

Day 3 → JavaScript Arrays and Objects:

JavaScript Arrays:

Arrays in JavaScript are used to store multiple values in a single variable. They can hold different types of data, such as numbers, strings, objects, and even other arrays. Arrays are indexed collections, where each item is assigned a numeric index starting from 0.

Example:

```
```javascript
```

```
// Creating an array
```

```
var fruits = ["Apple", "Banana", "Orange", "Mango"];
```

```
// Accessing array elements
```

```
console.log(fruits[0]); // Output: "Apple"
```

```
console.log(fruits[2]); // Output: "Orange"
```

```
// Modifying array elements
```

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

```
console.log(fruits); // Output: ["Apple", "Grapes", "Orange",
"Mango"]
```

```
// Array length
```

```
console.log(fruits.length); // Output: 4
```

```
// Adding elements to an array
```

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

```
console.log(fruits); // Output: ["Apple", "Grapes", "Orange",
"Mango", "Pineapple"]
```

```
// Removing elements from an array
```

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

```
console.log(fruits); // Output: ["Apple", "Grapes", "Orange",
"Mango"]
```

```
```
```

Array Methods:

JavaScript provides several built-in methods for working with arrays. These methods allow you to manipulate, add, remove, and transform array elements. Here are some commonly used array methods:

- 1. push(): Adds one or more elements to the end of an array.**
- 2. pop(): Removes the last element from an array.**
- 3. shift(): Removes the first element from an array.**
- 4. unshift(): Adds one or more elements to the beginning of an array.**
- 5. splice(): Changes the content of an array by adding, removing, or replacing elements.**
- 6. slice(): Extracts a portion of an array into a new array.**
- 7. concat(): Joins two or more arrays and returns a new array.**
- 8. indexOf(): Returns the first index at which a given element can be found in an array.**
- 9. forEach(): Executes a provided function once for each array element.**

- 10. **map():** Creates a new array by applying a function to each element of an array.
- 11. **reduce():** Applies a function to reduce the array to a single value.
- 12. **filter():** Creates a new array with all elements that pass a test.
- 13. **some():** Checks if at least one element in the array passes a test.
- 14. **every():** Checks if all elements in the array pass a test.
- 15. **includes():** Checks if an array contains a specific element.
- 16. **find():** Returns the first element in the array that satisfies a condition.
- 17. **findIndex():** Returns the index of the first element in the array that satisfies a condition.
- 18. **sort():** Sorts the elements of an array in place.
- 19. **reverse():** Reverses the order of the elements in an array.

- 1. **push():** Adds one or more elements to the end of an array.

```
```javascript
var fruits = ["Apple", "Banana", "Orange"];
fruits.push("Mango");
console.log(fruits); // Output: ["Apple", "Banana", "Orange", "Mango"]
```
```

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

```
```javascript
var fruits = ["Apple", "Banana", "Orange"];
```

```
var removedFruit = fruits.pop();
console.log(fruits); // Output: ["Apple", "Banana"]
console.log(removedFruit); // Output: "Orange"
...
```

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

```
``javascript
var fruits = ["Apple", "Banana", "Orange"];
var removedFruit = fruits.shift();
console.log(fruits); // Output: ["Banana", "Orange"]
console.log(removedFruit); // Output: "Apple"
...
```

**4. unshift(): Adds one or more elements to the beginning of an array.**

```
``javascript
var fruits = ["Apple", "Banana", "Orange"];
fruits.unshift("Mango");
console.log(fruits); // Output: ["Mango", "Apple", "Banana",
"Orange"]
...
```

**5. splice(): Changes the content of an array by adding, removing, or replacing elements.**

```
``javascript
var fruits = ["Apple", "Banana", "Orange"];
fruits.splice(1, 1, "Mango", "Grapes");
console.log(fruits); // Output: ["Apple", "Mango", "Grapes",
"Orange"]
...
```

**6. slice():** Extracts a portion of an array into a new array.

```
```javascript
var fruits = ["Apple", "Banana", "Orange", "Mango",
"Grapes"];
var slicedFruits = fruits.slice(1, 4);
console.log(slicedFruits); // Output: ["Banana", "Orange",
"Mango"]
```
```

**7. concat():** Joins two or more arrays and returns a new array.

```
```javascript
var fruits1 = ["Apple", "Banana"];
var fruits2 = ["Orange", "Mango"];
var allFruits = fruits1.concat(fruits2);
console.log(allFruits); // Output: ["Apple", "Banana",
"Orange", "Mango"]
```
```

**8. indexOf():** Returns the first index at which a given element can be found in an array.

```
```javascript
var fruits = ["Apple", "Banana", "Orange"];
var index = fruits.indexOf("Banana");
console.log(index); // Output: 1
```
```

**9. forEach():** Executes a provided function once for each array element.

```
```javascript
var numbers = [1, 2, 3, 4, 5];
numbers.forEach(function (number) {
  console.log(number);
});
// Output:
// 1
// 2
// 3
// 4
// 5
```
```

**10. map():** Creates a new array by applying a function to each element of an array.

```
```javascript
var numbers = [1, 2, 3, 4, 5];
var doubledNumbers = numbers.map(function (number) {
  return number * 2;
});
console.log(doubledNumbers); // Output: [2, 4, 6, 8, 10]
```
```

**Here are some additional examples for the array methods `reduce()`, `filter()`, `some()`, `every()`, and `all()`:**

**1. reduce():** Applies a function to reduce the array to a single value.

```
```javascript
var numbers = [1, 2, 3, 4, 5];
```

```
var sum = numbers.reduce(function (accumulator,
currentNumber) {
  return accumulator + currentNumber;
}, 0);
console.log(sum); // Output: 15
...

```

2. filter(): Creates a new array with all elements that pass a test.

```
```javascript
var numbers = [1, 2, 3, 4, 5];
var evenNumbers = numbers.filter(function (number) {
 return number % 2 === 0;
});
console.log(evenNumbers); // Output: [2, 4]
...

```

**3. some():** Checks if at least one element in the array passes a test.

```
```javascript
var numbers = [1, 2, 3, 4, 5];
var hasEvenNumber = numbers.some(function (number) {
  return number % 2 === 0;
});
console.log(hasEvenNumber); // Output: true
...

```

4. every(): Checks if all elements in the array pass a test.

```
```javascript
var numbers = [1, 2, 3, 4, 5];

```

```
var allEvenNumbers = numbers.every(function (number) {
 return number % 2 === 0;
});
console.log(allEvenNumbers); // Output: false
...
```

**5. includes():** Checks if an array contains a specific element.

```
```javascript  
var fruits = ["Apple", "Banana", "Orange"];  
var hasBanana = fruits.includes("Banana");  
console.log(hasBanana); // Output: true  
...
```

6. find(): Returns the first element in the array that satisfies a condition.

```
```javascript  
var numbers = [1, 2, 3, 4, 5];
var firstEvenNumber = numbers.find(function (number) {
 return number % 2 === 0;
});
console.log(firstEvenNumber); // Output: 2
...
```

**7. findIndex():** Returns the index of the first element in the array that satisfies a condition.

```
```javascript  
var numbers = [1, 2, 3, 4, 5];  
var firstEvenNumberIndex = numbers.findIndex(function  
(number) {  
  return number % 2 === 0;  
});
```



```
});  
console.log(firstEvenNumberIndex); // Output: 1  
...
```

8. sort(): Sorts the elements of an array in place.

```
```javascript  
var fruits = ["Orange", "Apple", "Banana"];
fruits.sort();
console.log(fruits); // Output: ["Apple", "Banana", "Orange"]
...
```

**9. reverse(): Reverses the order of the elements in an array.**

```
```javascript  
var numbers = [1, 2, 3, 4, 5];  
numbers.reverse();  
console.log(numbers); // Output: [5, 4, 3, 2, 1]  
...
```

JavaScript Objects:

Objects in JavaScript are collections of key-value pairs, where each key is a string (or symbol) and each value can be any data type, including arrays and other objects. Objects provide a way to store and organize related data.

Example:

```
```javascript  
// Creating an object
var person = {
```

```
name: "John",
age: 30,
hobbies: ["reading", "coding", "traveling"],
address: {
 street: "123 Main St",
 city: "New York",
 country: "USA"
}

};

// Accessing object properties
console.log(person.name); // Output: "John"
console.log(person.hobbies[0]); // Output: "reading"
console.log(person.address.city); // Output: "New York"

// Modifying object properties
person.age = 35;
console.log(person.age); // Output: 35

// Adding new properties to an object
person.job = "Engineer";
console.log(person.job); // Output: "Engineer"
...

```

## **JSON (JavaScript Object Notation):**

**JSON is a lightweight data interchange format that is widely used for transmitting and storing data. It is based on**

**JavaScript object syntax and provides a way to represent structured data as text.**

**Example:**

```
```javascript
// JSON example
var jsonExample = {
  "name": "John",
  "age": 30,
  "hobbies": ["reading", "coding", "traveling"],
  "address": {
    "street": "123 Main St",
    "city": "New York",
    "country": "USA"
  }
};

// Converting an object to JSON string
var jsonString = JSON.stringify(jsonExample);
console.log(jsonString);
// Output:
{"name":"John","age":30,"hobbies":["reading","coding","traveling"],"address":{"street":"123 Main St","city":"New York","country":"USA"}}

// Converting a JSON string to an object
var jsonObject = JSON.parse(jsonString);
console.log(jsonObject.name); // Output: "John"
```
```

## Hands on Project:

```
// Student Database
var studentDatabase = [];

// Function to add a student
function addStudent(name, email, phone, regId) {
 var student = {
 name: name,
 email: email,
 phone: phone,
 regId: regId
 };
 studentDatabase.push(student);
}

// Adding students
addStudent("John Doe", "john.doe@example.com", "1234567890", "REG001");
addStudent("Jane Smith", "jane.smith@example.com", "9876543210",
"REG002");
addStudent("Mike Johnson", "mike.johnson@example.com", "5555555555",
"REG003");
addStudent("Sarah Williams", "sarah.williams@example.com", "9999999999",
"REG004");
addStudent("David Brown", "david.brown@example.com", "1111111111",
"REG005");
addStudent("Emily Davis", "emily.davis@example.com", "2222222222",
"REG006");
addStudent("Alex Wilson", "alex.wilson@example.com", "3333333333",
"REG007");
addStudent("Olivia Taylor", "olivia.taylor@example.com", "4444444444",
"REG008");
addStudent("Jacob Anderson", "jacob.anderson@example.com", "6666666666",
"REG009");
addStudent("Sophia Martin", "sophia.martin@example.com", "7777777777",
"REG010");

// Sort students based on name
var sortedStudents = studentDatabase.sort(function (a, b) {
```

```

 return a.name.localeCompare(b.name);
 });

console.log("Sorted Students (by name):");
console.log(sortedStudents);
console.log("-----");

// Filter students based on registration ID
var filteredStudents = studentDatabase.filter(function (student) {
 return student.regId.startsWith("REG00");
});

console.log("Filtered Students (by registration ID starting with REG00):");
console.log(filteredStudents);
console.log("-----");

// Map students' names to uppercase
var upperCaseNames = studentDatabase.map(function (student) {
 return student.name.toUpperCase();
});

console.log("Uppercase Names:");
console.log(upperCaseNames);
console.log("-----");

// Iterate over students using forEach()
console.log("Students:");
studentDatabase.forEach(function (student) {
 console.log("Name: " + student.name + ", Email: " + student.email + ", Phone: " + student.phone);
});
console.log("-----");

// Check if at least one student has a phone number starting with '555'
var hasPhoneNumberStartingWith555 = studentDatabase.some(function (student) {
 return student.phone.startsWith("555");
});

```

```
console.log("Does any student have a phone number starting with '555'? " +
hasPhoneNumberStartingWith555);
console.log("-----");

// Check if all students have an email address
var allStudentsHaveEmail = studentDatabase.every(function (student) {
 return student.email !== undefined;
});

console.log("Do all students have an email address? " +
allStudentsHaveEmail);
console.log("-----");

// Find a student by email
var studentByEmail = studentDatabase.find(function (student) {
 return student.email === "sarah.williams@example.com";
});

console.log("Student with email 'sarah.williams@example.com':");
console.log(studentByEmail);
console.log("-----");

// Calculate the total number of students
var totalStudents = studentDatabase.reduce(function (accumulator) {
 return accumulator + 1;
}, 0);

console.log("Total number of students: " + totalStudents);
```