

# React Nanodegree Syllabus

*Become a Professional React Developer*



---

## Before You Start

Congratulations on considering the React Nanodegree program! Before you get started, make sure to set aside adequate time on your calendar for focused work, and double-check that you meet the requirements: you should have prior programming experience that includes HTML, CSS, and JavaScript programming.

The React Nanodegree program is comprised of 3 courses and 3 projects. Each project you build will be an opportunity to demonstrate what you've learned in your lessons. Your completed projects become part of a career portfolio that will demonstrate your mastery of React to potential employers.

# Course: React Fundamentals

Mastering React begins with learning your fundamentals, and this can pose a bit of a challenge, because while the modularity of the React ecosystem makes it really powerful for building applications, there is a great deal to learn. So we'll break everything down, and enable you to learn the foundational parts of the React ecosystem that are necessary to build production-ready apps.

As this is a project-based course, you're going to start building right away. This gives you an opportunity to get your hands dirty with React, and start mastering the skills you'll need. Plus, every project you build is reviewed by an expert Project Reviewer, and their detailed feedback will be instrumental in helping you to advance.

Lesson Title	Learning Outcomes
Why React	<ul style="list-style-type: none"><li>→ Identify why React was built</li><li>→ Outline concepts that are the building blocks of React</li><li>→ Recognize that React is just JavaScript</li></ul>
Rendering UI with React	<ul style="list-style-type: none"><li>→ Use <i>create-react-app</i> to create a new React application</li><li>→ Create <i>Class components</i></li><li>→ Use <i>JSX</i> to describe UI</li></ul>
State Management	<ul style="list-style-type: none"><li>→ Manage state in applications</li><li>→ Use <i>props</i> to pass data into a component</li><li>→ Create <i>functional components</i></li><li>→ Add <i>state</i> to components</li><li>→ Use the <i>this</i> keyword</li><li>→ Update state with <i>setState</i></li><li>→ Use <i>PropTypes</i></li><li>→ Use <i>controlled components</i></li></ul>
Render UI with External Data	<ul style="list-style-type: none"><li>→ Conceptualize the <i>lifecycle</i> of a component</li><li>→ Use React's <i>componentDidMount</i> lifecycle hook</li></ul>
Manage App Location with React Router	<ul style="list-style-type: none"><li>→ Use React Router to add different routes to applications</li><li>→ Use state to dynamically render a different "page"</li><li>→ Use React Router's <i>Route</i> component</li><li>→ Use React Router's <i>Link</i> component</li></ul>

## Project: MyReads: A Book Lending App

In this project, you will create a React application from scratch and utilize React components to manage the user interface. You'll create a virtual bookcase to store your books and track what you're reading. Using the provided Books API, you'll search for books and add them to a bookshelf as a React component. Finally, you'll use React's *setState* to build the functionality to move books from one shelf to another.

## Course: React & Redux

Classroom opens August 2017

Redux excels at state management, and in this course, you'll learn how Redux and React work together to make your application's state bulletproof.

As with the previous course, this is hand-on curriculum, and building projects are what it's all about. Here, you'll leverage React with Redux to build a "Would you Rather" application.

Lesson Title	Learning Outcomes
Why Use Redux	<ul style="list-style-type: none"><li>→ Build intuition for when to use Redux</li><li>→ Identify when to use component state vs Redux state</li></ul>
Redux With React	<ul style="list-style-type: none"><li>→ Create Reducers</li><li>→ Create Actions and Action Creators</li><li>→ Use createStore to create a Redux store</li><li>→ Tie Redux into a React app with react-redux</li><li>→ Use React DevTools to debug Redux state</li></ul>
Architect A Redux Store	<ul style="list-style-type: none"><li>→ Identify the strengths and weaknesses of normalized data</li><li>→ Architect a redux store with combineReducers</li></ul>

### Project: Readable

Leverage the strengths of Redux to build a "Readable" application where users can post text content and comment on each other's posts. You'll build this dynamic application from scratch while combining the state management features of Redux with the declarative component model from React. When complete, you'll be able to submit your own posts, comment on existing posts, and edit and delete posts and comments.

## Course: React Native

Classroom opens September 2017

In this course, you'll learn how to develop React applications that run on both iOS and Android devices. We'll explore everything from Authentication and the Facebook SDK, to Firebase and Flexbox. You'll learn about animations and styling in React Native, and even how to submit to Google Play and App Store.

### Project: Fitness Tracker

In this project you'll use React Native features to build a health and fitness tracker application for both iOS and Android. You'll leverage native OS system access features to track step count using the mobile OS access to the phone's hardware that allows step tracking.