

Task 4: Arrays and Objects — the backbone of working with data in any application, including React and backend services.

✅ Task 4: Arrays & Objects in JavaScript

🎯 Goals:

- Understand how to create, access, and manipulate arrays and objects
 - Use important built-in methods for real-world data tasks
 - Prepare for working with JSON and backend data
-

◆ Part A: Objects

📘 What is an Object?

An object is a collection of **key-value pairs**. It's used to group related data and functions.

🧩 Example:

```
const user = {  
  name: "Ali",  
  age: 25,  
  isAdmin: true,  
  greet: function () {  
    console.log("Hello!");  
  }  
};
```

🔍 Key Concepts

✅ 1. Accessing Properties

```
console.log(user.name); // "Ali"  
console.log(user["age"]); // 25
```

✅ 2. Adding/Updating/Deleting

```
user.email = "ali@email.com"; // Add
```

```
user.age = 26; // Update
```

```
delete user.isAdmin; // Delete
```

✅ 3. Object Methods

Functions inside objects are called **methods**.

```
const car = {  
  start: function() {  
    console.log("Engine started.");  
  }  
};  
  
car.start(); // Call method
```

✅ 4. for...in Loop (object keys)

```
for (let key in user) {  
  console.log(` ${key}: ${user[key]} `);  
}
```

◆ Part B: Arrays

📘 What is an Array?

An array is an ordered list of values (elements), used to store collections.

🧱 Example:

```
const services = ["Car Wash", "Plumbing", "Electrician"];
```

🔍 Key Concepts

✅ 1. Access Elements by Index

```
console.log(services[0]); // "Car Wash"
```

✅ 2. Modify Arrays

```
services.push("Painting"); // Add to end
```

```
services.pop(); // Remove last
```

```
services.unshift("Gardening"); // Add to start
```

```
services.shift(); // Remove first
```

✅ 3. Looping Over Arrays

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

```
for (let service of services) {  
  console.log(service);  
}
```

✅ 4. Important Array Methods

- `.map()`: Transforms each element and returns a new array
- `.filter()`: Returns a new array with elements that match a condition
- `.reduce()`: Reduces the array to a single value

```
// map
```

```
const upperServices = services.map(service => service.toUpperCase());
```

```
// filter
```

```
const shortServices = services.filter(service => service.length < 10);
```

```
// reduce
```

```
const numbers = [1, 2, 3];
```

```
const sum = numbers.reduce((total, num) => total + num, 0); // 6
```

Practice Exercises

✅ Exercise 1: Object Basics

Create an object customer with:

- name, email, isPremium
 - Add a method greet() that logs "Welcome, [name]!"
-

✅ Exercise 2: Array Methods

Create an array of 5 service names. Use:

- .map() to make them uppercase
 - .filter() to return only those with more than 7 letters
 - .reduce() to count total number of characters in all services
-

✅ Exercise 3: for...in Loop

Write a function printDetails(obj) that takes an object and prints each key and value using a for...in loop.

Reflection Questions

1. What is the difference between dot notation and bracket notation when accessing object properties?
2. When would you use .map() vs .forEach()?
3. What is the difference between .push() and .unshift()?