Main Points/Key Points	Notes
	Introducing React Native 1. **What is React Native?** a. It is a framework to build native mobile application in JavaScript by using React JavaScript library. b. It is open sourced by Facebook and backed by large community. c. Mobile apps developed using React Native are Instagram, Uber, Walmart, Amazon, and Soundcloud. d. React Native is half-hybrid and half-native. e. It is built using JavaScript and JSX. 2. **Native vs. Without Native Code** i. **Native** 1. React Native code is converted to the native language of each platform (`Android–Java, iOS–Objective-C`). Refer to the created `android` and `ios` folders. 2. Developer would be able to continue React Native project in Native Language of each platform (`Android – Android Studio, iOS – Xcode`) 3. Developer would be able to configure Native settings such as App Icon, Splash Screen, App deployment to the actual device. 4. Developer would be able to utilize a large number of supported and new APIs.
	Summary

Main Points/Key Points	Notes
	Introducing React Native ii. **Without Native Code** 1. React Native code is retained in its original form for rendering. 2. Developer is required to use the defined APIs set by Expo (Limited). 3. Not all UI toolkits support Expo. 4. Developer is required to install an Expo client app to render the developed react native code. 5. Developer would be able to use **`snack.expo.io`** online code editor. 3. **React Native Development Tools.** **Required Software** 1. Node.js (version 8). 2. Python2. 3. Yarn. 4. JDK 1.8 or latest. **Native Code** 1. Android Studio (Android SDK). 2. Xcode (iOS SDK). **Without Native Code** a. Expo App (expo.io).
	Summary

Main Points/Key Points	Notes
	Introducing React Native a. Install react-native-cli through node.js command prompt. **Native Code** **1. Windows** npm install –g react-native-cli **2. Mac** sudo npm install –g react-native-cli **Without Native Code** **1. Windows** npm install –g expo-cli **2. Mac** sudo npm install –g expo-cli b. Set Android SDK Manager to Android 6.0 Marshmallow or higher. I. Access Android SDK Manager through Android Studio. II. Make sure it is updated and agreed to the licenses provided to avoid warning. c. Configure Android Virtual Device (AVD) running Android 6.0 and above. I. Access Android Virtual Device (AVD) through Android Studio. II. Select any device definition with pre-default configuration and select Android 6.0 and above.
	Summary

Main Points/Key Points	Notes
	Introducing React Native d. Set ANDROID_HOME in Windows Environment Variables and include in the system path. I. Create a new system variable ANDROID_HOME is pointing to **ANDROID SDK** directory. a. `C:\Users\<login>\AppData\Local\Android\Sdk` II. Windows System Path must include the following **ANDROID SDK** subfolders: a. `%ANDROID_HOME%/tools` b. `%ANDROID_HOME%/tools/bin` c. `%ANDROID_HOME%/platform-tools` e. Set ANDROID_HOME in Mac OS Bash Profile I. `sudo nano ~/.bash_profile` II. Add the following environment variables: a. `export ANDROID_HOME=/Users/<login>/Library/Android/sdk` b. `export PATH=$PATH:$ANDROID_HOME/platform-tools` c. `export PATH=$PATH:$ANDROID_HOME/tools` d. `export PATH=$PATH:$ANDROID_HOME/tools/bin` III. `source ~/.bash_profile` f. Emulator (Android) or Simulator (iOS) Testing by executing the following commands: I. Windows - `emulator -avd Pixel-2-XL` (Android) II. Mac OS - `./emulator -avd Pixel-2-XL` (Android) III. Mac OS - `open -a simulator` (iOS)
	Summary

Main Points/Key Points	Notes

	Introducing React Native
	
	4. **React Native Project Development.**
	
	Native Code
	1. Windows & Mac
	`react-native init HelloWorld`
	
	Without Native Code
	1. Windows & Mac
	`expo init HelloWorld`
	
	a. The created React Native project located inside home folder.
	
	Windows
	`C:/Users/<windows login>/HelloWorld`
	
	Mac
	`Macintosh HD/Users/<mac login>/HelloWorld`
	
	b. Modify **App.js** file created inside **HelloWorld** folder using **Microsoft Visual Studio Code**. For Without Native Code approach you can use http://snack.expo.io online code editor.

	Summary

Main Points/Key Points	Notes
	Introducing React Native c. To deploy or run the app, execute the following commands: **Native Code** **1. Windows** cd HelloWorld react-native run-android **2. Mac** cd HelloWorld react-native run-android or react-native run-ios **Without Native Code** **1. Windows & Mac** cd HelloWorld npm start or yarn start or expo start
	Summary

Main Points/Key Points	Notes

	Introducing React Native
	
	5. **A Basic React Native Component**.
	a. Components are the building blocks of React Native mobile application. There are exported classes for JavaScript.
	
	b. There are two types of components:
	
	Stateful
	`class Greeting extends Component { constructor(){ super(); this.state = { name: 'Chris' }; } render(){ return( <SomeComponent /> ); } }`
	
	Stateless
	`const Greeting = () => ( <SomeComponent /> );`

	Summary

Main Points/Key Points	Notes
	Introducing React Native 6. **A Flow of a Basic React Native Component**.

React Native Component

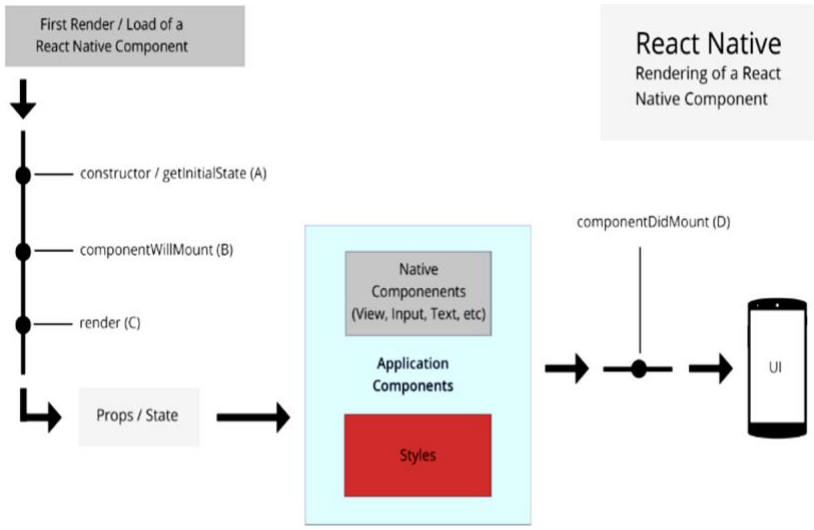
```
import React, {Component} from 'react';
import { View, Text, StyleSheet } from 'react-
native';

class HelloWorld extends Component{

constructor () { // A
super()
 this.state = {
 name: 'React Native in Action'
 }
}

componentDidMount () { // C
console.log('mounted..')
}

render () { // B
return (
 <View style={styles.container}>
 <Text>{this.state.name}</Text>
 </View>
 )
 }
}

const styles = StyleSheet.create({
container: {
marginTop: 100,
flex: 1 }
})
```

Summary

Main Points/Key Points	Notes
	Introducing React Native 7. **Code Explanation**.

View
The fundamental block for building React Native Application especially for the UI. (* .. *)
Text
The text element for React Native Application (..).
StyleSheet
The styling element for React Native Application. Similar to CSS.
A
1. Set the state of the object with a variable ***name*** in the constructor. 2. Dynamic data must set as a state or props.
B
It will examine props and state, later combine components, styles and data to be rendered as the UI.
C
ComponentDidMount is the last lifecycle of a React Native Class. Any APIs to be called will be placed in here.

Summary

Main Points/Key Points	Notes
	Introducing React Native 8. **A Flow Diagram**.  *Figure 1. Rendering a Basic React Native Class.* Dabit, N. (2018). *React Native in Action*. New York, NY: Manning Publications Co.
	Summary

Main Points/Key Points	Notes

Introducing React Native

9. **JSX vs. HTML**

JSX	HTML
Text Component	
`<Text>Hello</Text>`	`<span>Hello</span>`
View Component	

`<View>`
`<Text>Hello 2</Text>`
`</View>`
	`<div>`
	`<span>Hello 2</span>`
	`</div>`

Touchable highlight	

`<TouchableHighlight>`
`<Text>Hello 2 </Text>`
`</TouchableHighlight>`
	`<button>`
	`<span>Hello 2</span>`
	`</button>`

10. **References**

a. Dabit, N. (2018). *React Native in Action*. New York, NY: Manning Publications Co.

b. Abernathy, C. (2018). *React Native Tutorial: Building Android Apps with JavaScript*. Retrieved from https://www.raywenderlich.com/247-react-native-tutorial-building-android-apps-with-javascript

c. Facebook Inc. React Native Reference, version 0.57. Available at https://facebook.github.io/react-native/

Summary