

React.js 小书

[<-- 返回首页](#)

使用 JSX 描述 UI 信息

- 作者: [胡子大哈](#)
- 原文链接: <http://huziketang.com/books/react/lesson6>
- 转载请注明出处, 保留原文链接和作者信息。

这一节我们通过一个简单的例子讲解 React.js 描述页面 UI 的方式。把

`src/index.js` 中的代码改成:

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import './index.css'

class Header extends Component {
  render () {
    return (
      <div>
        <h1>React 小书</h1>
      </div>
    )
  }
}

ReactDOM.render(
  <Header />,
  document.getElementById('root')
)
```

我们在文件头部从 `react` 的包当中引入了 `React` 和 `React.js` 的组件父类 `Component`。记住, 只要你要写 `React.js` 组件, 那么就必须要引入这两个东西。

`ReactDOM` 可以帮助我们把 `React` 组件渲染到页面上去, 没有其它的作用了。你可以发现它是从 `react-dom` 中引入的, 而不是从 `react` 引入。有些朋友可能会疑惑, 为什么不把这些东西都包含在 `react` 包当中呢? 我们稍后会回答这个问题。

接下来的代码你看起来会比较熟悉, 但又会有点陌生。你看其实它跟我们前几节里面讲的内容其实很类似, 一个组件继承 `Component` 类, 有一个 `render` 方法, 并且把这个组件的 HTML 结构返回; 这里 `return` 的东西就比较奇怪了, 它并不是一个字符串, 看起来像是纯 HTML 代码写在 JavaScript 代码里面。你也许会说, 这不就有语法错误了么? 这完全不是合法的 JavaScript 代码。这种看起来“在 JavaScript 写的标签的”语法叫 JSX。

JSX 原理

为了让大家深刻理解 JSX 的含义。有必要简单介绍了一下 JSX 稍微底层的运作原理，这样大家可以更加深刻理解 JSX 到底是什么东西，为什么要有这种语法，它是经过怎么样的转化变成页面的元素的。

思考一个问题：如何用 JavaScript 对象来表现一个 DOM 元素的结构，举个例子：

```
<div class='box' id='content'>
  <div class='title'>Hello</div>
  <button>Click</button>
</div>
```

每个 DOM 元素的结构都可以用 JavaScript 的对象来表示。你会发现一个 DOM 元素包含的信息其实只有三个：标签名，属性，子元素。

所以其实上面这个 HTML 所有的信息我们都可以用合法的 JavaScript 对象来表示：

```
{
  tag: 'div',
  attrs: { className: 'box', id: 'content' },
  children: [
    {
      tag: 'div',
      attrs: { className: 'title' },
      children: ['Hello']
    },
    {
      tag: 'button',
      attrs: null,
      children: ['Click']
    }
  ]
}
```

你会发现，HTML 的信息和 JavaScript 所包含的结构和信息其实是一样的，我们可以用 JavaScript 对象来描述所有能用 HTML 表示的 UI 信息。但是用 JavaScript 写起来太长了，结构看起来又不清晰，用 HTML 的方式写起来就方便很多了。

于是 React.js 就把 JavaScript 的语法扩展了一下，让 JavaScript 语言能够支持这种直接在 JavaScript 代码里面编写类似 HTML 标签结构的语法，这样写起来就方便很多了。编译的过程会把类似 HTML 的 JSX 结构转换成 JavaScript 的对象结构。

上面的代码：

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import './index.css'
```

```
class Header extends Component {
  render () {
    return (
      <div>
        <h1 className='title'>React 小书</h1>
      </div>
    )
  }
}

ReactDOM.render(
  <Header />,
  document.getElementById('root')
)
```

经过编译以后会变成：

```
import React, { Component } from 'react'
import ReactDOM from 'react-dom'
import './index.css'

class Header extends Component {
  render () {
    return (
      React.createElement(
        "div",
        null,
        React.createElement(
          "h1",
          { className: 'title' },
          "React 小书"
        )
      )
    )
  }
}

ReactDOM.render(
  React.createElement(Header, null),
  document.getElementById('root')
);
```

`React.createElement` 会构建一个 JavaScript 对象来描述你 HTML 结构的信息，包括标签名、属性、还有子元素等。这样的代码就是合法的 JavaScript 代码了。所以使用 React 和 JSX 的时候一定要经过编译的过程。

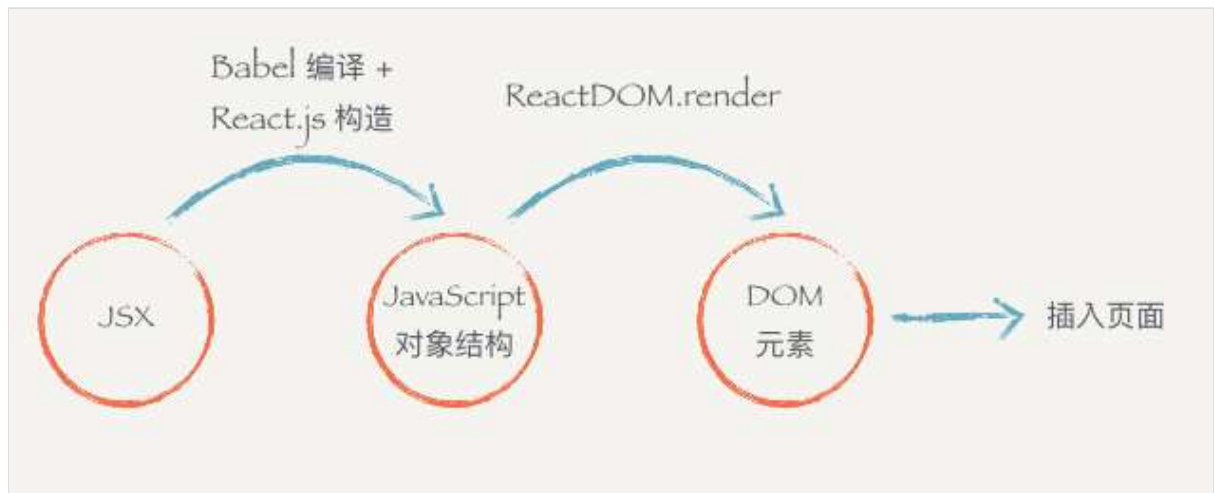
这里再重复一遍：所谓的 **JSX** 其实就是 **JavaScript 对象**。每当在 JavaScript 代码中看到这种 JSX 结构的时候，脑子里面就可以自动做转化，这样对你理解 React.js 的组件写法很有好处。

有了这个表示 HTML 结构和信息的对象以后，就可以拿去构造真正的 DOM 元素，然后把这个 DOM 元素塞到页面上。这也是我们最后一段代码中 `ReactDOM.render` 所干的事情：

```
ReactDOM.render(  
  <Header />,  
  document.getElementById('root')  
)
```

`ReactDOM.render` 功能就是把组件渲染并且构造 DOM 树，然后插入到页面上某个特定的元素上（在这里是 id 为 `root` 的 `div` 元素）。

所以可以总结一下从 JSX 到页面到底经过了什么样的过程：



有些同学可能会问，为什么不直接从 JSX 直接渲染构造 DOM 结构，而是要经过中间这么一层呢？

第一个原因是，当我们拿到一个表示 UI 的结构和信息的对象以后，不一定会把元素渲染到浏览器的普通页面上，我们有可能把这个结构渲染到 canvas 上，或者是手机 App 上。所以这也是为什么要把 `react-dom` 单独抽离出来的原因，可以想象有一个叫 `react-canvas` 可以帮我们把 UI 渲染到 canvas 上，或者是有一个叫 `react-app` 可以帮我们把它转换成原生的 App（实际上这玩意叫 `ReactNative`）。

第二个原因是，有了这样一个对象。当数据变化，需要更新组件的时候，就可以用比较快的算法操作这个 JavaScript 对象，而不用直接操作页面上的 DOM，这样可以尽量少的减少浏览器重排，极大地优化性能。这个在以后的章节中我们会提到。

总结

要记住几个点：

1. JSX 是 JavaScript 语言的一种语法扩展，长得像 HTML，但并不是 HTML。
2. React.js 可以用 JSX 来描述你的组件长什么样的。
3. JSX 在编译的时候会变成相应的 JavaScript 对象描述。

4. `react-dom` 负责把这个用来描述 UI 信息的 JavaScript 对象变成 DOM 元素，并且渲染到页面上。

课后练习题

- [用 React.js 在页面上渲染标题](#)

因为第三方评论工具有问题，对本章节有任何疑问的朋友可以移步到 [React.js 小书的论坛](#) 发帖，我会回答大家的疑问。

[下一节：组件的 render 方法](#)

[上一节：React.js 基本环境安装](#)

如果你觉得小书写得还不错，可以请胡子大哈喝茶 :)

赞赏

或者传播一下知识也是一个很好的选择

4 条评论，3 人参与。



我有话说...

使用社交帐号登录

发布前先点击左边的按钮登录

最新评论



冬、 • 6月27日 12:56

受益匪浅

顶 • 回复 • 分享»



冬、 • 6月27日 12:56

太好了

顶 • 回复 • 分享»



librAquarius • 4月17日 11:29

感谢作者分享，受教了。

顶 • 回复 • 分享»



oneminxyz • 4月12日 15:34

NICE

顶 • 回复 • 分享»

友言?