Array in Javascript

1. What is an Array?

- **Definition:** A list-like object for storing multiple values in a single variable.
- **Syntax:**

```
let arr = [10, 20, 30];
```

- Arrays can hold mixed types: numbers, strings, booleans, even other arrays or objects.
- **Access & modify:** Use index, zero-based:

```
console.log(arr[0]); // 10
arr[1] = 25;        // Now arr = [10, 25, 30]
```

2. Array Properties

- **length:** Gives number of elements.

```
console.log(arr.length); // 3
```

3. Adding & Removing Items

- **push():** Add at end.

```
arr.push(40); // [10, 25, 30, 40]
```

- **pop():** Remove from end.

```
arr.pop(); // returns 40, arr = [10, 25, 30]
```

- **unshift():** Add at start.

```
arr.unshift(5); // [5, 10, 25, 30]
```

- **shift():** Remove from start.

```
arr.shift(); // returns 5, arr = [10,25,30]
```

4. Searching for Values

- **indexOf(item):** Returns first index or -1 if not found.

```
arr.indexOf(25); // 1
```

- **includes(item):** true or false.

```
arr.includes(50); // false
```

5. Looping Through Arrays

- **for loop:**

```
for (let i = 0; i < arr.length; i++) {
  console.log(arr[i]);
}
```

- **forEach():**

```
arr.forEach(function(item, index) {
  console.log(index, item);
});
```

6. Important Array Methods

- **map():** Transforms items and returns a new array.

```
let doubled = arr.map(x ⇒ x * 2);
```

- **filter():** Returns items that satisfy a condition.

```
let big = arr.filter(x ⇒ x > 20);
```

- **reduce():** Aggregates values.

```
let sum = arr.reduce((acc, curr) ⇒ acc + curr, 0);
```

7. Real-Life Use Case

- Dynamic lists (like to-do apps).
- Data transforms (e.g. calculating totals).
- Filtering based on criteria (e.g. products under ₹500).

✅ Key Takeaways: Beginner Notes

Concept	Explanation
Array Declaration	`let arr = [value1, value2]`
Access Elements	`arr[index]`
Modify Elements	`arr[index] = newValue`
Add Elements	`.push()` / `.unshift()`
Remove Elements	`.pop()` / `.shift()`
Search Elements	`.indexOf()` / `.includes()`
Looping	`for` , `.forEach()`
Transform Arrays	`.map()` , `.filter()` , `.reduce()`

Example Code

```
let nums = [1, 2, 3];
nums.push(4);          // [1,2,3,4]
```

```
let evens = nums.filter(x ⇒ x % 2 === 0);  // [2,4]
let sum = nums.reduce((a, b) ⇒ a + b, 0);  // 10
```

📘 Notes for Revision (Copy-Paste Friendly)

- Array = list of values: let arr = [a, b, c]
- Access via arr[0], arr[1], ..., length via arr.length
- Add/remove: push(), pop(), unshift(), shift()
- Search: arr.indexOf(item), arr.includes(item)
- Loop: for(i=0;i<arr.length;i...), arr.forEach((it,i)⇒...)
- Transforms:
  - arr.map(x ⇒ ...) – creates new array
  - arr.filter(x ⇒ condition) – subset
  - arr.reduce((acc,cur)⇒..., init) – single value
- Example:

```
let nums=[1,2,3];
nums.push(4);                // [1,2,3,4]
let evens = nums.filter(x⇒x%2===0);
let sum = nums.reduce((a,b)⇒a+b, 0);
```
