

# JavaScript Coding Questions and Answers

Last Updated : 05 Aug, 2025

JavaScript is the most commonly used interpreted, and scripted Programming language. It is used to make web pages, mobile applications, web servers, and other platforms. Developed in 1995 by Brendan Eich. Developers should have a solid command over this because many job roles need proficiency in [JavaScript](#).

We will see the **Top 50 JavaScript Coding questions and answers** including basic and medium JavaScript coding questions and answers. In this article, we will cover everything like JavaScript core concepts- arrays, strings, arrow functions, and classes. These Top 50 coding questions and answers will help you to improve your coding concept in JavaScript.

## Basic JavaScript Coding Questions and Answers

If you're a beginner and want to gather some real-time coding examples, then start here. This section will help you understand the fundamentals and solve simple coding problems.

### 1. Write a Program to reverse a string in JavaScript.

This code splits the string into an array of characters using the `split()` method, reverses the array, and joins it back into a string using the [join\(\)](#) method.

```
function reverseString(str)
{
    return str.split("").reverse().join("");
}

console.log(reverseString("GeeksForGeeks"));
```



## Output

skeeGroFskeeG

## 2. Write a Program to check whether a string is a palindrome string.

- A palindrome is a word that reads the same word from forward and backward. This ignores spaces and capitalization.
- The code below checks if a string is a palindrome by reversing it and comparing it to the original string. If both are equal, it returns true otherwise, it is false. "GFG" is a palindrome string so it returns true.

```
function isPalindrome(str) {  
    const reversed = str.split("").reverse().join("");  
    return str === reversed;  
}
```

```
console.log(isPalindrome("GFG"));
```

× ▶ 📄

## Output

true

## 3. Find the largest number in an array in JavaScript.

Using a For loop:

The code defines a function findLargest that finds the largest number in an array. It starts by assuming the first element is the largest, then iterates through the array. If a larger number is found, it updates the largest value. Finally, it returns the largest number which is 100.

```
function findLargest(arr) {  
    //Suppose first element is the largest  
    let largest = arr[0];  
    for (let i = 1; i < arr.length; i++) {  
        if (arr[i] > largest) {
```

```
        // Update the largest if a bigger element is
found
        largest = arr[i];
    }
}
return largest;
}

console.log(findLargest([99, 5, 3, 100, 1]));
```

## Output

100

Using the spread operator (...) or Math.max:

The findLargest function uses Math.max() to find the largest number in an array. The spread operator (...arr) expands the array elements so Math.max() can evaluate each value. In the given code, it returns 100, the largest number from [99, 5, 3, 100, 1].

```
function findLargest(arr) {
    // Math.max() is used to find the largest number
    return Math.max(...arr);
}

console.log(findLargest([99, 5, 3, 100, 1]));
```

## Output

100

## 4. How Remove the first element from an array in JavaScript?

The code initializes an array arr with values [5, 6, 7]. The slice(1) method creates a new array, excluding the first element (5), resulting in [6, 7]. Finally, console.log(arr) outputs the modified array [6, 7] to the console.

```
// Initialize an array
let arr = [5, 6, 7];
arr = arr.slice(1);
// Create a new array without the first element
```

```
console.log(arr);
```



## Output

```
[ 6, 7 ]
```

## 5. Write a Program to use a callback function?

This code defines a greet function that takes two arguments- a name and a callback function. It calls the callback with a greeting message using the name. When greet('Geek', message => console.log(message)) is executed, it prints "Hello, Geek!" as output.

```
function greet(name, callback) {  
    callback(`Hello, ${name}!`);  
}  
greet('Geek', message => console.log(message));
```



## Output

```
Hello, Geek!
```

## 6. Write a code to create an arrow function?

The code defines an arrow function add that takes two arguments a and b and returns their sum (a + b). When console.log(add(6, 2)) is executed, it calls the add function with 6 and 2, and prints the result which is 8, to the console.

```
const add = (a, b) => a + b;  
console.log(add(6, 2));
```



## Output

```
8
```

## 7. Write a Program to add a property to an object?

The code creates an object `obj` with a `name` property set to `'Riya'`. Then, it adds a new property `age` with the value `21`. Finally, `console.log(obj)` prints the updated object, which now contains both `name` and `age` properties: `{ name: 'Riya', age: 21 }`.

```
const obj = { name: 'Riya' };  
obj.age = 21;  
console.log(obj);
```



## Output

```
{ name: 'Riya', age: 21 }
```

## 8. Write a Program to delete a property from an object?

In this code, an object `obj` with properties `name` and `age` is created. The `delete` keyword removes the `age` property from the object. After deletion, `console.log(obj)` outputs the object, which now only contains the `name` property: `{ name: 'Riya' }`.

```
const obj = { name: 'Riya', age: 21 };  
delete obj.age;  
console.log(obj);
```



## Output

```
{ name: 'Riya' }
```

## 9. What will be the output of the given code?

The code uses the `reduce` method on the array `[1, 2, 3]` to sum its elements. It takes two parameters, `a` (accumulator) and `b` (current value), adding them together. The final result `6` is printed to the console, representing the total of the array's numbers.

```
console.log([1, 2, 3].reduce((a, b) => a + b)); //adds  
numbers in the array, together using the reduce me. X d ▶
```



## Output

6

## 10. What will be the output of the given code?

The code `console.log('gfg'.repeat(3));` uses the `repeat()` method to create a new string by repeating the string 'gfg' three times. The output will be 'gfggfgg' which is displayed in the console. This method is useful for printing repeated text easily.

```
console.log('gfg'.repeat(3));  
//creates a new string by repeating the original string in 3  
times
```

### Output

gfggfggfg

## 11. What will be the output of the given code?

```
console.log(1 + '2');
```

The code `console.log(1 + '2');` adds the number 1 and the string '2'. In JavaScript, when a number and a string are combined with `+`, the number is converted to a string. Finally, it results in the string '12', which is then prints to the console.

## 12. What will be the output of the given code?

```
console.log('6' - 1);
```

'6' is a string, so when you use the `-` operator with string and number, JavaScript convert the string to a number automatically which is called type coercion. '6' gets converted to the number 6, then  $6 - 1 = 5$ . So, 5 is the answer.

### Output:

5

### 13. What will be the output of the given code?

```
console.log(1 === '1');
```

'1' is a integer and '1' is a string. Strict equality operator(===) check both the type and the value. They look similar but they have different data type. So =, the answer in false.

#### Output:

false

### 14. What will be the output of the given code?

'null' represents the absence of any value and 'undefined' represents a variable that has been declared but not assigned any value. The answer is true because JavaScript treats them equal because of '==' loose equality operator.

```
console.log(null == undefined);
```

× ▶ 📄

#### Output

true

### 15. Write a Program to find a sum of an array?

The sumArray function takes an array arr as input and initializes a variable sum to 0. It loops through each element of the array, adding each element's value to sum. Finally, it returns the total sum which is 33.

```
function sumArray(arr) {  
    let sum = 0;  
    for (let i = 0; i < arr.length; i++) {  
        sum += arr[i];  
    }  
}
```

```
        return sum;
    }

    console.log(sumArray([15, 6, 10, 2]));
```

  

## Output

33

## 16. Write a Program to check if a number is prime or not?

The `isPrime()` function checks if a number `num` is prime. It returns `false` for numbers less than or equal to 1. It loops starts from 2 to `num - 1`, checking if `num` is divisible by any number in that range. If it is, it returns `false` otherwise, it returns `true`.

```
function isPrime(num) {
    if (num <= 1)
        return false;
    for (let i = 2; i < num; i++)
    {
        if (num % i === 0)
            return false;
    }
    return true;
}

console.log(isPrime(7));
```

  

## Output

true

## 17. Write a Program to print Fibonacci sequence up to n terms?

The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones, starting from 0 and 1.

The fibonacci function generates the Fibonacci sequence up to `n` terms. It initializes two variables `num1` (0) and `num2` (1). In a loop, it prints `num1`,



then calculates the next number as the sum of num1 and num2, updating them for the next iteration. Here, it prints 7 terms.

```
function fibonacci(n) {  
    let num1 = 0, num2 = 1, nextNum;  
  
    console.log("Fibonacci Sequence:");  
  
    for (let i = 1; i <= n; i++) {  
        console.log(num1);  
        nextNum = num1 + num2;  
        num1 = num2;  
        num2 = nextNum;  
    }  
}  
  
fibonacci(7);
```



## Output

Fibonacci Sequence:

0  
1  
1  
2  
3  
5  
8

## 18. Write a Program to find factorial of a number?

The factorial function calculates the factorial of a given number num. It initializes answer to 1, then multiplies it by each integer from 2 to num in a loop. Finally, it returns the computed factorial. The console.log statement prints the factorial of 7, which is 5040.

```
function factorial(num) {  
    let answer = 1;  
    for (let i = 2; i <= num; i++) {  
        answer *= i;  
    }  
}
```

```
        return answer;
    }

    console.log(factorial(7));
```

  

## Output

5040

## 19. Calculate the Power of a Number in JavaScript?

The power function takes two arguments- base and exponent. It calculates the result of raising base to the power of exponent using the exponentiation operator \*\*.

```
function power(base, exponent)
{
    return base ** exponent;
}

console.log(power(3, 4));
```

  

## Output

81

## 20. Write a Program to print the frequency of elements in an array?

The frequency function counts how many times each number appears in an array. It creates an empty object freq, iterates through the array, and either increments the count for existing numbers or adds new numbers with a count of 1. Finally, it returns the freq object with the counts.

```
function frequency(arr) {
    const freq = {};
    for (let i = 0; i < arr.length; i++) {
        if (freq[arr[i]]) {
            freq[arr[i]] += 1;
        } else {
```

```
        freq[arr[i]] = 1;
    }
}
return freq;
}

console.log(frequency([1, 1, 2, 3, 3, 4]));
```

## Output

```
{ '1': 2, '2': 1, '3': 2, '4': 1 }
```

## Medium JavaScript Coding Questions and Answers

Now that you've learned the basics, you can move to more advanced topics. Here, we will dive into complex coding problems and explore deeper concepts to strengthen your skills.

### 21. Write a Program to count the occurrences of a character in a string in JavaScript?

**using split() method**

The countChar() function counts how many times a specified character (char) appears in a string (str). It splits the string into an array using the character, then returns the length of the array (length-1), which gives the count of the character. The given code counts 'G' in 'GeeksForGeeks'.

```
function countChar(str, char)
{
    return str.split(char).length - 1;
}

console.log(countChar('GeeksForGeeks', 'G'));
```

## Output

```
2
```

**Using a for loop**

The countChar() function counts how many times a given character (char) appears in a given string (str). It initializes a counter (count) to

zero, iterates through each character in the string, increments the counter when it finds a match and returns the total count.

```
function countChar(str, char) {  
    let count = 0;  
    for (let i = 0; i < str.length; i++) {  
        if (str[i] === char) {  
            count++;  
        }  
    }  
    return count;  
}  
  
console.log(countChar('GeeksForGeeks', 'G'));
```

## Output

2

## 22. Write a Program to convert Celsius to Fahrenheit in JavaScript?

We are using the formula **Fahrenheit=(Celsius×9/5)+32** to convert Celsius to Fahrenheit.

```
function celsiusToFahrenheit(celsius) {  
    // find the conversion from Celsius to  
    // Fahrenheit Fahrenheit=(Celsius×9/5)+32  
    return (celsius * 9 / 5) + 32;  
}  
  
console.log(celsiusToFahrenheit(20));
```

## Output

68

## 23. Write a Program to convert Fahrenheit to Celsius in JavaScript?

We are using the formula **Celsius=(Fahrenheit-32)\*5/9** to convert Fahrenheit to Celsius.

```
function fahrenheitToCelsius(fahrenheit) {  
    return (fahrenheit - 32) * 5 / 9;  
}
```

```
console.log(fahrenheitToCelsius(68));
```

× ▶ 📄

## Output

20

## 24. Write a Program to sort an array in Ascending Order in JavaScript?

The sortArray function sorts an array in ascending order using a nested loop. It compares each element with the others and swaps them if they are out of order. After iterating through the array, it returns the sorted array. For example, [5, 3, 8, 1] becomes [1, 3, 5, 8].

```
function sortArray(arr) {  
    for (let i = 0; i < arr.length; i++) {  
        for (let j = i + 1; j < arr.length; j++) {  
            if (arr[i] > arr[j]) {  
                // swap the elements  
                let temp = arr[i];  
                arr[i] = arr[j];  
                arr[j] = temp;  
            }  
        }  
    }  
    return arr;  
}
```

```
console.log(sortArray([5, 3, 8, 1]));
```

× ▶ 📄

## Output

[ 1, 3, 5, 8 ]

## 25. write a Program to sort an array in Descending Order in JavaScript?

The code sorts an array in descending order using a bubble sort algorithm. It repeatedly swaps adjacent elements if they are in the wrong order then returning the sorted array.

```
function sortArrayDesc(arr) {  
    let n = arr.length;  
    for (let i = 0; i < n - 1; i++) {  
        for (let j = 0; j < n - 1 - i; j++) {  
            if (arr[j] < arr[j + 1]) {  
                // Swap the elements  
                let temp = arr[j];  
                arr[j] = arr[j + 1];  
                arr[j + 1] = temp;  
            }  
        }  
    }  
    return arr;  
}
```

```
console.log(sortArrayDesc([5, 3, 8, 1]));
```

× ▶ 📄

### Output

```
[ 8, 5, 3, 1 ]
```

## 26. Write a Program to merge two arrays in JavaScript?

The mergeArrays function combines two arrays, arr1 and arr2, by using the concat method, which adds all elements of arr2 to the end of arr1. The function returns the merged array. In the given code [5, 6] and [7, 8] combine to form [5, 6, 7, 8].

```
function mergeArrays(arr1, arr2) {  
    // this method merges all the elements  
    // of arr2 at the end of arr1.  
    return arr1.concat(arr2);  
}
```

```
console.log(mergeArrays([5, 6], [7, 8]));
```

  

## Output

```
[ 5, 6, 7, 8 ]
```

## 27. Find the Intersection of Two Arrays in JavaScript?

In the given code set is used to store a unique values from arr2 then filter checks each element in arr1 to see if it is also exist in the set and keeping only those that matches. The output is an array of common elements from both arr1 and arr2.

```
function arrayIntersection(arr1, arr2)
{
  const set2 = new Set(arr2);
  return arr1.filter(value => set2.has(value));
}
```

```
console.log(arrayIntersection([5, 6, 7], [6, 7, 8 ]
```

  

## Output

```
[ 6, 7 ]
```

## 28. Find the Union of Two Arrays in JavaScript?

The arrayUnion function merges two arrays (arr1 and arr2) using the spread operator, combines them into a single array, and removes duplicates using Set. It then returns the unique values as a new array. In the given code it outputs [1, 2, 3, 4].

```
function arrayUnion(arr1, arr2) {
  // merges two arrays then removes duplicates
  // and returns the output as a new array.
  return [...new Set([...arr1, ...arr2])];
}
```

```
console.log(arrayUnion([1, 2, 3], [2, 3, 4]));
```

  

## Output

```
[ 1, 2, 3, 4 ]
```

## 29. Check if a Number is Even or Odd in JavaScript?

The function `isEven(num)` checks if a number is even by dividing it by 2. If the remainder (`num % 2`) is 0, the function returns `true`, means the number is even. Otherwise, it returns `false`. `console.log(isEven(10))` prints `true` because 10 is even number.

```
function isEven(num) {  
    return num % 2 === 0;  
}  
console.log(isEven(10));
```

× ▶ 📄

## Output

```
true
```

## 30. Write a Program to find the minimum value in an array in JavaScript?

### Using for loop

The function `findMin()` takes an array and finds the smallest value. It starts by assuming the first element is the minimum value, then loops through the array, comparing each element. If a smaller value is found, it updates `min`. value Finally, it returns the smallest value which is -1.

```
function findMin(arr) {  
    // Assume the first element is the minimum  
    let min = arr[0];  
    for (let i = 1; i < arr.length; i++) {  
        if (arr[i] < min) {  
            // Update min if a smaller value is found  
            min = arr[i];  
        }  
    }  
    return min;  
}
```



```
}  
  
console.log(findMin([5, 10, -1, 8]));
```



## Output

-1

### Using Math.min() method

This code defines a function findMin() that takes an array arr as input. It uses Math.min(...arr) to find and return the minimum value from the array by spreading its elements. In this code, it finds -1 as the smallest number in the array [5, 10, -1, 8].

```
function findMin(arr)  
{  
  return Math.min(...arr); // find minimum value  
}  
  
console.log(findMin([5, 10, -1, 8]));
```



## Output

-1

## 31. Check if a String Contains Another String in JavaScript?

The containsSubstring function checks if a substring exists within a given string. It uses indexOf to search for the substring inside the string. If found, it returns true otherwise, it returns false. In the code, it finds "For" in "GeeksForGeeks."

```
function containsSubstring(str, substring) {  
  //searches for the substring within str  
  return str.indexOf(substring) !== -1;  
}  
  
console.log(containsSubstring('GeeksForGeeks', 'Fo × ) > 
```



## Output

true

## 32. Find the First Non-Repeated Character in a String in JavaScript?

This code finds the first non-repeated character in a string. It first counts how many times each character appears, then checks the string again to return the first character that appears only once. If no non-repeated character is found, it returns null, but in the given string 'F' is a non-repeated character so the output is 'F'.

```
function fun(str) {  
    const charCount = {};  
  
    // count the occurrences of each character  
    for (let char of str) {  
        charCount[char] = (charCount[char] || 0) + 1;  
    }  
  
    // find the first non-repeated character  
    for (let char of str) {  
        if (charCount[char] === 1) {  
            return char;  
        }  
    }  
  
    return null;  
}  
  
console.log(fun('GeeksForGeeks'));
```

### Output

F

## 33. Find the Longest Word in a String in JavaScript?

This code finds the longest word in a given string. It splits the string into an array of words, then iterates through the array. For each word, it

checks if its length is greater than the current longest word, if yes then it updating the longest word accordingly. Finally, it returns the longest word.

```
function longestWord(str) {  
    //str is split into an array of words using the split  
    method.  
    const words = str.split(' ');  
    let longest = '';  
  
    for (let word of words) {  
        if (word.length > longest.length) {  
            longest = word;  
        }  
    }  
    return longest;  
}  
  
console.log(longestWord('GeeksForGeeks is great'))
```

## Output

GeeksForGeeks

## 34. Capitalize the First Letter of Each Word in a Sentence in JavaScript?

The function capitalizeFirstLetter() takes a sentence as input, splits it into an array of words, and capitalizes the first letter of each word. It uses a loop to modify each word and then joins the words back into a sentence, then returning the result.

```
function capitalizeFirstLetter(sentence) {  
    const words = sentence.split(' ');  
    for (let i = 0; i < words.length; i++) {  
        words[i] = words[i].charAt(0).toUpperCase() +  
words[i].slice(1);  
    }  
    return words.join(' ');  
}
```

```
console.log(capitalizeFirstLetter('hello geeks')); × ▶ 📄
```

## Output

Hello Geeks

## 35. Convert an Array of Strings to Uppercase in JavaScript?

The `toUpperCaseArray()` function takes an array of strings as input. It creates a new array and converts each string to uppercase using a loop. The uppercase strings are stored in the new array, which is returned. The console logs prints the result for the input `['g', 'f', 'g']`.

```
function toUpperCaseArray(arr)
{
  const upperCaseArray = [];
  for (let i = 0; i < arr.length; i++)
  {
    upperCaseArray[i] = arr[i].toUpperCase();
  }
  return upperCaseArray;
}
```

```
console.log(toUpperCaseArray(['g', 'f', 'g'])); × ▶ 📄
```

Full Stack Course   HTML   CSS   JavaScript   TypeScript   jQuery   AngularJS

Sign In

## Output

[ 'G', 'F', 'G' ]

## Hard JavaScript Coding Questions and Qnswers

After mastering the basics and advanced topics, you're ready for the hard coding challenges. This section will test your problem-solving abilities with real-world scenarios, incorporating some logic of Data Structures and Algorithms (DSA) that will help you become a proficient developer.

## 36. Write a Program to reverse an array in JavaScript?

The `reverseArray` function takes an array `arr` as input and creates a new empty array called `reversed[]`. It then loops through `arr` backwards, pushing each element into `reversed`. Finally it returns the reversed array as output.

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

console.log(reverseArray([5, 6, 7, 8]));
```

### Output

```
[ 8, 7, 6, 5 ]
```

## 37. Get the last element of an array in JavaScript?

The function `lastElement()` takes an array `arr` as input and returns its last element. It does this by accessing the element at the index "`arr.length - 1`", which shows the last position in the array. The console logs prints the last element of the array `[6, 2, 9, 5]` which is 5.

```
function lastElement(arr)
{
  return arr[arr.length - 1];
}

console.log(lastElement([6, 2, 9, 5]));
```

### Output

```
5
```

## 38. Remove falsy Values from an array in JavaScript?

The falsy values in JavaScript are values which are false, 0, "" (empty string), null, undefined, and NaN.

The removeFalsyValues() function takes an array and filters out falsy values (0, false, ""). It creates an empty array answer[], then iterates through the input array. If an element is truthy, it adds it to answer[]. Then, it returns the array of truthy values

```
function removeFalsyValues(arr) {  
    const answer = [];  
    for (let i = 0; i < arr.length; i++) {  
        if (arr[i]) {  
            answer[answer.length] = arr[i];  
        }  
    }  
    return answer;  
}  
  
console.log(removeFalsyValues([0, 5, false, 6, '', 7]));
```

## Output

```
[ 5, 6, 7 ]
```

## 39. Calculate the factorial of a number using recursion in JavaScript?

The factorial function calculates the factorial of a given number n. If n is 0 or 1, it returns 1. Otherwise, it multiplies n by the factorial of n - 1, effectively reducing the problem recursively until it reaches the base case. console.log(factorial(4)) outputs 24.

```
function factorial(n)  
{  
    if (n === 0 || n === 1) return 1;  
    return n * factorial(n - 1);  
}  
  
console.log(factorial(4));
```

## Output

24

### 40. Create an object and print the property?

This code creates an object named "person" with two properties- name ("GFG") and age ( 25). The console.log(person.name) line prints the value of the name property, which is "GFG", as output.

```
let person = { name: "GFG", age: 25 };  
console.log(person.name);
```

× ▶ 📄

## Output

GFG

### 41. Use the map function on an array in JavaScript?

The given creates an array called "numbers" containing the values 5, 6 and 7. It then uses the map function to create a new array "ans", where each number is multiplied by 2. Then, it prints the new array, which outputs [10, 12, 14] to the console.

```
let numbers = [5, 6, 7];  
let ans = numbers.map(function (num) {  
    return num * 2;  
});  
console.log(ans);
```

× ▶ 📄

## Output

[ 10, 12, 14 ]

### 42. Write a Program to create a simple class in JavaScript?

This code defines a class called "Animals" with a constructor that initializes the animal's name. The speak method prints a message

including the animal's name and a generic noise. A new instance of `Animals`, named `dog`, is created, and its `speak` method is called, so the output is "Dog makes a noise."

```
class Animals {  
  constructor(name) {  
    this.name = name;  
  }  
  speak() {  
    console.log(`${this.name} makes a noise`);  
  }  
}  
let dog = new Animals("Dog");  
dog.speak();
```

## Output

Dog makes a noise

## 43. Use JSON to parse and stringify data in JavaScript?

The code defines a JSON string `jsonData` containing a name. It uses `JSON.parse()` to convert the JSON string into a JavaScript object, storing it in `parsedData`. Finally, it logs the value of the `name` property from the object, which outputs "Geeks" to the console.

```
let jsonData = '{"name": "Geeks"}';  
let parsedData = JSON.parse(jsonData);  
console.log(parsedData.name);
```

## Output

Geeks

## 44. Convert a string to an array of words in JavaScript?

This code defines a string sentence containing "Geeks For Geeks". The `split()` method divides the string into individual words using spaces as



separators then creating an array called "wordsArray". The resulting array, which contains ["Geeks", "For", "Geeks"], is an output.

```
let sentence = "Geeks For Geeks";  
let wordsArray = sentence.split(" ");  
console.log(wordsArray);
```



## Output

```
[ 'Geeks', 'For', 'Geeks' ]
```

## 45. Write a switch statement code in JavaScript?

This code defines a variable course with the value "javascript". The switch statement checks the value of course. If it matches "javascript", it prints "This is a javascript course" to the console. If it does not match, the default case prints "Not a javascript course".

```
let course = "javascript";  
switch (course) {  
  case "javascript":  
    console.log("This is a javascript course");  
    break;  
  default:  
    console.log("Not a javascript course");  
}
```



## Output

```
This is a javascript course
```

## 46. Check if Two Strings are Anagrams or not in JavaScript?

The areAnagrams() function checks if two strings are anagrams. First it compares their lengths if they are not same then it returns false. Then, it counts character frequencies for each string. Then, it compares these counts. If they match, the strings are anagrams(true), otherwise they are not anagrams(false).

```
function areAnagrams(str1, str2) {
```

```
if (str1.length !== str2.length) {  
    return false;  
}  
  
let count1 = {};  
let count2 = {};  
  
// Count frequency of each character in str1  
for (let i = 0; i < str1.length; i++) {  
    let char = str1[i];  
    count1[char] = (count1[char] || 0) + 1;  
}  
  
// Count frequency of each character in str2  
for (let i = 0; i < str2.length; i++) {  
    let char = str2[i];  
    count2[char] = (count2[char] || 0) + 1;  
}  
  
// Compare the two frequency objects  
for (let char in count1) {  
    if (count1[char] !== count2[char]) {  
        return false;  
    }  
}  
  
return true;  
}  
console.log(areAnagrams("listen", "silent"));
```

## Output

true

## 47. Find the maximum difference between two numbers in an array in JavaScript?

The `maxDifference()` function finds the maximum difference between elements in an array, where the larger element comes after the smaller one. It initializes `min` variable with the first element of an array, then iterates through the array, updating the `maxDiff` and the minimum value.

```
function maxDifference(arr) {  
    let min = arr[0]  
    let maxDiff = 0;  
  
    for (let i = 1; i < arr.length; i++) {  
        const diff = arr[i] - min;  
        maxDiff = Math.max(maxDiff, diff);  
        min = Math.min(min, arr[i]);  
    }  
    return maxDiff;  
}  
  
console.log(maxDifference([1, 2, 90, 10, 110]));
```

## Output

109

## 48. Remove Duplicates from an Array in JavaScript?

The removeDuplicates function takes an array arr as input and creates a new array "uniqueArray". It iterates through arr and checking if each element is already in uniqueArray or not. If not, it adds the element in uniqueArray. Finally, it returns uniqueArray which contains only unique values from the original array.

```
function removeDuplicates(arr)  
{  
    const uniqueArray = [];  
    for (let i = 0; i < arr.length; i++)  
    {  
        if (!uniqueArray.includes(arr[i]))  
        {  
            uniqueArray.push(arr[i]);  
        }  
    }  
    return uniqueArray;  
}  
  
console.log(removeDuplicates([5, 2, 5, 6, 6, 7]));
```

## Output

```
[ 5, 2, 6, 7 ]
```

## 49. Count Vowels in a String in JavaScript?

The countVowels function counts the number of vowels in a given string. It initializes a count variable as 0 and iterates through each character of the string. If a character is found in the vowels string (which includes both lowercase and uppercase vowels), it increments the count variable. Atlast, it returns the total count.

```
function countVowels(str) {  
    let count = 0;  
    // Include both lowercase and uppercase vowels  
    const vowels = 'aeiouAEIOU';  
    for (let i = 0; i < str.length; i++) {  
        if (vowels.includes(str[i])) {  
            count++;  
        }  
    }  
  
    return count;  
}  
  
console.log(countVowels("hello geek"));
```

## Output

```
4
```

## 50. Get Unique Characters from a String in JavaScript?

The uniqueCharacters function takes a string as input and finds unique characters. It initializes an empty array, uniqueChars, and iterates through the string. If a character is not already in uniqueChars, it adds it. Finally, it joins the unique characters into a string and returns the string.

```
function uniqueCharacters(str) {  
    let uniqueChars = [];  
    for (let i = 0; i < str.length; i++) {  
        if (!uniqueChars.includes(str[i])) {  
            uniqueChars.push(str[i]);  
        }  
    }  
    return uniqueChars.join('');  
}
```

```
const uniqueChars = [];  
for (let i = 0; i < str.length; i++) {  
    if (!uniqueChars.includes(str[i])) {  
        uniqueChars.push(str[i]);  
    }  
}  
return uniqueChars.join('');  
}  
  
console.log(uniqueCharacters("geeksforgeeks"));
```

## Output

geksfor

[Comment](#)[More info](#)[Campus Training Program](#)

Corporate & Communications Address:

A-143, 7th Floor, Sovereign Corporate  
Tower, Sector- 136, Noida, Uttar Pradesh  
(201305)

Registered Address:

K 061, Tower K, Gulshan Vivante  
Apartment, Sector 137, Noida, Gautam  
Buddh Nagar, Uttar Pradesh, 201305



Advertise with us

## Company

About Us  
Legal  
Privacy Policy  
Careers  
Contact Us  
Corporate Solution  
Campus Training Program

## Tutorials

Programming Languages  
DSA  
Web Technology  
AI, ML & Data Science  
DevOps  
CS Core Subjects  
Interview Preparation  
GATE  
School Subjects

## Offline Centers

Noida  
Bengaluru  
Pune  
Hyderabad  
Patna

## Explore

POTD  
Job-A-Thon  
Connect  
Community  
Videos  
Blogs  
Nation Skill Up

## Courses

IBM Certification  
DSA and Placements  
Web Development  
Data Science  
Programming Languages  
DevOps & Cloud  
GATE  
Trending Technologies

## Preparation Corner

Aptitude  
Puzzles  
GfG 160  
DSA 360  
System Design

---

@GeeksforGeeks, Sanchhaya Education Private Limited, All rights reserved