


ARRAY IN JS

Arrays are the collections of items of same datatypes.

- **Create: let arr = [1, 2, 3];**
- **Common methods:**
 - **push(), pop()**
 - **shift(), unshift()**
 - **forEach()**
 - **map(), filter(), reduce()**


```
let nums = [1, 2, 3];  
let doubled = nums.map(n => n * 2);
```

- **push()**

- **What it does:** Adds one or more elements to the end of an array.
- **Changes the original array?**  Yes
- **Returns:** The new length of the array.

```
let fruits = ["apple", "banana"];  
let length = fruits.push("orange");  
console.log(fruits); // ["apple", "banana", "orange"]  
console.log(length); // 3
```

- **pop()**

- **What it does:** Removes the last element from an array.
- **Changes the original array?**  Yes
- **Returns:** The removed element.

```
let fruits = ["apple", "banana", "orange"];  
let last = fruits.pop();  
console.log(fruits); // ["apple", "banana"]  
console.log(last);   // "orange"
```

- **shift()**

- **What it does:** Removes the first element from an array.
- **Changes the original array?**  Yes
- **Returns:** The removed element.

```
let fruits = ["apple", "banana", "orange"];  
let first = fruits.shift();  
console.log(fruits); // ["banana", "orange"]  
console.log(first);  // "apple"
```

- **unshift()**

- **What it does:** Adds one or more elements to the start of an array.
- **Changes the original array?**  Yes
- **Returns:** The new length of the array.

```
let fruits = ["banana", "orange"];  
let length = fruits.unshift("apple");  
console.log(fruits); // ["apple", "banana", "orange"]  
console.log(length); // 3
```

- **forEach()**

- **What it does:** Loops through each array element and runs a function.
- **Changes the original array?** **✗ No** (unless you modify it manually inside the loop).
- **Returns:** Nothing (undefined).

```
let fruits = ["apple", "banana", "orange"];
fruits.forEach((item, index) => {
  console.log(index, item);
});
// Output:
// 0 apple
// 1 banana
// 2 orange
```

- **map()**

- **What it does:** Creates a new array by applying a function to each element.
- **Changes the original array?** ❌ No
- **Returns:** A new array.

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

- **filter()**

- **What it does:** Creates a new array with elements that pass a condition.
- **Changes the original array?** ❌ No
- **Returns:** A new filtered array.

```
let numbers = [1, 2, 3, 4, 5];  
let evens = numbers.filter(num => num % 2 === 0);  
console.log(evens); // [2, 4]  
console.log(numbers); // [1, 2, 3, 4, 5]
```

- **reduce()**
 - **What it does:** Reduces an array to a single value by running a function for each element.
 - **Changes the original array?** **✗ No**
 - **Returns:** A single value.

```
let numbers = [1, 2, 3, 4];  
let sum = numbers.reduce((accumulator, current) => accumulator + current, 0);  
console.log(sum); // 10
```

How reduce works here:

- **Start with accumulator = 0**
- **Step 1: $0 + 1 \rightarrow 1$**
- **Step 2: $1 + 2 \rightarrow 3$**
- **Step 3: $3 + 3 \rightarrow 6$**
- **Step 4: $6 + 4 \rightarrow 10$**

