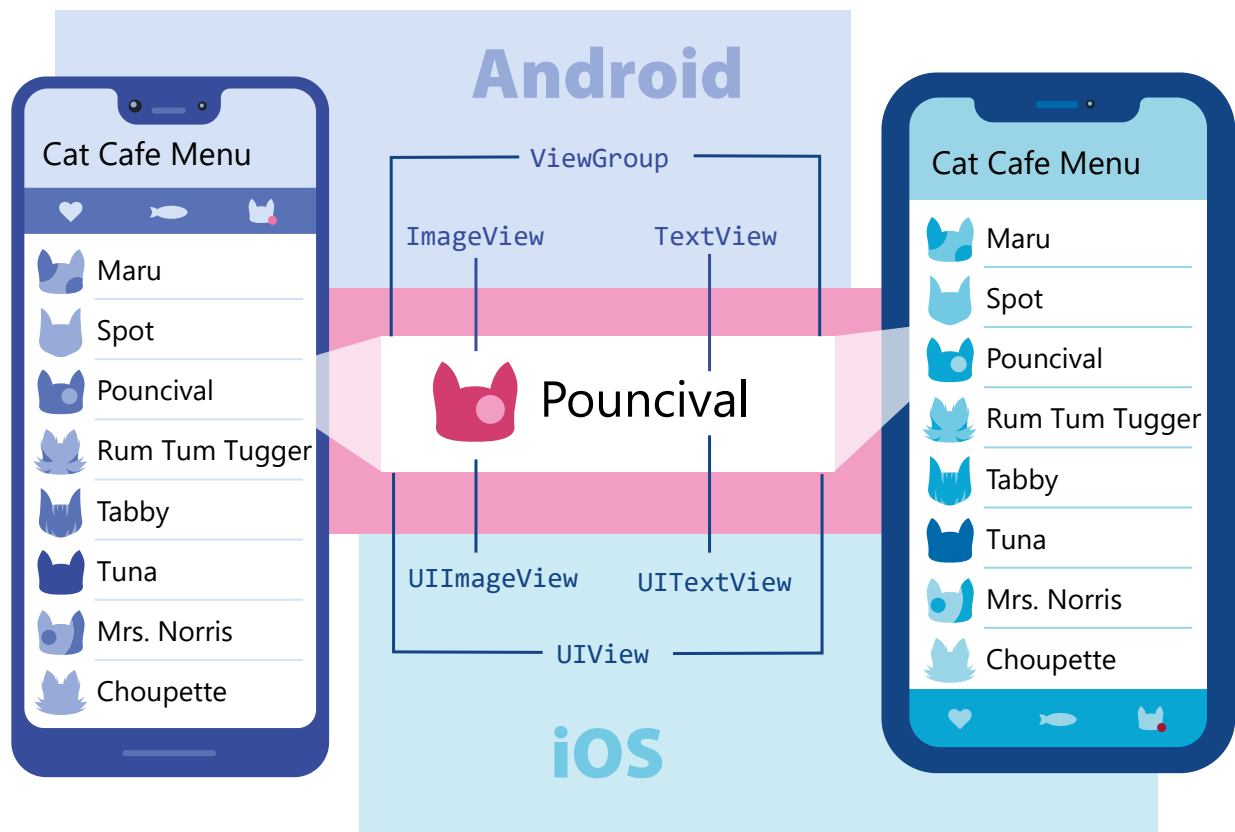


Core Components and Native Components

React Native is an open source framework for building Android and iOS applications using React and the app platform's native capabilities. 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.

Views and mobile development

In Android and iOS development, a **view** is the basic building block of UI: a small rectangular element on the screen which can be used to display text, images, or respond to user input. Even the smallest visual elements of an app, like a line of text or a button, are kinds of views. Some kinds of views can contain other views. It's views all the way down!



Just a sampling of the many views used in Android and iOS apps.

Native Components

In Android development, you write views in Kotlin or Java; in iOS development, you use Swift or Objective-C. With React Native, you can invoke these views with JavaScript using React components. At runtime, React Native creates the corresponding Android and iOS views for those components. Because React Native components are backed by the same views as Android and iOS, React Native apps look, feel, and perform like any other apps. We call these platform-backed components **Native Components**.

React Native comes with a set of essential, ready-to-use Native Components you can use to start building your app today. These are React Native's **Core Components**.

React Native also lets you build your own Native Components for [Android](#) and [iOS](#) to suit your app's unique needs. We also have a thriving ecosystem of these **community-contributed components**. Check out [Native Directory](#) to find what the community has been creating.

Core Components

React Native has many Core Components for everything from controls to activity indicators. You can find them all [documented in the API section](#). You will mostly work with the following Core Components:

REACT NATIVE UI COMPONENT	ANDROID VIEW	IOS VIEW	WEB ANALOG	DESCRIPTION
<code><View></code>	<code><ViewGroup></code>	<code><UIView></code>	A non-scrolling <code><div></code>	A container that supports layout with flexbox, style, some touch handling, and accessibility controls
<code><Text></code>	<code><TextView></code>	<code><UITextView></code>	<code><p></code>	Displays, styles, and nests strings of text and

REACT NATIVE UI COMPONENT	ANDROID VIEW	IOS VIEW	WEB ANALOG	DESCRIPTION
				even handles touch events
<code><Image></code>	<code><ImageView></code>	<code><UIImageView></code>	<code></code>	Displays different types of images
<code><ScrollView></code>	<code><ScrollView></code>	<code><UIScrollView></code>	<code><div></code>	A generic scrolling container that can contain multiple components and views
<code><TextInput></code>	<code><EditText></code>	<code><UITextField></code>	<code><input type="text"></code>	Allows the user to enter text

In the next section, you will start combining these Core Components to learn about how React works. Have a play with them here now!

Hello World
Expo

```

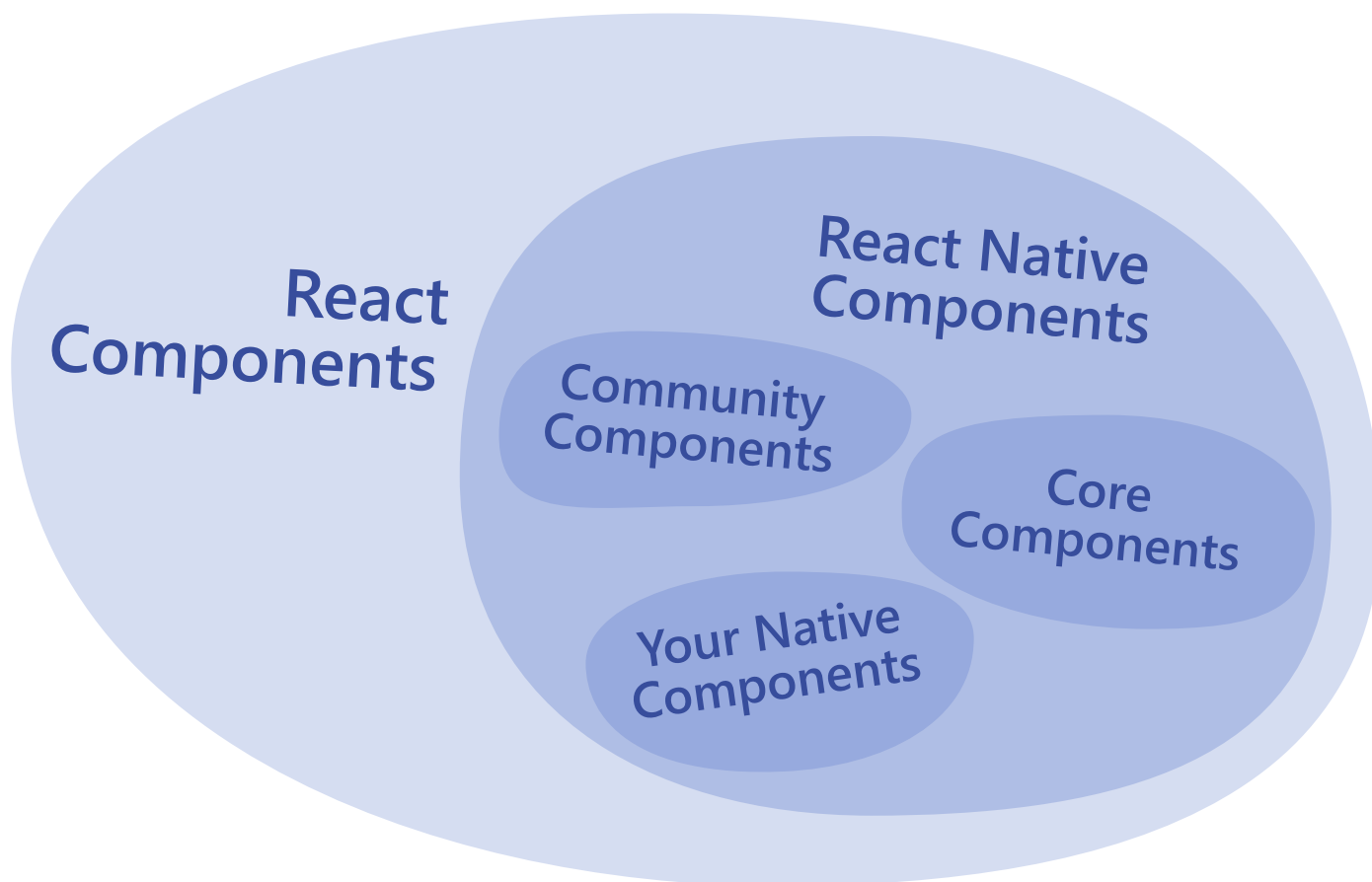
import React from 'react';
import {View, Text, Image, ScrollView, TextInput} from 'react-native';

const App = () => {
  return (
    <ScrollView>
      <Text>Some text</Text>
      <View>
        <Text>Some more text</Text>
        <Image
          source={{
            uri: 'https://reactnative.dev/docs/assets/p_cat2.png',
          }}
          style={{width: 200, height: 200}}
        />
      </View>
      <TextInput
        style={{
          height: 40,
          borderColor: 'gray',
          borderWidth: 1,
        }}
        defaultValue="You can type in me"
      />
    </ScrollView>
  );
}


```

Preview
My Device
iOS
Android
Web

Because React Native uses the same API structure as React components, you'll need to understand React component APIs to get started. The [next section](#) makes for a quick introduction or refresher on the topic. However, if you're already familiar with React, feel free to [skip ahead](#).



Is this page useful?  

 Edit this page

Last updated on **Jun 21, 2023**