

10. Arrays and Objects – Working with Data

Arrays and Objects – Working with Data

1. Arrays in JavaScript

An **array** is an ordered collection of elements that can store multiple values under a single variable name. Arrays are indexed, meaning each element has a numerical position (starting from 0). They allow storage of various data types, including numbers, strings, objects, and even other arrays. JavaScript arrays are dynamic, meaning they can grow or shrink as needed.

1.1 Array Characteristics

- Indexed from 0 (zero-based indexing).
- Can contain mixed data types.
- Resizable dynamically.
- Have built-in methods for manipulation.
- Stored in contiguous memory locations, providing efficient access.

1.2 Array Methods

JavaScript provides several methods for manipulating arrays:

- **push()** and **pop()**: Adding and removing elements from the end.
- **shift()** and **unshift()**: Adding and removing elements from the beginning.
- **slice()**: Extracting a portion of an array without modifying the original.
- **splice()**: Adding or removing elements at a specific index.
- **concat()**: Merging multiple arrays.
- **sort()**: Sorting array elements.
- **reverse()**: Reversing the order of elements.

2. Higher-Order Array Methods

Higher-order methods are functions that operate on arrays to transform or filter data. They provide a functional programming approach to data manipulation.

2.1 map()

A method that creates a new array by applying a function to each element. It does not modify the original array.

2.2 filter()

Creates a new array containing only elements that satisfy a given condition.

2.3 reduce()

Applies a function to each element, accumulating the result into a single value. It is useful for computing sums, averages, or other aggregate values.

3. Objects in JavaScript

An **object** is a collection of key-value pairs where each key (also called a property) maps to a specific value. Objects provide a structured way to store and manage data.

3.1 Object Characteristics

- Keys are strings (or symbols).
- Values can be of any data type (strings, numbers, functions, or even other objects).
- Used to represent real-world entities like users, products, and configurations.
- Can contain methods (functions as values).
- Allow nesting (objects within objects).

3.2 Object Properties and Methods

- Properties define the attributes of an object.

- Methods are functions associated with an object.
- Objects can be modified dynamically by adding or deleting properties.

3.3 Object Manipulation

- Accessing properties using dot notation or bracket notation.
- Adding new properties dynamically.
- Deleting properties using the delete operator.
- Checking for the existence of a property using the in operator.

4. Prototype and Prototype Chain

JavaScript uses **prototype-based inheritance**, where objects inherit properties and methods from a prototype. Each object in JavaScript has a hidden `__proto__` property pointing to its prototype.

4.1 Understanding Prototypes

- Objects can inherit methods from their prototype.
- If a property or method is not found in the object, JavaScript looks for it in the prototype.
- Modifying the prototype affects all instances of an object.

4.2 Prototype Chain

- When accessing a property or method, JavaScript searches the object itself first.
- If not found, it checks the object's prototype.
- This process continues up the prototype chain until the property is found or null is reached.

5. Iterating Over Arrays and Objects

5.1 Iterating Over Arrays

- **forEach()**: Executes a function for each element in the array.
- **for...of**: Iterates over values of an array.
- **map()**: Returns a new transformed array.
- **filter()**: Returns a subset of the array.

5.2 Iterating Over Objects

- **for...in**: Iterates over object properties.
- **Object.keys()**: Retrieves an array of keys.
- **Object.values()**: Retrieves an array of values.
- **Object.entries()**: Retrieves an array of key-value pairs.

6. Key Concepts Summary

Concept	Description
Arrays	Ordered collection of values.
Objects	Key-value pairs for structured data.
Array Methods	Manipulation tools like push, pop, map, reduce.
Prototype	Mechanism for inheritance in JavaScript.
Iteration	Methods to loop over arrays and objects.