



React JavaScript Programming

Course ISI-1514 Two Days Instructor-Led, Hands-On

Introduction

This two day class introduces the React JavaScript framework, sometimes referred to as ReactJS or React.JS. React is a popular component-based JavaScript framework used for easy creation of powerful, interactive UIs.

This React class will introduce the benefits of the React JavaScript framework, so course participants can start to develop applications quickly using the framework.

At Course Completion

In this course, students will learn the following:

- Understand the programming model provided by the React framework
- Define React components
- Use the React framework to handle events and stateful data

Prerequisites

Attendees should have some prior understanding of web development, HTML, AJAX, and JavaScript.

Course Materials

The student kit includes a comprehensive workbook.

Course Outline

Module 1: React Overview

- What is React?
- What's in a Name?
- React Component Model
- What React Is Not
- What You Will Not Find in React
- Motivation for Creating React
- A React JavaScript Example
- One-Way Data Flow
- JSX
- A JSX Example
- The Virtual (Mock) DOM
- Only Sub-components that Actually Change are Re-Rendered

Contact ISInc for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>



- React Libraries

Module 2: ES6 in a Nutshell

- What is ES6?
- ES6 Features
- Using ES6
- Transpiling
- Major Syntax Changes
- let and const
- Variable Scope
- Shadowing Variables
- Arrow Functions
- Arrow Functions As Parameters
- Using 'this' Within Arrow Functions
- Template Literals
- Spread Operator
- ES6 Classes
- Declaring Classes
- Declaring Instance Methods
- Accessor Methods
- Static Methods
- Inheritance With Classes

Module 3: Babel Command-Line Interface

- Babel Transpiler
- Usage Options
- Presets and Plug-ins
- Babel CLI Installation
- Babel Configuration
- Running Babel Command-Line
- A Basic ES6 Development Setup with Babel
- Test the Babel Development Setup
- Adding React to the Development Setup
- Create a Minimal React App - Index.html
- Create a Minimal React App - app.js

Module 4: Basic Components and JSX

- What is JSX?
- JSX Transpilation to React Code Example
- Running the Transpiled Code
- Babel
- The Babel Runtime JavaScript Library

Contact ISINC for more information at 916.920.1700 or by visiting our website at <http://www.isinc.com>

- Script Import Skeleton Code
- Playing Around in CodePen
- React Components
- Ways to Create UI Components
- Creating a Functional Component Example
- Component Names Must Be Capitalized
- Creating a React Class-Based Component in ES5
- The render Method
- Creating a UI Component Using ES6 Class Notation
- Using ES6 Classes with React
- Which UI Component Creation Syntax Should I Use?
- Components vs Elements
- Elements Are Immutable
- Properties
- Property Naming Convention
- Properties Default to 'True'
- Spread Attributes (an ES6 Feature)
- Expressions
- Fragments

Module 5: React Component Concepts

- Nesting JSX Elements
- Example of JSX Nesting
- Comments in JSX Code
- JSX Escapes Values
- Working with Lists of Items
- Keys in Lists
- Example List With Key
- Container vs. Presentational Components
- State
- Types of State Data
- State Hierarchy
- Lifting State Up
- Props vs. State
- Pass Down a Function
- Immutability
- Immutability – Why?
- Virtual DOM and State
- Setting state
- Updating Input fields
- Passing Props to Components
- Passing Functions to Components
- Event Handling

- Event Handler Example
- Event Binding - DOs
- Event Binding – Don'ts
- Passing Parameters to Event Handlers
- App Development Workflow – 1/3
- App Development Workflow – 2/3
- App Development Workflow – 3/3

Module 6: React Components with ES6 Classes

- Classes in ES6
- Functional Components
- Extending React.Component
- The render() Method
- state
- props
- defaultProps
- propTypes
- The Component Lifecycle
- componentWillMount()
- componentDidMount()
- setState(newStateValue)

Module 7: React Router

- Routing and Navigation
- react-router
- Creating a react-router based project
- A Basic Routed Component
- Router vs. BrowserRouter
- The Route component
- <Switch>
- Redirect Route
- Navigating with <Link>
- Navigating with <NavLink>
- Route Parameters
- Retrieving Route Parameters
- QueryString Parameters
- Using Router with Redux

Module 8: Building React Apps with Redux

- Redux
- Redux Terminology
- Redux Principles

- Redux: Actions
- Redux Action Types
- Action Creators
- Dispatching Actions
- Data Flow Basics
- Redux Reducers
- Pure Functions
- Reducer Example
- Returning Default State
- Creating a Development Environment with create-react-app
- Using Redux with React
- Initializing the Store
- Immutability
- Benefits of Immutable State
- Mutability of Standard types
- Copying Objects in JavaScript
- Copying Arrays in JavaScript
- One Store - Multiple Reducers
- Combining Reducers
- Components and Redux
- The React-Redux Package
- Wrapping App with Provider
- mapStateToProps
- mapDispatchToProps
- Using Mapped Properties and Methods
- Wrapping Components with Connect
- Configure Store
- Programming Advice - MultiTab Console

Module 9: Unit Testing React with Jest

- Jest Testing Framework
- Features
- Snapshot Testing
- Code Coverage
- Interactive Mode
- Projects created with
- create-react-app
- Installing Jest Manually
- Run Test
- Unit Tests
- Anatomy of a Unit Test
- Common Matchers
- Combining Tests



- Running Tests
- Testing async code with done
- Testing Promise Based Async Code
- Setup and Teardown
- A Simple Snapshot Test
- react-test-renderer
- ReactTestUtils
- Enzyme