Sure! Here's a handy guide to **common array methods in JavaScript** — especially useful in React when managing and displaying data. These methods help you work with arrays efficiently: adding, removing, updating, or transforming data.

### ◆ 1. `push()`

**Adds** an element to the **end** of an array.

javascript Copy Edit

```javascript
const fruits = ["Apple", "Banana"]; fruits.push("Orange"); console.log(fruits); // ["Apple",
"Banana", "Orange"]
```

### ◆ 2. `pop()`

**Removes** the **last** element from an array.

javascript Copy Edit

```javascript
fruits.pop(); console.log(fruits); // ["Apple", "Banana"]
```

### ◆ 3. `unshift()`

**Adds** an element to the **beginning** of an array.

javascript Copy Edit

```javascript
fruits.unshift("Mango"); console.log(fruits); // ["Mango", "Apple", "Banana"]
```

### ◆ 4. `shift()`

**Removes** the **first** element from an array.

javascript Copy Edit

```javascript
fruits.shift(); console.log(fruits); // ["Apple", "Banana"]
```

## ◆ **5.** `map()`

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

javascript Copy Edit

```javascript
const numbers = [1, 2, 3]; const doubled = numbers.map(num => num * 2); console.log(doubled); //
[2, 4, 6]
```

✅ Commonly used in React to render lists:

jsx Copy Edit

```jsx
{fruits.map(fruit => <li>{fruit}</li>)}
```

## ◆ **6.** `filter()`

Returns a **new array** with only the elements that pass a condition.

javascript Copy Edit

```javascript
const numbers = [1, 2, 3, 4]; const even = numbers.filter(num => num % 2 === 0);
console.log(even); // [2, 4]
```

## ◆ **7.** `find()`

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

javascript Copy Edit

```javascript
const users = [{ id: 1 }, { id: 2 }]; const user = users.find(u => u.id === 2); console.log(user);
// { id: 2 }
```

## ◆ 8. includes()

Checks if an array contains a value.

javascript Copy Edit

```javascript
const colors = ["red", "green", "blue"]; console.log(colors.includes("green")); // true
```

## ◆ 9. forEach()

Runs a function on **each item** in the array. (No return value.)

javascript Copy Edit

```javascript
fruits.forEach(fruit => console.log(fruit));
```

## ◆ 10. slice()

Returns a **portion** of the array (does **not** modify original).

javascript Copy Edit

```javascript
const someFruits = fruits.slice(0, 2);
```

## ◆ 11. splice()

**Changes** the original array by adding/removing items.

javascript Copy Edit

```javascript
fruits.splice(1, 1, "Kiwi"); // remove 1 at index 1 and add "Kiwi"
```

## ◆ 12. `reduce()`

Applies a function to accumulate a **single value** from an array.

javascript Copy Edit

```javascript
const total = [1, 2, 3].reduce((sum, val) => sum + val, 0); console.log(total); // 6
```