

# ReactJS Web Application Development

**Duration:** 4 Days (*Face-to-Face & Remote-Live*), or 28 Hours (*On-Demand*)

**Price:** \$2095 (*Face-to-Face & Remote-Live*), or \$895 (*On-Demand*)

**Discounts:** We offer multiple discount options. [Click here](#) for more information.

**Delivery Options:** Attend face-to-face in the classroom, [remote-live](#) or via [on-demand training](#).

## Description

React (a.k.a. ReactJS or React.js) is a popular component-based JavaScript library used for easy creation of powerful, interactive UIs. This React training course will provide an introduction to the benefits of the React JavaScript framework, so course participants can start to develop applications quickly using the framework.

Students will be introduced to the React JavaScript library and covers essentials such as using `Create React App`, defining components, writing and styling JSX elements, passing props, using state and registering event handlers. Students will also learn how to use React Hooks, the Context API, Lifecycle Methods and how to implement global state using the Redux JavaScript library.

Comprehensive hands on exercises are integrated throughout to reinforce learning and develop real competency.

## Prerequisites

Knowledge of HTML and JavaScript equivalent to attending the [Website Development with HTML5, CSS and Bootstrap](#) and [JavaScript Programming](#) courses.

## Course Overview

<

### React Fundamentals

- Overview of React
- Components

### React Development Environment

- Environment Setup
  - NodeJS

- React Tree Reconciliation
- Uni-directional Data Flow
- React and ReactDOM

- VS Code
- VS Code Extensions for React
- Installing React
  - Create React App
- Overview of Babel and Webpack
- Using Create React App

## ES6 Features in React

- ES Modules
- Declaring variables with `let` and `const`
- Arrow functions
- Template Literals
- Rest Parameters and Spread Operator
- Array/Object Destructuring
- ES6 Classes
- Callbacks
- Promises
- `Async/Await`

## Responding to User Events

- Defining Event Handlers
- Issues with Events
- Meaning of this
- `SyntheticEvent`
- Accessing Event Properties
- Event Pooling

## Styling JSX

- How to style JSX Elements
- Applying CSS Classes to JSX
- Defining Style Objects
- Styling with Bootstrap and other CSS Frameworks
- Animating React Components

## Fetching External Data

- Using the Fetch API
- `Async/Await`
- Using Promises
- Communicating with External APIs
- Using `axios` for fetching data

## JSX

- Role of JSX
- Rendering JSX Elements
- Writing JSX Expressions
- Comments in JSX
- React Fragments

## Working with Forms

- Using Interactive Properties
  - `value`
  - `defaultValue`
  - `checked`
  - `selected`
- Controlled inputs
- Uncontrolled inputs
- Capturing Component Updates with `onChange`
- Intercepting Form Submission

## Working with State

- What is State?
- Setting Initial State
- Mutating State with `setState()`
- Problems with State

## Defining Components

- What are Components?
- Presentational vs Container Components
- Rules for Defining Components
- Creating Functional Components

- Limitations of Functional Components
- Creating Class-based Components
- Defining a `constructor()` Method

## Lifecycle Methods

- What are Lifecycle Methods?
- Mounting
  - `constructor`
  - `componentWillMount`
  - `render`
  - `componentDidMount`
- Rendering
  - `componentWillReceiveProps`
  - `shouldComponentUpdate`
  - `componentWillUpdate`
  - `render`
  - `componentDidUpdate`
- Unmounting
  - `componentWillUnmount`
- Dealing with State and Prop Changes
- Error Handling
  - `componentDidCatch`

## Using the Context API

- Why use Context?
- Creating a Context Object
- Defining Providers and Consumers
- Using the Render props pattern

## Passing Data with Props

- What are Props?
- Passing Props to Components
- Receiving Props
- Handling Children
- Validating props with `PropTypes`
  - Data Type Validation
  - Making Properties Required
- "Prop drilling" Explained
- Simplifying Prop drilling with the spread operator
- Communicating with Parent Components

## Using React Router

- Creating Single Page Applications in React
- What is React Router?
- Fundamentals of React Router
- Defining Routers
- Route and Switch components
- React-router objects
  - `Match`
  - `History`
  - `Location`
- Authenticating Routes
- Route Parameters

## Using React Hooks

- What are Hooks?
- Adding state to functional components with `useState`
- `useEffect`
- `useContext`
- `useReducer`

## Deploying a React Application

- Using `npm run build`
- Using `npm run deploy`
- Configuring Client-side Routing
- Deploying to GitHub Pages

Software Skills Training, Inc.  
6 Hemlock Drive  
Chelmsford, MA 01824  
978.250.4983  
[www.software-skills-training.com](http://www.software-skills-training.com)

Copyright © 2021 Software Skills Training, Inc.