React
编程风格指南

前言

> 原文出处：http://segmentfault.com/a/1190000003899845

本文讨论的是 React.js 编程中的一些编程风格，团队开发中遵循统一的风格有利于提高代码的阅读体验。本文译自React style guide。下面将从三个方面来讲。

语法

方法顺序遵循生命周期放在前面， render() 方法放在最后

在 react 组件内部，方法的顺序如下：

1. 生命周期方法（按照时间先后顺序依次为： getDefaultProps , getInitialState , componentWillMount , componentDidMount , componentWillReceiveProps , shouldComponentUpdate , componentWillUpdate , componentDidUpdate , componentWillUnmount ）
2. 其他的方法
3. render 方法

事件处理函数的命名

采用 "handle" + "EventName" 的方式来命名
Example：

```
<Component onClick={this.handleClick} onLaunchMissiles={this.handleLaunchMissiles} />
```

事件函数作为属性时的命名

为了跟 react 的事件命名保持一致： onClick , onDrag , onChange , 等等，采用如下格式：

```
<Component onLaunchMissiles={this.handleLaunchMissiles} />
```

元素跟 return 放在同一行

为了节约空间，采用下面的写法：

```
return <div>
    ...
</div>;
```

而不是：

```
return (     // "div" 与 "return" 不在同一行
    <div>
        ...
    </div>
);
```

对HTML的属性进行对齐和排序

如果属性不是太多，那就放在同一行，否则就把每一个属性都单独写一行：

```
<div className="highlight" key="highlight-div">
<div
    className="highlight"
    key="highlight-div"
>
<Image
    className="highlight"
    key="highlight-div"
/>
```

而不是：

```
<div className="highlight"      // 属性没有在单独行
     key="highlight-div"
>
<div                            // 闭合便签不在单独的行
    className="highlight"
    key="highlight-div">
<div                            // 属性没有排序（一般重要的属性写在前面）
    key="highlight-div"
    className="highlight"
>
```

一个文件只导出一个 react 类

每一个 `.jsx` 应该只能导出单独的 `react 类`。这样有利于测试，因为这些测试框架要求一个文件导出的就是一个函数。

注意：你依然可以在一个文件中定义多个类，只要保证导出的只有一个即可。

语言特色

确保“呈现型”的组件功能单一

把 react 组件 分为“逻辑型组件”和“呈现型组件” 是很有必要的。前者包含的是业务逻辑，里面不应该包含HTML；后者一般是可复用的，可以包含HTML。前者可以拥有自己的内部的 state ，而后者不应该拥有。

多用 props

如果能用 props 就不要用 state ，这一定程度上可以减少应用程序的复杂度。
一般的模式是：创建一个“无状态”的组件（呈现型组件），只负责呈现数据，把包含 state 的“逻辑型组件”做为这些组件的父级组件。 然后把它内部的 state 作为 props 传递给下面的呈现型组件。这些逻辑型组件包含了所有的交互逻辑。

使用 propTypes

react 组件 都应该完成 propTypes 验证。每一个 this.props 的属性都应该有一个与之对应的 propTypes 。
避免使用这些没有描述意义的 prop-types:

- React.PropTypes.any
- React.PropTypes.array
- React.PropTypes.object

最好使用：

- React.PropTypes.arrayOf
- React.PropTypes.objectOf
- React.PropTypes.instanceOf
- React.PropTypes.shape

永远不要在DOM中保存 state

不要通过 data- 属性或class类。所有的信息应该都存储在javascript中，或者在React组件中，或者在 React store 中，如果使用了类似 Redux 这样的框架的话。

React 库和组件

不要使用 backbone 模型

直接使用 flux action，或者 `$.ajax` 来代替。

尽量少用 jQuery 就少用

永远也不要用jquery去操作DOM。
尝试避免jquery插件的使用。有必要的话，把jquery插件包装在React组件中。
你可以使用 `$.ajax`（但是不要用其他方法，像 `$.post`）来进行网络通信。

复用组件

你可以从react-components.com获取第三方React组件。