

## Lesson 3: Arrays and Objects in JavaScript

### Objective

In this lesson, you'll learn about two of the most fundamental data structures in JavaScript: arrays and objects. By the end of this lesson, you'll be able to store and manipulate collections of data using arrays and objects, and understand how to access and modify data within them.

## 1. Arrays

### What are Arrays?

- Arrays are ordered collections of values. They are used to store multiple values in a single variable, and these values can be accessed by their index.

#### 1.1 Creating Arrays

##### Array Declaration

- You can create an array using square brackets `[]` or the `Array` constructor.

```
let fruits = ["Apple", "Banana", "Cherry"];
let numbers = new Array(1, 2, 3, 4, 5);
```

##### Accessing Array Elements

- Array elements can be accessed by their index, with the first element having an index of `0`.

```
console.log(fruits[0]); // Output: Apple
console.log(numbers[2]); // Output: 3
```

#### 1.2 Array Methods

##### Common Array Methods

- **push()** : Adds an element to the end of an array.

```
fruits.push("Orange");
console.log(fruits); // Output: ["Apple", "Banana", "Cherry", "Orange"]
```

- **pop()** : Removes the last element from an array.

```
fruits.pop();
console.log(fruits); // Output: ["Apple", "Banana", "Cherry"]
```

- **shift()** : Removes the first element from an array.

```
fruits.shift();  
console.log(fruits); // Output: ["Banana", "Cherry"]
```

- **unshift()** : Adds an element to the beginning of an array.

```
fruits.unshift("Mango");  
console.log(fruits); // Output: ["Mango", "Banana", "Cherry"]
```

- **length** : Returns the number of elements in an array.

```
console.log(fruits.length); // Output: 3
```

- **indexOf()** : Returns the index of a specified element in the array, or `-1` if it is not found.

```
console.log(fruits.indexOf("Banana")); // Output: 1
```

- **slice()** : Returns a new array containing a portion of the original array.

```
let someFruits = fruits.slice(1, 3);  
console.log(someFruits); // Output: ["Banana", "Cherry"]
```

## 2. Objects

### What are Objects?

- Objects are collections of key-value pairs, where each key (also called a property) is associated with a value. Objects allow you to group related data and functions together.

### 2.1 Creating Objects

#### Object Literal Syntax

- Objects can be created using curly braces `{}` with key-value pairs separated by a colon `:`. Keys are typically strings, and values can be any data type.

```
let person = {  
  firstName: "John",  
  lastName: "Doe",  
  age: 30,  
  isStudent: false  
};
```

#### Accessing Object Properties

- Object properties can be accessed using dot notation or bracket notation.

```
console.log(person.firstName); // Output: John  
console.log(person["age"]);    // Output: 30
```

## 2.2 Modifying Objects

### Adding and Modifying Properties

- You can add new properties to an object or modify existing ones.

```
person.firstName = "Jane"; // Modify property
person.job = "Developer"; // Add new property
console.log(person);
// Output: { firstName: "Jane", lastName: "Doe", age: 30, isStudent: false,
job: "Developer" }
```

### Deleting Properties

- Properties can be removed from an object using the `delete` keyword.

```
delete person.isStudent;
console.log(person);
// Output: { firstName: "Jane", lastName: "Doe", age: 30, job: "Developer" }
```

## 3. Arrays of Objects

### What are Arrays of Objects?

- You can store objects inside arrays, allowing you to manage collections of data with multiple attributes.

```
let students = [
  { name: "Alice", age: 20 },
  { name: "Bob", age: 22 },
  { name: "Charlie", age: 23 }
];
```

### Accessing Objects in Arrays

- You can access objects in arrays by their index and then access their properties.

```
console.log(students[0].name); // Output: Alice
```

### Looping Through Arrays of Objects

- Use loops to iterate through arrays of objects and access their properties.

```
for (let i = 0; i < students.length; i++) {
  console.log(students[i].name + " is " + students[i].age + " years old.");
}
```

## 4. Hands-On Practice

### Exercise 1: Array Manipulation

- Create an array of your favorite fruits. Add a new fruit to the end of the array, remove the first fruit, and print the final array.

### Exercise 2: Working with Objects

- Create an object to represent a book with properties for title, author, and year published. Add a new property for the genre, and then delete the year published property.

### Exercise 3: Array of Objects

- Create an array of objects to represent a list of movies with properties for title, director, and year. Write a function that prints the title of each movie released after the year 2000.

## 5. Homework/Assignment

### Assignment 1: Create a To-Do List

- Create an array of objects where each object represents a to-do item with properties for the task description and whether it is completed. Write a function to add a new to-do item, mark an item as completed, and display all tasks.

### Assignment 2: Simple Inventory System

- Create an inventory system where you store objects representing items in an array. Each object should have properties for item name, quantity, and price. Write functions to add items to the inventory, update the quantity of an item, and calculate the total value of the inventory.

## 6. Recommended Resources

### Documentation and Tutorials

- [MDN Web Docs: Arrays](#)
  - An in-depth look at JavaScript arrays, including all methods and properties, with examples.
- [MDN Web Docs: Objects](#)
  - Comprehensive documentation on JavaScript objects, explaining how to create, access, and manipulate objects.
- [JavaScript.info: Arrays](#)
  - A beginner-friendly guide to understanding arrays, including useful array methods and examples.
- [JavaScript.info: Objects](#)
  - Detailed explanations and examples on working with objects in JavaScript.

### Video Tutorials

- [Traversy Media: JavaScript Arrays Crash Course](#)
  - A video tutorial covering the basics of JavaScript arrays, including methods and common use cases.
- [Academind: JavaScript Objects in Detail](#)

- A thorough introduction to JavaScript objects, explaining their structure, use cases, and how to work with them effectively.