

JavaScript Array Methods

Arrays are fundamental data structures in JavaScript, allowing you to store ordered collections of elements. JavaScript provides a rich set of methods to manipulate, iterate over, and transform arrays effectively. Here's a comprehensive guide to some of the most common array methods, along with explanations, syntax, and examples:

push(): Adds one or more elements to the end of an array and returns the new length.

```
const fruits = ["apple", "banana"];
fruits.push("orange", "mango");
console.log(fruits); // Output: ["apple", "banana", "orange", "mango"]
```

pop(): Removes the last element from an array and returns it. If the array is empty, pop() returns undefined.

```
const colors = ["red", "green", "blue"];
const removedColor = colors.pop();
console.log(colors); // Output: ["red", "green"]
console.log(removedColor); // Output: "blue"
```

shift(): Removes the first element from an array and returns it. If the array is empty, shift() returns undefined.

```
const numbers = [1, 2, 3, 4];
const removedNumber = numbers.shift();
console.log(numbers); // Output: [2, 3, 4]
console.log(removedNumber); // Output: 1
```

unshift(): Adds one or more elements to the beginning of an array and returns the new length.

```
const letters = ["b", "c"];
letters.unshift("a");
console.log(letters); // Output: ["a", "b", "c"]
```

join(): Converts an array of elements into a string, separated by a specified separator (comma by default).

```
const vegetables = ["tomato", "potato", "onion"];
const veggieString = vegetables.join(", ");
console.log(veggieString); // Output: "tomato, potato, onion"
```

```
const numbers = [1, 2, 3];
const numberString = numbers.join("-");
console.log(numberString); // Output: "1-2-3"
```

`indexOf(element)`: Returns the first index at which a given element can be found in the array, or -1 if it's not present.

```
const animals = ["dog", "cat", "rabbit", "cat"];
const firstCatIndex = animals.indexOf("cat"); // Case-sensitive
console.log(firstCatIndex); // Output: 1
```

`lastIndexOf(element)`: Returns the last index at which a given element can be found in the array, or -1 if it's not present.

```
const fruits = ["apple", "banana", "orange", "cat", "banana"];
const lastBananaIndex = fruits.lastIndexOf("banana");
console.log(lastBananaIndex); // Output: 4
```