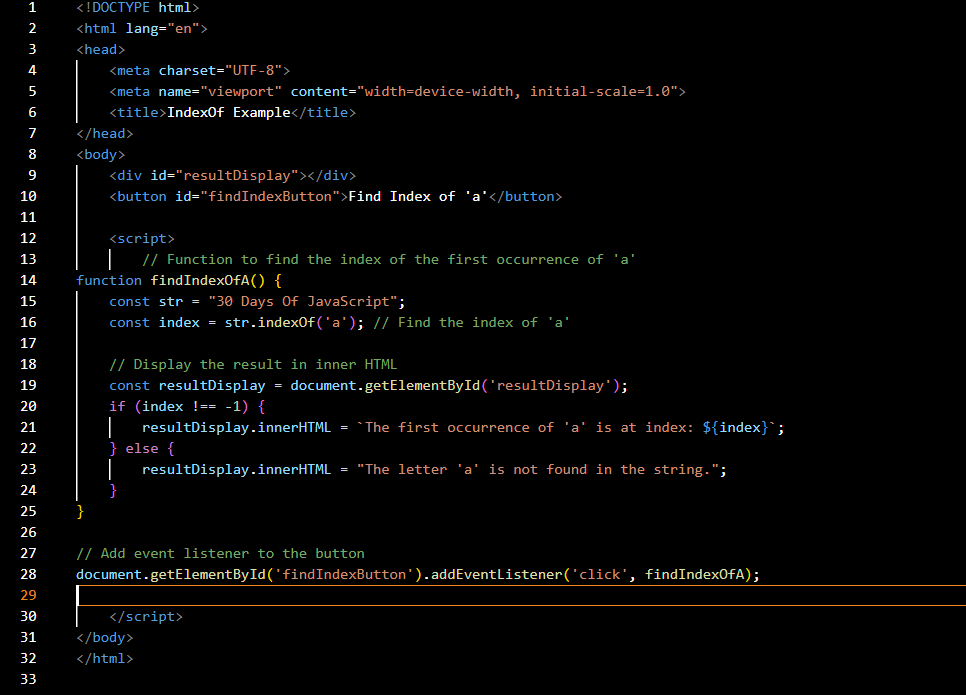


output

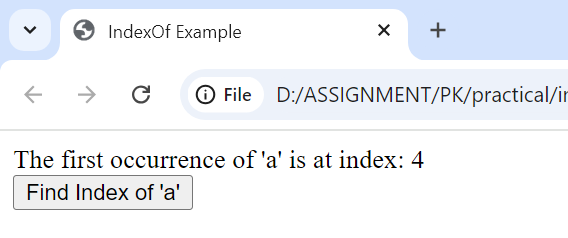
: 

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

Of JavaScript?

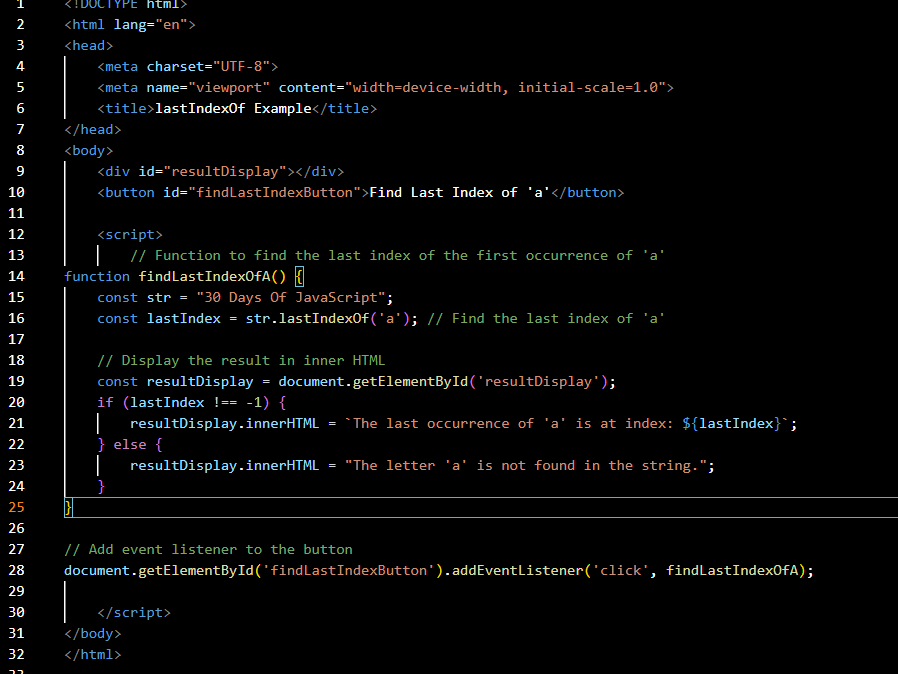


Output :

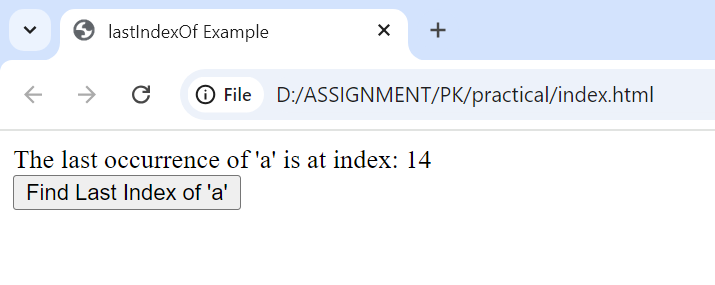


Q,53 Use lastIndexOf to determine the position of the last occurrence of a in 30 Days Of

JavaScript?



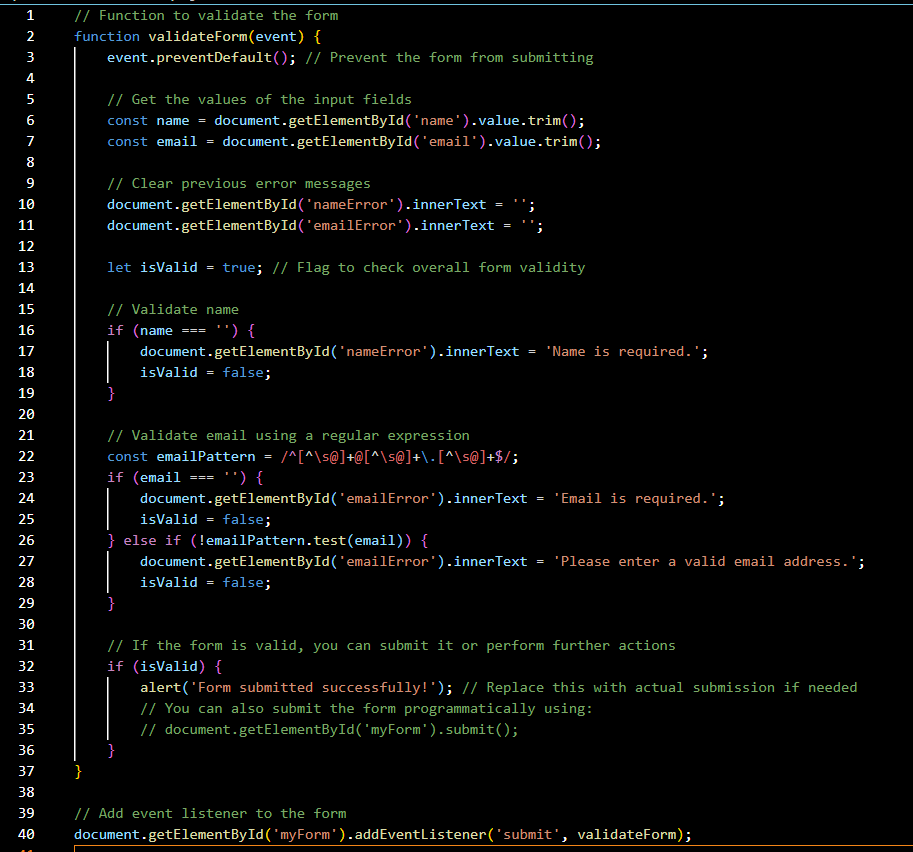
Output :



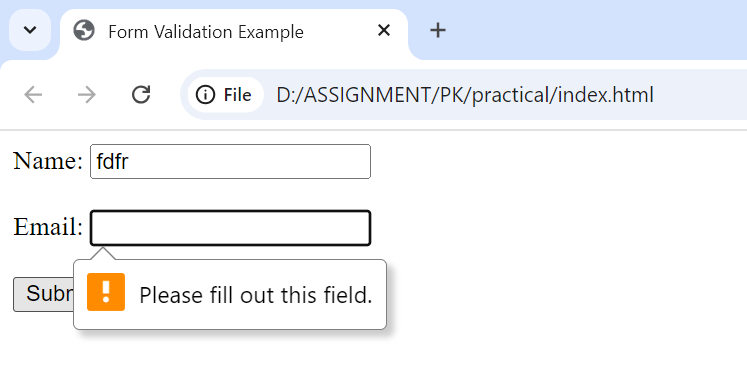
Q.54 Form Validtion in JS?

Ans :





Output :



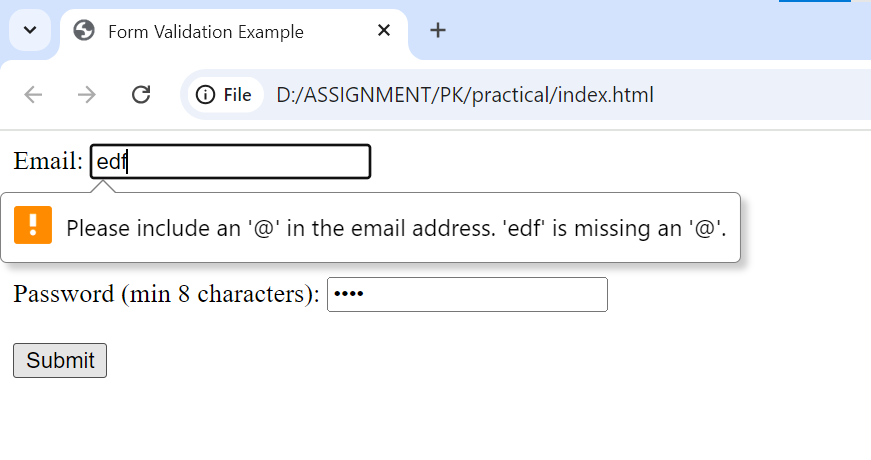
Q.55 Form in Email, number, Password, Validation?

Ans –





Output :



Q.56 Dynamic Form Validation in JS?

Dynamic form validation in JavaScript is about providing real-time feedback to users as they interact with form fields.

This makes forms more user-friendly and helps prevent submission of invalid data.

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

1.onchange = An HTML element has been changed

<button onchange ="document.getElementById('demo').innerHTML = Date()">The time is?</button>

2.onclick = The user clicks an HTML element

<button onclick="document.getElementById('demo').innerHTML = Date()">The time is?</button>

3.onmouseover = The user moves the mouse over an HTML element

<button onmouseover ="document.getElementById('demo').innerHTML = Date()">The time is?</button>

4.onmouseout=The user moves the mouse away from an HTML element

<button onmouseout ="document.getElementById('demo').innerHTML = Date()">The time is?</button>

5.onkeydown = The user pushes a keyboard key

<button onkeydown ="document.getElementById('demo').innerHTML = Date()">The time is?</button>

6.onload = The browser has finished loading the page

<button onload ="document.getElementById('demo').innerHTML = Date()">The time is?</button>

Q.60 What is Bom vs Dom in JS?

**BOM (Browser Object Model):**

* BOM refers to objects provided by the browser that allow interaction with the browser itself.
* Examples include window, navigator, screen, location, history, and alert().
* It allows JavaScript to manipulate the browser window and other browser functionalities, like navigating, getting browser information, etc.

**DOM (Document Object Model):**

* DOM is a programming interface for web documents. It represents the structure of a webpage as a tree of objects (nodes).
* JavaScript can interact with and manipulate the content, structure, and style of a webpage using the DOM.
* Examples include document.getElementById(), document.querySelector(), and element.innerHTML.

Q.61 Array vs object defences in JS?

### Arrays

### Arrays are ordered collections of values, which can be accessed by their index

### They are created using square brackets [] or the Array constructor.

### Syntax:

### let arr = [1, 2, 3, 4];

### let arr2 = new Array(1, 2, 3, 4);

### Each element in an array has a numerical index starting from 0.

### Arrays have a length property that dynamically updates as elements are added or removed.

### Objects

### Objects are collections of key-value pairs, where keys (also called properties) are strings or symbols, and values can be of any data type.

### They are created using curly braces {} or the Object constructor.

### Syntax:

### let obj = { name: "Alice", age: 25 };

### let obj2 = new Object();

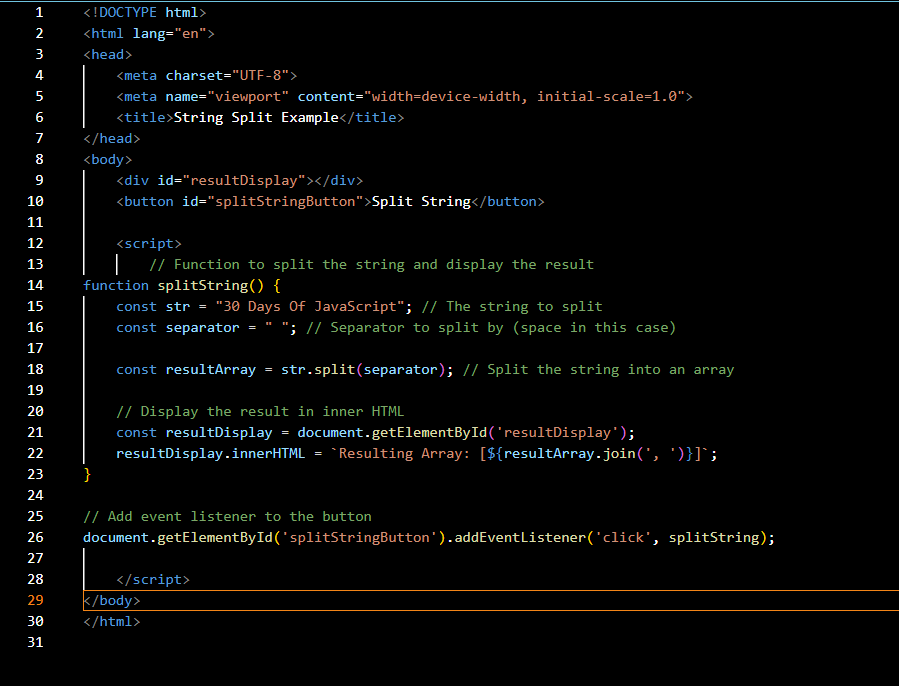
### obj2.name = "Alice";

### obj2.age = 25;

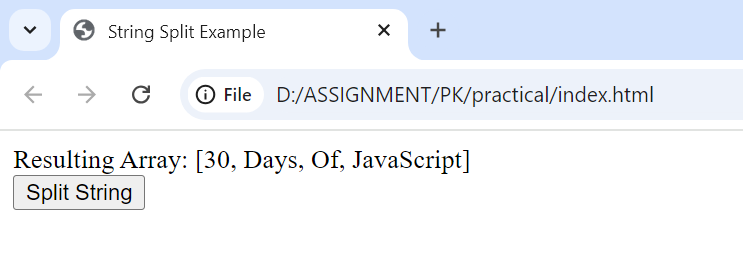
### Each value is accessed using a unique key.

### Properties can be added, modified, or deleted at runtime.

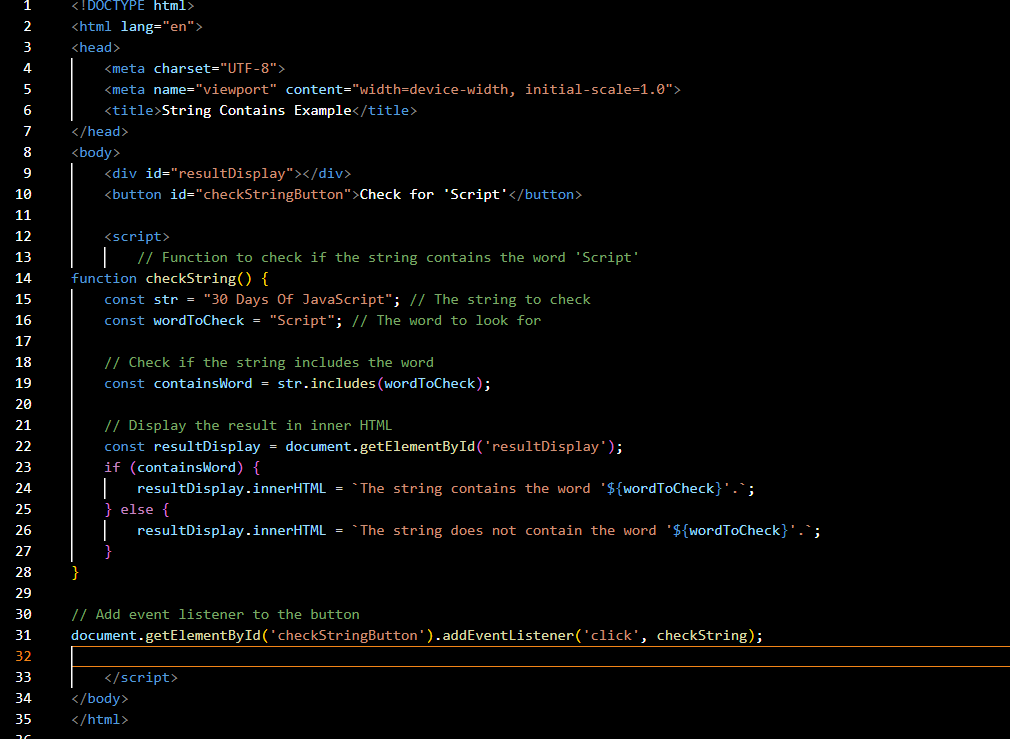
Q.62 Split the string into an array using split() Method?

Ans - 

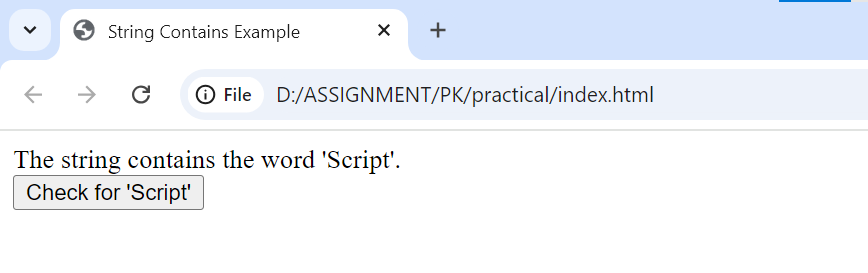
Output :



Q.63 Check if the string contains a word Script using includes() method?

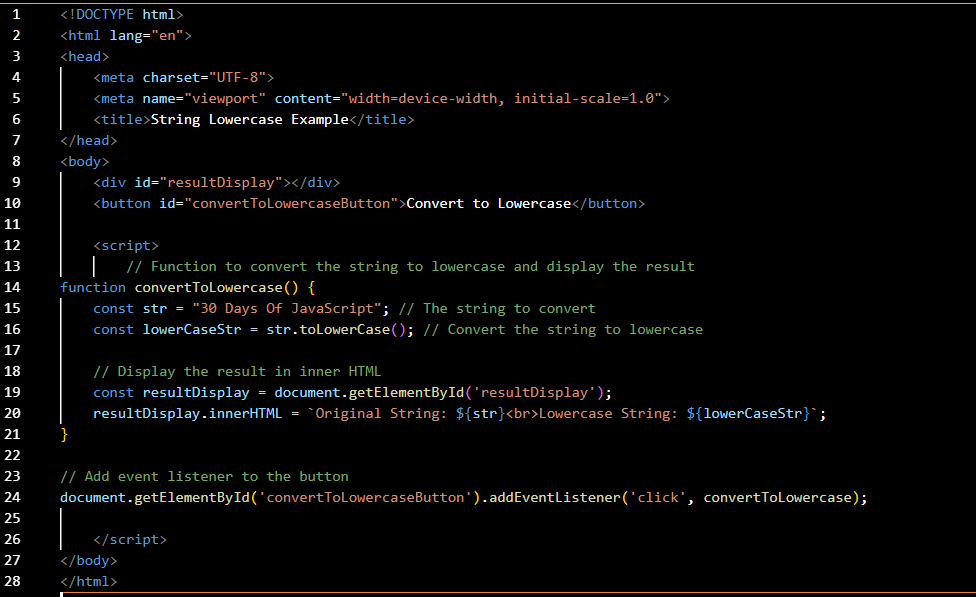
Ans - 

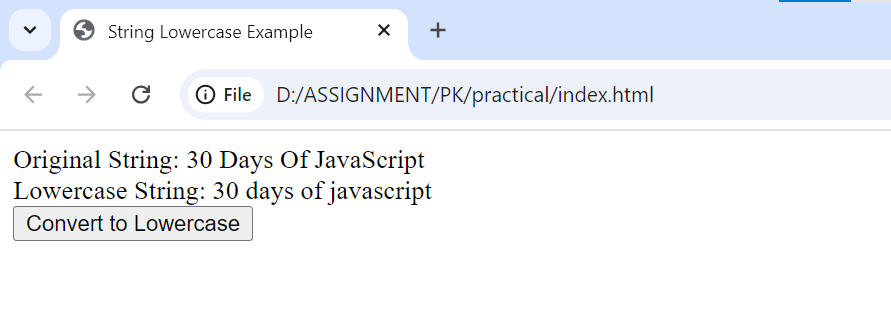
Output :



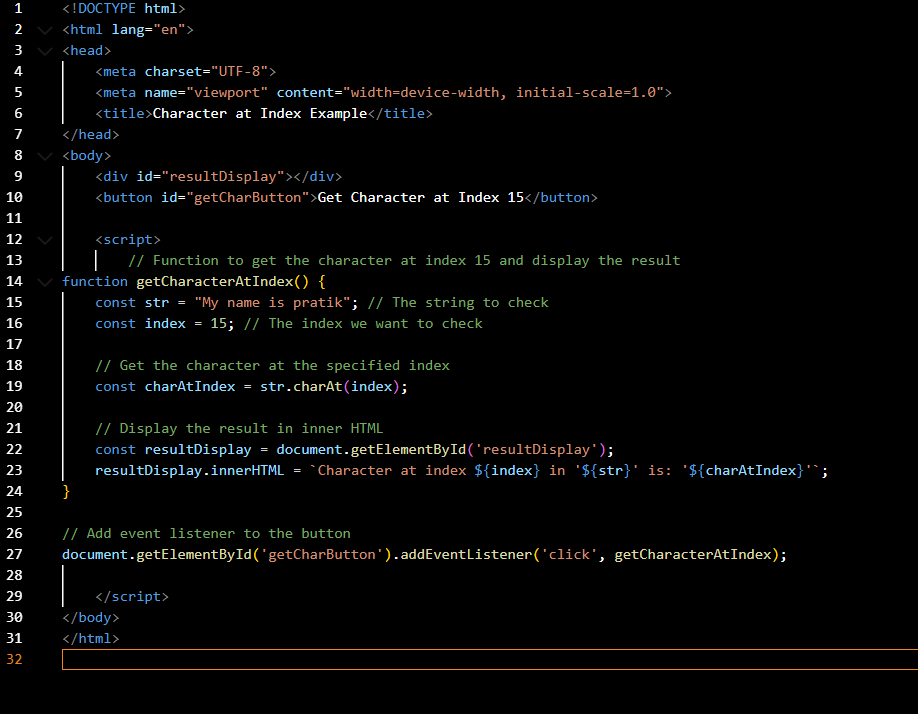
Q.64 Change all the string characters to lowercase letters using toLowerCase() Method.

Ans :

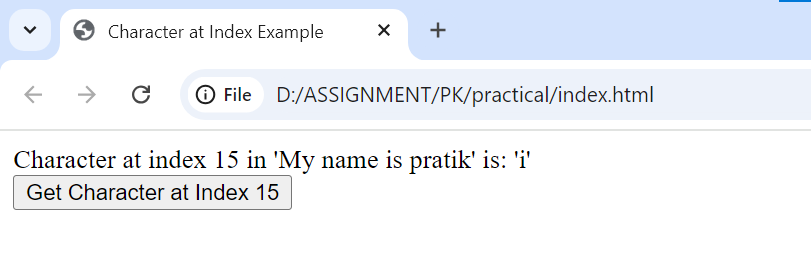


Ans - 

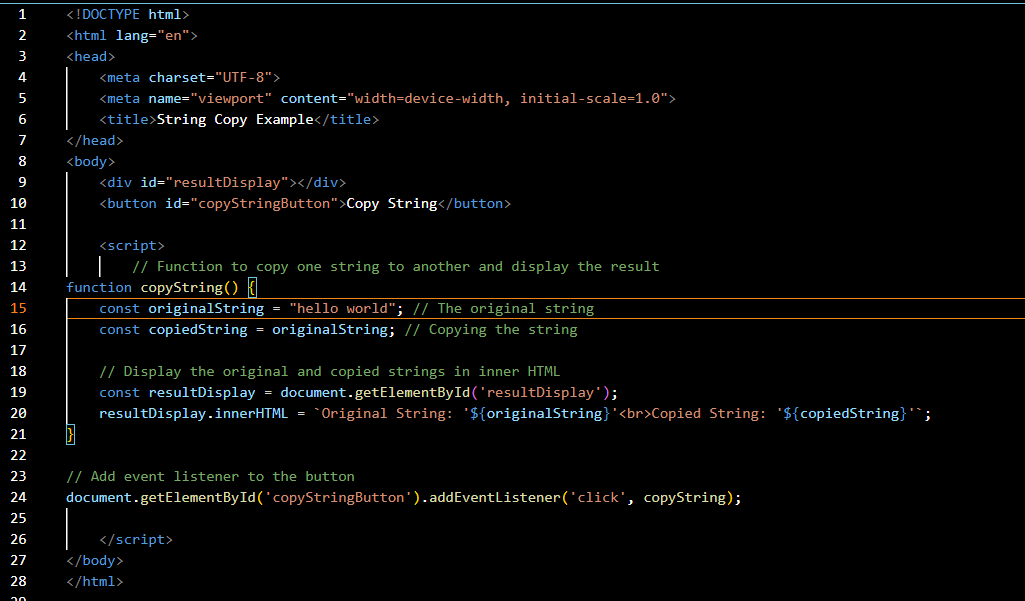
Q.65 What is Character at index 15 in ’30 Days of JavaScript’ string? Use charAt() method.



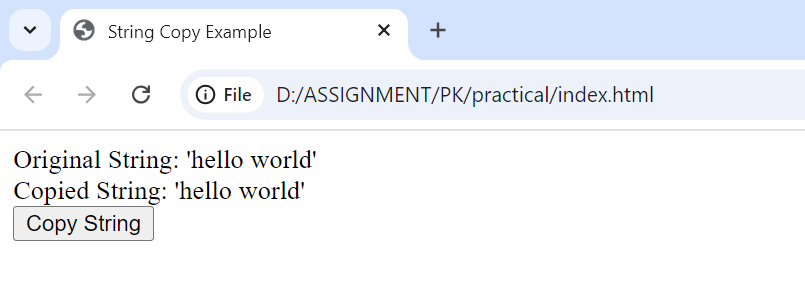
Output :

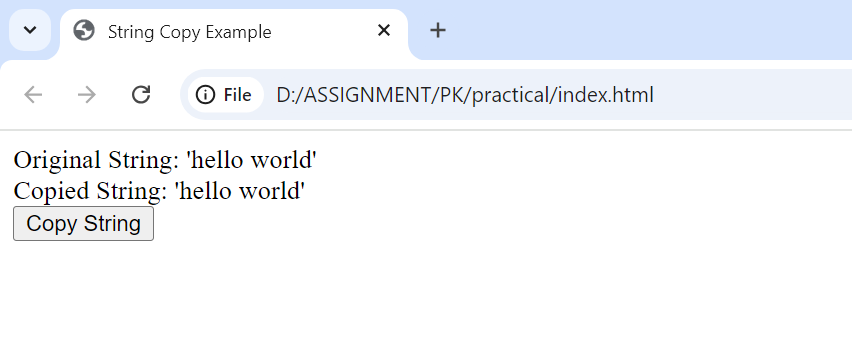


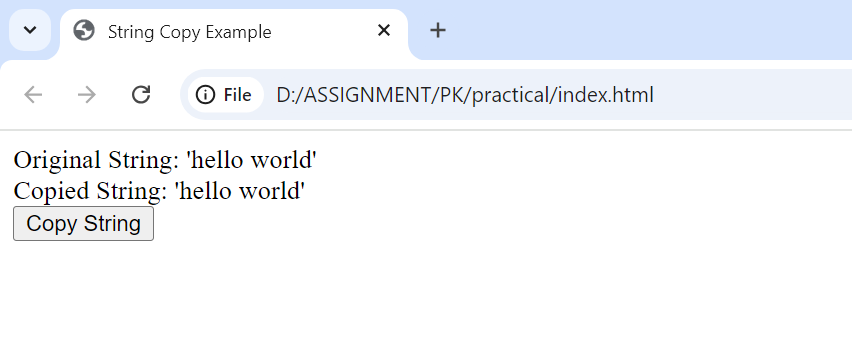
Q.66 copy to one string to another string in JS?

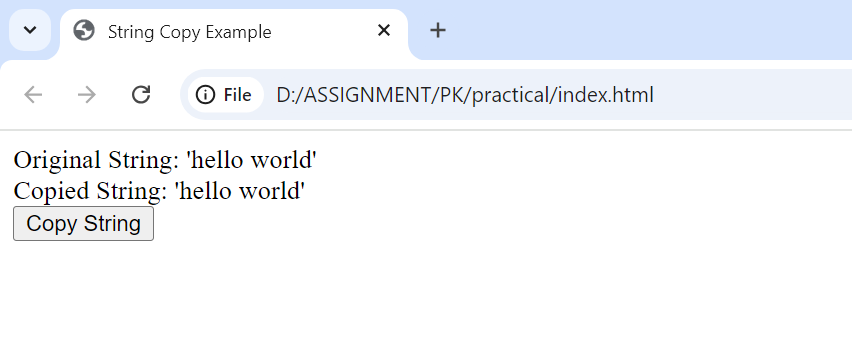
Ans - 

Output :

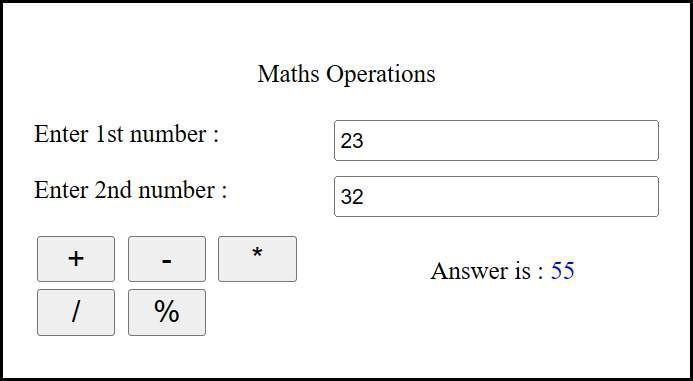








● Create basic math operation in JS





• Create result

)1arkshcct for Informatio n Technology

:Enter Marks

**I. C: T.angul!ge**

**2. C:++ rJlngllngc**

3. Database

4. IIT.t\<IL

5. css

u.pbp



34

44

22

4

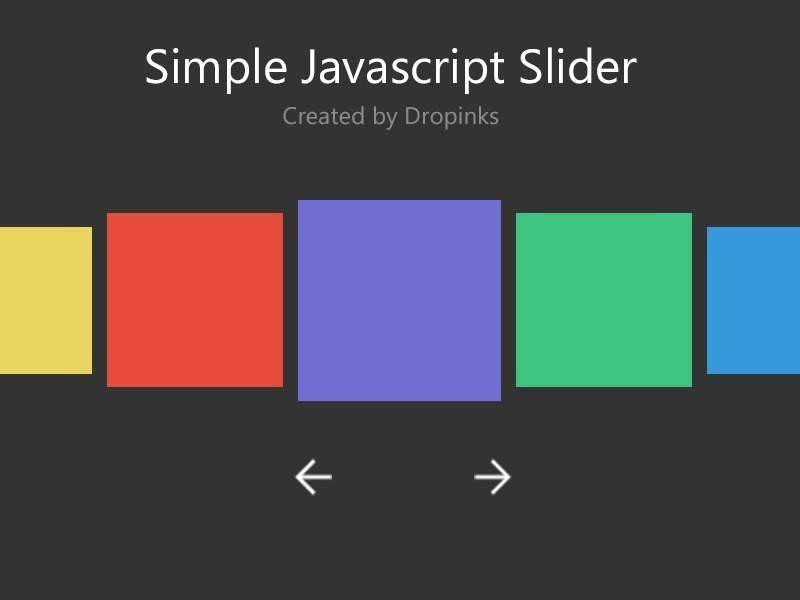
5()

Ro,r.;ull

To:al is : 272 / 350 **P rcentage** is. : 7:) ..



• Create a slider using JavaScript



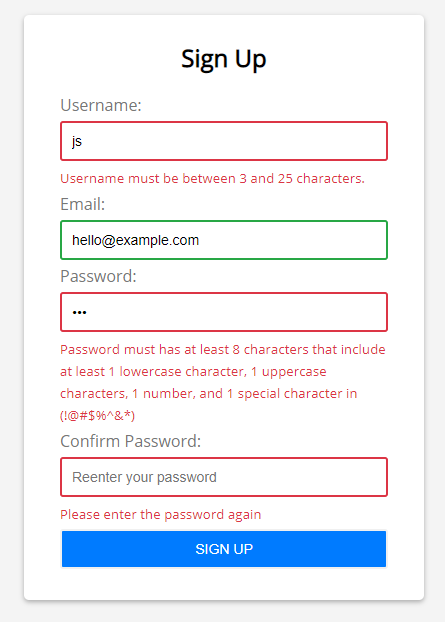
● Change background using java script



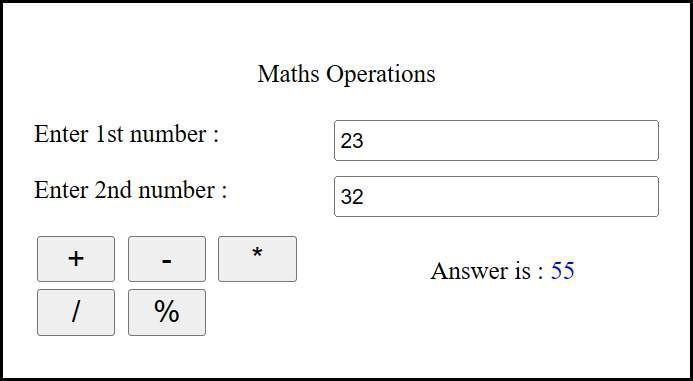
● Create a hide show password functionalities.



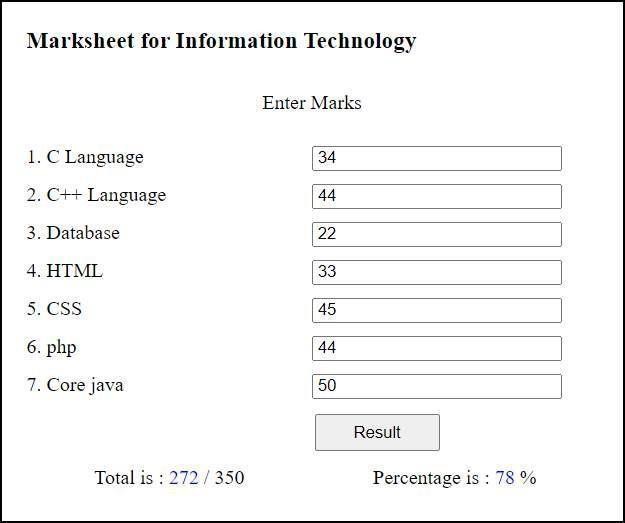
● Create a form and apply validations as shown in below example.



● Create basic math operation in JS

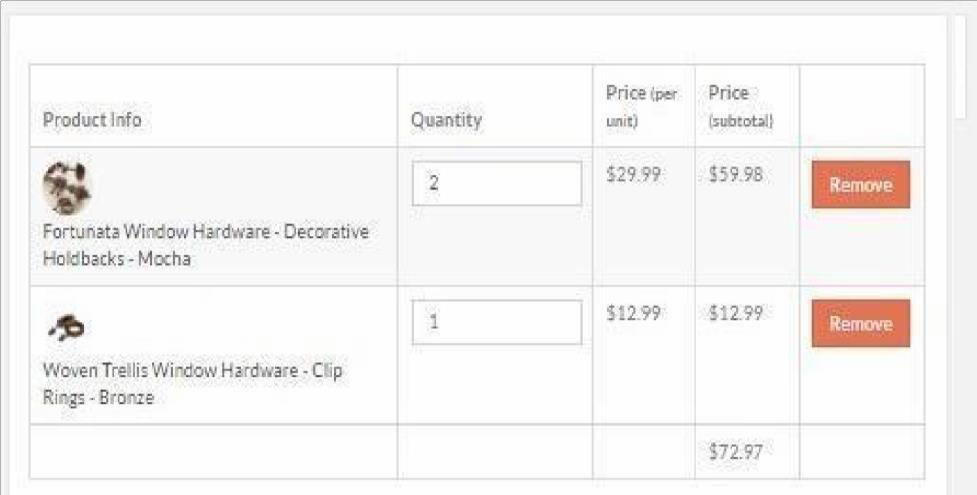


● Create result

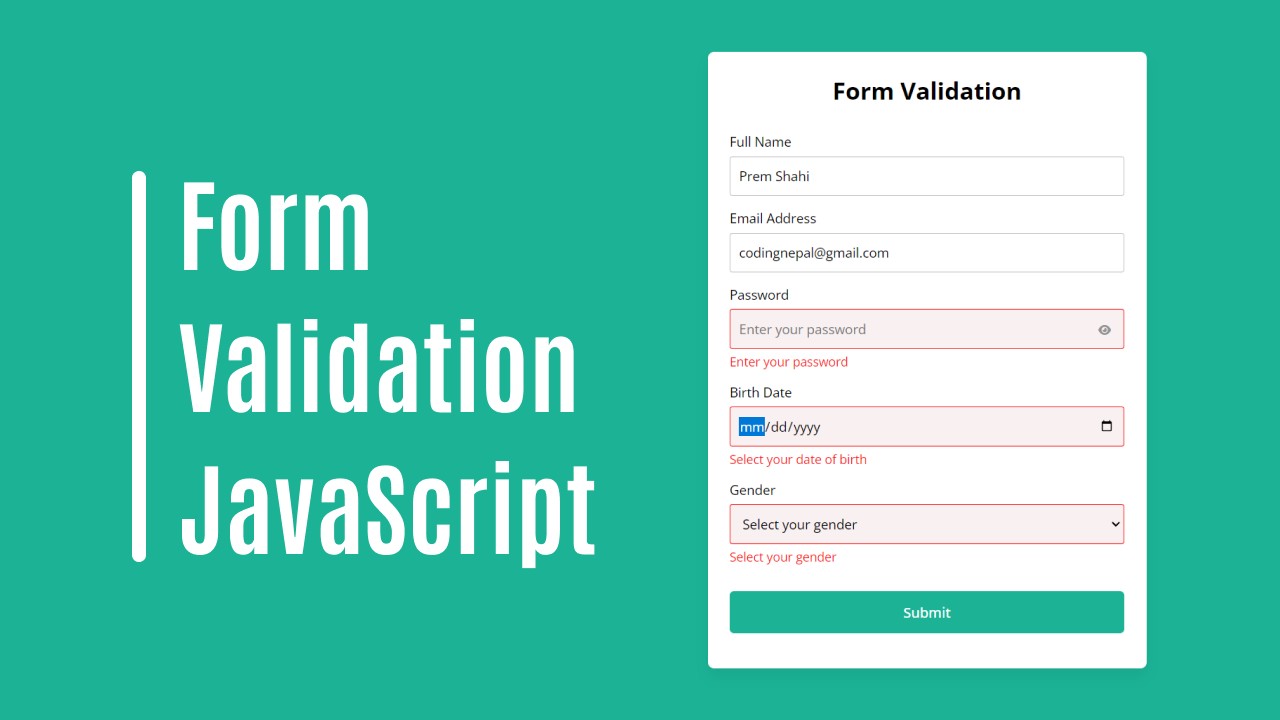


**Advance JavaScript**

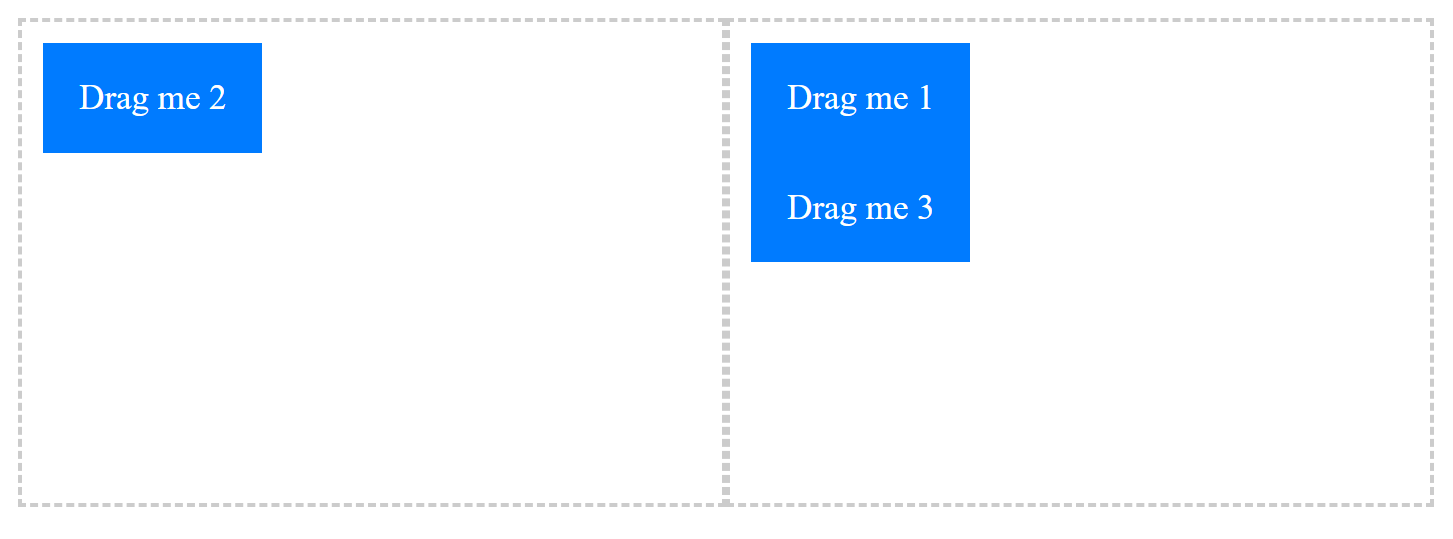
• Calculate subtotal price of quantity show in below image



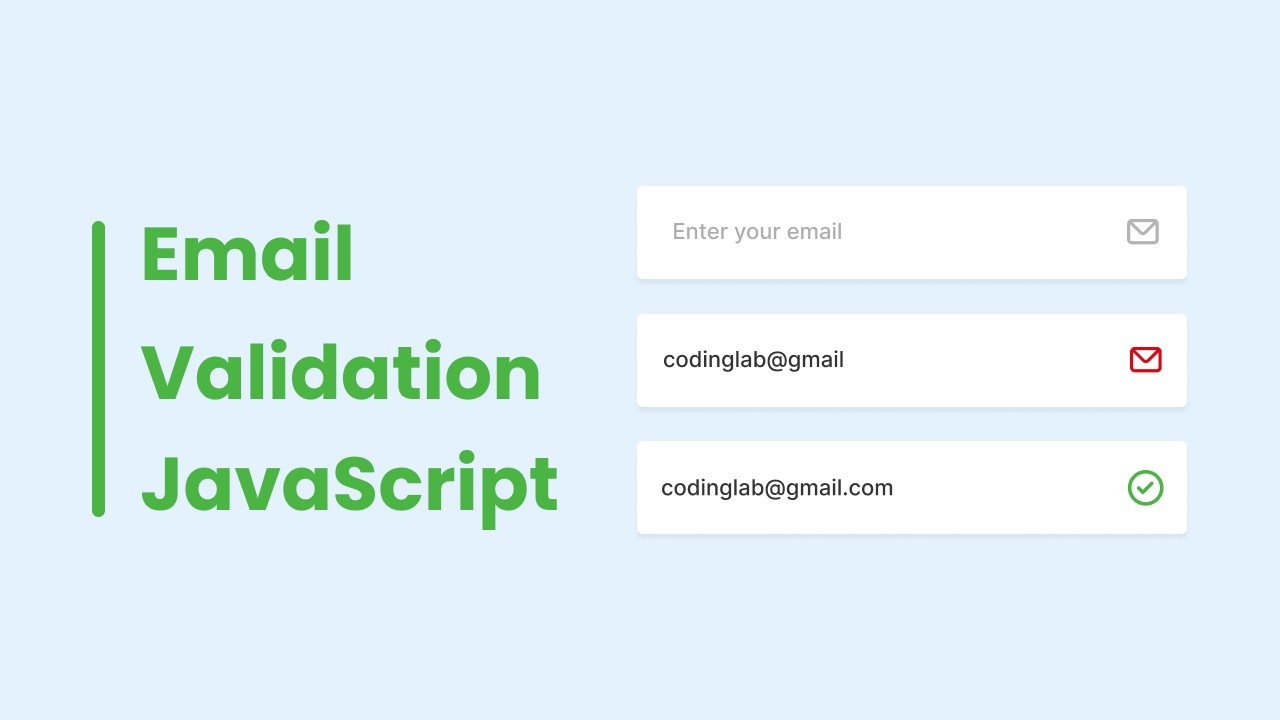
• Create below form and do apply validation show in below image



• Create two containers and drag from one to another show in below image



• Createt Email validation program show in below image



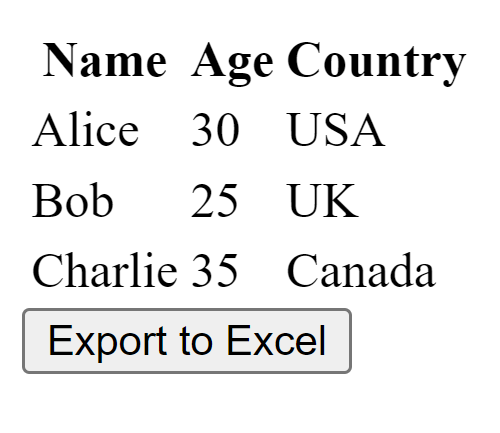
• Create Password Strength checker show in below image



• Create program to generate Random quote show in below image



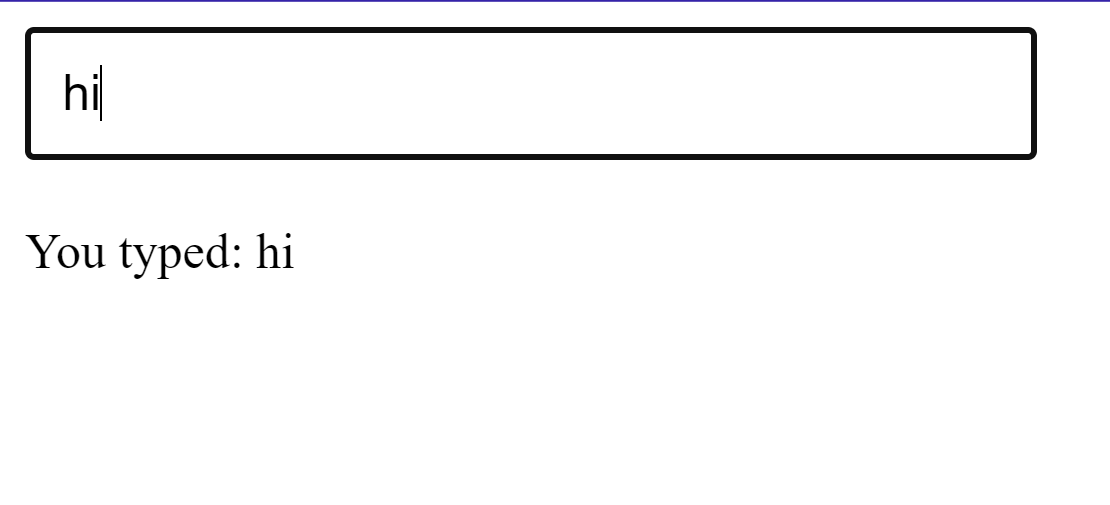
• Create program to Export html table to excel show in below image



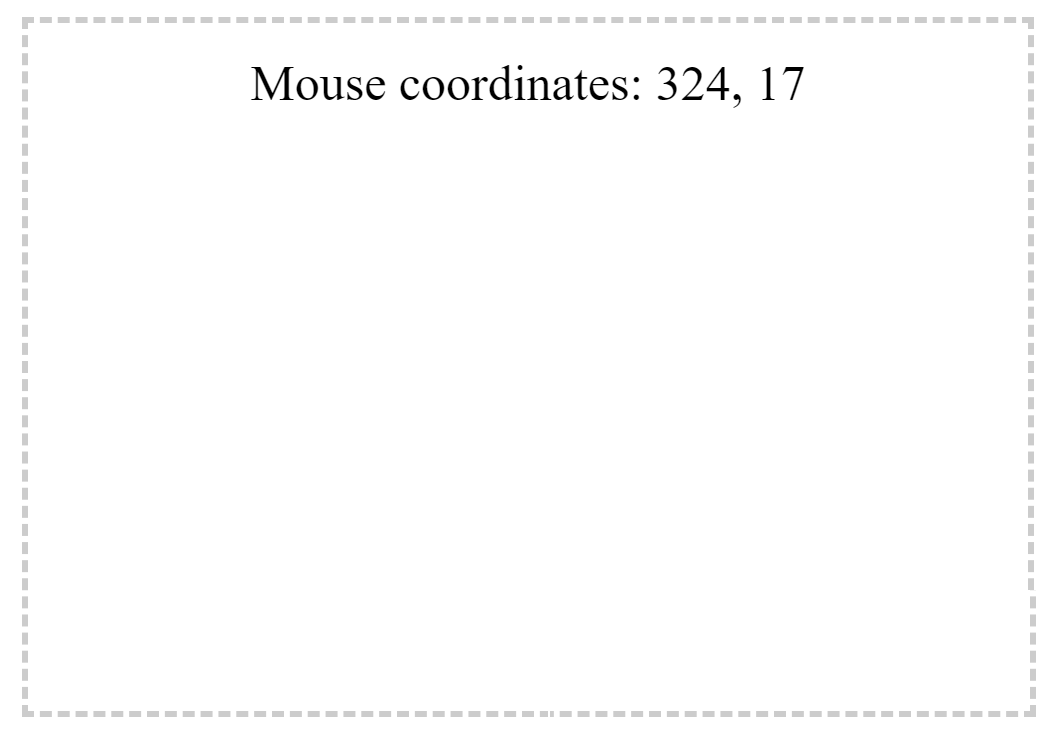
• Create Currency converter show in below image



• Delay function result using debouncing



• Achieve Throttling using below example (on mouse over display mouse cordinates.)



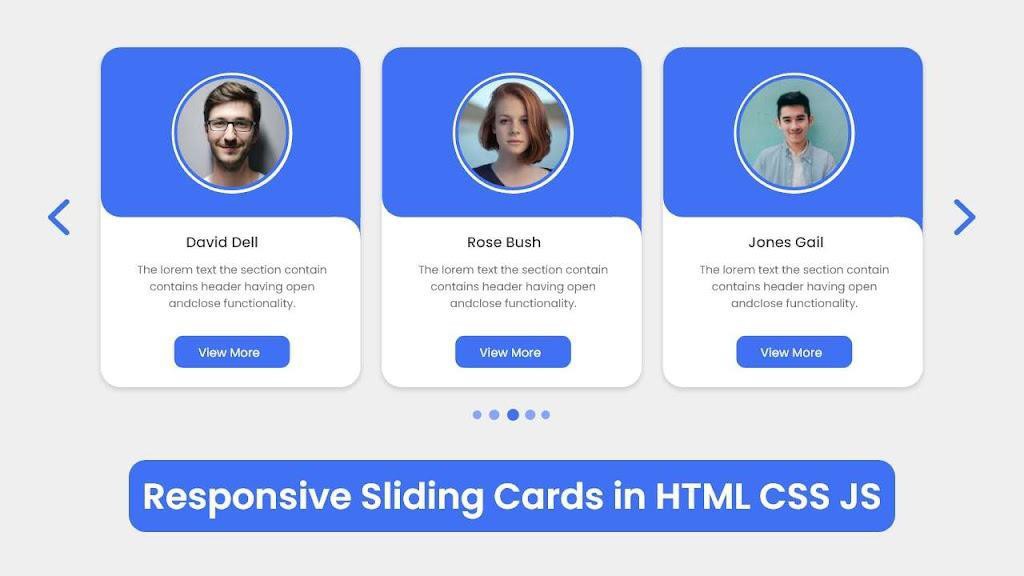
• Create Simple Alarm Clock show in below image



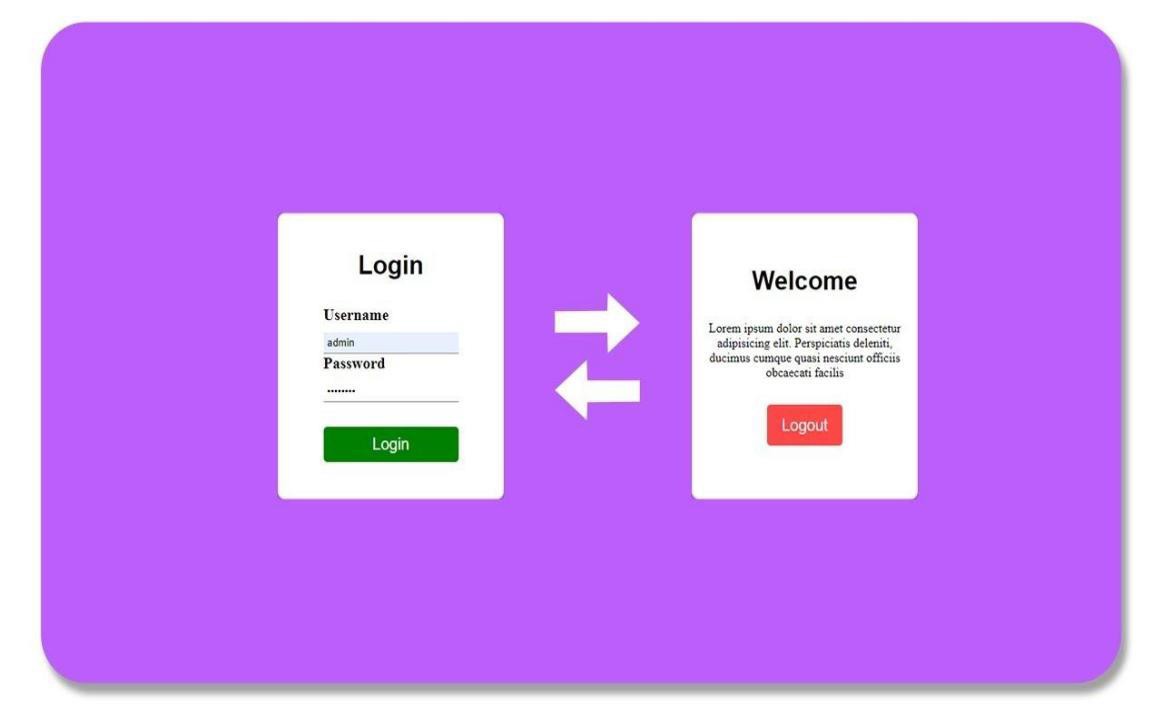
• Create Responsive Light box Gallery shown in below image



• Create review Card slider shown in below image



• Create Login Logout system shown below



• Create Image Editor shown below and apply filters like brightness,contrast etc.



• Perform Localstorage CRUD as below

