ReactNative从入门到发疯

目录

一、 前言 1

1. 作者简介 1

2. 本书初衷 1

3. 本书内容简介 1

二、 ReactNative入门必备 2

1. ReactNative介绍 2

1.1 ReactNative 2

1.2 什么是React Native？ 3

1.3 React Native开发注意事项 4

2. ReactNative安装 5

安装Genymotion 7

三、 ReactNative IOS UI/GUI开发 13

1. 你必须要会点FlexBox布局 13

2. React Native常用组件之View 20

3. React Native常用组件之Text 20

4. React Native常用组件之Image 20

5. React Native常用组件之TextInput 20

6. React Native之常用Touchable系列组件 20

四、 ReactNative开发进阶 20

五、 ReactNative 项目实战 20

六、 ReactNative 实战拓展 20

七、 ReactNative”入门到发疯”结尾 20

## 前言

### 作者简介

作者姓名:王衡杰

邮箱: hundredsir@gmail.com

博客名:香蕉大大(简书)

微信公众号:大大家的iOS说

微信号:hundreda

毕业于:广东某工业大学

一枚一直在路上的开发者,希望能用自己微小的力量给大家带来一些借鉴以及经验参考!

### 本书初衷

想了很久,一方面以前一直看大神的书成长,但是觉得感觉还是那一句,人在江湖走,出来混迟早要还的,所以我想法很简单,在如今蔚然大观的人潮中,怎么总结一套自己的资料并且供大家传阅,可能就是我今年年末最后的目标.

如果说本书的想法,不知道有没有机会录制成视频,但是鉴于如今参考资料繁多,所以我还是站在一个点,希望能和大家分享,并且通过开源的方式一起把此书做好,也希望大家能给我提点建议,给大家多一些正确的参考.

本书于2016年11月26日完结第一稿,很多不足的地方希望大家能多给点建议,再次谢谢大家拜读,也希望大家能通过鄙人短浅的认识,开始reactnative的世界!

### 本书内容简介

## ReactNative入门必备

### ReactNative介绍

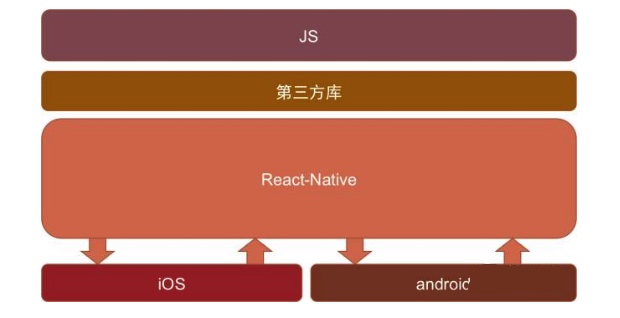
尽管在移动开发中，原生App的开发成本很高，但现阶段基于原生开发仍然是必须的，因为Web的用户体验仍无法超越Native，主要体现在：

* Native的原生控件有更好的体验；
* Native有更好的手势识别；
* Native有更合适的线程模型，尽管Web Worker可以解决一部分问题，但如图像解码、文本渲染仍无法多线程渲染，这影响了Web的流畅性。

### ReactNative

React Native于F8大会开源,在短短不到一年的时间里,它成为手机端必不可少的开发模式之一。 它充分利用了Facebook现有的业务轮子, 其核心设计理念：**既拥有Native的用户体验、又保留React的开发效率**。

目前，React Native基本完成了对多端的支持，实现了真正意义上的面向配置开发: 开发者可以灵活的使用HTML和CSS布局,使用React语法构建组件,实现：**H5, Android, iOS 多端代码的复用**。



此外，使用流畅度和原生的保持在同一层次,这不是我们梦寐以求的开发模式吗?让我们从无到有开启新的学习篇章吧!

### 什么是React Native？

**（一）时下两大移动互联网主流技术**

           BAT等一线互联网公司的插件化，热修改等技术

           React Native技术（**2016年，随着Android版本的稳定，更加火爆**）

**（二）React Native介绍**

          Facebook于2015年9月15日发布**React Native**

          广大开发者可以使用JavaScript和React开发跨平台移动应用

          React Native提倡**组件化**开发: 即提供一个个封装好的组件，组件相互嵌套形成新的组件

（**三）React Native的优势**

**3.1 跨平台开发**

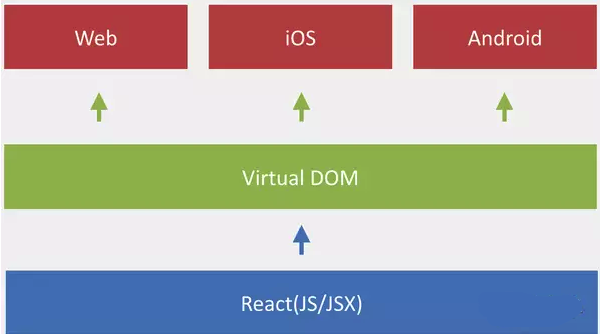
         运用React Native,我们可以使用同一份业务逻辑核心代码来创建原生应用运行在Web端，Android端和iOS端；

**3.2 追求极致的用户体验：实时热部署**

**3.3 learn once,write everywhere** （最具魅力）

        React Native不强求一份原生代码支持多个平台，所以不提“Write once, run anywhere”（Java），提出了“Learn once, write anywhere”。

**React Native开发注意事项**



### React Native开发注意事项

* + 1. 目前react native在iOS上仅支持**ios7**以上，Android仅支持**Android4.1**以上版本:

github地址:  https://github.com/facebook/react-native

       官网文档: http://facebook.github.io/react-native/docs/getting-started.html

* + 1. 由于React Native的版本更新速度很快，如果没有深厚的JavaScript基础，建议选择:

**a、功能适中，交互一般，不需要特别多的系统原生支持；**

**b、对于部分复杂的应用，可以考虑原生+React Native混合开发**

### ReactNative安装

**(一)、环境需求**

**1.1  安装Homebrew**

       Homebrew是OS X的套件(包)管理器，我们可以通过它获取并且安装很多组件

**安装方式:**

      ruby -e "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/master/install)"

**1.2  安装npm 和 Node.js**

   Node.js最好安装**4.0**及其以上更高版本，node安装成功后npm自动也就有了，直接下载安装Node.js，网址：https://nodejs.org/en/download/ （资料中已有

**1.3  安装WatchMan**

该插件用于监控bug文件和文件变化 ，并且可以触发指定的操作

**安装方式**

      brew install watchman



      注意：在Max OS X 10.11（El Capitan)版本中，homebrew在安装软件时可能会碰到/usr/local目录不可写的权限问题。可以使用下面的命令修复：

sudo chown -R `whoami` /usr/local

**1.3  安装Flow**

       flow是一个 JavaScript 的静态类型检查器，建议安装它，以方便找出代码中可能存在的类型错误

安装方式

brew install flow



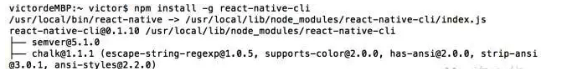
**(注意：如果提示command not found,请加上sudo获得最高权限)**

**(二)、React Native安装**

**2.1 安装React Native**

        npm install -g react-native-cli

**2.2 安装截图如下:**



**(三)、 ios开发环境需求**

Xcode 7 及其以上更高版本

**(四)、 Android开发环境需求**

**安装最新版的JDK:**

      下载安装地址：http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html （资料中已经存放）

      可以安装Android studio省略以下步骤，**建议安装Android studio**。

### ****安装Genymotion****

       Genymotion是一个第三方模拟器，它比Google官方的模拟器更易设置且性能更好。但是，它只针对个人用户免费。

1）下载并安装Genymotion

      https://www.genymotion.com/

2）打开Genymotion,如果你尚未安装VirtualBox,它有可能会提示你安装

3）创建一个模拟器并启动

4）按下⌘+M可以打开开发者菜单(在安装并启动了React Native应用之后)

**(五)、 React Native的第一个应用**

**5.1 执行命令,生成一个工程**

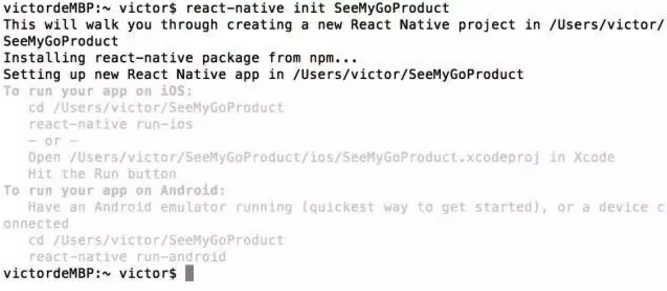
     react-native init 项目名称

      注意:由于众所周知的网络原因，需要等待一段时间（具体视网络情况而定）。react-native命令行从npm官方源拖代码时会遇上麻烦，可以将npm仓库源替换为国内镜像：

     npm config set registry https://registry.npm.taobao.org

     npm config set disturl https://npm.taobao.org/dist

**5.2  运行截图**



**5.3 目录结构截图**

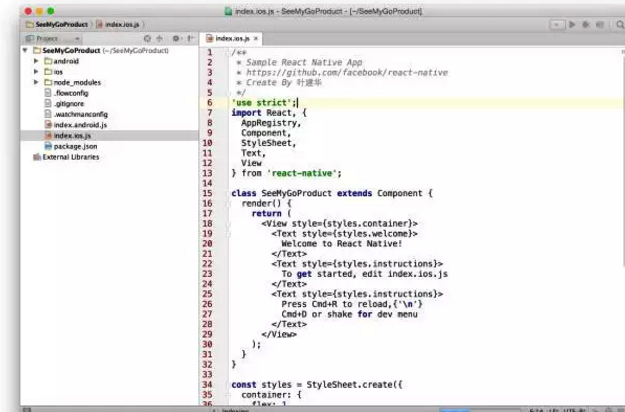
**目录结构分析:**

a）默认生成**android**和**ios**两个平台的原生项目；

b）其中，**index.android.js**和**index.ios.js**文件为Android和iOS的空壳应用文件；

c）此外，**node\_modules**文件夹，是为**Node.js**存放和管理npm包资源，也包含React Native框架文件。

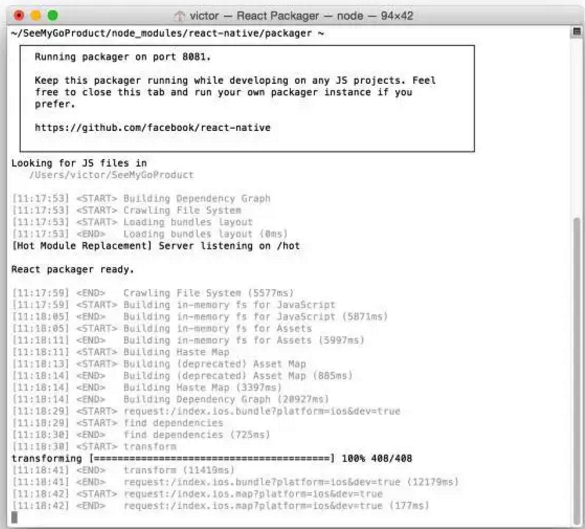
**查看index.ios.js中的代码:**



**(六)、运行工程文件**

      不管是 iOS 还是 Android，在开发调试阶段，都需要在 Mac 上启动一个 HTTP 服务，称为**Debug Server**，默认运行在 **8081** 端口，APP 通 Debug Server 加载 js。

**6.1 打开Xcode，运行你的第一个React Native创建的iOS应用**



初始化成功



启动页面

**6.2 把React Native创建的应用跑在Android上**

a) 命令行执行cd SeeMyGoProduct,路径切换到项目主目录

b) 命令行执行react-native run-android进行加载运行android 应用。

c) 使用编辑器进行打开和修改index.android.js文件，接着通过菜单按钮选择Reload JS来进行刷新修改

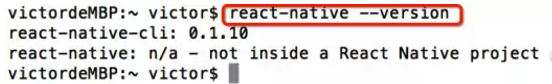
**(七)、管理React Native库的版本**

       在开发中，会经常的去控制React Native的版本库，得以适配各种条件下的开发，那该如何查看、控制ReactNative的版本呢？

**7.1 查看本地的React Native的版本**

  命令行输入

   react-native –version



**7.2 更新本地的React Native的版本**

    命令行输入

    npm update -g react-native-cli

**7.3 查询react-native的npm包最新版本**

NPM的全称是Node Package Manager ，是一个NodeJS包管理和分发工具，已经成为了非官方的发布Node模块（包）的标准。

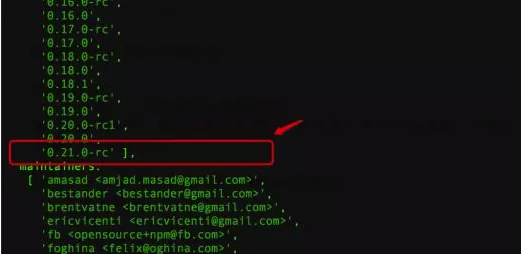
    npm包地址 ：

     https://www.npmjs.com/package/react-native

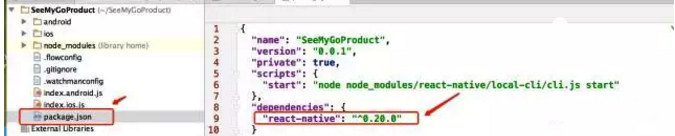
    命令行查询

      npm info react-native

    查询效果



 项目中查看



**7.4 升级或者降级npm包的版本**

    npm install --save react-native@0.18

**7.5 更新项目templates文件（可选）**

       新的npm包会包含更新在运行react-native init命令生成的一些动态文件，例如init创建项目的时候会生成iOS和Android的子项目，我们可以通过以下的命令进行获取最新的代码

     命令行查询

      react-native upgrade

**(八)、WebStom设置React Native代码提示**

**8.1  从gitHub上下载xml插件**

 git clone https://github.com/virtoolswebplayer/ReactNative-LiveTemplate

**8.2  安装**

将ReactNative.xml复制到 ~/Library/Preferences/WebStorm11/templates ，然后重启 WebStrom

请注意,webStorm中可能没有templates 安全起见就自己创建一个templates 文件夹就可以

## ReactNative IOS UI/GUI开发

### 你必须要会点FlexBox布局

在H5中，界面的搭建都是采用CSS的布局，基于**盒子模型**，依赖 **display属性 ， position属性， float属性**。但对于那些特殊布局非常不方便，比如，垂直居中。

一种全新的针对web和移动开发布局应运而生：**Flex布局**，可以简便、完整、响应式地实现各种页面布局。目前，它已经得到了所有浏览器的支持，这意味着，现在就能很安全地使用这项功能。



#### 一、FlexBox布局

**1.1 FlexBox是什么意思呢？**

**flexible**（形容词）：能够伸缩或者很容易变化，以适应外界条件的变化

**box**（名词）：通用的矩形容器

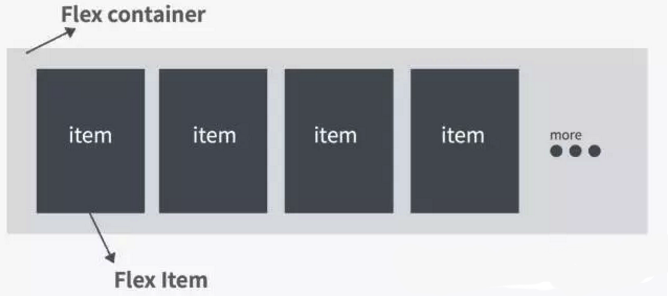
**1.2  什么是FlexBox布局?**

弹性盒模型（The Flexible Box Module）,又叫Flexbox，意为“弹性布局”，旨在通过弹性的方式来对齐和分布容器中内容的空间，使其能适应不同屏幕，为盒装模型提供最大的灵活性。

Flex布局主要思想是：让容器有能力让其子项目能够改变其宽度、高度（甚至是顺序），以最佳方式填充可用空间；

React native中的FlexBox是这个规范的一个**子集**。

**1.3大部分情况下是处理图中FlexItem在FlexContainer中的位置和尺寸关系**



**二、Flexbox在开发中的应用场景**

**2.1 Flexbox在布局中能够解决什么问题？**

     浮动布局

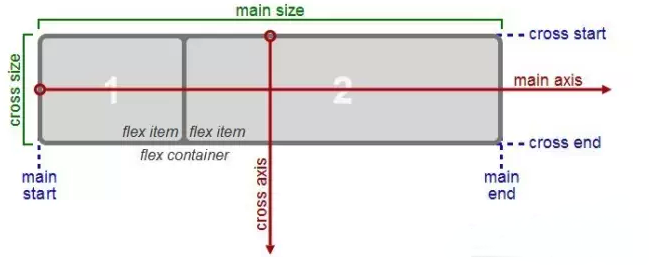
     各种机型屏幕的适配

     水平和垂直居中

     自动分配宽度

     ......

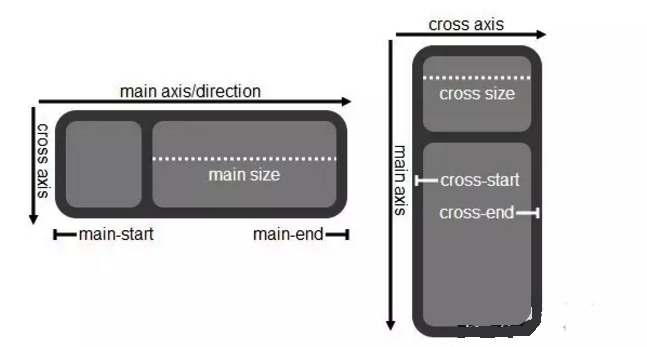
**2.2在CSS中，常规的布局是基于块和内联流方向，而Flex布局是基于flex-flow流,下图很好解释了Flex布局的思想：**



容器默认存在两根轴：**水平的主轴（main axis）**和**垂直的交叉轴（cross axis）**。主轴的开始位置（与边框的交叉点）叫做main start，结束位置叫做main end；交叉轴的开始位置叫做cross start，结束位置叫做cross end。

       项目默认沿主轴排列，单个项目占据的主轴空间叫做main size，占据的交叉轴空间叫做cross size。

* 1. **根据伸缩项目排列方式的不同，主轴和侧轴方向也有所变化：**



#### 三、Flexbox的常用属性

**3.1 容器属性**

**a)  flexDirection: `row | row-reverse | column | column-reverse`**

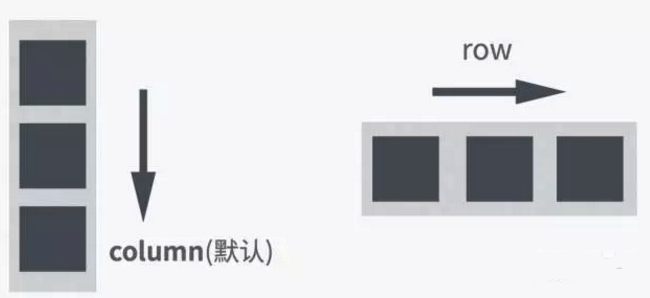
     该属性决定主轴的方向（即项目的排列方向）。

     row：主轴为水平方向，起点在左端。

     row-reverse：主轴为水平方向，起点在右端。

    column(默认值)：主轴为垂直方向，起点在上沿。

    column-reverse：主轴为垂直方向，起点在下沿。



**b）justifyContent:`flex-start | flex-end | center | space-between | space-around`**

      定义了伸缩项目在主轴线的对齐方式

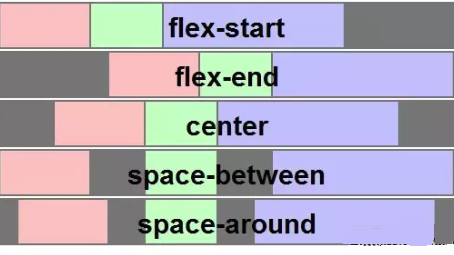
      flex-start(默认值)：伸缩项目向一行的起始位置靠齐。

      flex-end：伸缩项目向一行的结束位置靠齐。

      center：伸缩项目向一行的中间位置靠齐。

      space-between：两端对齐，项目之间的间隔都相等。

      space-around：伸缩项目会平均地分布在行里，两端保留一半的空间。



**c） alignItems:  `flex-start | flex-end | center | baseline | stretch`**

       定义项目在交叉轴上如何对齐，可以把其想像成侧轴（垂直于主轴）的“对齐方式”。

      flex-start：交叉轴的起点对齐。

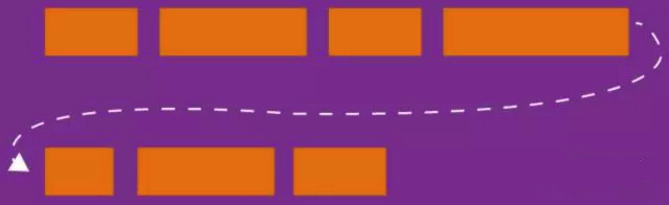
      flex-end：交叉轴的终点对齐 。

      center：交叉轴的中点对齐。

      baseline：项目的第一行文字的基线对齐。

      stretch（默认值）：如果项目未设置高度或设为auto，将占满整个容器的高度。 **d)  flexWrap: `nowrap | wrap | wrap-reverse`**

       默认情况下，项目都排在一条线（又称"轴线"）上。flex-wrap属性定义，如果一条轴线排不下，如何换行。



nowrap(默认值)：不换行。



wrap：换行，第一行在上方。



wrap-reverse：换行，第一行在下方。（和wrap相反）



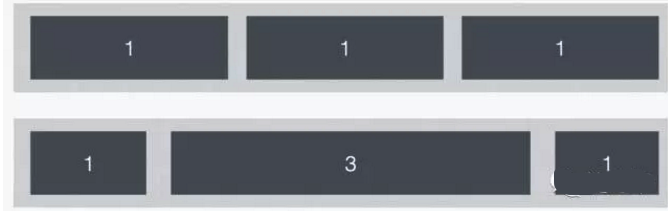
**3.2 元素属性**

**a）flex**

       “flex-grow”、“flex-shrink”和“flex-basis”三个属性的缩写， 其中第二个和第三个参数（flex-shrink、flex-basis）是可选参数。

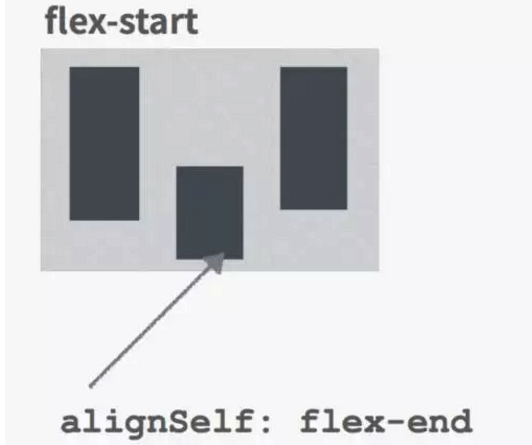
   默认值为“0 1 auto”。

   宽度 ＝ 弹性宽度 \* ( flexGrow / sum( flexGorw ) )



**b）alignSelf:  “auto | flex-start | flex-end | center | baseline | stretch”**

align-self属性允许单个项目有与其他项目不一样的对齐方式，可覆盖align-items属性。默认值为auto，表示继承父元素的align-items属性，如果没有父元素，则等同于stretch。



#### 四、在React Native中使用Flexbox

**4.1  获取当前屏幕的宽度、高度、分辨率**

### React Native常用组件之View

### React Native常用组件之Text

### React Native常用组件之Image

### React Native常用组件之TextInput

### React Native之常用Touchable系列组件

       在前面的登录界面中，我们发现所有的组件不会对用户的点击、触摸、拖拽做出合适的响应，这是十分不友好的。那么，在React Native中如何让视图对触发做出合适的响应呢？

**一、高亮触摸**  **TouchableHighlight**

      当手指点击按下的时候，该视图的不透明度会进行降低同时会看到相应的颜色，其实现原理则是在底层新添加了一个View。此外，TouchableHighlight只能进行一层嵌套，不能多层嵌套。

**常用属性：**

**activeOpacity** number

设置组件在进行触摸的时候，显示的不透明度(取值在0-1之间)

**onHideUnderlay**  function  方法

 当底层被隐藏的时候调用

**onShowUnderlay**  function 方法

 当底层显示的时候调用

**style**

可以设置控件的风格演示，该风格演示可以参考View组件的style

**underlayColor**

 当触摸或者点击控件的时候显示出的颜色

代码演示：



**二、不透明触摸**  **TouchableOpacity**

     该组件封装了响应触摸事件；当点击按下的时候，该组件的透明度会降低。

**常用属性：**

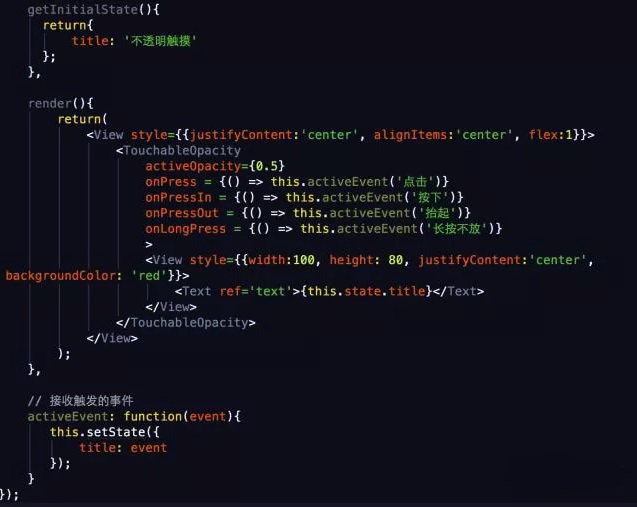
**activeOpacity  number**

      设置当用户触摸的时候，组件的透明度

**三、常见的触摸事件**

      在开发中会经常的用到点击、按下、抬起、长按等触发事件，那么在TouchableOpacity中又是改如何展示呢？

       在下面代码中分别演示了如何添加各种触摸事件：



## ReactNative开发进阶

## ReactNative 项目实战

## ReactNative 实战拓展

## ReactNative”入门到发疯”结尾