## **Advanced Array Methods in JavaScript**

JavaScript provides several powerful array methods for performing operations efficiently. Below are some important ones with descriptions, examples, and a summary table at the end.

### **1. forEach()**

Executes a provided function once for each array element but does **not return a new array**.

const numbers = [1, 2, 3];

numbers.forEach(num => console.log(num \* 2));

// Output: 2, 4, 6

### **2. map()**

Creates a **new array** by applying a function to each element.

const numbers = [1, 2, 3];

const doubled = numbers.map(num => num \* 2);

console.log(doubled); // Output: [2, 4, 6]

### **3. filter()**

Creates a **new array** containing elements that pass a given condition.

const numbers = [10, 20, 30, 40];

const filtered = numbers.filter(num => num > 20);

console.log(filtered); // Output: [30, 40]

### **4. reduce()**

Reduces an array to a single value by applying a function.

const numbers = [1, 2, 3, 4];

const sum = numbers.reduce((acc, num) => acc + num, 0);

console.log(sum); // Output: 10

### **5. find()**

Returns **the first element** that satisfies a condition.

const numbers = [10, 20, 30, 40];

const result = numbers.find(num => num > 25);

console.log(result); // Output: 30

### **6. findIndex()**

Returns the **index** of the first element that satisfies a condition.

const numbers = [10, 20, 30, 40];

const index = numbers.findIndex(num => num > 25);

console.log(index); // Output: 2

### **7. some()**

Checks if **at least one** element satisfies a condition.

const numbers = [10, 20, 30];

const hasLargeNumber = numbers.some(num => num > 25);

console.log(hasLargeNumber); // Output: true

### **8. every()**

Checks if **all** elements satisfy a condition.

const numbers = [10, 20, 30];

const allLarge = numbers.every(num => num > 5);

console.log(allLarge); // Output: true

### **9. sort()**

Sorts the array **in place**. Default sorting is lexicographic (as strings), so numbers should be sorted using a compare function.

const numbers = [40, 100, 1, 5, 25, 10];

numbers.sort((a, b) => a - b); // Ascending order

console.log(numbers); // Output: [1, 5, 10, 25, 40, 100]

### **10. reverse()**

Reverses the array **in place**.

const numbers = [1, 2, 3];

numbers.reverse();

console.log(numbers); // Output: [3, 2, 1]

### **11. concat()**

Returns a **new array** by merging two or more arrays

const arr1 = [1, 2];

const arr2 = [3, 4];

const merged = arr1.concat(arr2);

console.log(merged); // Output: [1, 2, 3, 4]

### **12. slice()**

Returns a **new array** containing selected elements.

const numbers = [10, 20, 30, 40, 50];

const sliced = numbers.slice(1, 4);

console.log(sliced); // Output: [20, 30, 40]

### **13. splice()**

Modifies the array by adding/removing elements.

const numbers = [10, 20, 30];

numbers.splice(1, 1, 25, 27);

console.log(numbers); // Output: [10, 25, 27, 30]

### **14. includes()**

Checks if an element exists in an array.

const numbers = [10, 20, 30];

console.log(numbers.includes(20)); // Output: true

console.log(numbers.includes(50)); // Output: false

### **15. flat()**

Flattens nested arrays into a single-level array.

const nested = [1, [2, [3, 4]], 5];

const flatArray = nested.flat(2);

console.log(flatArray); // Output: [1, 2, 3, 4, 5]

### **16. from()**

Creates an array from an iterable or string.

const str = "Hello";

const arr = Array.from(str);

console.log(arr); // Output: ['H', 'e', 'l', 'l', 'o']

### **17. fill()**

Fills an array with a specific value.

const arr = new Array(5).fill(0);

console.log(arr); // Output: [0, 0, 0, 0, 0]

### **Summary Table**

| **Method** | **Returns New Array?** | **Modifies Original?** | **Use Case** |
| --- | --- | --- | --- |
| forEach() | ❌ No | ❌ No | Loop through elements |
| map() | ✅ Yes | ❌ No | Transform elements |
| filter() | ✅ Yes | ❌ No | Get elements that match a condition |
| reduce() | ❌ No | ❌ No | Reduce array to a single value |
| find() | ❌ No | ❌ No | Find first matching element |
| findIndex() | ❌ No | ❌ No | Find index of first match |
| some() | ❌ No | ❌ No | Check if at least one element meets condition |
| every() | ❌ No | ❌ No | Check if all elements meet condition |
| sort() | ❌ No | ✅ Yes | Sort elements |
| reverse() | ❌ No | ✅ Yes | Reverse order |
| concat() | ✅ Yes | ❌ No | Merge arrays |
| slice() | ✅ Yes | ❌ No | Extract portion of array |
| splice() | ❌ No | ✅ Yes | Add/remove elements |
| includes() | ❌ No | ❌ No | Check for value existence |
| flat() | ✅ Yes | ❌ No | Flatten nested arrays |
| from() | ✅ Yes | ❌ No | Convert iterable to array |
| fill() | ❌ No | ✅ Yes | Fill array with values |