**React** – This is a framework for implementing web and mobile application using JavaScript.

**Native** – We can use native iOS and android components, which are controlled by JavaScript.

**Pros**:

* Code sharing
* Real time code update

**Cons**:

* Single thread application

**Setup**:

1. Install Homebrew – Use to install node and watchman
   * /usr/bin/ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"
2. Install node
   * brew install node
3. Install Watchman – Used for watching file system changes
   * brew install watchman
4. Install react native
   * npm install -g react-native-cli
5. Install XCode and android studio for app development
6. Create app using following command:
   * react-native init <app name>
7. Run React native packager
   * cd <app name>
   * react-native start
8. Run simulator:
   * react-native run-ios
   * react-native run-android
9. Create iOS build using: Product -> Archive
10. Create android build using command:
    * cd android && ./gradlew assembleRele

**State Vs Props:**

There are two types of data that control a component.

* Props: props are set by the parent and they are fixed throughout the lifetime of a component.
* State: For data that is going to change, we have to use state.

**Styling:**

*style* props is used for styling a component. *StyleSheet.create* is sued to create all style at a place.

const styles = StyleSheet.create({});

**FlexBox:**

To accommodate different screen sizes, React Native offers Flexbox support. To achieve the desired layout, flexbox offers three main properties:

* *flexDirection*: Used to align element vertical or horizontal
* *justifyContent*: Used to determine elements distribution inside a container along the primary axis
* *alignItems*: Used to determine elements distribution inside a container along the secondary axis

**Height and width:**

* *width*: Set component width
* *height*: Set component height
* *flex*: space is divided all among views according value.

**Components:**

* View, ScrollView
* TextInput
* Button, TouchableHighlight
* FlatList, SectionList, ListView
* Text
* Image

**Debugging:**

You can open the developer menu on the IOS simulator by pressing *command + D*. On Android emulator, you need to press *command + M*.

* Reload − Used for reloading simulator. You can use shortcut command + R
* Debug JS Remotely − Used for activating debugging inside browser developer  console.
* Enable Live Reload − Used for enabling live reloading whenever your code is  saved. The debugger will open at localhost:8081/debugger-ui.
* Systrace − Used for starting Android marker based profiling tool.
* Show Inspector − Used for opening inspector where you can find info about your components. You can use shortcut command + I
* Show Perf Monitor − Perf monitor is used for keeping track of the performance of your app.