

React.js 小书

[<-- 返回首页](#)

组件的组合、嵌套和组件树

- 作者: [胡子大哈](#)
- 原文链接: <http://huziketang.com/books/react/lesson8>
- 转载请注明出处, 保留原文链接和作者信息。

继续拓展前面的例子, 现在我们已经有了 `Header` 组件了。假设我们现在构建一个新的组件叫 `Title`, 它专门负责显示标题。你可以在 `Header` 里面使用 `Title` 组件:

```
class Title extends Component {
  render () {
    return (
      <h1>React 小书</h1>
    )
  }
}

class Header extends Component {
  render () {
    return (
      <div>
        <Title />
      </div>
    )
  }
}
```

我们可以直接在 `Header` 标签里面直接使用 `Title` 标签。就像是一个普通的标签一样。React.js 会在 `<Title />` 所在的地方把 `Title` 组件的 `render` 方法表示的 JSX 内容渲染出来, 也就是说 `<h1>React 小书</h1>` 会显示在相应的位置上。如果现在我们在 `Header` 里面使用三个 `<Title />`, 那么就会有三个 `<h1 />` 显示在页面上。

```
<div>
  <Title />
  <Title />
  <Title />
</div>
```

这样可复用性非常强, 我们可以把组件的内容封装好, 然后灵活在使用在任何组件内。另外这里要注意的是, 自定义的组件都必须要用大写字母开头, 普通的 HTML 标签都用小写字母开头。

现在让组件多起来。我们来构建额外的组件来构建页面，假设页面是由 `Header`、`Main`、`Footer` 几个部分组成，由一个 `Index` 把它们组合起来。

```
import React, { Component } from 'react';
import ReactDOM from 'react-dom';

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

class Header extends Component {
  render () {
    return (
      <div>
        <Title />
        <h2>This is Header</h2>
      </div>
    )
  }
}

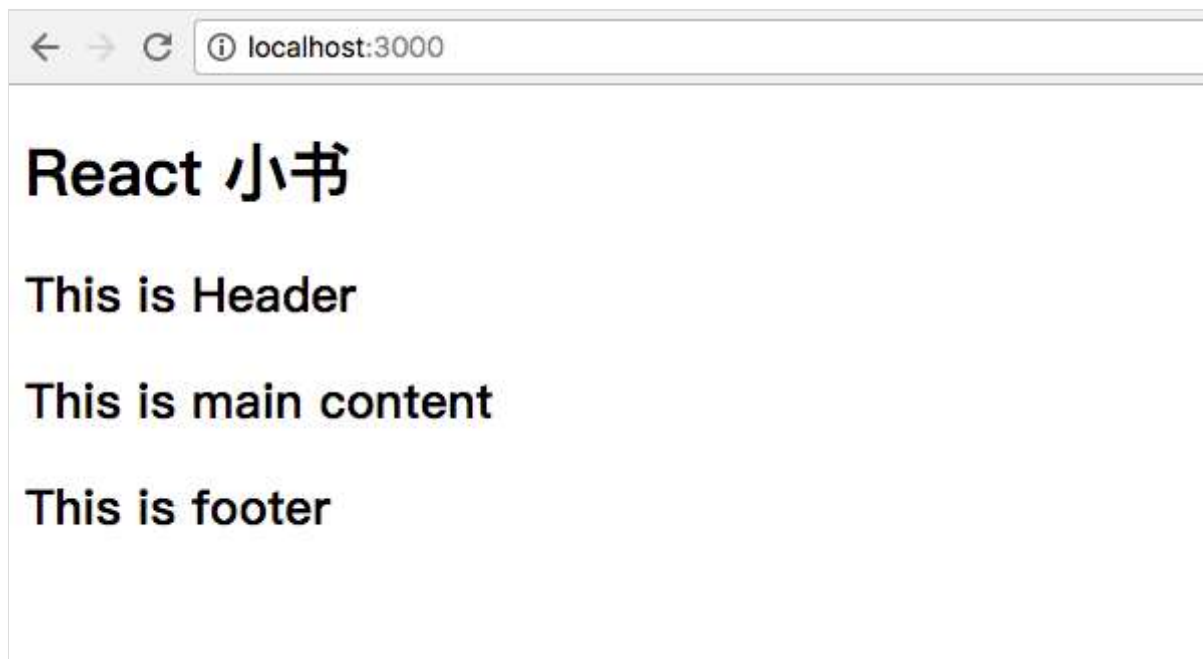
class Main extends Component {
  render () {
    return (
      <div>
        <h2>This is main content</h2>
      </div>
    )
  }
}

class Footer extends Component {
  render () {
    return (
      <div>
        <h2>This is footer</h2>
      </div>
    )
  }
}

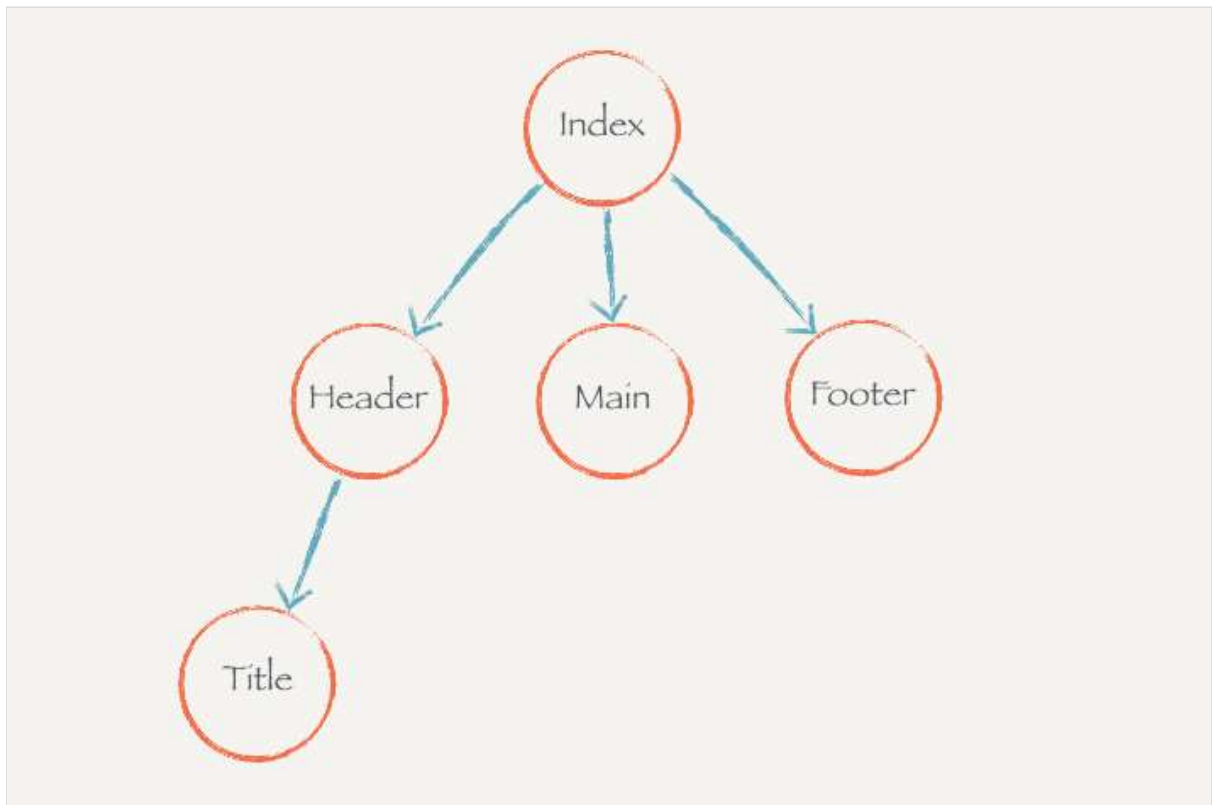
class Index extends Component {
  render () {
    return (
      <div>
        <Header />
        <Main />
        <Footer />
      </div>
    )
  }
}
```

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

最后页面会显示内容：



组件可以和组件组合在一起，组件内部可以使用别的组件。就像普通的 HTML 标签一样使用就可以。这样的组合嵌套，最后构成一个所谓的组件树，就正如上面的例子那样，`Index` 用了 `Header`、`Main`、`Footer`，`Header` 又使用了 `Title`。这样用这样的树状结构表示它们之间的关系：



这里的结构还是比较简单，因为我们的页面结构并不复杂。当页面结构复杂起来，有许多不同的组件嵌套组合的话，组件树会相当的复杂和庞大。理解组件树的概念对后面理解数据是如何在组件树内自上往下流动过程很重要。

课后练习

- [用 React.js 组建的房子](#)

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：事件监听](#)

[上一节：组件的 render 方法](#)

如果你觉得小书写得还不错，可以请胡子大哈喝杯茶 :)

赞赏

或者传播一下知识也是一个很好的选择