

INTRODUCTION TO REACT NATIVE

with Chris Minnick

@chrisminnick



INTRODUCTION



ABOUT ME



INTRODUCTIONS

- What's your name?
- What do you do?
- JavaScript level (beginner, intermediate, advanced)?
- What do you want to know at the end of this course?
- Favorite food?

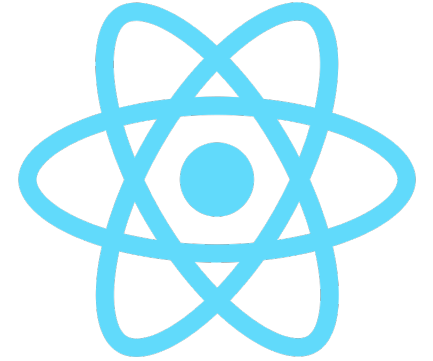


DAILY SCHEDULE

- 08:30 - 10:30
- 15 minute break
- 10:45 - 12:00
- 1 hour lunch break
- 1:00 - 2:00
- 15 minute break
- 2:15 - 3:15
- 15 minute break
- 3:30 - 4:30



WHAT IS REACT NATIVE?



- Built using React
- Build cross platform applications using JavaScript
- Target not only iOS & Android, but also Apple TV, VR, AR, Windows, & Desktop
- Released March 2015
- Rapidly gaining in popularity & adoption
- Learn once write anywhere
- Real world projects typically see 80% – 95% code reuse



WHO'S USING REACT NATIVE

- Facebook
- Instagram
- Walmart
- Salesforce
- Amazon
- Skype
- Baidu
- American Express
- Tesla
- Adidas
- Discord
- Wix
- Bloomberg
- Visa
- Microsoft
- JD.com



WHY REACT NATIVE

- Faster speed of development
- Lower cost of development
 - Code reuse
 - Easier to find developers
- Ship across multiple platforms
- Possible to ship over the air updates bypassing App Store / Play Store



BENEFITS

- More people know JavaScript than any other language
- Can compile the same code to run on multiple platforms
- The resulting app is a Native app
 - Not a hybrid
 - Not a mobile web app



BENEFITS OF NATIVE

- Performance
- Platform API access
- Naive UI components
- If you need native code, you can use it
 - Just drop in native components
 - written in Objective C, Java, or Swift



HOW DOES REACT NATIVE WORK?

- Write your application in JavaScript
- JavaScript is transpiled & minified
- Separate threads for UI, layout, and JavaScript
- Threads communicate asynchronously through the native bridge



CREATING A NEW REACT NATIVE PROJECT

Create React Native App CLI

- Builds without native projects & code
- Native code not configurable
- Will run without all SDKs & IDEs installed
- Great for newer developers
- Includes features like Camera, Location, Push Notifications, Social Authentication, Accelerometer, File System, & Image Picker configured out of the box
- Can eject into a native project at any time



CREATING A NEW REACT NATIVE PROJECT

React Native CLI

- Builds with native projects & code
- Native code configurable
- Needs proper environment setup in order to run
- Does not include some important APIs without additional configuration



CREATING A NEW REACT NATIVE PROJECT

Creating a new project

```
react-native init MyProjectName
```

```
cd MyProjectName
```

```
react-native run-ios / run-android
```



EXPO

- A native and JavaScript SDK for accessing device functionality.
- XDE
 - Expo Development Environment
 - helps with publishing and previewing
- Command Line Tools
- Expo Client
 - Lets you view your project as you're developing it



GET STARTED WITH EXPO

- `npx create-expo-app my-app`
- `cd my-app`
- `npx expo start`



TESTING AND RUNNING REACT NATIVE APPS

- Simulators

- Android

- Emulation

- Genymotion

- iOS

- iOS simulator comes with XCode

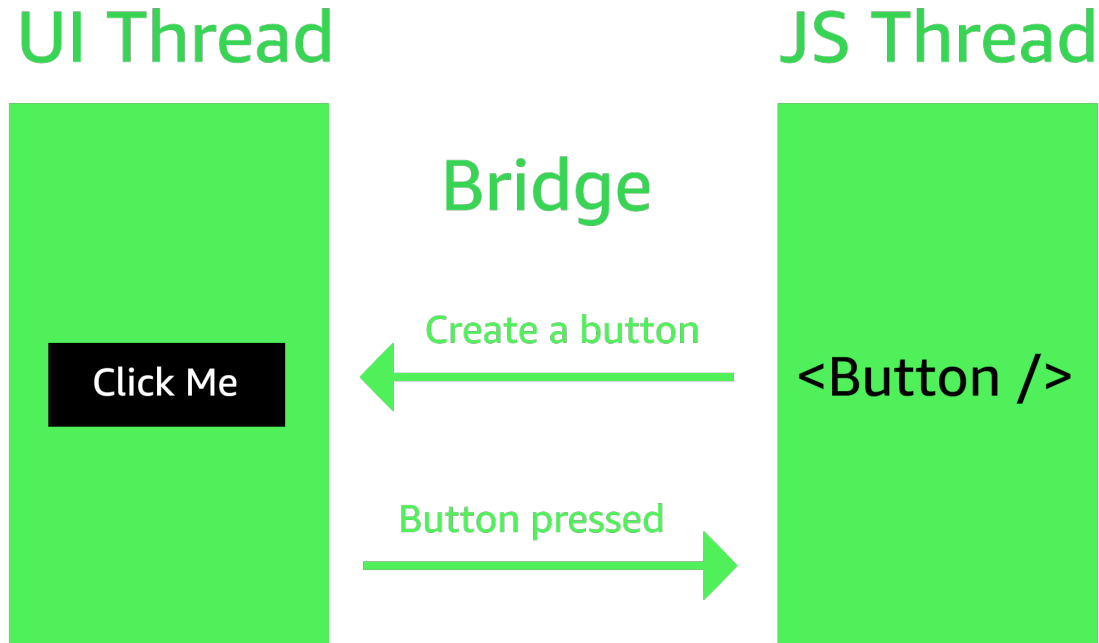


HOT RELOADING

- Hot Reloading is **AWESOME!**



HOW DOES REACT NATIVE WORK?



REACT NATIVE API

- Over 30 UI Primitives

- View
- Text
- ScrollView
- Image
- TextInput
- Switch



REACT NATIVE API

- Mapping to Web UI primitives

- View == div
- Text == span / p / h*
- ScrollView == div + styling
- Image == img
- TextInput == input



REACT NATIVE API - COMPONENTS

- Primitives not globally in scope must be imported

```
import {  
  View  
  Text  
  ScrollView  
  Image  
  TextInput  
  Switch  
} from 'react-native'
```



REACT NATIVE API — DEVICE APIS

Device APIs

CameraRoll

Keyboard

NetInfo

Vibration

PanResponder



REACT NATIVE VS WEB



REACT NATIVE VS WEB

- Different primitives
- Styling
- Most browser APIs do not exist
 - Some browser APIs polyfilled (geolocation, fetch, timers, console)
- Navigation
- Mobile UI / UX
- Native IDEs / configuration
- Deployment



REACT NATIVE VS WEB

Event handling

- Unlike web, not every component has every interaction
- Handful of “touchable” components
 - Button, TouchableHighlight, TouchableOpacity, TouchableWithoutFeedback
- Create custom touchable components with PanResponder API



REACT NATIVE VS WEB

Styling

- Based on CSS styles
- Layout is done with FlexBox
- Style prop can be one style or an array of styles
- Styles can be created & updated dynamically
- Use StyleSheet API for perf optimization



REACT NATIVE VS WEB

Styling

```
<View
  style={{
    margin: 20,
    backgroundColor: 'red'
  }}
/>
```



REACT NATIVE VS WEB

Styling

```
const styles = {  
  container: {  
    margin: 20,  
    backgroundColor: 'red'  
  }  
}
```

```
<View  
  style={styles.container}  
>
```



REACT NATIVE VS WEB

Styling - StyleSheet

```
import { StyleSheet } from 'react-native'

const styles = StyleSheet.create({
  container: {
    margin: 20,
    backgroundColor: 'red'
  }
})

<View style={styles.container} />
```



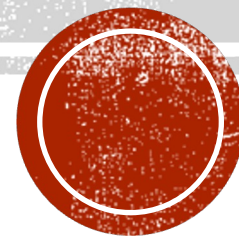
REACT NATIVE VS WEB

Styling – Array of styles

```
<View style={[styles.container, styles.blueBG]} />
```



LET'S MAKE AN APP!



THANKS!

chrisminnick.com

[@chrisminnick](https://twitter.com/chrisminnick)

github.com/chrisminnick

