Array Masterclass

Instructor: Pramod Kumar Jena

---

1. Introduction to Arrays

What is an Array?

- **Definition**: An array is a collection of elements stored in a single variable. Arrays can hold multiple values at once, making them an efficient way to organize data.
- **Structure**: Arrays are ordered, meaning each element has an index. In JavaScript, indices start from `0`.

Why Use Arrays?

- **Data Organization**: Useful for storing lists of related information (like a list of names, ages, or scores).
- **Efficient Data Access**: Quickly access, add, modify, or remove data.

Real-Life Examples:

1. **Library System**: An array holds a list of book titles in a library.
2. **Calendar App**: An array can represent a week's days or appointments.
3. **Grocery List**: A shopper’s list of items to buy.

---

2. Creating Arrays

Methods of Creating Arrays:

Literal Notation:
```
let fruits = ["apple", "banana", "mango"];
```

Using `new Array()`:
```
let numbers = new Array(1, 2, 3, 4, 5);
```

Empty Array:
```
let emptyArray = [];
```

Example:

Create a list of cities:

```
let cities = ["New York", "London", "Tokyo", "Sydney"];
```

---

3. Accessing and Modifying Array Elements

Accessing Elements:
Access elements using index notation.

```
console.log(fruits[0]); // "apple"
```

Modifying Elements:
Directly assign a new value to an element.

```
fruits[1] = "strawberry";
```

Adding and Removing Elements:
Using `push()`: Adds to the end.

```
fruits.push("grape");
```

Using `pop()`: Removes from the end.

```
fruits.pop();
```

---

4. Array Methods: Core Operations

4.1 `push()` and `pop()`
`push()` adds elements to the end.

```
fruits.push("pineapple");
```

`pop()` removes the last element.

```
fruits.pop();
```

4.2 `unshift()` and `shift()`

`unshift()` adds elements to the beginning.

```
fruits.unshift("kiwi");
```

`shift()` removes the first element.

```
fruits.shift();
```

4.3 `concat()`

Merges two or more arrays.

```
let combined = fruits.concat(["pear", "peach"]);
```

4.4 `slice()`

Extracts a portion without modifying the original.

```
let tropicalFruits = fruits.slice(1, 3);
```

4.5 `splice()`

Adds/removes elements at a specified position.

```
fruits.splice(1, 0, "papaya"); // Adds "papaya" at index 1
```

---

5. Advanced Array Methods

5.1 `forEach()`

Executes a function for each element.

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

5.2 `map()`

Returns a new array with modified elements.

```
let upperFruits = fruits.map(fruit => fruit.toUpperCase());
```

5.3 `filter()`

Creates a new array with elements that pass a test.

```
let filteredFruits = fruits.filter(fruit => fruit.startsWith("p"));
```

5.4 `reduce()`

Reduces all elements to a single value.

```
let totalLength = fruits.reduce((sum, fruit) => sum + fruit.length,
0);
```

5.5 `find()` and `findIndex()`

`find()` returns the first match.

```
let mango = fruits.find(fruit => fruit === "mango");
```

`findIndex()` returns the index of the first match.

```
let mangoIndex = fruits.findIndex(fruit => fruit === "mango");
```

5.6 `sort()`

Sorts elements.

```
let sortedFruits = fruits.sort();
```

5.7 `reverse()`

Reverses the order of elements.

```
fruits.reverse();
```

---

6. Real-Life Array Examples

Example 1: Storing Scores in a Game

```
let scores = [95, 88, 73, 84, 99];
let averageScore = scores.reduce((sum, score) => sum + score) /
scores.length;
```

Example 2: Creating a To-Do List App

```
let tasks = ["Buy groceries", "Do laundry", "Study JavaScript"];
tasks.push("Walk the dog");
tasks.splice(1, 1, "Do the dishes"); // Modify a task
```

---

7. Common Interview Patterns and Exercises

Pattern 1: Printing a Triangle

```
for (let i = 1; i <= 5; i++) {
  console.log('*'.repeat(i));
}
```

Pattern 2: Printing an Inverted Triangle

```
for (let i = 5; i >= 1; i--) {
  console.log('*'.repeat(i));
}
```

Exercise 1: Sum of an Array

```
let numbers = [10, 20, 30, 40];
let sum = numbers.reduce((acc, num) => acc + num, 0);
```

Exercise 2: Finding the Maximum Number

```
let max = Math.max(...numbers);
```

Exercise 3: Count Occurrences in an Array

```
let fruits = ["apple", "banana", "apple", "orange", "banana"];
let count = fruits.reduce((acc, fruit) => {
  acc[fruit] = (acc[fruit] || 0) + 1;
  return acc;
}, {});
```

Exercise 4: Reverse Array

```
let reversed = fruits.reverse();
```

---

8. Hands-On Exercises and Q&A

Exercise: Student Attendance

Track and update attendance using an array:

```
let attendance = [true, false, true, true, false];
let presentStudents = attendance.filter(status => status).length;
```

Exercise: Real-Time Updates Using Arrays

Demonstrate dynamic updates to an array by adding, removing, and modifying elements in real-time with user input.

Exercise: Inventory System

Build a simple inventory management example with array methods to add, remove, and display items.