

Useful Code Snippets with JavaScript Learn more Code



- Write a function that converts a number to a string with commas separating thousands in JavaScript. **2**
- Write a function that removes the first occurrence of a specified element from an array in JavaScript. **2**
- Write a function that returns the last n elements of an array in JavaScript. **3**
- Write a function that generates a random alphanumeric string of a specified length in JavaScript. **3**
- Write a function that returns the sum of all numbers in an array in JavaScript. **4**
- Write a function that sorts an array of objects by a specified property in JavaScript. **4**
- Write a function that returns the longest word in a string in JavaScript. **5**
- Write a function that converts a string to title case (i.e., capitalizes the first letter of each word) in JavaScript. **5**

Write a function that returns the number of occurrences of a specified element in an array in JavaScript. **6**

Write a function that checks whether a given string is a palindrome (i.e., reads the same forwards and backwards) in JavaScript. **6**

Write a function that finds the first non-repeating character in a string in JavaScript. **6**

Write a function that converts a number to a string with commas separating thousands in JavaScript.

```
function addCommas(num) {  
    return  
    num.toString().replace(/\B(?=(\d{3})+(?!\d))/g, ",");  
}
```

Explanation: The function first converts the input number to a string using the `toString()` method. Then, it uses a regular expression `(/\B(?=(\d{3})+(?!\d))/g)` to match any digits that are not preceded by the beginning of a number (`\B`) and are followed by groups of three digits `((\d{3})+)` that are not followed by more digits `((?!\d))`. The `replace()` method then replaces each match with a comma.

Write a function that removes the first occurrence of a specified element from an array in JavaScript.

```
function removeElement(arr, element) {
```

```
const index = arr.indexOf(element);
if (index > -1) arr.splice(index, 1);
return arr;
}
```

Explanation: The function uses the `indexOf()` method to find the index of the first occurrence of the specified element in the input array. If it exists (i.e., its index is greater than -1), the `splice()` method is used to remove it from the array.

Write a function that returns the last n elements of an array in JavaScript.

```
function lastNElements(arr, n) {
  return arr.slice(-n);
}
```

Explanation: The function uses the `slice()` method with a negative index to return the last n elements of the input array.

Write a function that generates a random alphanumeric string of a specified length in JavaScript.

```
function generateRandomString(length) {
  const chars =
    "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01
    23456789";
  let result = "";
  for (let i = 0; i < length; i++) {
```

```
    result += chars.charAt(Math.floor(Math.random() *
chars.length));
  }
  return result;
}
```

Explanation: The function generates a random alphanumeric string of the specified length by first creating a string of all possible characters

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

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

Explanation: The function uses the `reduce()` method to add up all the numbers in the input array, starting with an initial value of 0.

Write a function that sorts an array of objects by a specified property in JavaScript.

```
function sortByProperty(arr, property) {
  return arr.sort((a, b) => a[property] - b[property]);
}
```

Explanation: The function sorts an array of objects by a specified property using the `sort()` method with a custom comparison function that subtracts the property value of the second object from that of the first object.

Write a function that returns the longest word in a string in JavaScript.

```
function longestWord(str) {  
  const words = str.split(" ");  
  return words.reduce((longest, word) => (word.length >  
longest.length ? word : longest), "");  
}
```

Explanation: The function first splits the input string into an array of words using the `split()` method. It then uses the `reduce()` method to compare the length of each word to the current longest word and return the longest one.

Write a function that converts a string to title case (i.e., capitalizes the first letter of each word) in JavaScript.

```
function toTitleCase(str) {  
  return str.replace(/\b\w/g, match =>  
match.toUpperCase());  
}
```

Explanation: The function uses a regular expression (`/\b\w/g`) to match the first letter of each word in the input string (`\b` matches a word boundary and `\w` matches a word character). The `replace()` method then replaces each match with its uppercase equivalent using a callback function.

Write a function that returns the number of occurrences of a specified element in an array in JavaScript.

```
function countOccurrences(arr, element) {  
  return arr.filter(item => item === element).length;  
}
```

Explanation: The function uses the `filter()` method to create a new array containing only the occurrences of the specified element and then returns its length.

Write a function that checks whether a given string is a palindrome (i.e., reads the same forwards and backwards) in JavaScript.

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

Explanation: The function first reverses the input string by splitting it into an array of characters, reversing the array, and joining it back into a string. It then compares the original string to the reversed string and returns true if they are the same.

Write a function that finds the first non-repeating character in a string in JavaScript.

```
function firstNonRepeatingChar(str) {
```

```
const counts = {};  
for (const char of str) {  
  counts[char] = counts[char] ? counts[char] + 1 : 1;  
}  
for (const char of str) {  
  if (counts[char] === 1) {  
    return char;  
  }  
}  
return null;  
}
```

Explanation: The function first creates an object to store the number of occurrences of each character in the input string. It then iterates through the string twice: once to count the occurrences of each character and once to find the first character with a count of 1.