Introduction to React Native

March 29th, 2019

ralfi
SALHON
Developer & Graphic Designer

Education
Tufts University
Computer Science (BS)

Website
tufts.io/ralfi

Github
github.com/ralfisalhon

Email
rifat.salhon@tufts.edu

01 PROGRAMMING LANGUAGES

Proficient: C, C++, HTML, Javascript, Python

Familiar: C#, Java, React.js, Swift

02 SOFTWARE & LIBRARIES

Proficient: Photoshop, PyxelEdit, React Native, Unity

Familiar: After Effects, Final Cut Pro, Illustrator, Xcode

ABOUT ME
Projects

01 RED or BLUE

2D Puzzle platformer game in a dynamically changing world. Currently free to download on our website.

Qualified for Phase 2 submissions of Boston Festival of Indie Games.

Technologies Used:
Unity (C#), PyxelEdit

bit.ly/redorbluegame
github.com/mohsr/red-or-blue

02 Hearo

Mobile app that automatically summarizes conference calls using voice transcription. Currently unreleased.

Technologies Used:
React Native (JS), Xcode, Voximplant API, Google Speech-to-Text API, Twilio API

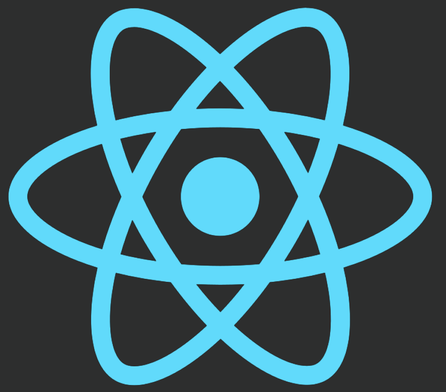
03 Jamblr

Mobile app that allows for easy music discovery and playlist creation. Currently on the App Store.

Won the HubSpot special award for best user experience and engineering values at Tufts Polyhack 2018.

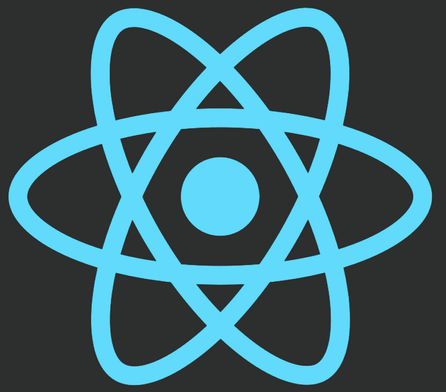
Technologies Used:
React Native (JS), Node, Xcode, Spotify Web API

https://apple.co/2JCioyX



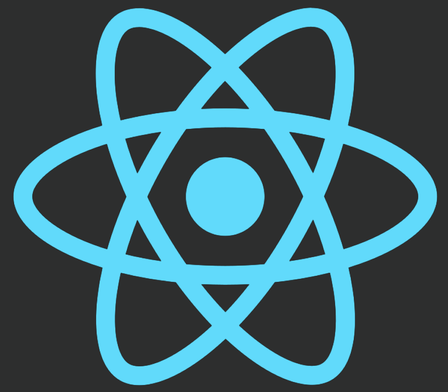
What is React Native?

- Built using React
- Build cross platform applications using Javascript
- Released on March 2015 by Facebook
- Rapidly gaining in popularity & adoption



Why React Native?

- Fast speed of development (hot reloading)
- Low cost of development (code reuse)
- Ship across multiple platforms
- Possible to ship “over the air” updates



Who’s using React Native?

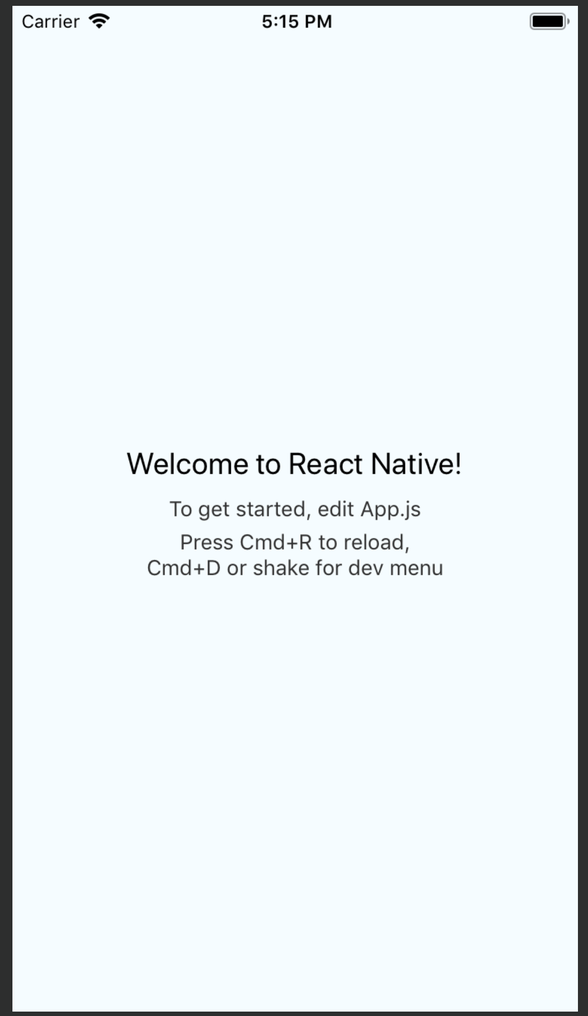
- Instagram
- AirBNB
- Skype
- Discord

Mapping to Web UI Primitives

- View == div
- Text == span / p / h
- Image == img
- TextInput == input

Creating a new Project

- *brew install node*
- *brew install watchman*
- *npm install -g react-native-cli*
- react-native init AwesomeProject



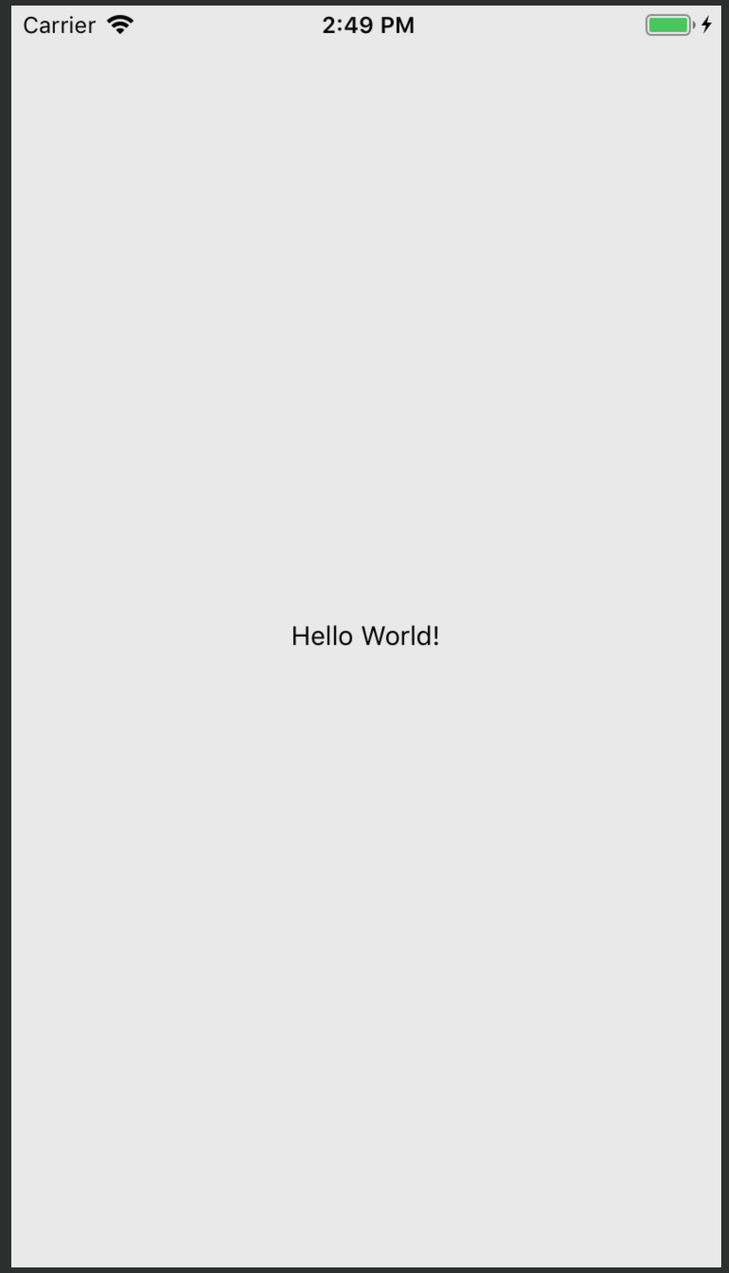
https://facebook.github.io/react-native/docs/getting-started.html

Package Imports

Basic Class Structure

Styling (like CSS)

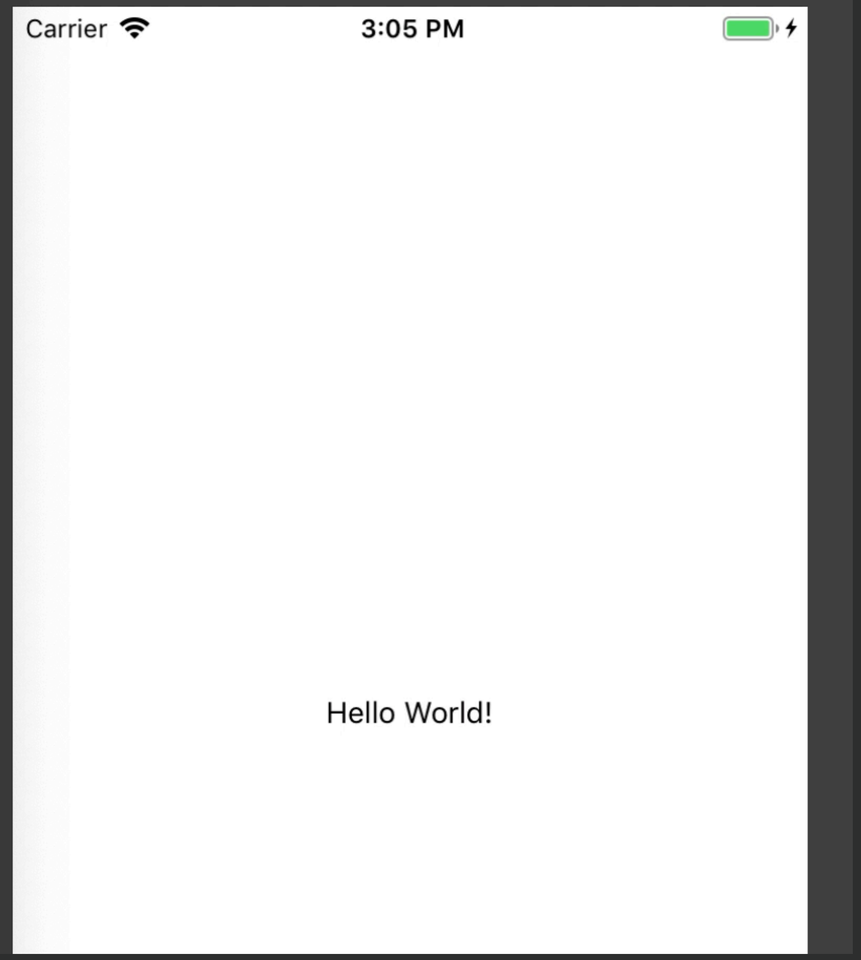
```
1  import React from "react";
2  import {
3      StyleSheet,
4      View,
5      Image,
6      Text,
7  } from "react-native";
8
9  export class HomeScreen extends React.Component {
10     render() {
11         return (
12             <View style = {styles.container}>
13                 <Text> Hello World! </Text>
14             </View>
15         );
16     }
17 }
18
19 const styles = StyleSheet.create({
20     container: {
21         flex: 1,
22         justifyContent: 'center',
23         alignItems: 'center',
24     },
25 });
```



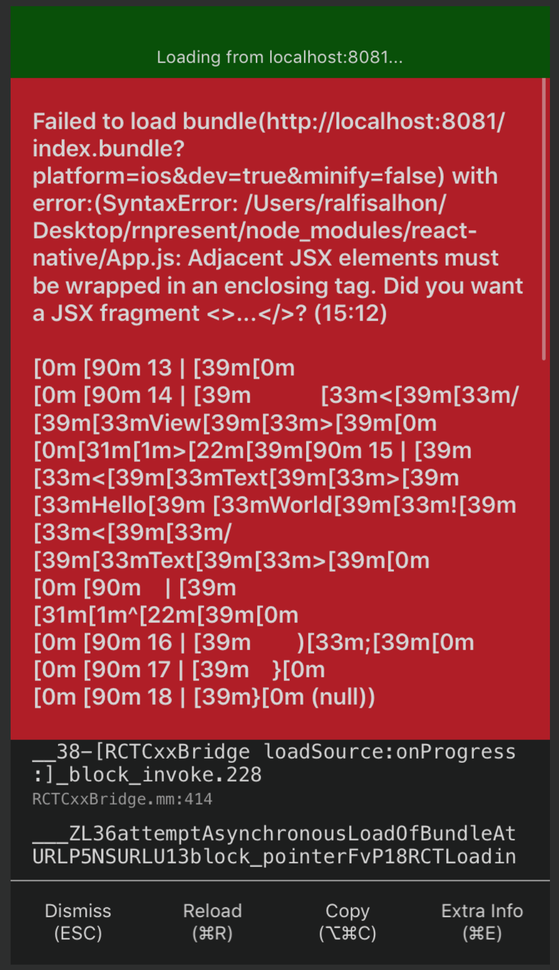
The Power of Hot Loading

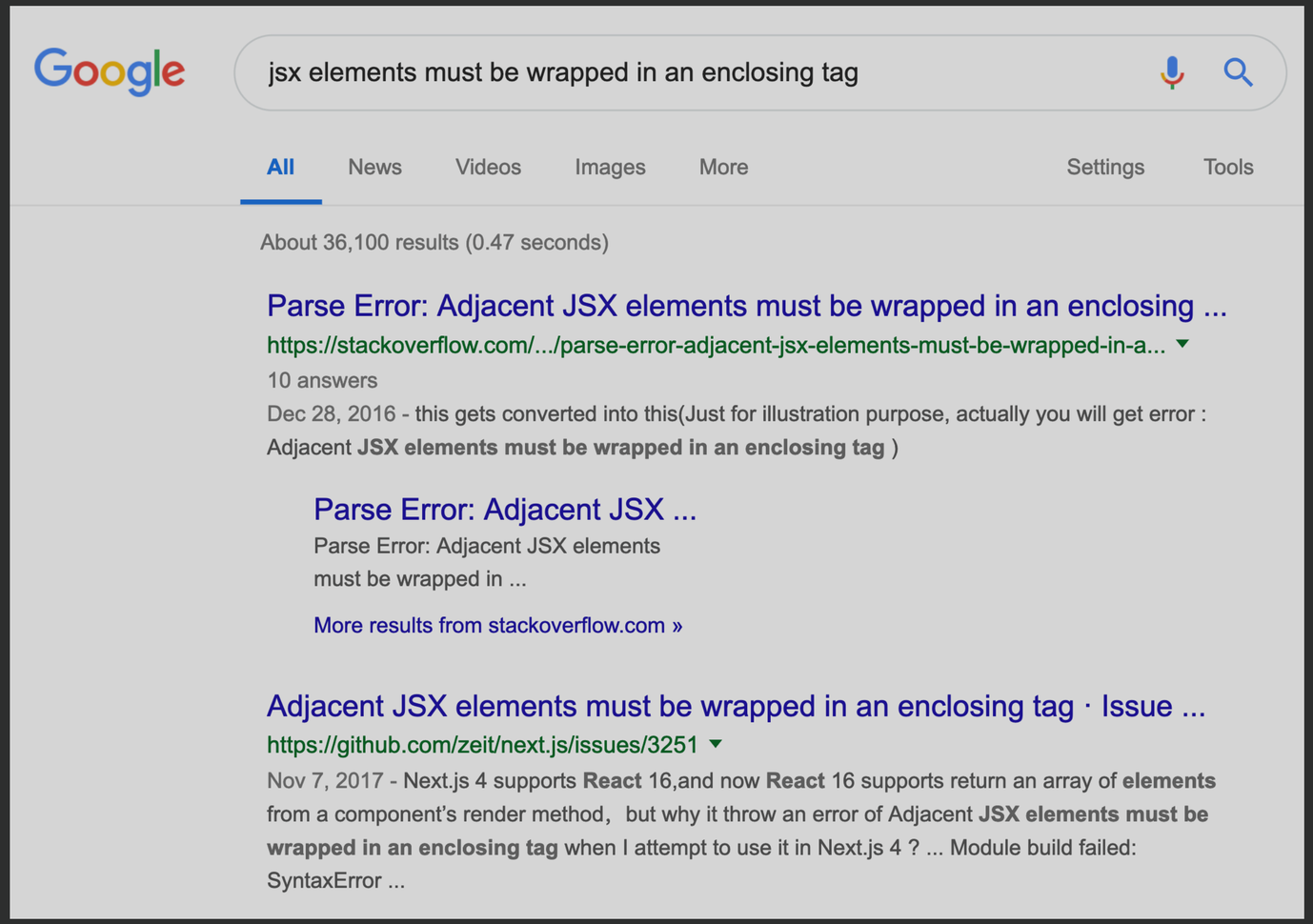
App.js

```
1  import React from "react";
2  import {
3      StyleSheet,
4      View,
5      Image,
6      Text,
7  } from "react-native";
8
9  export default class HomeScreen extends React.Component {
10     render() {
11         return (
12             <View style = {styles.container}>
13                 <Text> Hello World! </Text>
14             </View>
15         );
16     }
17 }
18
19 const styles = StyleSheet.create({
20     container: {
21         flex: 1,
22         justifyContent: 'center',
23         alignItems: 'center',
24     },
25 });
26
```



CONS of React Native





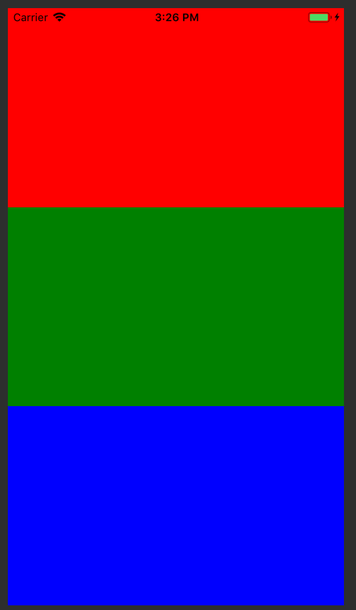
React Native is not beginner friendly –> Google is your best friend.

Debugging can take hours –> WRITE CLEAN CODE!

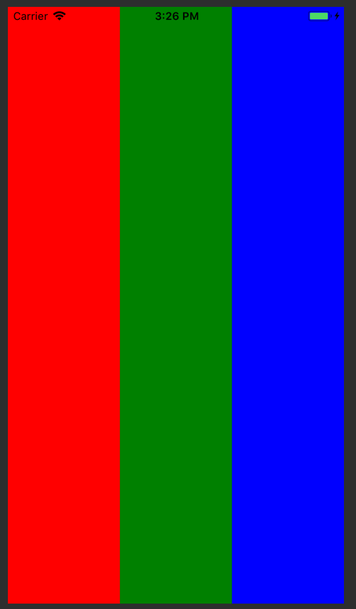
Learning flex

```
export default class HomeScreen extends React.Component {
    render() {
        return (
            <View style = {{flex: 1}}>
                <View style = {{flex: 1, backgroundColor: 'red'}}>
                </View>
                <View style = {{flex: 1, backgroundColor: 'green'}}>
                </View>
                <View style = {{flex: 1, backgroundColor: 'blue'}}>
                </View>
            </View>
        );
    }
}
```

by default,
flexDirection: ‘column’



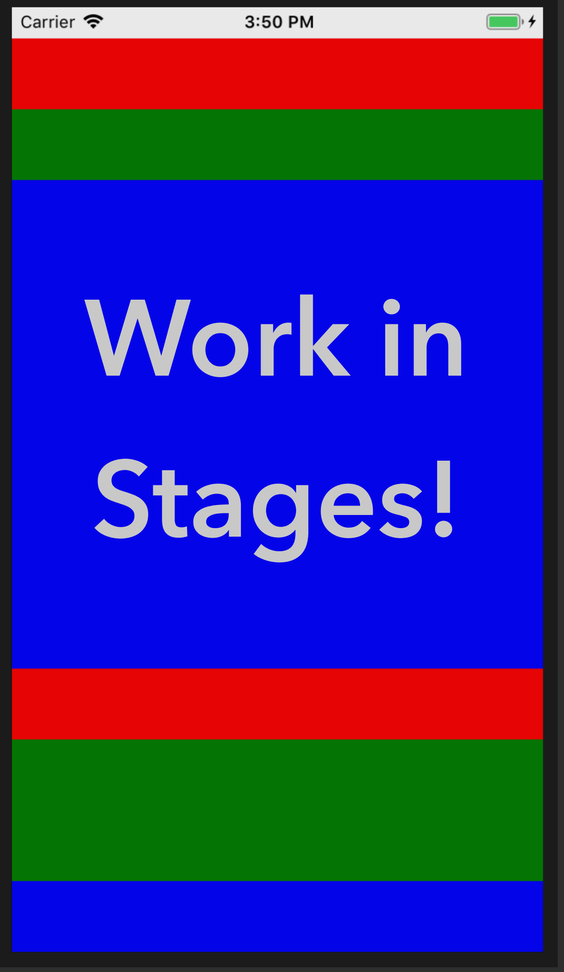

```
export default class HomeScreen extends React.Component {
    render() {
        return (
            <View style = {{flex: 1, flexDirection: 'row'}}>
                <View style = {{flex: 1, backgroundColor: 'red'}}>
                </View>
                <View style = {{flex: 1, backgroundColor: 'green'}}>
                </View>
                <View style = {{flex: 1, backgroundColor: 'blue'}}>
                </View>
            </View>
        );
    }
}
```



Exploring Instagram's <View>'s

App.js

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

export default class HomeScreen extends React.Component {
    render() {
        return (
            <View style = {styles.container}>
                <View style = {{height: 50, backgroundColor: 'red'}}>
                </View>
                <View style = {{height: 50, backgroundColor: 'green'}}>
                </View>
                <View style = {{flex: 1, backgroundColor: 'blue'}}>
                </View>
                <View style = {{height: 50, backgroundColor: 'red'}}>
                </View>
                <View style = {{height: 100, backgroundColor: 'green'}}>
                </View>
                <View style = {{height: 50, backgroundColor: 'blue'}}>
                </View>
            </View>
        );
    }
}

const styles = StyleSheet.create({
    container: {
        flex: 1,
        marginTop: 22,
    },
});

```



height: 50

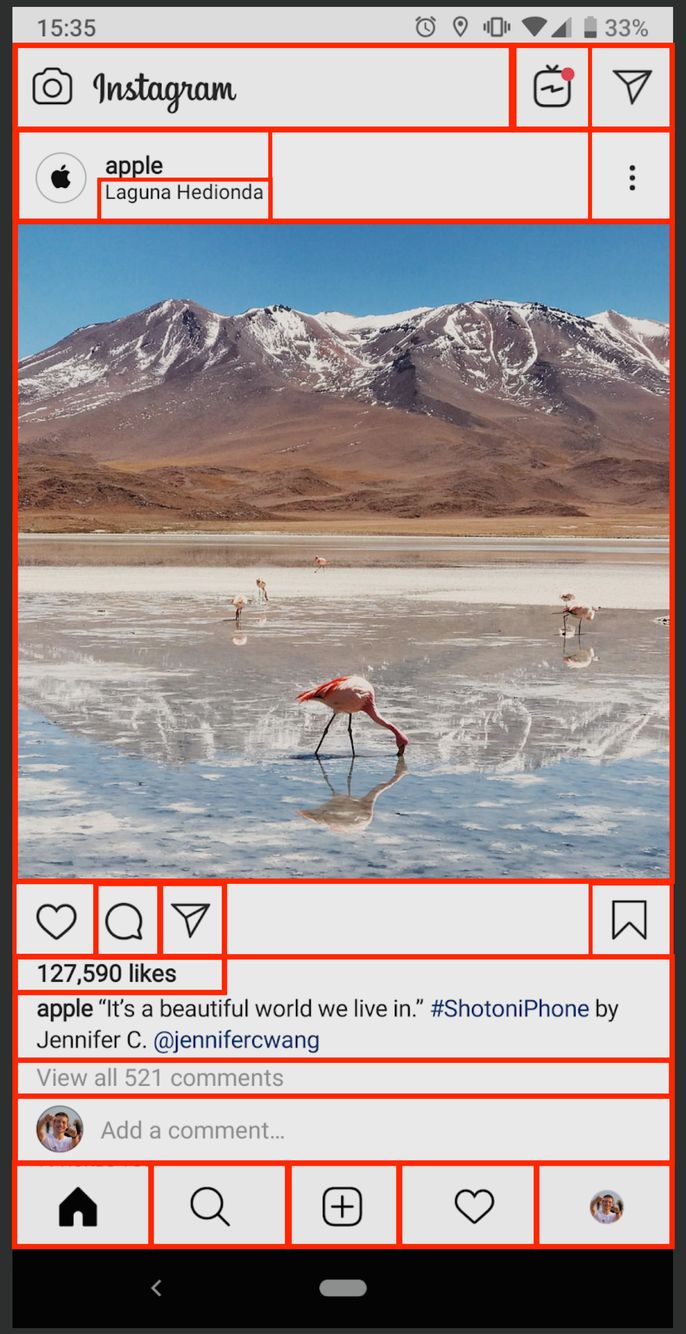
height: 50

flex: 1

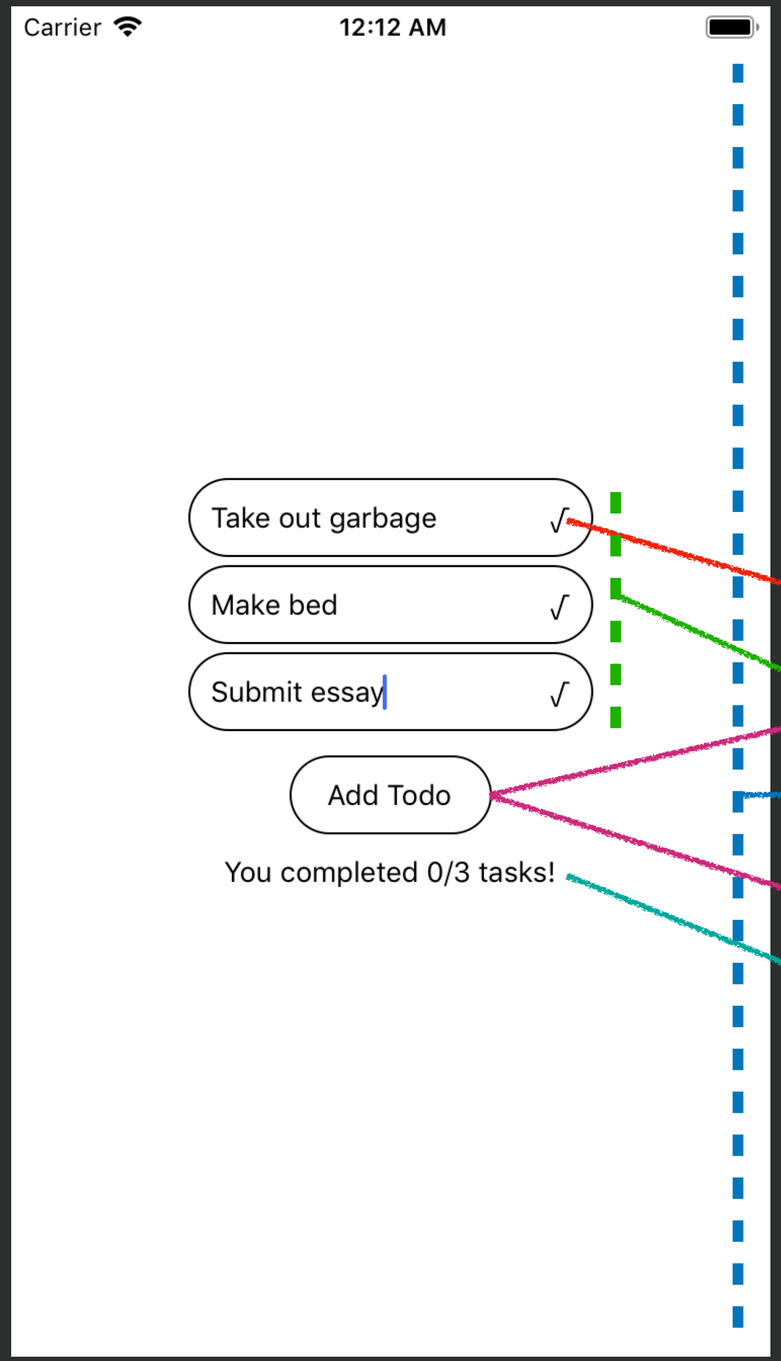
height: 50

height: 100

height: 50

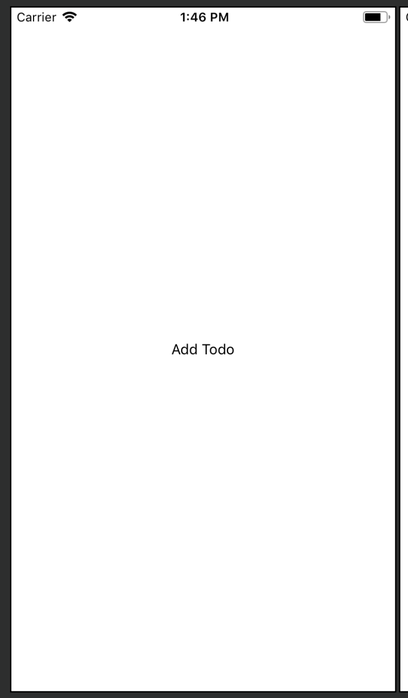


Let's make a
To-Do Application



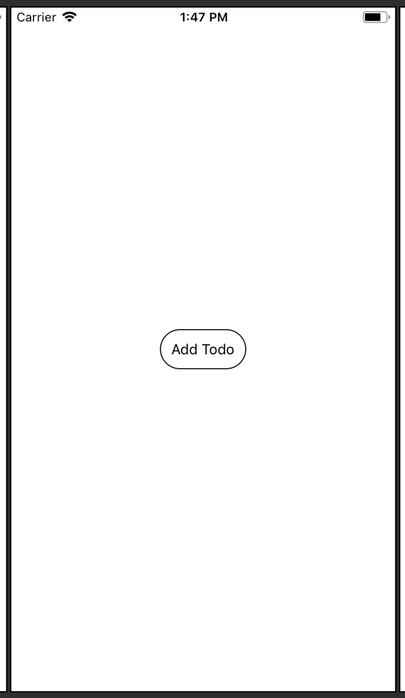

```
import React from "react";
import {
    StyleSheet,
    View,
    Text,
    TouchableOpacity,
    TextInput,
} from "react-native";

export default class HomeScreen extends React.Component {
    constructor()
    {
        super();
        this.state = {…};
    }

    isCompleted(index) {…}
    completeItem(index) {…}
    addItem() {…}

    renderTodos = () => {…}
    render() {
        return (
            <View style = {styles.container}>
                {this.renderTodos()}
                <TouchableOpacity
                style = {styles.button}
                onPress={() => this.addItem()}>
                    <Text>Add Todo</Text>
                </TouchableOpacity>
                <Text>…</Text>
            </View>
        );
    }
}

const styles = StyleSheet.create({…});
```

Stages of Development



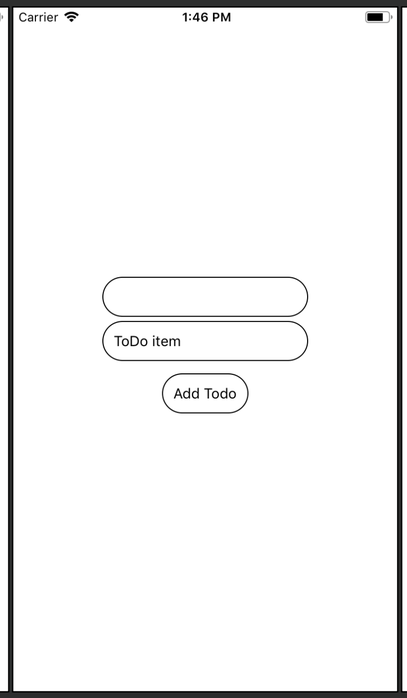
01 Start new React Native App. For your main view, set Justify Content and Align Items to ‘center’.

Test by rendering a Text element in the middle. Call it “Add ToDo”

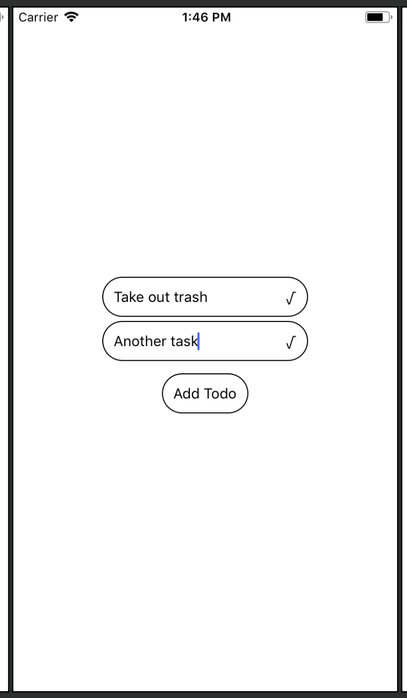


02 Wrap the text in a TouchableOpacity. Give it some styling like a borderWidth.

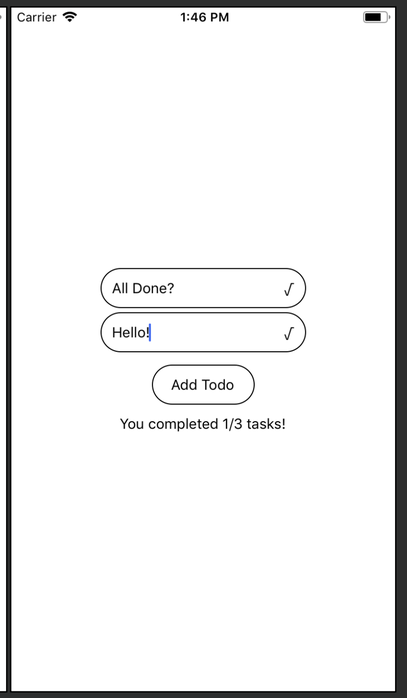
Test by clicking on the item. You should see the Opacity change.



03 Create a state to keep track of how many ToDo items you have. Make a render method to create that many ToDo boxes. Use TextInput inside of the boxes to be able to write text.



04 Add new button next to each button to complete it. This should call a function to remove that item. You could keep track of the deleted items in another state variable, like an array of completed item indexes.



05 Create a Text component on the bottom that displays the amount of tasks left, and the amount of tasks completed.

Live Coding

Introduction to React Native

March 29th, 2019

ralfi
SALHON
Developer & Graphic Designer

Education
Tufts University
Computer Science (BS)

Website
tufts.io/ralfi

Github
github.com/ralfisalhon

Email
rifat.salhon@tufts.edu