

JAVASCRIPT OBJECTS, ARRAYS, AND LOOPS

FULL STACK SKILLS BOOTCAMP

LOOPS, OBJECTS, AND ARRAYS IN JAVASCRIPT

- **Lesson Overview:**

- In this lesson, we will be introduced to:
 1. Traditional for loops and how they work.
 2. Introduction to objects and arrays.
 3. Adding custom methods to objects.
 4. Using built-in methods for objects and arrays.
 5. Common string methods

UNDERSTANDING THE TRADITIONAL FOR LOOP

- **What is a for loop?**

A for loop allows you to run a block of code a specified number of times.

- **Example:**

```
for (initialization; condition; increment) {  
    // code to be executed  
}
```

INCLUDING JAVASCRIPT IN A WEB PAGE

■ Syntax of a for loop:

Initialization:

A variable is initialized (e.g., `var i = 0`).

Condition: The loop runs as long as the condition is true (e.g., `i < 10`).

Increment: The value is updated after every iteration (e.g., `i++`).

Demo...

```
for (let i = 0; i < 5; i++) {  
  console.log(i);  
}
```

OBJECTS IN JAVASCRIPT

■ Definition:

Objects are collections of key-value pairs.

Objects allow you to store related data and functions (called methods) in one structured format.

Objects can store **primitive values** (strings, numbers, booleans) and functions.

```
const person = {  
  name: 'John',  
  age: 30,  
  isStudent: false  
};  
console.log(person.name); // Output: John
```

ARRAYS IN JAVASCRIPT

- **Definition:**

Arrays are ordered lists of items.

Arrays are used to store multiple values in a single variable.

Arrays are zero-indexed, meaning the first element is at index 0.

```
const fruits = ['apple', 'banana', 'cherry'];  
console.log(fruits[0]); // Output: apple
```

ADDING METHODS TO OBJECTS

■ Definition:

A method is simply a function stored as a property in an object.

Methods allow objects to perform actions based on their properties.

Example: Adding a method to an object

The *this* keyword refers to the current object.

```
const person = {  
  name: "John",  
  age: 30,  
  isStudent: false,  
  greet: function() {  
    return "Hello, my name is " + this.name;  
  }  
};
```

EXAMPLE OF OBJECT USAGE

- **Example:**

Accessing and manipulating an object:

Use dot notation (`person.name`) or bracket notation (`person["name"]`) to access properties.

```
console.log(person.name); // Output: John  
person.age = 31; // Updates the age property  
console.log(person.greet()); // Calls the greet method,
```


COMMON OBJECT METHODS

- **Object.keys():**

Returns an array of the object's property names

```
const keys = Object.keys(person);  
console.log(keys); // ["name", "age", "isStudent", "greet"]
```

- **Object.values():**

Returns an array of the object's property values.

```
const values = Object.values(person);  
console.log(values); // ["John", 30, false, f]
```

- **Object.entries():**

Returns an array of the object's key-value pairs.

```
const entries = Object.entries(person);  
console.log(entries); // [["name", "John"], ["age", 30], ...]
```

COMMON ARRAY METHODS

- **push():** Adds an item to the end of the array..
- **pop():** Removes the last item from the array..
- **shift():** Removes the first item from the array..
- **unshift():** Adds an item to the beginning of the array.
- **map():** Creates a new array by applying a function to each element

```
const numbers = [1, 2, 3];  
const doubled = numbers.map(num => num * 2);  
console.log(doubled); // Output: [2, 4, 6]
```

COMMON STRING METHODS

- **length**: Returns the length of the string
- **toUpperCase()**: Converts the string to uppercase.
- **toLowerCase()**: Converts the string to lowercase
- **indexOf()**: Returns the index of the first occurrence of a substring.
- **slice()**: Extracts a portion of the string.

PRACTICAL EXAMPLE

- **Task:**

Create a function that accepts an object representing a student and returns a message with the student's name and grade.

```
function getStudentInfo(student) {  
  return `Student ${student.name} has a grade of ${student.grade}`;  
}  
  
const student = {  
  name: "Alice",  
  grade: "A"  
};  
  
console.log(getStudentInfo(student)); // Output: Student Alice has a grade of A
```

CONCLUSION

- **Objects** are collections of key-value pairs.
 - **Arrays** are ordered lists of items.
 - **Common methods** help manipulate both objects and arrays efficiently.
 - **Strings** in JavaScript come with many built-in methods for text processing.
- **Objects in JavaScript:**
 - [MDN Web Docs: Working with Objects](#)
 - **Arrays:**
 - [W3Schools: Arrays](#)
 - **Object Methods:**
 - [MDN Web Docs: Object Methods](#)
 - **String Methods:**
 - [W3Schools: JavaScript String Methods](#)

QUESTIONS?