# Module 6) JAVASCRIPT BASIC & DOM

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

**Ans :** JavaScript is a lightweight programming language commonly used by web developers to add dynamic interactions to web pages, applications, servers, and even games. It works seamlessly alongside HTML and CSS, complementing CSS in formatting HTML elements while providing user interaction, a capability that CSS alone lacks.

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

**Ans :**

1. var: Historically, var was the original way to declare variables in JavaScript. Variables declared with var are function-scoped or globally scoped, meaning their scope is limited to the function where they are declared or to the global scope if declared outside of any function.
2. let: Introduced in ECMAScript 6 (ES6), let allows you to declare block-scoped variables. Block-scoped variables are limited in scope to the block, statement, or expression on which it is used, rather than the entire function. This helps in avoiding issues related to variable hoisting and unintended variable access.
3. const: Also introduced in ECMAScript 6 (ES6), const allows you to declare variables whose values are constant and cannot be re-assigned. Like let, const is also block-scoped. Once a value is assigned to a const variable, it cannot be changed or re-declared.



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

**Ans :** In JavaScript, data types represent the kind of values that can be used and manipulated within a program. JavaScript is a loosely-typed language, meaning variables can hold values of any data type without explicit declaration. Here are the main data types in JavaScript

1. **Primitive Data Types:**

* String: Represents textual data, enclosed within single (' '), double (" "), or backtick (``) quotes.
* Number: Represents numeric values, including integers, floating-point numbers, and special numeric values like NaN (Not-a-Number) and Infinity.
* Boolean: Represents a logical value, either true or false.
* Undefined: Represents a variable that has been declared but has not been assigned a value. Variables are automatically initialized to undefined.
* Null: Represents the intentional absence of any value.

1. **Non Primitive Data Types :**

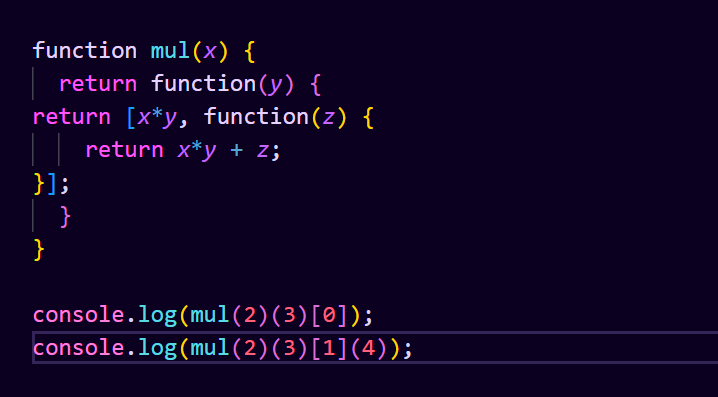
* Object: Objects are complex data types that can store collections of data and functionalities. They are composed of key-value pairs, where keys are strings (or Symbols) and values can be of any data type, including other objects, functions, and primitive values. Objects are used extensively in JavaScript for organizing and manipulating data.
* Array: Arrays are specialized objects in JavaScript that store collections of elements, typically ordered by numeric indices. They have properties and methods specifically designed for working with ordered collections of data.
* Function: Functions in JavaScript are also objects, but they have the additional ability to be invoked. They can be assigned to variables, passed as arguments to other functions, and returned from other functions. Functions are commonly used to encapsulate reusable pieces of code and perform tasks.

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

**Ans :** The MUL function is a miniature of the multiplication function. In this function, we call the function that required an argument as a first number, and that function calls another function that required another argument and this step goes on.

The first function’s argument is x, the second function`s argument is y and the third is z, so the return value will be xyz.

There's nothing more to it than that. Which way do you mean? In javascript, a function can be a return value from another function. In this case mul is returning a function which in case returns an array. The first element in the array is x\*y and the second element is a function.



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

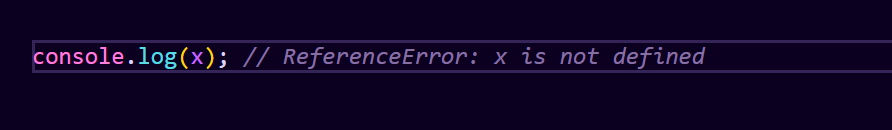
**Ans :** In JavaScript Undeclared − It occurs when a variable which hasn’t been declared using var, let or const is being tried to access. Undefined − It occurs when a variable has been declared using var, let or const but isn’t given a value. Following is the code for undeclared and undefined in JavaScript

1. **Undeclared:**

An undeclared variable is one that has been not been declared using the var, let, or const keywords.

If you try to access the value of an undeclared variable, it will result in a ReferenceError.

For Example :

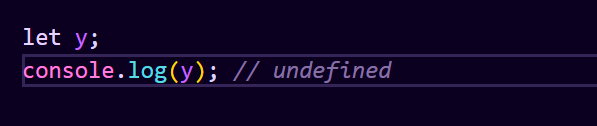


1. **Undefined:**

Undefined is a special value in JavaScript that represents the absence of a value or an uninitialized variable.

If a variable is declared but not assigned a value, or if a function does not explicitly return a value, its value is undefined.

For Example :



In this case, y is declared but has not been assigned a value, so its value is undefined.

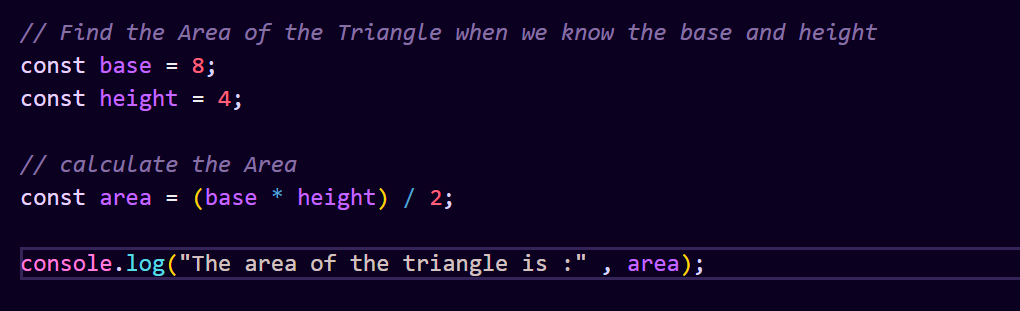
**Q.6 Using console.log() print out the following statement: The quote 'There is no exercise better for the heart than reaching down and lifting people up.' by John Holmes teaches us to help one another. Using console.log() print out the following quote by Mother Teresa:Ans :**

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

**Ans :** In our case 10 == "10" after type conversion both sides of the comparison are equal so it returns true. After type conversion conversion of which side to which type? == means check for values. === means check for values as well as type. so when comparing 10 =='10', javascript engine only checks for values, hence true

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

**Ans :** In geometry, the area of a triangle is calculated by taking half the product of the base and the height of the triangle. In JavaScript, it is possible to write a program that calculates the area of a triangle based on user input. This can be useful for creating applications that require users to input geometric data and perform calculations with that data. In this tutorial, we will learn how to write a JavaScript program that calculates the area of a triangle.

****

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

**Ans :**

****

We define a function daysUntilChristmas that calculates the number of days left until the next Christmas.

Inside the function, we check if Christmas has already passed this year. If so, we adjust the target date to Christmas of the next year.

We calculate the difference in milliseconds between today and Christmas and convert it to days.

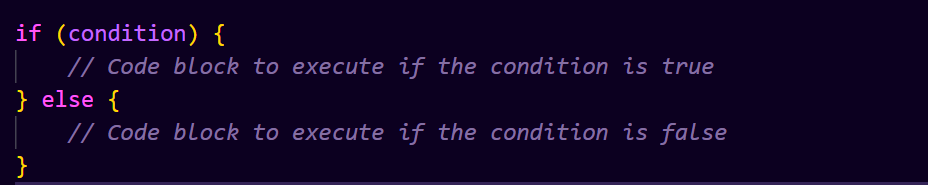
Finally, we use if-else statements to output the number of days left until Christmas, with proper grammar for singular and plural forms.

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

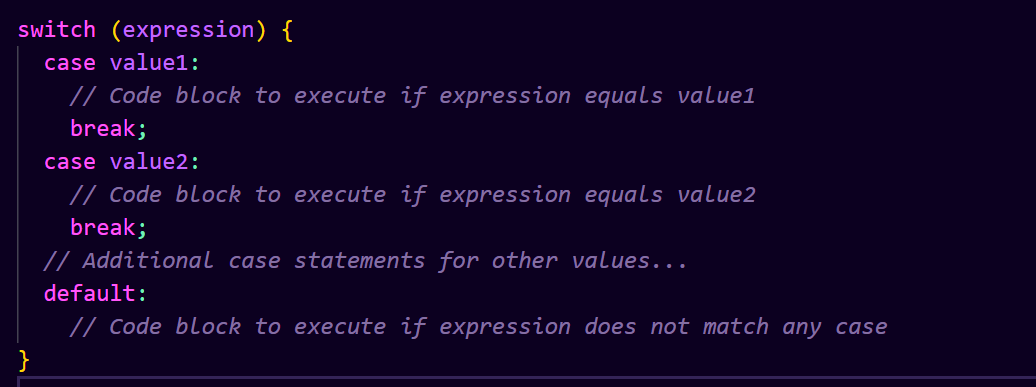
**Ans :**

A condition statement, also known as a conditional statement or control structure, is a programming construct that allows a program to execute different sequences of code based on whether a specified condition evaluates to true or false. Condition statements are fundamental to implementing decision-making logic in computer programs.

In most programming languages, including JavaScript, condition statements typically take the form of an if statement, which has the following basic structure:



The switch statement is another type of conditional statement in JavaScript. It provides a way to compare a single value against multiple possible conditions and execute different blocks of code based on the matched condition. The switch statement has the following syntax:



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

**Ans :**

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

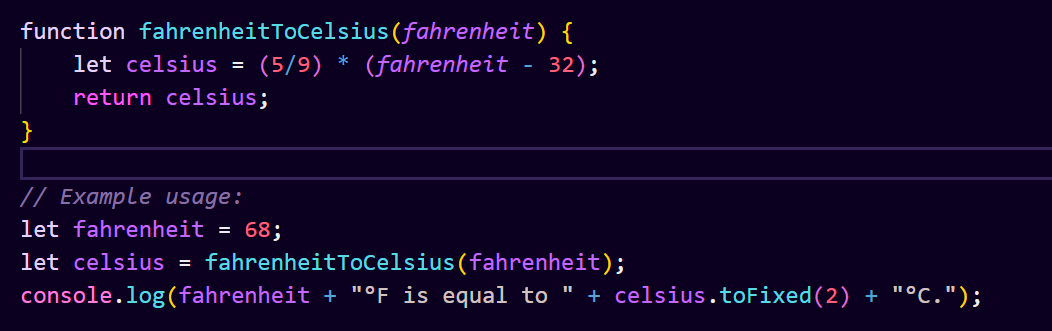
**Ans :**

****

These functions assume that a year has 365 days. In reality, a year can have 365 or 366 days, depending on whether it's a leap year. If you need more accurate calculations, you can adjust the functions accordingly.

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

**Ans :**

****

In this function:

We first calculate the temperature in Celsius using the formula.

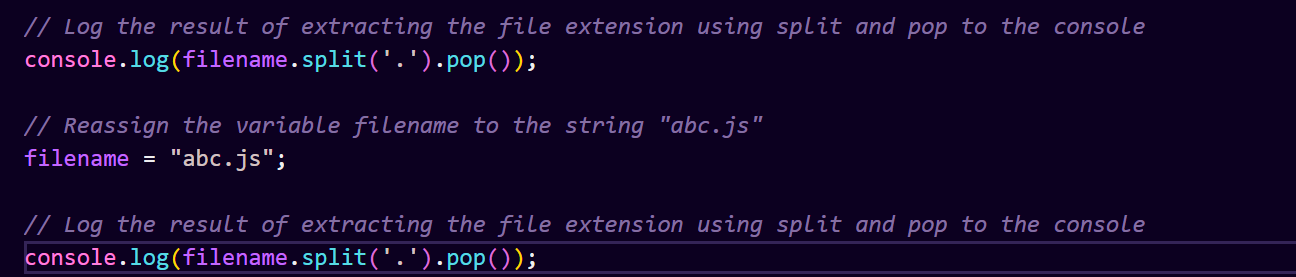
We then return the result.

The toFixed(2) method is used to round the result to two decimal places for better readability.

You can call this function with any temperature in Fahrenheit, and it will return the equivalent temperature in Celsius.

**Q.14 Write a JavaScript exercise to get the extension of a filename.?**

**Ans :** Using these extensions, you will understand what types of files your website is handling. These are the 3 ways to get file extension from a filename in JavaScript: Using split () and pop () methods in JavaScript. Get file extensions with substring () and lastIndexOf () methods. Using regular expression ( Regex) to find the file extension.

****

**Q.15 What is the result of the expression (5 > 3 && 2 < 4)?**

**Ans :**

The expression (5 > 3 && 2 < 4) is a logical AND operation. In JavaScript, the && operator returns true if both of its operands are true, otherwise it returns false.

Let's evaluate the expression:

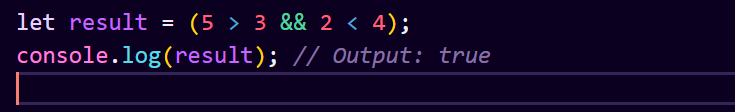
5 > 3 evaluates to true because 5 is greater than 3.

2 < 4 evaluates to true because 2 is less than 4.

Now, applying the logical AND operator:

Both operands are true, so the expression (5 > 3 && 2 < 4) evaluates to true.

Therefore, the result of the expression (5 > 3 && 2 < 4) is true.



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

**Ans :**

The expression (true && 1 && "hello") is a logical AND operation. In JavaScript, the && operator returns true if all of its operands are truthy values, otherwise it returns the first falsy value encountered.

Let's evaluate the expression:

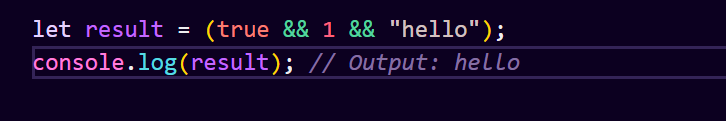
true is a truthy value.

1 is also a truthy value.

"hello" is a truthy value.

Since all operands are truthy values, the expression (true && 1 && "hello") evaluates to the last truthy value encountered, which is "hello".

Therefore, the result of the expression (true && 1 && "hello") is "hello".

****

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

**Ans :**

To evaluate the expression true && false || false && true, we first need to understand the precedence of logical operators in JavaScript.

The logical AND (&&) operator has higher precedence than the logical OR (||) operator. Therefore, expressions are evaluated from left to right, and the && operations are evaluated before the || operation.

Let's break down the expression:

true && false:

The result of true && false is false because both operands must be true for the && operator to return true.

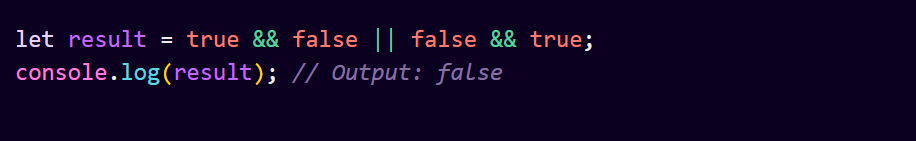
false || false:

The result of false || false is false because one of the operands must be true for the || operator to return true, but both are false.

Now, let's combine the results:

false || false: The result is false.

Therefore, the overall result of the expression true && false || false && true is `false`.



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

**Ans :**

1. **Switch case**

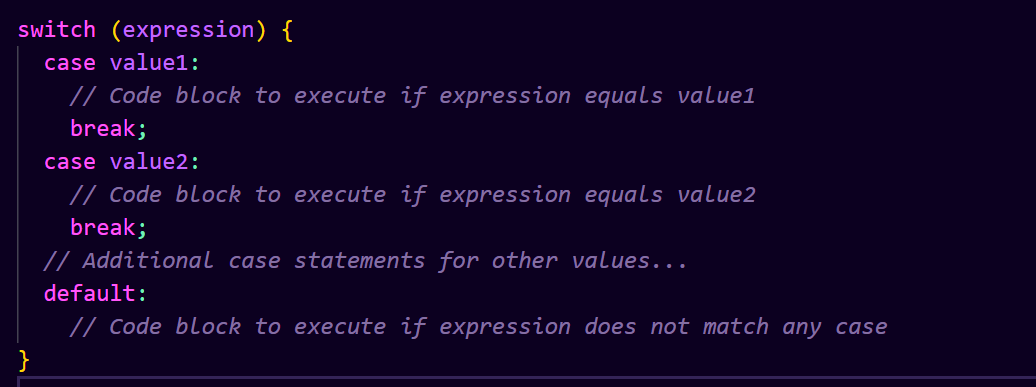
JavaScript switch statement is used to execute a block of code from multiple expressions. JavaScript switch statement evaluates an expression. The expression’s value is compared with the values of each case in the structure. If a match is found, the related block of code is executed.

Usage of JavaScript switch statement

JavaScript switch statement is used with a break or default keyword (optional and both can be used together also).

break: This keyword is used to break out of the switch block. This stops the execution inside the code block.

default: This keyword is used to specify a piece of code if no case matches the given condition. There can be only one default keyword in a switch statement.



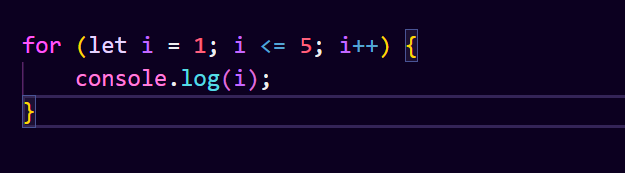
1. **Loop:**

A loop is a programming construct that allows you to execute a block of code repeatedly until a specified condition is met.

There are several types of loops in JavaScript, including for, while, and do-while loops.

Loops are used when you need to perform an operation multiple times without writing the same code repeatedly.

Here's an example of a for loop that prints numbers from 1 to 5:



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

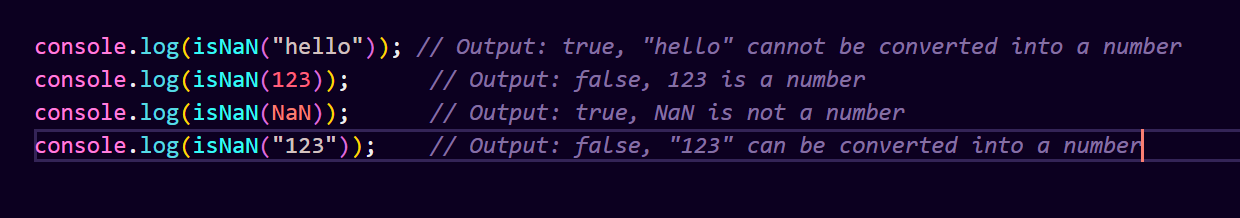
**Ans :** In JavaScript, the isNaN() function is used to determine whether a value is NaN (Not-a-Number) or not. NaN is a special value representing "Not-a-Number" and is returned when a mathematical operation cannot produce a meaningful result.

The isNaN() function takes one argument and returns true if the argument is NaN or cannot be converted into a number; otherwise, it returns false.Here's how the isNaN() function works:

If the argument passed to isNaN() is not a number (or cannot be converted into a number), it returns true.

If the argument is a number or can be converted into a number, it returns false.

Example usage of isNaN():



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

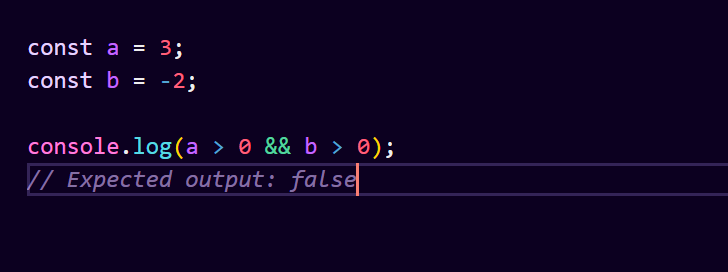
**Ans :**

1. **Logical AND (&&)**

The logical AND (&&) (logical conjunction) operator for a set of boolean operands will be true if and only if all the operands are true. Otherwise it will be false.

More generally, the operator returns the value of the first falsy operand encountered when evaluating from left to right, or the value of the last operand if they are all truthy.

Snytax : x && y

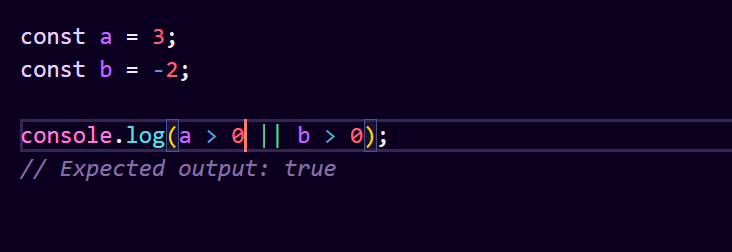


Even though the && operator can be used with non-Boolean operands, it is still considered a boolean operator since its return value can always be converted to a boolean primitive. To explicitly convert its return value (or any expression in general) to the corresponding boolean value, use a double NOT operator or the Boolean constructor.

1. **Logical OR (||)**

The logical OR (||) (logical disjunction) operator for a set of operands is true if and only if one or more of its operands is true. It is typically used with boolean (logical) values. When it is, it returns a Boolean value. However, the || operator actually returns the value of one of the specified operands, so if this operator is used with non-Boolean values, it will return a non-Boolean value.

Snytax : x || y



Even though the || operator can be used with operands that are not Boolean values, it can still be considered a boolean operator since its return value can always be converted to a boolean primitive. To explicitly convert its return value (or any expression in general) to the corresponding boolean value, use a double NOT operator or the Boolean() constructor.

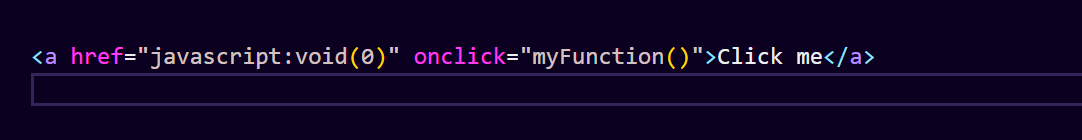
**Q.21 What is the use of Void (0)?**

**Ans :** The void operator evaluates the given expression and then returns undefined. The void operator is often used merely to obtain the undefined primitive value, usually using “ void (0) ” (which is equivalent to “ void 0 ”).

The void operator is used to evaluate an expression and returns the undefined. Generally, this operator is used for obtaining the undefined primitive value. It is often used with hyperlinks. Usually the browser refreshes the page or loads a new page on clicking a link. The javascript:void(0) can be used when we don't want to refresh or load a new page in the browser on clicking a hyperlink.

We can use the operand 0 in two ways that are void(0) or void 0. Both of the ways work the same. The JavaScript:void(0) tells the browser to "do nothing" i.e., prevents the browser from reloading or refreshing the page. It is useful when we insert links that have some important role on the webpage without any reloading. So, using void(0) on such links prevents the reloading of the page but allows to perform a useful function such as updating a value on the webpage.

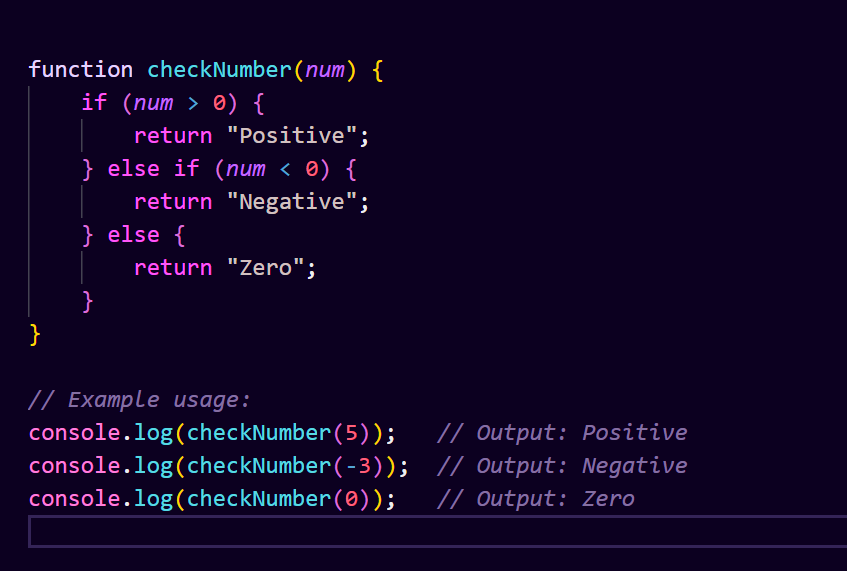
Here's an example of how the void operator can be used in a hyperlink:



In this example, clicking the hyperlink will execute the JavaScript function myFunction(), but it won't cause the browser to navigate anywhere. This can be useful for creating interactive elements on a webpage without affecting the page's navigation behavior.

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

**Ans :** To check whether a number is positive, negative, or zero in JavaScript, you can use a simple conditional statement, such as an if-else statement. Here's how you can do it:

****

In this code:

The function checkNumber takes a parameter num representing the number to be checked.

Inside the function, it checks if num is greater than 0. If true, it returns "Positive".

If num is less than 0, it returns "Negative".

If neither of the above conditions is true, it means num is equal to 0, so it returns "Zero".

You can call this function with any number, and it will return whether the number is positive, negative, or zero.

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

**Ans :** To determine whether a character is a vowel or not in JavaScript, you can create a function that checks if the character is included in a list of vowels. Here's how you can do it:

****

In this code:

The function isVowel takes a parameter char representing the character to be checked.

The character is converted to lowercase using toLowerCase() to handle both uppercase and lowercase vowels.

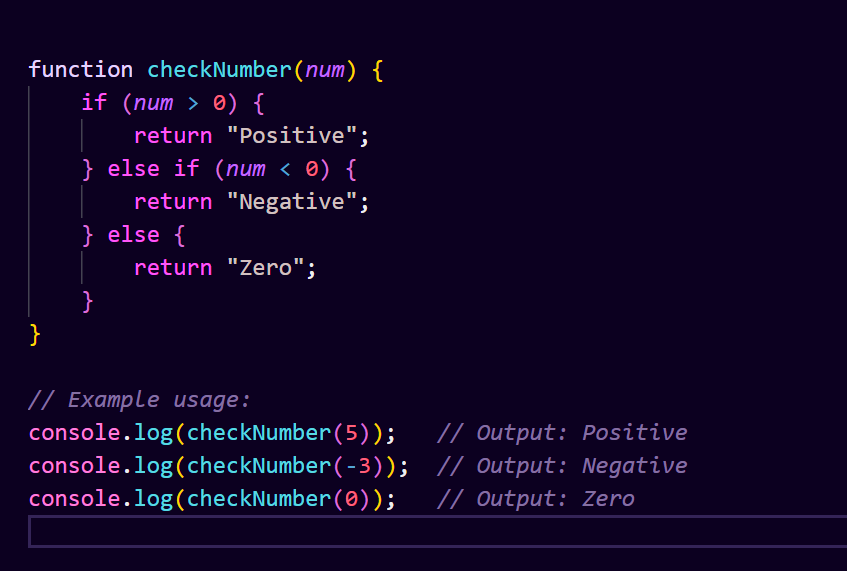
A list of vowels is defined as an array.

The includes() method is used to check if the character is included in the list of vowels.

If the character is found in the list of vowels, the function returns true; otherwise, it returns false.

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

**Ans :**

****

In this code:

The function checkNumber takes a parameter num representing the number to be checked.

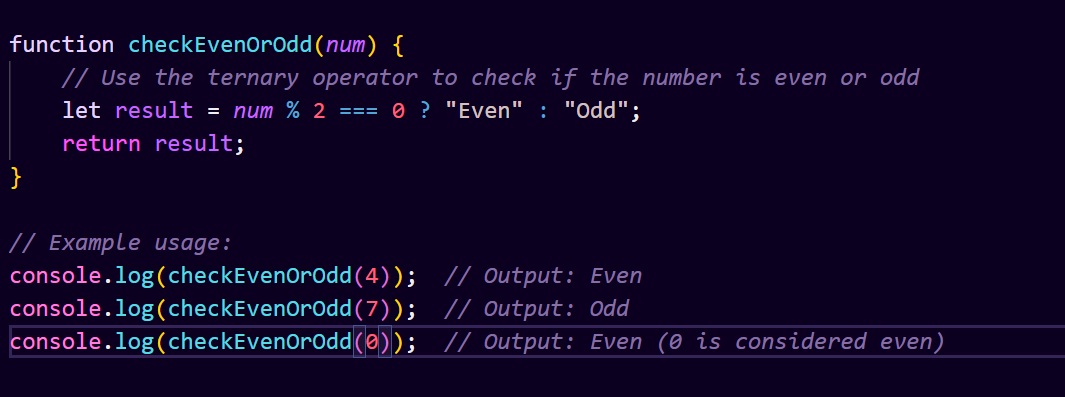
Inside the function, it checks if num is greater than 0. If true, it returns "Positive".

If num is less than 0, it returns "Negative".

If neither of the above conditions is true, it means num is equal to 0, so it returns "Zero".

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

**Ans :**

****

In this code:

The function checkEvenOrOdd takes a parameter num representing the number to be checked.

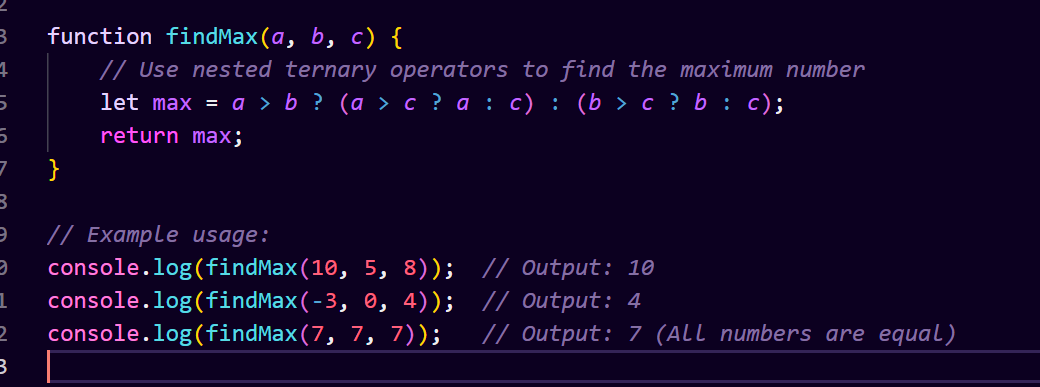
It uses the ternary operator (condition ? expression1 : expression2) to check if num % 2 === 0. If true, it returns "Even"; otherwise, it returns "Odd".

The result is then stored in the variable result and returned from the function.

You can call this function with any number, and it will return whether the number is even or odd.

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

**Ans :**

****

In this code

The function findMax takes three parameters a, b, and c representing the three numbers to be compared.

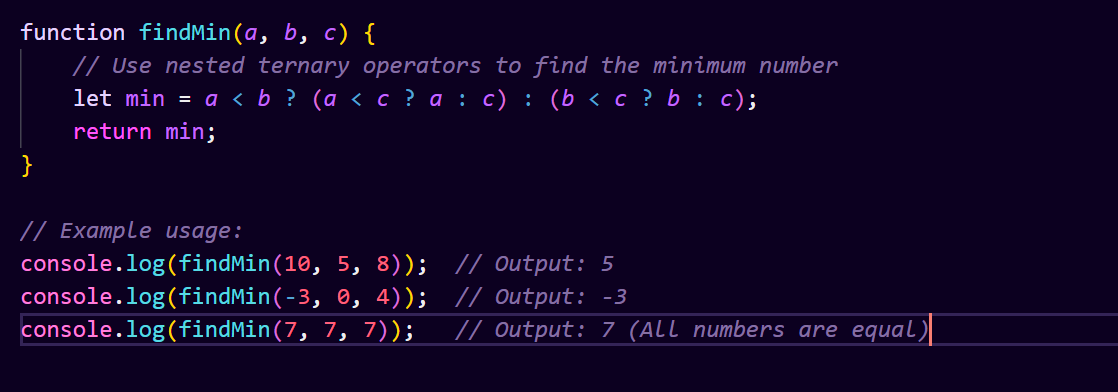
It uses nested ternary operators to compare a with b and c. If a is greater than b, it then compares a with c to find the maximum among a, b, and c. Otherwise, it compares b with c.

The maximum number is then stored in the variable max and returned from the function.

You can call this function with any three numbers, and it will return the maximum number among them.

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

**Ans :**



In this code:

The function findMin takes three parameters a, b, and c representing the three numbers to be compared.

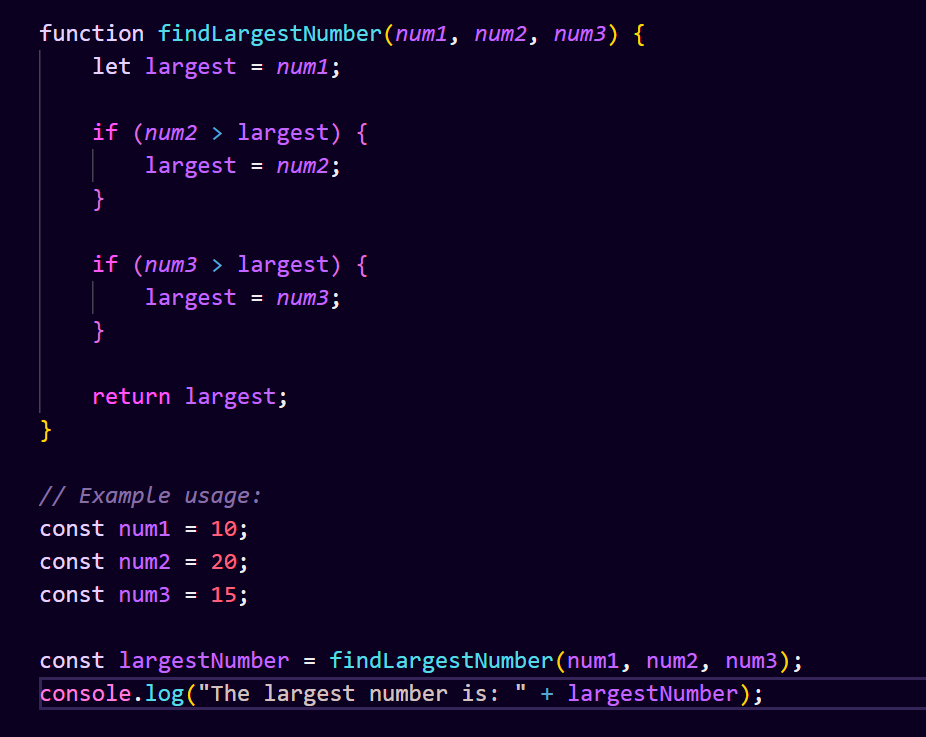
It uses nested ternary operators to compare a with b and c. If a is less than b, it then compares a with c to find the minimum among a, b, and c. Otherwise, it compares b with c.

The minimum number is then stored in the variable min and returned from the function.

You can call this function with any three numbers, and it will return the minimum number among them.

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

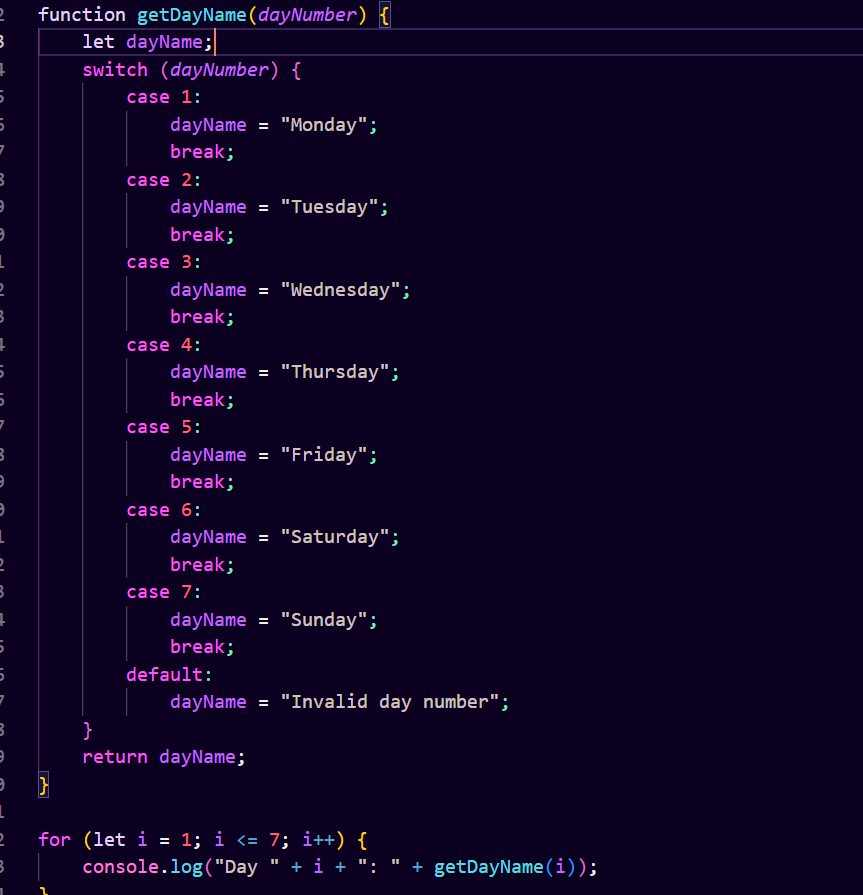
**Ans :** Below are the approaches to finding the largest of three numbers using JavaScript: This is a straightforward approach using if-else statements to compare the numbers and find the largest one. Example: In this example, we are using a Conditional Statement (if-else). The Math.max () method can be used to find the maximum of a list of numbers.

****

**Q.29 Write to show**

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

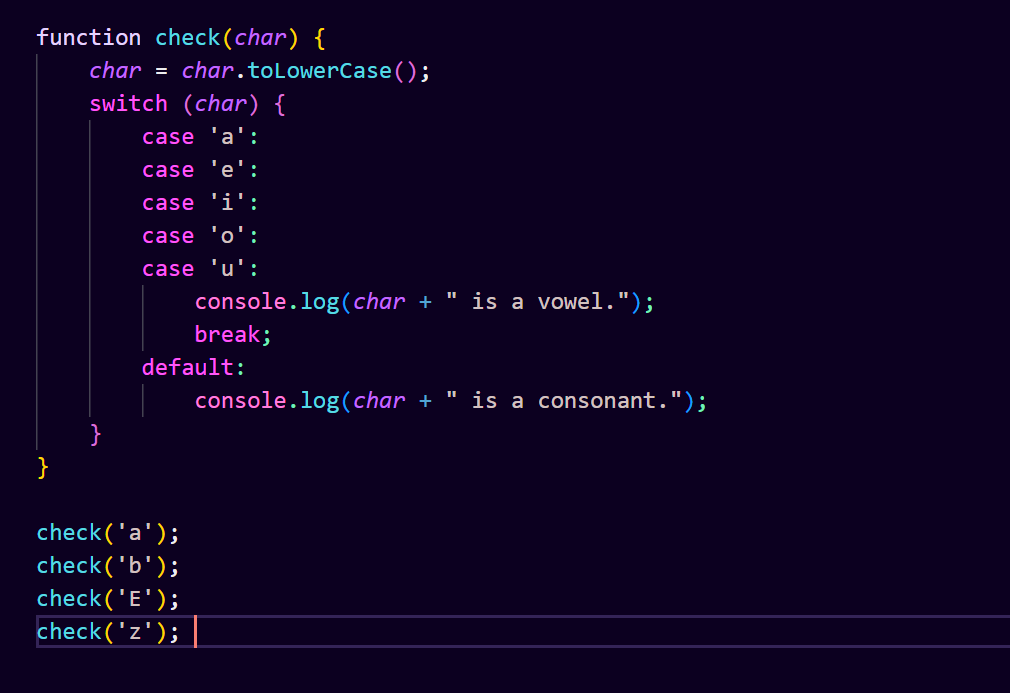
**Ans :**

****

This function takes a numeric value representing the day of the week (1 for Monday, 2 for Tuesday, etc.) and returns the corresponding name of the day. The switch-case statement checks the value of dayNumber and assigns the appropriate day name to the dayName variable. If the input is not within the range of 1 to 7, it returns "Invalid day number".

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

**Ans :**

****

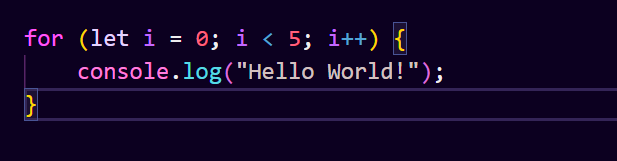
in this function, we take a character as input and use a switch-case statement to check whether it's a vowel ('a', 'e', 'i', 'o', 'u') or not. If it's one of these vowels, we print that it's a vowel. Otherwise, we print that it's a consonant. uppercase and lowercase inputs.

## (Conditional looping logic Question)

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

**Ans :** JavaScript Loops are powerful tools for performing repetitive tasks efficiently. Loops in JavaScript execute a block of code again and again while the condition is true.

For example, suppose we want to print “Hello World” 5 times. This can be done using JS Loop easily. In Loop, the statement needs to be written only once and the loop will be executed 5 times as shown below:



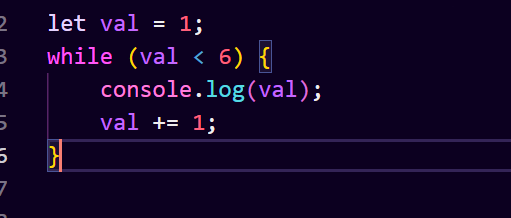
1. **JavaScript For Loop**

Looping in programming languages is a feature that facilitates the execution of a set of instructions repeatedly until some condition evaluates and becomes false. We come across for loop which provides a brief and systematic way of writing the loop structure.

JavaScript for loop is used to iterate the elements for a fixed number of times. JavaScript for loop is used if the number of the iteration is known.

1. **JavaScript While Loop**

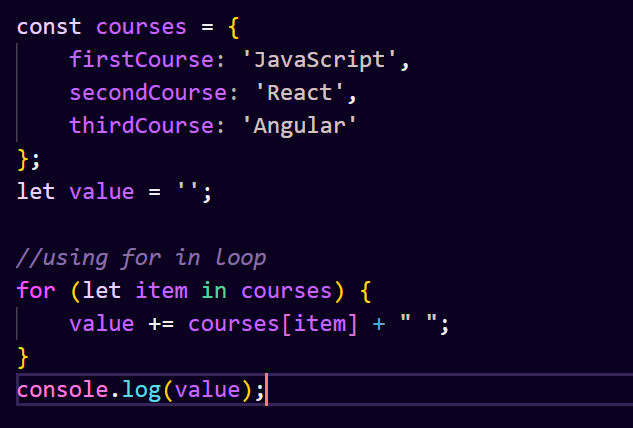
A While Loop in JavaScript is a control flow statement that allows the code to be executed repeatedly based on the given boolean condition. The while loop can be thought of as a repeating if statement. The loop can be used to execute the specific block of code multiple times until it fails to match the condition.



1. **JavaScript for in Loop**

JavaScript for in loop is used to iterate over the properties of an object. JavaScript for in loop iterates only over those keys of an object which have their enumerable property set to “true”.

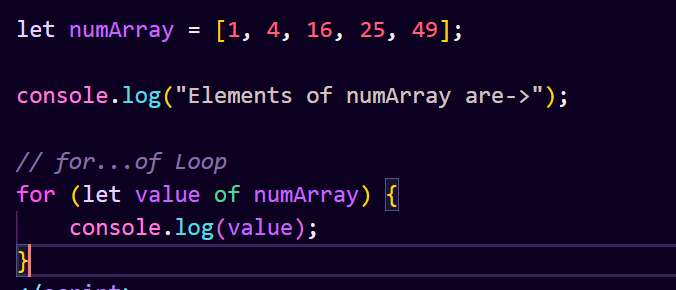
The key values in an object have four attributes (value, writable, enumerable, and configurable). It can be a great debugging tool if we want to show the contents of an object.



1. **JavaScript for…of Loop**

JavaScript for…of statement iterates over the values of an iterable object (like Array, Map, Set, arguments object, …,etc), executing statements for each value of the object.

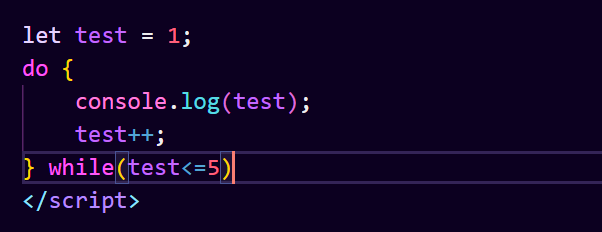
JavaScript for…of loop makes it easy to loop through the elements without needing to handle the index or iteration logic which makes the code short and easier to understand.



1. **JavaScript do…while Loop**

A do… while loop in JavaScript is a control statement in which the code is allowed to execute continuously based on a given boolean condition. It is like a repeating if statement.

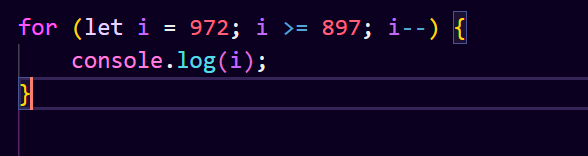
The do…while loop can be used to execute a specific block of code at least once



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

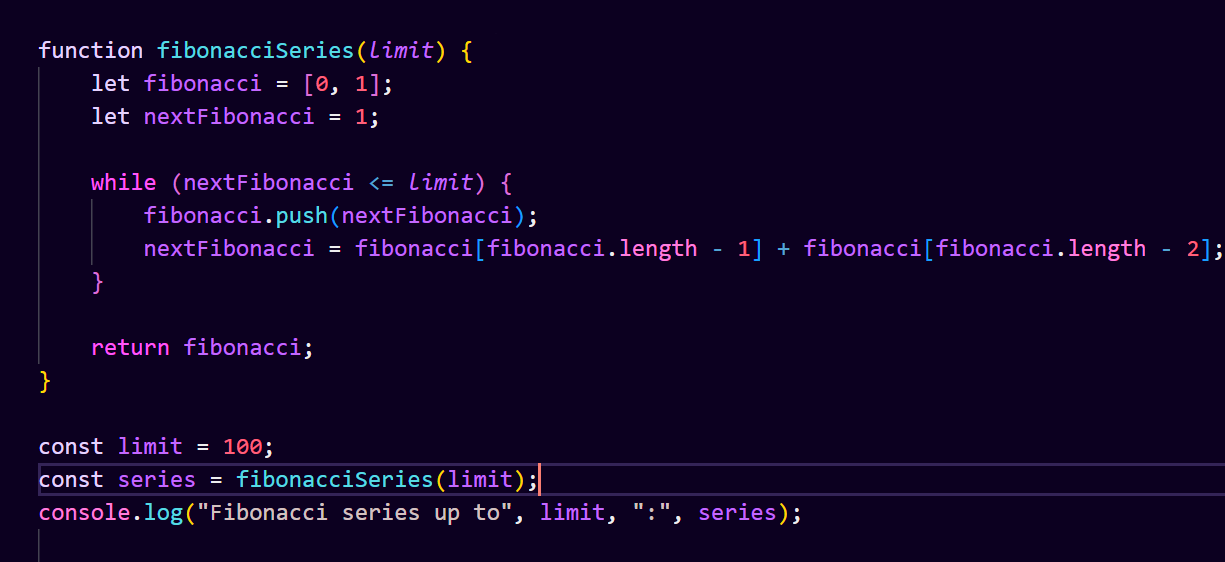
**Ans :** To print the numbers from 972 to 897 using a for loop in JavaScript, you can start the loop from 972 and decrement by 1 until reaching 897. Here's how you can do it:

This loop will print numbers from 972 down to 897, inclusive. Each iteration of the loop will print one number.

****

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

**Ans :**

****

We define a function fibonacciSeries that takes the limit as an argument.

We initialize an array fibonacci with the first two Fibonacci numbers: 0 and 1.

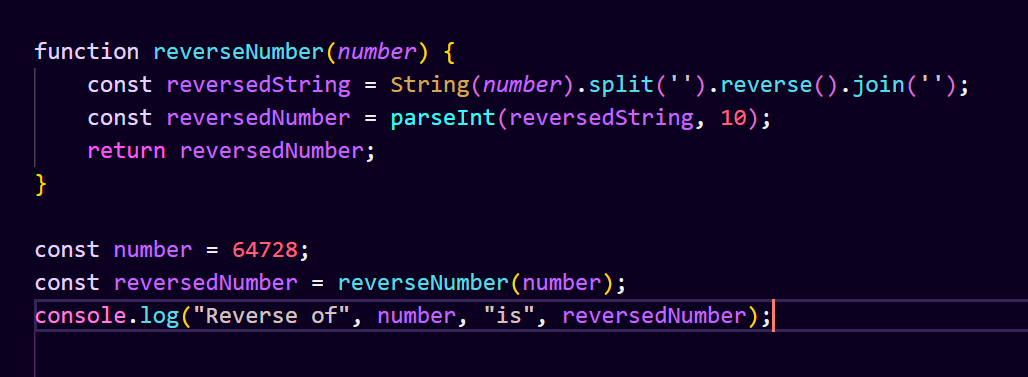
We use a while loop to generate the Fibonacci numbers until the nextFibonacci number exceeds the limit.

Inside the loop, we calculate the next Fibonacci number by adding the last two numbers in the fibonacci array.

Finally, we return the fibonacci array containing the Fibonacci series up to the specified limit.

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

**Ans :**



We define a function reverseNumber that takes a number as an argument.

We convert the number to a string using String(number).

We split the string into an array of characters using split('').

We reverse the array using reverse().

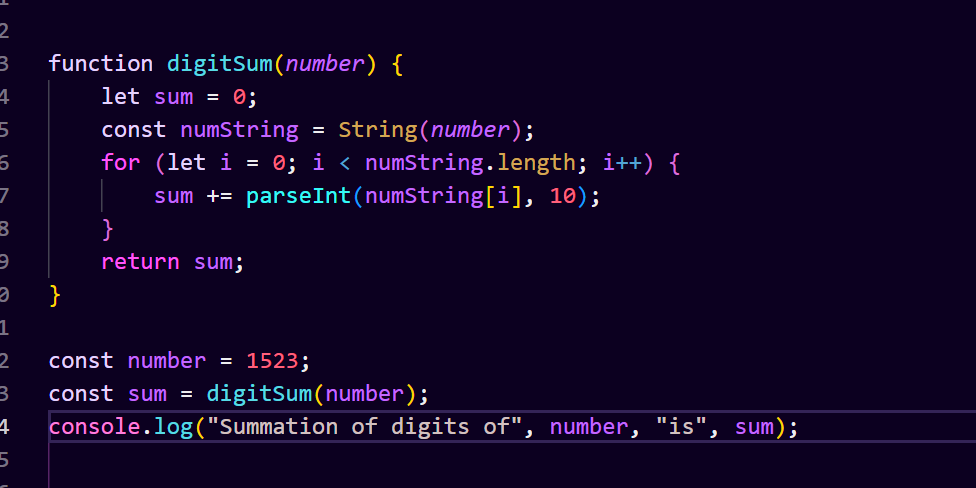
We join the reversed array back into a string using join('').

We convert the reversed string back to a number using parseInt(reversedString, 10).

Finally, we return the reversed number.

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

**Ans :**

****

We define a function digitSum that takes a number as an argument.

We initialize a variable sum to store the summation of the digits.

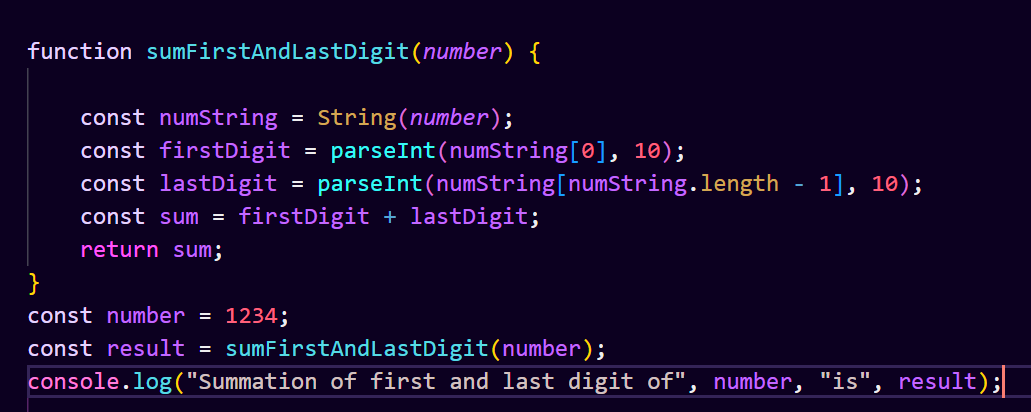
We convert the number to a string to iterate through its digits.

We iterate through each digit of the number using a for loop.

Inside the loop, we convert each digit back to a number using parseInt() and add it to the sum.

Finally, we return the sum.

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



We define a function sumFirstAndLastDigit that takes a number as an argument.

We convert the number to a string using String(number).

We extract the first and last characters of the string using array indexing.

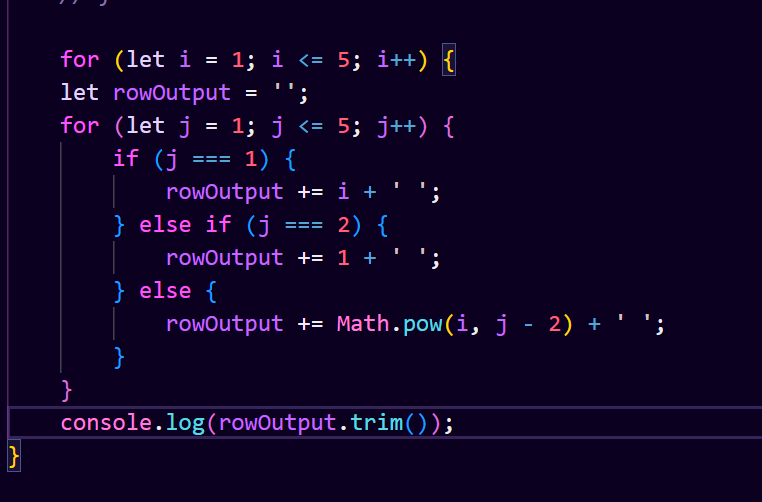
We convert these characters back to numbers using parseInt().

We calculate the sum of the first and last digits.

Finally, we return the sum.

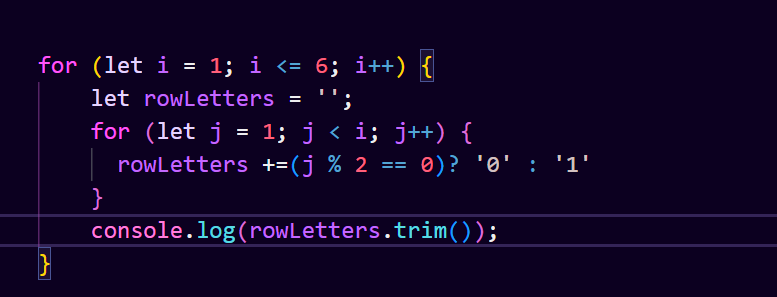
**Q.37 Use console.log() and escape characters to print the following pattern in JS?**

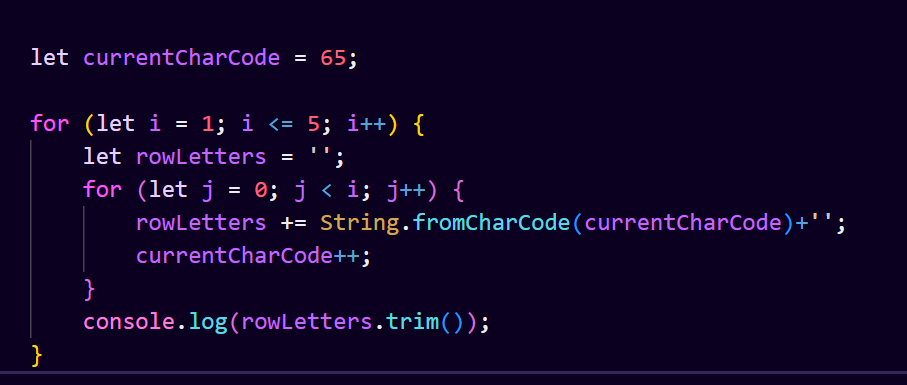
**Ans :**

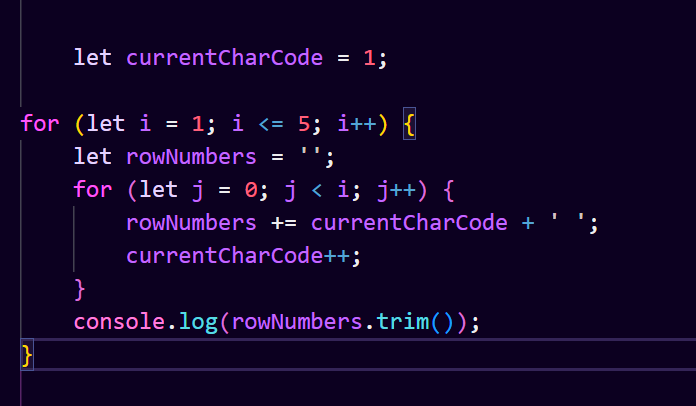
****

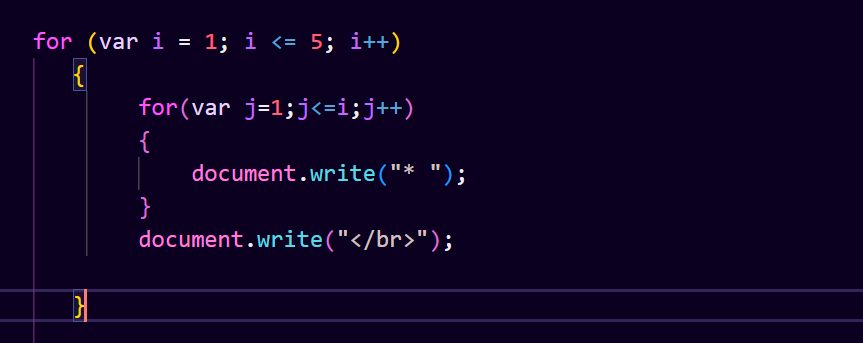
**Q.38 Use pattern in console.log in JS?**

**Ans :**

****

****

****

****

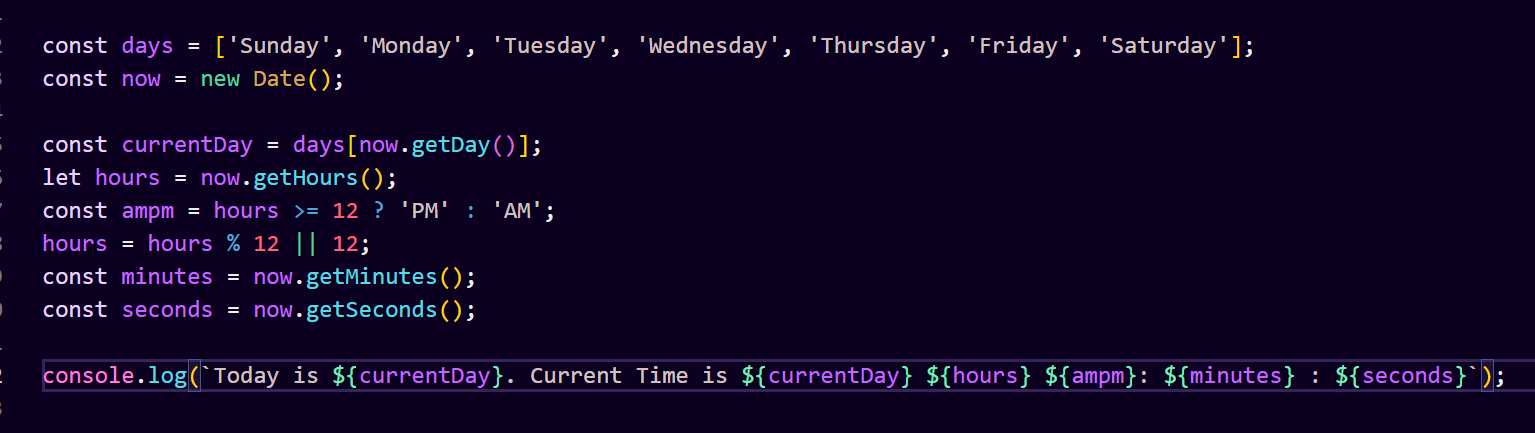
**Q.39 Accept 3 numbers from user using while loop and check each numbers palindrome?**

**Ans :­­**

# (Array and object Question)

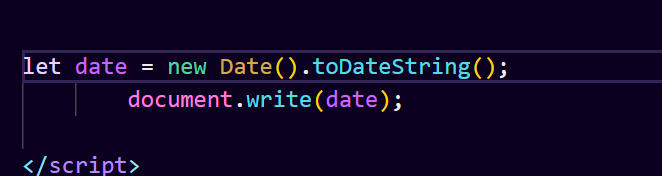
**Q.40 Write a JavaScript Program to display the current day and time in the following format. Sample Output: Today is Friday. Current Time is 12 PM: 12 : 22 2 ?**

**Ans :**

****

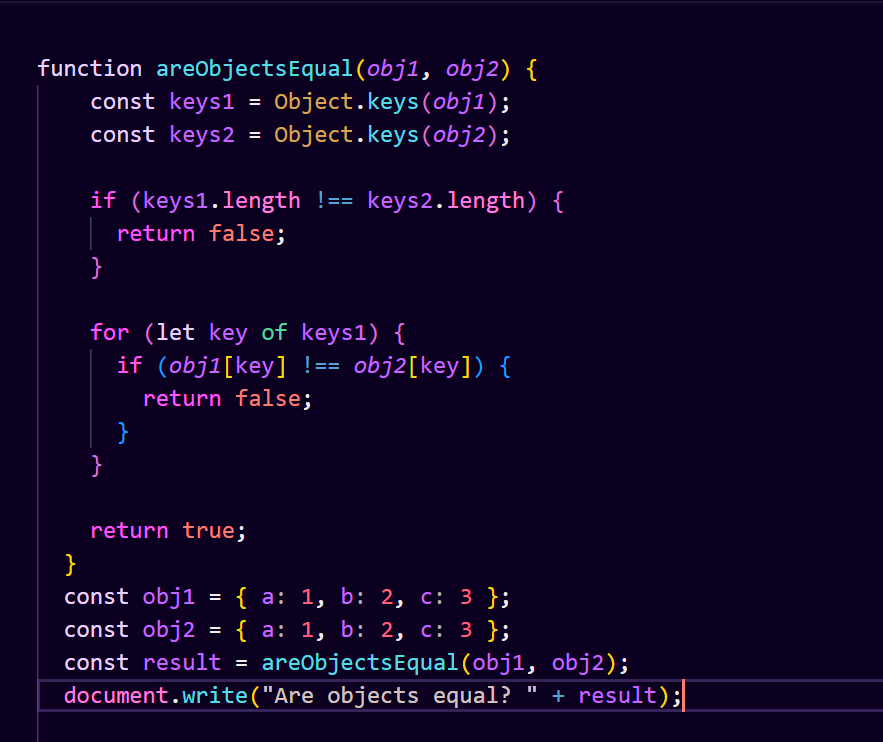
**Q.41 Write a JavaScript program to get the current date?**

**Ans :**

****

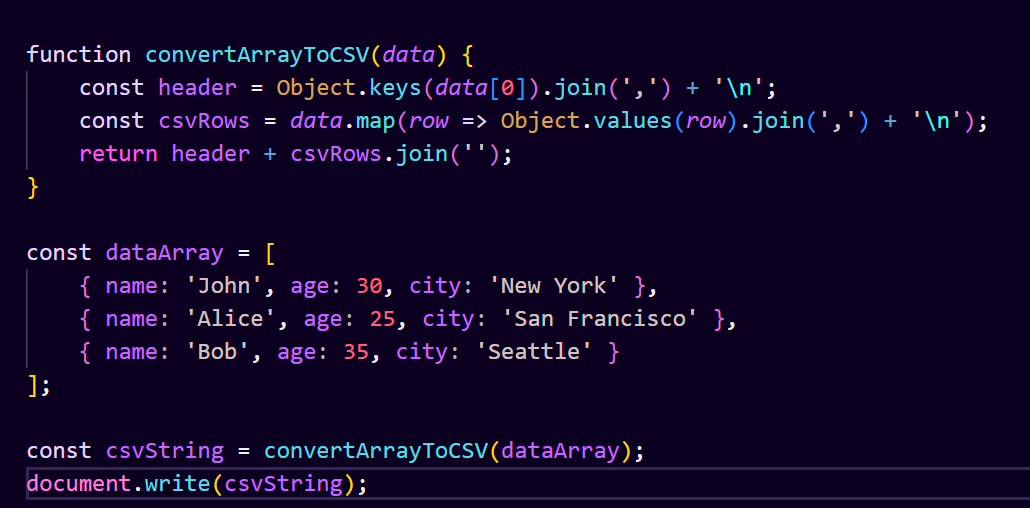
**Q.42 Write a JavaScript program to compare two objects?**

**Ans :**

****

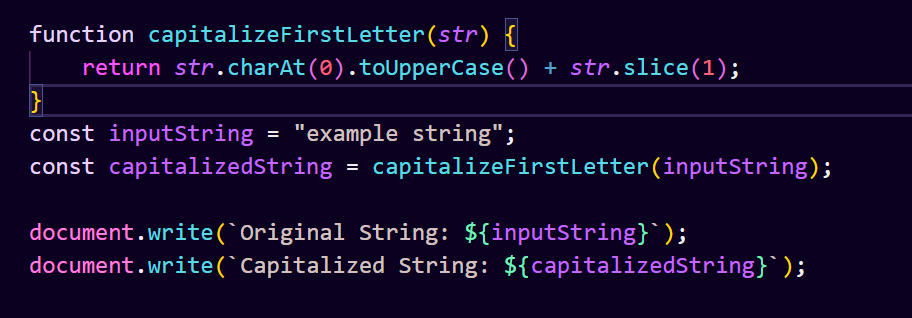
**Q.43 Write a JavaScript program to convert an array of objects into CSV string?**

**Ans :**

****

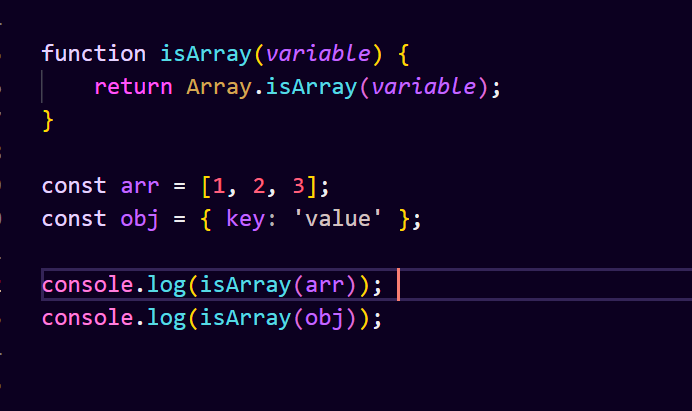
**Q.44 Write a JavaScript program to capitalize first letter of a string?**

**Ans :**

****

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

**Ans :**

****

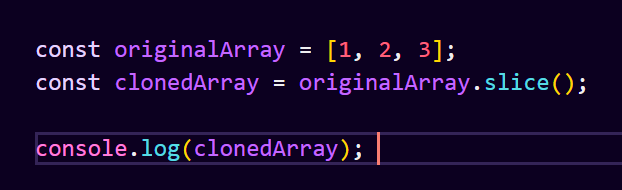
The isArray() function takes a variable as input and checks if it is an array using the Array.isArray() method.

If the variable is an array, isArray() returns true; otherwise, it returns false.

Example usage demonstrates checking whether an array (arr) and an object (obj) are arrays.

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

**Ans :**

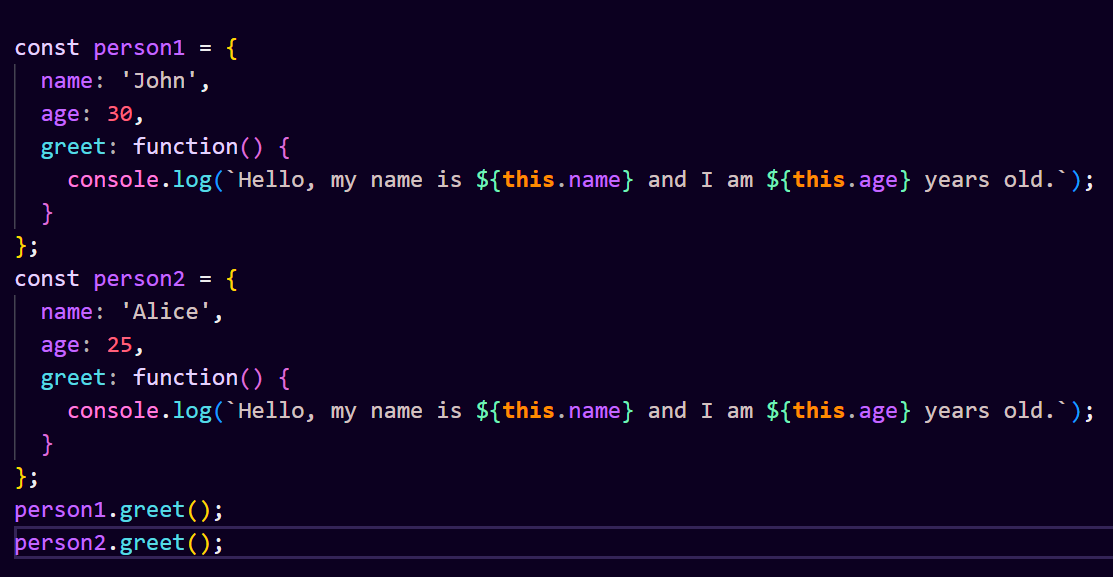
****

This code creates a new array clonedArray that contains the same elements as originalArray by using the slice() method without any arguments, effectively cloning the array.

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

**Ans :** One drawback of declaring methods directly in JavaScript objects is that it can lead to code duplication and inefficiency, especially when multiple instances of the object are created.

When a method is declared directly in an object, it gets created separately for each instance of the object. This means that each object instance will have its own copy of the method, consuming more memory than necessary.

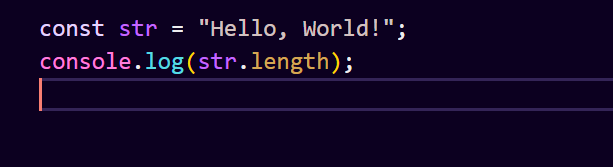


In this example, both person1 and person2 have their own greet() method, even though the implementation is exactly the same. This results in code duplication.

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

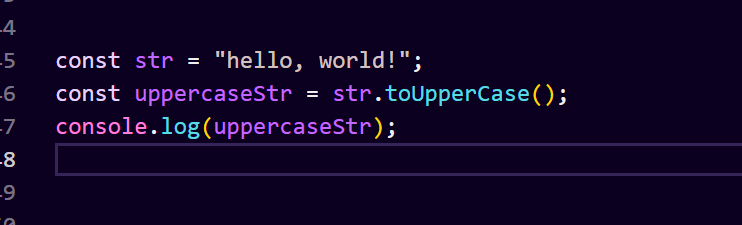
**Ans :** To print the length of a string on the browser console using console.log(), you can simply use the length property of the string. Here's how you can do it

When you run this code in a browser environment and check the console, it will display the length of the string "Hello, World!".



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

**Ans :**

****

We have a string "hello, world!".

We use the toUpperCase() method on the string to convert all characters to uppercase.

The resulting uppercase string is stored in the variable uppercaseStr.

We then print uppercaseStr to the console, which will display "HELLO, WORLD!".

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

**Ans :**

Declaring methods directly in JavaScript objects, especially in the context of constructor functions or object literals, has some drawbacks

1. Memory Consumption:- When you declare methods directly in the object, each instance of the object will have its own copy of the methods. This can lead to increased memory consumption, especially if you have many instances of the object. If the methods don't rely on instance-specific data, it's more memory-efficient to attach them to the prototype.

2. Inefficiency for Repeatedly Created Objects:- If you create multiple instances of an object with methods declared directly in the object, JavaScript has to allocate memory for the methods every time a new instance is created. This can be less efficient compared to attaching methods to the prototype, where the methods are shared among all instances.

3. Difficulty in Achieving Method Overriding :- When methods are declared directly in an object, it can be challenging to achieve method overriding or inheritance, especially if you want to extend the functionality of an existing method. Prototypal inheritance provides a more flexible way to handle method overriding.

4. Readability and Maintainability :- As your object grows, declaring methods directly in the object may lead to code that is less readable and maintainable. Separating the definition of methods from the object creation can enhance code organization and readability.

**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?**

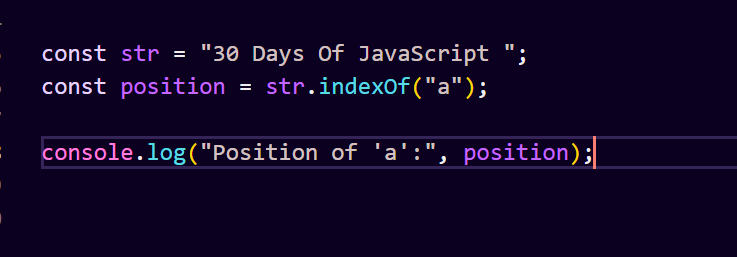
**Ans :**

****

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

**Ans :**

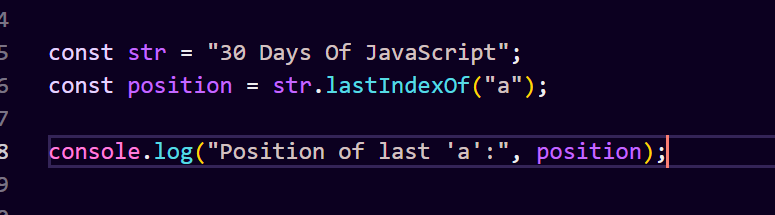
You can use the indexOf() method to determine the position of the first occurrence of a specific substring within a string. Here's how you can use it to find the position of the first occurrence of the letter "a" in the string "30 Days Of JavaScript":

****

When you run this code, it will output the position of the first occurrence of the letter "a" in the string "30 Days Of JavaScript". If the letter "a" is not found in the string, it will return -1.

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

**Ans :** You can use the lastIndexOf() method to determine the position of the last occurrence of a specific substring within a string. Here's how you can use it to find the position of the last occurrence of the letter "a" in the string "30 Days Of JavaScript":

****

When you run this code, it will output the position of the last occurrence of the letter "a" in the string "30 Days Of JavaScript". If the letter "a" is not found in the string, it will return -1.

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

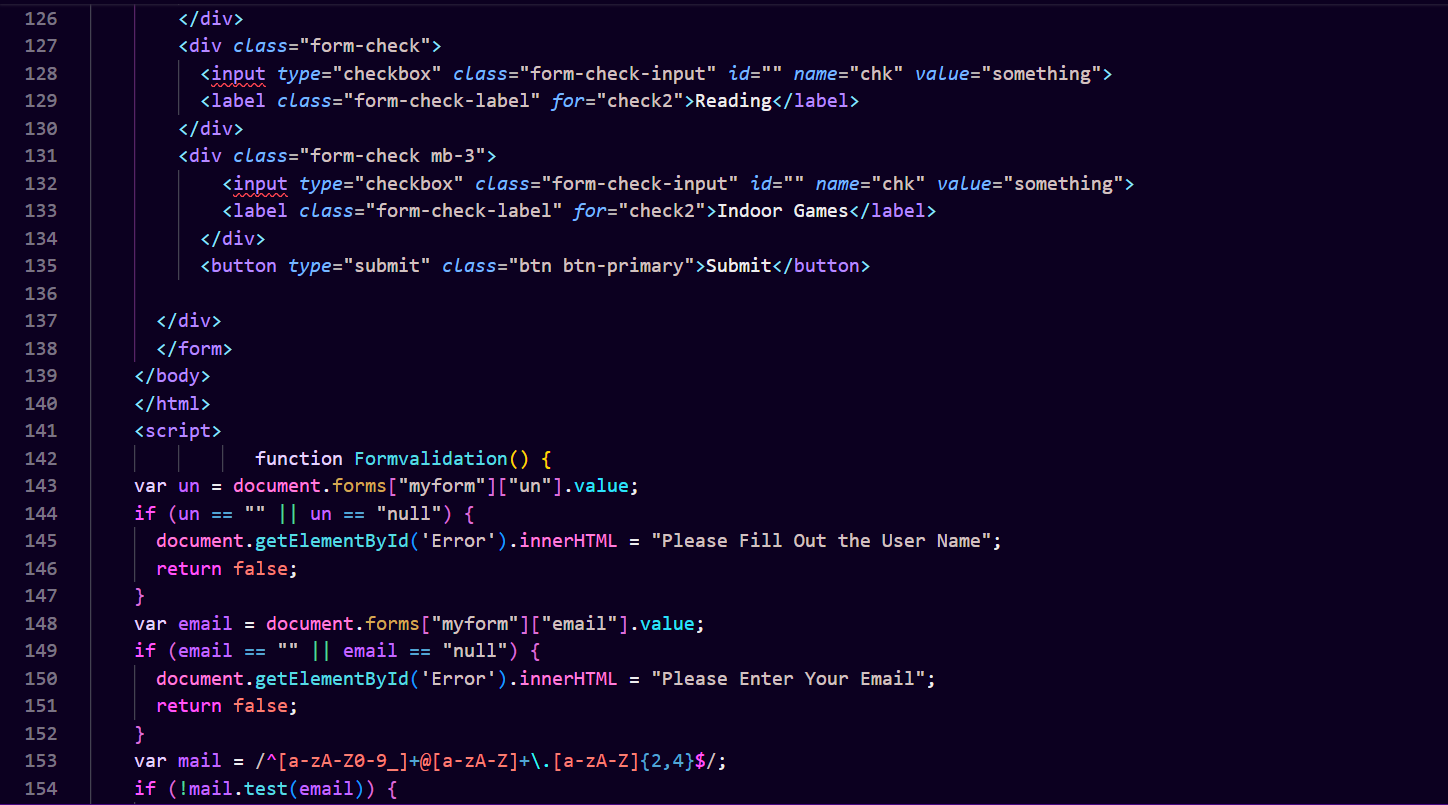
**Q.55 Form in Email, number, Password, Validation?**

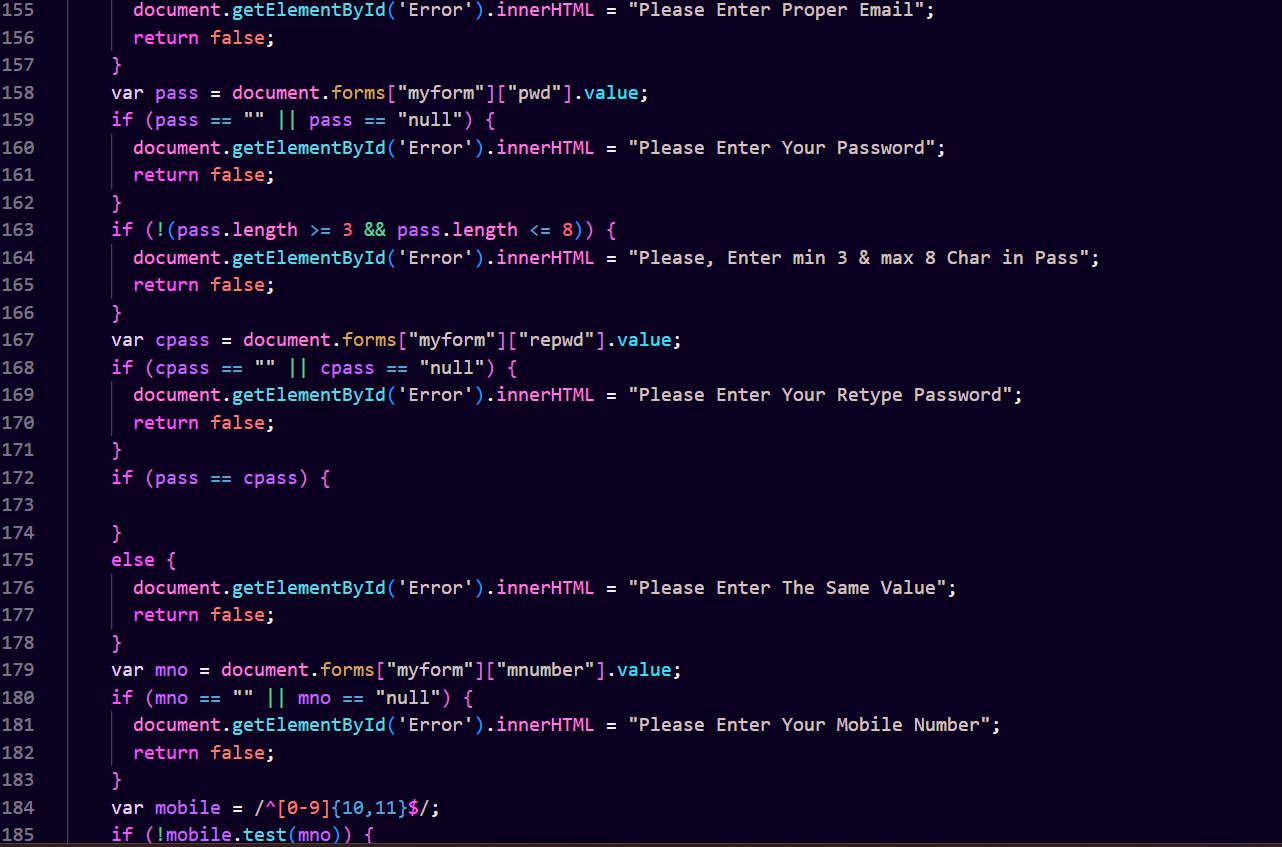
**Q.56 Dynamic Form Validation in JS?**

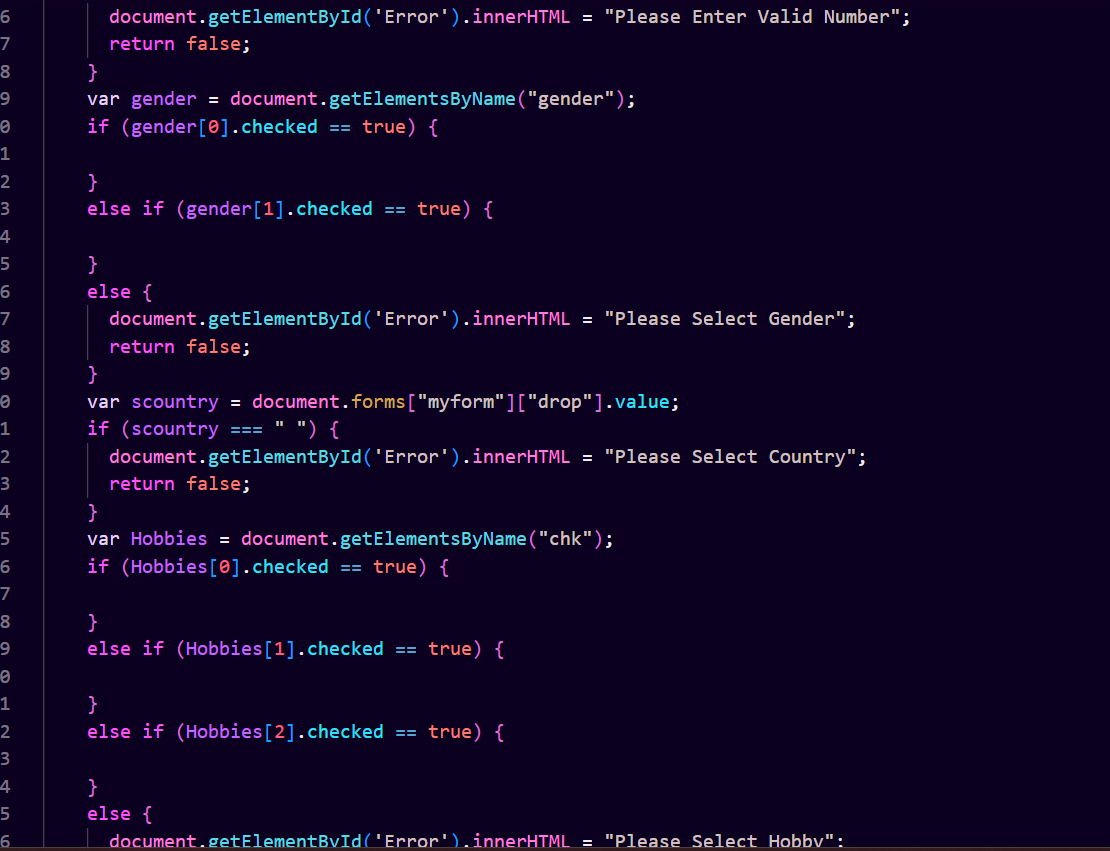
**Ans :**

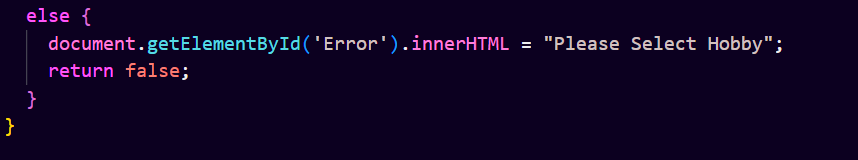
****

****

****

****

****

****

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

**Ans** : JavaScript events can be categorized into several types based on their sources or triggers. Here are some common types of JavaScript events:

1. Mouse Events:

click: Triggered when the user clicks an element.

dblclick: Triggered when the user double-clicks an element.

mouseover: Triggered when the mouse pointer enters an element.

mouseout: Triggered when the mouse pointer leaves an element.

mousedown: Triggered when the mouse button is pressed down over an element.

mouseup: Triggered when the mouse button is released over an element.

mousemove: Triggered when the mouse pointer moves while it is over an element.

1. Keyboard Events:

keydown: Triggered when a keyboard key is pressed down.

keyup: Triggered when a keyboard key is released.

keypress: Triggered when a keyboard key that produces a character value is pressed.

1. Form Events:

submit: Triggered when a form is submitted.

reset: Triggered when a form is reset.

focus: Triggered when an element gets focus.

blur: Triggered when an element loses focus.

change: Triggered when the value of an input element changes.

input: Triggered when the value of an input element is being changed by the user.

1. Document/Window Events:

load: Triggered when a resource and its dependent resources have finished loading.

resize: Triggered when the browser window is resized.

scroll: Triggered when an element's scrollbar is being scrolled.

unload: Triggered when a page is unloaded (closed or navigated away from).

1. Touch Events:

**touchstart: Triggered when a touch point is placed on the touch surface.**

**touchend: Triggered when a touch point is removed from the touch surface.**

**touchmove: Triggered when a touch point is dragged along the touch surface.**

**touchcancel: Triggered when a touch point has been disrupted in an unexpected way (e.g., an event of catastrophic proportions occurs).**

1. **Media Events:**

**play: Triggered when media playback begins.**

**pause: Triggered when media playback is paused.**

**ended: Triggered when media playback has reached the end.**

**volumechange: Triggered when the volume of the media changes.**

**durationchange: Triggered when the duration of the media changes.**

**Q.59 What is Bom vs Dom in JS?**

**Ans :** BOM and DOM are both integral parts of JavaScript when working within a web browser, but they serve different purposes and handle different aspects of web development:

1. DOM (Document Object Model):

The DOM represents the structure of an HTML document as a tree-like structure.

It provides a structured representation of the document, allowing JavaScript to interact with and manipulate the document's elements, attributes, and text content dynamically.

With the DOM, you can access and modify the content, structure, and style of HTML documents using JavaScript.

For example, you can use DOM methods like getElementById(), querySelector(), appendChild(), setAttribute(), etc., to interact with HTML elements and their attributes.

1. BOM (Browser Object Model):

The BOM represents everything else that the browser provides beyond the document itself.

It includes objects such as window, document, navigator, history, screen, location, etc.

The BOM provides JavaScript access to various aspects of the browser environment and the client system.

It allows JavaScript to control the browser window, navigate to different URLs, manage browser history, interact with user input devices, display alerts and prompts, handle cookies, and more.

Unlike the DOM, which is standardized by the W3C, the BOM is not standardized and can vary between different browsers.

**Q.60 Array vs object defences in JS?**

**Ans :** In JavaScript, arrays and objects are both data structures used to store and manipulate data, but they have different characteristics and use cases:

1. Arrays:

Arrays are ordered collections of values, where each value is identified by an index.

The index of an array starts from 0 and increments by 1 for each subsequent element.

Arrays are ideal for storing collections of similar items or sequential data.

They provide methods for adding, removing, and accessing elements, such as push(), pop(), shift(), unshift(), splice(), and slice().

Arrays are suitable for situations where you need to maintain the order of elements and perform operations like iteration, sorting, filtering, and mapping over the elements.

1. Objects:

Objects are collections of key-value pairs, where each key is unique and associated with a value.

Objects are not ordered, meaning there is no guarantee of the order in which key-value pairs are stored or retrieved.

Objects are ideal for storing data with named properties or attributes.

They provide methods for accessing and manipulating properties, such as dot notation (object.property) or bracket notation (object['property']), Object.keys(), Object.values(), and Object.entries().

Objects are suitable for situations where you need to access data based on descriptive keys or perform operations like lookup and retrieval based on keys.

In summary:

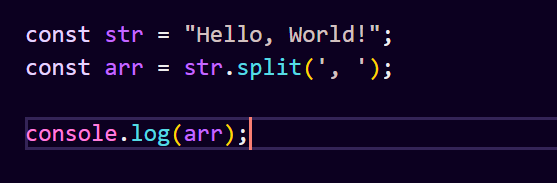
Use arrays when you need to store ordered collections of similar items or sequential data.

Use objects when you need to store data with named properties or attributes and need to access it using descriptive keys.

It's worth noting that JavaScript arrays are a specialized type of object. While they have additional array-specific methods and behaviors, they can still have arbitrary properties and behave like regular objects.

**Q.61 Split the string into an array using split() Method?**

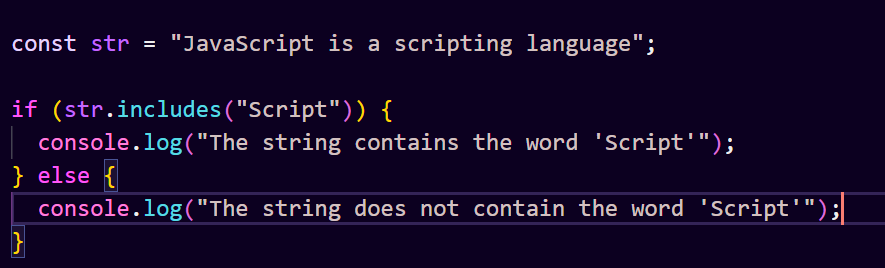
**Ans :**

****

in this example, the split() method is called on the string str, using ', ' as the separator. This splits the string wherever ', ' occurs, creating an array with "Hello" and "World!" as its elements.

**Q.62 Check if the string contains a word Script using includes() method?**

**Ans :**

****

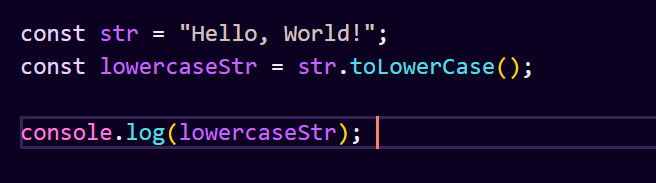
The includes() method is called on the string str, with the argument "Script".

If the string contains the substring "Script", includes() returns true, otherwise it returns false.

Based on the result, the appropriate message is logged to the console.

**Q.63 Change all the string characters to lowercase letters using toLowerCase() Method**

**Ans :**

****

The toLowerCase() method is called on the string str.

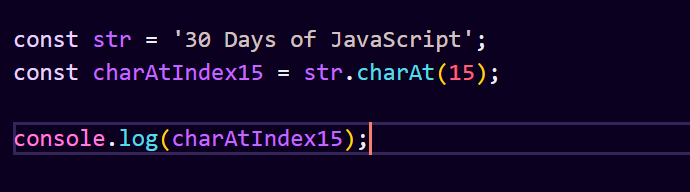
It returns a new string with all characters converted to lowercase.

The resulting lowercase string is stored in the variable lowercaseStr.

Finally, the lowercase string is printed to the console.

**Q.64 What is Character at index 15 in ’30 Days of JavaScript’ string? Use charAt() method**.

**Ans :**

****

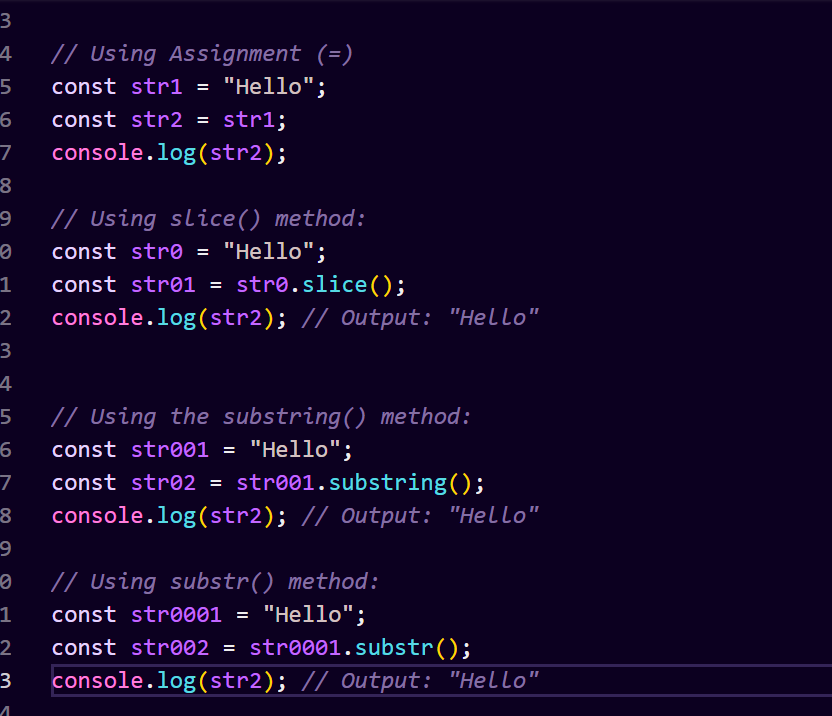
The charAt() method is called on the string str, with the argument 15.

This returns the character at index 15 in the string, which is 'v'.

The character 'v' is then printed to the console.

**Q.65 copy to one string to another string in JS?**

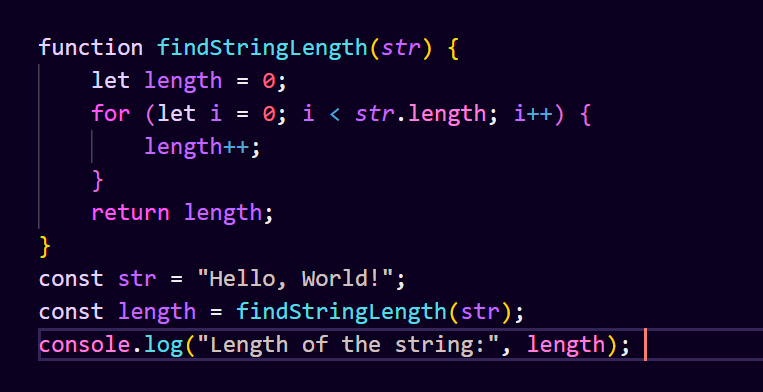
**Ans :** You can copy the content of one string to another string in JavaScript using various methods, such as assignment (=), slice(), substring(), or substr() methods. Here are a few examples:

****

All of these methods create a new string that contains a copy of the content of the original string. The resulting string (str2) is independent of the original string (str1), so changes to one string will not affect the other.

**Q.66 Find the length of a string without using libraryFunction?**

**Ans :** You can find the length of a string in JavaScript without using library functions by iterating over the characters of the string and counting them manually. Here's how you can do it:



The findStringLength function takes a string str as input.

It initializes a variable length to 0.

It iterates over each character of the string using a for loop and increments the length variable for each character.

After iterating through the entire string, it returns the final length of the string.

Example usage demonstrates finding the length of the string "Hello, World!" without using any library functions.

* **What is JavaScript?**

**Ans :** JavaScript is a high-level, interpreted programming language primarily used for creating dynamic and interactive content on webpages. It is one of the core technologies of the World Wide Web, alongside HTML and CSS. JavaScript allows developers to add behavior to web pages, control the browser, manipulate content dynamically, and interact with users.

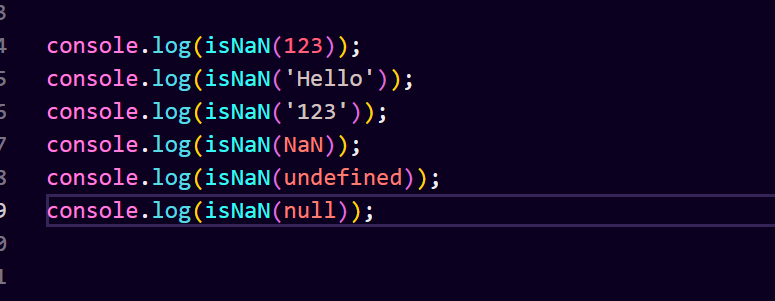
JavaScript (JS) is the most popular programming language for web development. It can be used for both Client-side and Server-side. It is also known as a scripting language for web pages. It is commonly used to create dynamic and interactive content on websites. JS is an essential component of modern web browsers, enabling client-side scripting to enhance user experience by manipulating the content of web pages in real time.

* **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. It returns true if the value is NaN, and false otherwise.

NaN is a special value in JavaScript that represents "Not a Number". It is returned when a mathematical operation or function that expects a number receives an operand that is not a valid number. For example, dividing zero by zero, dividing infinity by infinity, or attempting to parse a string that is not a valid number will result in NaN.

Here's how you can use the isNaN() function:



* **What is negative Infinity?**

**Ans :** If it is, there are some serious issues that we need to deal with as we’ll see in a bit. Subtraction with negative infinity can also be dealt with in an intuitive way in most cases as well. A really, really large negative number minus any positive number, regardless of its size, is still a really, really large negative number.

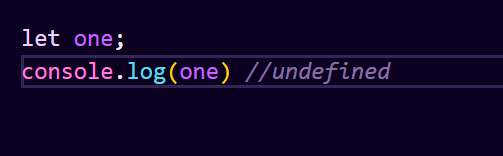
* **Which company developed JavaScript?**

**Ans :** JavaScript was invented by Brendan Eich in 1995. It was developed for Netscape 2, and became the ECMA-262 standard in 1997. After Netscape handed JavaScript over to ECMA, the Mozilla foundation continued to develop JavaScript for the Firefox browser. Mozilla's latest version was 1.8.5. (Identical to ES5).

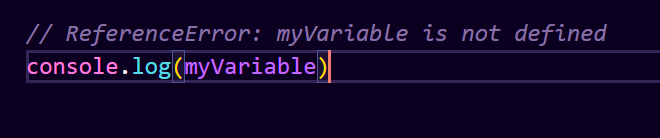
* **What are undeclared and undefined variables?**

**Ans :**

**Undefined:** It occurs when a variable has been declared but has not been assigned any value. Undefined is not a keyword.



**Undeclared:** It occurs when we try to access any variable that is not initialized or declared earlier using the var or const keyword. If we use ‘typeof’ operator to get the value of an undeclared variable, we will face the runtime error with the return value as “undefined”. The scope of the undeclared variables is always global.



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

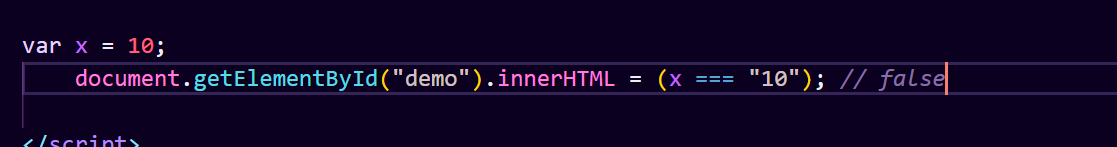
**Ans :**

****

This code will add a new <p> element with a text node containing the label "New Element" followed by a number representing the count of existing elements + 1 every time you click the "Add New Element" button. You can adapt this example to add different types of elements or modify the content of the dynamically added elements as needed.

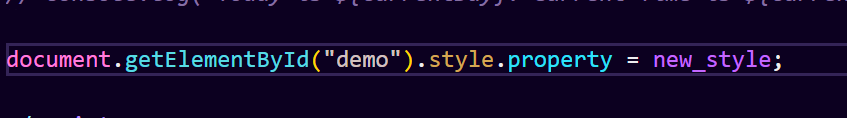
* **What is === operator?**

**Ans :** === (Triple equals) is a strict equality comparison operator in JavaScript, which returns false for the values which are not of a similar type. This operator performs type casting for equality. If we compare 2 with “2” using ===, then it will return a false value.

****

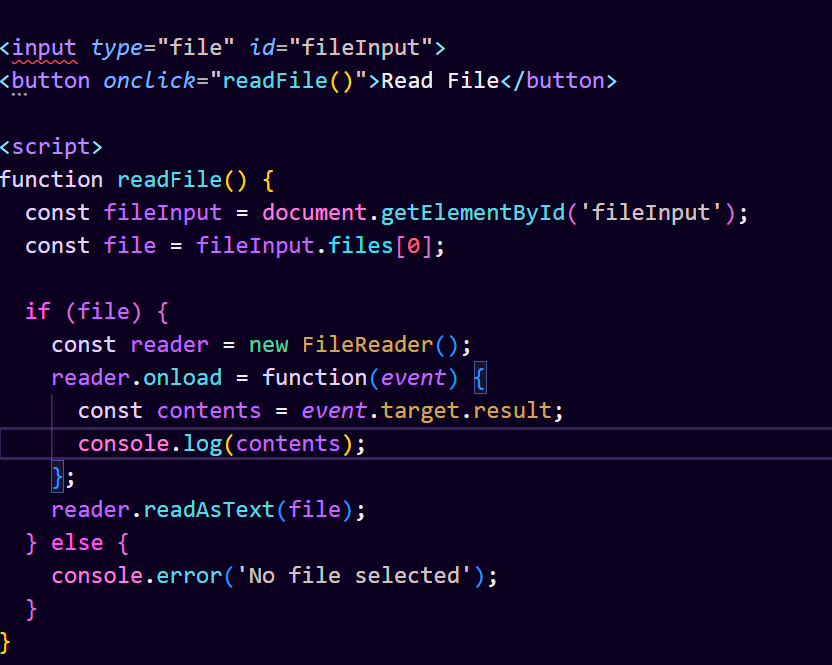
* **How can the style/class of an element be changed?**

**Ans :** In this article, we will learn how we can change the style/class of an element. If you want to build a cool website or app then UI plays an important role. We can change, add or remove any CSS property from an HTML element on the occurrence of any event with the help of JavaScript. There are two common approaches that allow us to achieve this task.

****

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

**Ans :**

****

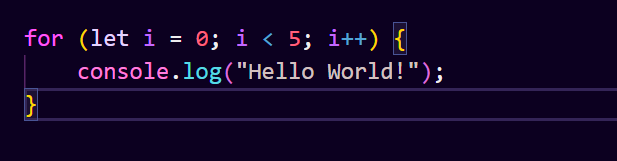
Writing a File (Not supported in browser for security reasons):

In a browser environment, writing files directly from JavaScript is generally not permitted due to security restrictions. You would typically use techniques like Blob URLs or FileSaver.js to prompt the user to download a file instead of directly writing to the filesystem.

**• What are all the looping structures in JavaScript?**

**Ans :** JavaScript Loops are powerful tools for performing repetitive tasks efficiently. Loops in JavaScript execute a block of code again and again while the condition is true.

For example, suppose we want to print “Hello World” 5 times. This can be done using JS Loop easily. In Loop, the statement needs to be written only once and the loop will be executed 5 times as shown below:



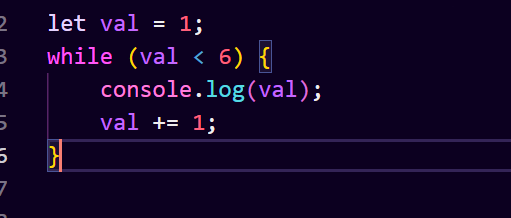
1. **JavaScript For Loop**

Looping in programming languages is a feature that facilitates the execution of a set of instructions repeatedly until some condition evaluates and becomes false. We come across for loop which provides a brief and systematic way of writing the loop structure.

JavaScript for loop is used to iterate the elements for a fixed number of times. JavaScript for loop is used if the number of the iteration is known.

1. **JavaScript While Loop**

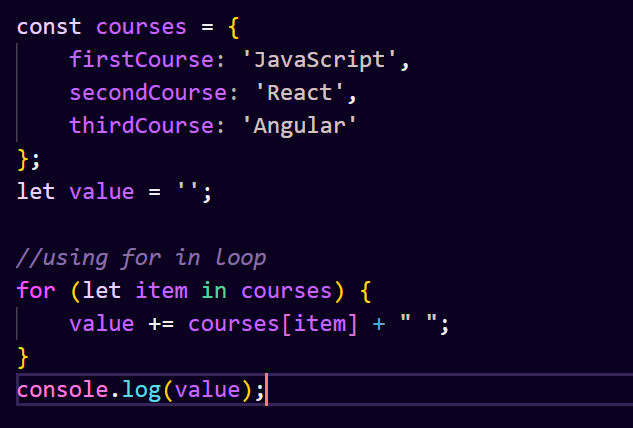
A While Loop in JavaScript is a control flow statement that allows the code to be executed repeatedly based on the given boolean condition. The while loop can be thought of as a repeating if statement. The loop can be used to execute the specific block of code multiple times until it fails to match the condition.



1. **JavaScript for in Loop**

JavaScript for in loop is used to iterate over the properties of an object. JavaScript for in loop iterates only over those keys of an object which have their enumerable property set to “true”.

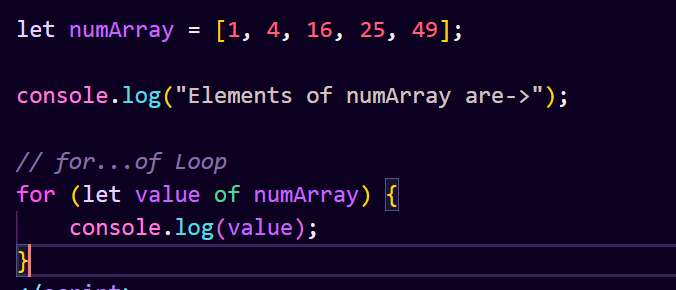
The key values in an object have four attributes (value, writable, enumerable, and configurable). It can be a great debugging tool if we want to show the contents of an object.



1. **JavaScript for…of Loop**

JavaScript for…of statement iterates over the values of an iterable object (like Array, Map, Set, arguments object, …,etc), executing statements for each value of the object.

JavaScript for…of loop makes it easy to loop through the elements without needing to handle the index or iteration logic which makes the code short and easier to understand.



1. **JavaScript do…while Loop**

A do… while loop in JavaScript is a control statement in which the code is allowed to execute continuously based on a given boolean condition. It is like a repeating if statement.

The do…while loop can be used to execute a specific block of code at least once

**• How can you convert the string of any base to an integer in JavaScript?**

**Ans :**

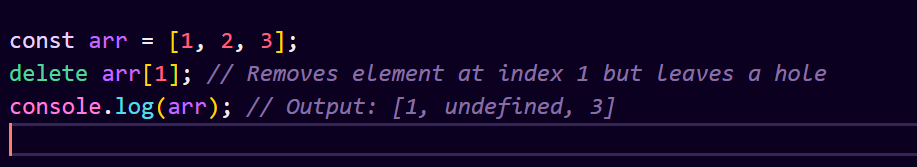
* **What is the function of the delete operator?**

**Ans :**

Deleting Variables:

The delete operator cannot delete variables declared with var, let, or const. It's used specifically for removing properties from objects or elements from arrays.

In summary, the delete operator allows you to remove properties from objects or elements from arrays, but it's important to understand its behavior and limitations, especially regarding array manipulation and non-configurable properties.

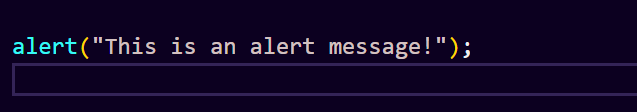


* **What are all the types of Pop up boxes available in JavaScript?**

**Ans :** In JavaScript, there are three main types of popup boxes that you can use for interacting with users:

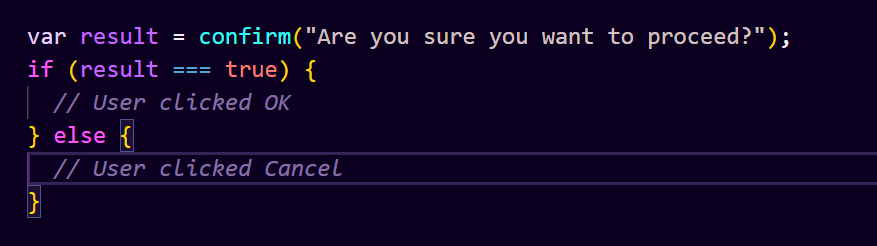
1. Alert Box:

An alert box is used to display a message to the user. It only has one button, which is usually labeled "OK".



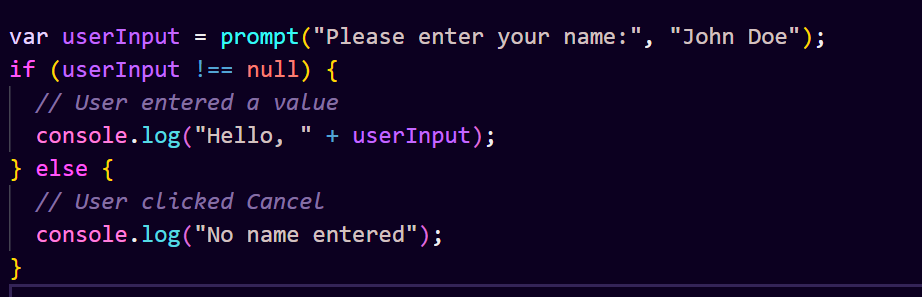
1. Confirm Box:

A confirm box is used to get confirmation from the user. It typically has two buttons, "OK" and "Cancel". It returns true if the user clicks "OK" and false if the user clicks "Cancel".



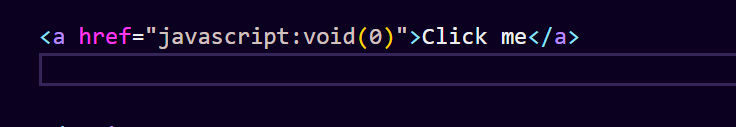
1. Prompt Box:

A prompt box is used to get input from the user. It provides a text field for the user to enter data. It returns the text entered by the user or null if the user clicks "Cancel".



**• What is the use of Void (0)?**

**Ans :** In JavaScript, void(0) is a unary operator used to evaluate an expression and return undefined. It's typically used in the context of anchor (<a>) elements in HTML to prevent the browser from navigating to a new page when the anchor is clicked without needing to specify a href attribute.



In this example, clicking the link will execute the JavaScript expression void(0). Since void(0) always returns undefined, clicking the link won't cause the browser to navigate anywhere.

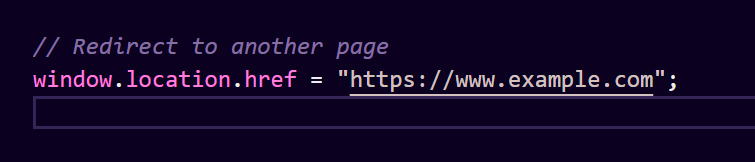
Using javascript:void(0) as the href value is a way to create a non-navigating link, often used for JavaScript-driven functionality or to prevent the default behavior of an anchor link without having to define a specific action.

However, it's worth noting that the use of void(0) for this purpose has become less common in modern JavaScript and HTML development. Instead, many developers prefer using event listeners or other mechanisms to handle user interactions and prevent default browser behavior.

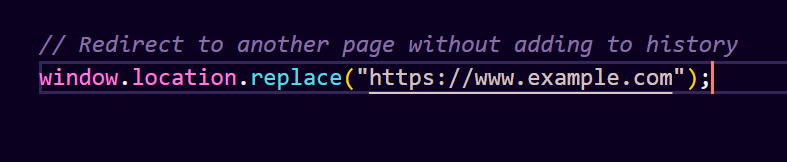
* **How can a page be forced to load another page in JavaScript?**

**Ans :**

In JavaScript, you can force a page to load another page using the window.location object. You can set its href property to the URL of the page you want to navigate to. Here's how you can do it:

This line of code will immediately navigate the browser to the specified URL, loading the content of that page and replacing the current page in the browser's history.

You can also use other properties of the window.location object to manipulate the browser's location. For example, you can use window.location.replace() to navigate to a new page without adding a new entry to the browser's history:



Or you can use window.location.assign() for similar behavior to window.location.href:



All of these methods can be used to force a page to load another page in JavaScript. However, keep in mind that this can be an abrupt user experience, so it should be used judiciously and considerately.

**• What are the disadvantages of using innerHTML in JavaScript?**

**Ans :**

Disadvantages of using innerHTML property in JavaScript:

The use of innerHTML very slow: The process of using innerHTML is much slower as its contents as slowly built, also already parsed contents and elements are also re-parsed which takes time.

Preserves event handlers attached to any DOM elements: The event handlers do not get attached to the new elements created by setting innerHTML automatically. To do so one has to keep track of the event handlers and attach it to new elements manually. This may cause a memory leak on some browsers.

Content is replaced everywhere: Either you add, append, delete or modify contents on a webpage using innerHTML, all contents is replaced, also all the DOM nodes inside that element are reparsed and recreated.

Appending to innerHTML is not supported: Usually, += is used for appending in JavaScript. But on appending to an Html tag using innerHTML, the whole tag is re-parsed.

Old content replaced issue: The old content is replaced even if object.innerHTML = object.innerHTML + ‘html’ is used instead of object.innerHTML += ‘html’. There is no way of appending without reparsing the whole innerHTML. Therefore, working with innerHTML becomes very slow. String concatenation just does not scale when dynamic DOM elements need to be created as the plus’ and quote openings and closings becomes difficult to track.

Can break the document: There is no proper validation provided by innerHTML, so any valid HTML code can be used. This may break the document of JavaScript. Even broken HTML can be used, which may lead to unexpected problems.

Can also be used for Cross-site Scripting(XSS): The fact that innerHTML can add text and elements to the webpage, can easily be used by malicious users to manipulate and display undesirable or harmful elements within other HTML element tags. Cross-site Scripting may also lead to loss, leak and change of sensitive information.