**React native 组件**

1. React Native内置了对ES2015标准的支持，使用了JSX语法（一种在JavaScript中嵌入XML结构的语法），组件通过名为AppRegistry的内置模块进行了“注册”。
2. React native中两种控制组件特性的数据：props和state。props是在父组件中指定，一经指定，在被指定组件的生命周期中不再改变。而对于需要改变的数据，我们需要使用state。
3. 样式style属性用法
4. <View style={{width: 50, height: 50}} />
5. <View style={styles.myStyle} />，其中styles的定义如下

const styles = StyleSheet.create({

myStyle: {

color: 'blue',

fontWeight: 'bold',

},

red: {

color: 'red',

},

});

1. style可接受一个变量，也可接受一个数组，且数组中越往后优先级越高

<View style={[styles.myStyle, {backgroundColor: '#fff’}]} />

1. flex弹性宽度，可动态地扩张或收缩。flex:1来指定某个组件扩张以撑满所有剩余的空间。如果有多个并列的子组件使用了flex属性，则flex值越大，占据剩余空间的比例也越大。
2. Flexbox可以在不同屏幕尺寸上提供一致的布局结构。
3. flexDirection可以决定布局的**主轴**。子元素是沿着**水平轴(row)**排列，还是沿**竖直轴(column)**排，默认为**竖直轴(column)**方向。
4. justifyContent可以决定其子元素沿着**主轴**的**排列方式**。{flex-start、center、flex-end、space-around、space-between}。
5. alignItems可以决定其子元素沿着**次轴**（与主轴垂直的轴）的**排列方式。{**flex-start、center、flex-end、stretch**}**

**注：**alignItems: stretch，子元素在次轴方向上不能有固定的尺寸。

1. style属性可以是一个普通的JavaScript对象，也可以是一个数组，在数组中位置居后的样式对象比居前的优先级更高。
2. React Native中的尺寸都是无单位的，表示的是与设备像素密度无关的逻辑像素点。
3. flex可以使其在可利用的空间中动态地扩张或收缩，子控件占据剩余空间的比等于并列组件间flex值的比。
4. flexDirection可以决定布局的主轴（row/column），justifyContent可以决定其子元素沿着主轴的排列方式(flex-start、center、flex-end、space-around以及space-between)，alignItems可以决定其子元素沿着次轴（flex-start、center、flex-end以及stretch）。  
   **注**：要使stretch选项生效的话，子元素在次轴方向上不能有固定的尺寸。
5. 图片  
   a) **静态图片**：<Image source={require('./my-icon.png')} />  
    I. 实际被用到（即被require）的图片才会被打包到你的app；  
    II. require中的图片名字必须是一个静态字符串  
   b) **混合app图片** <Image source={{uri: 'app\_icon'}} style={{width: 40, height: 40}} />  
    确保图片在应用中确实存在，而且还需要指定尺寸。  
   c) 网络图片 <Image source={{uri: ‘https://facebook.github.io/react/img/logo\_og.png'}} style={{width: 400, height: 400}} /> （需要制定图片大小）
6. 动画：全局的布局动画LayoutAnimation和用于创建更精细的交互控制的动画Animated。
7. Animated有三种类型的动画：spring（弹跳动画），decay（衰减动画）和timing（从时间范围映射到渐变的值）。