

2.7 平台适配

编写一个跨平台的 app，你肯定希望代码的复用的越多越好，如果代码不一样，就要为 ios 和 android

各准备一套组件了。

React Native 提供了两种管理代码的方法：

- 用平台模块
- 用 `platform-specific file extensions`

有些组件只能在一个平台运行，这些属性都用 `@platform`，在网址上这些属性旁边都有个小标记。

平台模块

React Native 有个模块可以检测 app 是运行在哪个平台，如果组件只有一小部分不同就可以用这个区

分平台的方案。

```
import { Platform, StyleSheet } from 'react-native';
```

```
const styles = StyleSheet.create({  
  height: (Platform.OS === 'vr') ? 1 : 100,  
});
```

在 WebVR 中 `Platform.OS` 就是 `vr`

还有一个 `Platform.select` 方法可以用，它给出了把 `Platform.OS` 作为键的对象，返回现在运行的平台的值。

```
import { Platform, StyleSheet } from 'react-native';
```

```
const styles = StyleSheet.create({
  container:{
    flex: 1,
    ...Platform.select({
      ios: {
        backgroundColor: 'red',
      },
      android: {
        backgroundColor: 'blue',
      },
      vr: {
        backgroundColor: 'orange',
      },
    }),
  },
});
```

在三个平台中，都有一个 `flex` 为 1 的容器，只不过三个容器的颜色各不相同而已。

特定平台扩展名

当你的代码比较复杂，需要考虑分开放在不同的文件中时，就可以用这种方式了，在文件名后面加

上平台的后缀，如下所示：

```
BigButton.ios.js
```

```
BigButton.android.js
```

```
BigButton.vr.js
```

这样你就可以在组件引用了：

```
const BigButton = require('./BigButton');
```

这样不同的平台就会自动选择不同的文件加载了。