

JavaScript and React Development

Course ISI-1528

5 Days

Instructor-led, Hands on

Introduction

Today, JavaScript is used in 89.9% of all websites, including the most heavily-trafficked sites like Google, Facebook, YouTube, and Amazon. That's why every web developer should know how to use JavaScript.

Since the introduction of JavaScript in the mid 90's, several JavaScript libraries and frameworks have emerged to help developers create robust applications. One of the most popular libraries is React.

Prerequisites

To get the most out of this course you'll need to be familiar with HTML and CSS, and how to use these two technologies to create web pages. JavaScript lives inside the web browser, and it's used to interact with a web page loaded into the browser. So being able to write HTML and style it with CSS is an important skill to have before you learn JavaScript. Students should attend Course 1372B, Introduction to HTML5, CSS3 for Web Designers and Developers or have equivalent skills and knowledge.

Course Objectives

Students will learn the following skills:

- Add JavaScript to a web page
- Execute code based on conditions
- Repeat code with loops
- Use object oriented programming with JavaScript
- Dynamically create new HTML elements
- Navigate the document object model
- Handle errors and debug
- React overview
- ES6 in a nutshell
- Babel command-line interface
- Basic components and JSX
- React functional component concepts
- React components with ES6 classes
- React router
- State management
- Using React hooks
- Unit testing

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

Course Outline

Module 1: Introduction to Web Development

- How a web application works
- The components of a JavaScript application
- The HTML Skills you need
- The CSS Skills you need
- How to test a JavaScript application
- How to use Visual Studio Code to develop JavaScript applications

Module 2: Getting Started with JavaScript

- How to include JavaScript in an HTML document
- The JavaScript syntax
- How to work with JavaScript data

Module 3: The Essential JavaScript Statements

- How to code the basic control statement
- How to work with arrays

Module 4: How to work with JavaScript objects, functions and events

- How to use objects to work with data
- How to use functions
- How to handle events

Module 5: How to Script Forms and Controls

- DOM scripting properties and methods
- How to script forms and controls
- How to add new nodes to the DOM

Module 6: How to Test and Debug a JavaScript Application

- An introduction to testing and debugging
- How to debug with Chrome's developer tools
- Other debugging methods

Module 7: How to Work with Numbers, Strings and Dates

- How to work with numbers
- How to work with strings
- How to work with dates and times

Module 8: How to Code Control Statements

- How to code conditional expressions
- How to code the selection structures
- How to code the iteration structures

Module 9: How to Work with Arrays and Web Storage

- How create and use an array
- How to use the methods of an Array object
- Other skills for working with arrays
- How to use web storage

Module 10: How to Create and Use Functions

- Basic skills for working with objects
- What you need to know about JavaScript prototypes
- Advanced skills for working with objects

Module 11: 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
- React Libraries

Module 12: 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 13: 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 14: Basic Components and JSX

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

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

- Components vs Elements
- Elements Are Immutable
- Properties
- Property Naming Convention
- Properties Default to 'True'
- Spread Attributes (an ES6 Feature)
- Expressions
- Fragments

Module 15: React Functional Component Concepts

- Functional Components
- Nesting JSX Elements
- Example of JSX Nesting
- Comments in JSX Code
- Setting CSS Styles Using Classes
- Setting CSS Styles Directly
- 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
- Component Life-cycle
- Life-cycle in Functional Components

Module 16: React Components with ES6 Classes

- Classes in ES6
- Functional Components
- Extending React.Component
- The render() Method
- state
- props
- defaultProps
- propTypes
- Component Lifecycle
- Component Life-cycle: Overview
- Component Life-cycle – Render Phase
- Component Life cycle – Commit Phase
- Component Life-cycle – Unmounting
- constructor()
- example
- componentDidMount()
- example
- setState(newStateValue)

Module 17: 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 18: State Management for React

- React State Basics – Props and State
- Props
- State in Class Based Components
- Managing State with Hooks in Functional Components

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

- The Problem with Props and State
- Redux State Library
- Redux Advantages
- Redux Disadvantages
- Basic Rules for State Management
- Types of State
- Data State
- Communication State
- Control State
- Session State
- Location State
- Location State Side Effects

Module 19: Using React Hooks

- Functional Component Shortcomings
- Hooks Overview
- Functional Component Props
- The useState Hook
- Functional Component using the useState hook
- useState with Multiple Variables
- useState can also be used with Objects
- The Effect Hook
- Effect Hook Example
- Using Effect Hook to Load Data
- Restricting when useEffect is Called
- Hook Rules
- Linter Example
- Custom Hooks
- Custom Hook Function Example
- Using the Custom Hook
- Additional Hooks

Module 20: Unit Testing React with React Testing Library

- React Testing Framework
- Features
- Snapshot Testing
- Code Coverage
- Interactive Mode
- Projects created with
- create-react-app
- Default App Component Test
- Unit Tests

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

- Anatomy of a Unit Test
- Common Matchers
- Combining Tests
- Running Tests
- Testing Promise based async code with 'done'
- Setup and Teardown
- react-testing-library
- A Simple Component Test
- A Simple Snapshot Test
- Running and Updating SnapShot Tests
- Building Component Tests
- Calling Render
- Render Properties
- Simulating Events
- Testing Results
- Using Query Functions
- Text Matching
- Counter Component
- counter-test.js