

10 Helpful JavaScript Code Snippets



- Write a function that converts a string to title case (i.e., capitalizes the first letter of each word) in JavaScript. **2**
- Write a function that calculates the sum of all digits in a given integer in JavaScript. **2**
- Write a function that removes all whitespace from a given string in JavaScript. **3**
- Write a function that checks whether a given year is a leap year in JavaScript. **3**
- Write a function that returns the factorial of a given number in JavaScript. **4**
- Write a function that removes duplicates from an array in JavaScript. **4**
- Write a function that converts a number to a string with commas separating thousands in JavaScript. **4**
- Write a function that removes the first occurrence of a specified element from an array in JavaScript. **5**

Write a function that returns the last n elements of an array in JavaScript. **5**

Write a function that generates a random alphanumeric string of a specified length in JavaScript. **6**

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, c => c.toUpperCase());  
}
```

Explanation: The function uses the `replace()` method with a regular expression (`/\b\w/g`) to match the first character of each word in the input string (`\b` represents a word boundary and `\w` represents a word character). The second parameter of the `replace()` method is a function that returns the matched character converted to uppercase.

Write a function that calculates the sum of all digits in a given integer in JavaScript.

```
function sumDigits(num) {  
  return num.toString().split("").reduce((acc, cur) =>  
    acc + parseInt(cur), 0);  
}
```

Explanation: The function first converts the input integer to a string using the `toString()` method. Then, it uses the `split()` method to convert the string to an array of individual digits.

Finally, it uses the `reduce()` method to accumulate the sum of all digits in the array.

Write a function that removes all whitespace from a given string in JavaScript.

```
function removeWhitespace(str) {  
  return str.replace(/\s/g, "");  
}
```

Explanation: The function uses the `replace()` method with a regular expression (`/\s/g`) to match all whitespace characters in the input string and replace them with an empty string.

Write a function that checks whether a given year is a leap year in JavaScript.

```
function isLeapYear(year) {  
  return year % 4 === 0 && (year % 100 !== 0 || year %  
400 === 0);  
}
```

Explanation: The function uses the modulo operator (`%`) to check if the input year is divisible by 4. If it is, it then checks whether the year is divisible by 100 and not divisible by 400 to exclude years that are divisible by 100 but not by 400 (e.g., 1700, 1800, 1900).

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

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

Explanation: The function recursively calculates the factorial of a given number by multiplying it with the factorial of the preceding number until it reaches 0 or 1. The base cases handle negative numbers and 0 (whose factorial is defined as 1).

Write a function that removes duplicates from an array in JavaScript.

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

Explanation: The function creates a Set object from the input array to remove duplicates and then converts it back to an array using the spread operator (...).

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