**(0:00) What This Guide Covers (React Native Crash Course):**

* What is React Native?
* React Native Installation & Setup
  + Even though we use Windows. Using React Native CLI (Command-line Interface) is the same on Mac or Linux. It is NPM-based.
* Creating Components
* Key React Native UI Features (such as input controls, list views and touchable elements)
* Key API Features (such as async. storage and messaging API)

**(1:20) What React Native Is:**

* A framework for building mobile apps using JavaScript and the React library
* For building cross-platform apps (Android/iOS)
  + Instead of building an iOS app using Swift and an Android app in Java for Android
* Uses almost all React.js concepts including components, state, props, lifecycles, etc.

**(2:30) Real Mobile Apps:**

* Some frameworks one may use for hybrid app development: Cordova/PhoneGap, Ionic and Sencha Touch. They are basically web apps that run inside of a native container/WebView.
* React Native apps are not hybrid apps
* They do not run in a WebView
* Use the same fundamental building blocks as a native app built with Swift/Objective-C/Java
* Better and faster than Cordova/PhoneGap and Ionic

**(3:30) Advantages of React Native:**

* Cross Platform
* Much Less Expensive
* Easier to Code
* Save Massive Amount of Time
  + One does not need to recompile things all the time
* Open Source

**(5:09) UI Component Examples:**

* TextInput, Picker, Switch, Slider
* Touchable Elements
  + When you tap a button in a web app, you get only the standard browser effect
  + With a native app, you get a more natural effect. We have the touchable highlight, touchable opacity, etc.
* ListView & ScrollView
  + We can use Flexbox to style ListView
* Alerts, Modals, ProgressBars
* StatusBar
* TabBarIOS
* ToolbarAndroid

**(7:15) APIs for Device Interaction:**

* CameralRoll
* AsyncStorage
* Geolocation
* ImageEditor
* PushNotifications
* Vibration
* Share/Messages

**(7:51) Development Environment & Specs:**

* Windows
  + Android Studio
  + Android SDK (Software Development Kit) (6.0 Marshmallow)
  + AVD (Android Virtual Device)
* Mac
  + Xcode
  + Simulator

**Android SDK & AVD Setup for React Native (Extra):**

You need Android Studio, SDK and AVD set up if you want to use React Native, Cordova or Ionic (refer: https://facebook.github.io/react-native/docs/getting-started.html)

1. Make sure Java, Node is installed
2. Install Python2 via Chocolatey (“choco install -y python2”)
3. Install Android Studio
   * We do not use it (it is more for Java development)
4. Install SDK, HAXM (Hardware Accelerated Execution Manager), build tools, system image
   * The SDK version is Android 6.0 (Marshmallow)
   * HAXM does AVD acceleration
   * Google APIs, Google APIs Intel x86 Atom\_64 System Image, Android SDK Platform-tools and Android Build-tools 23.0.1, 23.0.2, 23.0.3 is also needed
   * System image (Intel x86 Atom\_64 System Image) is for the AVD
   * Android 6.0 SDK, HAXM, Google APIs, Google APIs Intel x86 Atom\_64 System Image, Android SDK Platform-tools, Android Build-tools 23.0.1, 23.0.2, 23.0.3, Intel x86 Atom\_64 System Image can be installed via Android Studio
5. Edit PATH variable (so that we can run the SDK manager (no longer available) and other things through the command line)
   * Go to Control Panel >> System and Security >> System >>Advanced system settings >> Environment Variables
   * Edit “Path” under “User variables for (username)”
   * Some entries should appear (like “C:\ProgramData\Android\Sdk\platform-tools” and “C:\ProgramData\Android\Sdk\tools”)
   * Some entries should appear under “System variables” (like “ANDROID\_HOME: C:\ProgramData\Android\Sdk” and “JAVA\_HOME: C:\Program Files\Java\jdk1.8.0\_144”)
6. Create Android virtual device
   * Create a new empty project via Android Studio (in the location such as “C:\ProgramData\Android\AndroidStudioProjects\MyApplication”)
   * Click the phone icon in the toolbar to open the AVD manager
   * Create a virtual device
   * Set the AVD name (e.g. my android device), the device (e.g. Nexus 6), Android version to 6.0 x86\_64
   * Click the play icon

**(9:08) MyApp Component (Code Explanation):**

import React, { Component } from 'react';

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

// We use AppRegistry to register a component

class MyApp extends Component {

render() {

return (

<View>

<Text>Hello World!</Text>

</View>

);

// We are not using something like “div”

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

// We pass the class name to the function,

// the second parameter is the callback function

**(11:00) Let's Code... (Install and Setup):**

* An AVD should be activated
* Install “react-native-cli” and “yarn” globally (“npm install -g react-native-cli” & “npm install -g yarn”)
* Enter “react-native init MyApp”
* Change “"react-native": "0.56.0"” to “"react-native": "0.55.4"”, “"babel-preset-react-native": "5"” to “"babel-preset-react-native": "2.1.0"” in “package.json” and run “npm install”. The app does not load if latest version is used.
* Get into the project directory, enter “react-native run-android” in the console
* The line “BUILD SUCCESSFUL” should be shown. An app will be opened in the AVD.

**(17:56) React Native Structure:**

We have “android” and “ios” files. We do not need to edit them most of the time.

/\* ./App.js \*/

/\* ... \*/

type Props = {};

export default class App extends Component<Props> {

render() {

return (

<View style={styles.container}>

<Text style={styles.welcome}>Welcome to React Native!</Text>

<Text style={styles.instructions}>To get started, edit App.js</Text>

<Text style={styles.instructions}>{instructions}</Text>

</View>

);

// Try changing “React Native” to “MyApp”

}

}

/\* ... \*/

After changing “App.js”, double tap “R” on the keyboard to reload the app. Also, try “Ctrl + M” (there is an option which allows hot reloading).

**(20:23) Coding from Scratch:**

/\* ./App.js \*/

// This is our main component and other component we build

// will just be imported to this file

import React, {Component} from 'react';

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

export default class MyApp extends Component{

render(){

return(

<View>

<Text>Hello World!</Text>

</View>

);

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

**(24:30) Component 1 - Simple TextView:**

/\* ./app/components/Component\_1/Component\_1.js \*/

import React, {Component} from 'react';

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

export default class Component1 extends Component{

render(){

return(

<View>

<Text>This is Component 1.</Text>

</View>

);

}

}

AppRegistry.registerComponent('Component1', () => Component1);

/\* ./App.js \*/

import React, {Component} from 'react';

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

import Component1 from './app/components/Component\_1/Component\_1';

export default class MyApp extends Component{

render(){

return(

<View>

<Text>Hello World!</Text>

<Component1/>

</View>

);

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

**(26:45) Component 1 - Properties (Props):**

/\* ./App.js \*/

import React, {Component} from 'react';

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

import Component1 from './app/components/Component\_1/Component\_1';

export default class MyApp extends Component{

render(){

return(

<View>

<Component1 message="Hello World!"/>

</View>

);

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

/\* ./app/components/Component\_1/Component\_1.js \*/

import React, {Component} from 'react';

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

export default class Component1 extends Component{

constructor(props){

super(props);

// Usually, we want the props passed in become state

this.state = {

name: 'David',

showName: true,

message: this.props.message

}

}

static defaultProps = {

// Try to get rid of the message passed in from "App.js"

message: 'Hi there.'

}

render(){

let name = this.state.showName ? this.state.name : '(No name)';

return(

<View>

<Text>{this.state.message}</Text>

<Text>{name}</Text>

</View>

);

}

}

AppRegistry.registerComponent('Component1', () => Component1);

**(31:37) How to Debug Console on Terminal:**

* Start a new cmd.exe and get into the project folder
* Enter “react-native log-android”
* One can view console logs in the console

**(33:20) Component 2 - In Line Styles:**

/\* ./App.js \*/

import React, {Component} from 'react';

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

import Component2 from './app/components/Component\_2/Component\_2';

export default class MyApp extends Component{

render(){

return(

<View>

<Component2/>

</View>

);

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

/\* ./app/components/Component\_2/Component\_2.js \*/

import React, {Component} from 'react';

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

export default class Component2 extends Component{

render(){

return(

<View style={{backgroundColor:'#000000'}}>

<Text style={{color: 'white'}}>Hello David.</Text>

</View>

);

// Note that there is a double curly braces

}

}

AppRegistry.registerComponent('Component2', () => Component2);

**(35:33) Component 2 - StyleSheet:**

/\* ./App.js \*/

import React, {Component} from 'react';

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

import Component2 from './app/components/Component\_2/Component\_2';

export default class MyApp extends Component{

render(){

return(

<View>

<Component2/>

</View>

);

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

/\* ./app/components/Component\_2/Component\_2.js \*/

import React, {Component} from 'react';

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

export default class Component2 extends Component{

render(){

return(

<View>

<View style={styles.myView}>

<Text style={styles.myText}>Hello David.</Text>

</View>

<View style={styles.container}>

<View style={styles.v1}>

<Text>View 1</Text>

</View>

<View style={styles.v2}>

<Text>View 2</Text>

</View>

<View style={styles.v3}>

<Text>View 3</Text>

</View>

</View>

</View>

);

}

}

const styles = StyleSheet.create({

myView: {

backgroundColor: 'blue'

// It is "background-color" in CSS (Cascading Style Sheets)

},

myText: {

color: 'white'

},

container: {

flexDirection: 'row',

height: 100

},

v1: {

flex: 1,

backgroundColor: 'red',

padding: 10

},

v2: {

flex: 1,

backgroundColor: 'green',

padding: 10

},

v3: {

flex: 1,

backgroundColor: 'grey',

padding: 10,

// "color: 'white'" does not work

}

});

AppRegistry.registerComponent('Component2', () => Component2);

**(43:06) Touchable Highlight:**

One can put touchable highlight to view, text or button

/\* ./app/components/Component\_2/Component\_2.js \*/

import React, {Component} from 'react';

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

export default class Component2 extends Component{

onPress(){ // New

console.log("Area pressed.");

}

render(){

return(

<View>

<View style={styles.myView}>

<Text style={styles.myText}>Hello David.</Text>

</View>

<View style={styles.container}>

<TouchableHighlight

style={styles.v1}

onPress={this.onPress}

underlayColor="grey">

<View>

<Text>View 1</Text>

</View>

</TouchableHighlight>

<View style={styles.v2}>

<Text>View 2</Text>

</View>

<View style={styles.v3}>

<Text>View 3</Text>

</View>

</View>

</View>

);

}

}

const styles = StyleSheet.create({

myView: {

backgroundColor: 'blue'

},

myText: {

color: 'white'

},

container: {

flexDirection: 'row',

height: 100

},

v1: {

flex: 1,

backgroundColor: 'red',

padding: 10

},

v2: {

flex: 1,

backgroundColor: 'green',

padding: 10

},

v3: {

flex: 1,

backgroundColor: 'grey',

padding: 10,

}

});

AppRegistry.registerComponent('Component2', () => Component2);

**(47:12) Touchable Opacity:**

/\* ./app/components/Component\_2/Component\_2.js \*/

import React, {Component} from 'react';

import {

AppRegistry,

StyleSheet,

Text,

TouchableHighlight,

TouchableOpacity,

View

} from 'react-native';

export default class Component2 extends Component{

onPress1(){

console.log("Area 1 pressed.");

}

onPress2(){ // New

console.log("Area 2 pressed.")

}

render(){

return(

<View>

<View style={styles.myView}>

<Text style={styles.myText}>Hello David.</Text>

</View>

<View style={styles.container}>

<TouchableHighlight

style={styles.v1}

onPress={this.onPress1}

underlayColor="grey">

<View>

<Text>View 1</Text>

</View>

</TouchableHighlight>

<TouchableOpacity

onPress={this.onPress2}

style={styles.v2}>

<View>

<Text>View 2</Text>

</View>

</TouchableOpacity>

<View style={styles.v3}>

<Text>View 3</Text>

</View>

</View>

</View>

);

}

}

const styles = StyleSheet.create({

myView: {

backgroundColor: 'blue'

},

myText: {

color: 'white'

},

container: {

flexDirection: 'row',

height: 100

},

v1: {

flex: 1,

backgroundColor: 'red',

padding: 10

},

v2: {

flex: 1,

backgroundColor: 'green',

padding: 10

},

v3: {

flex: 1,

backgroundColor: 'grey',

padding: 10,

}

});

AppRegistry.registerComponent('Component2', () => Component2);

**(49:49) Component 3 - TextInput:**

/\* ./App.js \*/

import React, {Component} from 'react';

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

import Component3 from './app/components/Component\_3/Component\_3';

export default class MyApp extends Component{

render(){

return(

<View>

<Component3/>

</View>

);

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

/\* ./app/components/Component\_3/Component\_3.js \*/

import React, {Component} from 'react';

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

export default class Component3 extends Component{

constructor(){

super();

this.state = {

textValue: 'Hello',

}

}

onChangeText(value){

this.setState({

textValue:value

});

// The text under the text input changes accordingly

}

onSubmit(){

console.log('Input submitted.')

}

render(){

return(

<View>

<TextInput

placeholder="Enter text:"

onChangeText={(value) => this.onChangeText(value)}

onSubmitEditing={this.onSubmit}

value={this.state.textValue}/>

<Text>{this.state.textValue}</Text>

</View>

);

// Other than "onChangeText" and "onSubmitEditing",

// We also have "onBlur", "onFocus", etc.

}

}

AppRegistry.registerComponent('Component3', () => Component3);

**(57:07) Component 3 - Switch:**

/\* ./app/components/Component\_3/Component\_3.js \*/

import React, {Component} from 'react';

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

export default class Component3 extends Component{

constructor(){

super();

this.state = {

textValue: 'Hello',

switchValue: false // New

}

}

onChangeText(value){

this.setState({

textValue:value

});

}

onSubmit(){

console.log('Input submitted.')

}

onSwitchChange(value){

this.setState({

switchValue:value

});

}

render(){

return(

<View>

<TextInput

placeholder="Enter text:"

onChangeText={(value) => this.onChangeText(value)}

onSubmitEditing={this.onSubmit}

value={this.state.textValue}/>

<Text>{this.state.textValue}</Text>

<Switch

value={this.state.switchValue}

onValueChange={(value) => this.onSwitchChange(value)}/>

</View>

);

// The switch is newly added

}

}

AppRegistry.registerComponent('Component3', () => Component3);

**(1:00:37) Component 4 - ListView:**

/\* ./App.js \*/

import React, {Component} from 'react';

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

import Component4 from './app/components/Component\_4/Component\_4';

export default class MyApp extends Component{

render(){

return(

<View>

<Component4/>

</View>

);

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

/\* ./app/components/Component\_4/Component\_4.js \*/

import React, {Component} from 'react';

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

const users = [

{name: 'John Doe'},

{name: 'Brad Traversy'},

{name: 'Steve Smith'},

{name: 'Janet Williams'}

]

export default class Coomponent4 extends Component{

constructor(){

super();

const ds = new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2});

this.state = {

userDataSource: ds.cloneWithRows(users),

};

}

renderRow(user, sectionId, rowId, highlightRow){

// There are some extra parameters

return(

<View style={styles.row}>

<Text style={styles.rowText}>{user.name}</Text>

</View>

);

}

render(){

return(

<View>

<ListView

dataSource={this.state.userDataSource}

renderRow={this.renderRow.bind(this)}/>

</View>

);

}

}

const styles = StyleSheet.create({

row: {

flexDirection: 'row',

justifyContent: 'center',

padding: 10,

backgroundColor: '#f4f4f4',

marginBottom: 3,

},

rowText: {

flex: 1

}

});

AppRegistry.registerComponent('Coomponent4', () => Coomponent4);

**(1:09:50) Component 5 - Get Data from API:**

/\* ./App.js \*/

import React, {Component} from 'react';

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

import Component5 from './app/components/Component\_5/Component\_5';

export default class MyApp extends Component{

render(){

return(

<View>

<Component5/>

</View>

);

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

/\* ./app/components/Component\_5/Component\_5.js \*/

import React, {Component} from 'react';

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

export default class Component5 extends Component{

constructor(){

super();

const ds = new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2});

this.state = {

userDataSource: ds,

};

}

componentDidMount(){

this.fetchUsers();

}

fetchUsers(){

fetch('https://jsonplaceholder.typicode.com/users')

.then((response) => response.json())

.then((response) => {

this.setState({

userDataSource: this.state.userDataSource.cloneWithRows(response)

})

})

}

renderRow(user, sectionId, rowId, highlightRow){

return(

<View style={styles.row}>

<Text style={styles.rowText}>{user.name}: {user.email}</Text>

</View>

);

}

render(){

return(

<ListView

dataSource={this.state.userDataSource}

renderRow={this.renderRow.bind(this)}/>

);

}

}

const styles = StyleSheet.create({

row: {

flexDirection: 'row',

justifyContent: 'center',

padding: 10,

backgroundColor: '#f4f4f4',

marginBottom: 3,

},

rowText: {

flex: 1

}

});

AppRegistry.registerComponent('Component5', () => Component5);

**(1:16:37) Component 6 - Navigator:**

/\* ./App.js \*/

import React, {Component} from 'react';

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

import {Navigator} from 'react-native-deprecated-custom-components';

// Navigator is currently deprecated...

import Component5 from './app/components/Component\_5/Component\_5';

import Component6 from './app/components/Component\_6/Component\_6';

export default class MyApp extends Component{

renderScene(route, navigator){

switch(route.id){

case 'component5':

return(<Component5 navigator={navigator} title='component5'/>)

case 'component6':

return(<Component6 navigator={navigator} title='component6'

user={route.user}/>)

}

}

render(){

return(

<Navigator

initialRoute={{id: 'component5'}}

renderScene={this.renderScene}

configureScreen={(route, routeStack) =>

Navigator.SceneConfigs.FloatFromBottom}/>

);

// We want to be able to tap on the item in component 5 and

// open up the details page

}

}

AppRegistry.registerComponent('MyApp', () => MyApp);

/\* ./app/components/Component\_5/Component\_5.js \*/

import React, {Component} from 'react';

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

export default class Component5 extends Component{

constructor(){

super();

const ds = new ListView.DataSource({rowHasChanged: (r1, r2) => r1 !== r2});

this.state = {

userDataSource: ds,

};

}

componentDidMount(){

this.fetchUsers();

}

fetchUsers(){

fetch('https://jsonplaceholder.typicode.com/users')

.then((response) => response.json())

.then((response) => {

this.setState({

userDataSource: this.state.userDataSource.cloneWithRows(response)

})

})

}

onPress(user){

// console.log(user);

this.props.navigator.push({

id: 'component6',

user: user

});

}

renderRow(user, sectionId, rowId, highlightRow){

// There are some extra parameters

return(

<TouchableHighlight onPress={() => {this.onPress(user)}}>

<View style={styles.row}>

<Text style={styles.rowText}>{user.name}: {user.email}</Text>

</View>

</TouchableHighlight>

);

}

render(){

return(

<ListView

dataSource={this.state.userDataSource}

renderRow={this.renderRow.bind(this)}/>

);

}

}

const styles = StyleSheet.create({

row: {

flexDirection: 'row',

justifyContent: 'center',

padding: 10,

backgroundColor: '#f4f4f4',

marginBottom: 3,

},

rowText: {

flex: 1

}

});

AppRegistry.registerComponent('Component5', () => Component5);

/\* ./app/components/Component\_6/Component\_6.js \*/

import React, {Component} from 'react';

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

export default class Component6 extends Component{

constructor(props){

super(props);

this.state = {

name: this.props.user.name,

email: this.props.user.email

}

}

onPress(){

this.props.navigator.push({

id: 'component5'

})

}

render(){

return(

<View>

<Text>{this.state.name}</Text>

<Text>{this.state.email}</Text>

<Button

onPress={this.onPress.bind(this)}

title="Go Back"/>

</View>

);

}

}

AppRegistry.registerComponent('Component6', () => Component6);