Arrays & Objects

Why Use Arrays and Objects?

In JavaScript, we often need to **store and manage multiple values**. Instead of creating separate variables for each piece of data, we can use **arrays** and **objects** to organize information efficiently.

- **Arrays** → best for **ordered lists** of items.

- **Objects** → best for **describing entities** with properties.

Arrays

An *array* is a list-like object that can hold multiple values in a single variable.

Creating an Array:

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

Accessing Items:

```
console.log(fruits[0]); // "Apple"
console.log(fruits[2]); // "Mango"
```

Common Array Methods:

- push() → Add item to the end

- pop() → Remove item from the end

- shift() → Remove item from the start

- unshift() → Add item to the start

- length → Get number of items

- forEach() → Loop through items

```
fruits.push("Orange"); // ["Apple", "Banana", "Mango", "Orange"]
fruits.pop();          // removes "Orange"
```

Objects

An *object* stores data in **key-value pairs**.

Creating an Object:

```
let person = {
  name: "Alice",
  age: 25,
  isStudent: true
};
```

Accessing Properties:

console.log(person.name); // "Alice"

console.log(person["age"]); // 25

Adding/Updating Properties:

```
person.country = "Philippines";
person.age = 26;
```

Arrays of Objects

Often, we combine arrays and objects to handle structured data.

```
let students = [
  { name: "John", grade: 90 },
  { name: "Maria", grade: 85 },
  { name: "David", grade: 92 }
];

console.log(students[1].name); // "Maria"
```

Looping Through Arrays and Objects

Array Example:

```
let colors = ["Red", "Green", "Blue"];
colors.forEach(function(color) {
  console.log(color);
});
```

Object Example:

```
for (let key in person) {
  console.log(key + ": " + person[key]);
}
```

When to Use Arrays vs Objects?

- Use an **array** when you just need a list of items.

- Use an **object** when you need to describe something with **properties**.

- Use **arrays of objects** when working with structured collections (e.g., a list of users, products, or records).

Ex:

Simulating a **shopping cart** where we calculate the total cost.

```javascript
let cart = [
  { item: "Laptop", price: 50000 },
  { item: "Mouse", price: 800 },
  { item: "Keyboard", price: 1500 }
];

let total = 0;
cart.forEach(function(product) {
  total += product.price;
});

console.log("Total Price: " + total);
```