Dr. D. Y. Patil Institute of Engineering, Management & Research, Akurdi , Pune – 411044.

Department of Electronics and Telecommunication Engineering

JavaScript (Lab Practice – 2)

LAB MANUAL

Code: 404187

B.E. E&TC

Academic Year 2023-24

Name	
Roll no	
Subject	

INDEX

Experiment No: 01

Aim:- Write a JavaScript program to calculate area of triangle, area of rectangle and area of circle.

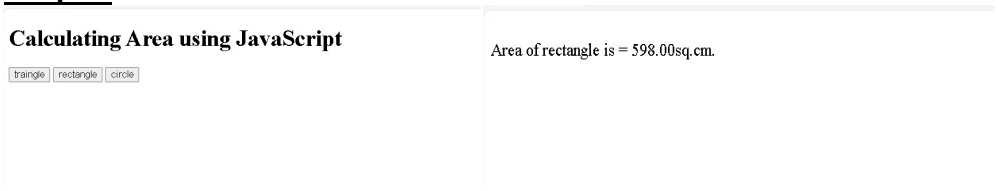
Algorithm :

Program Code:

```
function calculateTriangleArea(base, height) {
  return 0.5 * base * height;
}

function calculateRectangleArea(length, width) {
  return length * width;
}

function calculateCircleArea(radius) {
  return Math.PI * radius * radius;
}

function main() {
  console.log("Area Calculator");
  console.log("1. Triangle");
  console.log("2. Rectangle");
  console.log("3. Circle");

  const choice = prompt("Enter the number of the shape for which you want to calculate the area: ");

  switch (choice) {
    case "1":
      const triangleBase = parseFloat(prompt("Enter the base of the triangle: "));
      const triangleHeight = parseFloat(prompt("Enter the height of the triangle: "));
```

```
    const triangleArea = calculateTriangleArea(triangleBase, triangleHeight);
    console.log(`The area of the triangle is: ${triangleArea}`);
    break;

  case "2":
    const rectangleLength = parseFloat(prompt("Enter the length of the rectangle: "));
    const rectangleWidth = parseFloat(prompt("Enter the width of the rectangle: "));
    const rectangleArea = calculateRectangleArea(rectangleLength, rectangleWidth);
    console.log(`The area of the rectangle is: ${rectangleArea}`);
    break;

  case "3":
    const circleRadius = parseFloat(prompt("Enter the radius of the circle: "));
    const circleArea = calculateCircleArea(circleRadius);
    console.log(`The area of the circle is: ${circleArea}`);
    break;

  default:
    console.log("Invalid choice. Please enter 1, 2, or 3.");
 }
}

main();
```

Output:



Conclusion:

Experiment No: 02

Aim:- Write a JavaScript program to generate the multiplication table of a given number

Algorithm :

Program Code:

```
function generateMultiplicationTable(number, length) {
  console.log(`Multiplication Table for ${number}:\n`);
  for (let i = 1; i <= length; i++) {
    const result = number * i;
    console.log(`${number} * ${i} = ${result}`);
  }
}

const number = parseInt(prompt("Enter a number to generate its multiplication table: "));
const tableLength = parseInt(prompt("Enter the number of terms in the table: "));

generateMultiplicationTable(number, tableLength);
```

Output:

Multiplication Table

Enter Number | Generate Multiplication Table

3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30

Conclusion:

Experiment No: 03

Aim:- Write a JavaScript program to following operations on a given string,

- Reverse string
- Replace characters of a string.
- String is Palindrome

Theory:

reverse() Method:

JavaScript 1.1+ and JScript 2.0+ also allow you to reverse the elements of the array in place. The reverse() method, as one might expect, reverses the elements of the array it is invoked on:

```
var myArray = ["red", "green", "blue"];

myArray.reverse();

alert(myArray);
```

replace()

The replace() method returns the string that results when you replace text matching its first argument (a regular expression) with the text of the second argument (a string). If the g (global) flag is not set in the regular expression declaration, this method replaces only the first occurrence of the pattern. For example,

```
var s = "Hello. Regexps are fun.";
s = s.replace(/\./, "!"); // replace first period with an exclamation pointalert(s);
```

produces the string ―Hello! Regexps are fun.‖ Including the g flag will cause the interpreter to perform a global replace, finding and replacing every matching substring. For example,

```
var s = "Hello. Regexps are fun.";s = s.replace(/\./g, "!"); // replace all periods with exclamation pointsalert(s);
```

yields this result: ―Hello! Regexps are fun!‖

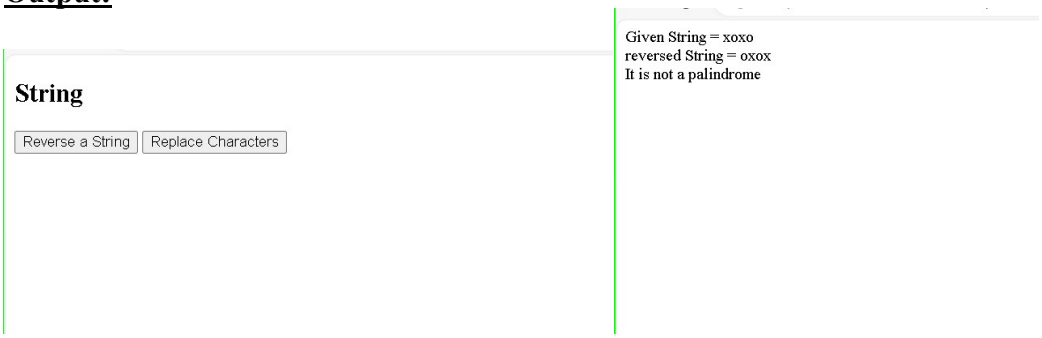
Algorithm :

Program Code:

```
<!DOCTYPE html>
<html>
<body>

<h2>String</h2>

<button onclick="RevStr()">Reverse a String</button>
<button onclick="replace()">Replace Characters</button>

<p id = "test"></p>
<p id="demo"></p>

<script>
function RevStr(){

   // empty string
   let revString = "";
   var str = prompt("Enter String");
   for (let i = str.length - 1; i >= 0; i--) {
      revString += str[i];
   }
  console.log("Given String  = " + str + "<BR>" + "reversed String = " + revString);
   document.write("Given String  = " + str  +"<br> "+ "reversed String = " + revString);
  PalStr(str, revString);
}

function PalStr(str, revString){

   // find the length of a string

      if (str === revString) {
         document.write( "<br>"+'It is a palindrome');
       }
      else {
        document.write( "<br>"+'It is not a palindrome');
       }
}

function replace(){
   var originaltext=prompt("enter original text ");
   var chartoreplace=prompt("enter char to be replaced in the original text");
   var newchar=prompt("enter new char to be inserted")
```

```
    var newstring=originaltext.replace(chartoreplace, newchar);
    document.write(newstring);
}
</script>

</body>
</html>
```

Output:



Conclusion:

Experiment No: 04

Aim:- Write a JavaScript program to compare two strings using various methods.

Theory:

Comparing Strings

While it is clear what comparison operators mean for numbers, what about strings? For example, is the following expression true? "thomas" >> "fritz"

When you compare strings, JavaScript evaluates the comparison based on strings' lexicographic order. Lexicographic order is essentially alphabetic order, with a few extra rules thrown in to deal with upper- and lower-case characters as well as to accommodate strings of different lengths.

The following general rules apply:

- Lowercase characters are less than uppercase characters.
- Shorter strings are less than longer strings.
- Letters occurring earlier in the alphabet are less than those occurring later.
- Characters with lower ASCII or Unicode values are less than those with larger values.

The interpreter examines strings on a character-by-character basis. As soon as one of the previous rules applies to the strings in question (for example, the two characters are different), the expression is evaluated accordingly.

The following comparisons are all true: "b" >> "a"

"thomas" >> "fritz"

"aaaa" >> "a"

"abC" >> "abc"

While this ordering might seem confusing at first blush, it is quite standard and consistent across most programming languages.

Algorithm :

Program Code:

```
<!DOCTYPE html>
<html>
<head>
   <title>exp4</title>
</head>
<body>
   <button onclick="exp4()">Compare</button>

   <script>
      function exp4() {
         let str1 = "apple";
         let str2 = "banana";

         if (str1 < str2) {
            alert("str1 is less than str2.");
         } else if (str1 > str2) {
            alert("str1 is greater than str2.");
         } else {
            alert("The strings are equal.");
         }
      }
   </script>
</body>
</html>
```

Output:

Conclusion:

Experiment No: 05

Aim:- Write a JavaScript program that will create a countdown timer

Algorithm :

Program Code:

```
function calculateCountdown(targetDate) {
  const currentDate = new Date();
  const timeDifference = targetDate - currentDate;

  if (timeDifference <= 0) {
    return "Countdown expired!";
  }

  const days = Math.floor(timeDifference / (1000 * 60 * 60 * 24));
  const hours = Math.floor((timeDifference % (1000 * 60 * 60 * 24)) / (1000 * 60 * 60));
  const minutes = Math.floor((timeDifference % (1000 * 60 * 60)) / (1000 * 60));
  const seconds = Math.floor((timeDifference % (1000 * 60)) / 1000);

  return `${days}d ${hours}h ${minutes}m ${seconds}s`;
}
```

```
const targetDate = new Date("2023-12-31T23:59:59"); // Change this to your target date
and time

function updateCountdown() {
  const countdown = calculateCountdown(targetDate);
  document.getElementById("countdown").textContent = countdown;

  if (countdown === "Countdown expired!") {
    clearInterval(countdownInterval);
  }
}

updateCountdown(); // Initial calculation
const countdownInterval = setInterval(updateCountdown, 1000); // Update every second
```

Output:

Building Coundown Timer Using Date Function

81d 17h 31m 13s

Todays Date is Sun Oct 15 2023 22:06:08 GMT+0530 (India Standard Time)
Month is 9
Day is 0
Time is 1697387768585msec
Time in msec from 1 Jan 1970 till Jan 5, 2024 is 1704449245000
Difference 7061476415

Conclusion:

Experiment No: 06

Aim:- Write a JavaScript program that will create an array and perform following operations

- To remove specific element from the array.
- Check if an array contains a specified value.
- To empty an array

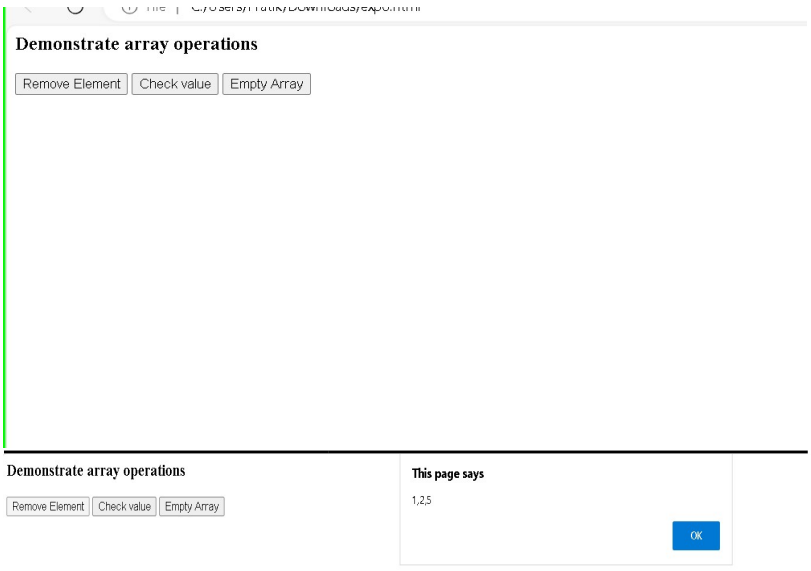
Algorithm :

Program Code:

```
<html>
<body>
<h3>Demonstrate array operations</h3>

<button onclick="removeelement(myarr, 2)">Remove Element</button>
<button onclick=" containsvalue(myarr,2)">Check value</button>
<button onclick="emptyArray(myarr)">Empty Array</button>

<script>
let myarr=[1,2,3,4,5];

function removeelement(myarr, index){
   if(myarr[index]!=null){
      myarr.splice(index, 1);

   }
   alert(myarr);
}
```

```
function containsvalue(myarr, value){
   for(var i=0;i<myarr.length;i++){
     if(myarr[i]==value){
      alert("value found at index"+ i);
      return;
     }

   }
   alert("not found");
}

function emptyArray(arr) {
   myarr.length = 0;
   alert("array is empty"+myarr );
}
</script>
</body>
</html>
```

Output:



Conclusion:

Experiment No: 07

Aim:- Write a JavaScript program that will append an object to an array and will check if an object is an array

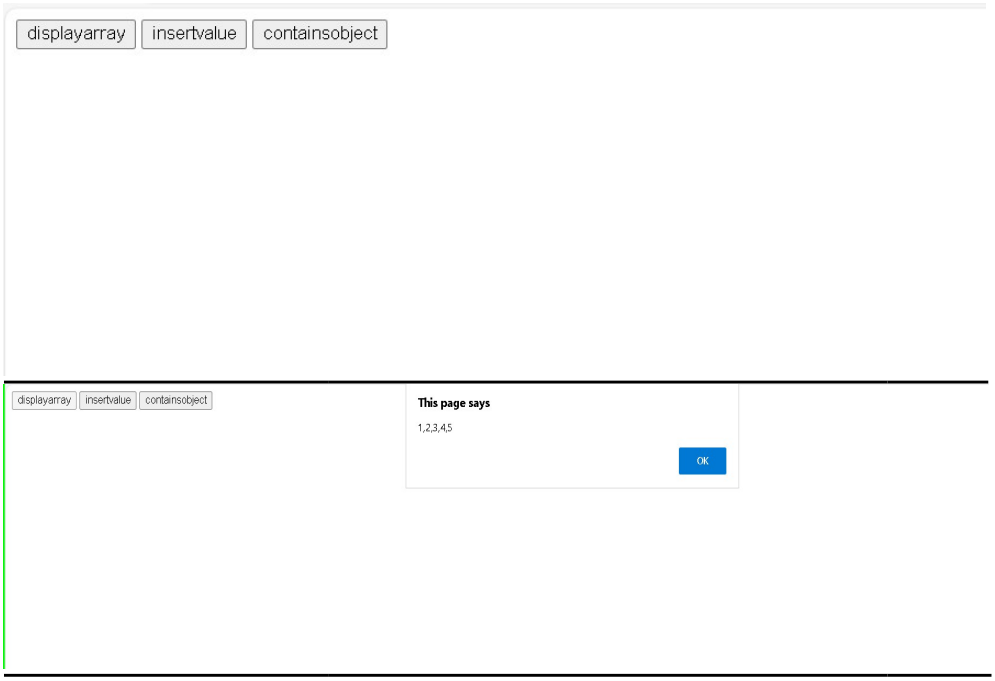
Algorithm :

Program Code:

```
<!DOCTYPE html>

<html>
   <head>
      <meta charset="utf-8">
      <meta http-equiv="X-UA-Compatible" content="IE=edge">
      <title></title>
      <meta name="description" content="">
      <meta name="viewport" content="width=device-width, initial-scale=1">
      <link rel="stylesheet" href="">
   </head>
   <body>
      <button onclick="display()">displayarray</button>
      <button onclick="appendObjectToArray(arr, 2)">insertvalue</button>
      <button onclick="isObjectArray(2)">containsobject</button>

      <script>
         let arr=[1,2,3,4,5];
```

```
        function display(){
           alert(arr);
        }


        function appendObjectToArray(arr, obj) {
   arr.push(obj);
   alert(arr);
}

let obj = { key: 'value' };

function isObjectArray(obj) {
 for(var i=0;i<arr.length;i++){
   if(arr[i]==obj){
      alert("true");
      return;
   }
 }
 alert("false");

}

      </script>
   </body>
</html>
```

Output:


displayarray
insertvalue
containsobject
displayarray
insertvalue
containsobject
This page says
1,2,3,4,5
OK

Conclusion:

Experiment No: 08

Aim:- Write a JavaScript program to create a Home page of any website and change background color using

- On mouse over event
- On focus event

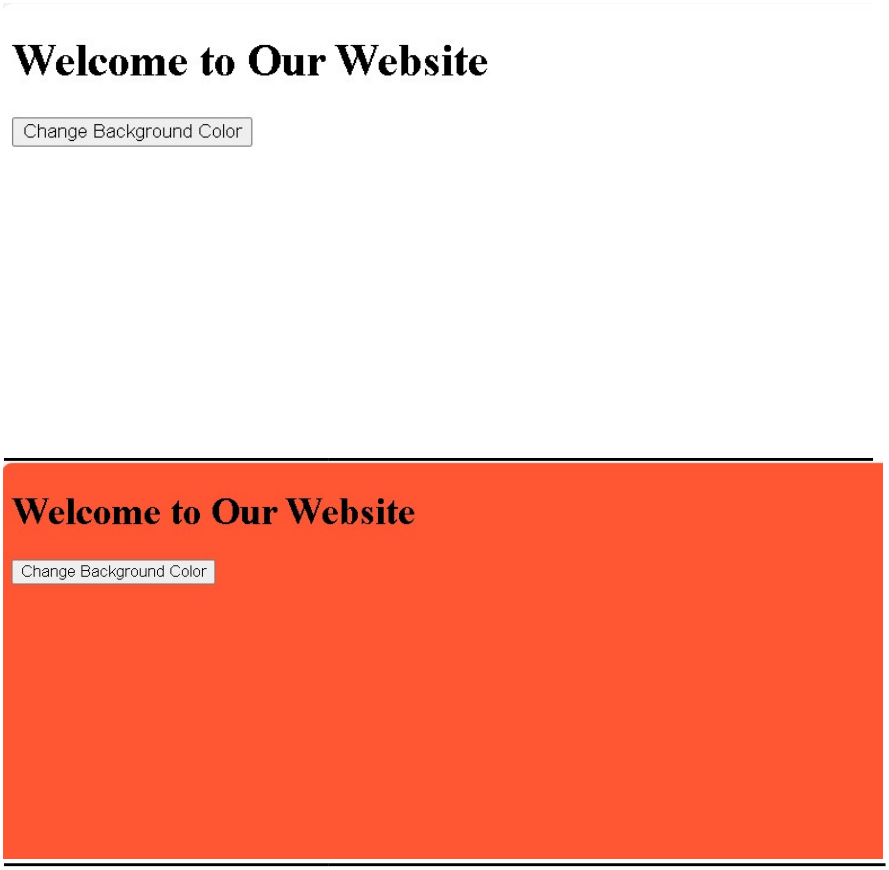
Algorithm :

Program Code:

```
<!DOCTYPE html>
<html>
<head>
   <title>Website Home Page</title>
   <style>
      body {
         font-family: Arial, sans-serif;
         text-align: center;
         padding: 50px;
         background-color: #f0f0f0;
         transition: background-color 0.3s;
      }

      h1 {
         color: #333;
      }

      button {
```

```
            padding: 10px 20px;
            font-size: 16px;
            background-color: #0074d9;
            color: #fff;
            border: none;
            cursor: pointer;
        }

        button:focus {
            outline: none;
        }
    </style>
</head>
<body>
    <h1>Welcome to Our Website</h1>
    <p>This is the home page of our website. Hover over or focus on the button to change
the background color.</p>

    <button onmouseover="changeBackgroundColor()"
onfocus="changeBackgroundColor()">Hover or Focus Me</button>

    <script>
        function changeBackgroundColor() {
            const colors = ['#f0f0f0', '#ff5733', '#33ff57', '#5733ff'];
            const body = document.body;
            const currentColor = body.style.backgroundColor;
            const newColor = colors[Math.floor(Math.random() * colors.length)];

            while (newColor === currentColor) {
                // Ensure the new color is different from the current color
                newColor = colors[Math.floor(Math.random() * colors.length)];
            }

            body.style.backgroundColor = newColor;
        }
    </script>
</body>
</html>
```

Output:



Conclusion:

Experiment No: 09

Aim:-

Design and implement a simple calculator using Java script for operations like addition multiplication, subtraction, division, square of a number etc:

- Design a calculator like text field for input and output, buttons for numbers and operations etc.
- Validate input values
- Prompt / Alerts for invalid values etc.

Algorithm :

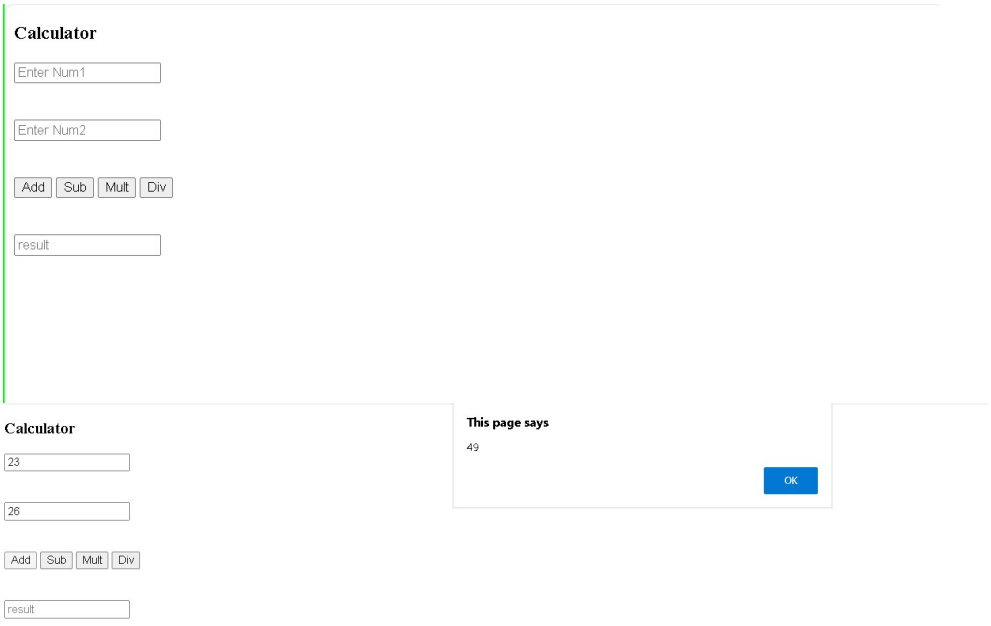
Program Code:

```
<!DOCTYPE html>
<html>
<head>
   <title>Simple Calculator</title>
   <style>
      .calculator {
         width: 300px;
         margin: 0 auto;
         text-align: center;
      }
   </style>
</head>
<body>
   <div class="calculator">
      <h2>Simple Calculator</h2>
```

```
    <input type="text" id="display" readonly>
    <br>
    <button onclick="appendToDisplay('1')">1</button>
    <button onclick="appendToDisplay('2')">2</button>
    <button onclick="appendToDisplay('3')">3</button>
    <button onclick="appendToDisplay('+')">+</button>
    <br>
    <button onclick="appendToDisplay('4')">4</button>
    <button onclick="appendToDisplay('5')">5</button>
    <button onclick="appendToDisplay('6')">6</button>
    <button onclick="appendToDisplay('-')">-</button>
    <br>
    <button onclick="appendToDisplay('7')">7</button>
    <button onclick="appendToDisplay('8')">8</button>
    <button onclick="appendToDisplay('9')">9</button>
    <button onclick="appendToDisplay('*')">*</button>
    <br>
    <button onclick="appendToDisplay('0')">0</button>
    <button onclick="appendToDisplay('.')">.</button>
    <button onclick="clearDisplay()">C</button>
    <button onclick="appendToDisplay('/')">/</button>
    <br>
    <button onclick="calculateResult()">=</button>
    <button onclick="calculateSquare()">x²</button>
</div>

<script>
    const display = document.getElementById("display");

    function appendToDisplay(value) {
        display.value += value;
    }

    function clearDisplay() {
        display.value = "";
    }

    function calculateResult() {
        try {
            const result = eval(display.value);
            display.value = result;
        } catch (error) {
            alert("Invalid input");
            clearDisplay();
        }
```

```
    }

    function calculateSquare() {
      try {
        const inputValue = parseFloat(display.value);
        if (!isNaN(inputValue)) {
          display.value = inputValue * inputValue;
        } else {
          alert("Invalid input");
          clearDisplay();
        }
      } catch (error) {
        alert("Invalid input");
        clearDisplay();
      }
    }
  </script>
</body>
</html>
```

Output:



Conclusion: