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

**Ans:-**JavaScript is a high-level, versatile programming language primarily used for front-end web development. It allows developers to add interactivity, manipulate the Document Object Model (DOM), and create dynamic content on web pages.

How to use it

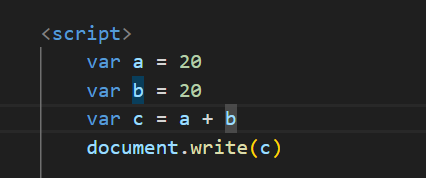
* Open a text editor: Use a code editor like Visual Studio Code, Sublime Text, other editor of your choice.
* Create an HTML file: Start with a basic HTML template to structure your web page.

****

**Q.2 How many type of Variable in JavaScript?**

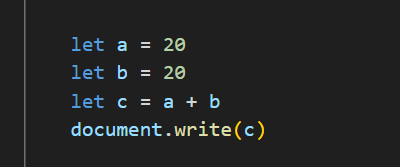
**Ans:-var (function-scoped):-**

* **var was historically used for variable declaration in JavaScript.**
* **Variables declared with var are function-scoped, meaning they are only**

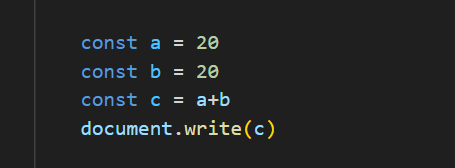
****

**let (block-scoped):-**

* let was introduced in ECMAScript 6 (ES6) to address some of the issues associated with var.
* Variables declared with let are block-scoped, meaning they are limited to the block (enclosed by curly braces) in which they are defined.
* let is generally preferred over var for variable declaration.

****

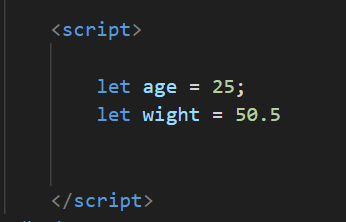
**const (block-scoped, constant):-**

* const is used to declare constants in JavaScript.
* Variables declared with const are block-scoped like let, but once a value is assigned to them, it cannot be reassigned.
* ****

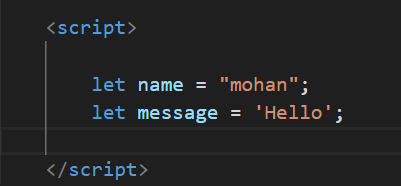
**Q.3 Define a Data Types in js?**

**Ans:- Primitive Data Types:-**

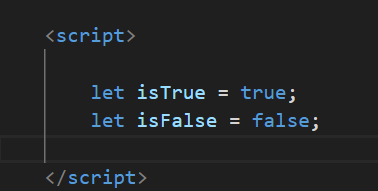
* **Number**: Represents numeric values. It includes integers and floating-point numbers.



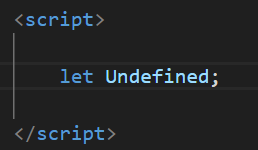
* **String:** Represents sequences of characters enclosed in single or double quotes.



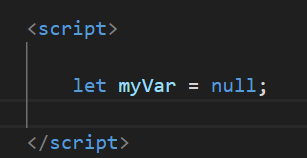
* **Boolean:** Represents either **true** or **false**.



* **Undefined:** Represents a variable that has been declared but has not been assigned a value.

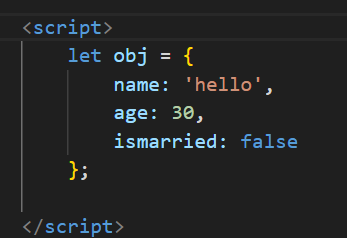


* **Null:** Represents the intentional absence of any object value.

****

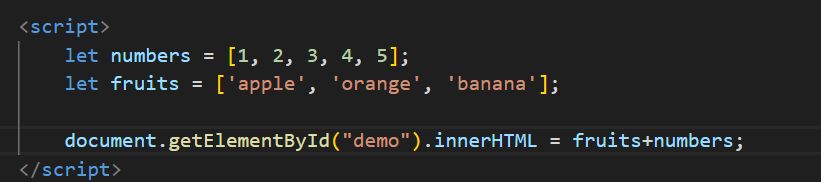
**Object Data Type:**

* Objects are complex data types that can hold key-value pairs and represent more complex structures.



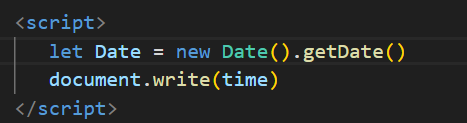
**Array Data Type:**

* Arrays are used to store ordered collections of values.



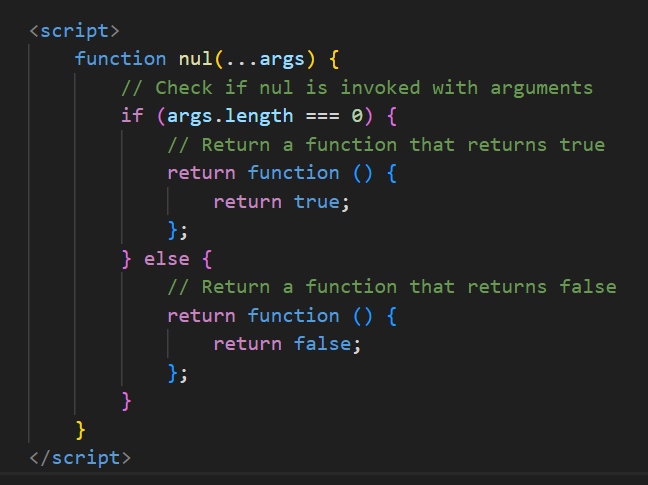
**Date Data Type:**

* Represents a specific point in time.

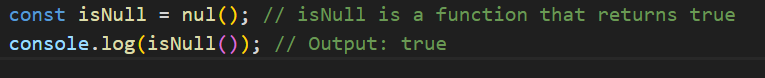


**Q.4 Write a nul Function Which will Work Properly When invoked With Following Syntax.**

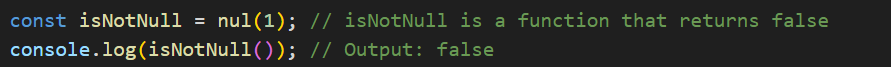
**Ans:-**nul function in JavaScript that works as requested, where it can be invoked in various ways, you can define it like this:-

` 

**With no arguments:**

****

**With arguments:-**

****

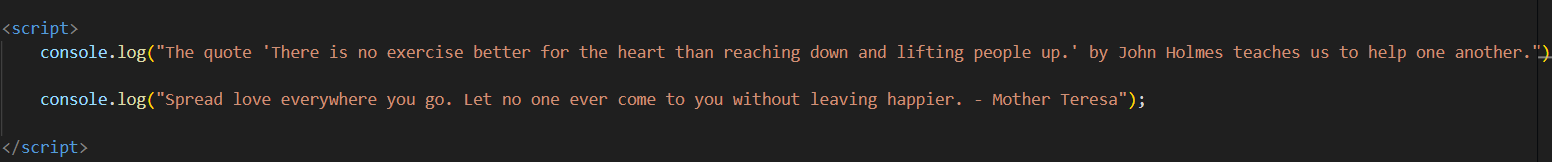
**Q.5 What the deference between undefined and undeclare in JavaScript?**

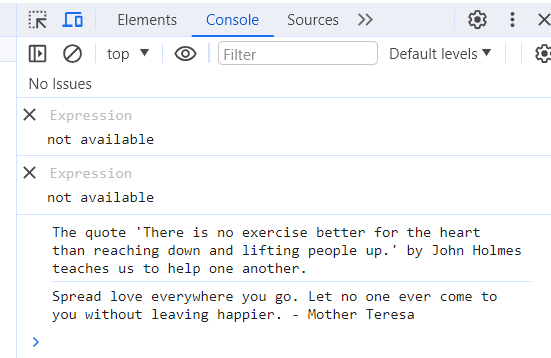
**Ans:-** In JavaScript, "undefined" and "undeclared" refer to different concepts.

1. **Undefined:**
   * A variable is said to be "undefined" when it has been declared but has not been assigned a value.
   * In other words, if you declare a variable using var, let, or const, the variable exists in the scope, but it hasn't been given a value. In this case, its default value is undefined.
2. **Undeclared:**

* A variable is considered "undeclared" if it is used without being formally declared using var, let, or const.
* This usually results in a ReferenceError. When you try to access a variable that has not been declared, JavaScript doesn't recognize it.

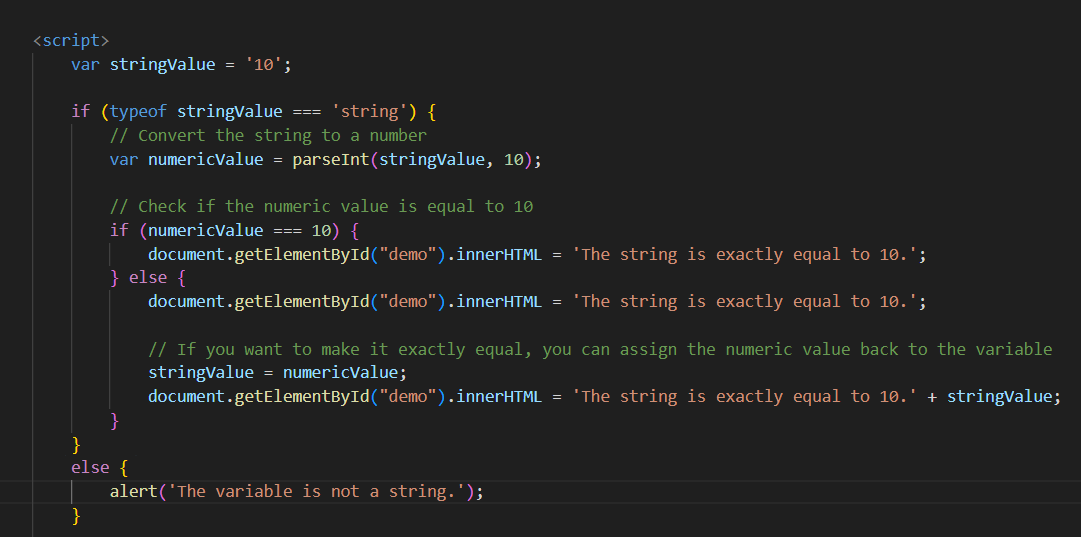
**Q.6 Using console.log() print out the following statement: The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another. Using console.log() print out the following quote by Mother Teresa:**

**Ans:-** ****

****

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

**Ans:-** In JavaScript, the typeof operator returns a string indicating the type of the operand. If you use typeof '10', it will return the string 'string'. If you want to check if the string representation of a number is equal to a numeric value, you can use the parseInt or parseFloat functions to convert the string to a number.

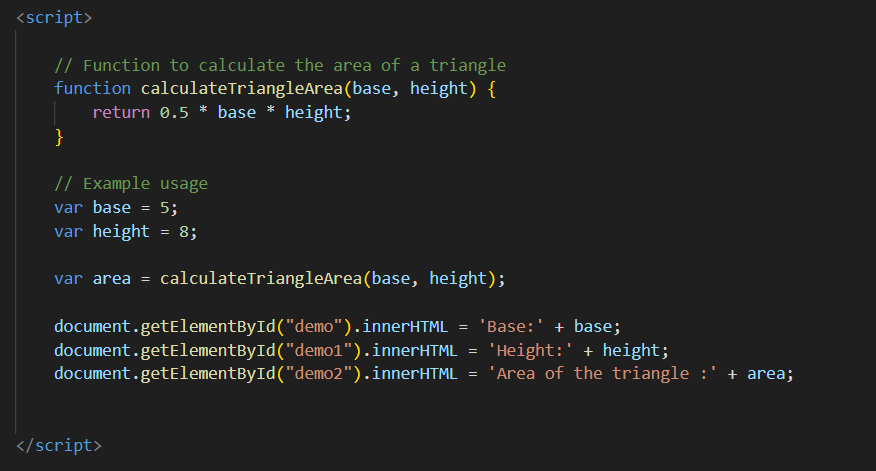


**Q.8 Write a JavaScript Program to find the area of a triangle?**

**Ans:-**

Area=21​×base×height;

This is a simple JavaScript program to find the area of a triangle:

****

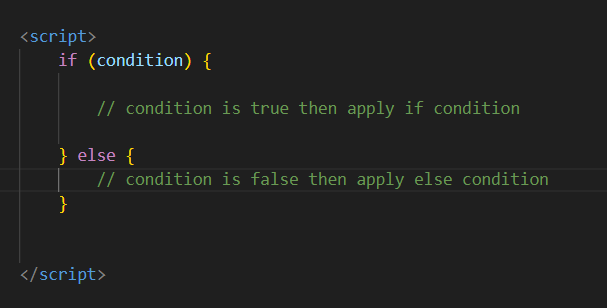
**Q.9 Write a JavaScript program to calculate days left until next Christmas?**

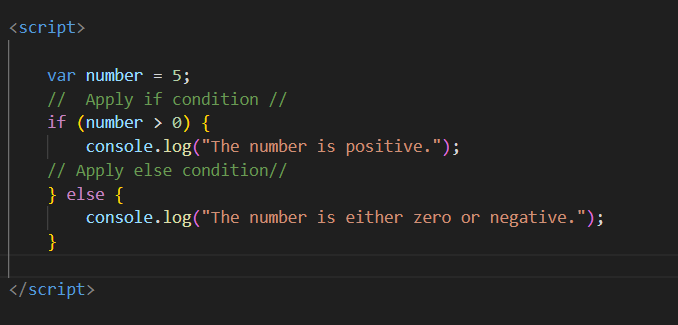
**Ans:-** ****

In this example, the **daysUntilChristmas** function calculates the number of days left until the next Christmas. It checks if Christmas has already passed for the current year, and if so, it sets the date for next year's Christmas. The program then calculates the difference in milliseconds and converts it to days.

**Q.10 What is Condition Statement?**

**Ans:-** A condition statement, often referred to as an "if statement," is a fundamental programming construct that allows you to execute a block of code conditionally based on a specified condition.

****



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

**Ans:-** The formula you've provided,*C*=4×*a*, appears to be the formula for calculating the perimeter of a rectangle, not the circumference. The perimeter of a rectangle is the total length of all its sides.

**For a rectangle, the formula for the perimeter (circumference) is given by:**

*C*=2×(*length*+*width*)

So, if *l* is the length and *w* is the width of the rectangle, the correct formula for the circumference is:

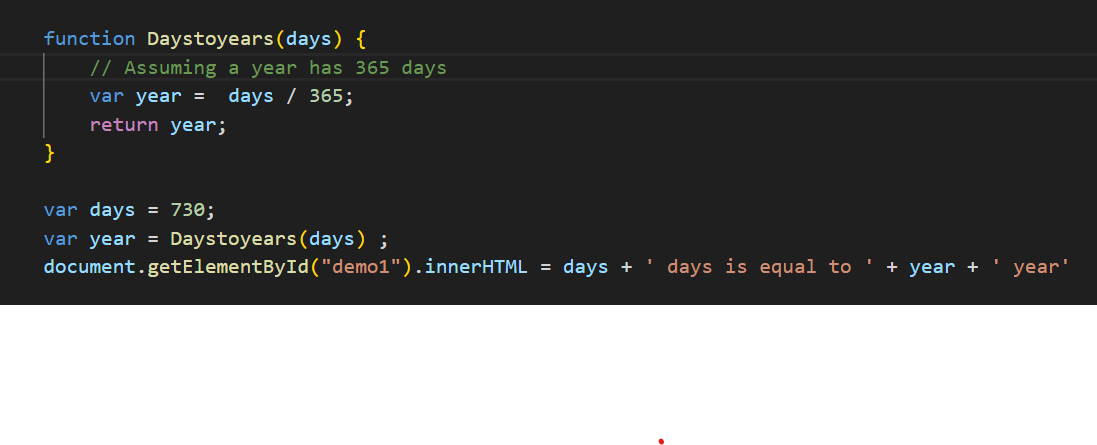
*C*=2×(*l*+*w*);

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

**Ans:-** **Convert years into days:-**

****

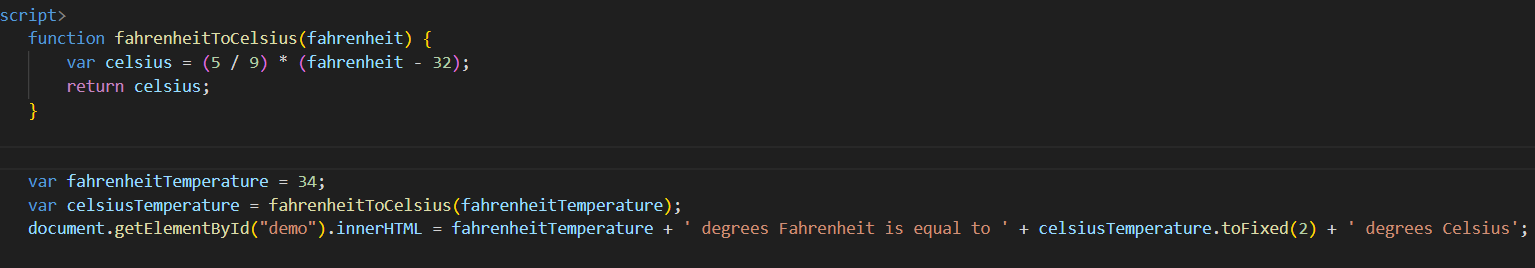
**Conver days to year:-**

****

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

**Ans**:- To convert temperature from Fahrenheit to Celsius,

**Formula *:-C=95​×(F−32)***

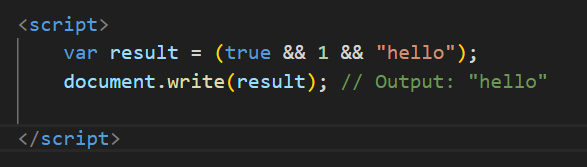
****This function takes a temperature value in Fahrenheit as input and returns the equivalent temperature in Celsius. The toFixed(2) method is used to round the result to two decimal places for better readability.

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

**Ans:-** **The result of the expression (true && 1 && "hello"):**

1. true is truthy.
2. 1 is truthy.
3. "hello" is truthy.

Since all operands are truthy, the && operator returns the last truthy value, which is "hello".

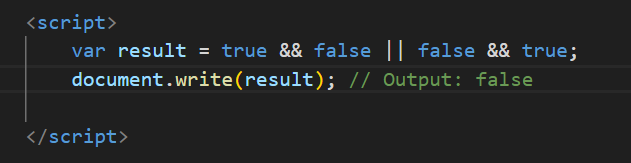


**Q.17 What is the result of the expression true && false || false && true?**

**Ans:- The result of the expression (true && false || false && true)**

1. true && false: This evaluates to false.
2. false && true: This also evaluates to false.
3. false || false: The logical OR of two false values is false.

**So, the overall result of the expression is false.**

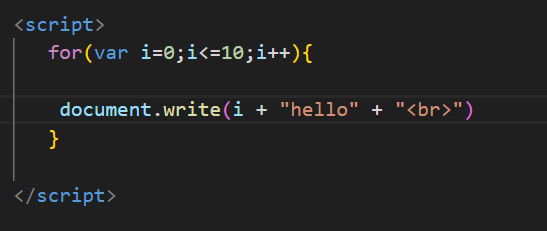
****

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

**Ans:-** In JavaScript, a loop is a programming construct that allows you to repeatedly execute a block of code. It helps automate repetitive tasks by iterating over a sequence of values or until a certain condition is met.

* **Three types of Loop:-**

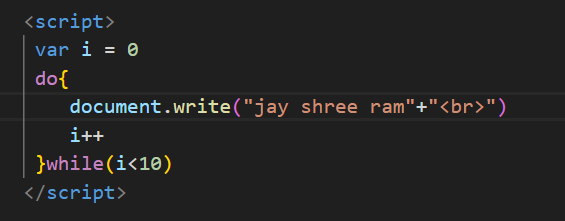
1. **For loop:-** The for loop is commonly used when you know in advance how many times the loop should run. It consists of an initialization, a condition, and an iteration statement.



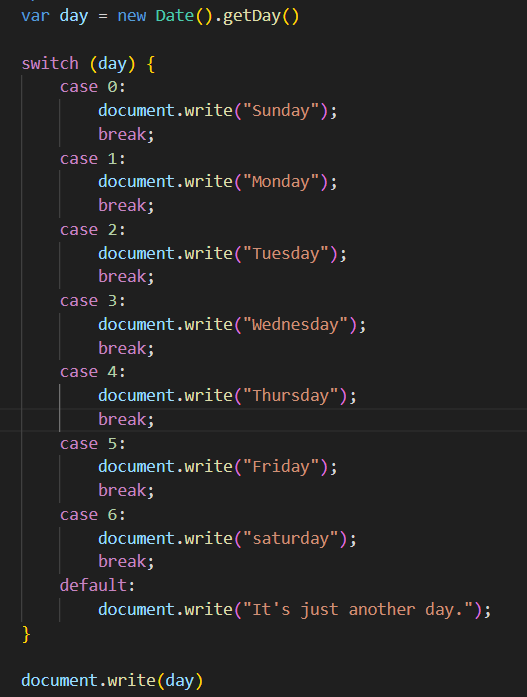
1. **While Loop:-** The while loop repeatedly executes a block of code as long as a specified condition is true.

****

1. **Do-While Loop:-** The do-while loop is similar to the while loop, but it guarantees that the block of code is executed at least once, even if the condition is initially false.

****

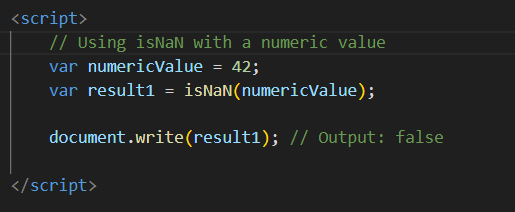
* **Switch Case:-** Switch Case is another control structure in JavaScript that provides a way to handle multiple conditions based on the value of an expression. It's often used as an alternative to long chains of if-else statements when you need to compare a single value against multiple possible values.

****

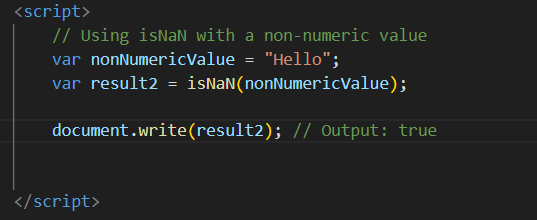
**Q.19 What is the use of is Nan function?**

**Ans:-** The isNaN() function in JavaScript is used to check whether a given value is NaN (Not-a-Number). It returns true if the value is NaN and false if it is a valid number or can be converted to one. This function is particularly helpful when you want to validate whether a variable or expression represents a numeric value.

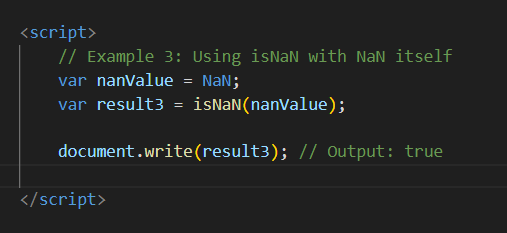
**Example:**-

****

In this example, isNaN(numericValue) returns false because numericValue is a valid number.



In this example, **isNaN(nonNumericValue)** returns **true** because the value stored in **nonNumericValue** is not a valid number.



In this example, isNaN(nanValue) returns true because nanValue is explicitly set to NaN.

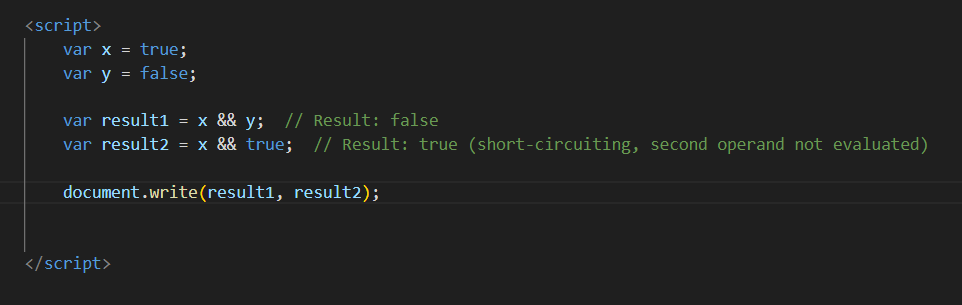
**Q.20 What is the difference between && and || in JavaScript?**

**Ans:-** In JavaScript, && and || are logical operators used for boolean operations. They operate on boolean values and return a boolean result based on the evaluation of the expressions they connect.

**Logical AND (&&):**

* Returns true if both operands are true.
* Returns false if at least one operand is false.
* If the first operand is false, the second operand is not evaluated (short-circuiting).

**Example:-**

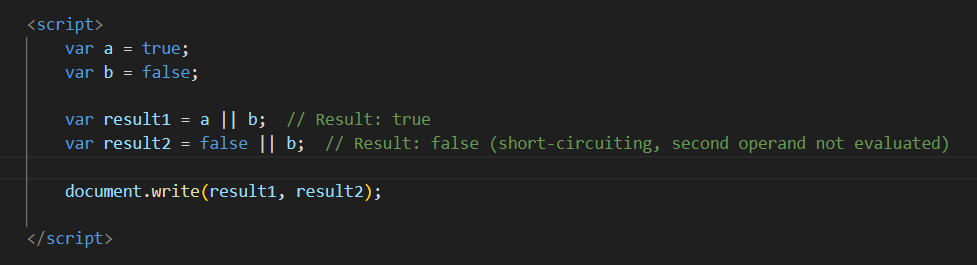


| **Expression** | **Result** |
| --- | --- |
| **true && true** | **true** |
| **true && false** | **false** |
| **false && true** | **false** |
| **false && false** | **false** |

**Logical OR (||):**

* Returns **true** if at least one operand is true.
* Returns **false** if both operands are false.
* If the first operand is true, the second operand is not evaluated (short-circuiting).

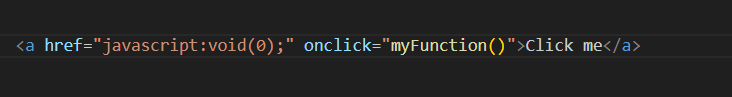
**Example:-**



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

**Ans:-**In JavaScript, void(0) is often used as a way to create a "void" or undefined value. The void operator takes an expression as its operand and evaluates it, then returns undefined. It's commonly used to create a hyperlink that doesn't navigate anywhere when clicked, preventing the browser from changing the current page.

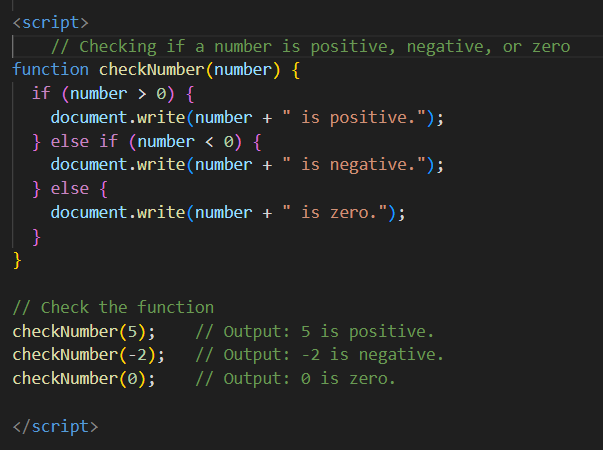
**The use of void(0) in an HTML anchor (<a>) tag:**

****

Inthis example, clicking the "Click me" link will trigger the myFunction() JavaScript function. The href attribute is set to "javascript:void(0);". The purpose of void(0) here is to ensure that clicking the link doesn't cause the browser to navigate to a new page. The void(0) expression is evaluated, and since it always returns undefined, it essentially prevents any navigation.

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

**Ans:-** In JavaScript, you can use an if statement to check whether a number is positive, negative Here's an example:

****

In this example, the checkNumber function takes a parameter number and uses an if statement to check its value. If number is greater than 0, it's considered positive. If it's less than 0, it's considered negative. If it's neither greater nor less than 0, it must be zero.

**Q.23 Find the Character Is Vowel or Not ?**

**Ans:-** To check the character is a vowel or not in JavaScript, you can use a function that compares the character against a set of known vowels.

**example:-**

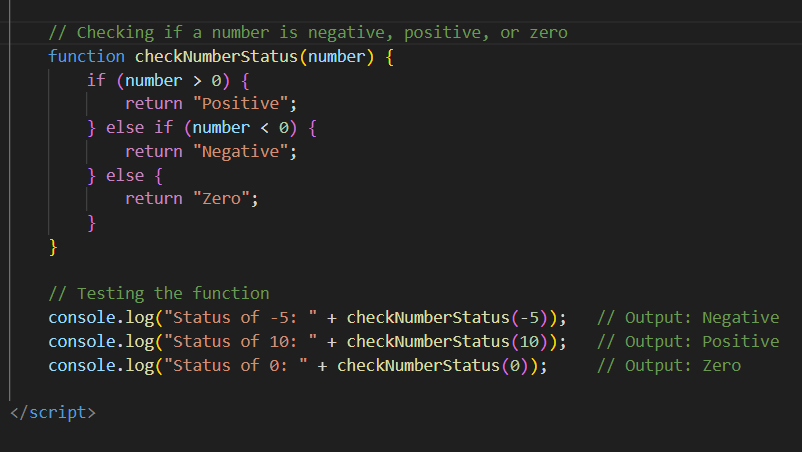
****

In this example, the isVowel function takes a character as input, converts it to lowercase using toLowerCase(), and then checks if it matches any of the vowels ('a', 'e', 'i', 'o', 'u'). The result is true if the character is a vowel and false .

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

**Ans:-**JavaScript function to check whether a number is negative, positive, or zero.

**example:-**

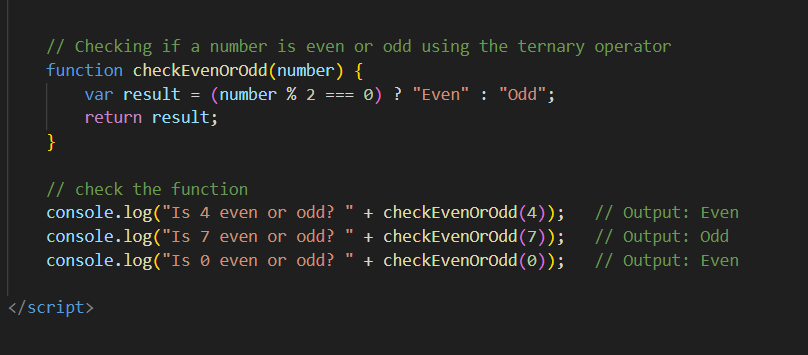
****

In this example, the checkNumberStatus function takes a number as input and uses an if-else statement to determine whether it's negative, positive, or zero. The function returns a string indicating the status.

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

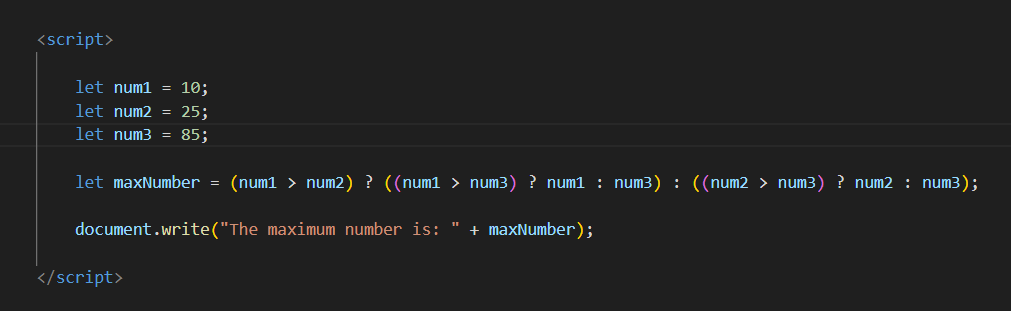
**Ans:-use the ternary operator to determine whether a number is even or odd in JavaScript.**

**Example:-**

****

In this example, the checkEvenOrOdd function takes a number as input and uses the ternary operator (? :) to check if the number is even or odd. If the remainder of the division by 2 is 0, the number is even; otherwise, it's odd.

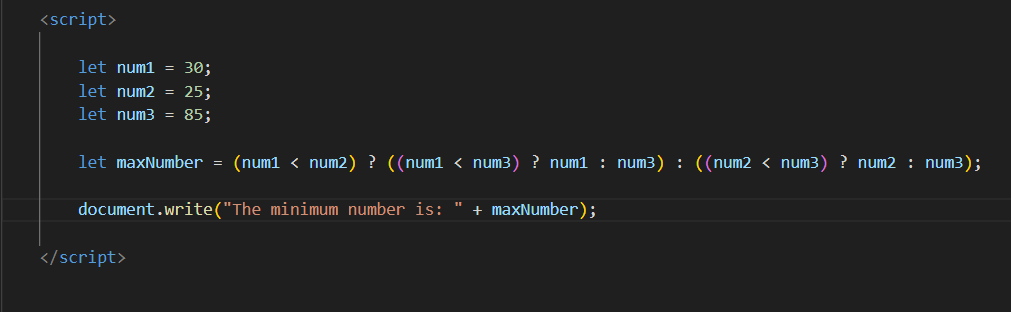
**Q.26 Write find maximum number among 3 numbers using ternary operator in JS?**

**Ans:-** Use the ternary operator to find the maximum number among 3 numbers in JavaScript****

In this example, the ternary operator is used to compare each pair of numbers and find the maximum among them. The structure is (condition) ? true-value : false-value.

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

**Ans:-** Use the ternary operator to find the minimum number among 3 numbers in JavaScript

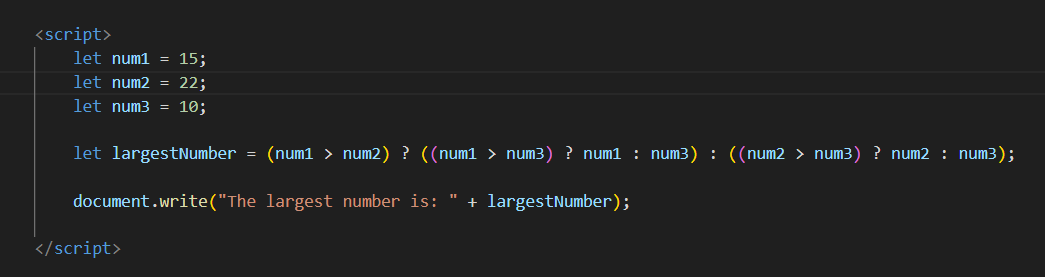
****

In this example, the ternary operator is used to compare the three numbers and determine the minimum value. The format of the ternary operator is (condition) ? (value if true) : (value if false).

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

**Ans:-**finding the largest of three numbers in JavaScript:

**Example:**

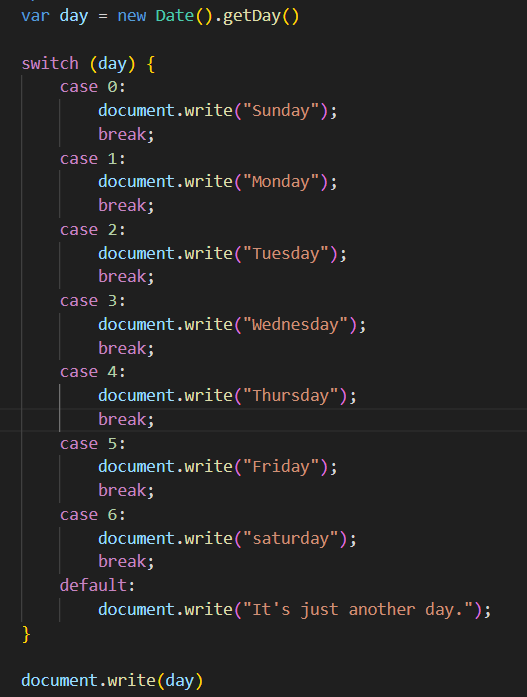


In this example, we have three variables **num1**, **num2**, and **num3** representing three numbers.

**Q.29 Write to show**

**i. Monday to Sunday using switch case in JS?**

**Ans:-**

****

**ii. Vowel or Consonant using switch case in JS?**

**Ans:-**use a switch case statement to check whether a given letter is a vowel or a consonant in JavaScript.

****

In this example, the switch statement is used to check the value of the variable letter. The toLowerCase() method is used to convert the letter to lowercase to handle both uppercase and lowercase letters.

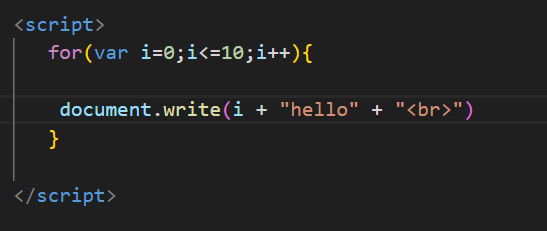
**Conditional looping logic Question)**

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

**Ans:-**In JavaScript, there are several looping structures that allow you to execute a block of code repeatedly. The most common looping structures are:

**for loop:** T

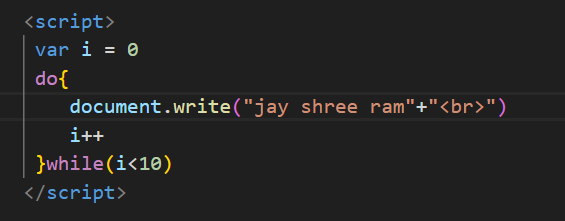
he for loop is used when you know the number of iterations you want to perform.



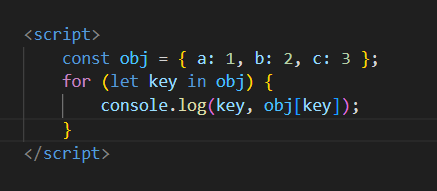
**while loop:-** The while loop is used when you don't know the number of iterations beforehand, and the loop continues as long as the specified condition is true.

****

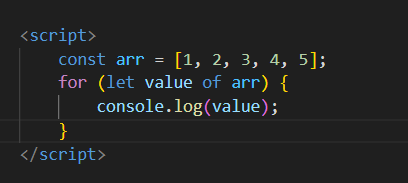
**do-while loop:** Similar to the **while** loop, but it guarantees that the block of code inside the loop is executed at least once, as the condition is checked after the execution.

****

**for...in loop:-** This loop is used for iterating over the properties of an object.

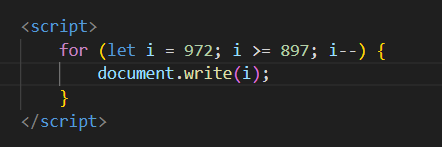


**for...of loop:** Introduced in ECMAScript 2015, this loop is used for iterating over iterable objects like arrays, strings, etc.



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

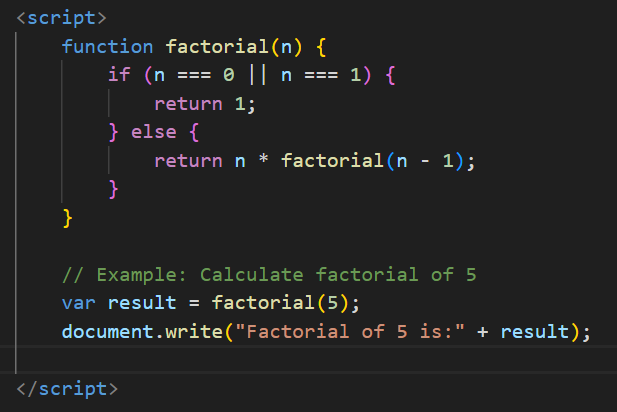
**Ans:-** To print the numbers from 972 to 897 using a for loop in JavaScript, you can set up the loop to iterate in reverse.

****

This loop starts with i initialized to 972 and decrements it by 1 in each iteration. The loop continues as long as i is greater than or equal to 897, printing the values from 972 down to 897

**Q.32 Write to print factorial of given number?**

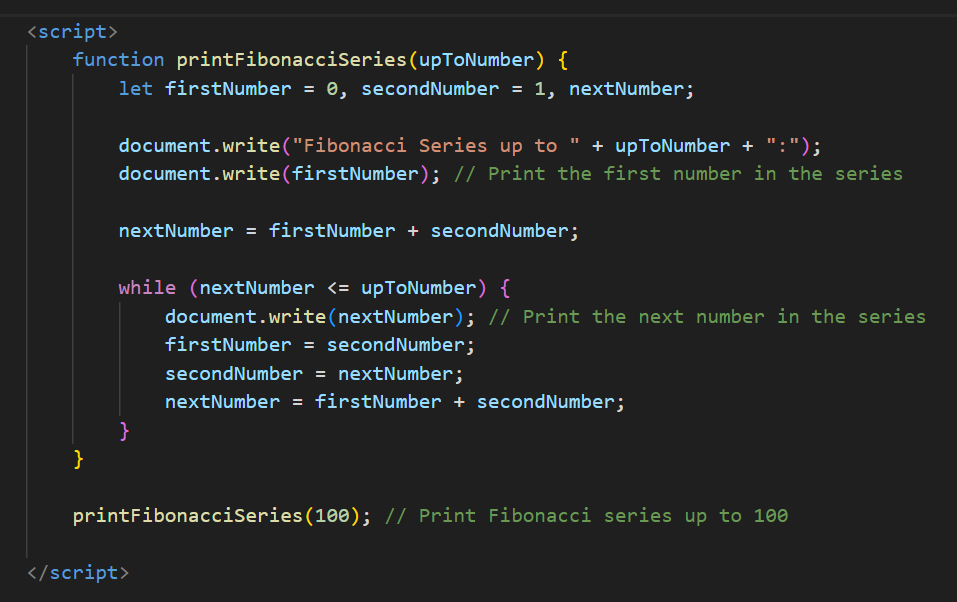
**Ans:-** print factorial of given number:



This code defines a function called **factorial** that recursively calculates the factorial of a given number **n**. The base case is when **n** is 0 or 1, in which case the factorial is 1. Otherwise, it recursively calls the **factorial** function with **n - 1** and multiplies the result by **n**.

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

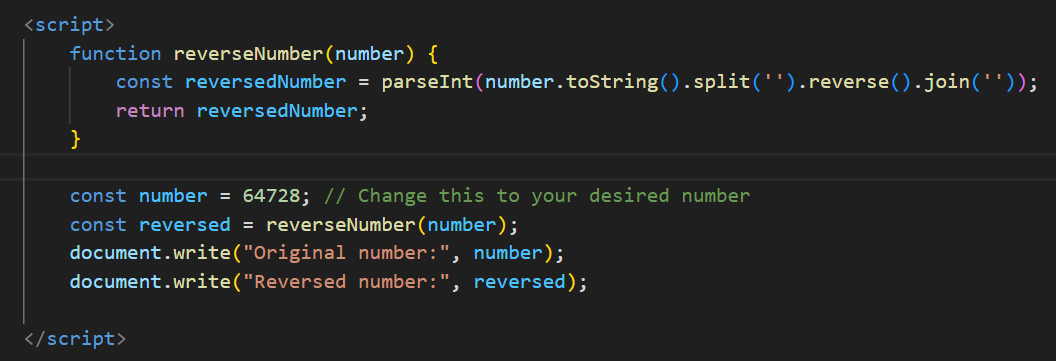
**Ans:-To print Fibonacci series up to given numbers:**

****

This code defines a function fibonacciSeries to generate the Fibonacci series up to a given limit and another function printFibonacciSeries to print the generated series. You can adjust the limit variable to print the Fibonacci series up to your desired number.

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

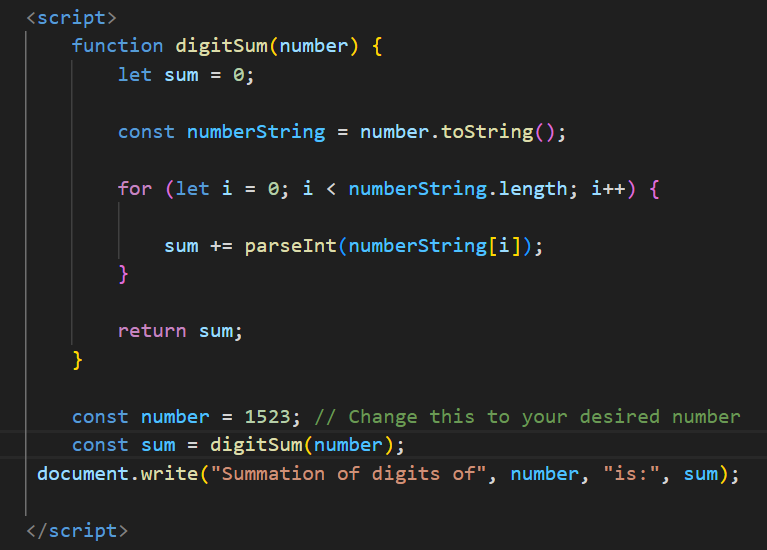
**Ans:-** **To print number in reverse order e.g.: number = 64728 ---> reverse =82746 in JS:**

****

This function first converts the number to a string, then splits it into an array of characters, reverses the array, and finally joins the characters back together into a string. Finally, it parses the string back into an integer.

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

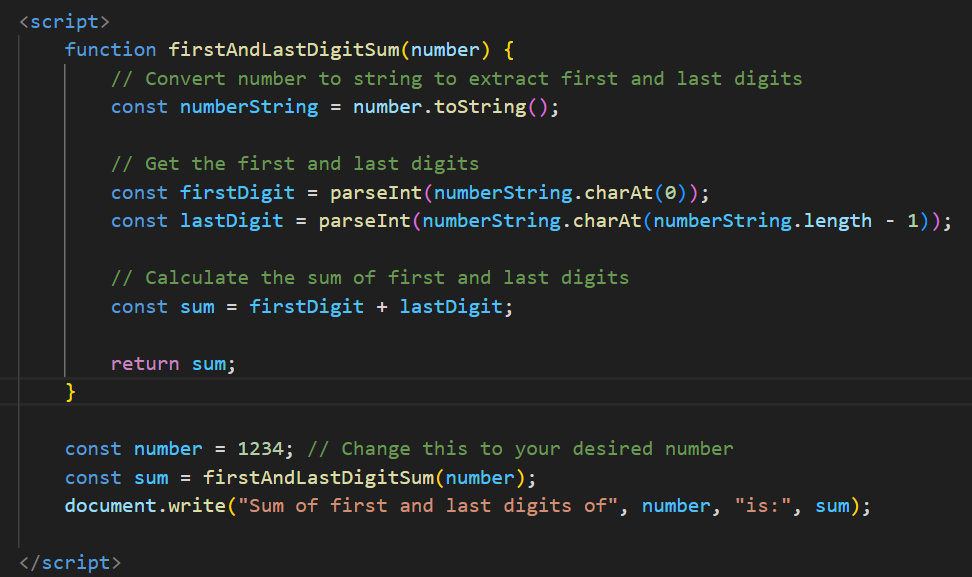
**Ans:-To find the summation of the digits of a given number in JavaScrip:**

****

This function iterates through each digit of the given number by converting it to a string and then back to a number. It adds each digit to the sum variable and returns the total sum of digits.

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

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

****

This program first converts the number to a string to extract the first and last digits using charAt(). Then it converts these characters back to integers using parseInt() and calculates their sum. Finally, it returns the sum of the first and last digits.

**Q.37 Use console.log() and escape characters to print the following pattern in JS?**

1 1 1 1 1

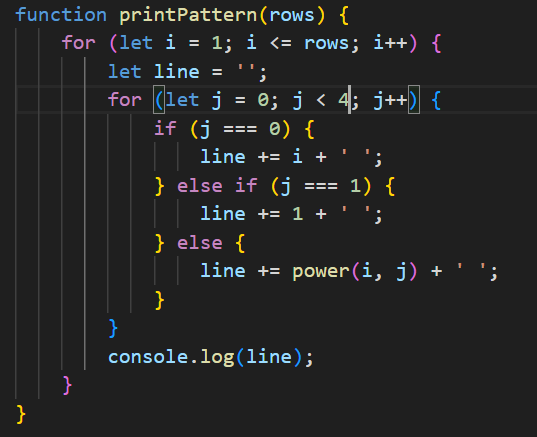
2 1 2 4 8

3 1 3 9 27

4 1 4 16 64

5 1 5 25 125

**Ans:-**

****

**Q.38 Use pattern in console.log in JS?**

1)

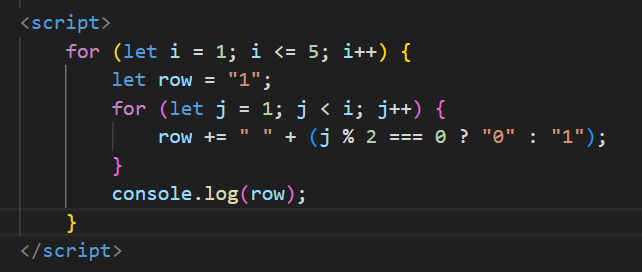
1

1 0

1 0 1

1 0 1 0

1 0 1 0 1

****

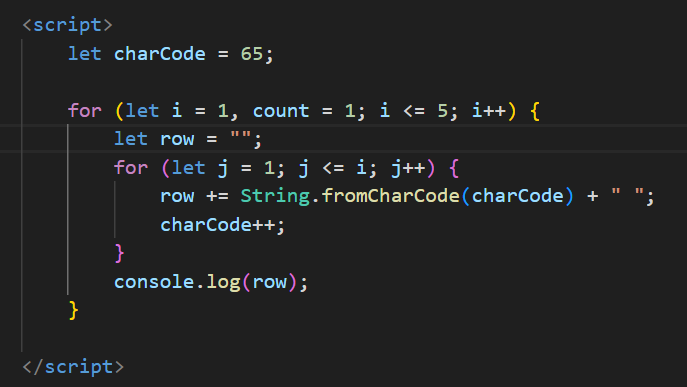
2) A

B C

D E F

G H I J

K L M N O

****

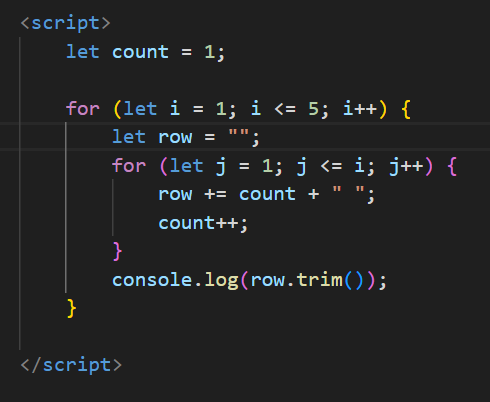
3) 1

2 3

4 5 6

7 8 9 10

11 12 13 14 15

****

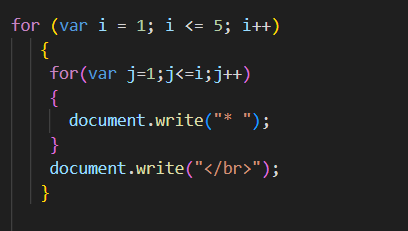
4) \*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

****

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

**Ans:-**

****

This code defines two functions: isPalindrome to check if a number is a palindrome, and checkPalindromes to prompt the user for three numbers and check each number if it's a palindrome or not using a while loop

**Q.40 Write a JavaScript Program to display the current day and time in the following format. Sample Output: Today is Friday. Current Time is 12 PM: 12 : 22 2 ?**

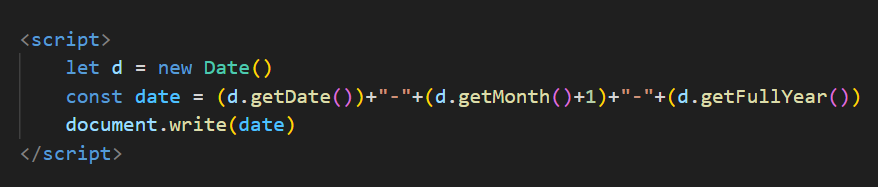
**Ans:-**

****

This program defines two functions: getCurrentDay to get the current day of the week and getCurrentTime to get the current time in 12-hour format with AM/PM. It then calls these functions to obtain the current day and time and logs them in the specified format using console.log().

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

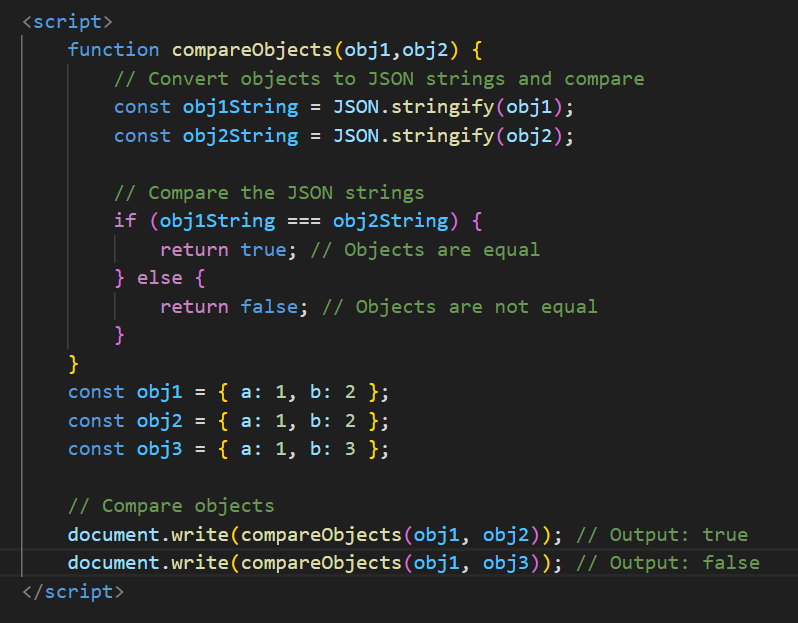
**Ans:-** **A JavaScript program to get the current date:**

****

This program creates a new Date object to get the current date and then extracts the year, month, and day components.

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

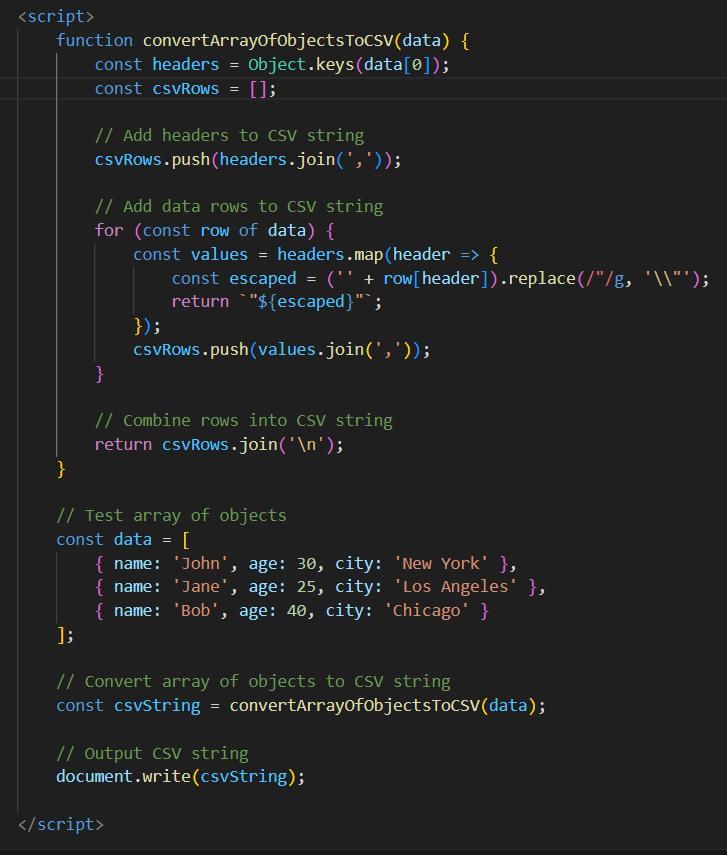
**Ans:-JavaScript program to compare two objects:**

****

This program defines a function compareObjects that takes two objects as arguments. Inside the function, it converts the objects to JSON strings using JSON.stringify() and then compares the strings. If the JSON strings are equal, it returns true, indicating that the objects are equal; otherwise, it returns false.

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

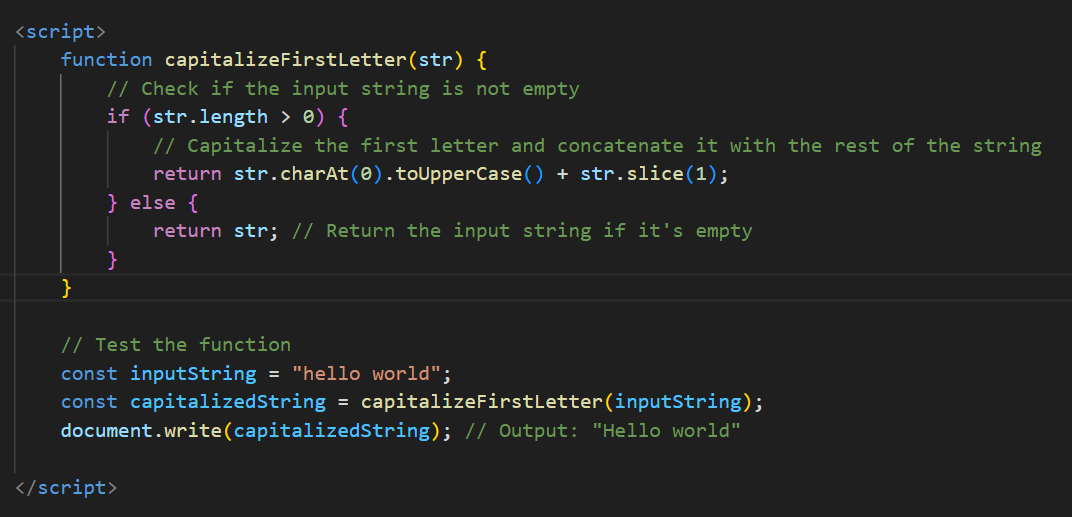
**Ans:-** **A JavaScript program to convert an array of objects into a CSV string:**

****

This program defines a function convertArrayOfObjectsToCSV that takes an array of objects as input. It first extracts the headers (property names) from the first object in the array.

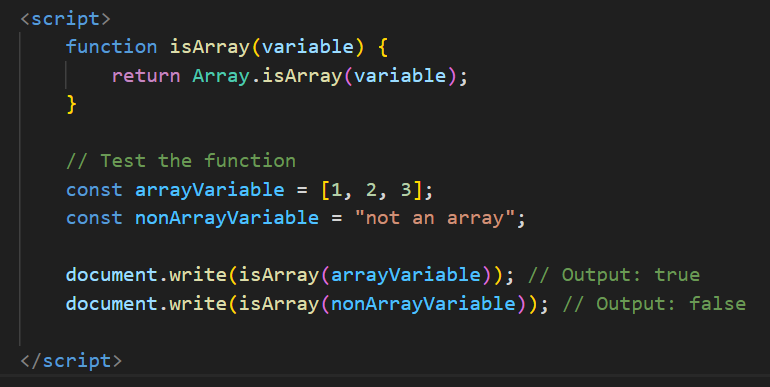
**Q.44 Write a JavaScript program to capitalize first letter of a string?**

**Ans:-You can capitalize the first letter of a string in JavaScript:**

****

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

**Ans:-** **A JavaScript program to determine if a variable is array:**

****

In this program, the isArray function takes a variable variable as input and returns true if the variable is an array, using the Array.isArray() method. Otherwise, it returns false. The function is tested with both an array variable and a non-array variable, and the results are logged to the console.

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

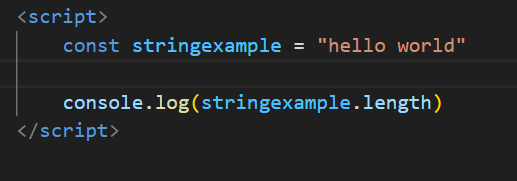
**Ans:-** A clone an array in JavaScript using various methods, such as using the spread operator (...), the slice() method, or the concat() method.

****

* cloneArrayUsingSpread clones the array using the spread operator ([...arr]).
* cloneArrayUsingSlice clones the array using the slice() method with no arguments.
* cloneArrayUsingConcat clones the array using the concat() method with an empty array (arr.concat([])).

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

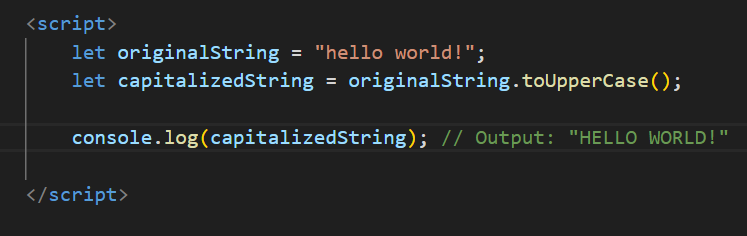
**Ans:-** The length of a string to the browser console using console.log():

****

it will display the length of the string "hello world"

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

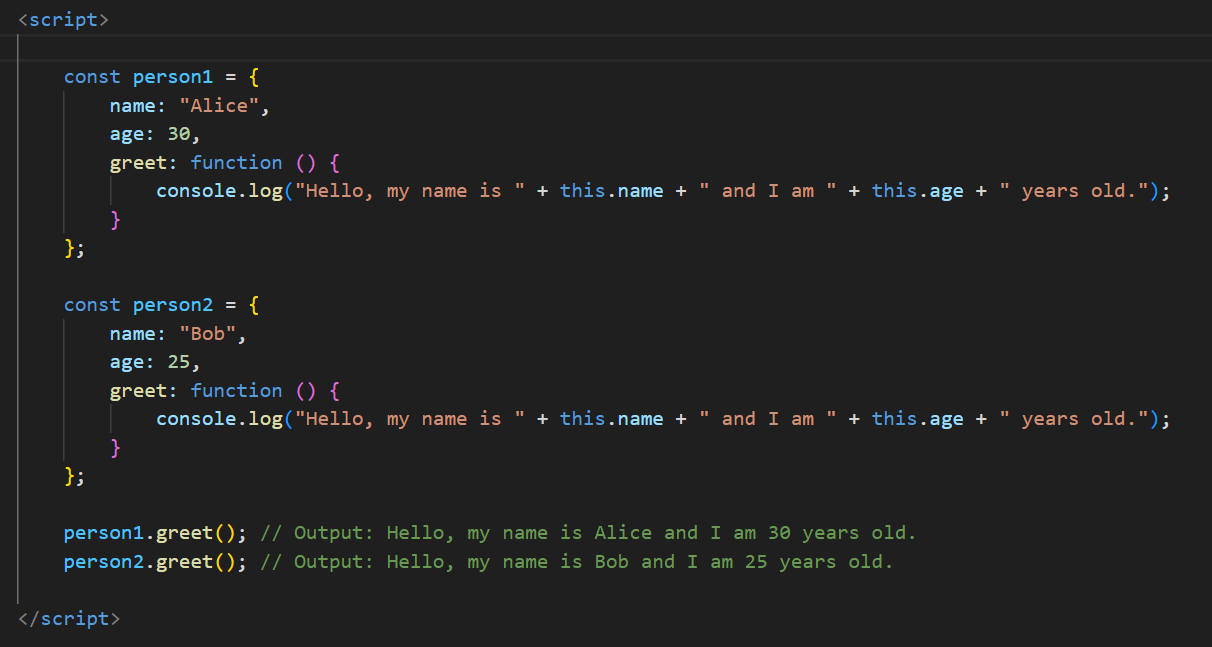
**Ans:- Change all the string characters to capital letters using toUpperCase() method:**

****

* originalString contains the original string.
* capitalizedString is assigned the result of calling toUpperCase() on originalString, which converts all characters in the string to uppercase.

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

**Ans:- The drawback of declaring methods directly in JavaScript objects:**



In this example, both person1 and person2 objects have a greet method defined directly within their object literals.

**Q.51 Write a JavaScript program to get the current date. Expected Output : mm-dd-yyyy, mm/dd/yyyy or dd-mm-yyyy, dd/mm/yyyy?**

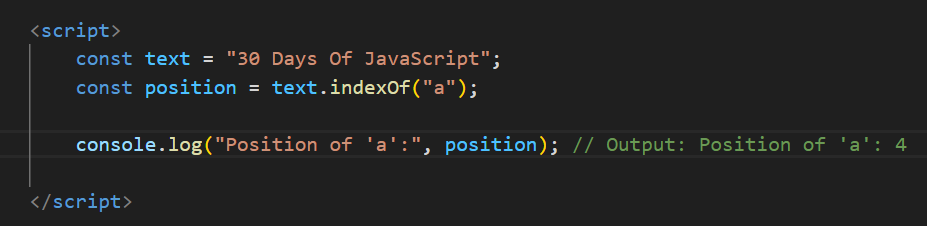
**Ans:-** AJavaScript's Date object to get the current date and then format it according to your desired output format.

****

* The getCurrentDate function takes a format parameter specifying the desired output format ("mm-dd-yyyy", "mm/dd/yyyy", "dd-mm-yyyy", or "dd/mm/yyyy").
* It retrieves the current date using the Date object and then formats it based on the specified format.
* The formatted date is returned as a string.

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

**Ans**:- Use indexOf to determine the position of the first occurrence of a in 30 Days Of JavaScript:



In this example, **indexOf("a")** returns the position of the first occurrence of the substring "a" within the string "30 Days Of JavaScript". The returned position is zero-based, so "a" is found at index 4(which is the fourth character in the string counting from 0).

**Q.53 Use lastIndexOf to determine the position of the last occurrence of a in 30 Days Of JavaScript?**

**Ans:-** Use lastIndexOf to determine the position of the last occurrence of a in 30 Days Of JavaScript:

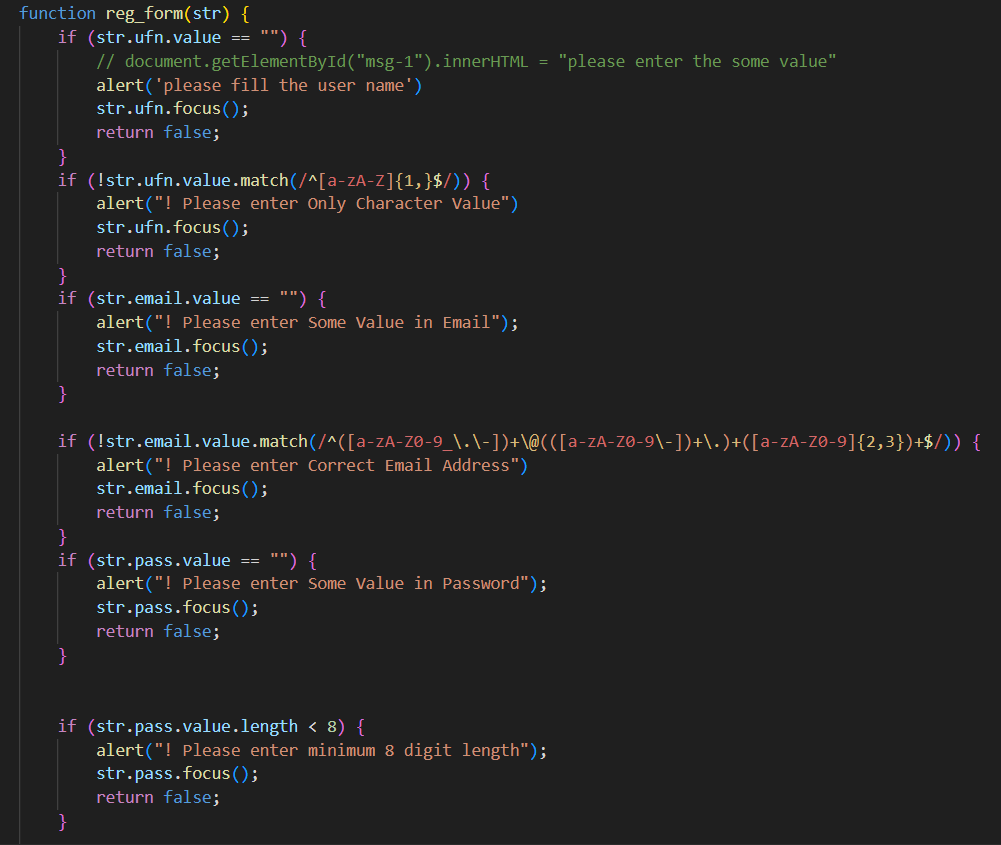
****

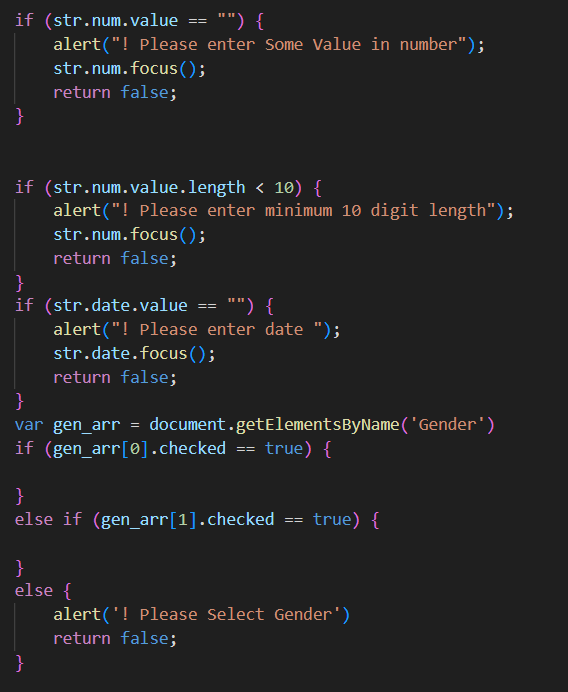
In this example, lastIndexOf("a") returns the position of the last occurrence of the substring "a" within the string "30 Days Of JavaScript". The returned position is zero-based, so "a" is found at index 14 (which is the twenty-fourth character in the string counting from 0).

**Q.54 Form Validtion in JS?**

**Ans:- Form validation:**

****

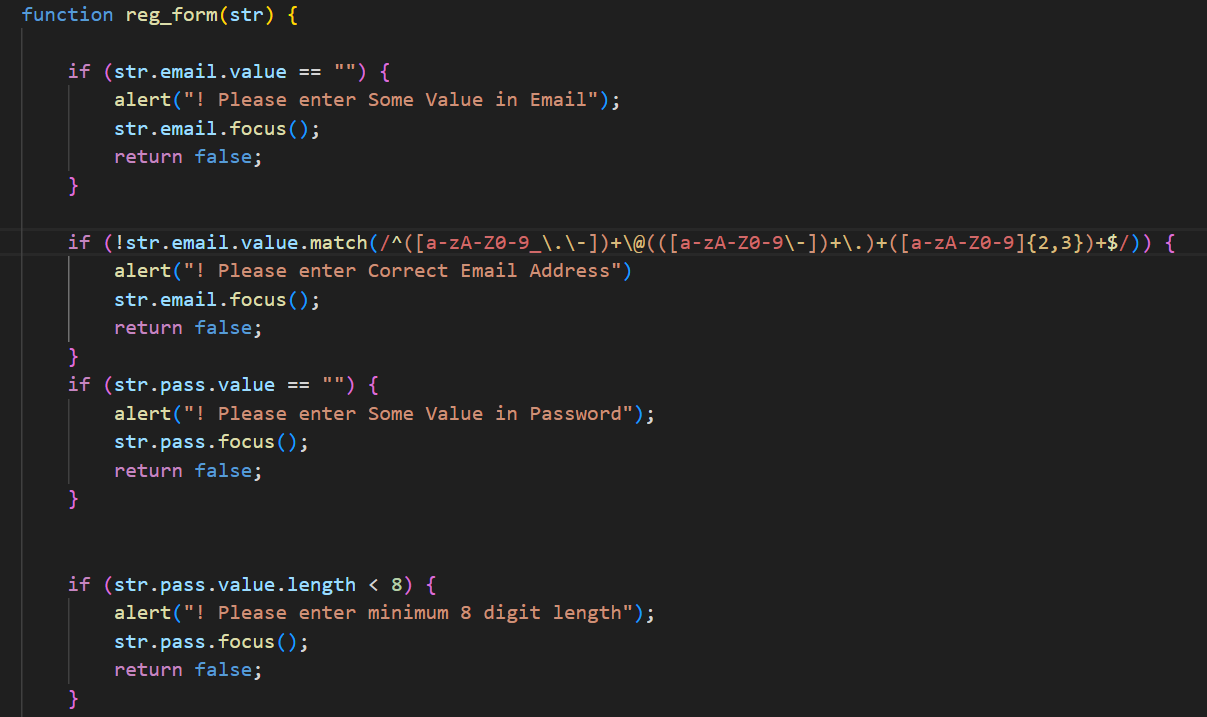
****

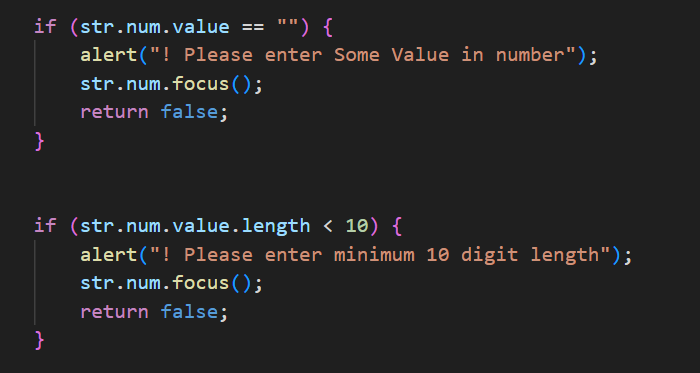
****

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

**Ans:-form in email,number,password,validation:**

****

****

****

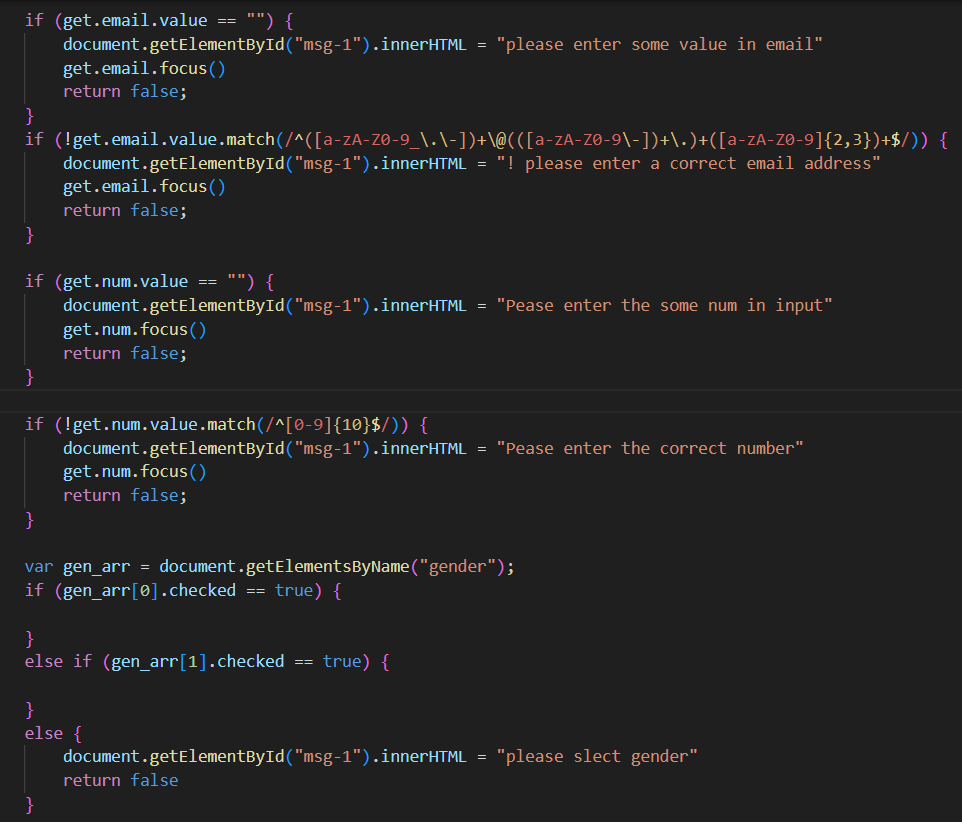
**Q.56 Dynamic Form Validation in JS?**

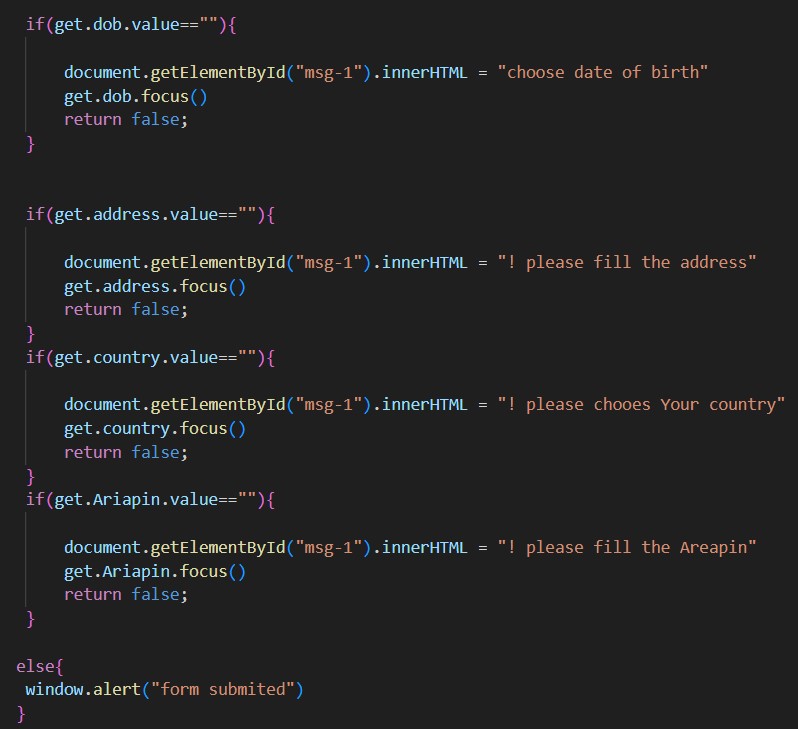
**Ans:-**

****

****

****

****



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

**Ans:-**In JavaScript, there are many types of events that can occur, and they are typically categorized based on their source or trigger. Here are some common types of events:

1. Mouse Events: These events are triggered by mouse actions such as clicks, movements, and scrolls. Examples include click, dblclick, mousedown, mouseup, mousemove, mouseover, mouseout, mouseenter, mouseleave, etc.
2. Keyboard Events: These events are triggered by keyboard actions such as key presses. Examples include keydown, keyup, and keypress.
3. Form Events: These events are triggered by actions within HTML forms such as submitting a form or changing the value of an input element. Examples include submit, reset, change, input, focus, blur, etc.
4. Touch Events: These events are triggered by touch interactions on touch-enabled devices. Examples include touchstart, touchmove, touchend, touchcancel, etc.
5. Media Events: These events are triggered by multimedia elements such as audio or video. Examples include play, pause, ended, loadedmetadata, volumechange, etc.
6. Drag Events: These events are triggered by drag-and-drop interactions. Examples include dragstart, drag, dragenter, dragleave, dragover, drop, dragend, etc.

**Q.58 What is Bom vs Dom in JS?**

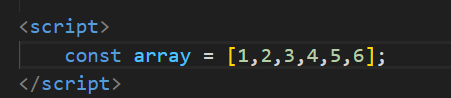
**Ans:-**

1. DOM (Document Object Model):-
   * The DOM represents the structure of an HTML document as a tree of nodes. Each node represents a part of the document, such as elements, attributes, and text.
   * JavaScript can manipulate the DOM to dynamically change the content, structure, and style of a webpage. This includes tasks like adding or removing elements, changing attributes, modifying styles, and handling events.
   * Examples of DOM manipulation methods and properties include document.getElementById(), element.innerHTML, element.style, element.addEventListener(), etc.
2. BOM (Browser Object Model):
   * The BOM represents everything about the browser environment that isn't directly related to the document being displayed. It provides objects and interfaces for interacting with the browser itself, such as controlling the browser window, managing cookies, and navigating to other pages.
   * Unlike the DOM, which deals with the structure and content of the webpage, the BOM deals with the browser as a whole.
   * Examples of BOM methods and properties include window.open(), window.alert(), window.location.href, etc.

**Q.59 Array vs object defences in JS?**

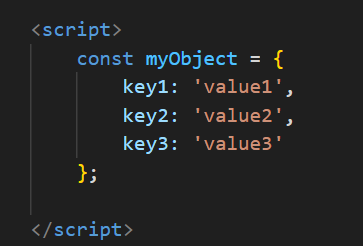
**Ans:-Arrays:-**

* Arrays are ordered collections of values, where each value is identified by an index. The indices of arrays are zero-based integers.
* Arrays are typically used to store lists of similar items or values that are accessed sequentially.
* They are mutable, meaning their contents can be modified (e.g., adding or removing elements, changing existing elements).
* Arrays have built-in methods and properties for manipulating their contents, such as push(), pop(), shift(), unshift(), splice(), forEach(), map(), etc.



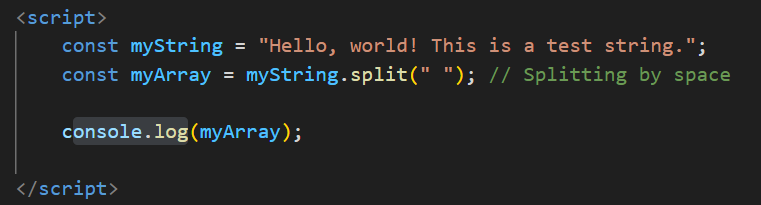
**Objects**:

* Objects are collections of key-value pairs, where each key is a unique string (or symbol) that maps to a value.
* Unlike arrays, the order of key-value pairs in objects is not guaranteed.
* Objects are typically used to represent structured data, where each value is associated with a specific identifier (key).
* They are mutable, meaning their properties can be added, modified, or deleted.
* Objects have various methods and properties for manipulation, but they are primarily accessed and modified using dot notation (**object.property**) or bracket notation (**object['property']**).



**Q.60** **Split the string into an array using split() Method?**

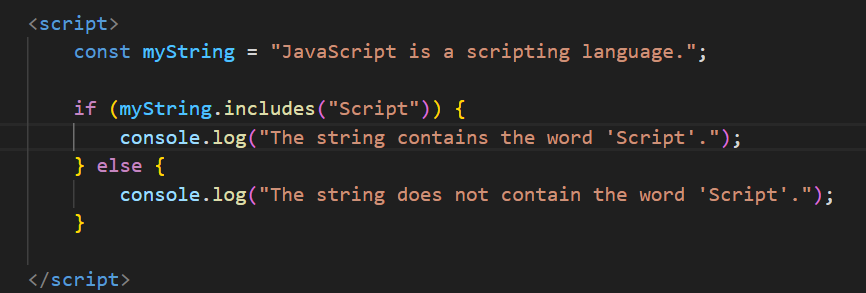
**Ans:-** Split the string into an array using split() Method:



In this example, the **split(" ")** method splits the **myString** into an array of substrings using a space (" ") as the separator. The resulting array will contain each word as a separate element.

**Q.61 Check if the string contains a word Script using includes() method?**

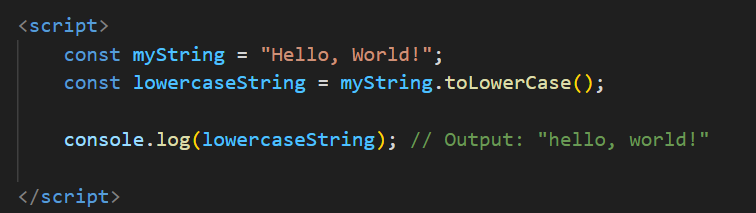
**Ans:-** Check if the string contains a word Script using includes() method:



In this example, the **includes()** method checks if the string **myString** contains the substring "Script". If it does, it returns **true**, indicating that the word "Script" is present in the string. Otherwise, it returns **false**. Based on the result, the appropriate message is logged to the console.

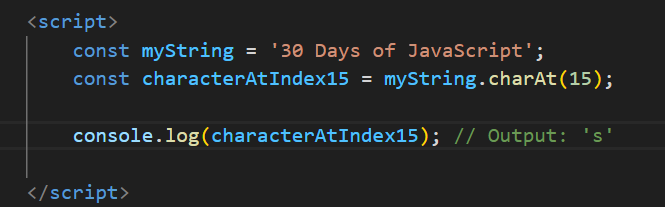
**Q.62 Change all the string characters to lowercase letters using toLowerCase() Method.**

**Ans:** The toLowerCase() method in JavaScript is used to convert all the characters in a string to lowercase:



**Q.63 What is Character at index 15 in ’30 Days of JavaScript’ string? Use charAt() method.**

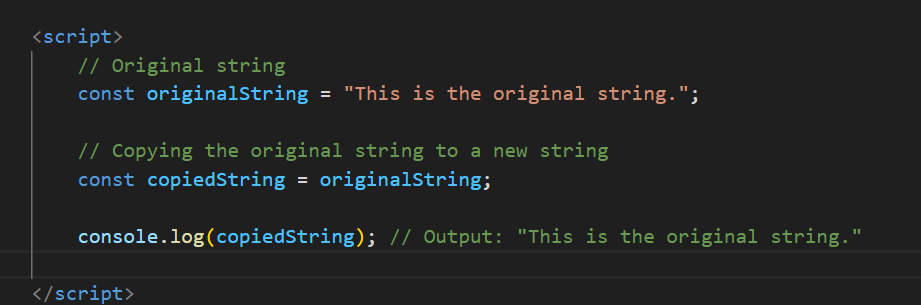
**Ans:-**The character at index 15 in the string '30 Days of JavaScript' using the charAt() method:

****

* The charAt(15) method is called on the myString variable, which returns the character at the specified index, in this case, index 15.
* The character at index 15 in the string '30 Days of JavaScript' is 's', as indicated by the output when logged to the console.

**Q.64 copy to one string to another string in JS?**

**Ans:-** To copy the content of one string to another string in JavaScript, you can simply assign the value of one string variable to another.

****

**In this example:**

* The content of the originalString variable is copied to the copiedString variable by assigning originalString to copiedString.
* Both originalString and copiedString now contain the same content, and changes made to one string will not affect the other.

**Q.65 Find the length of a string without using libraryFunction?**

**Ans:-** You can find the length of a string in JavaScript without using any library function by iterating through each character of the string and counting them.

****

* The getStringLength function takes a string str as its parameter.
* Within the function, a variable length is initialized to 0.
* A for loop iterates through each character of the input string str.
* For each character encountered, the length variable is incremented by 1.
* Once the loop finishes, the final value of length represents the length of the string.
* The length of the string "Hello, World!" is calculated using this function and stored in the lengthOfString variable, which is then logged to the console.

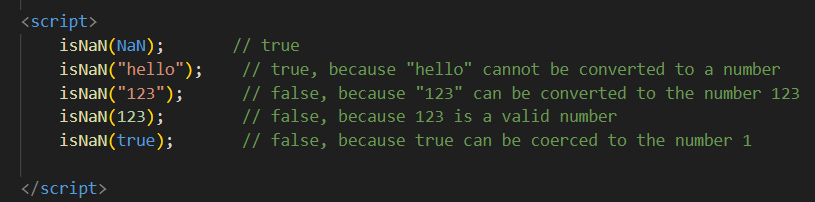
**66.What is JavaScript?**

**Ans:-** JavaScript is a high-level, dynamic, interpreted programming language primarily used for client-side web development. It was created by Brendan Eich while he was working at Netscape Communications Corporation in 1995. JavaScript is often abbreviated as JS.

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

**Ans:-** The isNaN() function in JavaScript is used to determine whether a value is NaN (Not-a-Number) or not. NaN is a special value representing "Not a Number" and is typically returned when a mathematical operation fails or when a value cannot be coerced into a valid number.

* f the argument passed to **isNaN()** is not a number (e.g., a string that cannot be converted to a number), or if it is the NaN value itself, the function returns **true**.
* If the argument is a number or a value that can be coerced into a number, the function returns **false**.



**68. What is negative Infinity?**

**Ans:-** In JavaScript, **Negative Infinity** is a special value that represents the mathematical concept of negative infinity. It is one of the possible values of the global **Infinity** property.

When a number exceeds the upper limit of what JavaScript can represent as a finite number, it becomes **Infinity**. Similarly, when a number goes below the lower limit of what JavaScript can represent as a finite number, it becomes **-Infinity**.

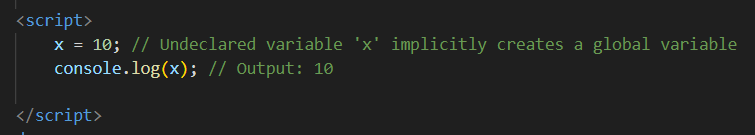
**69.** **Which company developed JavaScript?**

**Ans:-** JavaScript was developed by Brendan Eich while he was working at Netscape Communications Corporation in 1995. Netscape was a leading web browser company at the time, and JavaScript was initially created to add interactivity to web pages viewed in Netscape Navigator, one of the earliest popular web browsers. Later, Netscape submitted JavaScript to the Ecma International standards organization, leading to the standardization of the language as ECMAScript. Today, JavaScript is supported by all major web browsers and is widely used for web development.

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

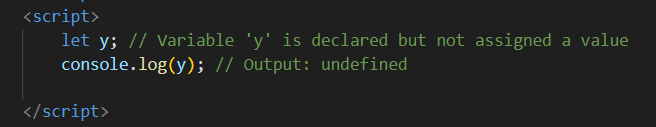
**Ans:- Undeclared Variables:**

* Undeclared variables are those that have not been declared using a var, let, or const keyword before they are used in a program.
* Using an undeclared variable in JavaScript doesn't throw an error by default. Instead, JavaScript implicitly creates a global variable with that name in non-strict mode. In strict mode, using an undeclared variable will result in a ReferenceError.



**Undefined Variables**:

* Undefined variables are those that have been declared, but they have not been assigned a value.
* When you try to access the value of an undefined variable, JavaScript returns the special value **undefined**.



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

**Ans:-** 

This code creates a button, and when clicked, it dynamically adds a new **<div>** element with the class "item" to the container **<div>**. You can customize this code according to your specific requirements, such as adding different types of elements or additional functionality.

**72.What is the difference between ViewState and SessionState?**

**Ans**.The basic difference between the two is that ViewState manages state at the client end, which makes state management easier and quickier for the end-user while SessionState manages state at the server end, making it easier to manage content from this end as well.

1. Viewstate: It is maintained at only one level which is page-level. Changes on other pages will not be visible if changes on one page had been done. Information stored in ViewState is stored only for clients and may not be transferred to any other location. Viewstate is simply a synonym for serializable data.

2. SessionState: It is maintained at session-level and data can be accessed across all pages in the web application. Any individual person can access but if only they have access this server and get the information stored within the server.

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

**Ans.** === is the strict equality operator in Javascript. It is used to compare two variables and check their value and data type. Returns a Boolean result (True or False) if the data types and values of the two variables match.

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

**Ans.** gettElementsByclassName() method is used to return the element in the document with the “className” attribute and the “id” attribute can used to change/append the class of the element

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

**Ans.** Write operation on a file:

After the File System file is imported then, the writeFile() operation is called. The writeFile() method is used to write into the file in JavaScript. The syntax of this method is as follows − writeFile(path,inputData,callBackFunction)

The writeFile() function accepts three parameters-

● Path − The first parameter is the file path or file name to which the input data is to be written. If the file already exists, the contents of the file are deleted and the user-supplied input will be updated, or if the file is not present, the corresponding file will be created at the given path and input. The information is written in it.

● inputData − It contains the data to be written in the file that is opened.

● callBackFuntion − The third parameter is the call back function which takes the error as the parameter and shows the mistake if the write operation fails.

**Reading from the file:**

The reading of the file in JavaScript can be done by using the readFile() function after the File System module is imported.

The syntax to read from a file is as follows –

readFile(path, format, callBackFunc)

The readFile() function accepts three parameters including one optional parameter.

● Path − The first parameter is the path of the test file from which the contents are to read. Only the file name has to be given, if the current location or directory is the same directory where the file which is to be opened and read is located.

● Format − The second parameter is the optional parameter which is the format of the text file. The format can be ASCII, utf-8 etc.

● CallBackFunc − The third parameter takes the error as the parameter and displays the fault is any raised due to the error.

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

**Ans**. JavaScript supports different kinds of loops:

● for - loops through a block of code a number of times

● for/in - loops through the properties of an object

● for/of - loops through the values of an iterable object

● while - loops through a block of code while a specified condition is true

● do/while - also loops through a block of code while a specified condition is correct.

**77.How can you convert the string of any base to an integer in JavaScript? Ans.** The parseInt() function is used to convert an arbitrary radix string to an integer in JavaScript. This function returns the radix integer that is specified in the second argument of the parseInt() function.

parseInt(string, radix)

A radix parameter specifies the number system to use:

2 = binary, 8 = octal, 10 = decimal, 16 = hexadecimal.

If radix is omitted, JavaScript assumes radix 10. If the value begins with "0x", JavaScript assumes radix 16.

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

**Ans**. The delete function is used on object properties. It deletes both value of property and property itself.

**Example code:**

var person = {

firstName:"Parth",

lastName:"Shinde",

age:23,

};

delete person.age; // or delete person["age"];

it will shows person age is undefined.

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

**Ans.** JavaScript has three types of pop-up boxes:

**● Alert Box:** The alert box is used to display a message to the user. It contains a message and an OK button.

**● Confirm Box:** The confirm box is used to ask the user for confirmation before taking an action. It contains a message, a Cancel button, and an OK button. A confirm box is used if you want the user to verify or accept something. When the box pops up, the user will have to press either “OK” or “CANCEL” to proceed. If the user presses “OK” it returns true or if the user presses “CANCEL” it returns false.(window.confirm();).

**● Prompt Box :** The prompt box is used to get input from the user. It contains a message, a text input field, a Cancel button, and an OK button.When a prompt box pops up, the user will have to click either "OK" or "Cancel" to proceed after entering an input value. If the user clicks "OK" the box returns the input value. If the user clicks "Cancel" the box returns null. (window.prompt();).

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

**Ans.**In JavaScript, void(0) is an expression that evaluates to undefined. The void operator takes an argument and returns undefined. In this case, the argument is 0, which has no effect on the expression. We use this to prevent any negative effectson a weird page when we insert some expression. For example, in the case of URL hyperlinks. Hyperlinks open by reloading the page when the user clicks on the link. Javascript is used when you need to run some other code

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

**Ans.** In JavaScript, you can use the window.location object to load a new page. The window.location object contains information about the current URL and provides methods to navigate to a new URL. It is a reference to a Location object that represents the current location of the document. By accessing it we can change the URL of the window.

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

Ans. Performance: Manipulating the innerHTML property can be slower than other DOM manipulation techniques, especially for larger HTML content. This is because setting innerHTML causes the browser to re-parse and re-render the entire content of the affected element.

Accessibility: Manipulating innerHTML can also cause accessibility issues if it is used to dynamically modify important page content, such as headings, links, or form elements. This can make it difficult or impossible for users with disabilities to navigate or interact with the page.

Event listeners: If an element has event listeners attached to it, replacing its innerHTML property can remove those event listeners and cause unexpected behavior.

Code maintainability: Manipulating innerHTML can also make code harder to maintain, as it mixes HTML markup with JavaScript code. This can make it difficult to debug or modify the code in the future

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

**Ans:-**

**• Create basic math operation in JS**

**Ans:-**

**• Create result**

**Ans:-**

* **Create a slider using JavaScript**

**Ans:-**

**Advance JavaScript**

**MODULE: 1 (Introduction and Code Quality)**

**● Write a program to Show an alert**

**Ans:-**Window .alert(‘Any message you can put here’)

**● What will be the result for these expressions?**

**1**. 5>4

**Ans.** True

**2**. “apple” > “pineapple”  
 **Ans.** False

**3**. “2” > “12”

**Ans**. True

**4**.undefined == null;

**Ans**. True

**5**. undefined === null;

**Ans**. False

**6**. null == “\n0\n”

**Ans.** False

**7**. null === “\n0\n”

**Ans.** False

**● Will alert be shown?** if ("0") { alert( 'Hello'); }

Ans:- Yes , if we put “0” in condition it is work as a string and code under this condition would execute. But if we put 0 in condition it is work as a number and code under this condition would never execute.

**• What is the code below going to output? alert( null || 2 || undefined ); Ans.** Result will be 2. Because null and undefined refers to empty string.

**• The following function returns true if the parameter age is greater than 18. Otherwise it asks for a confirmation and returns its result:**

**Ans.** function checkAge(age) { //making function to check age if (age> 18) { return true; } Else { return confirm (‘did parents allow you?'); } // open confirm box }

**• Replace Function Expressions with arrow functions in the code below: Function ask(question, yes, no)**

**{ if (confirm(question))yes();**

**else no(); }**

ask("Do you agree?",

function() { alert("You agreed."); },

function() { alert("You canceled the execution."); } }

**Ans.** Function ask(question,yes,no)

{ if (confirm(question)) yes();

else no();

}

Ask(“do you agree?”,

()=>{alert("You agreed."); },

()=>{alert("You canceled the execution."); })

**MODULE: 2 (Data Types and Objects)**

**● Write the code, one line for each action:**

**a) Create an empty object user.**

**Ans:-** let person = new Object(); //Create an object

**b) Add the property name with the value John.**

**Ans**. person.name = “John”; //Make property called name and assign value ‘John’

**c) Add the property surname with the value Smith.**

**Ans:-** **person.surname = “Smith”;** //Make property called surname and assign value ‘Smith’

**d) Change the value of the name to Pete.**

**Ans:-** **person.name = “Pete”;** //change value of property called name and assign value ‘Pete

**e) Remove the property name from the object.**

**Ans:-** **delete person.name;** //Delete property called name

**• Is array copied?**

let fruits = ["Apples", "Pear", "Orange"]; // push a new value into the "copy" let

shoppingCart = fruits; shoppingCart.push("Banana"); // what's in fruits?

alert( fruits.length ); // ?

**Ans.** Yes array is copied

Fruits = ["Apples", "Pear", "Orange","Banana"] //because both variable assigns to the same reference The fruits length will be 4.

**● Map to names**

let john = { name: "John", age: 25 }; let pete = { name: "Pete", age: 30 }; let mary = { name: "Mary", age: 28 }; let users = [ john, pete, mary ]; let names = /\* ... your code \*/ alert( names ); // John, Pete, Mary

**Ans:-**

let john = {name:”John”, age:25};

let pete = {name:”Pete”, age:30};

let mary = {name:”Mary” age:28};

let users = [john,pete,mary];

let names = users.map((item) => {item.name}); // carrying out name value from objects.

alert( names );

**● Map to objects**

**let john = { name: "John", surname: "Smith", id: 1 }; let pete = { name: "Pete", surname: "Hunt", id: 2 }; let mary = { name: "Mary", surname: "Key", id: 3 }; let users = [ john, pete, mary ]; let usersMapped = /\* ... your code ... \*/**

**/\* usersMapped = [**

**{ fullName: "John Smith", id: 1 },**

**{ fullName: "Pete Hunt", id: 2 },**

**{ fullName: "Mary Key", id: 3 }**

**]**

**\*/ alert( usersMapped[0].id ) // 1 alert( usersMapped[0].fullName ) // John Smith**

**Ans.** let usersMapped = users.map((user) => ({

fullname: `${user.name} ${user.surname}`,

id: user.id

}));

**● Sum the properties There is a salaries object with arbitrary number of salaries. Write the function sumSalaries(salaries) that returns the sum of all salaries using Object.values and the for..of loop.If salaries is empty, then the result must be 0. let salaries = {**

**"John": 100,**

**"Pete": 300,**

**"Mary": 250**

**};**

**alert( sumSalaries(salaries) ); // 650**

**Ans.** let salaries = {

"John": 100,

"Pete": 300,

"Mary": 250, };

function sumSalaries(salaries) {

var sum = 0; //made sure that sum will be 0

for (const salary of Object.values(salaries))

sum += salary; //addition of salaries }

return sum; //Output as sum }

alert( sumSalaries(salaries) );

**● Destructuring assignment We have an object: Write the Destructuring assignment that reads:**

a) Name property into the variable name.

b) Year’s property into the variable age.

c) isAdmin property into the variable isAdmin (false, if no such property)

d) let user = { name: "John", years: 30};

**Ans:**  let user = { name: "John", years: 30 };

let { name } = user // Extracting the 'name' property

let { years: age } = user; // Extracting the 'years' property and assign it to 'age'

let { isAdmin = false } = user; // Extracting the 'isAdmin' property with a default value of 'false'

**● Turn the object into JSON and back Turn the user into JSON and then read it back into another variable. user = { name: "John Smith", age: 35};**

**Ans:-** //JSON.parse converts object into json

let object = JSON.parse(user);

//JSON.stringyfy coverts json into object

let json = JSON.stringify(object);

**MODULE: 3 (Document, Event and Controls)**

**● Create a program to hide/show the password**

**Ans:https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/hide%20%26%20show%20password.html**

**● Create a program that will select all the classes and loop over and whenever i click the button the alert should show**

**Ans:-** **https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/class%20selct%20%26%20alert.html**

**● Create a responsive header using proper JavaScript**

**Ans:-** **https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/responsive%20header.html**

**● Create a form and validate using JavaScript**

**Ans:-** **https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/form%20validation%20.html**

**● Create a modal box using css and Js with three buttons**

**Ans:-** **https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/modal%20box.html**

**● Use external js library to show slider**

**Ans:-** **https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/js%20liberary%20slider.html**

**● Prevent the browser when i click the form submit button**

**Ans:-** **https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/form%20validation%20.html**

**MODULE: 4 (New Request)**

**● What is JSON ?**

**Ans:-** json is stand for javascript Notation . jsion is a storing and trasfring some datebject data are stored and transmitted using key-value pair and array data types. JSON format is text only, which makes easy to read and use with any programming language.

**● What is promises**

**Ans:-** JavaScript Promise Object combination of both the producing code and calls to the inner code Promise Object can be:

JavaScript Promise Object

A JavaScript Promise object contains both the

producing code and calls to the consuming code:

myPromise.then(

  function(value) { /\* code if successful \*/ },

  function(error) { /\* code if some error \*/ }

);

**● Write a program of promises and handle that promises also**

**Ans:https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/promis.html**

**● Use fetch method for calling an api** [**https://fakestoreapi.com/products**](https://fakestoreapi.com/products)

**Ans:-** **https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/fetch%20api.html**

**● Display all the product from the api in your HTML page**

**Ans:-** **https://github.com/premprakashmali/JAVASCRIPT/blob/main/javascript%20task/display%20data%20api.html**