

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。

我们先看下以下代码：

```
const element = <h1>Hello, world!</h1>;
```

这种看起来可能有些奇怪的标签语法既不是字符串也不是 HTML。

它被称为 JSX，一种 JavaScript 的语法扩展。我们推荐在 React 中使用 JSX 来描述用户界面。

JSX 是在 JavaScript 内部实现的。

我们知道元素是构成 React 应用的最小单位，JSX 就是用来声明 React 当中的元素。

与浏览器的 DOM 元素不同，React 当中的元素事实上是普通的对象，React DOM 可以确保 浏览器 DOM 的数据内容与 React 元素保持一致。

要将 React 元素渲染到根 DOM 节点中，我们通过把它们都传递给 ReactDOM.render() 的方法来将其渲染到页面上：

React 实例

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

尝试一下 »

注意:

由于 JSX 就是 JavaScript，一些标识符像 `class` 和 `for` 不建议作为 XML 属性名。作为替代，React DOM 使用 `className` 和 `htmlFor` 来做对应的属性。

使用 JSX

JSX 看起来类似 HTML，我们可以看下实例:

```
ReactDOM.render(
  <h1>Hello, world!</h1>,
  document.getElementById('example')
);
```

我们可以在以上代码中嵌套多个 HTML 标签，需要使用一个 div 元素包裹它，实例中的 p 元素添加了自定义属性 **data-myattribute**，添加自定义属性需要使用 **data-** 前缀。

React 实例

```
ReactDOM.render(  
  <div>  
    <h1>菜鸟教程</h1>  
    <h2>欢迎学习 React</h2>  
    <p data-myattribute = "somevalue">这是一个很不错的 JavaScript 库!</p>  
  </div>  
,  
  document.getElementById('example')  
);
```

[尝试一下 »](#)

独立文件

你的 React JSX 代码可以放在一个独立文件上，例如我们创建一个 `helloworld_react.js` 文件，代码如下：

```
ReactDOM.render(  
  <h1>Hello, world!</h1>,  
  document.getElementById('example')  
);
```

然后在 HTML 文件中引入该 JS 文件：

React 实例

```
<body>  
  <div id="example"></div>  
  <script type="text/babel" src="helloworld_react.js"></script>  
</body>
```

[尝试一下 »](#)

JavaScript 表达式

我们可以在 JSX 中使用 JavaScript 表达式。表达式写在花括号 `{}` 中。实例如下：

React 实例

```
ReactDOM.render(  
  <div>  
    <h1>{1+1}</h1>  
  </div>  
,  
  document.getElementById('example')  
);
```

[尝试一下 »](#)

在 JSX 中不能使用 **if else** 语句，但可以使用 **conditional (三元运算)** 表达式来替代。以下实例中如果变量 **i** 等于 **1** 浏览器将输出 **true**, 如果修改 **i** 的值，则会输出 **false**.

React 实例

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

[尝试一下 »](#)

样式

React 推荐使用内联样式。我们可以使用 **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**。以下实例演示了为 **h1** 元素添加 **myStyle** 内联样式：

React 实例

```
var myStyle = {  
  fontSize: 100,  
  color: '#FF0000'  
};  
ReactDOM.render(  
  <h1 style = {myStyle}>菜鸟教程</h1>,  
  document.getElementById('example')  
);
```

[尝试一下 »](#)

注释

注释需要写在花括号中，实例如下：

React 实例

```
ReactDOM.render(  
  <div>  
    <h1>菜鸟教程</h1>  
    { /*注释...*/ }  
  </div>,  
  document.getElementById('example')  
);
```

[尝试一下 »](#)

数组

JSX 允许在模板中插入数组，数组会自动展开所有成员：

React 实例

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

尝试一下 »

← React 安装

React 组件 →



3 篇笔记

✎ 写笔记