

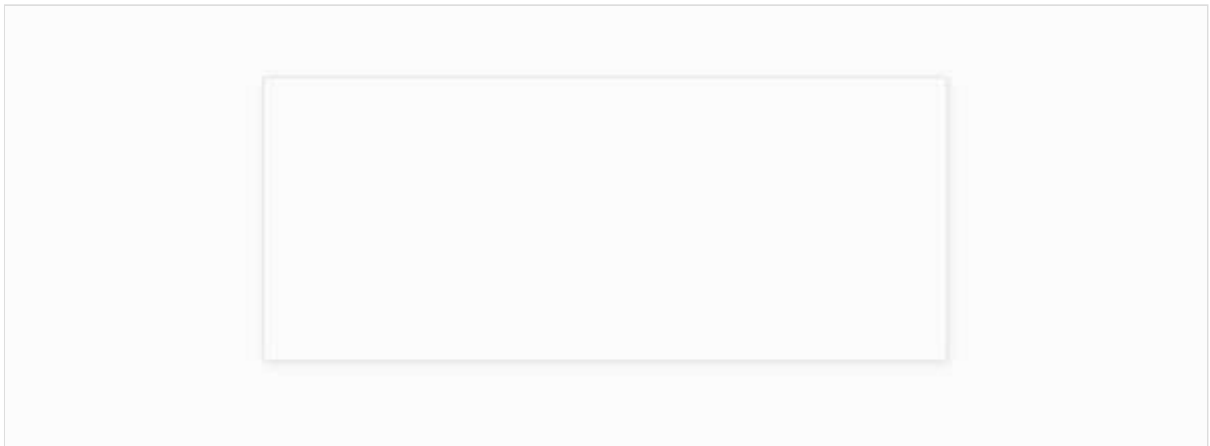
React.js 小书

[<-- 返回首页](#)

props.children 和容器类组件

- 作者: [胡子大哈](#)
- 原文链接: <http://huziketang.com/books/react/lesson22>
- 转载请注明出处, 保留原文链接和作者信息。

有一类组件, 充当了容器的作用, 它定义了一种外层结构形式, 然后你可以往里面塞任意的内容。这种结构在实际当中非常常见, 例如这种带卡片组件:



组件本身是一个不带任何内容的方形的容器, 我可以在用这个组件的时候给它传入任意内容:



基于我们目前的知识储备, 可以迅速写出这样的代码:

```
class Card extends Component {
  render () {
    return (
      <div className='card'>
        <div className='card-content'>
          {this.props.content}
        </div>
      </div>
    )
  }
}
```

```
        </div>
      )
    }
  }

ReactDOM.render(
  <Card content={
    <div>
      <h2>React.js 小书</h2>
      <div>开源、免费、专业、简单</div>
      订阅: <input />
    </div>
  } />,
  document.getElementById('root')
)
```

我们通过给 `Card` 组件传入一个 `content` 属性，这个属性可以传入任意的 JSX 结构。然后在 `Card` 内部会通过 `{this.props.content}` 把内容渲染到页面上。

这样明显太丑了，如果 `Card` 除了 `content` 以外还能传入其他属性的话，那么这些 JSX 和其他属性就会混在一起。很不好维护，如果能像下面的代码那样使用 `Card` 那想必也是极好的：

```
ReactDOM.render(
  <Card>
    <h2>React.js 小书</h2>
    <div>开源、免费、专业、简单</div>
    订阅: <input />
  </Card>,
  document.getElementById('root')
)
```

如果组件标签也能像普通的 HTML 标签那样编写内嵌的结构，那么就方便很多了。实际上，React.js 默认就支持这种写法，所有嵌套在组件中的 JSX 结构都可以在组件内部通过 `props.children` 获取到：

```
class Card extends Component {
  render () {
    return (
      <div className='card'>
        <div className='card-content'>
          {this.props.children}
        </div>
      </div>
    )
  }
}
```

把 `props.children` 打印出来，你可以看到它其实是个数组：

```
▼ Array[4]
  ▶ 0: Object
  ▶ 1: Object
  2: "订阅: "
  ▶ 3: Object
    length: 4
  ▶ __proto__: Array[0]
```

React.js 就是把嵌套的 JSX 元素一个个都放到数组当中，然后通过 `props.children` 传给了 `Card`。

由于 JSX 会把插入表达式里面数组中的 JSX 一个个罗列下来显示。所以其实就相当于在 `Card` 中嵌套了什么 JSX 结构，都会显示在 `Card` 的类名为 `card-content` 的 `div` 元素当中。

这种嵌套的内容成为了 `props.children` 数组的机制使得我们编写组件变得非常的灵活，我们甚至可以在组件内部把数组中的 JSX 元素安置在不同的地方：

```
class Layout extends Component {
  render () {
    return (
      <div className='two-cols-layout'>
        <div className='sidebar'>
          {this.props.children[0]}
        </div>
        <div className='main'>
          {this.props.children[1]}
        </div>
      </div>
    )
  }
}
```

这是一个两列布局组件，嵌套的 JSX 的第一个结构会成为侧边栏，第二个结构会成为内容栏，其余的结构都会被忽略。这样通过这个布局组件，就可以在各个地方高度复用我们的布局。

总结

使用自定义组件的时候，可以在其中嵌套 JSX 结构。嵌套的结构在组件内部都可以通过 `props.children` 获取到，这种组件编写方式在编写容器类型的组件当中非常有用。而在实际的 React.js 项目当中，我们几乎每天都需要用这种方式来编写组件。

课后练习

*[黑色边框的容器组件](#)

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: dangerouslySetHTML 和 style 属性

上一节: ref 和 React.js 中的 DOM 操作

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

0 条评论，0 人参与。

★ 2



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

还没有评论

友言?