**React Native**

React js is a library which is used to create a user interfaces. But its actually the react-dom which adds the web support. React just gives you tools for managing state, for building virtual component trees, but then we need extra library like react-dom for translating the result react produced to an actual platform like the browser.

React Native is alternative for React-dom. React Native gives a collection of special react components which we can use in our JSX code. So, React Native ships with built-in components you can use and those components are then compiled to native UI elements for the IOS and Android Platform. React Native takes care about this compilation step.

React Native is like React-Dom it just doesnt target the web, the browser as platform but instead IOS and Android. React Native gives the all components and the API's to interact with those platforms and to build apps for those platforms.

React Native provides some special components such as <View> & <Text> & <TextInput> etc. This elements are compiled by react native into underlying platforms like android and ios. Javascript code such as functions are not compiled by React Native, instead a special thread is being runned in which javascript code executes.

Expo CLI and React Native CLI are two tools which are used to create React Native projects and run React Native apps on testing devices and simulators as well as build React Native apps so that we can ship them to the app stores.

For reference <https://github.com/academind/react-native-practical-guide-code>

Official: <https://reactnative.dev/docs/components-and-apis>

Commands:

|  |  |
| --- | --- |
| Commands | Info about command |
| expo init "project name " | This command is used to create. |
| npm install @expo/ngrok@^4.1.0 | This make tunnel works |
| npm install @react-navigation/native | This make react-navigation possible. |
| expo install react-native-screens react-native-safe-area-context | Extra dependencies for react-navigation |
| npm install @react-navigation/native-stack | It is used as a history stack of navigating screens, with this we get same features of a browser and also animated effects. |
| npm install @react-navigation/drawer | For drawer navigation |
| expo install react-native-gesture-handler react-native-reanimated | Extra dependencies for drawer navigation. |
| npm install react-native-reanimated@1 --save --save-exact | Bug fixing while using drawer navigation. |

Step to steup app in mobile device:

1. Download expo app in your mobile device.

2. Open the http://localhost:19002 in the browser.

3. npm install @expo/ngrok@^4.1.0 , install this dependency and re-run the application.

4. Select tunnel on the browser and scan the QR code in expo app.

Steps to setup the simulator development:

1. Start the react code with npm start.

2. Open android-studio and create a new virtual android screen which contains playstore symbol. i.e. Pixal 3a API 32

3. go to /Users/vishalpalla/Library/Android/sdk/platform-tools and execute command ./adb kill-server

<https://stackoverflow.com/questions/23415746/how-to-resolve-the-adb-server-didnt-ack-error>

Note:

1. Like CSS and in react-web, in React Native child elements wont inherit the styles from parent element.

2. Some styles wont show any effect on IOS platform beacuse of underlying elements in IOS may wont support the styles. Like border radius wont support on TEXT element in React native, in that cases wrap <Text/> around the <View> and try.

**Basic Components:**

**View:** View is like div in the html, front-end web apps. View is used to wrap the components. By default View occupies the space which is required i.e. if you wrap a button inside a View, the View will takes the space based on the text of the button. If you want that button occupy all the available space use property flex:1 .

**Text**: Text is a built in react native component which is used to display the simple text. Border Radius Property dont work on Text, so wrap Text with View.

**TextInput**: TextInput is a built in react native component which is used to input the details. It is like a input tag in html.

**ScrollView**: ScrollView is used to give scroll effect to mobile. To use it, wrap your list around the ScrollView. But the height and area that scrollView uses decides by the parent of ScrollView, so wrap scrollView around <View> and give your styles on View.

<ScrollView>

{courseGoal.map((g, index) => (

<View style={styles.goalItem} key={index}>

<Text style={styles.goalText}>{g}</Text>

</View>

))}

</ScrollView>

**FlatList**: FlatList is replacement for ScrollView, flatList render only some specific which can be visible for screen length. So it improves performance. For unique ness we need to give key, for that we have two solutions.

1.First Solution: On the data i.e courseGoal list, on the each item we can add key property and give the unique value.

2. Second Solution: On the FlatList we have keyExtractor which takes a function as argument, the function receives the item, so we can return id or key which is unique.

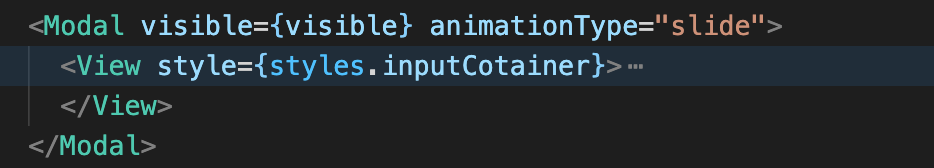
Note: Don't wrap FlatList around View, if you do then scrolling wont work.



**Pressable**: Pressable is built in react-native component which is used to handle onClick type event in react-native. It can be used like in the below image.



**Modal**: In Rect-Native we have a built in react-native modal to use. It takes a visible prop on which we need to pass a boolean to open and close.



**Dimensions:** Dimensions is a built in object from react-native used to set properties of css conditionally to adapt the different screen widths.

**useWindowDimensions:** This is a built in hook used to provide the dynamic screen information. Unlike Dimensions API, i wont lock the property details.

**Platform**: PlatForm is used to decide which things you wanna apply based on OS.



**Important Points**:

1. If a child <View> </View> we gave flex:1, then we need to make sure the Parent (in this case Pressable) actually takes space, otherwise we cant see anything. The verdict is flex:1 takes space from parent element. So make sure the parent element takes space.

<Pressable style={styles.button}>

<View style={styles.innerContainer}>

<Text>{title}</Text>

</View>

</Pressable>

2. useEffect runs after the component is mounted. But in some case we need a hook which has to execute when component is mounting. In that case useLayoutEffect can be used.

3. When using Image from react-native and getting image from url/internet height should be set manually because react-native dont know about height. If you dont set the height image wont be visible.

4. For Text we cant have underline so wrap it around View and give underline to View.