1 Introduction

Welcome to the very start of your React Native journey! If you're looking for environment setup instructions, they've moved to their own section. Continue reading for an introduction to the documentation, Native Components, React, and more!

2 Core Components

React Native is an open source framework for building Android and iOS applications using React and the app platform’s native capabiles. With React Native, you use JavaScript to access your platform’s APIs as well as to describe the appearance and behavior of your UI using React components: bundles of reusable, nestable code. You can learn more about React in the next section. But first, let’s cover how components work in React Native.

3 React Fundamentals

React Native runs on React, a popular open source library for building user interfaces with JavaScript. To make the most of React Native, it helps to understand React itself. This section can get you started or can serve as a refresher course.

We’re going to cover the core concepts behind React:

+ components. The rest of this introduction to React uses cats in its examples: friendly, approachable creatures that need names and a cafe to work in.

+ JSX. React and React Native use JSX, a syntax that lets you write elements inside JavaScript like so: <Text>Hello, I am your cat!</Text>. The React docs have a comprehensive guide to JSX you can refer to learn even more. Because JSX is JavaScript, you can use variables inside it. Here you are declaring a name for the cat, name, and embedding it with curly braces inside <Text>.

+ props. Props is short for “properties”.

+ state. While you can think of props as arguments you use to configure how components render, state is like a component’s personal data storage. State is useful for handling data that changes over time or that comes from user interaction. State gives your components memory!

4 Handling Text Input

TextInput is a Core Component that allows the user to enter text. It has an onChangeText prop that takes a function to be called every time the text changed, and an onSubmitEditing prop that takes a function to be called when the text is submitted.

For example, let's say that as the user types, you're translating their words into a different language. In this new language, every single word is written the same way: 🍕. So the sentence "Hello there Bob" would be translated as "🍕 🍕".