React JSX

React 使用 JSX 来替代常规的 JavaScript。

JSX 是一个看起来很像 XML 的 JavaScript 语法扩展。

我们不需要一定使用 JSX，但它有以下优点：

JSX 执行更快，因为它在编译为 JavaScript 代码后进行了优化。

它是类型安全的，在编译过程中就能发现错误。

使用 JSX 编写模板更加简单快速。

|  |
| --- |
| <!DOCTYPE html>  <html>  <head>  <title>hello world</title>  <script src="./build/react.min.js"></script>  <script src="./build/react-dom.min.js"></script>  <script src="./build/browser.min.js"></script>  </head>  <body>  <div id="container"></div>  <script type="text/babel">  ReactDOM.render(  <div>  <h1>hello, world!</h1>  <h2>welcome study</h2>  {/\* 注释:添加自定义属性需要使用 data- 前缀 \*/}  <p data-myattribute= "somevalue">JavaScript</p>  </div>,  document.getElementById('container')  );  </script>  </body>  </html> |

**JavaScript 表达式**

可在 JSX 中使用 JavaScript 表达式。写在花括号 **{}** 中。实例如下：

|  |
| --- |
| ReactDOM.render(  <div>  <h1>{1+1}</h1>  </div>,  document.getElementById('container')  ); |

JSX 中不能使用 if else 语句，但可用三元运算表达式来替代。

以下实例中如果变量 i = 1, 浏览器将输出 true, 如果修改 i 的值，则会输出 false.

|  |
| --- |
| var i = 1;  ReactDOM.render(  <div>  <h1>{i == 1 ? 'True!' : 'False'}</h1>  </div>,  document.getElementById('container')  ); |

**样式**

React 推荐使用内联样式。可使用 camelCase 语法来设置内联样式.

React 会在指定元素数字后自动添加 px 。

|  |
| --- |
| var mystyle = {  fontSize : 100,  color : "#FF0000"  };  ReactDOM.render(  <div>  <h1 style={mystyle}>菜鸟教程</h1>  </div>,  document.getElementById('container')  ); |

**数组**

JSX 允许在模板中插入数组，数组会自动展开所有成员：

|  |
| --- |
| var arr = [  <h1>react教程</h1>,  <h2>学的不仅是技术，更是梦想！</h2>,  ];  ReactDOM.render(  <div>{arr}</div>,  document.getElementById('container')  ); |

HTML 标签 vs. React 组件

React 可渲染 HTML 标签 (strings) 或 React 组件 (classes)。

要渲染 HTML 标签，只需在 JSX 里使用小写字母的标签名。

|  |
| --- |
| var myDivElement = <div className="foo"/>  ReactDOM.render(  myDivElement,  document.getElementById('container')  ); |

要渲染 React 组件，只需创建一个大写字母开头的本地变量。

|  |
| --- |
| var MyComponent = React.createClass({/\*...\*/});  var myElement = <MyComponent someProperty={true} />;  ReactDOM.render(  myElement,  document.getElementById('container')  ); |

# React 组件

使用组件使应用更容易来管理。

封装一个输出 "Hello World！" 的组件，组件名为 HelloMessage

|  |
| --- |
| var HelloMessage = React.createClass({  render : function() {  return <h1>hello world!</h1>;  }  });  ReactDOM.render(  <HelloMessage/>,  document.getElementById('container')  ); |

实例解析：

React.createClass 方法用于生成一个组件类 HelloMessage。

<HelloMessage /> 实例组件类并输出信息。

注意，原生 HTML 元素名以小写字母开头，而自定义的 React 类名以大写字母开头，比如 HelloMessage 不能写成 helloMessage。

还需要注意组件类只能包含一个顶层标签，否则也会报错。