Here's a comprehensive 14-day JavaScript learning plan going from basics to advanced, with 1.5 hours per day:

Week 1: Fundamentals to Intermediate

Day 1: JavaScript Basics

- Variables (var, let, const)
- Data Types
  - Primitives (string, number, boolean, null, undefined)
  - typeof operator
  - Type coercion
- Basic Operators
- Control Flow (if/else, switch)
  Practice Project: Build a basic calculator

Day 2: Functions & Scope

- Function declarations vs expressions
- Arrow functions
- Parameters & arguments
- Return values
- Global vs Local scope
- Hoisting basics
  Practice Project: Create a temperature converter

Day 3: Arrays & Objects

- Array methods (push, pop, shift, unshift)
- Advanced array methods (map, filter, reduce)
- Object creation and properties
- Object methods
- Destructuring
- Spread/Rest operators
  Practice Project: Build a todo list manager

Day 4: DOM Manipulation

- Selecting elements
- Creating/removing elements
- Event handling

- Event bubbling/capturing
- DOM traversal
- Attributes vs Properties
  Practice Project: Interactive form validation

Day 5: Asynchronous Basics

- setTimeout/setInterval
- Callbacks
- Promise basics
- async/await introduction
- Error handling
  Practice Project: Build a countdown timer

Day 6: ES6+ Features

- Template literals
- Classes
- Modules (import/export)
- Enhanced object literals
- Default parameters
- Optional chaining
  Practice Project: Build a quiz app

Day 7: Intermediate Concepts

- this keyword basics
- call, apply, bind
- Closure introduction
- Array methods deep dive
- String methods
- Regular expressions basics
  Practice Project: Build a search filter

Week 2: Advanced Concepts

Day 8: Advanced Functions

- Closure patterns
- Currying
- Composition

- Higher-order functions
- Memoization
- Generator functions

  Practice Project: Build a function utility library

Day 9: Advanced Objects & Prototypes

- Prototypal inheritance
- Constructor functions
- Object.create()
- Property descriptors
- Getters/setters
- Factory functions

  Practice Project: Build an object-oriented library system

Day 10: Advanced Async Patterns

- Promise chaining
- Promise.all/race/any/allSettled
- Async iterators
- Event loop deep dive
- Microtasks vs Macrotasks

  Practice Project: Build an API data fetcher with retry logic

Day 11: Error Handling & Debugging

- Try/catch patterns
- Custom error classes
- Debugging techniques
- Console methods
- Source maps
- Performance profiling

  Practice Project: Build an error tracking system

Day 12: Design Patterns

- Module pattern
- Singleton
- Observer
- Factory
- Pub/Sub

- MVC/MVVM basics
  Practice Project: Build a state management system

Day 13: Modern APIs & Performance

- Web Workers
- Service Workers
- LocalStorage/SessionStorage
- IndexedDB basics
- Performance optimization
- Memory management
  Practice Project: Build a caching system

Day 14: Testing & Tools

- Unit testing basics
- Jest introduction
- Webpack basics
- NPM ecosystem
- Git workflow
- Code quality tools
  Practice Project: Set up a complete project with testing

Daily Learning Structure (1.5 hours):

- 20 min: Theory and concept review
- 40 min: Coding exercises and practice
- 30 min: Project work

Additional Focus Areas Throughout:

1. Best Practices:
   - Clean code principles
   - DRY (Don't Repeat Yourself)
   - SOLID principles
   - Code organization
   - Documentation
2. Debugging Skills:
   - Browser DevTools
   - Debugging workflows
   - Common pitfalls

    - Performance bottlenecks
3. Essential Tools:
    - VS Code and extensions
    - Chrome DevTools
    - npm basics
    - Git fundamentals
4. Practical Exercises:

- Each day should include:
    - 2-3 coding challenges
    - 1 main project
    - Code review practice

Resources:

1. Documentation:
    - MDN Web Docs
    - JavaScript.info
    - V8 blog
2. Practice Platforms:
    - CodePen
    - JavaScript30
    - LeetCode
    - CodeWars