Intermediate Javascript Programming

Facilitated by Kent State University

Duration:

- 10 Hours
- Synchronous Virtual

Instructor:

- Gregory S. DeLozier, PhD,
- Associate Professor
- Department of Computer Science

COURSE OBJECTIVES

This intermediate JavaScript course is designed to enhance the understanding of modern web development. Participants will delve into advanced JavaScript concepts, including asynchronous programming, closures, and ES6+ features. The course will also cover practical applications such as building dynamic web applications using frameworks like React or Vue.js. Participants will be able to create robust, interactive web applications and tackle more complex programming challenges.

- Understand differences between 'var', 'let', and 'const' and scope
- Embed expressions in strings for cleaner code , Understand the limitations with 'this' binding
- Extract data from arrays and objects
- Import/export syntax
- Understand when to use static vs instance methods in classes
- Organize and modularize code effectively
- Simplity code with concise arrow function syntax
- Understand practical applications of array methods in data manipulation
- Create flexible functions and reusability
- Understand event loop, stack, queue and microtasks
- Chain promises for sequential async operations
- Work with 'error', TypeError', 'SyntaxError', and custom error classes
- Apply best practices to error handling , Use 'console' effectively
- Avoid global variables and maintaining clean code structure
- Setting up ESLint and Prettier for consistent code style
- Identify and refactor inefficient or repetitive code
- Balance new features with readability and browser compatibility
- ES6+ Features and Advanced Syntax
- Functional Programming and Array methods
- Error Handling, Debugging, and Code Quality

COURSE REQUIREMENTS

- An account at GitHub
    - Codespaces permitted

- A computer with a Chrome browser
- Optional laptop software:
    - Visual Studio Code
    - Node.js and related tools
    - Python 3.x
- Setup is covered in class

COURSE OUTLINE

Here's a structured breakdown of 10 one-hour lesson topics that efficiently group related concepts while maintaining a logical progression:

Lesson 1: Advanced JavaScript Fundamentals

- Differences between var, let, and const and scope
- Understanding closures and practical use cases
- Best practices for avoiding global variables
- Using modules with import/export for cleaner code organization

Lesson 2: ES6+ Features & Modern Syntax

- Template literals and embedding expressions in strings
- The limitations of this binding and arrow functions
- Concise arrow function syntax
- When to use static vs. instance methods in classes

Lesson 3: Working with Arrays and Objects Efficiently

- Extracting data using destructuring (arrays & objects)
- Using the spread/rest operators (...) for flexible function parameters
- Practical applications of array methods (map, filter, reduce) for data manipulation

Lesson 4: Functional Programming in JavaScript

- Higher-order functions and functional programming principles
- Array method chaining (map → filter → reduce)
- Writing pure functions and understanding immutability

Lesson 5: Asynchronous JavaScript & the Event Loop

- How the event loop, call stack, and microtasks work
- Understanding callbacks, Promises, and async/await
- Chaining Promises for sequential async operations

Lesson 6: Error Handling & Debugging Best Practices

- Handling runtime errors (Error, TypeError, SyntaxError)
- Creating and using custom error classes
- Best practices for console logging and debugging

Lesson 7: Writing Maintainable & Efficient Code

- Identifying and refactoring inefficient or repetitive code
- Avoiding common pitfalls like callback hell
- Balancing new ES6+ features with readability and browser compatibility

Lesson 8: Code Quality & Tooling

- Setting up ESLint and Prettier for code consistency
- Writing clean, modular, and reusable functions
- Organizing large projects with better code structure

Lesson 9: Introduction to JavaScript Frameworks

- Overview of React vs. Vue.js for dynamic web applications
- Understanding state, props, and component-based architecture
- Writing simple functional components

Lesson 10: Applying Knowledge – A Mini Project

- Building a small interactive web application
- Implementing best practices from previous lessons
- Review, Q&A, and next steps for continued learning