TRACY LEE | @ladyleet

SPEAKER & GOOGLE DEVELOPER EXPERT
DEV - ANGULAR, EMBER, REACT, RXJS
THIS DOT MEDIA / LABS
TECH WRITER & PODCASTER
MENTOR
React + React Native + Native
React Native is not ALL JavaScript
React Native is not ALL JavaScript.
React Native is powerful because we have a JavaScript interface for code implemented natively.
A PRACTICAL APPROACH TO REACT NATIVE
JS JEOPARDY!
OUR AGENDA

1. React Native Components
2. Building an app
3. Styling
4. Navigation
5. React Native Ecosystem

@ladyleet
React Native Components
Basic Components
User Interface
List Views
iOS Components & APIs
Android Components & APIs
@ladyleet
## Basic Components

<table>
<thead>
<tr>
<th>View</th>
<th>Text</th>
<th>Image</th>
</tr>
</thead>
<tbody>
<tr>
<td>The most fundamental component for building a UI.</td>
<td>A component for displaying text.</td>
<td>A component for displaying images.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>TextInput</th>
<th>ScrollView</th>
<th>Button</th>
</tr>
</thead>
<tbody>
<tr>
<td>A component for inputting text into the app via a keyboard.</td>
<td>Provides a scrolling container that can host multiple components and views.</td>
<td>A basic button component for handling touches that should render nicely on any platform.</td>
</tr>
</tbody>
</table>

@ladyleet
CREATING A PROJECT

1. Install Xcode
2. $ yarn global add react-native-cli
3. $ react-native init <appName>
$ react-native run-ios
Welcome to React Native!
To get started, edit index.ios.js
Press Cmd+R to reload,
Cmd+D or shake for dev menu

React Native packager is running.

Visit documentation

@ladyleet
THE MAGIC TRICK

1. kill all your processes
2. $ npm start — —reset-cache

(solves 99% of your problems)
SETUP TIPS

Create an /src folder with an App.js
Copy/paste index.ios.js into App.js
Import App.js into index.ios.js and index.android.js
import React, { Component } from 'react';

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

export default class JavaScriptJeopardyRn extends Component {
  render() {
    return {
      <View style={styles.container}>
        <Text style={styles.welcome}>
          Welcome to React Native!
        </Text>
        <Text style={styles.instructions}>
          To get started, edit index.ios.js
        </Text>
      </View>
    }
  }
}

const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5FCFF',
  },
  welcome: {
    fontSize: 20,
    textAlign: 'center',
    margin: 10,
  },
  instructions: {
    textAlign: 'center',
    color: '#333333',
    marginBottom: 5,
  },
});

AppRegistry.registerComponent('JavaScriptJeopardyRn', () => JavaScriptJeopardyRn);
STYLING
STYLING

Using styled-components 🎇

@ladyleet
const styles = StyleSheet.create({
  container: {
    flex: 1,
    justifyContent: 'center',
    alignItems: 'center',
    backgroundColor: '#F5FCFF',
  },
  welcome: {
    fontSize: 20,
    textAlign: 'center',
    margin: 10,
  },
  instructions: {
    textAlign: 'center',
    color: '#333333',
    marginBottom: 5,
  }
})
$ yarn add styled-components
import React, { Component } from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  View,
  Image,
  Linking
} from 'react-native';

import styled from 'styled-components/native';

$ yarn add styled-components

@ladyleet
Create a new variable, which becomes a React Native component.

```javascript
const Container = styled.View;
```
STYLED COMPONENTS

Write normal CSS

```javascript
const Container = styled.View
  .justifyContent: center;
  alignItems: center;
  backgroundColor: #F5FCFF;
  margin: 40px;
;
const TextStyled = styled.Text
  .fontSize: 20;
  textAlign: center;
  margin: 10px;
```
STYLED COMPONENTS

```javascript
<View style={styles.container}>
  <Text style={styles.welcome}>
    Welcome to React Native!
  </Text>
  <Text style={styles.instructions}>
    To get started, edit index.ios.js
    Press Cmd+R to reload, {'

    Cmd+D or shake for dev menu
  </Text>
</View>

<Container>
  <TextStyled>
    Welcome to React Native!
  </TextStyled>
  <Text>
    To get started, edit index.ios.js
    Press Cmd+R to reload, {'

    Cmd+D or shake for dev menu
  </Text>
</Container>
```
NAVIGATION
USING REACT-Navigation

$ yarn add react-navigation
NAVIGATION

Built-In
StackNavigator
TabNavigator
DrawerNavigator

+ Custom Navigators
import { TabNavigator } from 'react-navigation';
Create components you will be navigating to in /src.

Import those components into your App.js

Create a variable to define the TabNavigator.
```javascript
const mainNavigator = TabNavigator({
  Home: {
    screen: javascriptJeopardyRn,
    path: ''
  },

  Angular: {
    screen: Angular,
    path: 'angular'
  },

  React: {
    screen: ReactJS,
    path: 'react'
  },

  Ember: {
    screen: Ember,
    path: 'ember'
  },

  Vue: {
    screen: Vue,
    path: 'vue'
  }
});

AppRegistry.registerComponent('javascriptJeopardyRn', () => mainNavigator);
```
<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>tabBarComponent</td>
<td>activeTintColor</td>
</tr>
<tr>
<td>TabBarBottom</td>
<td>activeBackgroundColor</td>
</tr>
<tr>
<td>TabBarTop</td>
<td>showLabel</td>
</tr>
<tr>
<td>tabBarPosition</td>
<td>style</td>
</tr>
<tr>
<td>swipeEnabled</td>
<td>tabStyle</td>
</tr>
<tr>
<td>animationEnabled</td>
<td>tabBarLabel</td>
</tr>
<tr>
<td>lazy</td>
<td>tabBarIcon</td>
</tr>
</tbody>
</table>
import React, { Component } from 'react';
import {
  AppRegistry,
  StyleSheet,
  Text,
  View,
  Image
} from 'react-native';
tabBarIcon

```javascript
Vue.navigationOptions = {
  tabBarIcon: () => {
    <Image source={require('./img/vue.png')} />
  }
}
```
Find this project on
Github
Made with ❤️ by
@ladyleet
DISPLAYING DATA
Create 2 components Questions.js and Answer.js
Each component should have a `<View/>` and `<Text/>`
Use `{ this.props.q }` and `{ this.props.a }`
We’ll pass the data on the parent component
DISPLAYING DATA

```html
<View>
  <Text>{this.props.q}</Text>
</View>
```
DISPLAYING DATA

Import Questions.js and Answer.js into a component

Ex. the React.js component

Create an array of data with questions and answers

@ladyleet
DISPLAYING DATA

Simply write React!

```jsx
<ScrollView>
  {data.map(trivia =>
    <View>
      <Question q={trivia.Q}/>
      <Answer a={trivia.A}/>
    </View>
  )}
</ScrollView>
```
OUR AGENDA

1. React Native Components
2. Building an app
3. Styling
4. Navigation
5. React Native Ecosystem
React Native Ecosystem

State Management
Animations
Where to find help
State Management

Flux/Redux - redux.js.org
MobX - mobx.js.org
Morearty - github.com/moreartyjs
Animations

React Native Animatable
github.com/oblador/react-native-animatable

React Native’s Animated API
facebook.github.io/react-native/docs/animated.html

Lottie by Airbnb
airbnb.design/introducing-lottie
React Native’s Animated API

Animated gives you 4 animatable component types: Text, Image, View & ScrollView

Or you can create your own using Animated.createAnimatedComponent()
Lottie by Airbnb

Render After Effects animations in real time
Where to find help

Twitter
Documentation
Medium articles
egghead.io
Front End Masters
React Native is amazing for JavaScript developers who want high flexibility and the ability to build mobile apps with a powerful JavaScript interface.
React Native is best for people who want to have full control of their applications and understand the intricacies of their code’s inner workings without heavy abstractions.
Let’s get into building the actual app!