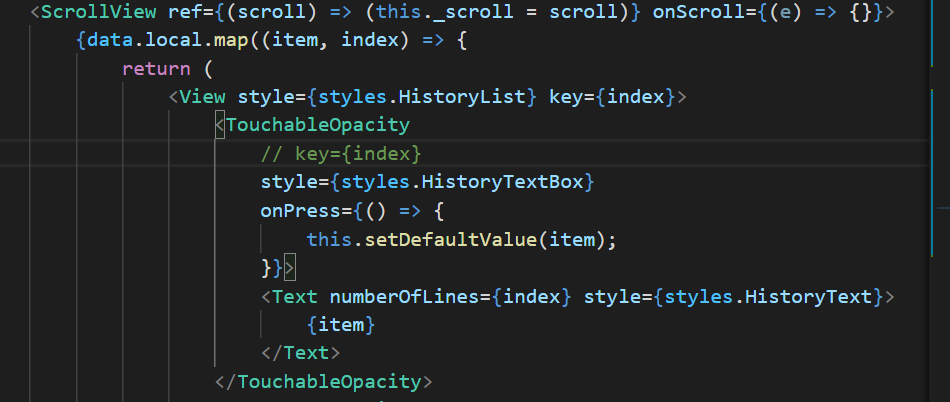
1. **propTypes为该组件props的类型，定义自己属性的类型（属性名：类型）**
2. **组件分为函数式组件和类组件，函数式组件主要用作展示UI，无状态所以不用做需要动态刷新的UI显示（用变量也可替代state，但是不能刷新UI，因其没有render函数，不能刷新UI）。类组件有状态，可用作需要数据显示的组件**
3. **使用遍历返回组件时最外层组件要加上key标记，不然会报错**
4. **React native宽高没有单位，运行后在Android和iOS上会被解析成不同的单位**
5. **使用state管理组件显示时，要注意setstate的顺序**
6. **React native父组件和子组件并不是继承关系，而是调用关系，在一个组件中调用另一个组件，被调用组件就是子组件**
7. **React native super(props)相当于执行父类的构造函数，进行绑定this等操作**
8. **React native props是组件的属性，用于接收调用组件传递过来的属性值，传递的属性值必须在被调用组件中声明过（使用时声明，不一定都是显示的属性，也可能是函数）**
9. **\*\*\* this.props.children获取组件包括起来的内容:<View><Text></Text></Text>。**
10. **\*\*\* this.refs获取使用ref标记的组件实例**
11. **\*\*\*\*\*react native提供的组件都是封装好的，可以直接用style属性，但是自定义的组件不能直接用style属性，需要自己封装或者在定义的组件内部设置样式，封装style能灵活设置，不封装只能用固定的**
12. **\*\*\*自定义的组件其实就相当于函数的return返回一个UI对象，分为类组件和函数组件（所以不能直接给子组件设置style）**
13. **子组件向父组件传值通过函数参数形式**
14. **页面跳转和使用组件控制显示的区别是：跳转是把页面的内容换成了跳转后的页面，之前的页面内容都不显示，使用组件的方法实现这种效果比较麻烦**
15. **Pubsub和redux都可解决组件通信问题，pubsub不是集中管理，redux集中管理数据**
16. **颜色rgb和rgba分清楚，一个是三个数值，一个是四个数值**
17. **布局不一定要固定使用某一种，根据需要选择布局方式**
18. **Flex: 1表示子组件等比例放大缩小**
19. **组件化开发，子组件首字母必须大写**
20. **setState({num:++this.state.num}),++必须在前**
21. **父组件给子组件赋的属性都必须是子组件定义过的，给子组件赋style属性不会设置样式，需要手动给子组件加上 style={this.props.style}**
22. **Flex布局，主轴是纵向，不设置容器高度，容器高度默认是子组件高度，宽度是父组件宽度**
23. **React native绝对定位和web端相同点：原位置都不占空间，不同点：组件相对于上一级父组件的左上角移动，而不是相对网页左上角移动**
24. **组件margin默认是0？**
25. **组件设置left、bottom属性后，原位置仍然占空间（类似相对布局），下面的元素如果设置相关属性可以占据该位置，定位解决层级问题就是这种方式**
26. **23、24、25总结：react native布局中定位问题：1、react native默认left、top等是web中的相对布局，原位置占据空间，其他组件设置定位可以占据该空间 。2、绝对定位：组件相对于父组件移动，其余和web端绝对定位一样**
27. **使用高德地图：1、注册账号，获取应用的key。 2、创建两个应用：Android应用和web服务应用。 3、下载react-native-amap-geolocation，配置Android文件。**
28. **定位出错原因：没有考虑返回数据格式是JSON，访问出错，没有仔细查看返回数据的对象属性格式**
29. **Npm全局包和本地包（项目内部的包）**
30. 项目分为开发模式（DEV/debug）和生产模式（release），\_DEV\_是react native提供的一种判断当前是哪一种模式的方式，为开发模式时是true，即使用react-native run-android运行项目时是true
31. **React native内置global对象，和nodejs的global一样，管理所有全局变量**
32. **DeviceEventEmitter是给app设置监听事件，在全局内访问，可在一个页面中触发另一个页面的监听事件，触发条件是用DeviceEventEmitter的方法，而Android端的BackHandler的触发条件是点击返回键时触发**
33. **React-redux：组件分为UI组件和容器组件，使用connect()方法把一个UI组件转换成容器组件，该组件就能使用redux API和状态，API和状态作为组建的props，通过props调用。Connect方法接收两个参数，分别是获取store和向store传递参数的方法。\*\*\*通过mapStateToProps函数获取state后存于props中，作为属性（是state不是reducer）**
34. **\*\*\*redux的reducer参数的state并不一定非要定义，组件变化传递action时并没有传入state，state可以在reducer中定义**
35. **Redux的action不只是传递的对象，而是传递对象的这个行为（函数），mapDispatchToProps就是把这个函数作为属性传递给组件**
36. **如果不设置mapDispatchToProps则会把dispatch作为属性传递给组件？**
37. **mapDispatchToProps相当于action，mapStateToProps相当于监听state（自动更新使用的state）**
38. **\*\*\*\*\*\*props更新后也会刷新UI！！！！！！！（是store原因还是props本身？）**
39. **dispatch作为mapDispatchToProps的参数，用于定义组件的action，redux的action就是在需要使用dispatch时调用store的dispatch，react-redux只是把dispatch单独作为参数传进组件中。如果组件不设置mapDispatchToProps，则会把dispatch作为组件的props？**
40. **\*\*\*\*\*\*mapStateToProps和mapDispatchToProps函数是都要作为connect函数的参数的，如果只需要mapDispatchToProps一个参数，则需要把第一个参数设置为null**
41. **生命周期：获取props、获取state、转载前、render、装载后，运行时：props或state变化、组件更新、render**
42. **Pentaxphotosync：Android模块中导入的rnfs.xxx库，是从node\_module的Android模块中导入的**