JavaScript
Array
Methods

Made with ❤️ By **Prince Meghani**

1. Map

The map() method **creates a new array** with the **results of calling** a provided function on every element in the array.

```javascript
const numbers = [1, 2, 3, 4];
const squares = numbers.map(x => x * x);
console.log(squares);
// 👉 Output: [1, 4, 9, 16]
```

2. Filter

The filter() method **creates a new array** with all **elements that pass the test** implemented by the provided function.

```javascript
const numbers = [1, 2, 3, 4];
const evens = numbers.filter(n => n % 2 === 0);
console.log(evens);
// 👉 Output: [2, 4]
```

3. Find

The find() method **returns the value of the first element** in the array that satisfies the provided testing function. Otherwise, it **returns undefined**.

```javascript
const numbers = [1, 2, 3, 4];
const found = numbers.find(n => n > 2);
console.log(found);
// 👉 Output: 3
```

4. FindIndex

The findIndex() method returns the **index of the first element** in the array that satisfies the provided testing function. Otherwise, it **returns -1**.

```javascript
const numbers = [1, 2, 3, 4];
const index = numbers.findIndex(n => n > 2);
console.log(index);
// 👉 Output: 2
```

5. Fill

The fill() method **changes all elements in an array to a static value**, from a start index (default 0) to an end index (default array.length).

```javascript
const numbers = [1, 2, 3, 4];
numbers.fill(0, 2, 4);
console.log(numbers);
// 👉 Output: [1, 2, 0, 0]
```

6. Some

The some() method **tests whether at least one element in the array passes the test** implemented by the provided function.

```javascript
const numbers = [1, 2, 3, 4];
const hasNegativeNumbers = numbers.some(n => n < 0);
console.log(hasNegativeNumbers);
// 👉 Output: false
```

7. Every

The every() method **tests whether all elements in the array pass the test** implemented by the provided function. It returns a Boolean value.

```javascript
const numbers = [1, 2, 3, 4];
const allPositive = numbers.every(n => n > 0);
console.log(allPositive);
// 👉 Output: true
```

Follow
For Related Content

Keep sharing, keep learning!