**Advanced Web Designing Technologies**

**Lab Manual**

**Bachelor of Technology**

**In**

**Computer Science Engineering**

Submitted by:

**ANOUSHKA SRIVASTAVA**

**Semester 5**

**Section A**

**Enrollment No. A50105219039**

Submitted to:

**Ms Swati Gupta**

**CSE Department**

****

**Department of Computer Science & Engineering**

**Amity School of Engineering & Technology**

**Amity University Haryana**

**Gurgaon, India**

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S.no.** | **Experiment** | **Date** | **Page Number** |
|  | **Understanding different output format in JS​** | **13-08-2021** | **3-16** |
|  | **To implement conditional statements by using different looping constructs, JS date operations and JS timers.** | **20-08-21** | **17-29** |
|  | **To implement conditional statements by using different looping constructs, JS date operations and JS timers.** | **27-08-2021** | **30-39** |
|  | **To perform the implementation of JS arrays and its associated methods.** | **3-09-2021** | **40-48** |
|  | **To implement JS programs using functions and to implement some function: Ceil, Random, Min-max**. | **10-09-2021** | **49-57** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**EXPERIMENT -1**

**AIM:** Understanding different output format in JS​

**Basic Introduction :**

You can embed the JS code directly within your web page by placing it between the <script> and </script> tags. The tag indicates the browser that the contained statements are to be interpreted as executable script and not HTML.

Generating output in JS

‘JavaScript’ can “display” data in different ways:

* Writing into an HTML element, using innerHTML.
* Writing into the HTML output using document.write().
* Writing into an alert box, using window.alert().
* Writing into the browser console, using console.log().

**Method 1 : Using innerHTML**

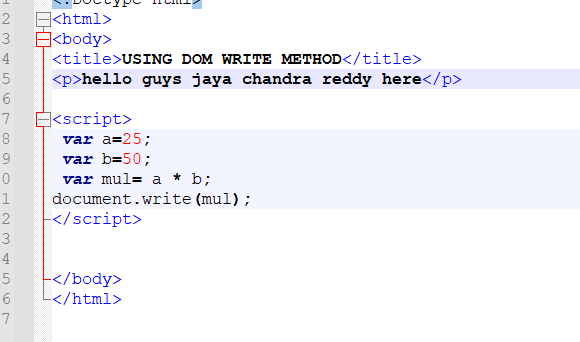
**Definition :**

The innerHTML property sets or returns the HTML content (inner HTML) of an element.

**Usage :**

To access an HTML element, JavaScript can use the document.getElementById(id) method. The id attribute defines the HTML element. The innerHTML property defines the HTML content.

SOURCE CODE



OUTPUT



**Method 2 : Using DOM Write**

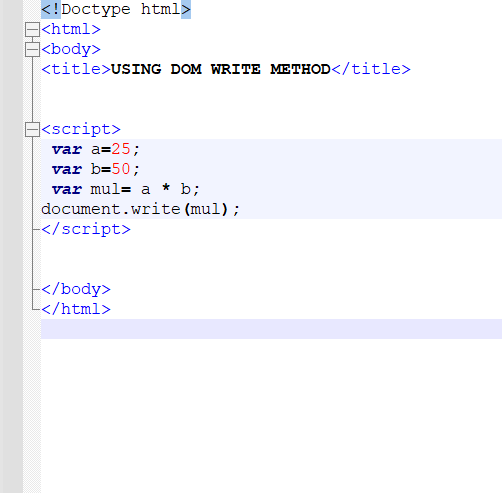
**Definition** : The write() method writes HTML expressions or JavaScript code to a document.

**Syntax** : document.write( exp1, exp2, exp3, ... )

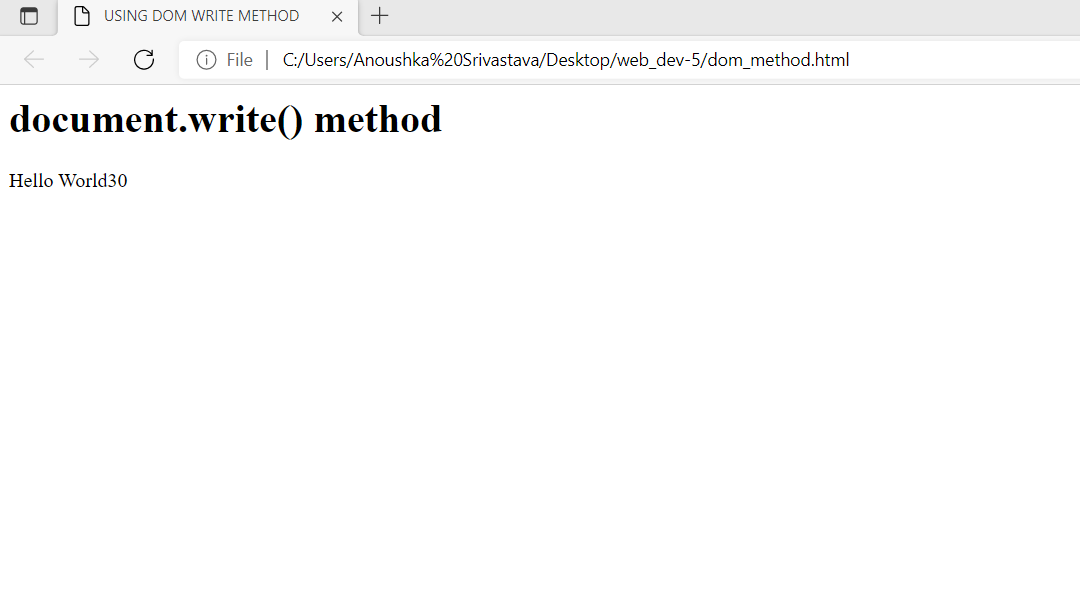
**Parameter Values**:

|  |  |
| --- | --- |
| Parameter | Description |
| exp1, exp2, exp3, ... | Optional. What to write to the output stream. Multiple arguments can be listed and they will be appended to the document in order of occurrence. |

SOURCE CODE



OUTPUT



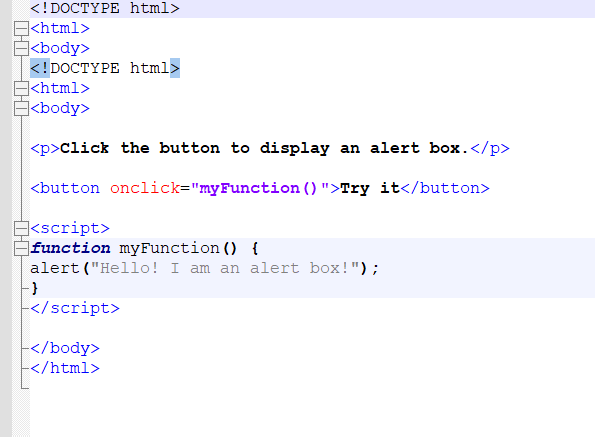
**Method 3 : Using alert**

**Definition** : The alert() method displays an alert box with a specified message and an OK button.

**Syntax :** window.alert(“SomeText”);

The window.alert(); method can be written without the window prefix.

SOURCECODE:



OUTPUT

**Method 4 : Using console.log**

**Definition :** The console.log() is a function in JavaScript which is used to print any kind of variables defined before in it or to just print any message that needs to be displayed to the user.

**Syntax** : console.log(A);

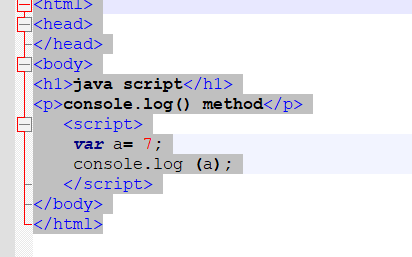
**Parameters** : It accepts a parameter which can be an array, an object or any message.

**Return Value :** It returns the value of the parameter given.

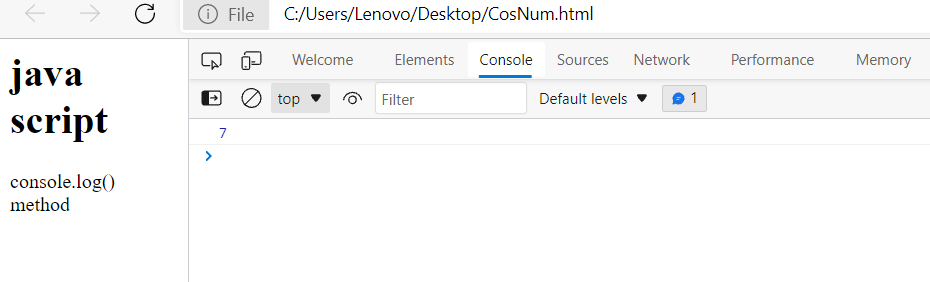
JavaScript codes to show the working of this function:

1. **Passing a number as an argument:**If the number is passed to the function console.log() then the function will display it.

SOURCE CODE

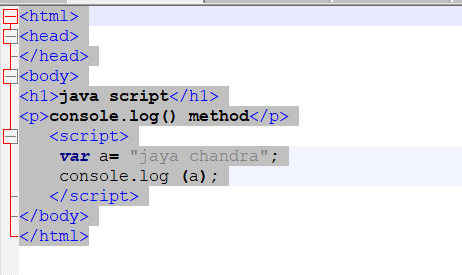


OUTPUT

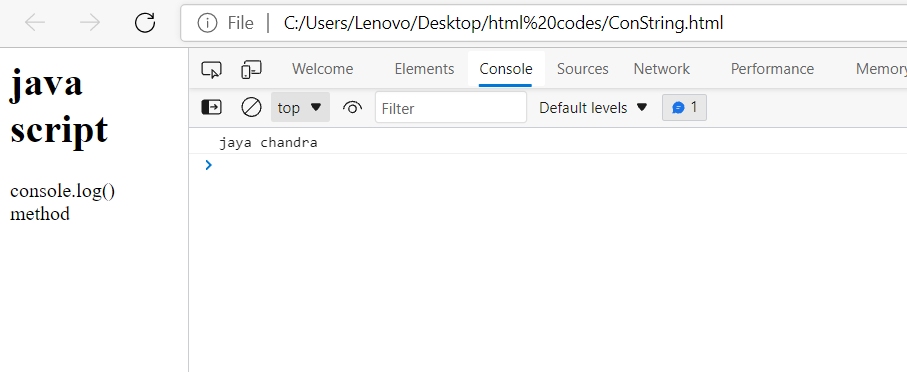


1. **Passing a string as an argument:**If the string is passed to the function console.log(), then the function will display it.

SOURCE CODE:

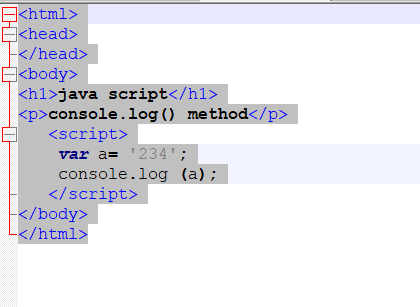
****

**Output :**

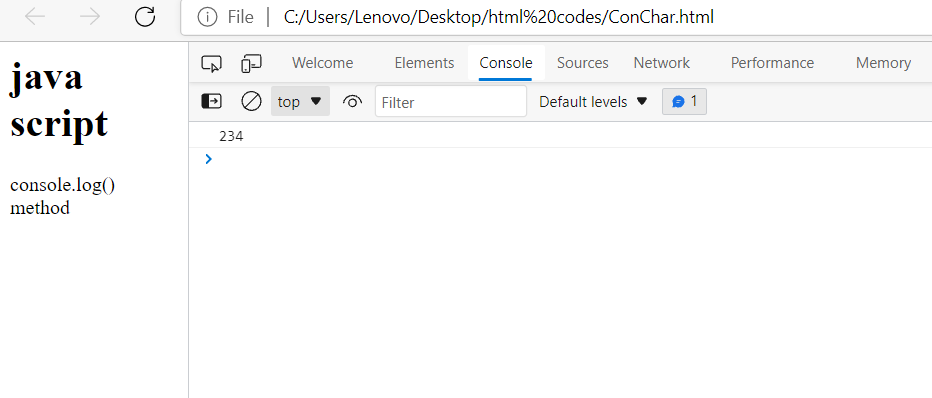


1. **Passing a char as an argument:**If the char is passed to the function console.log(), then the function will display it.

SOURCE CODE **:**

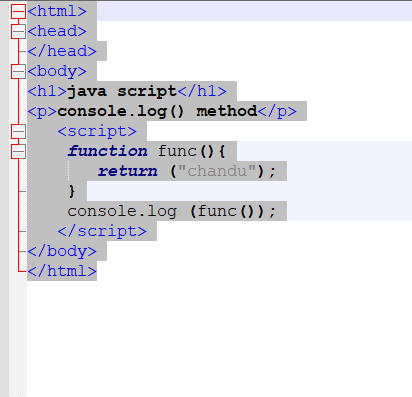


OUTPUT

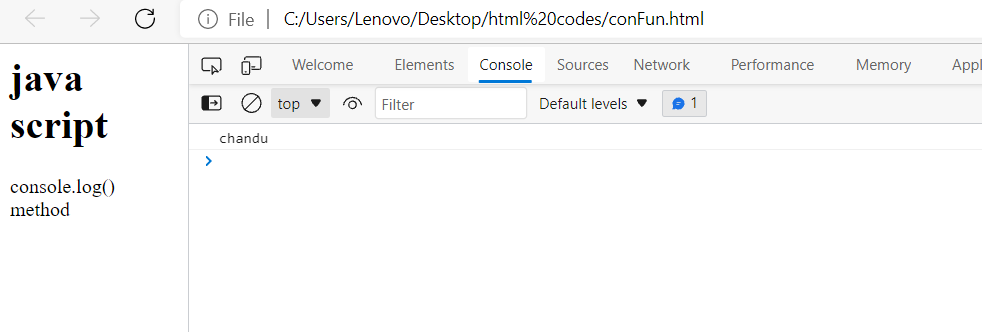


**4.Passing a function as an argument:**If the function is passed to the function console.log(), then the function will display the value of the passed function().

SOURCE CODE:

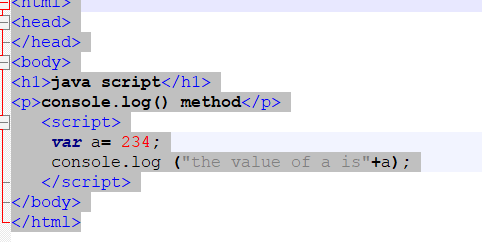


OUTPUT

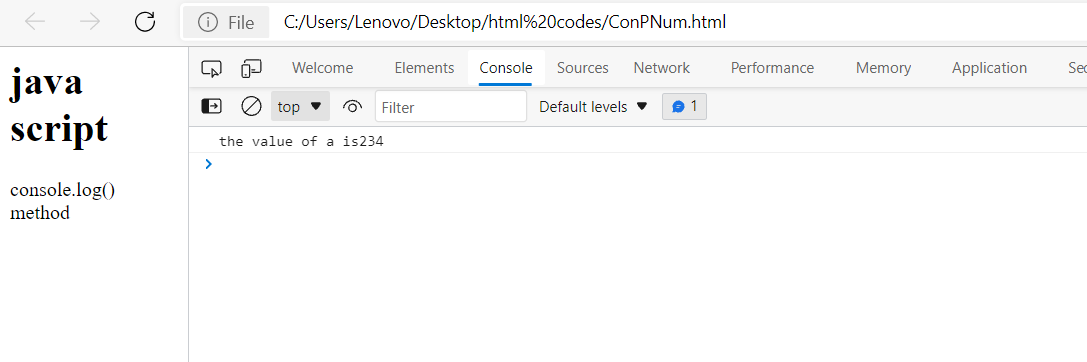


**5.Passing a number with message as an argument:**If the number is passed to the function console.log(), then the function will display it along with the given message.

SOURCE CODE

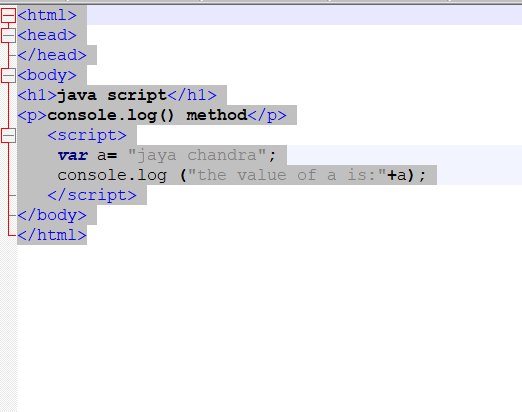


OUTPUT

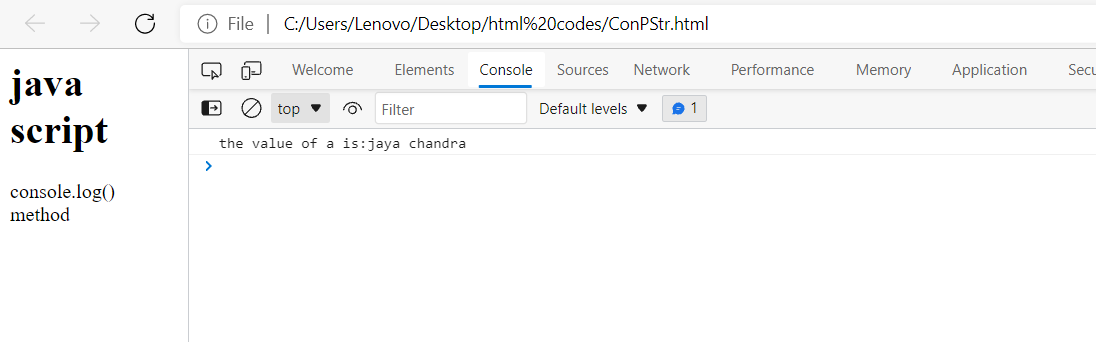


**6.Passing a string with message as an argument:**If the string is passed to the function console.log(), then the function will display it along with the given message.

**SOURCE CODE:**

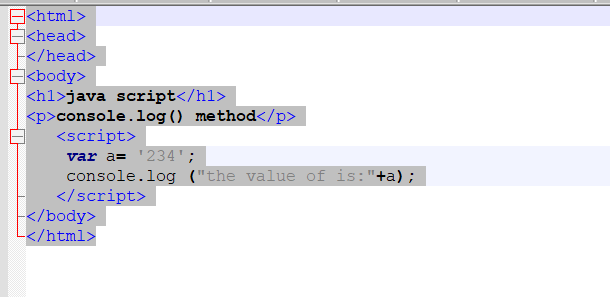
****

**OUTPUT :**

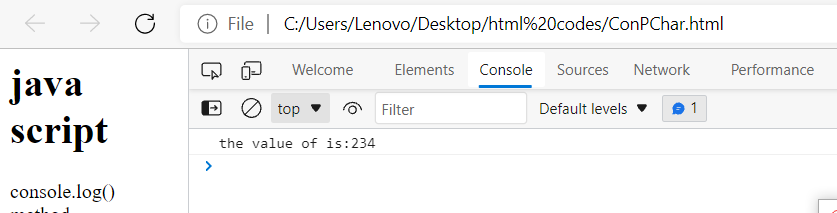


**7.Passing a char with message as an argument:** If the char is passed to the function console.log(), then the function will display it along with the given message.

SOURCE CODE:



OUTPUT

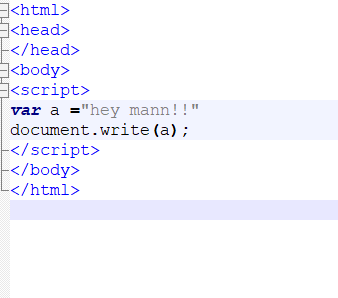


**5.There are two ways to include JavaScript in your HTML document.**

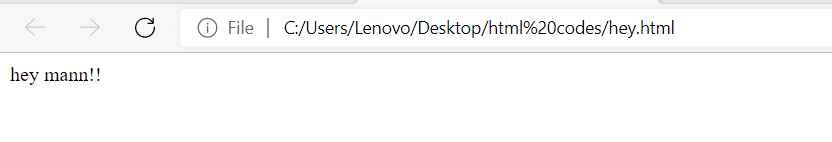
1. **Embedding javaScipt in your HTML document-**

You can embed the JavScript code directly within in your web page by placing it between the <script> and </script> tags. The <script> tag indicates the browser that the contained statements are to be interpreted as executable script and not HTML.

**SOURCE CODE:**



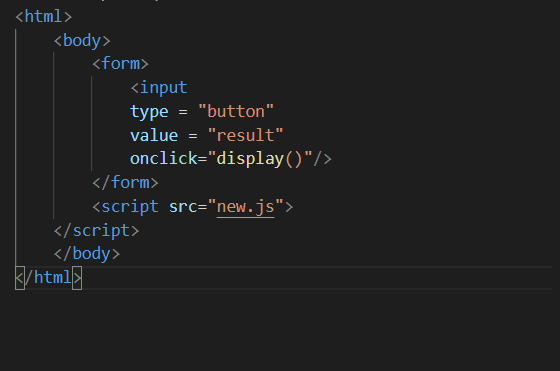
**Output :**



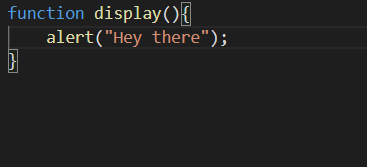
1. **Using External javaScript file and linking it in your HTML document-**

As you can see below there are two files – one the actual HTML document and other a separate external javascript file. The new.js file is linked in the HTML document using the script tag with the use of the attribute src as shown in the code below. When the javascript code is large, it is highly recommended to use a separate external file to keep the HTML doc and the JS code organised.

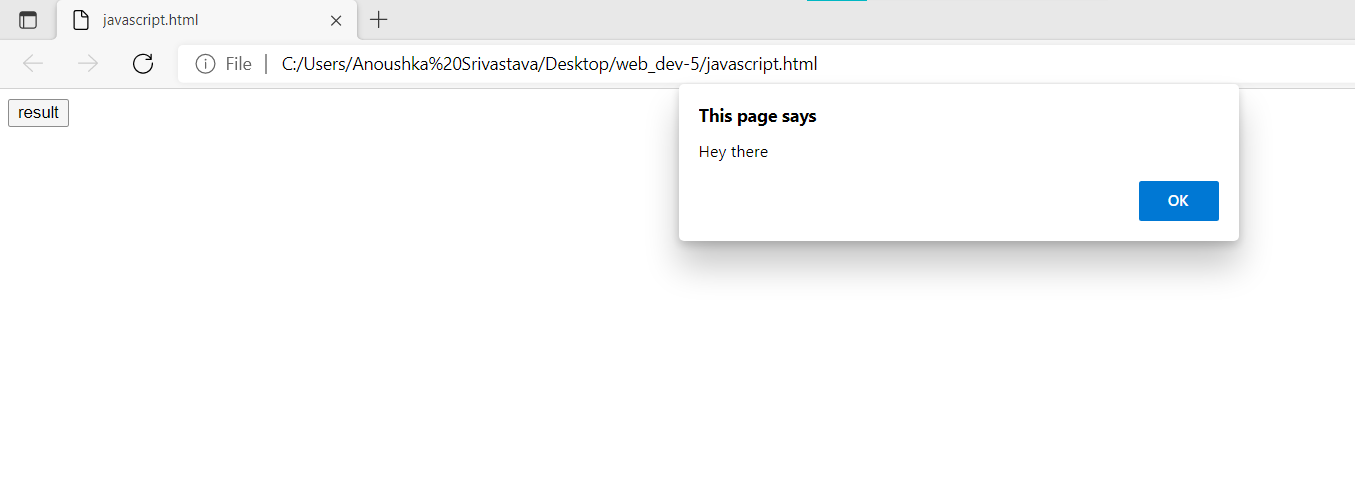
SOURCE CODE



**new.js :**



OUTPUT



**Experiment-2**

**AIM :** To implement conditional statements by using different looping constructs, JS date operations and JS timers.

**Basic Introduction :**

JavaScript includes operators same as other languages. An operator performs some operation on single or multiple operands (data value) and produces a result. There are many different types of operators.

JavaScript includes following categories of operators.

1. Arithmetic Operators
2. Comparison Operators
3. Logical Operators
4. Assignment Operators
5. String Operators

**1.Arithmetic Operators :** Arithmetic operators are used to perform mathematical operations between numeric operands.

**Program 1: WAP to perform different arithmetic operators**

SOURCE CODE

****

OUTPUT

****

**2. Comparison Operators**

JavaScript provides comparison operators that compare two operands and return a Boolean value true or false.

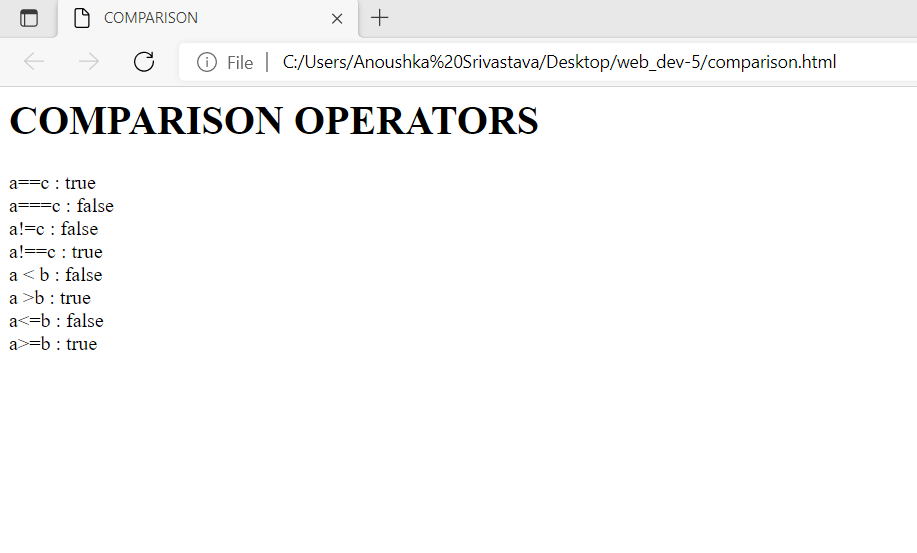
|  |  |
| --- | --- |
| **Operators** | **Description** |
| **==** | **Compares the equality of two operands without considering type.** |
| **===** | **Compares equality of two operands with type.** |
| **!=** | **Compares inequality of two operands.** |
| **>** | **Returns a Boolean value true if the left-side value is greater than the right-side value; otherwise, returns false.** |
| **<** | **Returns a Boolean value true if the left-side value is less than the right-side value; otherwise, returns false.** |
| **>=** | **Returns a Boolean value true if the left-side value is greater than or equal to the right-side value; otherwise, returns false.** |
| **<=** | **Returns a Boolean value true if the left-side value is less than or equal to the right-side value; otherwise, returns false.** |

**Program 2: WAP to perform different Comparison operators**

**OUTPUT**

****

**Output :**

****

**3. Assignment Operators**

JavaScript provides the assignment operators to assign values to variables with less key strokes.

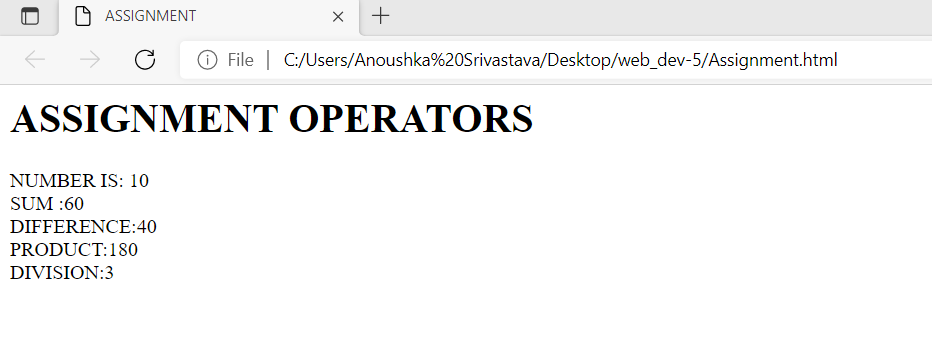
|  |  |
| --- | --- |
| Assignment operators | Description |
| = | Assigns right operand value to the left operand. |
| **+=** | Sums up left and right operand values and assigns the result to the left operand. |
| **-=** | Subtract right operand value from the left operand value and assigns the result to the left operand. |
| **\*=** | Multiply left and right operand values and assigns the result to the left operand. |
| **/=** | Divide left operand value by right operand value and assign the result to the left operand. |
| **%=** | Get the modulus of left operand divide by right operand and assign resulted modulus to the left operand. |

**Program 3: WAP to perform different Assignemnt operators**

**SOURCE CODE**

****

**OUTPUT**

****

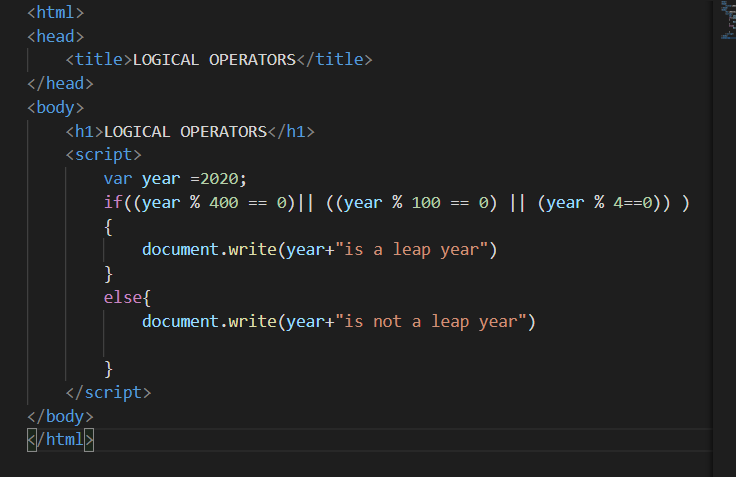
**4. Logical Operators**

In JavaScript, the logical operators are used to combine two or more conditions. JavaScript provides the following logical operators.

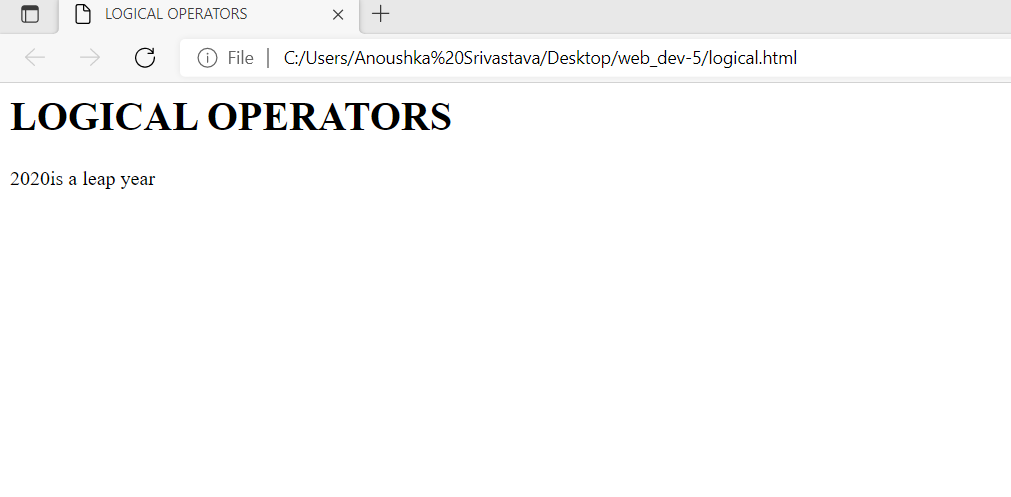
|  |  |
| --- | --- |
| **Operator** | **Description** |
| && | && is known as AND operator. It checks whether two operands are non-zero or not (0, false, undefined, null or "" are considered as zero). It returns 1 if they are non-zero; otherwise, returns 0. |
| || | || is known as OR operator. It checks whether any one of the two operands is non-zero or not (0, false, undefined, null or "" is considered as zero). It returns 1 if any one of them is non-zero; otherwise, returns 0. |
| ! | ! is known as NOT operator. It reverses the Boolean result of the operand (or condition). !false returns true, and !true returns false. |

**Program 3: WAP to perform different Logical operators**

**SOURCE CODE**

****

**OUTPUT**

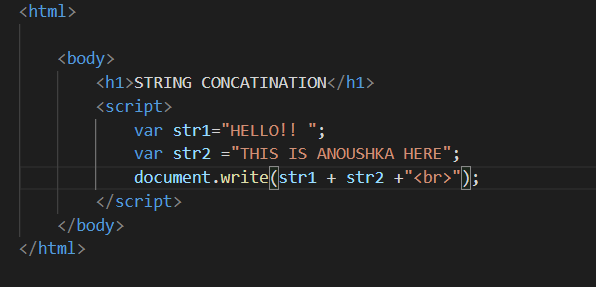
****

**5. String Operator**

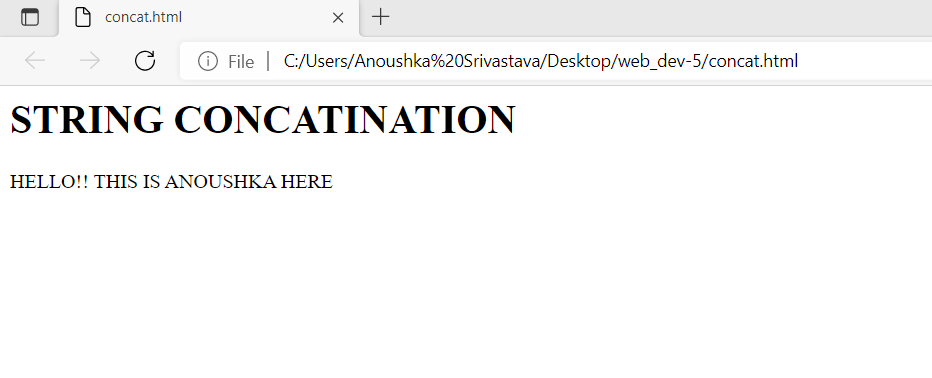
The + operator performs concatenation operation when one of the operands is of string type.

**Program 5: Write a program to perform different string operators in JS (Combining the strings).**

**Code :**

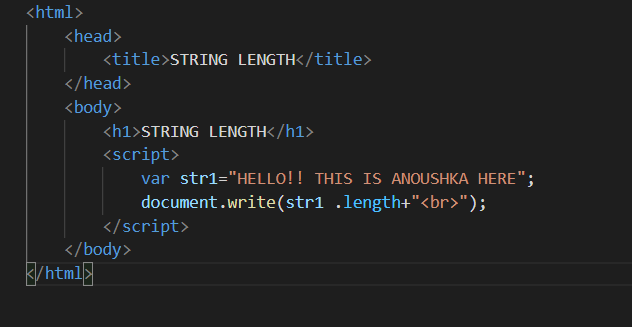


**Output :**

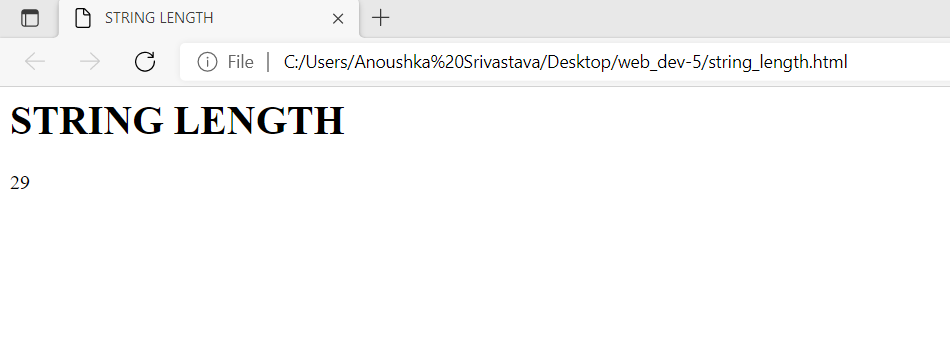
****

**Program 6: Getting the length of a string**

**Code :**

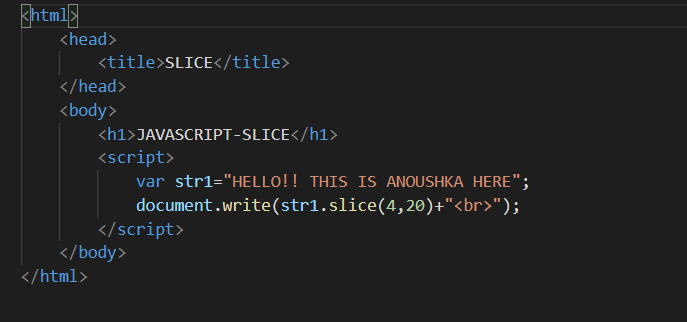


**Output :**

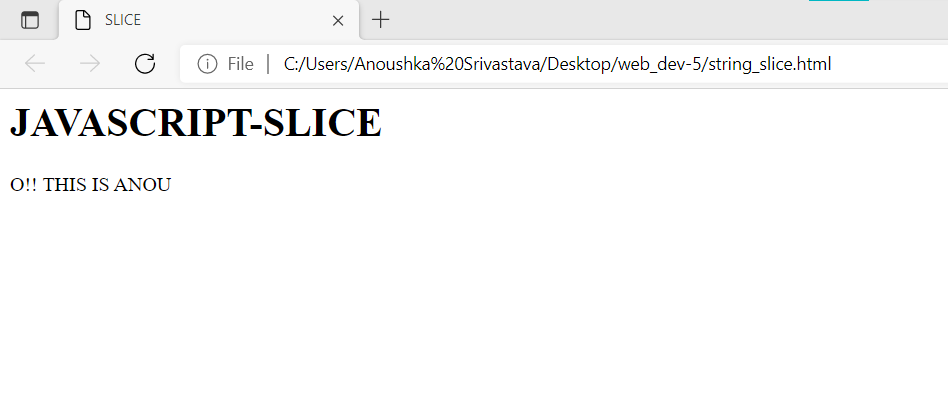
****

**Program 7: WAP to Slice out a portion of a string.**

**Code :**

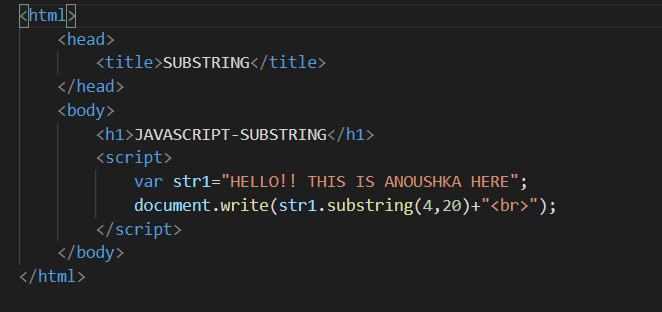


**Output :**

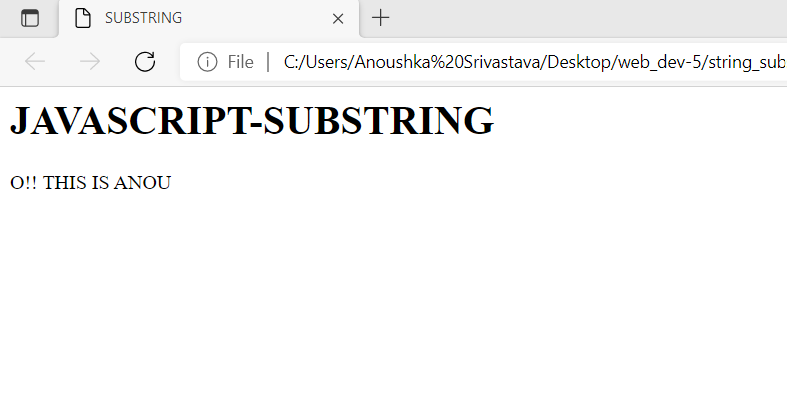
****

**Program 9: WAP to Extract substring from a String.**

**Code :**

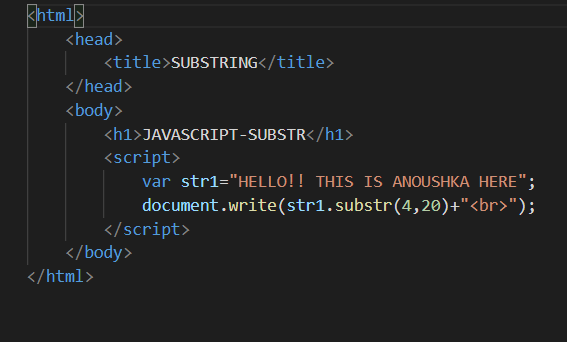


**Output :**

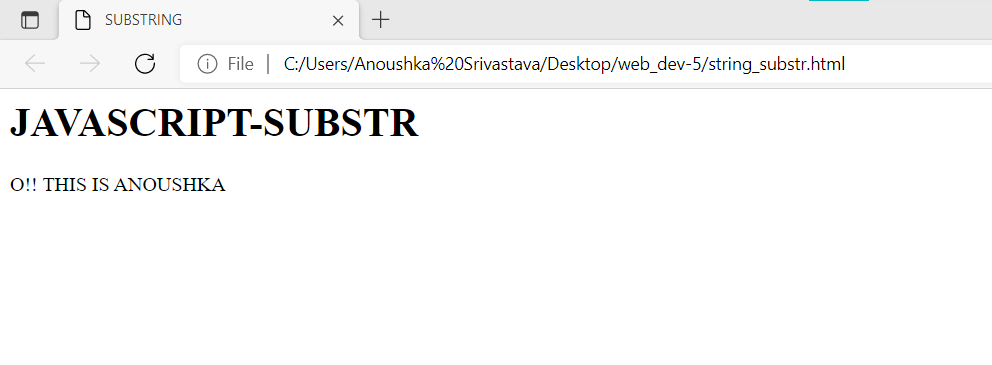
****

**Program 10: WAP to Extract Fixed Number of Characters from a String.**

**Code :**



**Output :**

****

**Experiment 3**

**AIM:** To implement conditional statements by using different looping constructs, JS date operations and JS timers.

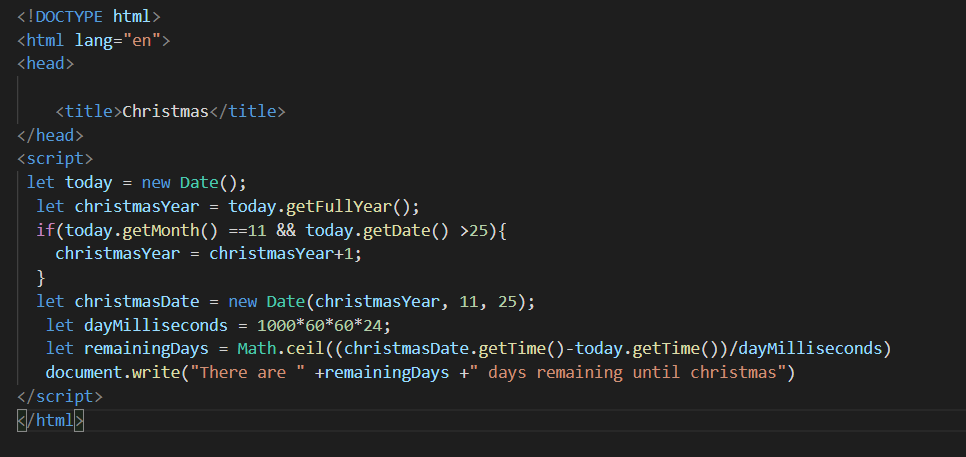
**Basic Introduction :**

Loops are used to execute the same block of code again and again, as long as a certain condition is met. The basic idea behind a loop is to automate the repetitive tasks within a program to save the time and effort. JavaScript now supports five different types of loops:

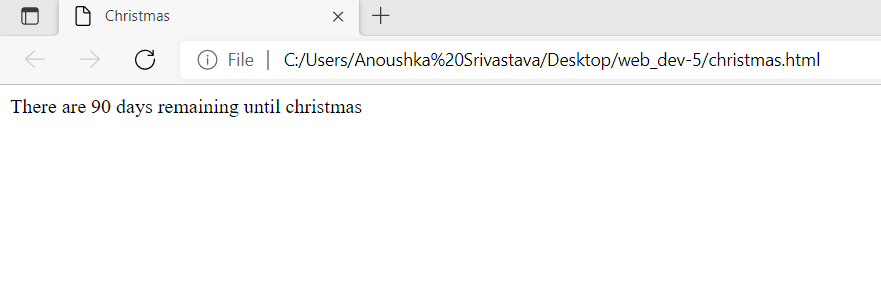
* **while** — loops through a block of code as long as the condition specified evaluates to true.
* **do…while** — loops through a block of code once; then the condition is evaluated. If the condition is true, the statement is repeated as long as the specified condition is true.
* **for** — loops through a block of code until the counter reaches a specified number.
* **for…in** — loops through the properties of an object.
* **for…of** — loops over iterable objects such as arrays, strings, etc.

**Program 1 : Predicting how many days remaining until next Christmas.**

**Code :**

****

**Output :**

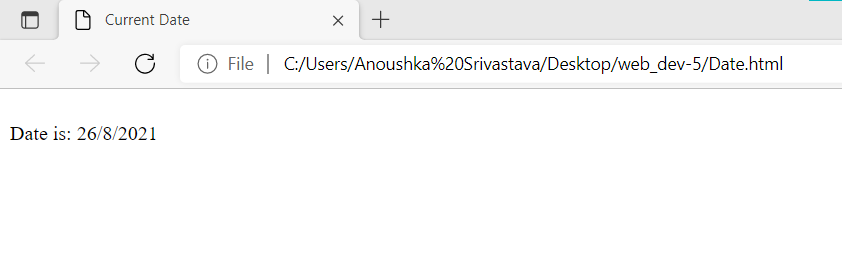
****

**Program 2 : Write a program to print the current date.**

**Code :**

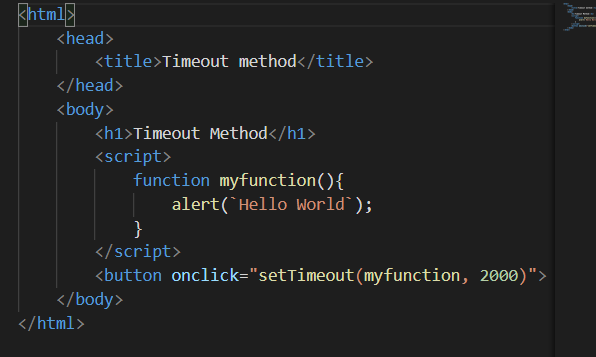
****

**Output :**

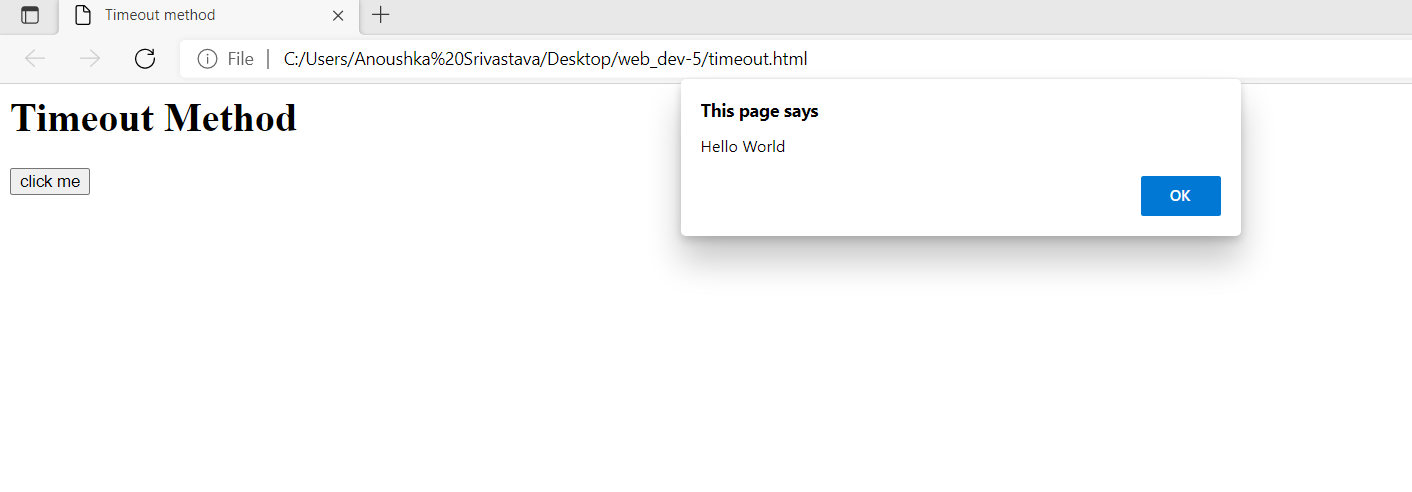
****

**Program 3 : Write a program to print Hello world using set timeout method.**

**Code :**

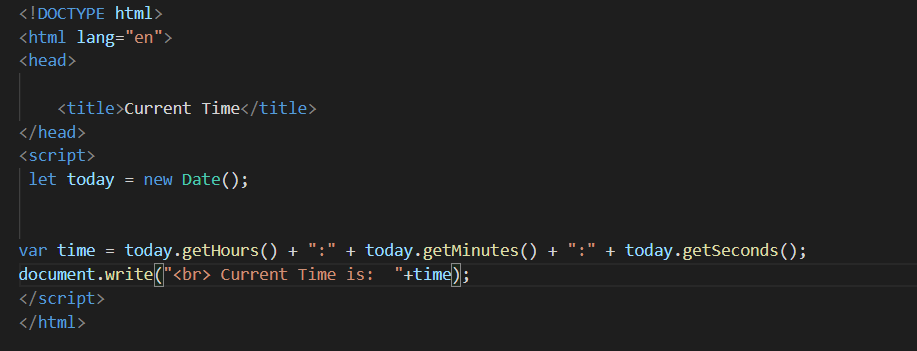
****

**Output :**

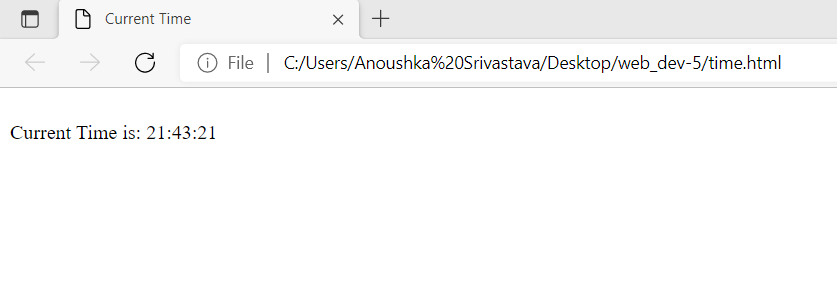
****

**Program 4 : Write a program to print current time of system.**

**Code :**

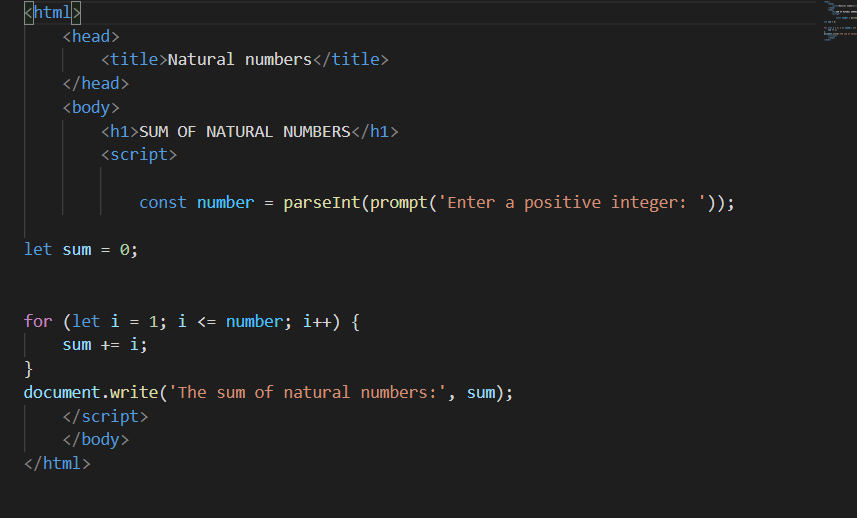
****

**Output :**

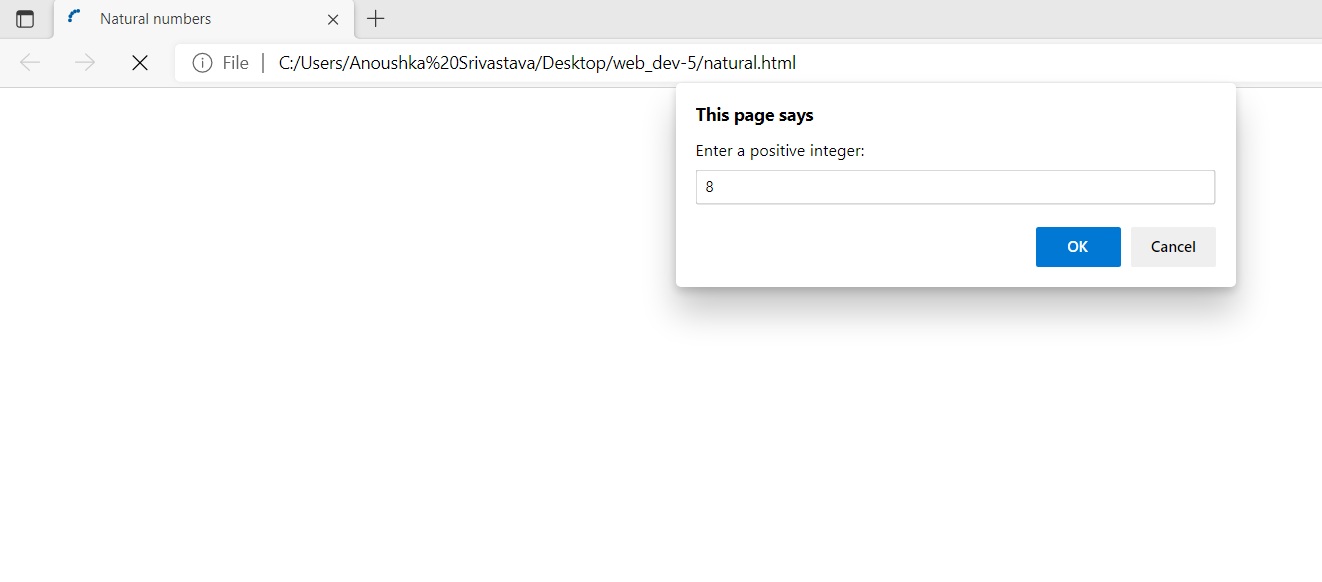
****

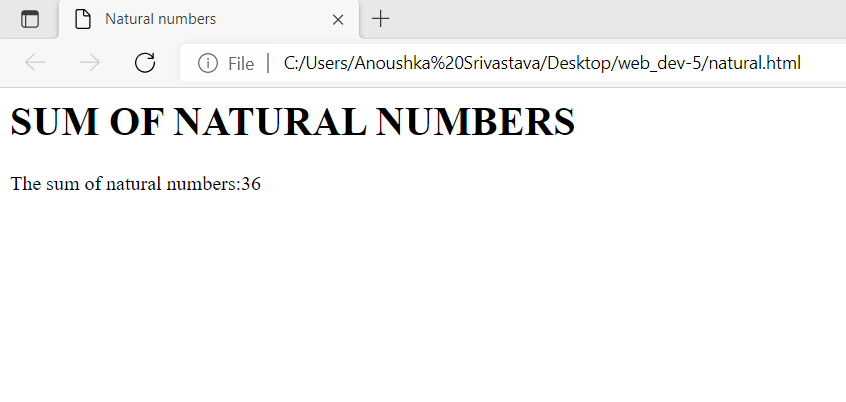
**Program 5 : Write a program to print the sum of natural numbers.**

**Code :**

****

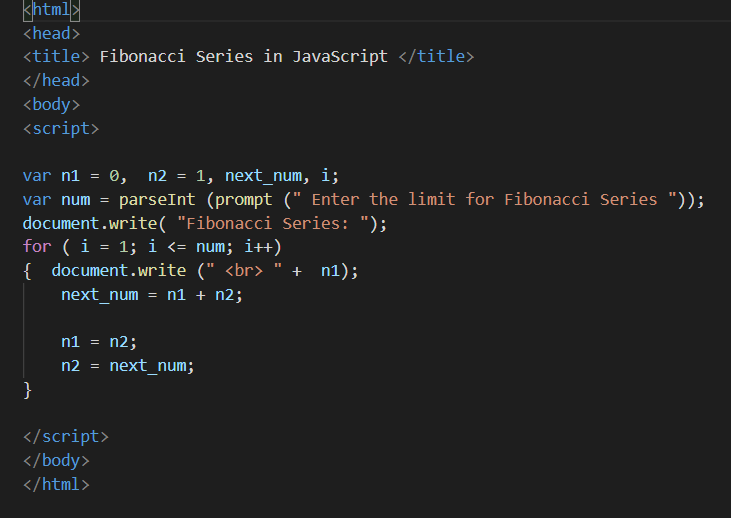
**Output :**

****

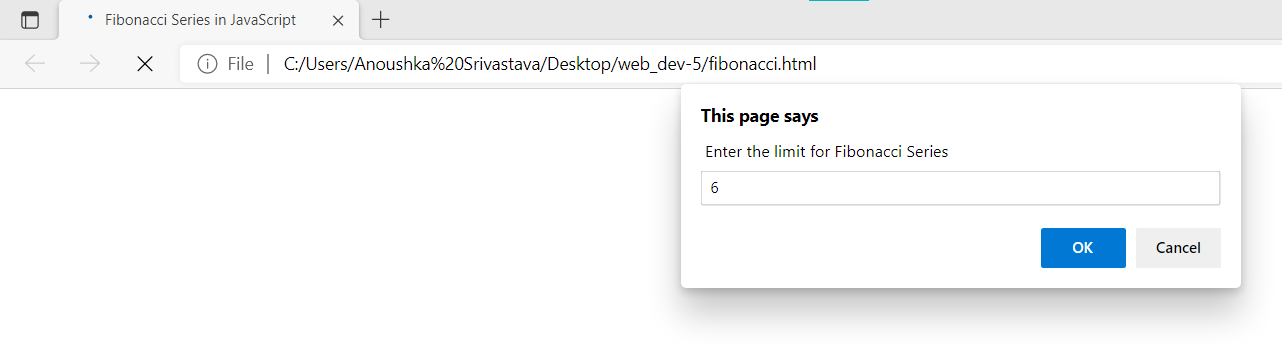
****

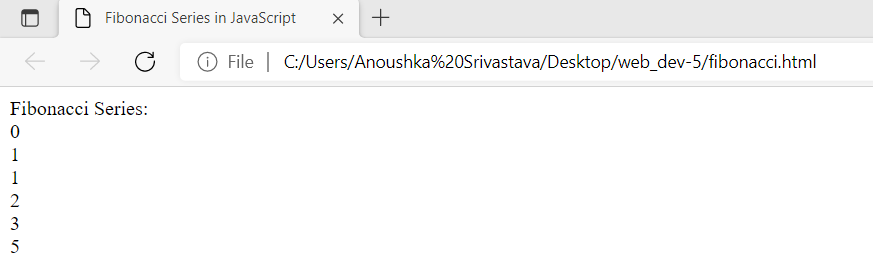
**Program 6 : Write a program to print Fibonacci series up to n terms.**

**Code :**



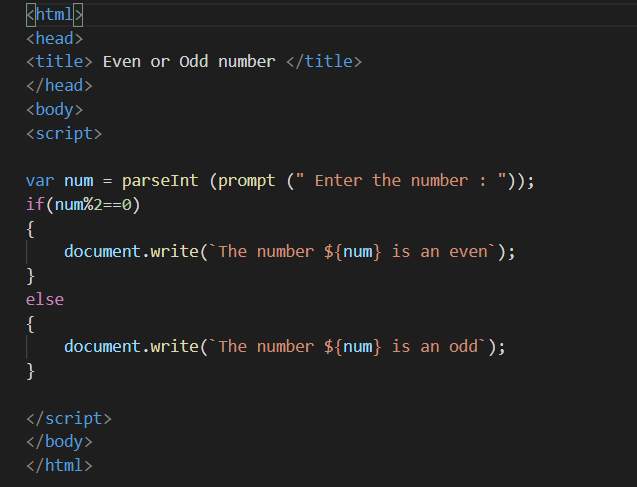
**Output :**

****

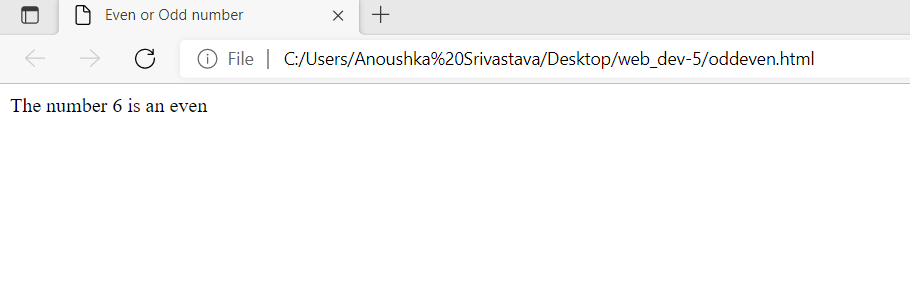
****

**Program 7 : Write a program to check whether a number is even or odd.**

**Code :**



**Output :**

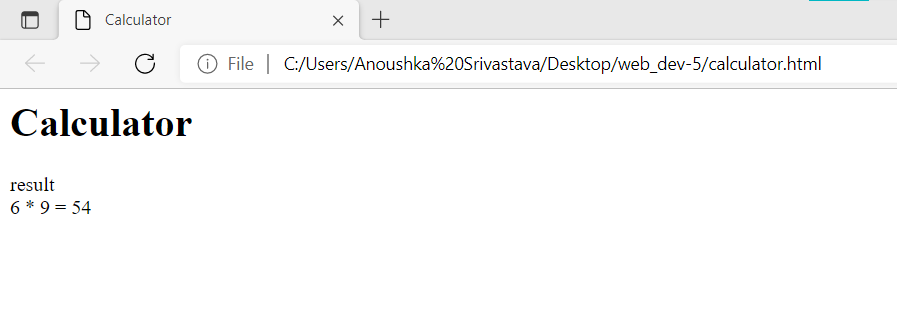
****

**Program 8 : Write a program to design a simple calculator using switch case.**

**Code :**



**Output :**

****

**Experiment 4**

**AIM :** To perform the implementation of JS arrays and its associated methods.

**Basic Introduction :**

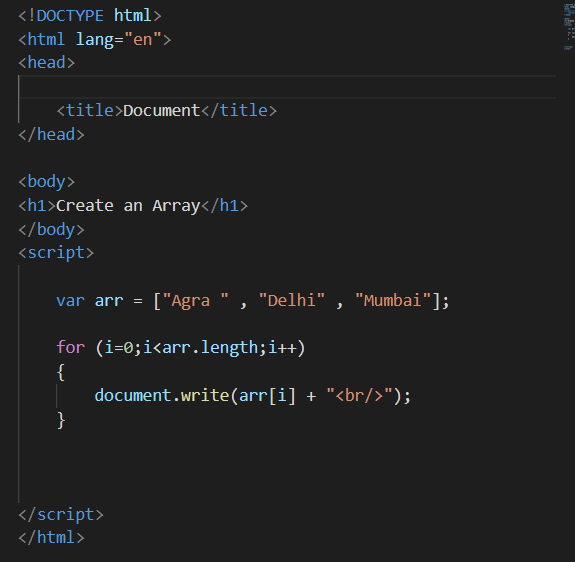
The Array object is used to store multiple values in a single variable. Array indexes are zero-based: The first element in the array is 0, the second is 1, and so on.

Some basic JavaScript array methods are :

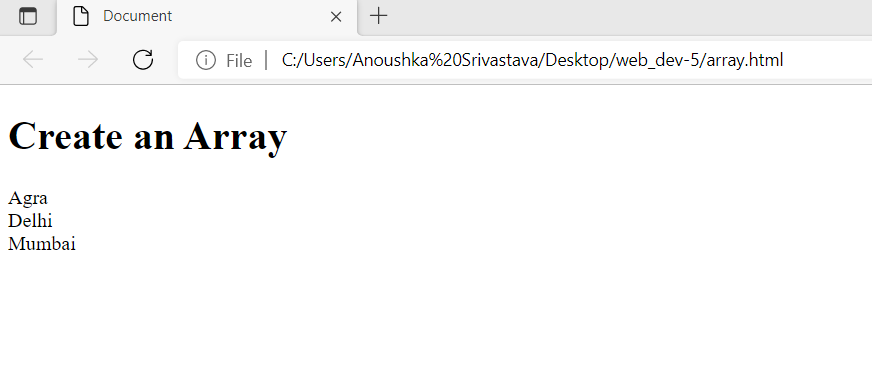
|  |  |
| --- | --- |
| **Method** | **Description** |
| **Array.push()** | To add elements to the end of the array. |
| **Array.pop()** | To remove elements from the end of the array. |
| **Array.unshift()** | To add elements to the front of the array |
| **Array.shift()** | To remove elements from the front of the array. |
| **Array.splice()** | To add or remove elements from the splice |

**Program 1 : Write a program to create an array in JS using array literal.**

**Code :**

****

**Output :**

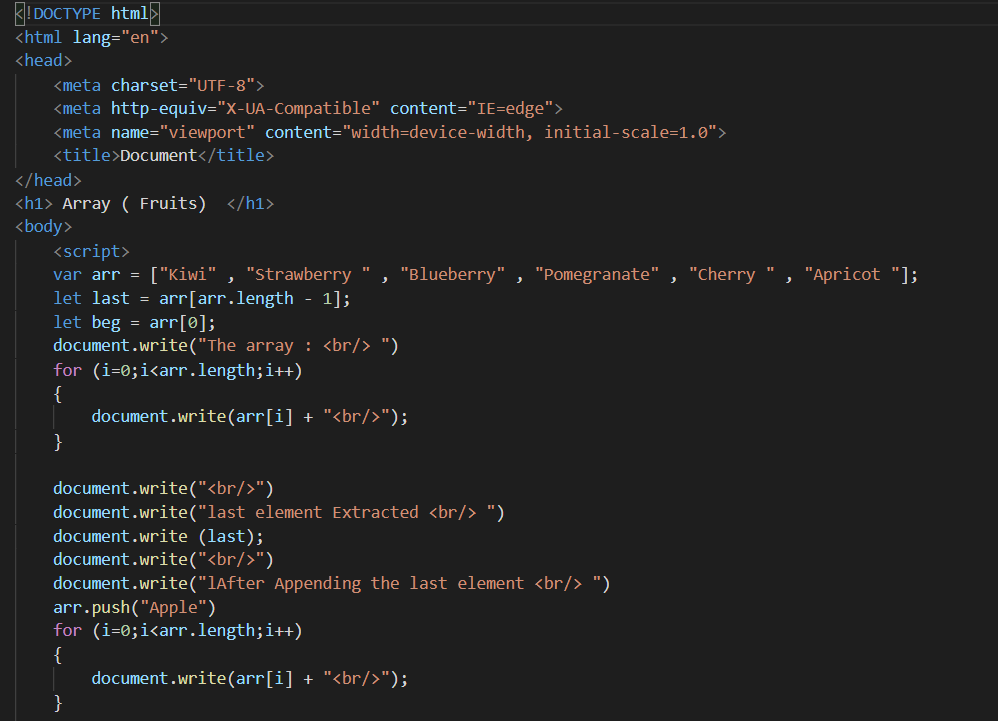


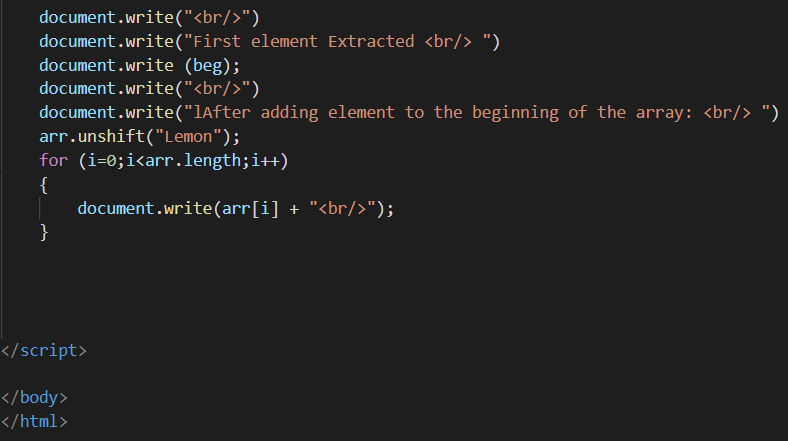
**Program 2 : To create a JS initializing arrays of fruits and display the same.**

**Performing the following:**

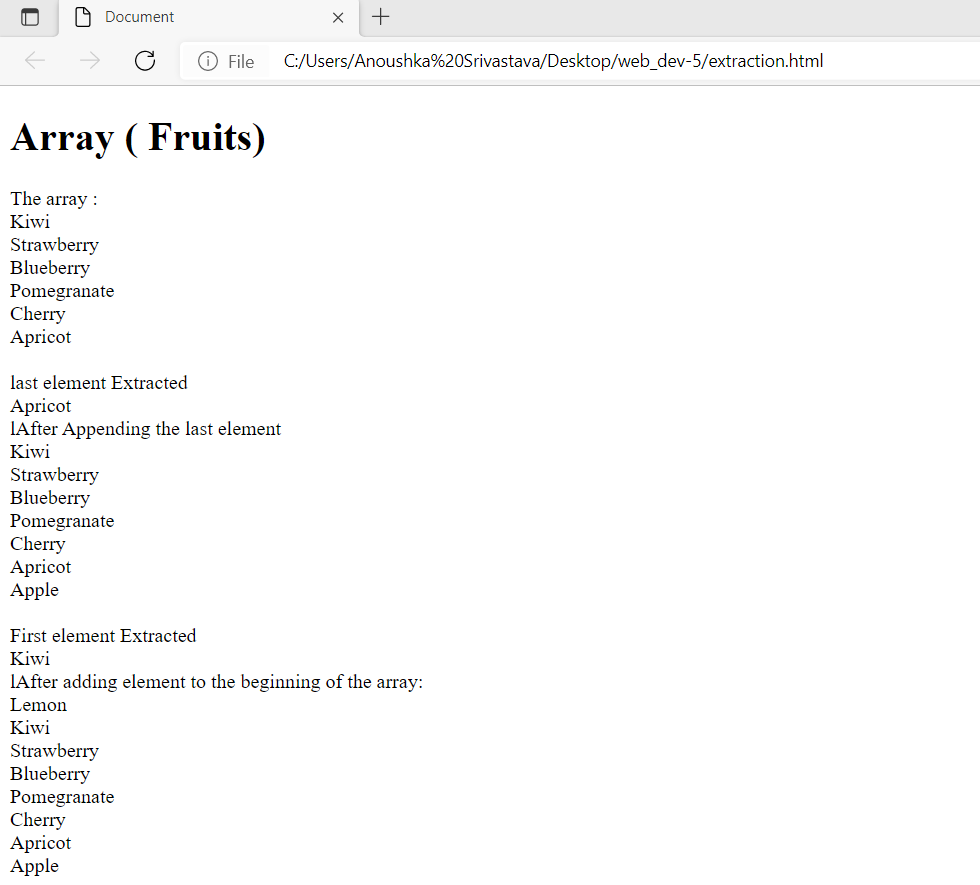
* **Extracts the last element of the array and return the same.**
* **Append the element of the array and returns it.**
* **Extracts the first element of the array and returns it.**
* **Add the element to the beginning of the array.**

**Code :**

****

****

**Output :**

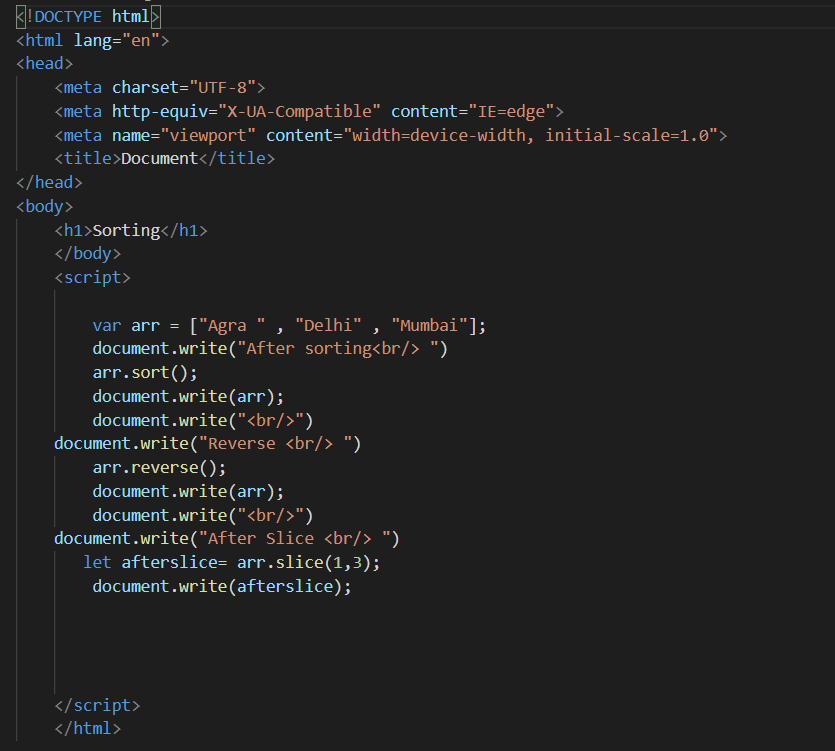


**Program 3 : Write a program to sort the array.**

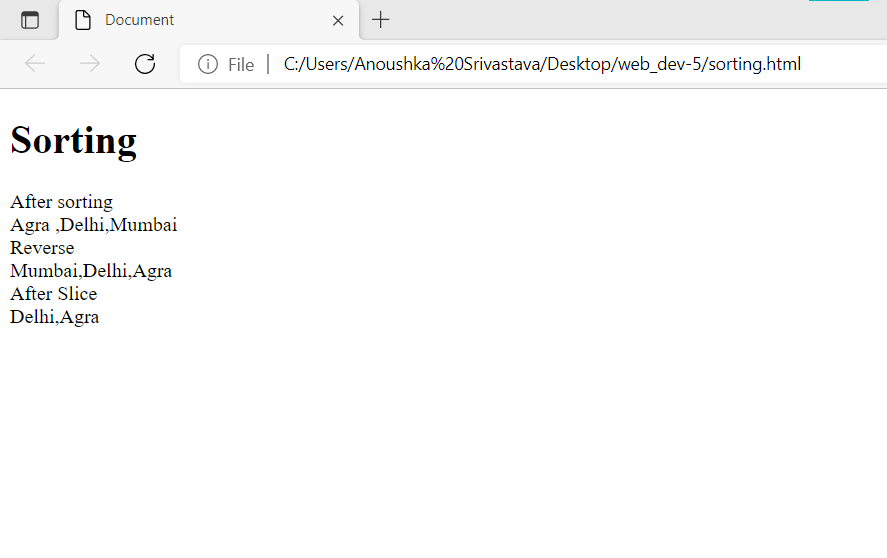
**Performing the following:**

* **Sort the array.**
* **Reverse the array.**
* **Perform the slice operation of the array.**

**Code :**

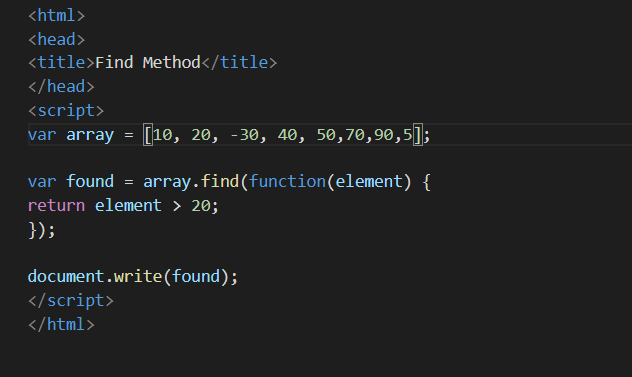
****

**Output :**

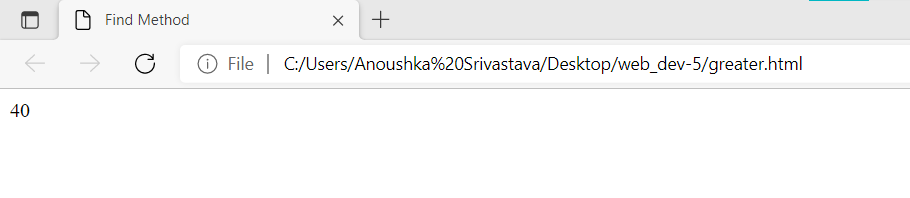


**Program 4 : Write a JS program the returns the value of the first element in the array which is greater than 20.**

**Code :**

****

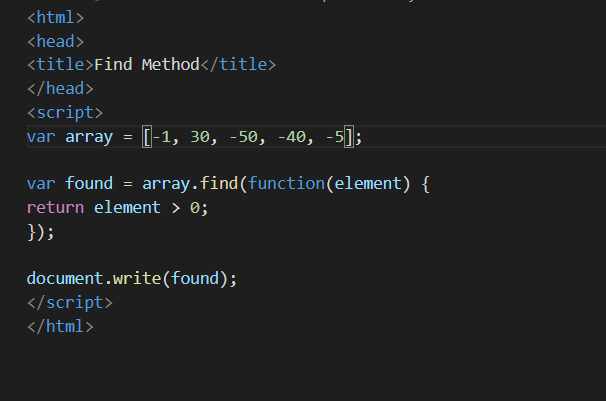
**Output :**



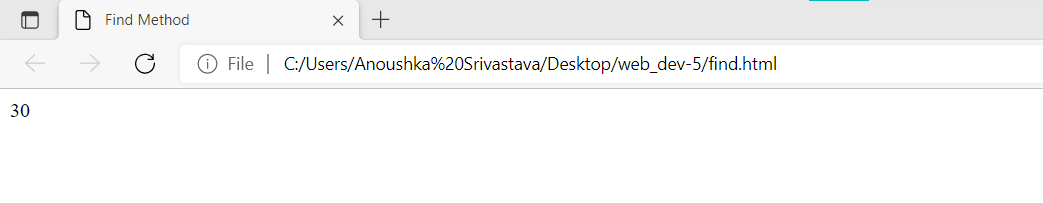
**Program 5 : Write a program that considers the following var array = -1, -30, 50, -40, -5}**

**Use the array find() method to find a positive number.**

**Code :**

****

**Output :**

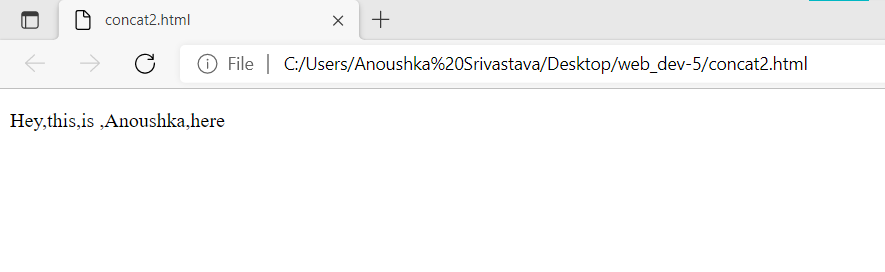


**Program 6 : Write a program to create an array in JS using array literal.**

**Code :**

****

**Output :**



**Experiment 5**

**AIM :** To implement JS programs using functions and to implement some function: Ceil, Random, Min-max.

**Basic Introduction :**

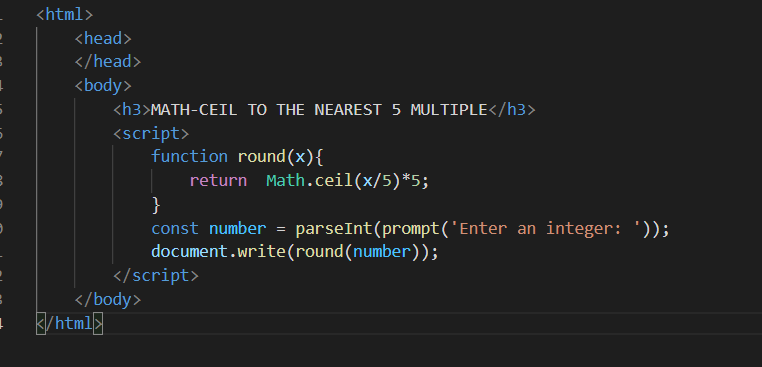
* **Math.ceil():-** The Math.ceil() function always rounds a number up to the next largest integer.

**Note:** Math.ceil(null) returns integer 0 and does not give a NaN error.

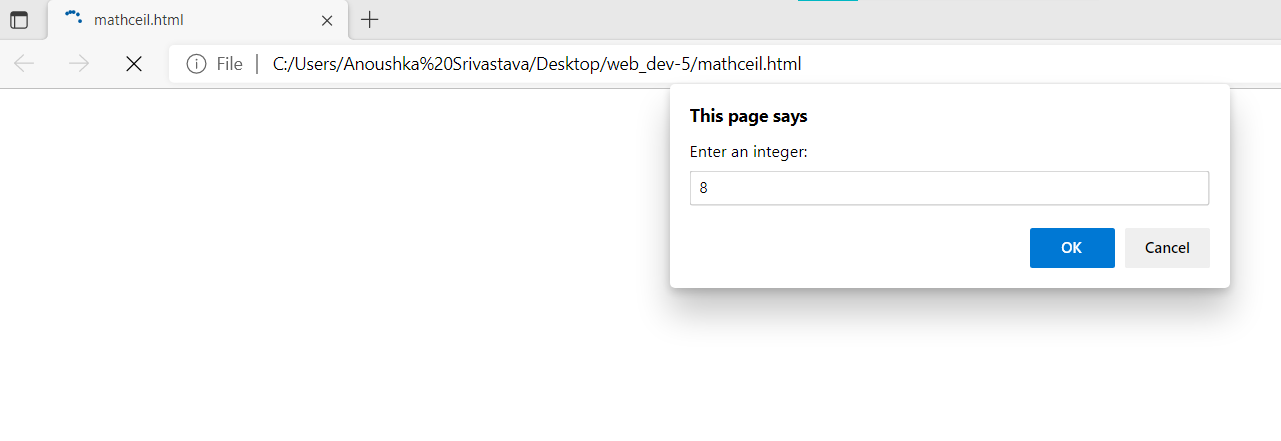
* **Math.random():-** The Math.random() function returns a floating-point, pseudo-random number in the range 0 to less than 1 (inclusive of 0, but not 1) with approximately uniform distribution over that range — which you can then scale to your desired range. The implementation selects the initial seed to the random number generation algorithm; it cannot be chosen or reset by the user.
* **Math.floor():-** The Math.floor() function returns the largest integer less than or equal to a given number.

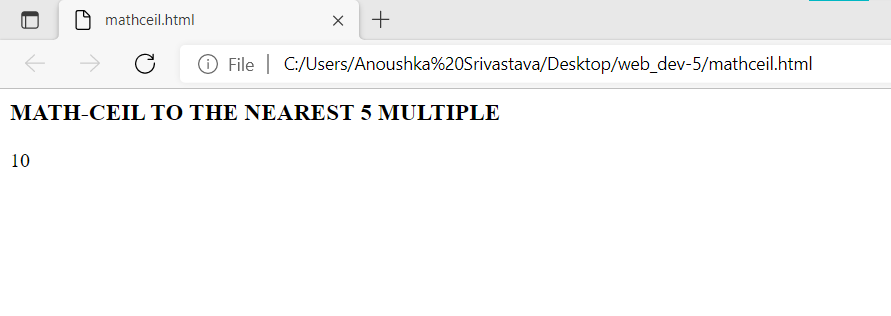
**Program 1 : Write a JS program to** **round off a number to the next multiple of 5.**

**Code :**

****

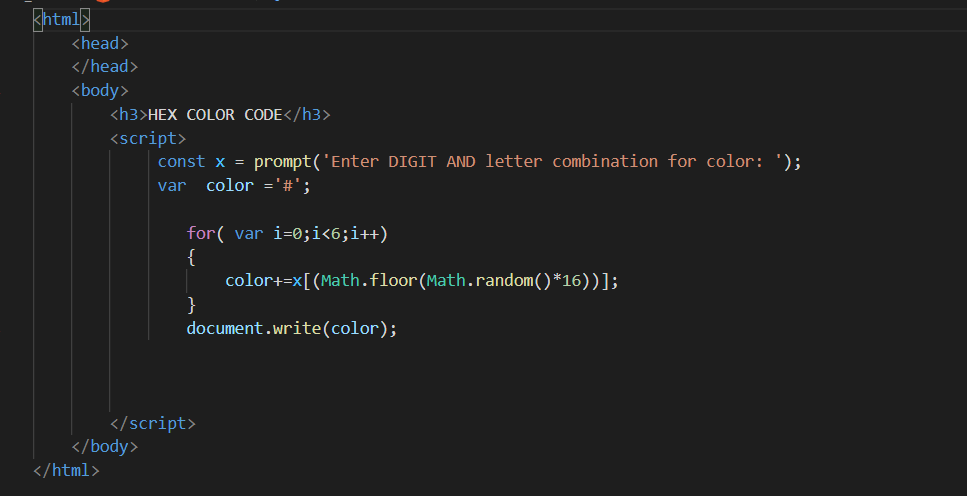
**Output :**



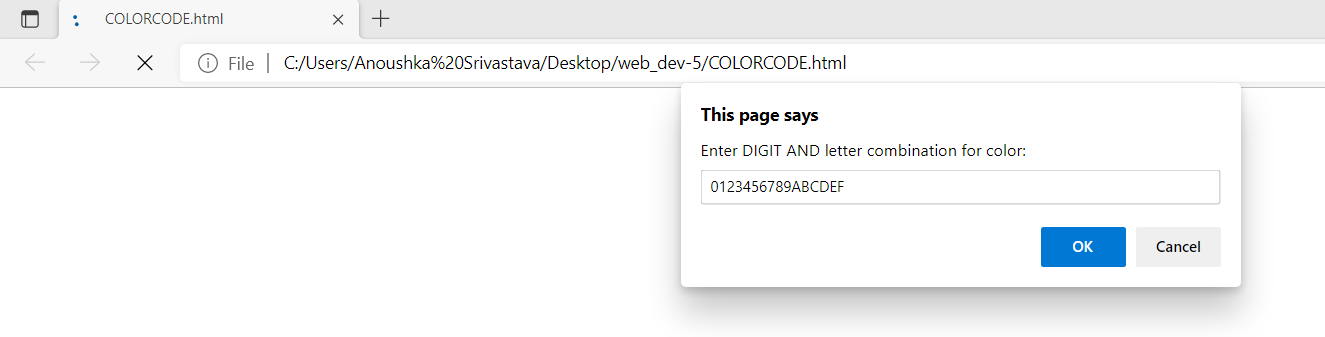


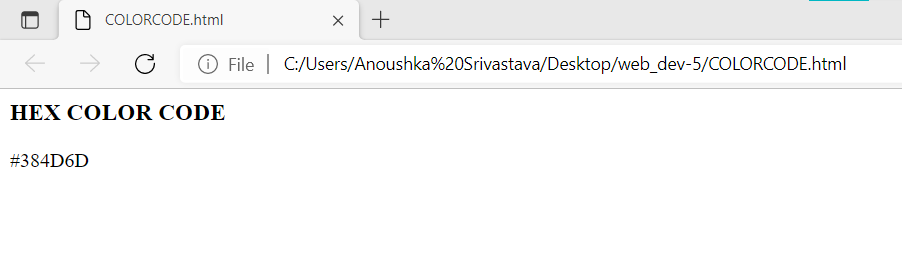
**Program 2 : Write a JS program to generate random hex codes of color.**

**Code :**

****

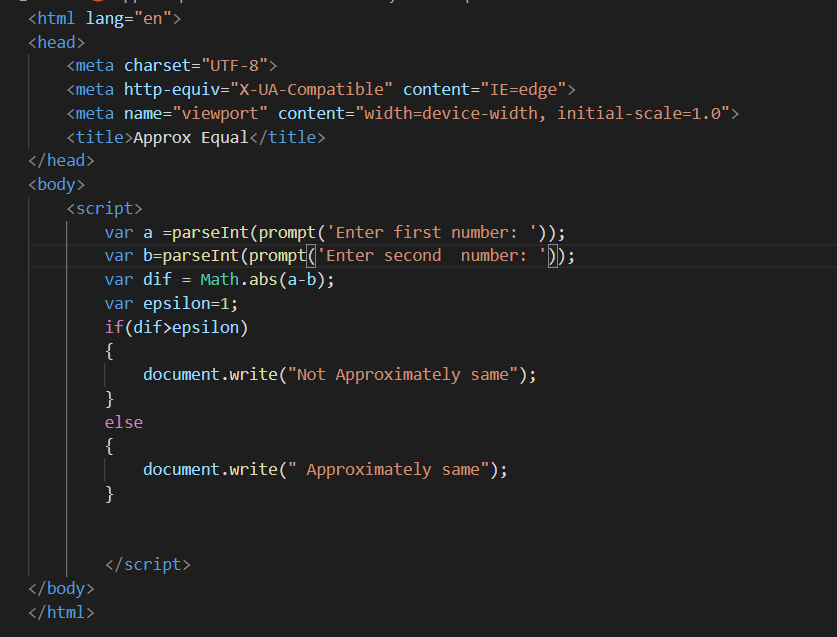
**Output :**



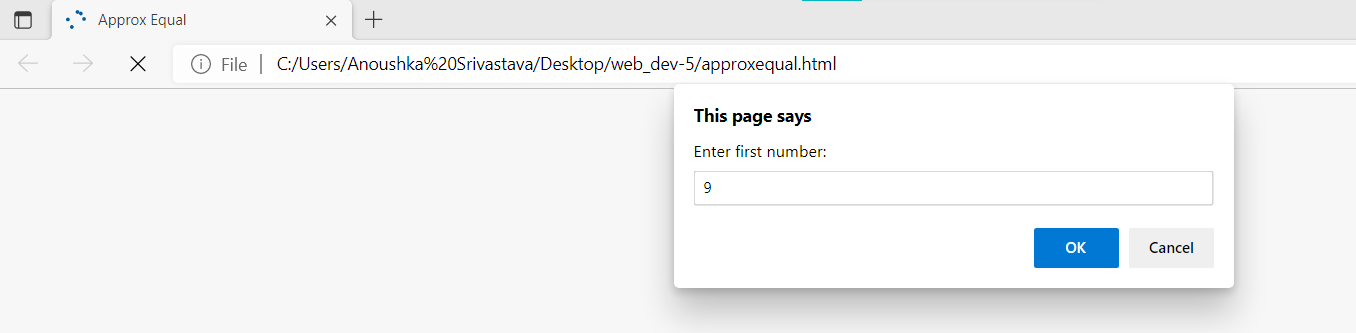


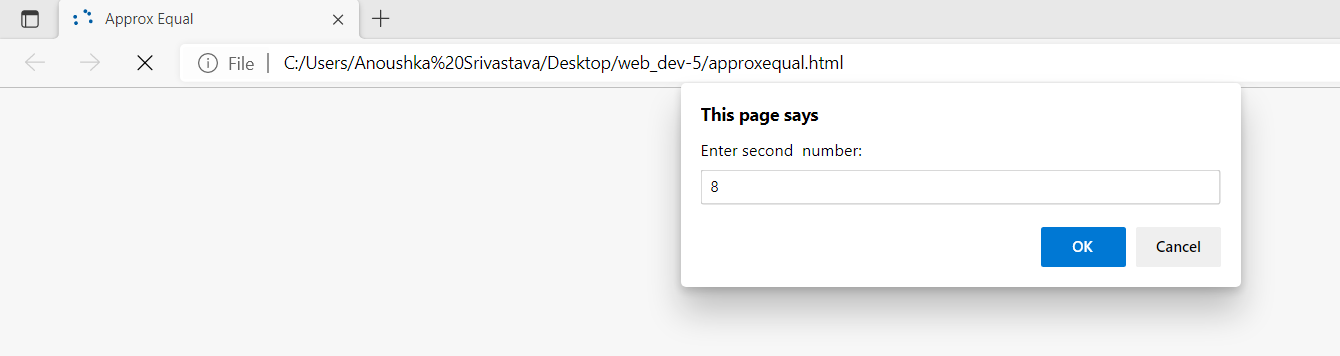
**Program 3 : Write a JS program to check two numbers are approximately equal in JavaScript.**

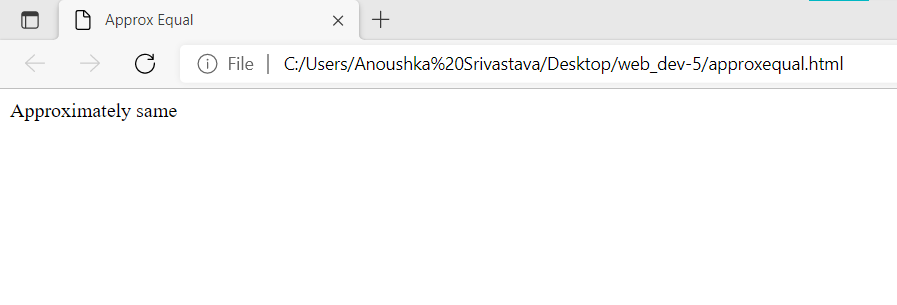
**Code :**

****

**Output :**

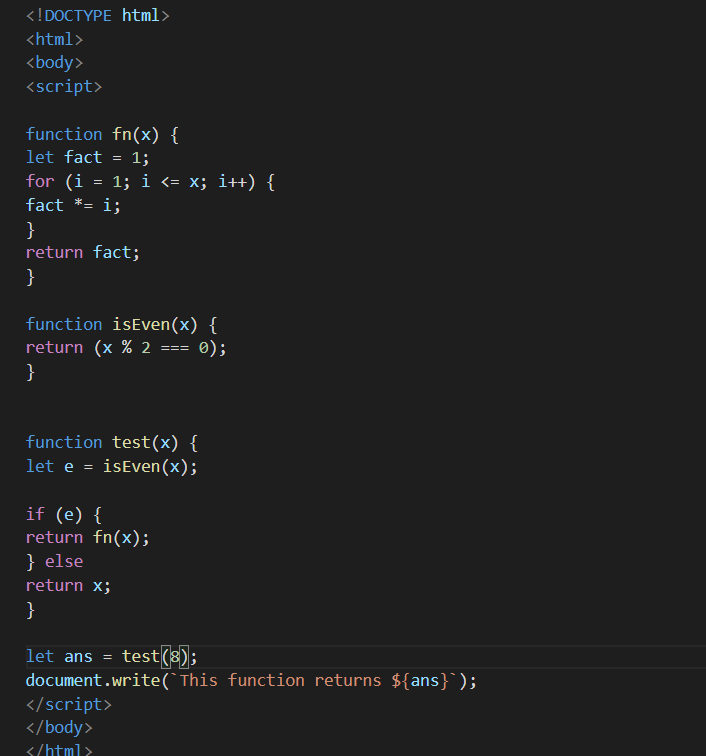




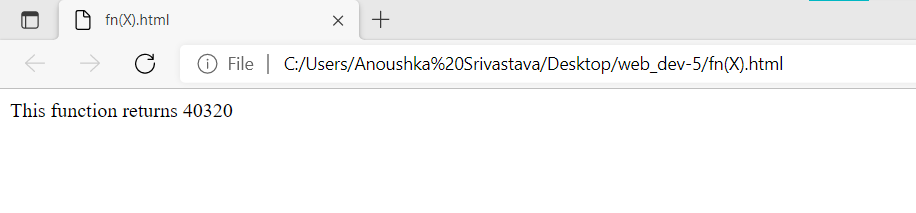


**Program 4 : Write a JS program to test a value x against predicate function and returns fn(x) or x in JavaScript.**

**Code :**

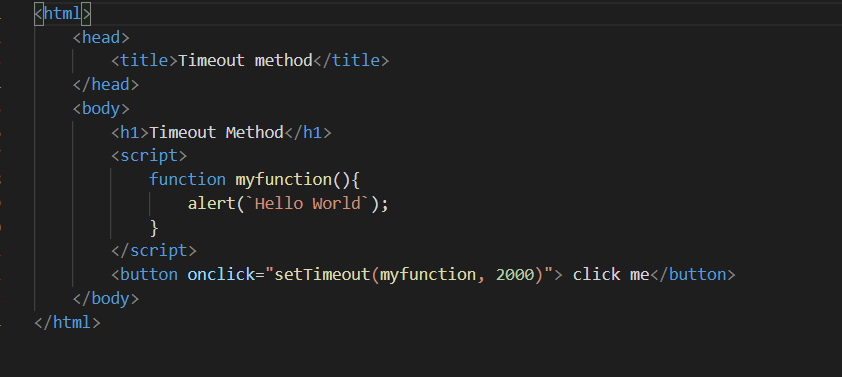
****

**Output :**

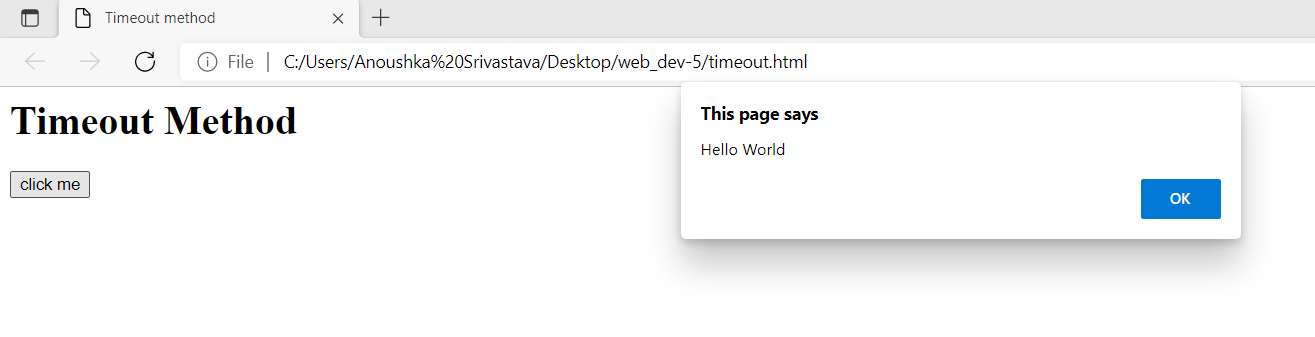


**Program 5 : Write a JS program to implement a function that enable another function after specified time.**

**Code :**

****

**Output :**

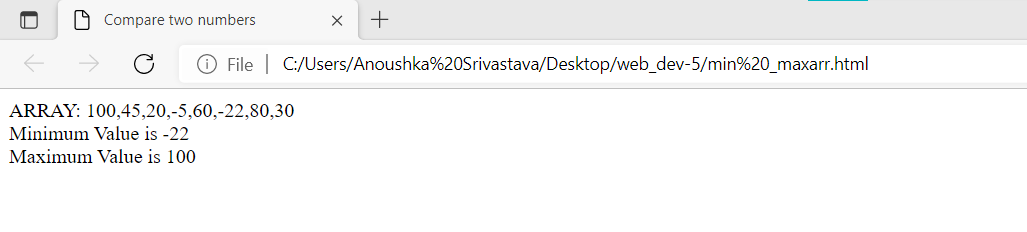


**Program 6 : Write a program to find minimum and maximum in an array.**

**Code :**

****

**Output :**

****

**Experiment 6**

**AIM :** Understanding different output format in JS.

**Basic Introduction :**

**Selecting the Topmost Elements**

The topmost elements in an HTML document are available directly as document properties. For example, the [<html>](https://www.tutorialrepublic.com/html-reference/html-html-tag.php) element can be accessed with document.documentElement property, whereas the [<head>](https://www.tutorialrepublic.com/html-reference/html-head-tag.php) element can be accessed with document.head property, and the [<body>](https://www.tutorialrepublic.com/html-reference/html-body-tag.php) element can be accessed with document.body property.

**Setting Inline Styles on Elements**

Inline styles are applied directly to the specific HTML element using the style attribute. In JavaScript the style property is used to get or set the inline style of an element.

**Navigating Between DOM Nodes**

DOM node provides several properties and methods that allow you to navigate or traverse through the tree structure of the DOM and make changes very easily. In the following section we will learn how to navigate up, down, and sideways in the DOM tree using JavaScript.

You can use the firstChild and lastChild properties of the DOM node to access the first and last direct child node of a node, respectively. If the node doesn't have any child element, it returns null.

**object.hasOwnProperty('myProp')**

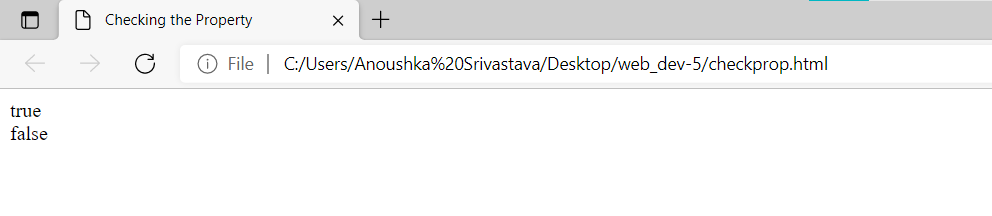
Every JavaScript object has a special method object.hasOwnProperty('myProp') that returns a Boolean indicating whether object has a property myProp.

**Program 1 : Write a JS program to check that an Object contains a particular property or not.**

**Code :**

****

**Output :**

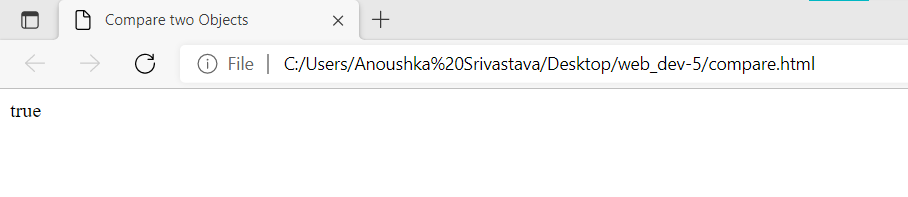
****

**Program 2 : Write a JS program to compare two objects to determine the first object contains equivalent property values to the second object in JavaScript.**

**Code :**

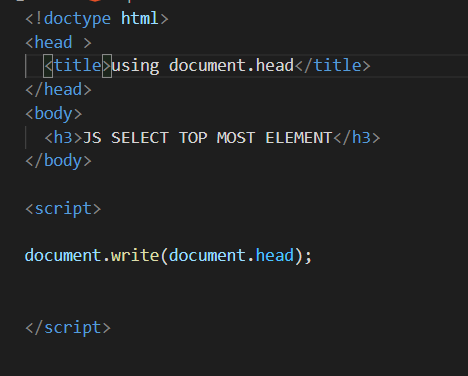
****

**Output :**

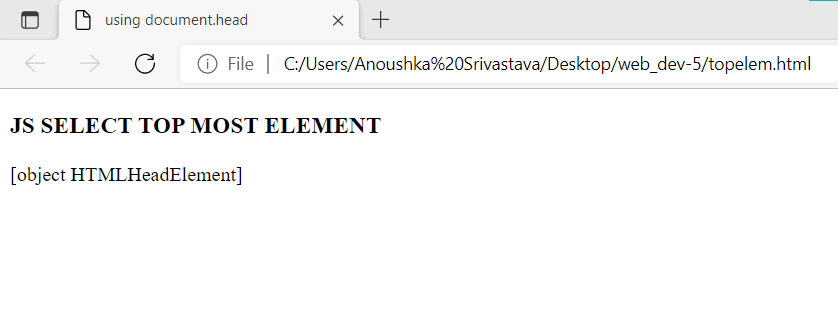
****

**Program 3 : Write a JS program to select the topmost elements in DOM structure.**

**Code :**

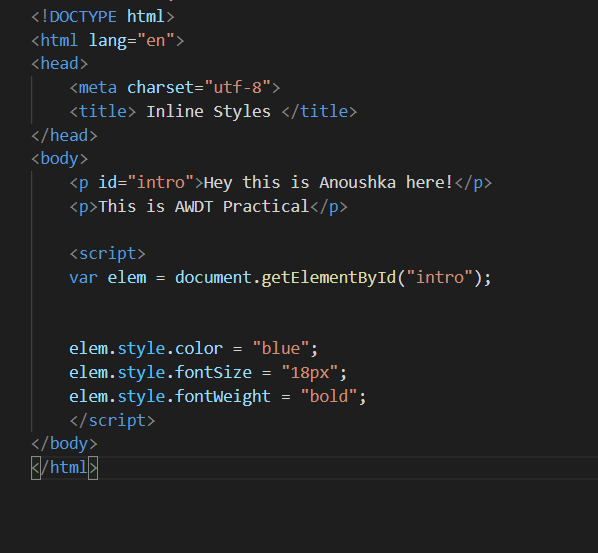


**Output :**

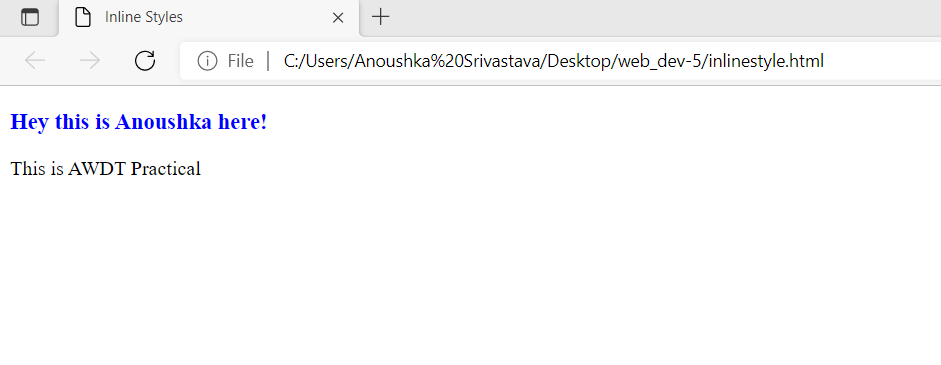
****

**Program 4 : Write a JS program to apply some inline styles in the HTML DOM.**

**Code :**

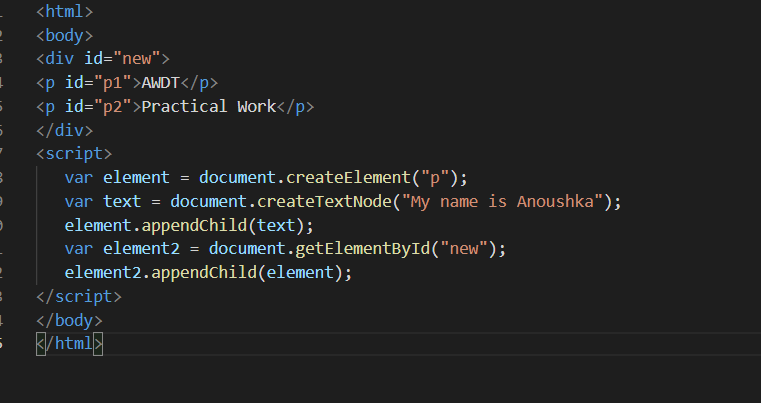


**Output :**

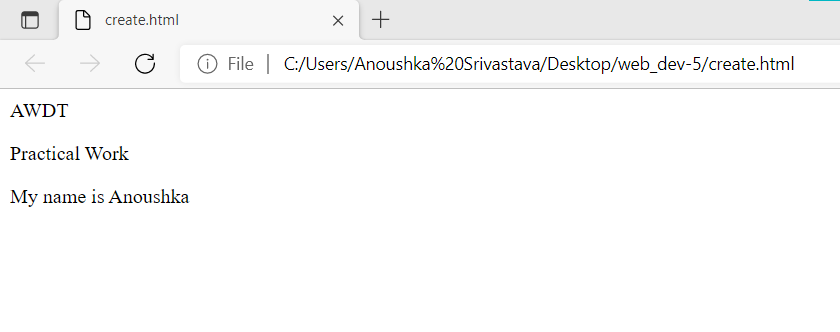
****

**Program 5 : Write a JS program to perform insertion of the new element in DOM structure.**

**Code :**

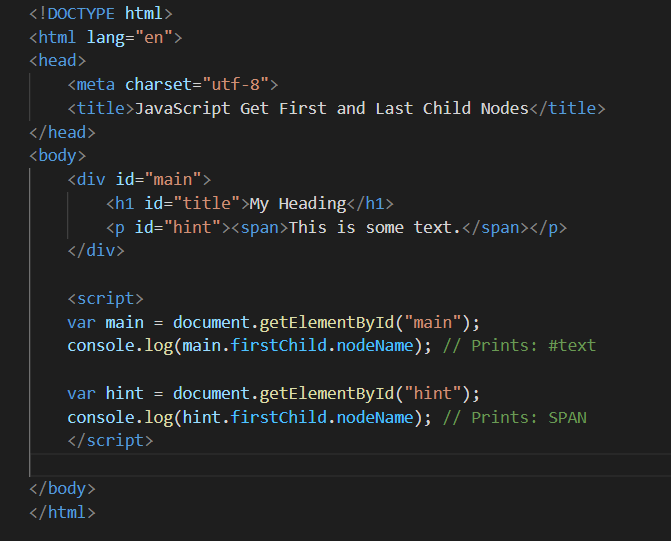
****

**Output :**

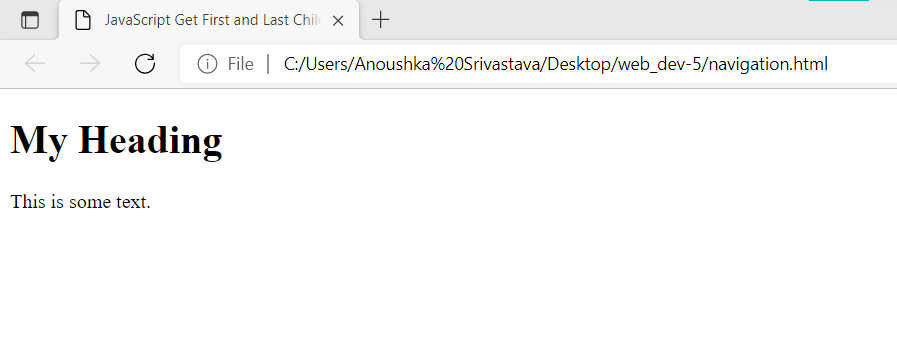
****

**Program 6 : Write a JS program to perform DOM Navigation.**

**Code :**



**Output :**

****