# React-Native

1. **创建项目**

npx react-native init 项目名

npx react-native run-android

### 监听

**16之前版本：**

componentWillReceiveProps：是在组件接收新值之前触发

componentDidUpdate：是在组件接收到新值之后触发

**16之后版本：**

**去除：**

componentWillMount  
componentWillReceiveProps  
componentWillUpdate

**新增：**

static getDerivedStateFromProps  
getSnapshotBeforeUpdate

static getDerivedStateFromProps：接受参数props，state

### 2.导航

* **插件安装：**

npm install @react-navigation/native

npm install react-native-reanimated react-native-gesture-handler react-native-screens react-native-safe-area-context @react-native-community/masked-view

npm install @react-navigation/stack

* **react-navigation分为三个部分**。  
  StackNavigator：类似顶部导航条，用来跳转页面和传递参数。  
  TabNavigator：类似底部标签栏，用来区分模块。  
  DrawerNavigator：抽屉，类似从App左侧滑出一个页面
* **API说明：**

**this.props.navigation**可以获取的一些方法：

**navigate**- 转到另一个页面, 计算出需要执行的操作　　（常用）

**goBack**- 关闭活动屏幕并在堆栈中向后移动　　（常用）

**addListener**- 订阅导航生命周期的更新

**isFocused**- 函数返回true如果屏幕焦点和false否则。

**state**- 当前状态/路由　　（常用）

**setParams**- 对路由的参数进行更改　　（常用）

**getParam**- 获取具有回退的特定参数　　（常用）

**dispatch**- 向路由发送 action　　（常用）

**dangerouslyGetParent**- 返回父级 navigator 的函数

注意： this.props.navigation并不是在所有页面（组件）中都可以使用，而是必须在StackNavigator、DrawerNavigator中声明的screen组件，才可以使用this.props.navigation。也就是说，screen组件会自动获得这个props

**this.props.navigation上还有一些方法取决于当前 navigator 的附加函数（StackNavigator 和 DrawerNavigator）**

**StackNavigator**：除了以上方法，还包含以下

**push**- 推一个新的路由到堆栈　　（常用）

**pop**- 返回堆栈中的上一个页面，如果提供一个参数n，则指定在堆栈内返回几层　　（常用）

**popToTop**- 跳转到堆栈中最顶层的页面，销毁其它所有页面　　（常用）

**replace**- 用新路由替换当前路由

**reset**- 清空堆栈，一般在退出登录的情况下操作　（常用）

**dismiss**- 关闭当前堆栈

**DrawerNavigator**：除了以上方法，还包含以下

**openDrawer**- 打开

**closeDrawer**- 关闭

**toggleDrawer**- 切换，如果是打开则关闭，反之亦然

* **参数传递：**

this.props.navigation.navigate(routeName, params, action)

* **接受参数**

**方式一**： this.props.route.parms（废弃：this.props.navigation.state.params.xxx）

**方式二**：this.props.navigation.getParam('title', 'profile');

**说明**： 过去, 当params未定义时, 你可能在获取params时遇到问题。 现在，你不必直接访问参数，可以调用getParam方法。

**功能**： 获取指定的参数，可以设置获取失败的返回值，比如上面的 title，当获取不到时，使用 profile 作为默认值

* 注意：3.x版本后，需要注意navigate和push的使用差别

**Push- 在堆栈顶部添加一条路由，并导航至该路由. 与navigate的区别在于，如果有已经加载的页面，navigate方法将跳转到已经加载的页面，而不会重新创建一个新的页面。push总是会创建一个新的页面，所以一个页面可以被多次创建。**

**3.暗黑模式**

https://www.freesion.com/article/1084712727/