

Array methods (push, pop, shift, unshift, splice, slice)

JavaScript Array Methods: `push`, `pop`, `shift`, `unshift`, `splice`, `slice` — Explained in Detail

These core array methods help you add, remove, or extract elements from an array. Let's explore each with syntax, examples, and use cases.

✓ 1. `push()`

Adds one or more elements to the **end** of an array.

```
let fruits = ["apple", "banana"];
fruits.push("cherry");

console.log(fruits); // ["apple", "banana", "cherry"]
```

 Returns the new length of the array.

✓ 2. `pop()`

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

```
let fruits = ["apple", "banana", "cherry"];
let removed = fruits.pop();

console.log(removed); // "cherry"
console.log(fruits); // ["apple", "banana"]
```

 Returns the removed element.

✓ 3. **shift()**

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

```
let colors = ["red", "green", "blue"];
let first = colors.shift();

console.log(first); // "red"
console.log(colors); // ["green", "blue"]
```

⚡ Slower than pop() for large arrays (as it re-indexes elements).

✓ 4. **unshift()**

Adds one or more elements to the **beginning** of the array.

```
let colors = ["green", "blue"];
colors.unshift("red");

console.log(colors); // ["red", "green", "blue"]
```

⚡ Returns the new length of the array.

✓ 5. **splice()**

Adds or removes elements at a specified index.

⚠ Modifies the original array.

```
let nums = [1, 2, 3, 4, 5];

// Remove 2 elements starting from index 1
let removed = nums.splice(1, 2);
```

```
console.log(removed); // [2, 3]
console.log(nums);    // [1, 4, 5]
```

✨ Add elements with `splice` :

```
let arr = [1, 4, 5];

// Insert at index 1
arr.splice(1, 0, 2, 3);

console.log(arr); // [1, 2, 3, 4, 5]
```

✅ 6. `slice()`

Returns a **shallow copy** of a portion of an array.

❌ Does **not** modify the original array.

```
let letters = ["a", "b", "c", "d", "e"];

let part = letters.slice(1, 4);

console.log(part); // ["b", "c", "d"]
console.log(letters); // Original array stays the same
```

✅ `slice(start, end)` copies from start index up to but not including end.

📖 Summary Table

Method	Action	Modifies Original?	Returns
<code>push()</code>	Add to end	✅ Yes	New length
<code>pop()</code>	Remove from end	✅ Yes	Removed item
<code>shift()</code>	Remove from start	✅ Yes	Removed item

<code>unshift()</code>	Add to start	✓ Yes	New length
<code>splice()</code>	Add/remove at index	✓ Yes	Removed items
<code>slice()</code>	Copy part of array	✗ No	New array

Use Cases

- `push` / `pop` → Stack operations
- `shift` / `unshift` → Queue operations
- `splice` → Insert, replace, or remove elements
- `slice` → Extract portions of arrays (pagination, cloning)

Let me know if you want a visual cheat sheet, practice problems, or comparison with functional array methods like `map`, `filter`, etc.