Here’s a logically structured roadmap arranged from beginner to advanced, ensuring a smooth progression through the topics:

Beginner

1. **JavaScript Basics**
   - Primitive and Non-Primitive Types
   - Variables: var, let, const
   - Pass by Value and Pass by Reference
   - Basic Data Types and Operators
   - Wrapper Objects: 0 vs new Number(0)
   - Numbers: 1_000_000, 1e9, Hex, Binary, Octal Numbers
   - Math.trunc, toString(base)
   - Basic Operators (+, -, *, /, %)
2. **Functions**
   - Function Statement / Declaration
   - Function Expression
   - Anonymous Function
   - Named Function Expression
   - IIFE (Immediately Invoked Function Expression)
3. **Scope**
   - Global Scope
   - Local Scope
   - Block Scope
   - Function Scope
   - Scope Chaining
   - Hoisting and Temporal Dead Zone (TDZ)
4. **The this Keyword**
5. **Object Basics**
   - Object Creation: {}
   - Property Access
   - Arrays and Indexing

6. **DOM Manipulation**
   - querySelector, querySelectorAll
   - textContent and innerHTML
   - Event Listeners: addEventListener
7. **Event Propagation**
   - Event Bubbling and Capturing
   - event.stopPropagation()

Intermediate

1. **Intermediate DOM Manipulation**
   - Creating and Manipulating DOM Elements
   - Using this in Event Listeners
2. **Advanced Functions**
   - Higher-Order Functions
   - First-Class Functions
   - Callback Functions and Callback Hell
   - Pure Functions and Functional Programming Basics
   - Decorator Functions
3. **Array Methods**
   - map, filter, reduce, forEach, find, sort
   - Spread and Rest Operators
4. **Objects and Object Methods**
   - Object Shorthand
   - Object Destructuring
   - Object Methods: Object.keys, Object.values, Object.entries, Object.assign
   - Deep Copy vs Shallow Copy
5. **Promises and Asynchronous JavaScript**
   - Callbacks vs Promises
   - Promise Methods: then, catch, finally
   - Promise.all, Promise.race, Promise.any, Promise.allSettled
   - async/await and Error Handling

6. **Closures**
    - Lexical Scope
    - Creating Functions with Memory
7. **Error Handling**
    - Syntax Errors vs Logic Errors
    - try, catch, finally
8. **Event Loop**
    - Call Stack
    - Callback Queue vs Microtask Queue
    - Synchronous vs Asynchronous JavaScript

Advanced

1. **JavaScript Architecture**
    - Execution Context and Memory Allocation
    - Synchronous Single-Threaded Nature
    - Global and Local Execution Context
    - Event Loop, Callbacks, and Promises
2. **Advanced Concepts**
    - Prototypal Inheritance and Prototype Chain
    - Currying Functions
    - Memoization
    - Iterators and Generators
    - Symbols and Iterables
3. **Advanced Objects and Data Structures**
    - Set, Map, WeakSet, WeakMap
    - Object.freeze vs Object.seal
    - Proxy and Reflect APIs
4. **Performance Optimization**
    - Debouncing and Throttling
    - Efficient DOM Updates
5. **Memory Management**

- Mark-and-Sweep Algorithm
- Reachability and Garbage Collection

6. **Advanced Asynchronous Concepts**
   - Async Iteration: for await...of
   - Web Workers and Service Workers
7. **New Features and ES6 Enhancements**
   - Template Literals, Default Parameters
   - Classes and Constructors
   - Modules: import and export
   - Optional Chaining (?.) and Nullish Coalescing (??)
8. **Frameworks and Tools**
   - Transpilers: Babel
   - Module Bundlers: Webpack
   - Polyfills vs Transpilers
9. **Security and Best Practices**
   - use strict
   - Avoiding Anti-Patterns and Common Pitfalls
10. **Additional Advanced Topics**
    - JavaScript Engines (e.g., V8, SpiderMonkey)
    - Just-In-Time Compilation and Abstract Syntax Tree
    - Concurrency Models

This structure moves learners step-by-step, introducing foundational concepts first before diving into deeper and more complex topics. Let me know if you'd like more focus on any specific area!