

JavaScript practical Assignment

Q. Set up a simple HTML file and include a JavaScript script. Display an alert box with a welcome message when the page loads.

```
<!DOCTYPE html>
<html>
<head>
    <title>Welcome Page</title>
</head>
<body>
    <h1>Hello Abhay</h1>
    <p>This is a simple webpage with a welcome alert.</p>
</body>
<script>
    window.onload = function() {
        alert("Welcome to my website!");
    };
</script>
</html>
```

Q . Create a JavaScript program that declares variables of different data types, performs basic arithmetic operations, and outputs the results to the console.

```
<!DOCTYPE html>
<html>
<head>
    <title>JavaScript Basics</title>
<script>
    // Variable declaration
    let name = "Abhay";           // String
    let age = 21;                 // Number
    let isStudent = true;         // Boolean

    // Arithmetic operations
    let num1 = 10;
    let num2 = 5;

    let sum = num1 + num2;
    let difference = num1 - num2;
    let product = num1 * num2;
```

```

let quotient = num1 / num2;

console.log("Name:", name);
console.log("Age:", age);
console.log("Is student:", isStudent);

console.log("Sum:", sum);
console.log("Difference:", difference);
console.log("Product:", product);
console.log("Quotient:", quotient);
</script>
</head>
<body>
    <h2>Open Console to See the Output</h2>
</body>
</html>

```

Q . Write a program that takes a student's score as input and outputs the corresponding grade based on the following criteria:

- **A: 90-100**
- **B: 80-89**
- **C: 70-79**
- **D: 60-69**
- **F: Below 60**

```

<!DOCTYPE html>
<html>
    <head>
        <title>Student Grade Checker</title>
    </head>
    <body>
        <h2>Grade will be shown in alert box</h2>
    </body>
    <script>
        function checkGrade() {
            let score = parseInt(prompt("Enter student's score:"));
            let grade;

            if (score >= 90 && score <= 100) {

```

```

        grade = "A";
    } else if (score >= 80 && score <= 89) {
        grade = "B";
    } else if (score >= 70 && score <= 79) {
        grade = "C";
    } else if (score >= 60 && score <= 59) {
        grade = "D";
    } else {
        grade = "F";
    }

    alert("Grade: " + grade);
}

// Call function on page Load
window.onLoad = checkGrade;
</script>
</html>

```

Q. : Create a program that classifies a person based on their age group: •
Child: 0-12 • Teen: 13-19 • Adult: 20-64 • Senior: 65 and above

```

<!DOCTYPE html>
<html>
  <head>
    <title>Age Group Classifier</title>
  </head>
  <body>
    <h2>Your age group will be shown in an alert box</h2>
  </body>
  <script>
    function classifyAge() {
      let age = parseInt(prompt("Enter your age:"));
      let group;

      if (age >= 0 && age <= 12) {
        group = "Child";
      } else if (age >= 13 && age <= 19) {
        group = "Teen";
      } else if (age >= 20 && age <= 64) {
        group = "Adult";
      } else if (age >= 65) {
        group = "Senior";
      }

      alert("Your age group is: " + group);
    }
  </script>

```

```

    } else {
        group = "Invalid age";
    }

    alert("You are classified as: " + group);
}

// Call the function when the page Loads
window.onLoad = classifyAge;
</script>
</html>

```

Q . Write a program that checks whether a given number is even or odd.

```

<!DOCTYPE html>
<html>
<head>
    <title>Even or Odd Checker</title>

</head>
<body>
    <h2>Even or Odd result </h2>
</body>
<script>
    function checkEvenOdd() {
        let num = parseInt(prompt("Enter a number:"));

        if (num % 2 === 0) {
            alert(num + " is Even");
        } else {
            alert(num + " is Odd");
        }
    }

    // Run the function when the page Loads
    window.onLoad = checkEvenOdd;
</script>
</html>

```

Q. Create a program that prompts the user for a number between 1 and 7 and outputs the corresponding day of the week.

```
<!DOCTYPE html>
```

```

<html>
<head>
    <title>Day Finder</title>

</head>
<body>
    <h2>Day will be shown in an alert box</h2>
</body>
<script>
    function showDay() {
        let num = parseInt(prompt("Enter a number (1 to 7):"));
        let day;

        switch(num) {
            case 1: day = "Sunday"; break;
            case 2: day = "Monday"; break;
            case 3: day = "Tuesday"; break;
            case 4: day = "Wednesday"; break;
            case 5: day = "Thursday"; break;
            case 6: day = "Friday"; break;
            case 7: day = "Saturday"; break;
            default: day = "Invalid input";
        }

        alert("Day: " + day);
    }

    // Call function on page Load
    window.onLoad = showDay;
</script>
</html>

```

Q. Develop a program that converts a temperature from Celsius to Fahrenheit or vice versa based on user input.

```

<!DOCTYPE html>
<html>
    <head>
        <title>Temperature Converter</title>
    </head>
    <body>
        <h2>Converted temperature will be shown in an alert box</h2>
    </body>
    <script>
        function convertTemperature() {

```

```

let type = prompt(
    "Enter conversion type: C to F or F to C"
).toUpperCase();
let temp = parseFloat(prompt("Enter temperature value:"));
let result;

if (type === "C TO F") {
    result = (temp * 9) / 5 + 32;
    alert(temp + "°C = " + result.toFixed(2) + "°F");
} else if (type === "F TO C") {
    result = ((temp - 32) * 5) / 9;
    alert(temp + "°F = " + result.toFixed(2) + "°C");
} else {
    alert("Invalid conversion type");
}
}

// Call the function when the page Loads
window.onLoad = convertTemperature;
</script>
</html>

```

Q. Create a simple calculator that performs basic arithmetic operations: addition, subtraction, multiplication, and division.

```

<!DOCTYPE html>
<html>
<head>
    <title>Simple Calculator</title>
</head>
<body>
    <h2>Calculator result will be shown in an alert box</h2>
</body>
<script>
    function simpleCalculator() {
        let num1 = parseFloat(prompt("Enter first number:"));
        let operator = prompt("Enter operator (+, -, *, /):");
        let num2 = parseFloat(prompt("Enter second number:"));
        let result;

        switch(operator) {
            case "+":
                result = num1 + num2;
                break;

```

```

        case "-":
            result = num1 - num2;
            break;
        case "*":
            result = num1 * num2;
            break;
        case "/":
            if (num2 === 0) {
                result = "Cannot divide by zero";
            } else {
                result = num1 / num2;
            }
            break;
        default:
            result = "Invalid operator";
    }

    alert("Result: " + result);
}

window.onload = simpleCalculator;
</script>
</html>

```

Q. Write a program that takes three numbers as input and determines the largest among them.

```

<script>
function findLargest() {
    let num1 = parseFloat(prompt("Enter first number:"));
    let num2 = parseFloat(prompt("Enter second number:"));
    let num3 = parseFloat(prompt("Enter third number:"));
    let largest;

    if (num1 >= num2 && num1 >= num3) {
        largest = num1;
    } else if (num2 >= num1 && num2 >= num3) {
        largest = num2;
    } else {
        largest = num3;
    }

    alert("The largest number is: " + largest);
}

```

```

        }

    window.onload = findLargest;
</script>
```

Q . Create a program that checks if a given year is a leap year

```

<!DOCTYPE html>
<html>
  <head>
    <title>Leap Year Checker</title>
  </head>
  <body>
    <h2>Leap year result will be shown in an alert box</h2>
  </body>
  <script>
    function checkLeapYear() {
      let year = parseInt(prompt("Enter a year:"));
      let message;

      if ((year % 4 === 0 && year % 100 !== 0) || year % 400 === 0) {
        message = year + " is a Leap Year";
      } else {
        message = year + " is NOT a Leap Year";
      }

      alert(message);
    }

    window.onload = checkLeapYear;
  </script>
</html>
```

Q. Simulate a traffic light system where the user inputs the color of the traffic light and the program outputs the corresponding action.

```

<script>
  function trafficSignal() {
    let color = prompt("Enter traffic light color (Red, Yellow,
Green):").toLowerCase();
    let action;

    switch(color) {
```

```

        case "red":
            action = "Stop";
            break;
        case "yellow":
            action = "Caution";
            break;
        case "green":
            action = "Go";
            break;
        default:
            action = "Invalid color";
    }

    alert("Action: " + action);
}

window.onload = trafficSignal;
</script>

```

Q. Write a program that simulates flipping a coin. The program should output either "Heads" or "Tails."

```

<!DOCTYPE html>
<html>
<head>
<title>Coin Flip</title>
<script>
    function flipCoin() {
        let result = Math.floor(Math.random() * 2); // 0 or 1
        if (result === 0) {
            alert("Heads");
        } else {
            alert("Tails");
        }
    }

    window.onload = flipCoin;
</script>
</head>
<body>
    <h2>Result will be shown in an alert box (Heads or Tails)</h2>
</body>
</html>

```

Q. Create a program that calculates the Body Mass Index (BMI) and categorizes the result.

```
<!DOCTYPE html>
<html>
<head>
    <title>BMI Calculator</title>

</head>
<body>
    <h2>BMI result will be shown in an alert box</h2>
</body>
<script>
    function calculateBMI() {
        // Step 1: Take user input
        let weight = parseFloat(prompt("Enter your weight in kilograms:"));
        let height = parseFloat(prompt("Enter your height in meters:"));

        // Step 2: Calculate BMI
        let bmi = weight / (height * height);
        let category;

        // Step 3: Categorize using if...else if...else
        if (bmi < 18.5) {
            category = "Underweight";
        } else if (bmi >= 18.5 && bmi < 24.9) {
            category = "Normal weight";
        } else if (bmi >= 25 && bmi < 29.9) {
            category = "Overweight";
        } else {
            category = "Obesity";
        }

        // Step 4: Show result
        alert("Your BMI is " + bmi.toFixed(2) + " - " + category);
    }

    // Run the function on page Load
    window.onLoad = calculateBMI;
</script>
</html>
```

Q. Build a simple quiz application where the user answers a question, and the program checks if the answer is correct.

```
<!DOCTYPE html>
<html>
<head>
    <title>Simple Quiz</title>
    <script>
        function startQuiz() {
            let answer = prompt("What is the capital of France?");

            if (answer.toLowerCase() === "paris") {
                alert("Correct! 🎉");
            } else {
                alert("Incorrect! The correct answer is Paris.");
            }
        }

        window.onload = startQuiz;
    </script>
</head>
<body>
    <h2>Answer the quiz question in the alert box</h2>
</body>
</html>
```

Q. Create a program that uses a for loop to print the numbers from 1 to 100. Additionally, implement a while loop that does the same

```
<!DOCTYPE html>
<html>
<head>
    <title>Loop Example</title>
    <script>
        function printNumbers() {
            console.log("Using for loop:");
            for (let i = 1; i <= 100; i++) {
                console.log(i);
            }

            console.log("Using while loop:");
        }
    </script>
</head>
<body>
</body>
</html>
```

```

let j = 1;
while (j <= 100) {
    console.log(j);
    j++;
}
}

window.onLoad = printNumbers;
</script>
</head>
<body>
    <h2>Check the console for output from 1 to 100</h2>
</body>
</html>

```

Q . Write a program that calculates the sum of all integers from 1 to a user-specified number.

```

<!DOCTYPE html>
<html>
<head>
    <title>Sum Calculator</title>
    <script>
        function calculateSum() {
            let num = parseInt(prompt("Enter a positive integer:"));
            let sum = 0;

            for (let i = 1; i <= num; i++) {
                sum += i;
            }

            alert("The sum of numbers from 1 to " + num + " is: " + sum);
        }

        window.onLoad = calculateSum;
    </script>
</head>
<body>
    <h2>Sum will be shown in an alert box</h2>
</body>
</html>

```

Q. Create a program that calculates the factorial of a given number.

```

<!DOCTYPE html>
<html>
<head>
    <title>Factorial Calculator</title>

</head>
<body>
    <h2>Factorial will be shown in an alert box</h2>
</body>
<script>
    function calculateFactorial() {
        let num = parseInt(prompt("Enter a positive integer:"));
        let factorial = 1;

        for (let i = 1; i <= num; i++) {
            factorial *= i;
        }

        alert("The factorial of " + num + " is: " + factorial);
    }

    window.onload = calculateFactorial;
</script>
</html>

```

Q. Generate the Fibonacci sequence up to a specified number .

```

<!DOCTYPE html>
<html>
<head>
    <title>Fibonacci Sequence</title>
<script>
    function generateFibonacci() {
        let terms = parseInt(prompt("Enter number of Fibonacci terms:"));
        let n1 = 0, n2 = 1, nextTerm;

        console.log("Fibonacci Sequence:");

        for (let i = 1; i <= terms; i++) {
            console.log(n1);
            nextTerm = n1 + n2;
            n1 = n2;
            n2 = nextTerm;
        }
    }
</script>
</html>

```

```
    window.onLoad = generateFibonacci;
  </script>
</head>
<body>
  <h2>Fibonacci sequence will be shown in the browser console</h2>
</body>
</html>
```

Q. Write a program that displays the multiplication table for a given number.

```
<!DOCTYPE html>
<html>
<head>
  <title>Multiplication Table</title>
  <script>
    function showTable() {
      let num = parseInt(prompt("Enter a number to print its multiplication table:"));

      console.log("Multiplication Table of " + num + ":");
      for (let i = 1; i <= 10; i++) {
        console.log(num + " x " + i + " = " + (num * i));
      }
    }

    window.onLoad = showTable;
  </script>
</head>
<body>
  <h2>Multiplication table will be shown in the browser console</h2>
</body>
</html>
```

Q. Create a program that counts down from a specified number to zero.

```
<!DOCTYPE html>
<html>
<head>
  <title>Countdown Program</title>
  <script>
```

```

function startCountdown() {
    let num = parseInt(prompt("Enter a starting number for countdown:"));

    console.log("Countdown begins:");
    while (num >= 0) {
        console.log(num);
        num--;
    }

    console.log("Blast off!");
}

window.onload = startCountdown;
</script>
</head>
<body>
    <h2>Countdown will be shown in the browser console</h2>
</body>
</html>

```

Q. Write a program that reverses a given string using a loop.

```

<!DOCTYPE html>
<html>
<head>
    <title>Reverse String</title>
    <script>
        function reverseText() {
            let input = prompt("Enter a string:");
            let reversed = "";

            for (let i = input.length - 1; i >= 0; i--) {
                reversed += input[i];
            }

            alert("Reversed string: " + reversed);
        }

        window.onload = reverseText;
    </script>
</head>
<body>
    <h2>Reversed string will be shown in an alert box</h2>
</body>
</html>

```

Q. Create a program that checks and displays all prime numbers up to a user-specified limit.

```
<!DOCTYPE html>
<html>
<head>
    <title>Prime Numbers</title>
    <script>
        function showPrimes() {
            let limit = parseInt(prompt("Enter a positive number:"));

            console.log("Prime numbers up to " + limit + ":");

            for (let num = 2; num <= limit; num++) {
                let isPrime = true;

                for (let i = 2; i <= Math.sqrt(num); i++) {
                    if (num % i === 0) {
                        isPrime = false;
                        break;
                    }
                }

                if (isPrime) {
                    console.log(num);
                }
            }
        }

        window.onload = showPrimes;
    </script>
</head>
<body>
    <h2>Prime numbers will be shown in the browser console</h2>
</body>
</html>
```

Q . Generate a pyramid of numbers.

```
<!DOCTYPE html>
<html>
<head>
    <title>Number Pyramid</title>
    <script>
        function generatePyramid() {
            let height = parseInt(prompt("Enter the height of the pyramid:"));

            for (let row = 1; row <= height; row++) {
                let stars = '';
                let spaces = ' '.repeat(height - row);

                for (let col = 1; col <= row; col++) {
                    stars += '*';
                }

                console.log(spaces + stars);
            }
        }
    </script>
</head>
<body>
    <h2>Number Pyramid</h2>
    <button onclick="generatePyramid()">Generate Pyramid</button>
</body>
</html>
```

```

console.log("Number Pyramid:");

for (let i = 1; i <= height; i++) {
    let row = "";

    // Add spaces for alignment
    for (let s = 1; s <= height - i; s++) {
        row += " ";
    }

    // Add numbers
    for (let j = 1; j <= i; j++) {
        row += j + " ";
    }

    console.log(row);
}
}

window.onLoad = generatePyramid;
</script>
</head>
<body>
    <h2>Number pyramid will be shown in the browser console</h2>
</body>
</html>

```

Q. Write a program that calculates the sum of even and odd numbers from 1 to a user-defined limit.

```

<!DOCTYPE html>
<html>
<head>
    <title>Even and Odd Sum</title>
    <script>
        function calculateSums() {
            let limit = parseInt(prompt("Enter a positive number:"));
            let evenSum = 0;
            let oddSum = 0;

            for (let i = 1; i <= limit; i++) {
                if (i % 2 === 0) {
                    evenSum += i;
                } else {
                    oddSum += i;
                }
            }
        }
    </script>
</head>
<body>
    <h2>Even and Odd Sum</h2>
    <input type="button" value="Calculate Sums" onclick="calculateSums()" />
    <p>Even Sum: <span id="evenSum"></span></p>
    <p>Odd Sum: <span id="oddSum"></span></p>
</body>

```

```

        alert("Sum of even numbers: " + evenSum + "\nSum of odd numbers: " +
oddSum);
    }

    window.onLoad = calculateSums;
</script>
</head>
<body>
    <h2>Even and odd sums will be shown in an alert box</h2>
</body>
</html>

```

Q. Create a program that prints each element of an array using a loop.

```

<!DOCTYPE html>
<html>
<head>
<title>Array Elements</title>
<script>
    function printArrayElements() {
        let items = [42, "JavaScript", true, "Abhay", 99];

        console.log("Array Elements:");
        for (let i = 0; i < items.length; i++) {
            console.log("Element " + (i + 1) + ": " + items[i]);
        }
    }

    window.onLoad = printArrayElements;
</script>
</head>
<body>
    <h2>Array elements will be shown in the browser console</h2>
</body>
</html>

```

Q. Develop a simple guessing game where the user has to guess a randomly generated number

```

<!DOCTYPE html>
<html>
<head>
    <title>Guessing Game</title>
    <script>
        function startGame() {
            let secret = Math.floor(Math.random() * 100) + 1;
            let guess;

            do {
                guess = parseInt(prompt("Guess a number between 1 and 100:"));

                if (guess < secret) {
                    alert("Too low! Try again.");
                } else if (guess > secret) {
                    alert("Too high! Try again.");
                } else {
                    alert("🎉 Correct! The number was " + secret);
                }
            } while (guess !== secret);
        }

        window.onload = startGame;
    </script>
</head>
<body>
    <h2>Guess the number using alerts</h2>
</body>
</html>

```

Q . Write a program that prints all odd numbers in a given range.

```

<!DOCTYPE html>
<html>
<head>
    <title>Odd Numbers in Range</title>
    <script>
        function printOddNumbers() {
            let start = parseInt(prompt("Enter the starting number:"));
            let end = parseInt(prompt("Enter the ending number:"));

            console.log("Odd numbers from " + start + " to " + end + ":");
        }
    </script>
</head>
<body>
    <h2>Odd Numbers in Range</h2>
</body>

```

```

    for (let i = start; i <= end; i++) {
      if (i % 2 !== 0) {
        console.log(i);
      }
    }

  window.onLoad = printOddNumbers;
</script>
</head>
<body>
  <h2>Odd numbers will be shown in the browser console</h2>
</body>
</html>

```

Q. Create a program that counts the number of vowels in a given string.

```

<!DOCTYPE html>
<html>
<head>
  <title>Vowel Counter</title>
  <script>
    function countVowels() {
      let input = prompt("Enter a string:");
      let count = 0;
      let vowels = "aeiou";

      for (let i = 0; i < input.length; i++) {
        let ch = input[i].toLowerCase();
        if (vowels.includes(ch)) {
          count++;
        }
      }

      alert("Total vowels in the string: " + count);
    }

    window.onLoad = countVowels;
  </script>
</head>
<body>
  <h2>Vowel count will be shown in an alert box</h2>
</body>
</html>

```

Q. Write a program that counts the frequency of each character in a string.

```
<!DOCTYPE html>
<html>
<head>
<title>Character Frequency</title>
<script>
    function countCharacterFrequency() {
        let input = prompt("Enter a string:");
        let freq = {};

        for (let i = 0; i < input.length; i++) {
            let ch = input[i];
            if (freq[ch]) {
                freq[ch]++;
            } else {
                freq[ch] = 1;
            }
        }

        console.log("Character Frequencies:");
        for (let key in freq) {
            console.log(key + ": " + freq[key]);
        }
    }

    window.onload = countCharacterFrequency;
</script>
</head>
<body>
    <h2>Character frequency will be shown in the browser console</h2>
</body>
</html>
```

Q. Write a JavaScript function that takes two numbers as parameters and returns their sum. Test the function with different values.

```
<!DOCTYPE html>
<html>
<head>
<title>Sum Function</title>
<script>
    // Function definition
```

```

function addNumbers(a, b) {
    return a + b;
}

// Testing the function with different values
console.log("Sum of 5 and 3: " + addNumbers(5, 3));
console.log("Sum of 10 and 15: " + addNumbers(10, 15));
console.log("Sum of -2 and 8: " + addNumbers(-2, 8));
</script>
</head>
<body>
    <h2>Check the browser console to see the sum results</h2>
</body>
</html>

```

Q. Create a basic calculator using functions for addition, subtraction, multiplication, and division.

```

<!DOCTYPE html>
<html>
<head>
    <title>Calculator</title>
    <script>
        // Function for addition
        function add(a, b) {
            return a + b;
        }

        // Function for subtraction
        function subtract(a, b) {
            return a - b;
        }

        // Function for multiplication
        function multiply(a, b) {
            return a * b;
        }

        // Function for division
        function divide(a, b) {
            if (b === 0) {
                return "Cannot divide by zero";
            }
        }
    </script>
</head>
<body>
    <h2>Calculator</h2>
    <p>Enter two numbers:</p>
    <input type="text" id="num1">
    <input type="text" id="num2">
    <button onclick="calculate()>Calculate</button>
    <div id="result"></div>
</body>

```

```

        return a / b;
    }

function calculate() {
    let num1 = parseFloat(prompt("Enter first number:"));
    let operator = prompt("Enter operator (+, -, *, /):");
    let num2 = parseFloat(prompt("Enter second number:"));
    let result;

    if (operator === "+") {
        result = add(num1, num2);
    } else if (operator === "-") {
        result = subtract(num1, num2);
    } else if (operator === "*") {
        result = multiply(num1, num2);
    } else if (operator === "/") {
        result = divide(num1, num2);
    } else {
        result = "Invalid operator!";
    }

    alert("Result: " + result);
}

window.onload = calculate;
</script>
</head>
<body>
    <h2>Result will be shown in alert box</h2>
</body>
</html>

```

Q. Write a function that converts temperatures between Celsius and Fahrenheit.

```

<!DOCTYPE html>
<html>
<head>
    <title>Temperature Converter</title>
    <script>
        // Function to convert temperature
        function convertTemperature(temp, scale) {
            if (scale === 'F') {

```

```

        return (temp * 9/5) + 32; // Celsius to Fahrenheit
    } else if (scale === 'C') {
        return (temp - 32) * 5/9; // Fahrenheit to Celsius
    } else {
        return "Invalid scale!";
    }
}

function startConversion() {
    let temperature = parseFloat(prompt("Enter the temperature value:"));
    let scale = prompt("Enter the scale to convert to (C or
F):").toUpperCase();

    let result = convertTemperature(temperature, scale);

    if (result === "Invalid scale!") {
        alert(result);
    } else {
        alert("Converted temperature: " + result.toFixed(2) + "°" + scale);
    }
}

window.onLoad = startConversion;
</script>
</head>
<body>
    <h2>Temperature conversion result will be shown in alert box</h2>
</body>
</html>

```

Q. Create a function that checks if a given string is a palindrome.

```

<!DOCTYPE html>
<html>
<head>
    <title>Palindrome Checker</title>
    <script>
        // Function to check if string is a palindrome
        function isPalindrome(str) {
            let original = str.toLowerCase();
            let reversed = original.split('').reverse().join('');
            return original === reversed;
        }
    </script>
</head>
<body>
    <h2>Enter a string to check if it's a palindrome</h2>
    <input type="text" id="inputString">
    <button type="button" onclick="checkPalindrome()>Check</button>
    <p id="result"></p>
</body>

```

```

function checkPalindrome() {
    let input = prompt("Enter a string to check for palindrome:");
    let result = isPalindrome(input);

    if (result) {
        alert("Yes! '" + input + "' is a palindrome ✅");
    } else {
        alert("No! '" + input + "' is not a palindrome ❌");
    }
}

window.onLoad = checkPalindrome;
</script>
</head>
<body>
    <h2>Palindrome result will be shown in an alert box</h2>
</body>
</html>

```

Q. Implement a function that calculates the factorial of a given number.

```

<!DOCTYPE html>
<html>
<head>
    <title>Factorial Calculator</title>
    <script>
        // Function to calculate factorial
        function factorial(n) {
            let result = 1;
            for (let i = 1; i <= n; i++) {
                result *= i;
            }
            return result;
        }

        function startFactorial() {
            let num = parseInt(prompt("Enter a positive integer:"));

            if (num < 0 || isNaN(num)) {
                alert("Please enter a valid positive number!");
            } else {
                let fact = factorial(num);
                alert("Factorial of " + num + " is: " + fact);
            }
        }
    </script>
</head>
<body>
    <h2>Factorial Calculator</h2>
    <p>Enter a positive integer:</p>
    <button onclick="startFactorial()">Calculate Factorial</button>
</body>
</html>

```

```

        }

    window.onLoad = startFactorial;
</script>
</head>
<body>
    <h2>Factorial result will be shown in an alert box</h2>
</body>
</html>

```

Q. Write a function that calculates the sum of all elements in an array.

```

<!DOCTYPE html>
<html>
<head>
    <title>Array Sum</title>
    <script>
        // Function to calculate sum of array elements
        function sumArray(arr) {
            let sum = 0;
            for (let i = 0; i < arr.length; i++) {
                sum += arr[i];
            }
            return sum;
        }

        // Sample array and function call
        function showSum() {
            let numbers = [5, 10, 15, 20, 25];
            let result = sumArray(numbers);
            alert("Sum of array elements: " + result);
        }

        window.onLoad = showSum;
    </script>
</head>
<body>
    <h2>Sum of array will be shown in alert box</h2>
</body>
</html>

```

Q. Create a function that returns the Fibonacci sequence up to a specified number of terms.

```
<!DOCTYPE html>
<html>
<head>
    <title>Fibonacci Sequence</title>
    <script>
        // Function to generate Fibonacci sequence
        function fibonacci(n) {
            let sequence = [];

            for (let i = 0; i < n; i++) {
                if (i === 0) {
                    sequence.push(0);
                } else if (i === 1) {
                    sequence.push(1);
                } else {
                    sequence.push(sequence[i - 1] + sequence[i - 2]);
                }
            }

            return sequence;
        }

        function showFibonacci() {
            let num = parseInt(prompt("Enter the number of terms:"));

            if (isNaN(num) || num <= 0) {
                alert("Please enter a valid positive number.");
            } else {
                let result = fibonacci(num);
                alert("Fibonacci sequence:\n" + result.join(", "));
            }
        }

        window.onload = showFibonacci;
    </script>
</head>
<body>
    <h2>Fibonacci sequence will be shown in an alert box</h2>
</body>
</html>
```

Q. Write a function that finds the maximum number in an array.

```
<!DOCTYPE html>
<html>
<head>
    <title>Find Max in Array</title>
    <script>
        // Function to find maximum value in an array
        function findMax(arr) {
            let max = arr[0]; // Assume first element is max

            for (let i = 1; i < arr.length; i++) {
                if (arr[i] > max) {
                    max = arr[i];
                }
            }

            return max;
        }

        function showMax() {
            let numbers = [12, 45, 7, 89, 23, 56];
            let maxValue = findMax(numbers);
            alert("Maximum number is: " + maxValue);
        }

        window.onLoad = showMax;
    </script>
</head>
<body>
    <h2>Maximum number will be shown in alert box</h2>
</body>
</html>
```

Q. Implement a function that reverses a given string.

```
<!DOCTYPE html>
<html>
<head>
    <title>Reverse String</title>
    <script>
        // Function to reverse a string
        function reverseString(str) {
            let reversed = "";
```

```

        for (let i = str.length - 1; i >= 0; i--) {
            reversed += str[i];
        }
        return reversed;
    }

    function showReversed() {
        let input = prompt("Enter a string to reverse:");
        let result = reverseString(input);
        alert("Reversed string: " + result);
    }

    window.onLoad = showReversed;
</script>
</head>
<body>
    <h2>Reversed string will be shown in alert box</h2>
</body>
</html>

```

Q. Create a function that counts the number of vowels in a string.

```

<!DOCTYPE html>
<html>
<head>
    <title>Vowel Counter</title>
    <script>
        // Function to count vowels in a string
        function countVowels(str) {
            let count = 0;
            let vowels = "aeiou";
            str = str.toLowerCase();

            for (let i = 0; i < str.length; i++) {
                if (vowels.includes(str[i])) {
                    count++;
                }
            }

            return count;
        }

        function showVowelCount() {

```

```

let input = prompt("Enter a string:");
let result = countVowels(input);
alert("Number of vowels: " + result);
}

window.onLoad = showVowelCount;
</script>
</head>
<body>
<h2>Vowel count will be shown in alert box</h2>
</body>
</html>

```

Q. Write a function that generates a random number within a specified range

```

<!DOCTYPE html>
<html>
<head>
<title>Random Number Generator</title>
<script>
// Function to generate random number between min and max
function getRandomNumber(min, max) {
    return Math.floor(Math.random() * (max - min + 1)) + min;
}

function showRandomNumber() {
    let min = parseInt(prompt("Enter minimum number:"));
    let max = parseInt(prompt("Enter maximum number:"));

    if (isNaN(min) || isNaN(max) || min > max) {
        alert("Please enter valid numbers where min <= max");
    } else {
        let random = getRandomNumber(min, max);
        alert("Random number between " + min + " and " + max + ": " + random);
    }
}

window.onLoad = showRandomNumber;
</script>
</head>
<body>
<h2>Random number will be shown in alert box</h2>
</body>
</html>

```

Q. Create a function that filters an array based on a specified condition.

```
<!DOCTYPE html>
<html>
<head>
    <title>Filter Array</title>
    <script>
        // Function to filter array based on condition
        function filterArray(arr, condition) {
            let result = [];
            for (let i = 0; i < arr.length; i++) {
                if (condition(arr[i])) {
                    result.push(arr[i]);
                }
            }
            return result;
        }

        function testFilter() {
            const numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10];

            // Condition: even numbers
            const evenNumbers = filterArray(numbers, function(num) {
                return num % 2 === 0;
            });

            // Condition: odd numbers
            const oddNumbers = filterArray(numbers, function(num) {
                return num % 2 !== 0;
            });

            alert("Even numbers: " + evenNumbers.join(", "));
            alert("Odd numbers: " + oddNumbers.join(", "));
        }

        window.onload = testFilter;
    </script>
</head>
<body>
    <h2>Filtered array results will be shown in alert boxes</h2>
</body>
</html>
```

Q . Write a function that concatenates two arrays into one.

```
<!DOCTYPE html>
<html>
<head>
    <title>Concatenate Arrays</title>
    <script>
        // Function to concatenate two arrays
        function concatArrays(arr1, arr2) {
            return arr1.concat(arr2);
        }

        function showCombinedArray() {
            const array1 = [1, 2, 3];
            const array2 = [4, 5, 6];

            const combined = concatArrays(array1, array2);

            alert("Combined array: " + combined.join(", "));
        }

        window.onload = showCombinedArray;
    </script>
</head>
<body>
    <h2>Combined array will be shown in alert box</h2>
</body>
</html>
```

Q . Create a function that removes duplicate values from an array.

```
<!DOCTYPE html>
<html>
<head>
    <title>Remove Duplicates</title>
    <script>
        // Function to remove duplicates from an array
        function removeDuplicates(arr) {
            let unique = [];

            for (let i = 0; i < arr.length; i++) {
                if (!unique.includes(arr[i])) {
                    unique.push(arr[i]);
                }
            }
        }
    </script>
</head>
<body>
    <h2>Duplicates removed from array</h2>
    <pre>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]</pre>
    <pre>[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]</pre>
</body>
</html>
```

```

        }

        return unique;
    }

function showUniqueArray() {
    const numbers = [1, 2, 3, 2, 4, 5, 1, 6, 4];
    const result = removeDuplicates(numbers);
    alert("Array without duplicates: " + result.join(", "));
}

window.onLoad = showUniqueArray;
</script>
</head>
<body>
    <h2>Unique array will be shown in alert box</h2>
</body>
</html>

```

Q. Implement a function that checks if a number is even or odd.

```

<!DOCTYPE html>
<html>
<head>
    <title>Even or Odd Checker</title>
    <script>
        // Function to check if a number is even
        function isEven(num) {
            return num % 2 === 0;
        }

        function checkEvenOdd() {
            let input = parseInt(prompt("Enter a number:"));

            if (isNaN(input)) {
                alert("Please enter a valid number.");
            } else {
                if (isEven(input)) {
                    alert(input + " is even.");
                } else {
                    alert(input + " is odd.");
                }
            }
        }
    </script>
</head>
<body>
    <h2>Even or Odd Checker</h2>
    <p>Enter a number:</p>
    <button onclick="checkEvenOdd()">Check</button>
</body>
</html>

```

```

        }

        window.onLoad = checkEvenOdd;
    </script>
</head>
<body>
    <h2>The result will be shown in an alert box</h2>
</body>
</html>

```

Q. Create a function that greets the user with their name

```

<!DOCTYPE html>
<html>
<head>
    <title>Greet User</title>
    <script>
        // Function to greet the user
        function greetUser(name) {
            alert("Hello, " + name + "! Welcome to our website.");
        }

        function askName() {
            let userName = prompt("Enter your name:");
            if (userName) {
                greetUser(userName);
            } else {
                alert("You didn't enter a name.");
            }
        }

        window.onLoad = askName;
    </script>
</head>
<body>
    <h2>Greeting will appear in an alert box</h2>
</body>
</html>

```

6. Arrays and Objects

Q. Create an array of objects representing students (name, age, grade). Write a program to loop through the array and display each student's details in the console.

```
<!DOCTYPE html>
<html>
<head>
    <title>Student Details</title>
    <script>
        function showStudents() {
            // Array of student objects
            let students = [
                { name: "Abhay", age: 20, grade: "A" },
                { name: "Minaxi", age: 19, grade: "B" },
                { name: "Raj", age: 21, grade: "A+" },
                { name: "Pooja", age: 18, grade: "C" }
            ];

            // Loop through the array and display details
            for (let i = 0; i < students.length; i++) {
                console.log("Student " + (i + 1));
                console.log("Name: " + students[i].name);
                console.log("Age: " + students[i].age);
                console.log("Grade: " + students[i].grade);
                console.log("-----");
            }
        }

        window.onload = showStudents;
    </script>
</head>
<body>
    <h2>Open console to see student details</h2>
</body>
</html>
```

Q. Create an array of five different fruits and display each fruit in the console

```
<!DOCTYPE html>
<html>
<head>
    <title>Fruit List</title>
    <script>
        function showFruits() {
```

```

// Step 1: Define an array of fruits
let fruits = ["Apple", "Banana", "Mango", "Orange", "Pineapple"];

// Step 2: Loop through the array and print each fruit
for (let i = 0; i < fruits.length; i++) {
    console.log("Fruit " + (i + 1) + ": " + fruits[i]);
}

window.onload = showFruits;
</script>
</head>
<body>
    <h2>Open the console to see the fruit list</h2>
</body>
</html>

```

Q. Write a program that calculates and displays the length of an array.

```

<!DOCTYPE html>
<html>
<head>
    <title>Array Length</title>
    <script>
        function showArrayLength() {
            // Step 1: Define an array with 6 elements
            let colors = ["Red", "Green", "Blue", "Yellow", "Purple", "Orange"];

            // Step 2: Use Length property to get array size
            let lengthOfArray = colors.length;

            // Display the result
            alert("The length of the array is: " + lengthOfArray);
        }

        window.onload = showArrayLength;
    </script>
</head>
<body>
    <h2>Array length will be shown in alert box</h2>
</body>
</html>

```

Q. Create a program that adds a new fruit to an array and removes the last fruit.

```
<!DOCTYPE html>
<html>
<head>
    <title>Even Number Filter</title>
    <script>
        // Step 2: Function to filter even numbers
        function filterEvenNumbers(arr) {
            let evenNumbers = [];

            for (let i = 0; i < arr.length; i++) {
                if (arr[i] % 2 === 0) {
                    evenNumbers.push(arr[i]);
                }
            }

            return evenNumbers;
        }

        function showEvenNumbers() {
            // Step 1: Define array with mixed numbers
            const numbers = [10, 3, 7, 8, 12, 5, 6, 13];

            // Step 3: Filter and Log even numbers
            const result = filterEvenNumbers(numbers);
            console.log("Even numbers:", result);
        }

        window.onload = showEvenNumbers;
    </script>
</head>
<body>
    <h2>Open console to see even numbers</h2>
</body>
</html>
```

Q. Write a program that sorts an array of strings alphabetically.

```
<!DOCTYPE html>
<html>
<head>
    <title>Sort Names</title>
    <script>
```

```

function sortNames() {
    // Step 1: Define array of names
    let names = ["Zara", "Abhay", "Minaxi", "Raj", "Priya"];

    // Step 2: Sort the array alphabetically
    names.sort();

    // Step 3: Display the sorted array in console
    console.log("Sorted names:", names);
}

window.onload = sortNames;
</script>
</head>
<body>
    <h2>Open console to see sorted names</h2>
</body>
</html>

```

Q. Create a program that merges two arrays into one.

```

<!DOCTYPE html>
<html>
<head>
    <title>Merge Arrays</title>
    <script>
        function mergeArrays() {
            // Step 1: Define two arrays
            let array1 = ["Apple", "Banana", "Mango"];
            let array2 = ["Orange", "Pineapple", "Grapes"];

            // Step 2: Merge arrays using concat
            let mergedArray = array1.concat(array2);

            // Step 3: Log the result
            console.log("Merged array:", mergedArray);
        }

        window.onload = mergeArrays;
    </script>
</head>
<body>
    <h2>Open the console to see merged array</h2>
</body>
</html>

```

Q. Write a function that calculates the sum of all elements in an array of numbers.

```
<!DOCTYPE html>
<html>
<head>
    <title>Sum of Array</title>
    <script>
        // Step 2: Define sumArray function
        function sumArray(arr) {
            let sum = 0;
            for (let i = 0; i < arr.length; i++) {
                sum += arr[i];
            }
            return sum;
        }

        function displaySum() {
            // Step 1: Define an array of numbers
            const numbers = [5, 10, 15, 20, 25];

            // Step 3: Call the function and show result
            const total = sumArray(numbers);
            alert("The sum of the array is: " + total);
        }

        window.onload = displaySum;
    </script>
</head>
<body>
    <h2>The total will be shown in an alert box</h2>
</body>
</html>
```

Q. Implement a function that reverses the order of elements in an array.

```
<!DOCTYPE html>
<html>
<head>
    <title>Reverse Array</title>
    <script>
        // Step 2: Function to reverse array using Loop
        function reverseArray(arr) {
            let reversed = [];
            for (let i = arr.length - 1; i >= 0; i--) {
                reversed.push(arr[i]);
            }
            return reversed;
        }
    </script>
</head>
<body>
    <h2>The reversed array is:</h2>
    <pre>[</pre>
</body>
</html>
```

```

        for (let i = arr.length - 1; i >= 0; i--) {
            reversed.push(arr[i]);
        }

        return reversed;
    }

    function displayReversedArray() {
        // Step 1: Define an array with 5 elements
        const items = [10, 20, 30, 40, 50];

        // Step 3: Reverse and Log the array
        const result = reverseArray(items);
        console.log("Original array:", items);
        console.log("Reversed array:", result);
    }

    window.onload = displayReversedArray;
</script>
</head>
<body>
    <h2>Open the console to see the reversed array</h2>
</body>
</html>

```

Q. Create a program that finds and displays the maximum value in an array of numbers

```

<!DOCTYPE html>
<html>
<head>
    <title>Find Max Value</title>
    <script>
        function findMaxValue() {
            // Step 1: Define an array of numbers
            let numbers = [12, 45, 7, 89, 34, 67];

            // Step 2: Assume first element is max
            let max = numbers[0];

            // Step 2: Loop to find max
            for (let i = 1; i < numbers.length; i++) {
                if (numbers[i] > max) {
                    max = numbers[i];
                }
            }
        }
    </script>
</head>
<body>
    <h2>Find Max Value</h2>
    <button onclick="findMaxValue()">Find Max</button>
    <p>The maximum value is: <span id="max"></span></p>
</body>
</html>

```

```

    // Step 3: Display max value in console
    console.log("Maximum value in array:", max);
}

window.onLoad = find.MaxValue;
</script>
</head>
<body>
<h2>Open console to see the maximum value</h2>
</body>
</html>

```

Q . Define an object that represents a book with properties like title, author, and year.

```

<!DOCTYPE html>
<html>
<head>
<title>Book Object</title>
<script>
    function showBookDetails() {
        // Step 1: Define the book object
        let book = {
            title: "The Alchemist",
            author: "Paulo Coelho",
            year: 1988
        };

        // Step 2: Log the object to console
        console.log("Book Details:", book);
    }

    window.onLoad = showBookDetails;
</script>
</head>
<body>
<h2>Open console to see book object details</h2>
</body>
</html>

```

Q . Write a program that accesses and displays the properties of an object.

```

<!DOCTYPE html>
<html>
<head>

```

```

<title>Car Object</title>
<script>
    function displayCarDetails() {
        // Step 1: Define the car object
        let car = {
            make: "Toyota",
            model: "Innova",
            year: 2022
        };

        // Step 2: Access and Log each property
        console.log("Car Make: " + car.make);
        console.log("Car Model: " + car.model);
        console.log("Car Year: " + car.year);
    }

    window.onload = displayCarDetails;
</script>
</head>
<body>
    <h2>Open console to see car details</h2>
</body>
</html>

```

Q . Create an object and update one of its properties

```

<!DOCTYPE html>
<html>
<head>
    <title>Update Object</title>
    <script>
        function updatePersonAge() {
            // Step 1: Create a person object
            let person = {
                name: "Abhay",
                age: 22
            };

            console.log("Before update:", person);

            // Step 2: Update the age property
            person.age = 23;

            // Step 3: Log updated object
            console.log("After update:", person);
        }
    </script>
</head>
<body>
</body>
</html>

```

```
window.onload = updatePersonAge;  
</script>  
</head>  
<body>  
  <h2>Open console to see person object update</h2>  
</body>  
</html>
```

Q. Define an object that has a method to display a greeting.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Greeting Method</title>  
  <script>  
    function showGreeting() {  
      // Step 1: Define an object with a method  
      let greeting = {  
        sayHello: function() {  
          console.log("Hello! Welcome to JavaScript Assignments!");  
        }  
      };  
  
      // Step 2: Call the method  
      greeting.sayHello();  
    }  
  
    window.onload = showGreeting;  
  </script>  
</head>  
<body>  
  <h2>Open the console to see the greeting message</h2>  
</body>  
</html>
```

Q . Create an object that contains another object as a property.

```
<!DOCTYPE html>  
<html>  
<head>  
  <title>Nested Object</title>  
  <script>  
    function showPrincipalDetails() {
```

```

// Step 1: Define the main object with nested object
let school = {
    name: "Sunrise Public School",
    address: "Ahmedabad, Gujarat",
    principal: {
        name: "Mr. Sharma",
        yearsOfService: 15
    }
};

// Step 2: Log the nested object
console.log("Principal Details:", school.principal);
}

window.onload = showPrincipalDetails;
</script>
</head>
<body>
    <h2>Open the console to see nested object output</h2>
</body>
</html>

```

Q . Create an object based on user input.

```

<!DOCTYPE html>
<html>
<head>
    <title>User Object from Input</title>
    <script>
        function createUserObject() {
            // Step 1: Prompt for name and age
            let name = prompt("Enter your name:");
            let age = prompt("Enter your age:");

            // Step 2: Create user object
            let user = {
                name: name,
                age: parseInt(age)
            };

            // Step 3: Log the object
            console.log("User Details:", user);
        }

        window.onload = createUserObject;
    </script>
</head>

```

```
<body>
  <h2>Check console for user object from input</h2>
</body>
</html>
```

Q . Write a program that loops through an object's properties and displays each property and its value.

```
<!DOCTYPE html>
<html>
<head>
  <title>Object Property Loop</title>
  <script>
    function displayObjectProperties() {
      // Step 1: Define an object with at least 3 properties
      let student = {
        name: "Abhay",
        age: 22,
        course: "Node.js"
      };

      // Step 2: Use for...in loop to display key-value pairs
      for (let key in student) {
        console.log(key + ": " + student[key]);
      }
    }

    window.onload = displayObjectProperties;
  </script>
</head>
<body>
  <h2>Open console to see object properties and values</h2>
</body>
</html>
```

Q . Create an array of objects representing different books.

```
<!DOCTYPE html>
<html>
<head>
  <title>Books Array</title>
```

```

<script>
  function displayBookTitles() {
    // Step 1: Define array of book objects
    let books = [
      { title: "The Alchemist", author: "Paulo Coelho", year: 1988 },
      { title: "Wings of Fire", author: "A.P.J. Abdul Kalam", year: 1999 },
      { title: "Harry Potter", author: "J.K. Rowling", year: 1997 }
    ];

    // Step 2: Use a Loop to display titles
    console.log("Book Titles:");
    for (let i = 0; i < books.length; i++) {
      console.log("- " + books[i].title);
    }
  }

  window.onload = displayBookTitles;
</script>
</head>
<body>
  <h2>Open console to see book titles</h2>
</body>
</html>

```

Q . Implement object destructuring to extract properties from an object.

```

<!DOCTYPE html>
<html>
<head>
  <title>Object Destructuring</title>
<script>
  function destructureLaptop() {
    // Step 1: Define the object
    const laptop = {
      brand: "Dell",
      model: "Inspiron 15",
      price: 55000
    };

    // Step 2: Destructure the object
    const { brand, model, price } = laptop;

    // Step 3: Log the values
    console.log("Brand:", brand);
    console.log("Model:", model);
  }
</script>

```

```

        console.log("Price:", price);
    }

    window.onload = destructureLaptop;
</script>
</head>
<body>
<h2>Open console to see destructured laptop properties</h2>
</body>
</html>

```

Q. Create a program that merges two objects into one.

```

<!DOCTYPE html>
<html>
<head>
<title>Merge Objects</title>
<script>
    function mergeObjects() {
        // Step 1: Define two objects
        const person = { name: "Abhay", age: 22 };
        const contact = { phone: "9876543210", city: "Ahmedabad" };

        // Step 2: Merge using Object.assign()
        const merged1 = Object.assign({}, person, contact);
        console.log("Merged using Object.assign():", merged1);

        // Step 3: Merge using spread operator
        const merged2 = { ...person, ...contact };
        console.log("Merged using spread operator:", merged2);
    }

    window.onload = mergeObjects;
</script>
</head>
<body>
<h2>Open console to see merged object results</h2>
</body>
</html>

```

Q. Write a higher-order function that takes an array and a callback function as arguments. The function should apply the callback to each element in the array and return a new array with the results.

```

<!DOCTYPE html>
<html>
<head>
    <title>Higher-Order Function</title>
    <script>
        function applyCallbackToArray(arr, callback) {
            let result = [];
            for (let i = 0; i < arr.length; i++) {
                result.push(callback(arr[i]));
            }
            return result;
        }

        function square(num) {
            return num * num;
        }

        function runExample() {
            let numbers = [1, 2, 3, 4, 5];

            let squaredNumbers = applyCallbackToArray(numbers, square);

            console.log("Original Array:", numbers);
            console.log("Squared Array:", squaredNumbers);
        }

        window.onload = runExample;
    </script>
</head>
<body>
    <h2>Open console to see higher-order function result</h2>
</body>
</html>

```

Q. Write a program that simulates an asynchronous operation using setTimeout. Create a promise that resolves after a specified delay and logs a message to the console.

```

<!DOCTYPE html>
<html>
<head>
    <title>Async Operation with Promise</title>
    <script>
        function simulateAsyncTask() {
            // Step 1: Create a promise that resolves after 2 seconds

```

```

let promise = new Promise(function(resolve) {
    setTimeout(function() {
        resolve("☑ Asynchronous operation completed after 2 seconds");
    }, 2000); // 2000 ms = 2 seconds
});

// Step 2: Handle the resolved promise
promise.then(function(message) {
    console.log(message);
});
}

window.onLoad = simulateAsyncTask;
</script>
</head>
<body>
    <h2>Check the console after 2 seconds</h2>
</body>
</html>

```

Q. Create a simple HTML page with buttons. Write JavaScript to change the backgroundcolor of the page when each button is clicked.

```

<!DOCTYPE html>
<html>
<head>
    <title>Background Color Changer</title>
    <style>
        button {
            margin: 10px;
            padding: 10px 20px;
            font-size: 16px;
            cursor: pointer;
        }
    </style>
    <script>
        function changeColor(color) {
            document.body.style.backgroundColor = color;
        }
    </script>
</head>
<body>
    <h2>Click a button to change background color</h2>
    <!-- Buttons -->

```

```

<button onclick="changeColor('lightblue')">Light Blue</button>
<button onclick="changeColor('lightgreen')">Light Green</button>
<button onclick="changeColor('lightyellow')">Light Yellow</button>
<button onclick="changeColor('lightpink')">Light Pink</button>
</body>
</html>

```

Q. Rewrite a set of traditional JavaScript functions using ES6 arrow functions and use template literals for string concatenation.

```

<!DOCTYPE html>
<html>
<head>
    <title>Arrow Functions & Template Literals</title>
    <script>
        // Traditional function
        /*
        function greet(name) {
            return "Hello, " + name + "!";
        }

        function add(a, b) {
            return "Sum is: " + (a + b);
        }
        */

        // ☑️ ES6 Arrow Functions with Template Literals
        const greet = (name) => `Hello, ${name}!`;

        const add = (a, b) => `Sum is: ${a + b}`;

        function runExample() {
            let userName = "Abhay";
            let result = greet(userName);
            console.log(result); // Hello, Abhay!

            let sumResult = add(10, 15);
            console.log(sumResult); // Sum is: 25
        }

        window.onload = runExample;
    </script>
</head>
<body>
    <h2>Open the console to see arrow function output</h2>
</body>
</html>

```

Q . Demonstrate the difference between let and const by creating variables.

```
<!DOCTYPE html>
<html>
<head>
    <title>Let vs Const</title>
    <script>
        function showLetConstDifference() {
            // Using Let
            let city = "Ahmedabad";
            console.log("Initial city (let):", city);

            city = "Surat"; // ✅ Changing value
            console.log("Updated city (let):", city);

            // Using const
            const country = "India";
            console.log("Country (const):", country);

            try {
                country = "USA"; // ❌ This will throw an error
                console.log("Updated country (const):", country);
            } catch (error) {
                console.log("Error when changing const:", error.message);
            }
        }

        window.onload = showLetConstDifference;
    </script>
</head>
<body>
    <h2>Open console to see let vs const behavior</h2>
</body>
</html>
```

Q. Convert a traditional function to an arrow function.

```
<!DOCTYPE html>
<html>
<head>
    <title>Arrow Function Conversion</title>
    <script>
        function convertFunction() {
            // Step 1: Traditional function
```

```

        function sumTraditional(a, b) {
            return a + b;
        }

        console.log("Traditional Function Result:", sumTraditional(10, 5)); // 15

        // Step 2: Converted to arrow function
        const sumArrow = (a, b) => a + b;

        console.log("Arrow Function Result:", sumArrow(10, 5)); // 15
    }

    window.onload = convertFunction;
</script>
</head>
<body>
    <h2>Open console to see traditional vs arrow function result</h2>
</body>
</html>

```

Q. Use template literals to create a formatted string.

```

<!DOCTYPE html>
<html>
<head>
    <title>Template Literals</title>
    <script>
        function showGreeting() {
            // Step 1: Define variables
            const firstName = "Abhay";
            const lastName = "Kalal";
            const age = 22;

            // Step 2: Use template literal
            const greeting = `Hello, my name is ${firstName} ${lastName} and I am
${age} years old.`;

            // Step 3: Log to console
            console.log(greeting);
        }

        window.onload = showGreeting;
    </script>
</head>
<body>
    <h2>Open console to see the greeting message</h2>

```

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

Q. Create a function that accepts parameters with default values.

```
<!DOCTYPE html>
<html>
<head>
  <title>Default Parameters</title>
  <script>
    function greet(name, greeting = "Hello") {
      return `${greeting}, ${name}!`;
    }

    function runExample() {
      // Call with both parameters
      console.log(greet("Abhay", "Hi"));           // Output: Hi, Abhay!

      // Call with only one parameter (uses default greeting)
      console.log(greet("Minaxi"));                 // Output: Hello, Minaxi!
    }

    window.onload = runExample;
  </script>
</head>
<body>
  <h2>Open console to see default parameter demo</h2>
</body>
</html>
```

Q. Use destructuring to extract properties from an object.

```
<!DOCTYPE html>
<html>
<head>
  <title>Object Destructuring</title>
  <script>
    function showPersonDetails() {
      // Step 1: Create object
      const person = {
        name: "Abhay",
        age: 22,
        city: "Ahmedabad"
```

```

};

// Step 2: Use destructuring
const { name, age } = person;

// Step 3: Log the extracted values
console.log("Name:", name);
console.log("Age:", age);
}

window.onload = showPersonDetails;
</script>
</head>
<body>
<h2>Open console to see destructured properties</h2>
</body>
</html>

```

Q. Use the spread operator to combine arrays.

```

<!DOCTYPE html>
<html>
<head>
<title>Spread Operator - Combine Arrays</title>
<script>
function combineArrays() {
    // Step 1: Create two arrays
    const array1 = [1, 2, 3];
    const array2 = [4, 5, 6];

    // Step 2: Combine using spread operator
    const combinedArray = [...array1, ...array2];

    // Step 3: Log result
    console.log("Combined Array:", combinedArray);
}

window.onload = combineArrays;
</script>
</head>
<body>
<h2>Open console to see the combined array</h2>
</body>
</html>

```

Q. Implement a function that uses rest parameters to accept an indefinite number of arguments.

```
<!DOCTYPE html>
<html>
<head>
  <title>Rest Parameters - Sum Function</title>
  <script>
    // Step 1: Define function using rest parameter
    function sumAll(...numbers) {
      let total = 0;
      for (let num of numbers) {
        total += num;
      }
      return total;
    }

    function runExamples() {
      // Step 2: Call function with different sets of numbers
      console.log("Sum of 1, 2, 3:", sumAll(1, 2, 3));           // 6
      console.log("Sum of 5, 10, 15, 20:", sumAll(5, 10, 15, 20)); // 50
      console.log("Sum of 100:", sumAll(100));                   // 100
    }

    window.onload = runExamples;
  </script>
</head>
<body>
  <h2>Open console to see sumAll results</h2>
</body>
</html>
```

Q. Create a simple class to represent a Rectangle.

```
<!DOCTYPE html>
<html>
<head>
  <title>Rectangle Class</title>
  <script>
    // Step 1: Define the Rectangle class
    class Rectangle {
      constructor(width, height) {
        this.width = width;
        this.height = height;
      }
    }
  </script>
</head>
<body>
  <h2>Open console to see Rectangle class</h2>
</body>
</html>
```

```

        this.height = height;
    }

    // Step 2: Method to calculate area
    calculateArea() {
        return this.width * this.height;
    }
}

function runExample() {
    // Create an instance of Rectangle
    const rect = new Rectangle(10, 5);

    // Log the area
    console.log(`Area of rectangle: ${rect.calculateArea()}`);
}

window.onload = runExample;
</script>
</head>
<body>
    <h2>Open console to see Rectangle area</h2>
</body>
</html>

```

Q. Create a promise that resolves after a specified time.

```

<!DOCTYPE html>
<html>
<head>
    <title>Promise with Delay</title>
    <script>
        // Step 1: Define a function that returns a promise
        function wait() {
            return new Promise((resolve) => {
                setTimeout(() => {
                    resolve("Promise resolved after 2 seconds");
                }, 2000); // 2 seconds = 2000 milliseconds
            });
        }

        // Step 2: Call the function and Log message on resolve
        function runExample() {
            console.log("Waiting for the promise...");

            wait().then((message) => {
                console.log(message);
            });
        }
    </script>
</head>
<body>
    <h2>Open console to see the delay</h2>
</body>
</html>

```

```
    });
}

window.onload = runExample;
</script>
</head>
<body>
<h2>Open console to see delayed promise result</h2>
</body>
</html>
```

Q. Use async/await syntax to work with a promise.

```
<!DOCTYPE html>
<html>
<head>
<title>Async/Await Example</title>
<script>
// Step 1: Create async function to fetch data
async function fetchData() {
  try {
    // Step 2: Use await with fetch
    const response = await
fetch("https://jsonplaceholder.typicode.com/posts/1");
    const data = await response.json();

    // Step 3: Log the fetched data
    console.log("Fetched Data:", data);
  } catch (error) {
    console.error("Error fetching data:", error);
  }
}

window.onload = fetchData;
</script>
</head>
<body>
<h2>Open console to see async/await API result</h2>
</body>
</html>
```

Q. Create a simple module and import it into another file.

```

// square.js
export function square(num) {
  return num * num;
}

// main.js
import { square } from './square.js';

console.log("Square of 5 is:", square(5));

<!DOCTYPE html>
<html>
<head>
  <title>JavaScript Module Example</title>
</head>
<body>
  <h2>Open console to see the square result</h2>

  <!-- Load main.js as a module -->
  <script type="module" src="main.js"></script>
</body>
</html>

```

Q. Demonstrate the usage of Set and Map.

```

<!DOCTYPE html>
<html>
<head>
  <title>Set and Map Example</title>
  <script>
    function showSetAndMap() {
      // Step 1: Create a Set and add elements
      const mySet = new Set();
      mySet.add("apple");
      mySet.add("banana");
      mySet.add("apple"); // Duplicate, will be ignored
      mySet.add("mango");

      console.log("Set contents:");
      for (let item of mySet) {
        console.log(item);
      }

      // Step 2: Create a Map and add key-value pairs
    }
  </script>
</head>
<body>
  <h2>Set and Map Example</h2>
  <p>This example demonstrates the usage of Set and Map objects in JavaScript.</p>
  <p>A Set is used to store unique values. In this example, we add several items to a Set and then log its contents to the console. Note that duplicates are ignored.</p>
  <p>A Map is used to store key-value pairs. In this example, we create a Map and add some key-value pairs to it, then log the entire Map to the console.</p>
</body>

```

```

const myMap = new Map();
myMap.set("name", "Abhay");
myMap.set("age", 22);
myMap.set("city", "Ahmedabad");

console.log("\nMap contents:");
for (let [key, value] of myMap) {
  console.log(` ${key} = ${value}`);
}

window.onload = showSetAndMap;
</script>
</head>
<body>
  <h2>Open console to see Set and Map output</h2>
</body>
</html>

```

Q. Use enhanced object literals to create an object.

```

<!DOCTYPE html>
<html>
<head>
  <title>Enhanced Object Literal Example</title>
  <script>
    function createPerson() {
      // Step 1: Create variables
      const name = "Abhay";
      const age = 22;

      // Step 2: Use enhanced object literal
      const person = {
        name, // same as name: name
        age, // same as age: age
        greet() {
          console.log(`Hello, my name is ${this.name} and I am ${this.age} years old.`);
        }
      };

      // Step 3: Log the object
      console.log("Person Object:", person);
      person.greet();
    }

    window.onload = createPerson;
  </script>
</head>
<body>
  <h2>Enhanced Object Literal Example</h2>
</body>
</html>

```

```
</script>
</head>
<body>
  <h2>Open console to see enhanced object literal output</h2>
</body>
</html>
```

Q. Use map, filter, and reduce to manipulate an array.

```
<!DOCTYPE html>
<html>
<head>
  <title>Array Manipulation</title>
  <script>
    function manipulateArray() {
      // Step 1: Define an array of numbers
      const numbers = [1, 2, 3, 4, 5];

      // Step 2: Square each number using map
      const squared = numbers.map(num => num * num);

      // Step 3: Filter even numbers
      const even = numbers.filter(num => num % 2 === 0);

      // Step 4: Sum of all numbers using reduce
      const sum = numbers.reduce((total, num) => total + num, 0);

      // Log the results
      console.log("Original Array:", numbers);
      console.log("Squared Array:", squared);
      console.log("Even Numbers:", even);
      console.log("Sum of All Numbers:", sum);
    }

    window.onload = manipulateArray;
  </script>
</head>
<body>
  <h2>Open console to see the output of map, filter, and reduce</h2>
</body>
</html>
```

Q. Use the for...of loop to iterate over an array.

```

<!DOCTYPE html>
<html>
<head>
    <title>for...of Loop Example</title>
    <script>
        function loopArray() {
            // Step 1: Define an array of strings
            const fruits = ["Apple", "Banana", "Mango", "Orange", "Grapes"];

            // Step 2: Use for...of loop to print each element
            console.log("Fruits List:");
            for (const fruit of fruits) {
                console.log(fruit);
            }
        }

        window.onload = loopArray;
    </script>
</head>
<body>
    <h2>Open console to see the output from for...of loop</h2>
</body>
</html>

```

Q. Write a function that attempts to parse a JSON string and uses try...catch to handle any errors that may occur.

```

<!DOCTYPE html>
<html>
<head>
    <title>JSON Parsing with Error Handling</title>
    <script>
        function parseJsonString(jsonStr) {
            try {
                const result = JSON.parse(jsonStr);
                console.log("Parsed JSON object:", result);
            } catch (error) {
                console.log("Error parsing JSON:", error.message);
            }
        }

        window.onload = function () {
            const validJson = '{"name": "Abhay", "age": 22}';
            const invalidJson = '{name: Abhay, age: 22}'; // invalid JSON format
        }
    </script>
</head>
<body>
</body>
</html>

```

```
        console.log("Parsing valid JSON:");
        parseJsonString(validJson);

        console.log("\nParsing invalid JSON:");
        parseJsonString(invalidJson);
    };
</script>
</head>
<body>
    <h2>Open console to see JSON parsing with try...catch</h2>
</body>
</html>
```