Course duration

- 5 days

Course Benefits

- Learn how to install, configure and use modern web tooling.
- Understand test-driven development.
- Learn to create test suites using Jasmine.
- Write ES6 code and compile it using Babel.
- Understand what React.js is and what problem it solves.
- Explore the basic architecture of a React.js application.
- Gain a deep knowledge of React.js components and JSX.
- Build a working application that uses React.js components.
- Use Redux for maintaining state in a React.js application.
- Learn React.js best practices.

Course Outline

1. React Quickstart with Create-React-App
2. Development Ecosystem
    1. Code Editors and IDEs
    2. Lab 1: Installing and Configuring WebStorm IDE
    3. Node.js
    4. EventEmitter
    5. Node Streams
    6. Node Modules
    7. Lab 2 - Getting Started with Node.js
    8. Git
    9. What is Version Control
    10. History of Git
    11. What is Git?
    12. 3 States of Git
    13. Git Workflow
    14. Lab 3 - Version Control With Git
    15. Command Prompt
    16. Know Your Shell
3. Reproducible Builds
    1. Why Automate Your Build?
    2. Build Requirements
    3. npm
    4. Lab 4 – Initialize npm

   5. node_modules
   6. package.json
   7. npm install
   8. Lab 5 – Using npm as a Build Tool
   9. Lab 6 - Managing External Dependencies
4. Static Code Analysis
   1. Lint tools
   2. Configuring ESLint
   3. ESLint: What Can Be Configured?
   4. ESLint Rules
   5. Lab 7 - Automate Linting
   6. Lab 8 - Configure a Local Web Server
   7. Browser Development Tools
5. Test-Driven Development
   1. Goal of TDD
   2. The TDD Cycle
   3. Red, Green, Refactor
   4. Assertions
   5. JavaScript Testing Frameworks
   6. JS Exception Handling
   7. Jasmine Overview
   8. How Jasmine Works
   9. Test Suites
   10. Specs
   11. Expectations
   12. Matchers
   13. TDD vs. BDD
   14. Lab 9 – Get Started with Jasmine
   15. Lab 10 - TDD Practice
   16. Automated Cross-browser Testing
   17. Karma
   18. Lab 11 - In-browser Testing with Karma
6. Modularity
   1. Why is Modularity Important?
   2. CommonJS
   3. RequireJS
   4. ES6 Modules
   5. Front-end Modules
   6. Manage Modules Manually
   7. Front End Package Management with npm
7. Building and Refactoring
   1. Building the dist directory
   2. webpack
   3. Lab 12: Deploying with Webpack
   4. Lab 13 README Update and Refactoring
8. ES2015 (ES6)
   1. Variable Scoping with const and let

   2. let vs. var
   3. Block-scoped Functions
   4. Arrow Functions
   5. Default Parameter Handling
   6. Rest Parameter
   7. Spread Operator
   8. Template Literals
   9. Enhanced Object Properties
   10. Array Matching
   11. Object Matching
   12. Symbol Primitive
   13. User-defined Iterators
   14. For-Of Operator
   15. Creating and Consuming Generator Functions
   16. Class Definition
   17. Class Declaration
   18. Class Expressions
   19. Class Inheritance
   20. Advanced JavaScript Topics
   21. “use strict”
   22. Understanding this
   23. 4 Rules of this
   24. What is this?
   25. Implicit Binding
   26. Explicit Binding
       1. new Binding
       2. window Binding
       3. Array.map()
       4. Promises
       5. Promises vs. Event Listeners
       6. Why use Promises?
       7. Demo: Callback vs. Promise
       8. Using Promises
       9. Babel
       10. Lab 14: Transpiling with Babel
       11. Lab 15: Converting to ES6

9. The Document Object Model
   1. What is the DOM?
   2. Understanding Nodes
   3. EventTarget
   4. DOM Events
   5. Other Events
   6. Element
   7. Manipulating HTML with the DOM
   8. Manipulating HTML with JQuery
   9. Manipulating HTML with React
10. Introduction to React.js

1. What is React.js
2. Imperative API vs. Declarative API
3. Imperative vs. Declarative Screen updates
4. One-way Data Flow
5. Virtual DOM
6. Virtual DOM vs. HTML DOM
7. State Machines
8. Lab 16, Part 1: Hello, React!
9. Understanding Components
10. React.render()
11. ReactDOM
12. React Development Process
13. Props vs. State
14. Setting Initial State
15. super()
16. Lab 16, Parts 2-3: Your first Component

11. JSX
    1. What is JSX?
    2. Using JSX
    3. JSX is not Exactly HTML
    4. Using React with JSX
    5. Using React without JSX
    6. Expressions in JSX
    7. Precompiled JSX
    8. Lab 17 - HTML to JSX
12. React Components
    1. Creating Components
    2. Pure Functions
    3. Benefits of Pure Functions
    4. F.I.R.S.T
    5. Single Responsibility
    6. Communication Between Components
    7. Props
    8. Ref Callback
    9. Lab 18: Passing Props
    10. Styles in React
    11. Style Components
    12. Lab 19: Style in React
    13. Forms
    14. Forms Have State
    15. Form Events
    16. Controlled Components
    17. Uncontrolled Components
    18. Lab 20: Controlling the Form
    19. Stateless Functional Components
    20. Lab 21: Refactoring the App
    21. Component Life-Cycle Events

    22. Life-Cycle Methods
    23. Mount/Unmount
    24. Mount/Unmount Life-Cycle Methods
    25. Data Life-Cycle Methods
    26. Component Life Cycle
        1. Events
        2. Lab 22: Life Cycle and Events
        3. Composition
        4. Reusable Components
        5. Presentational Components
        6. Container Components
        7. PropTypes
        8. Lab 23: PropTypes
        9. Testing React Components
        10. What to Test in a React Component
        11. Jest
        12. Mocking
        13. Snapshot Testing
        14. TestUtils
        15. Enzyme
        16. Shallow Rendering
        17. Lab 24: Testing React Components
        18. Lab 24.5: Testing with Jest and Enzyme
        19. Lab 25: Multiple Components
        20. React Router
        21. Lab 26: React Router
        22. Lab 27: React Router, Part 2

13. Flux and Redux
    1. Flux
    2. Flux Flow
    3. Flux Action
    4. Flux Dispatcher
    5. Flux Stores
    6. EventEmitter
    7. Redux
    8. Stores & Immutable State Tree
    9. Redux Actions
    10. Reducers
    11. Things You Should Never Do in a Reducer
    12. Reducer Composition
    13. Redux Store
    14. Redux Pros and Cons
    15. Lab 28: Redux Thermometer
    16. Lab 29: Implementing Redux
    17. React AJAX Best Practices
    18. Redux with Ajax
    19. What is Redux Middleware?

   20. What is Middleware Good For?
   21. Thunk
   22. Redux Saga
   23. Using Sagas
   24. Lab 30: SwimCalc App
   25. Lab 31: Redux Middleware
   26. create-react-app
       1. Lab 32: create-react-app and enzyme
       2. Relay and GraphQL

14. Advanced Topics
    1. Server-side React
    2. Using React with Other Libraries
    3. Performance Optimization
    4. Development vs. Production
    5. Perf Object
    6. Perf Object Methods
    7. Optimization Techniques
    8. Using pre-built Components
15. Further Study

Class Materials

Each student will receive a comprehensive set of materials, including course notes and all the class examples.

Class Prerequisites

Experience in the following *is required* for this React class:

- HTML and CSS
- Some JavaScript experience
- Familiarity with intermediate-to-advanced JavaScript programming topics, including objects, functions, closures, callbacks, prototypes and object-oriented JavaScript

Prerequisite Courses

Courses that can help you meet these prerequisites:

- Introduction to JavaScript Training
- Advanced JavaScript Programming
- Introduction to HTML Training - old
- Introduction to CSS Training
- Advanced CSS Training