

10 Helpful JavaScript Code Snippets



- | | |
|--|---|
| Write a function that reverses a string in JavaScript. | 2 |
| Write a function that checks whether a given string is a palindrome in JavaScript. | 2 |
| Write a function that returns the factorial of a given number in JavaScript. | 3 |
| Write a function that finds the largest number in an array in JavaScript. | 3 |
| Write a function that sorts an array of numbers in ascending order in JavaScript. | 3 |
| Write a function that calculates the sum of all numbers in an array in JavaScript. | 4 |
| Write a function that returns a new array with all duplicate values removed in JavaScript. | 4 |
| Write a function that returns the nth Fibonacci number in JavaScript. | 4 |

Write a function that checks whether a given number is prime in JavaScript.

5

Write a function that removes all falsy values (false, null, 0, "", undefined, and NaN) from an array in JavaScript.

6

Write a function that reverses a string in JavaScript.

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

Explanation: The function takes a string as input and splits it into an array of characters using the `split()` method. The `reverse()` method is then called on the array to reverse the order of its elements, and finally, the `join()` method is used to convert the array back into a string.

Write a function that checks whether a given string is a palindrome in JavaScript.

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

Explanation: The function first reverses the input string using the same approach as in the previous question. Then, it compares the reversed string to the original string and returns true if they are the same (i.e., the string is a palindrome).

Write a function that returns the factorial of a given number in JavaScript.

```
function factorial(num) {  
  if (num === 0 || num === 1) {  
    return 1;  
  } else {  
    return num * factorial(num - 1);  
  }  
}
```

Explanation: The function recursively calculates the factorial of a given number by multiplying it by the factorial of the previous number until it reaches 1 or 0 (in which case the factorial is 1).

Write a function that finds the largest number in an array in JavaScript.

```
function findMax(arr) {  
  return Math.max(...arr);  
}
```

Explanation: The function uses the `Math.max()` method along with the spread operator (`...`) to find the largest number in the input array.

Write a function that sorts an array of numbers in ascending order in JavaScript.

```
function sortArray(arr) {  
  return arr.sort((a, b) => a - b);  
}
```

Explanation: The function uses the `sort()` method on the input array with a comparator function that sorts the numbers in ascending order by subtracting `b` from `a`.

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

```
function sumArray(arr) {  
  return arr.reduce((acc, cur) => acc + cur, 0);  
}
```

Explanation: The function uses the `reduce()` method on the input array to accumulate the sum of all numbers in the array. The initial value of the accumulator is set to `0`.

Write a function that returns a new array with all duplicate values removed in JavaScript.

```
function removeDuplicates(arr) {  
  return [...new Set(arr)];  
}
```

Explanation: The function creates a new set from the input array using the `Set()` constructor, which automatically removes duplicates. The set is then spread into a new array using the spread operator (`...`).

Write a function that returns the `n`th Fibonacci number in JavaScript.

```
function fibonacci(n) {
```

```
if (n <= 1) {  
  return n;  
} else {  
  return fibonacci(n - 1) + fibonacci(n - 2);  
}  
}
```

Explanation: The function recursively calculates the nth Fibonacci number by adding the two previous Fibonacci numbers until it reaches the base cases of 0 or 1.

Write a function that checks whether a given number is prime in JavaScript.

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

Explanation: The function first checks if the input number is less than or equal to 1, in which case it is not prime. Then, it uses a for loop to iterate over all possible divisors of the number up to its square root. If the number is divisible by any of these divisors, it is not prime. Otherwise, the function returns true.

Write a function that removes all falsy values (false, null, 0, "", undefined, and NaN) from an array in JavaScript.

```
function removeFalsy(arr) {  
  return arr.filter(Boolean);  
}
```

Explanation: The function uses the filter() method on the input array with the built-in Boolean() function as the filter function. This filters out all falsy values from the array (since they evaluate to false when converted to a Boolean).