

## 1.7 组件、属性和状态

组件、属性和状态是 React Native 框架里面的内容，在 React VR 中也适用

### 属性和状态

组件里面有两种类型的数据：**props**(属性)和 **state**(状态)，属性是有父组件设置的，在组件的生命周

期内都有效，而那些需要改变的数据，我们用 **state**（状态）来改变

### props

大多数的组件在创建后都有不同的参数，这些参数也叫做 `props`。

你自己的组件也可以用属性，在你的 **app** 里面很多地方都可以用这个组件，在每个地方做些微的改

变就可以了，下面就是 `this.props` 的用法：

```
import React, { Component } from 'react';
import { AppRegistry, asset, Pano, Text, View } from 'react-vr';
```

```
class Greeting extends Component {
  render() {
    return (
      <Text>Hello {this.props.name}!</Text>
    );
  }
}
```

```
{  
  
class LotsOfGreetings extends Component {  
  render() {  
    return (  
      <View>  
        <Pano source={asset('chess-world.jpg')}/>  
        <View style={{transform: [{translate: [0, 0, -3]}]}}>  
          <Greeting name='Rexxar' />  
          <Greeting name='Jaina' />  
          <Greeting name='Valeera' />  
        </View>  
      </View>  
    );  
  }  
}
```

```
AppRegistry.registerComponent('LotsOfGreetings', () =>  
  LotsOfGreetings);
```

用 name 属性，我们可以复用 Greeting 组件

## state

一般情况下，我们在构造函数中初始化 state 数据，用 setState 来改变 state 的数据

```
import React, { Component } from 'react';  
import { AppRegistry, asset, Pano, Text, View } from 'react-vr';
```

```
class Blink extends Component {
  constructor(props){
    super(props);
    this.state = {showText: true};

    // Toggle the state every second
    setInterval(() => {
      this.state ({ showText: !this.state.showText });
    }, 1000);
  }

  render() {
    let display = this.state.showText ? this.props.text : ' ';
    return (
      <Text> {display}</Text>
    );
  }
}

class BlinkApp extends Component {
  render() {
    return (
      <View>
        <Pano source={asset('chess-world.jpg')}/>
        <View style={{transform: [{translate: [0, 0, -3]}]}}>
          <Blink name='I love to blink' />
          <Blink name='Yes blinking is so great' />
          <Blink name='Why did they ever take this out of HTML' />
        </View>
      </View>
    );
  }
}
```

```
      <Blink name='Look at me look at me look at me' />
    </View>
  </View>
);
}
}
```

```
AppRegistry.registerComponent('BlinkApp', () => BlinkApp);
```

state 的用法和 React 里面的用法是一样的，更多 state 的资料可以去查下 [React.Component API](#).