

JavaScript Arrays: Full Tutorial

1. What is an Array?

An Array is a special variable that can hold more than one value at a time.

Example:

```
let fruits = ["Apple", "Banana", "Orange"];
```

2. Creating Arrays

- Using array literals: `let cars = ["Toyota", "Honda", "Ford"];`
- Using new keyword: `let cars = new Array("Toyota", "Honda", "Ford");`

3. Accessing & Modifying Elements

Access: `cars[0] // "Toyota"`

Modify: `cars[1] = "BMW";`

4. Array Properties & Looping

- length: `arr.length`
- Looping: `for`, `for...of`, `forEach`

5. Common Methods

- `push()`, `pop()`, `shift()`, `unshift()`
- `indexOf()`, `includes()`, `find()`, `findIndex()`
- `map()`, `filter()`, `reduce()`
- `sort()`, `reverse()`, `concat()`, `slice()`, `splice()`
- `flat()`, `Array.isArray()`, spread operator

6. Advanced Methods & Tricks

- `flatMap()`: `arr.flatMap(x => [x * 2])`
- `from()`: `Array.from("hello") -> ["h", "e", "l", "l", "o"]`
- `every()`, `some()`, `fill()`, `copyWithin()`

JavaScript Arrays: Full Tutorial

- Destructuring: `let [a, b] = [1, 2]`
- Spread operator: `[...arr1, ...arr2]`

7. Practice Problems

1. Write a function to remove duplicates from an array.
2. Find the second largest number in an array.
3. Flatten a deeply nested array recursively.
4. Create a function that returns the frequency of elements.
5. Implement a custom map function without using `map()`.
6. Use `reduce()` to group items by property (e.g., age).
7. Sort an array of objects by a key (e.g., name).