Week 3: JavaScript Fundamentals & Core Logic

Day 1: Introduction to JavaScript and Variable Scoping

The Role of JavaScript in the MERN Stack

JavaScript is the only language that runs across the entire MERN stack: the browser (React), the server (Node/Express), and the database (MongoDB via Shell/Mongoose). In Week 3, we transition from static pages to dynamic, "living" applications.

Variable Declarations: var, let, and const

- **Variable Scope**: JavaScript variables have different accessibility levels depending on where they are declared.
- **Global vs. Local**: Global variables are accessible everywhere, while local variables are confined to the block or function where they are defined.
- **Hoisting and Shadowing**: We explored how JavaScript "hoists" declarations to the top of the script and how "shadowing" occurs when a variable in an inner scope has the same name as one in an outer scope.
- **Immutable Constants**: Using const ensures that the variable reference cannot be reassigned, which is a best practice for maintaining data integrity in React components.

Day 2: Data Types and Operators

Primitive vs. Reference Types

JavaScript handles data in two primary ways:

1. **Primitives**: Strings, Numbers, Booleans, Null, and Undefined. These are stored directly in the "stack" memory.

2. **Reference Types**: Objects, Arrays, and Functions. These are stored in the "heap," and variables hold a reference to the memory location.

The Importance of Immutability

In the context of state management, understanding that objects and arrays are reference types is crucial. Changing a value inside an array without creating a new copy can lead to bugs where the UI fails to update.

Day 3: Control Flow and Decision Making

Conditional Logic

- **If/Else Statements**: The primary way to execute code based on specific conditions.
- **Switch Statements**: Used for multiple branches of logic, such as handling different "Action Types" in a Redux reducer (which we will cover in later weeks).
- **Truthy and Falsy Values**: Understanding that values like 0, "", null, and undefined evaluate to false in a conditional check.

Loops and Iteration

- **For and While Loops**: Standard iteration for repetitive tasks.
- **The .forEach() Method**: A modern way to iterate through arrays, which is the precursor to the .map() function used heavily in React to render lists.

Day 4: Functions and Functional Programming

Defining Logic Units

Functions are the building blocks of JavaScript. We explored:

- **Function Declarations**: The traditional way to define a reusable block of code.

- **Function Expressions**: Assigning a function to a variable.
- **Parameters and Arguments**: Passing data into functions to make them dynamic.
- **Return Values**: How functions output data back to the caller.

The "This" Keyword

One of the most complex parts of JavaScript is the this keyword, which refers to the object that is currently executing the code. Misunderstanding this often leads to errors in event handlers.

Day 5: Objects and Basic Arrays

Storing Complex Data

Objects allow us to group related data into "key-value" pairs. For example, a "Product" object in your **GadgetShop** would contain keys for name, price, and category.

- **Object Literals**: Creating objects using curly braces {}.
- **Dot vs. Bracket Notation**: Different ways to access object properties.
- **Array Basics**: Storing ordered lists of data, which is essential for managing the contact list in your **Phone Book** assignment.