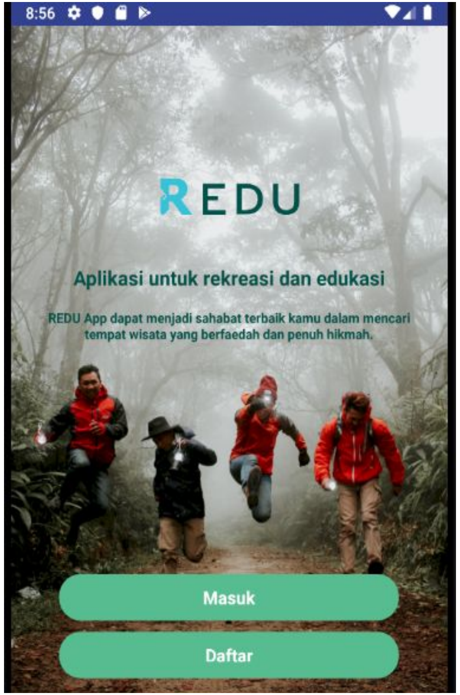
React Native Basic

Isumi
April 2020

GOALS

1. Intro
2. Installing (RN CLI)
3. Components
4. Debugging
5. Styling



Aplikasi untuk rekreasi dan edukasi

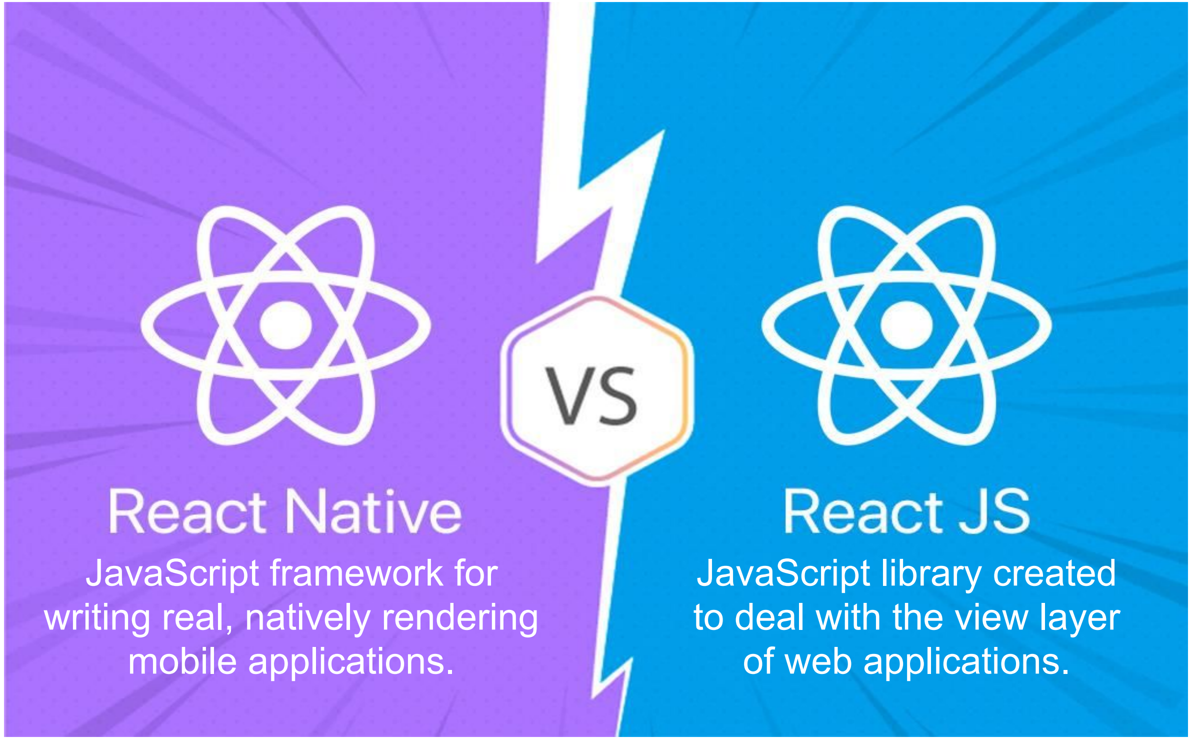
REDU App dapat menjadi sahabat terbaik kamu dalam mencari tempat wisata yang berfaedah dan penuh hikmah.

1. Intro

"The biggest mistake we made as a company was betting too much on HTML as opposed to native" (Mark Zuckerberg)

1. Intro

Why?

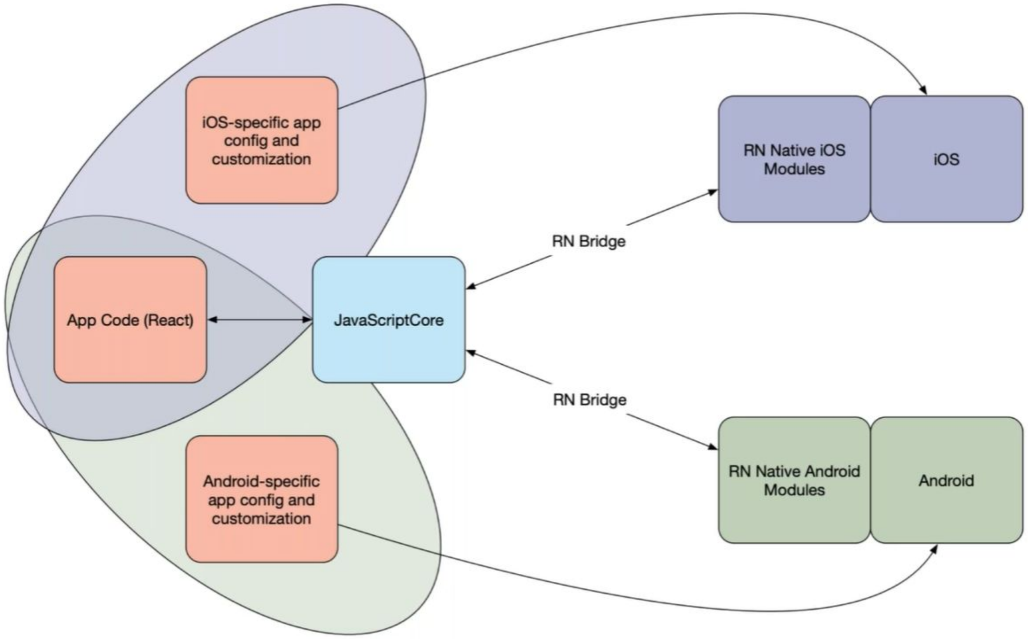


JavaScript framework for writing real, natively rendering mobile applications.

JavaScript library created to deal with the view layer of web applications.

1. Intro

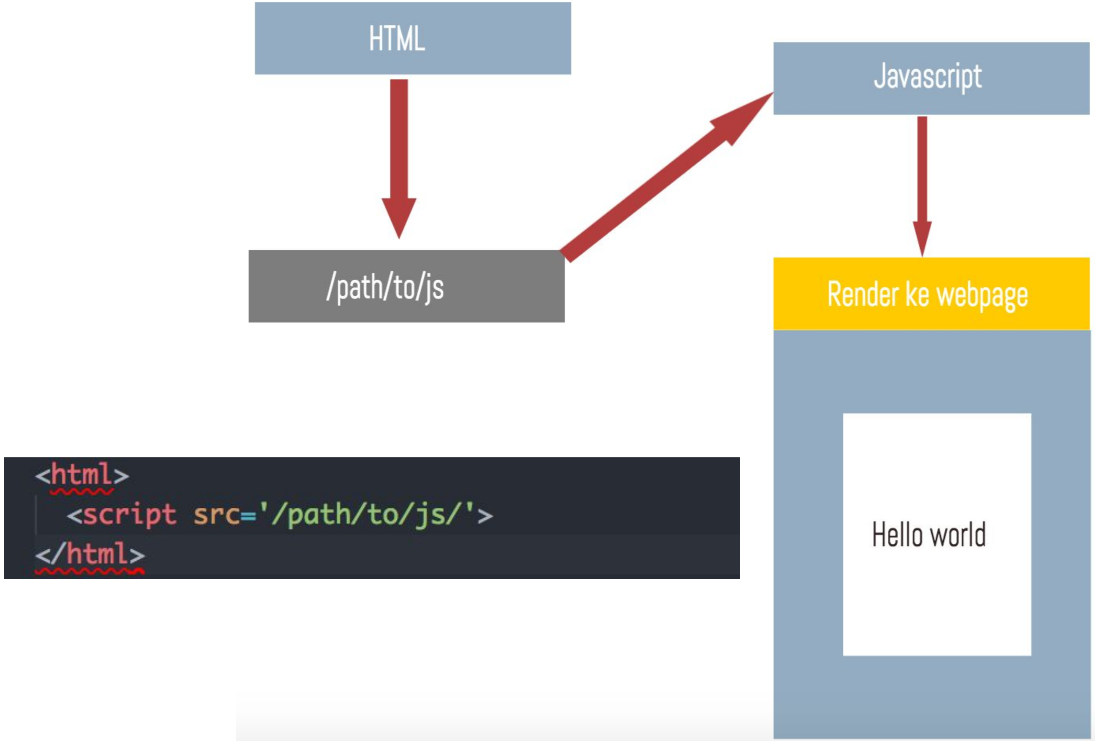
How does RN work?


iOS-specific app config and customization
App Code (React)
JavaScriptCore
Android-specific app config and customization
RN Bridge
RN Bridge
RN Native iOS Modules
iOS
RN Native Android Modules
Android

References:
https://medium.com/under10/react-native-cara-kerjanya-gimana-sih-8e4ab3542cff
https://www.headspin.io/blog/appium/testing-react-native-apps-with-appium/

1. Intro

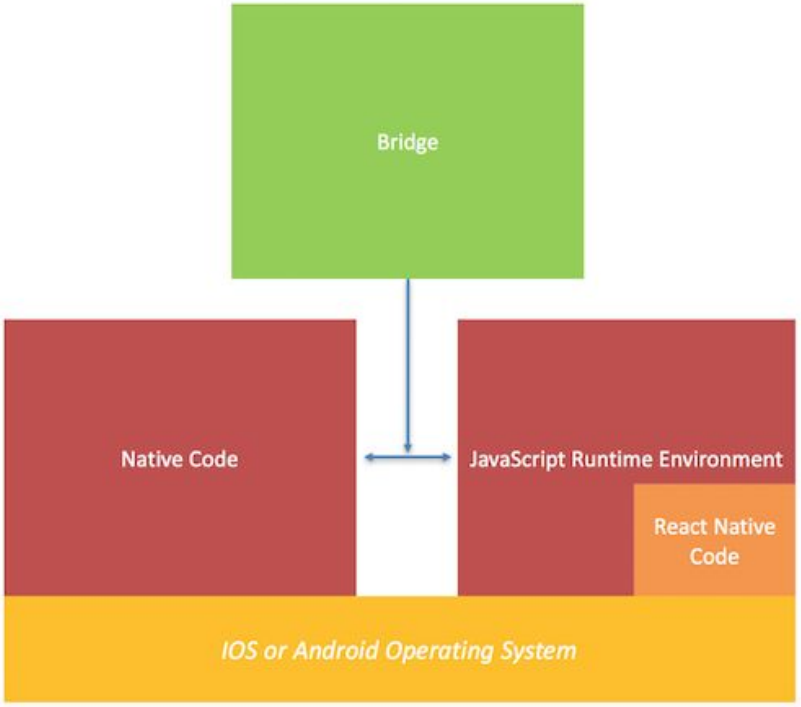
First: JS modern web app work

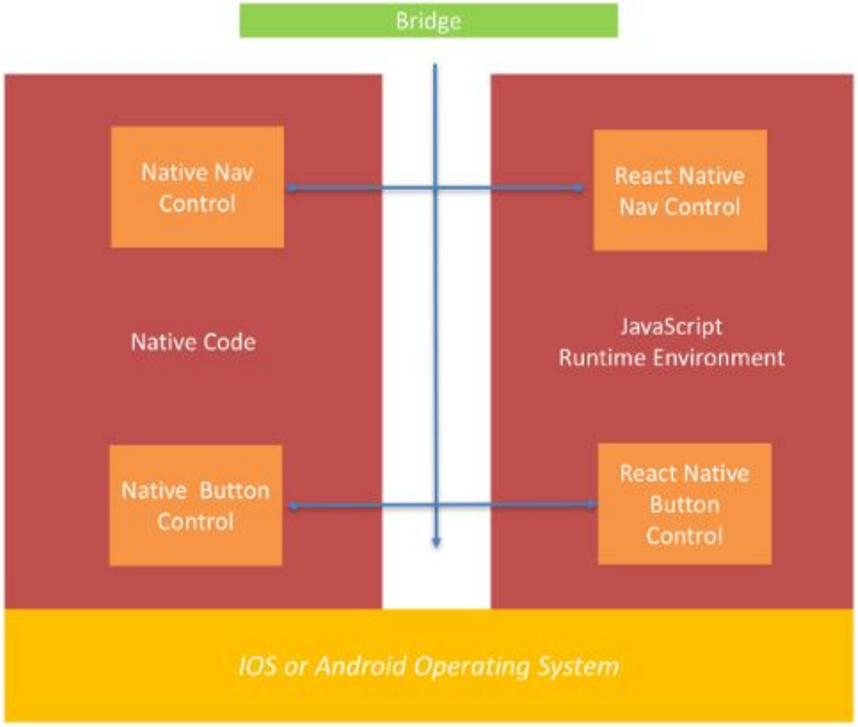


```
<html>
  <script src='/path/to/js/'>
</html>
```

1. Intro

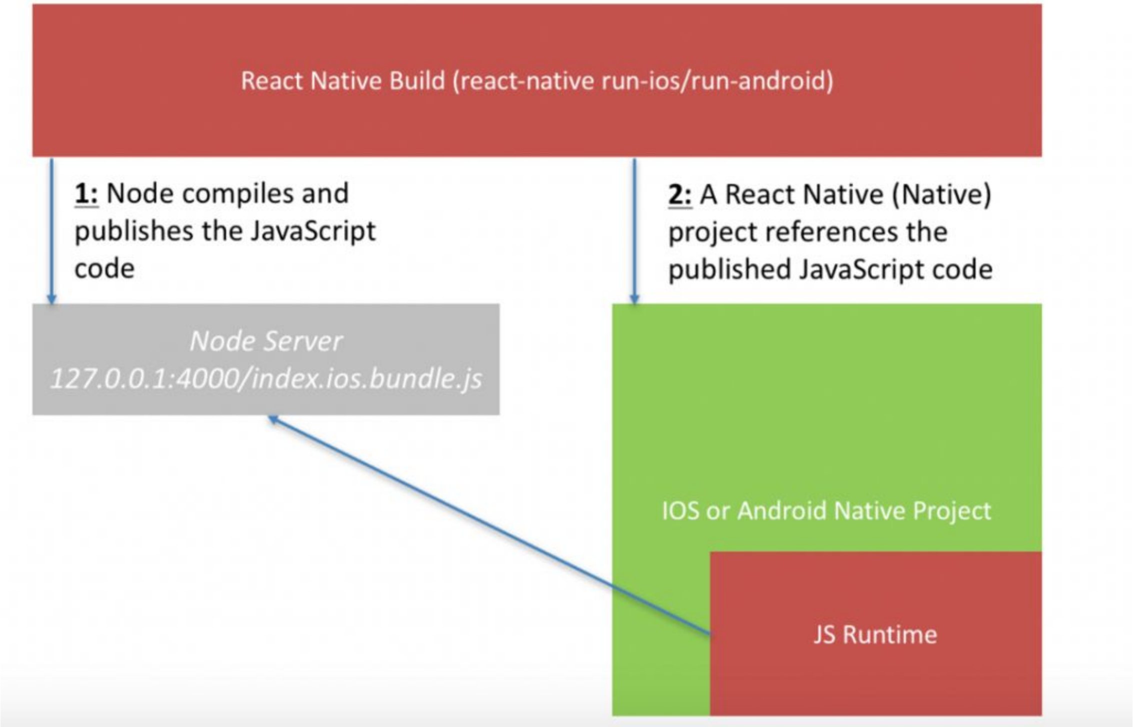
Second: JS executed


Bridge
Native Code
JavaScript Runtime Environment
React Native Code
IOS or Android Operating System


Bridge
Native Nav Control
React Native Nav Control
Native Code
JavaScript Runtime Environment
Native Button Control
React Native Button Control
IOS or Android Operating System

1. Intro

Third: The RN Build Process Architecture



React Native Build (react-native run-ios/run-android)

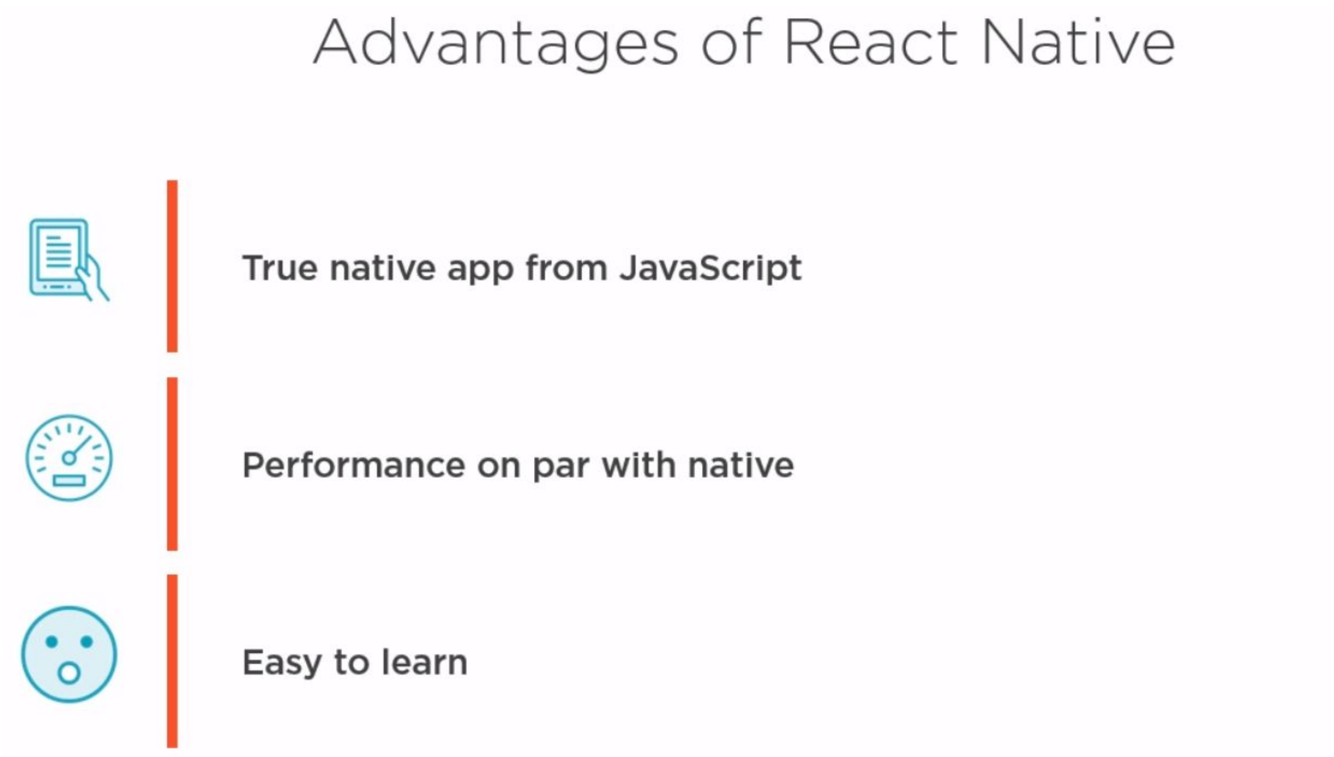
1: Node compiles and publishes the JavaScript code

2: A React Native (Native) project references the published JavaScript code

Node Server
127.0.0.1:4000/index.ios.bundle.js

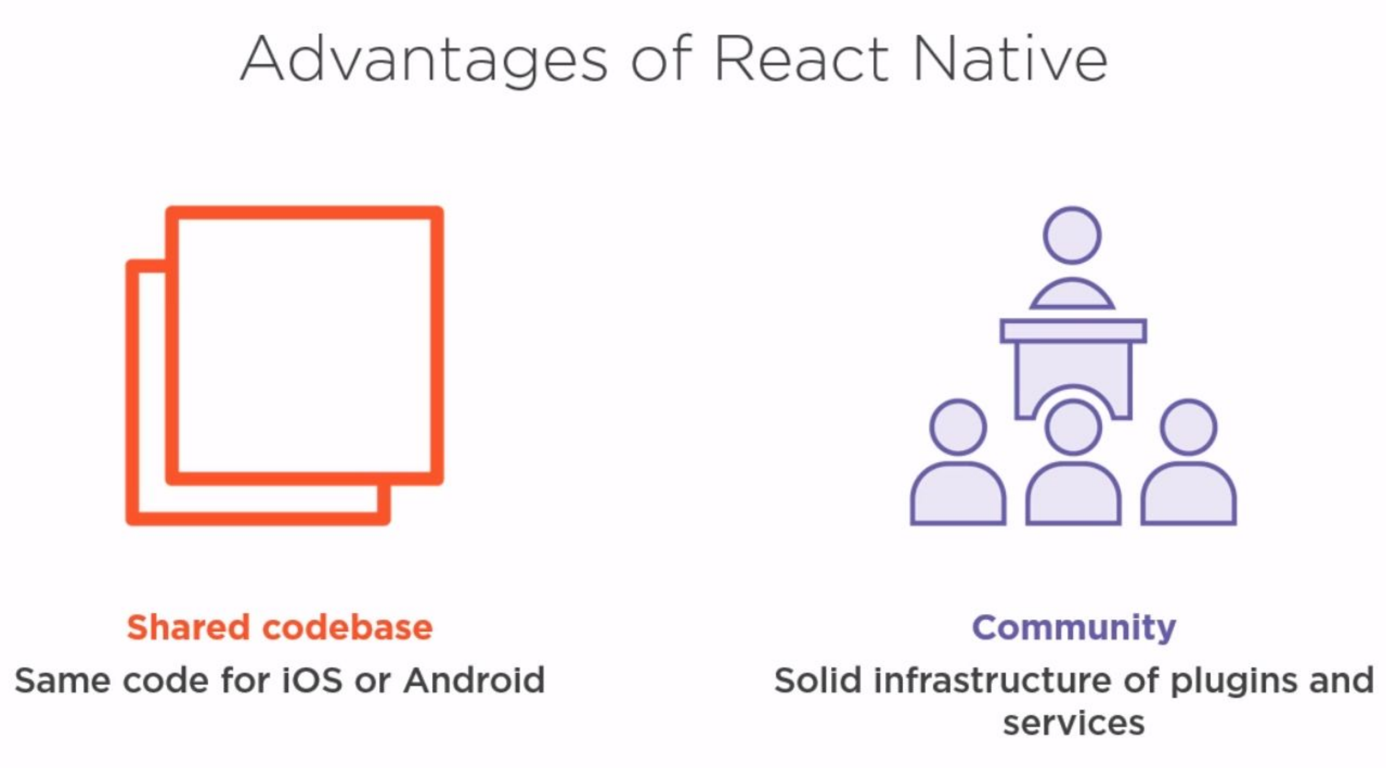
IOS or Android Native Project

JS Runtime



Advantages of React Native

- True native app from JavaScript
- Performance on par with native
- Easy to learn



Advantages of React Native

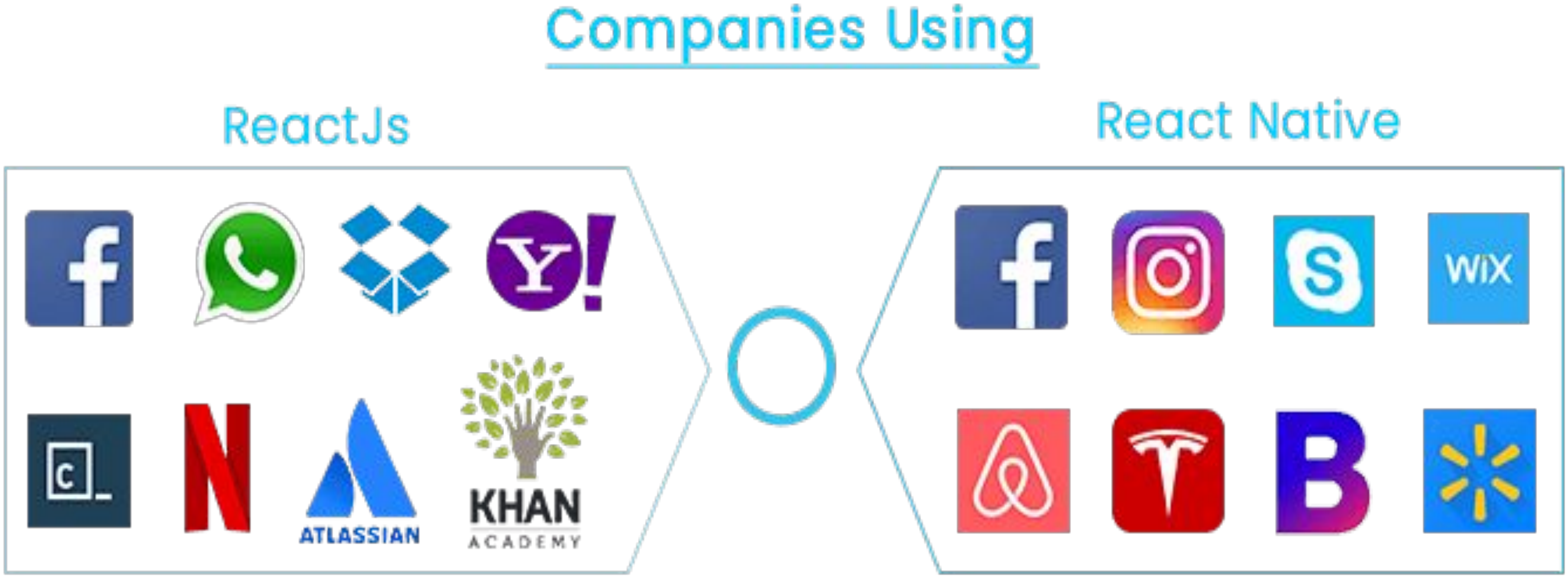
Shared codebase

Same code for iOS or Android

Community

Solid infrastructure of plugins and services

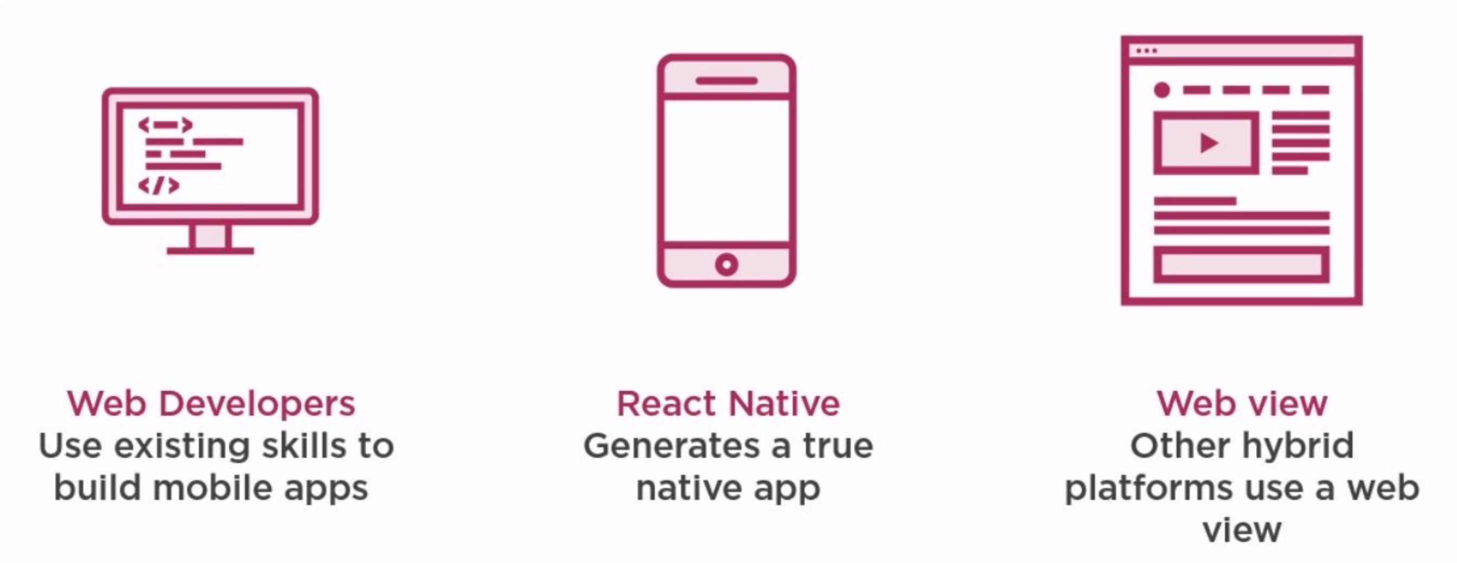
1. Intro



References:
https://www.youtube.com/watch?v=S4oggFrlFIw

1. Intro

Why is RN different?



Web Developers
Use existing skills to build mobile apps

React Native
Generates a true native app

Web view
Other hybrid platforms use a web view

References:
https://www.netguru.com/blog/when-react-native-is-not-a-good-choice-for-a-mobile-application-development

2. Installing (RN CLI)

- Requirements:
  - Android: NodeJS, JDK, Android Studio/Emulator/AVD/Physical device
  - iOS: NodeJS, JDK, XCode, Physical device
- Installing dependencies:
  - Android:

    Install Android SDK, Configure ANDROID_HOME environment
  - iOS:

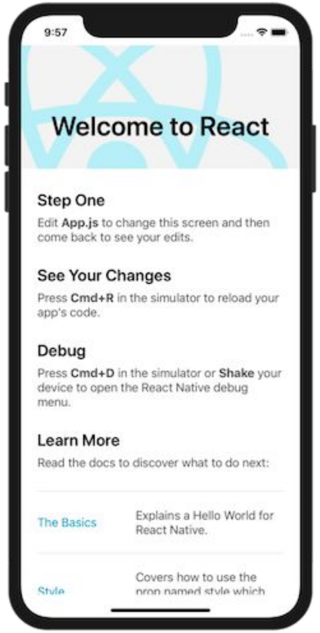
```
brew install node, tap AdoptOpenJDK/openjdk, cask install adoptopenjdk8
```

References:
https://reactnative.dev/docs/environment-setup

- `npm install -g react-native-cli`
- `react-native init AwesomeProject`

android

```
cd AwesomeProject
react-native run-android
```



9:57

Welcome to React

Step One

Edit **App.js** to change this screen and then come back to see your edits.

See Your Changes

Press **Cmd+R** in the simulator to reload your app's code.

Debug

Press **Cmd+D** in the simulator or **Shake** your device to open the React Native debug menu.

Learn More

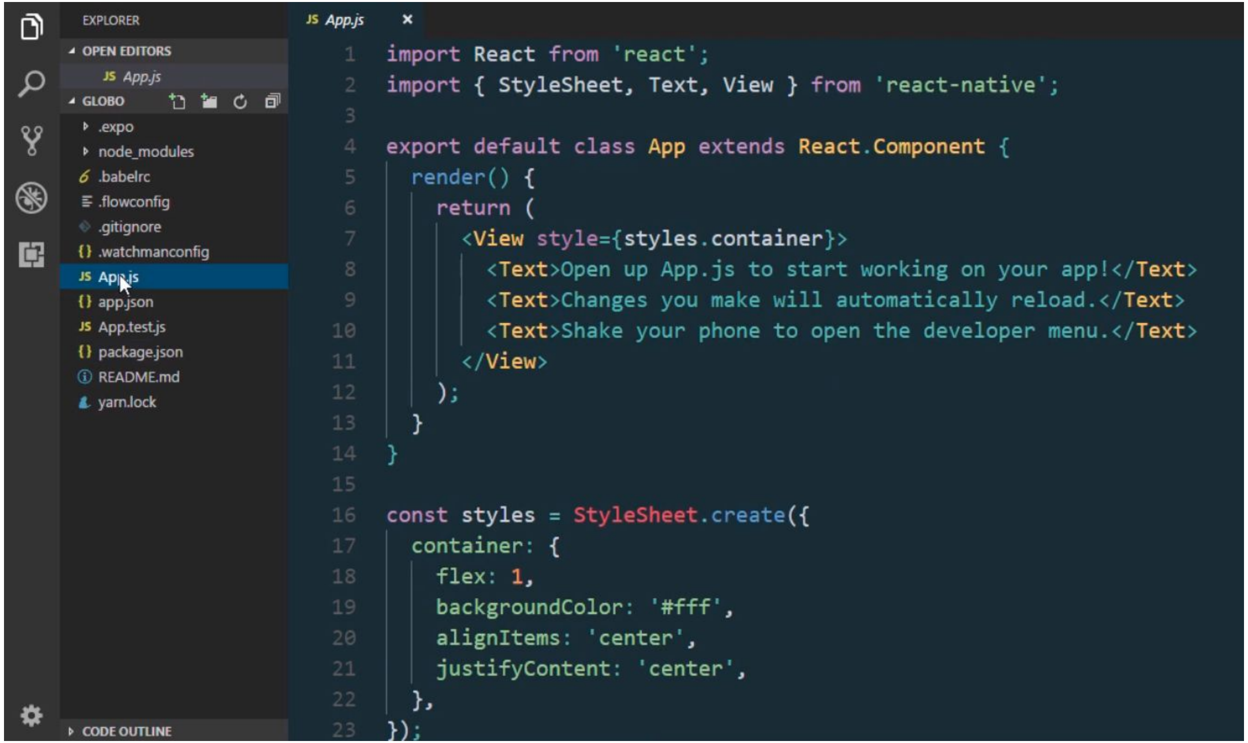
Read the docs to discover what to do next:

The Basics — Explains a Hello World for React Native.

Style — Covers how to use the prop named style which

```
react-native start
```

```
cd AwesomeProject
react-native run-ios
```



App.js

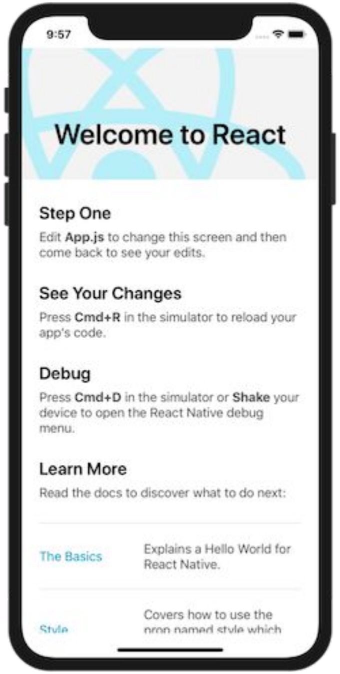
```
import React from 'react';
import { StyleSheet, Text, View } from 'react-native';

export default class App extends React.Component {
  render() {
    return (
      <View style={styles.container}>
        <Text>Open up App.js to start working on your app!</Text>
        <Text>Changes you make will automatically reload.</Text>
        <Text>Shake your phone to open the developer menu.</Text>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

3. Components

React uses components to build apps

React Native includes many components ready for use

React Native components translate to native features



9:57

Welcome to React

Step One

Edit **App.js** to change this screen and then come back to see your edits.

See Your Changes

Press **Cmd+R** in the simulator to reload your app's code.

Debug

Press **Cmd+D** in the simulator or **Shake** your device to open the React Native debug menu.

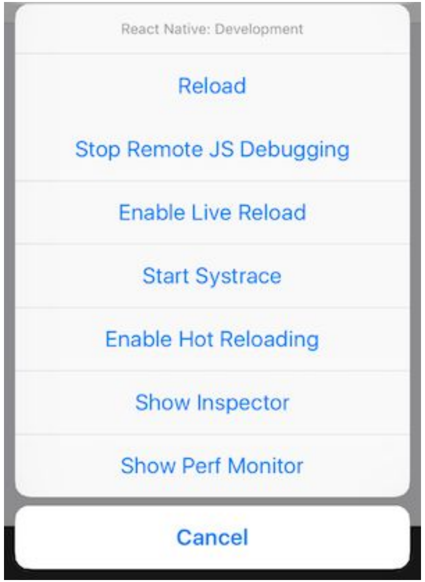
Learn More

Read the docs to discover what to do next:

The Basics	Explains a Hello World for React Native.
Style	Covers how to use the prop named style which

4. Debugging

Use the: ⌘D / ⌘M / Ctrl+M



React Native: Development

Reload

Stop Remote JS Debugging

Enable Live Reload

Start Systrace

Enable Hot Reloading

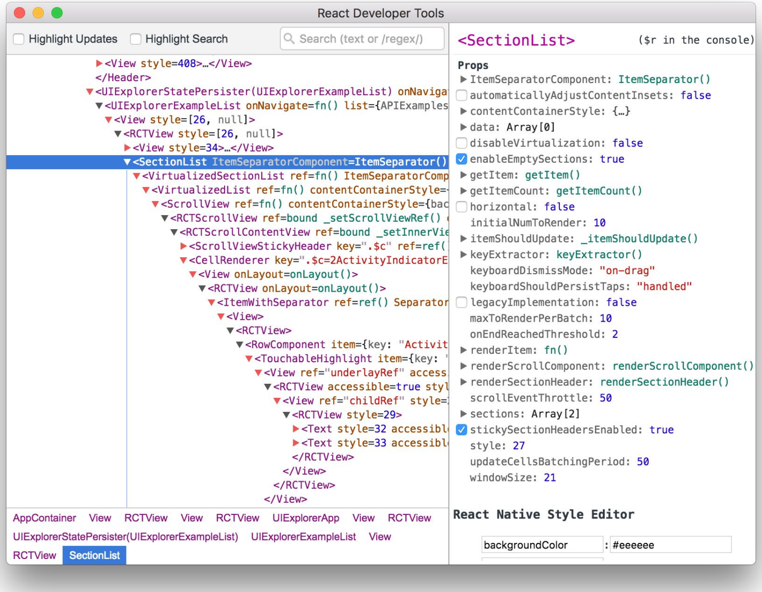
Show Inspector

Show Perf Monitor

Cancel

```
npm install -g react-devtools
```

```
react-devtools
```



React Developer Tools

□ Highlight Updates □ Highlight Search Search (text or /regex/)

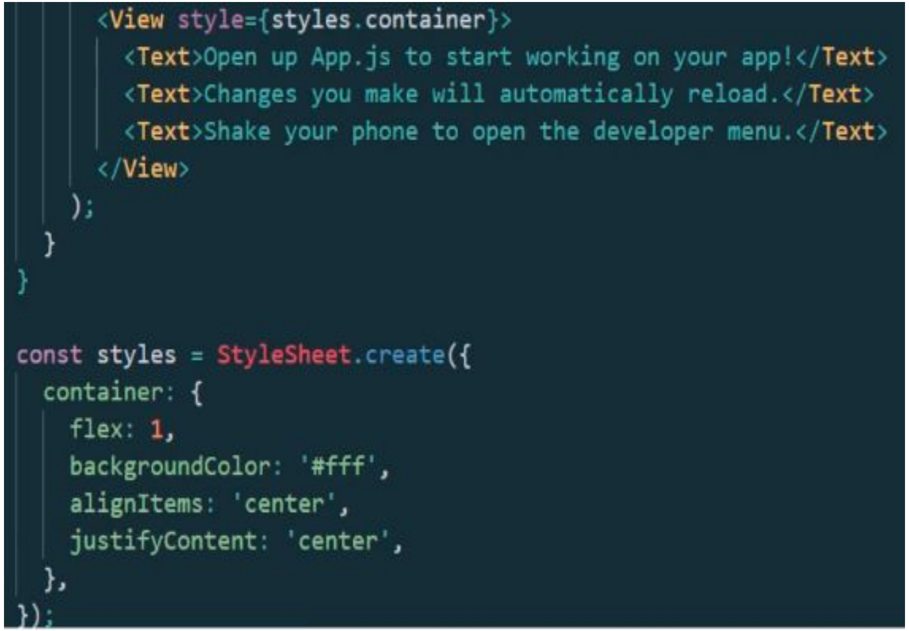
```
            ▶<View style=408>…</View>
            </Header>
           ▼<UIExplorerStatePersister(UIExplorerExampleList) onNaviga
            ▼<UIExplorerExampleList onNavigate=fn() list={APIExamples
             ▼<View style=[26, null]>
              ▼<RCTView style=[26, null]>
               ▶<View style=34>…</View>
               ▼<SectionList ItemSeparatorComponent=ItemSeparator()
                ▼<VirtualizedSectionList ref=fn() ItemSeparatorComp
                 ▼<VirtualizedList ref=fn() contentContainerStyle=
                  ▼<ScrollView ref=fn() contentContainerStyle={bac
                   ▼<RCTScrollView ref=bound _setScrollViewRef()
                    ▼<RCTScrollContentView ref=bound _setInnerVie
                     ▶<ScrollViewStickyHeader key=".$c" ref=ref(
                     ▼<CellRenderer key=".$c=2ActivityIndicatorE
                      ▼<View onLayout=onLayout()>
                       ▼<RCTView onLayout=onLayout()>
                        ▼<ItemWithSeparator ref=ref() Separator
                         ▼<View>
                          ▼<RCTView>
                           ▼<RowComponent item={key: "Activit
                            ▼<TouchableHighlight item={key: '
                             ▼<View ref="underlayRef" access
                              ▼<RCTView accessible=true styl
                               ▼<View ref="childRef" style=
                                ▼<RCTView style=29>
                                 ▶<Text style=32 accessibl
                                 ▶<Text style=33 accessibl
                                 </RCTView>
                                </View>
                               </RCTView>
                              </View>
```

AppContainer View RCTView View RCTView UIExplorerApp View RCTView UIExplorerStatePersister(UIExplorerExampleList) UIExplorerExampleList View RCTView SectionList

```
<SectionList>                ($r in the console)
Props
▶ ItemSeparatorComponent: ItemSeparator()
□ automaticallyAdjustContentInsets: false
▶ contentContainerStyle: {…}
▶ data: Array[0]
□ disableVirtualization: false
☑ enableEmptySections: true
▶ getItem: getItem()
▶ getItemCount: getItemCount()
□ horizontal: false
  initialNumToRender: 10
▶ itemShouldUpdate: _itemShouldUpdate()
▶ keyExtractor: keyExtractor()
  keyboardDismissMode: "on-drag"
  keyboardShouldPersistTaps: "handled"
□ legacyImplementation: false
  maxToRenderPerBatch: 10
  onEndReachedThreshold: 2
▶ renderItem: fn()
▶ renderScrollComponent: renderScrollComponent()
▶ renderSectionHeader: renderSectionHeader()
  scrollEventThrottle: 50
▶ sections: Array[2]
☑ stickySectionHeadersEnabled: true
  style: 27
  updateCellsBatchingPeriod: 50
  windowSize: 21

React Native Style Editor

backgroundColor : #eeeeee
```

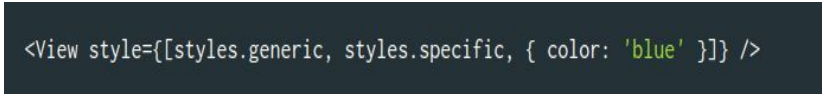
References:
https://reactnative.dev/docs/debugging

5. Styling



```
      <View style={styles.container}>
        <Text>Open up App.js to start working on your app!</Text>
        <Text>Changes you make will automatically reload.</Text>
        <Text>Shake your phone to open the developer menu.</Text>
      </View>
    );
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    backgroundColor: '#fff',
    alignItems: 'center',
    justifyContent: 'center',
  },
});
```

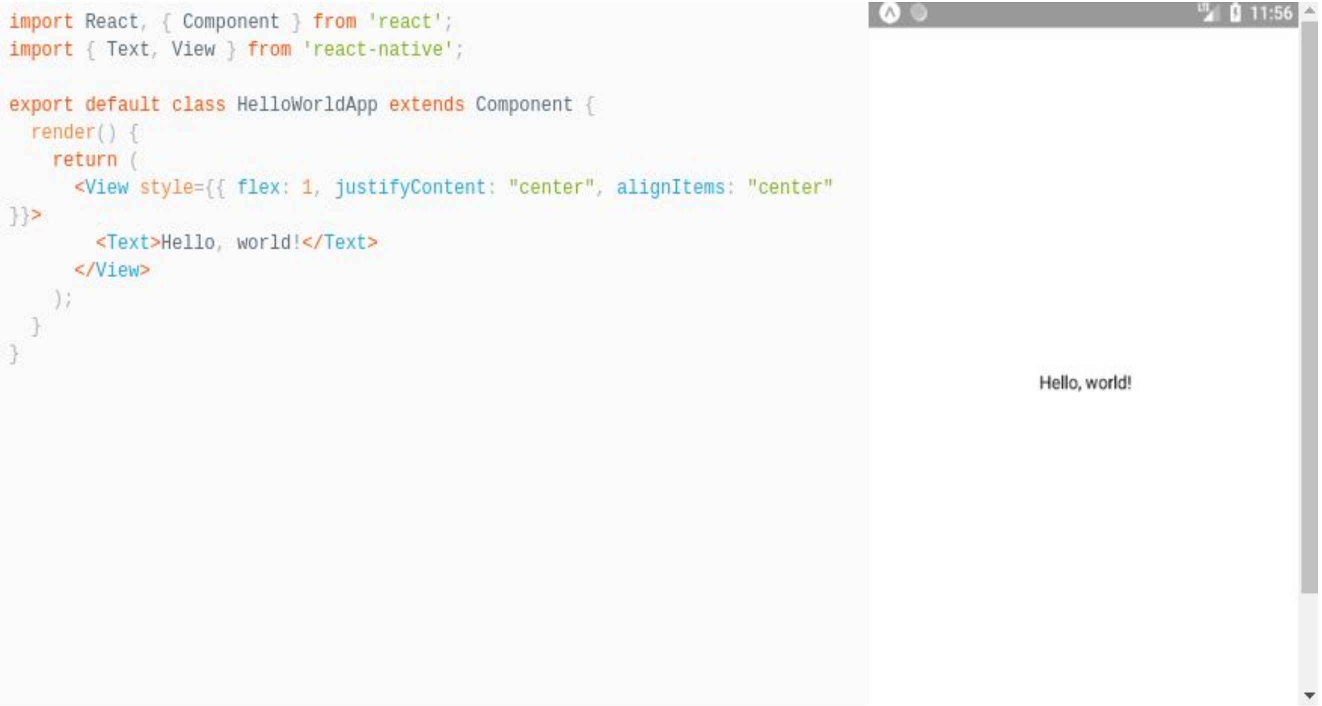
Applying multiple styles to a component



```
<View style={[styles.generic, styles.specific, { color: 'blue' }]} />
```

Explore: Flexbox & React Native UI Library

Let’s practice together!



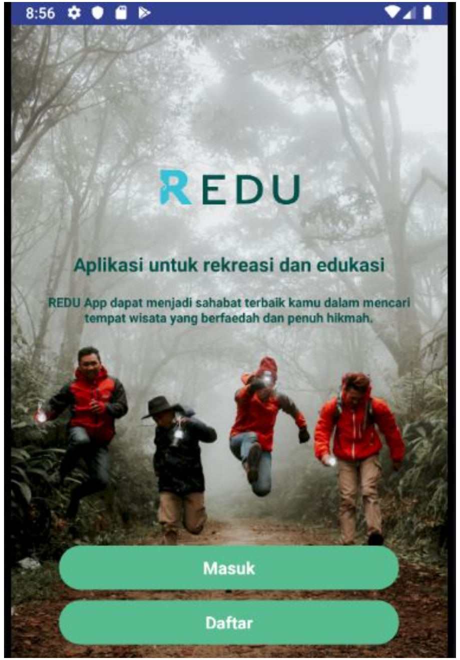
```
import React, { Component } from 'react';
import { Text, View } from 'react-native';

export default class HelloWorldApp extends Component {
  render() {
    return (
      <View style={{ flex: 1, justifyContent: "center", alignItems: "center"
}}>
        <Text>Hello, world!</Text>
      </View>
    );
  }
}
```

```
Hello, world!
```

QnA

Exercise

Create Form Login Page ;)



8:56

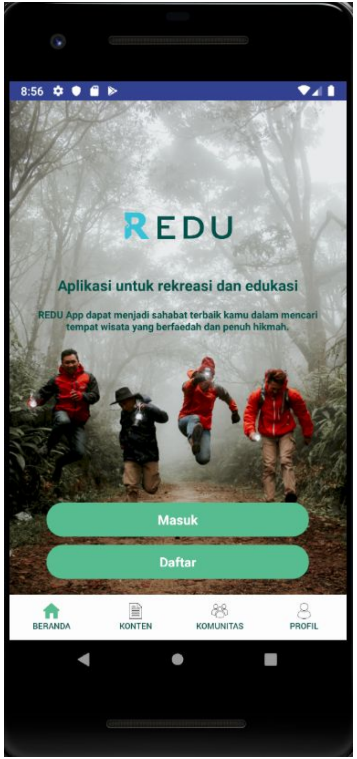
REDU

Aplikasi untuk rekreasi dan edukasi

REDU App dapat menjadi sahabat terbaik kamu dalam mencari tempat wisata yang berfaedah dan penuh hikmah.

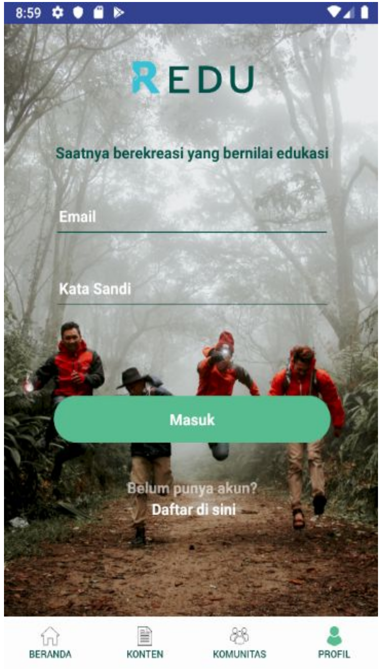
1. Text
2. Image (Logo & Background)
3. Button/TouchableOpacity
4. Debugging
5. Styling

Spoiler Next Session

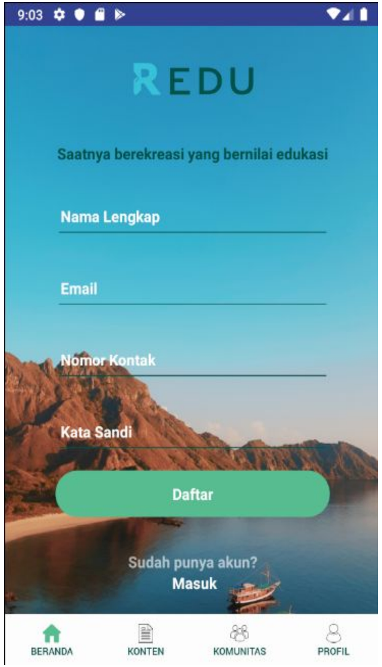


Aplikasi untuk rekreasi dan edukasi

REDU App dapat menjadi sahabat terbaik kamu dalam mencari tempat wisata yang berfaedah dan penuh hikmah.



Saatnya berekreasi yang bernilai edukasi



Saatnya berekreasi yang bernilai edukasi

Managing the presentation of, and transition between, multiple screens is typically handled by what is known as a **navigator**.

See u again. Stay healthy!
Thank u!