React 的设计思想

React 是目前最火的用户界面开发库之一。它获得今天这样的业界地位并不是一个偶然，最根本的原因，是 React 汇聚了几十年用户界面开发的经验思想，贯彻了众多被证明行之有效的模式和原则。

这本小册，就是向大家讲解 React 应用中的模式和原则。

内容范围

读者应该已经对 React 有一个初步了解，熟悉 React 的基本用法，这本小册的目的是帮助读者进一步认识 React。读者在看完这本小册之后，应该不只是利用 React 来开发网页应用，还能够应用各种设计模式，开发出高效易于维护的代码。

如果读者对 React 还并不了解，可以先去看作者写的《深入浅出React和Redux》，本小册的目的是提高开发者水平，所以不会花太多篇幅讲解 React 的基础入门知识。

当然，React 技术的应用并不只限于网页，可以利用 React Native 来开发原生应用，但是受篇幅限制，本小册不探讨 React Native 的用法，而是集中介绍 React 技术在网页中的应用。

React 的基础原则

要真正理解 React，开发者必须要明白这几点：

1. React 界面完全由数据驱动；
2. React 中一切都是组件；
3. props 是 React 组件之间通讯的基本方式。

好的，让我们开始吧！

界面完全由数据驱动

初学 React 的开发者，尤其是用惯了 jQuery 的开发者，往往对 React 的工作方式很难接受，但是，请试着接受这一点，React 的哲学，简单说来可以用下面这条公式来表示：

```
UI = f(data)
```

等号左边的 UI 代表最终画出来的界面；等号右边的 f 是一个函数，也就是我们写的 React 相关代码；data 就是数据，在 React 中，data 可以是 state 或者 props。

UI 就是把 data 作为参数传递给 f 运算出来的结果。这个公式的含义就是，如果要渲染界面，不要直接去操纵 DOM 元素，而是修改数据，由数据去驱动 React 来修改界面。

我们开发者要做的，就是设计出合理的数据模型，让我们的代码完全根据数据来描述界面应该画成什么样子，而不必纠结如何去操作浏览器中的 DOM 树结构。

这样一种程序结构，是声明式编程（Declarative Programming）的方式，代码结构会更加容易理解和维护。

组件：React 世界的一等公民

React 是一个用 JavaScript 语言开发的库，而我们知道，在 JavaScript 的世界里，一切皆是对象，甚至连一个函数都是一个对象。

我们可以把一个函数当做对象来赋值给一个变量，可以把对象作为参数传递给一个函数，也可以访问一个函数的属性，函数完全就是一个对象。

比如，你是否知道，你可以访问一个函数的 length 来获知这个函数声明的参数？

```
function foo(a, b) {
  return a + b;
}

console.log(foo.length); // 输出为2
```

因为函数本身就是对象，所以，在 JavaScript 中，函数拥有一等公民的地位。

那么，在 React 的世界中，什么是一等公民呢？

答案就是**组件（Component）**。

可以这么说，在 React 中一切皆为组件。这是因为：

1. 用户界面就是组件；
2. 组件可以嵌套包装组成复杂功能；
3. 组件可以用来实现副作用。

接下来，我们逐个来看组件的这三个方面。

第一点用户界面就是组件，很好理解，在界面上看到的任何一个“块”，都需要代码来实现，而这部分代码最好是独立存在的，与其他代码之间的纠葛越少越好，所以要把这个“块”的相关代码封装在一个代码单元里。这样的代码单元，在 React 里就是一个“组件”。

在上面的图中，一个 Button 是一个界面元素，对应的就是一个 React 组件。在 React 中，一个组件可以是一个类，也可以是一个函数，这取决于这个组件是否有自己的状态。

第二点，组件可以嵌套包装组成复杂功能。现实中的应用是很复杂的，界面设计中包含很多元素，一个“块”套着另一个“块”，React 中的组件可以重复嵌套，就是为了支持现实中的用户界面需要。

第三点，组件可以用来实现副作用。并不是说组件必须要在界面画一些东西，一个组件可以什么都不画，或者把画界面的事情交给其他组件去做，自己做一些和界面无关的事情，比如获取数据。

下面是一个 Beacon 组件，它的 render 函数返回为空，所以它实际上并不渲染任何东西。

```
class Beacon extends React.Component {
  render() {
    return null;
  }

  componentDidMount() {
    const beacon = new Image();
    beacon.src = 'https://domain.name/beacon.gif';
  }
}
```

不过，Beacon 的 componentDidMount 函数中创造了一个 `Image` 对象，访问了一个特定的图片资源，这样就可以对应服务器上留下日志记录，用于记录这一次网页访问。

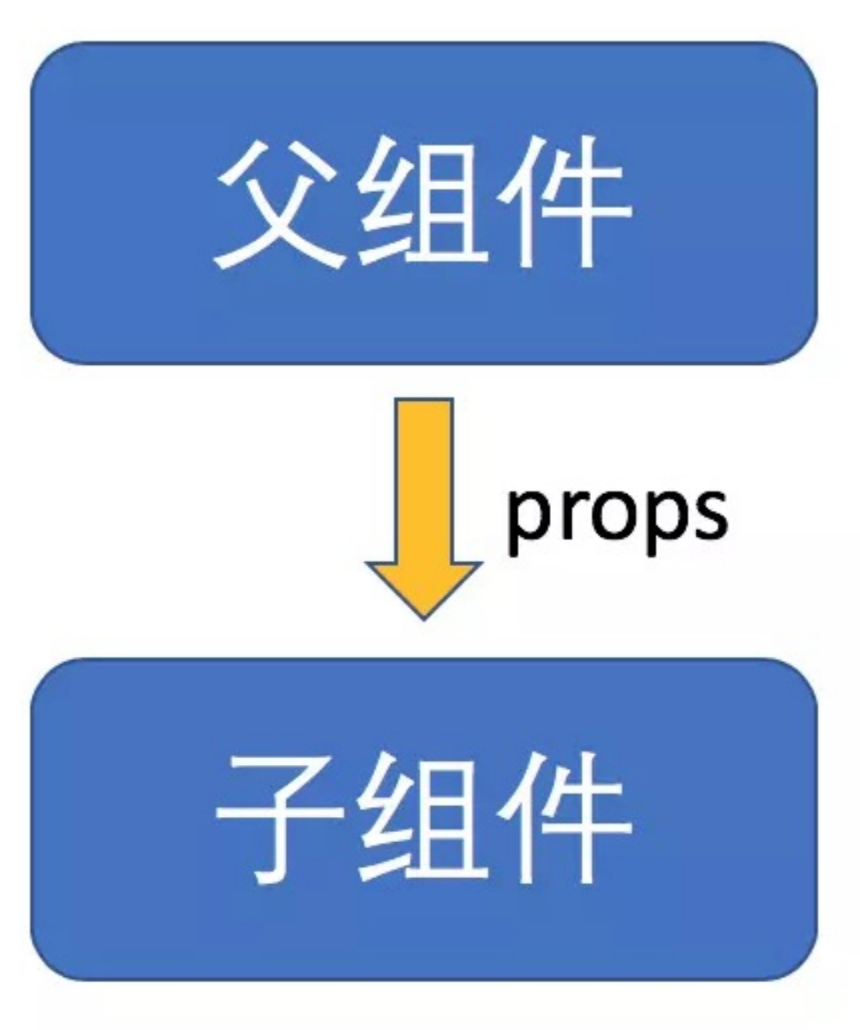
Beacon 组件的使用方式和普通组件别无二致，但是却能够轻松实现对网页访问的跟踪。

```
<div>
  <Beacon />
</div>
```

组件之间的语言：props

如果说组件是 React 世界的一等公民，这些公民之间也肯定是需要交流的，他们通过什么语言交流呢？答案就是 props。

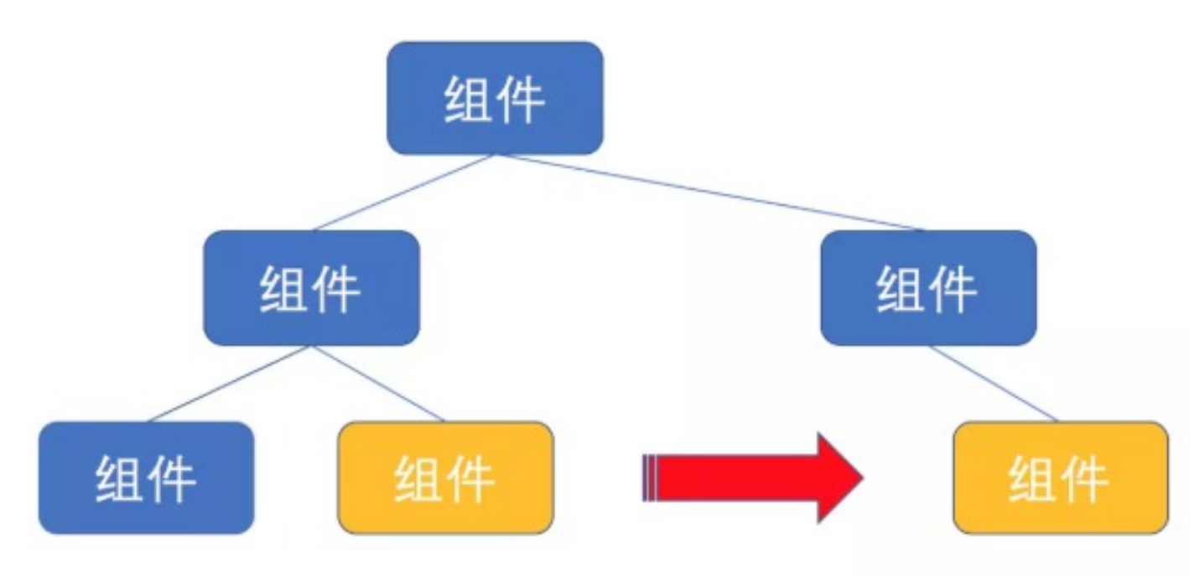
如果一个父组件有话要对子组件说，也就是，想要传递数据给子组件，则应该通过 props。



当然，你可以给子组件增加一个新的函数，然后让父组件去调用这个函数，但是，这种方法很拙劣。如果直接调用子组件的函数，执行过程也处于 React 生命周期之外，所以，不应该使用这种方法。

同样，如果子组件有话要同父组件说，那应该支持函数类型的 props。身为 JavaScript 里一等公民的函数可以作为参数传递，当然也可以作为 props 传递。让父组件传递一个函数类型的 props 进来，当子组件要传递数据给父组件时，调用这个函数类型 props，就把信息传递给了父组件。

如果两个完全没有关系的组件之间有话说，情况就复杂了一点，比如下图中，两个橙色组件之间如果有话说，就没法直接通过 props 来传递信息。



一个比较土的方法，就是通过 props 之间的逐步传递，来把这两个组件关联起来。如果之间跨越两三层的关系，这种方法还凑合，但是，如果这两个组件隔了十几层，或者说所处位置多变，那让 props 跨越千山万水来相会，实在是得不偿失。

另一个简单的方式，就是建立一个全局的对象，两个组件把想要说的话都挂在这个全局对象上。这种方法当然简单可行，但是，我们都知道全局变量的危害罄竹难书，如果不想将来被难以维护的代码折磨，我们最好对这种方法敬而远之。

一般，业界对于这种场景，往往会采用第三方数据管理工具来解决，在本小册的第 12 和第 13 小节会详细介绍用 Redux 和 Mobx 解决这些问题的方法。

其实，不依赖于第三方工具，React 也提供了自己的跨组件通讯方式，这种方式叫 Context，在第 8 小节会详细介绍。

小结

本节介绍了使用 React 的基本思想，希望读者能够学习到：

1. 在 React 中，界面完全由数据驱动；
2. 在 React 中，一切都是组件；
3. props 是 React 组件之间通讯的基本方式。

在下一小节中，我们会投入实践，用 React 的基本原则来设计一个比较复杂的应用。

留言

评论将在后台进行审核，审核通过后对所有人可见

霍森笙箫 前端工程师
买过书，听了live，程大出品一如既往支持。言简意赅，温故知新
▲ 0 评论 7天前

yitiaohui0226 高级Android开发 @ PDD
支持一下老兵，在读RxJS
▲ 0 评论 8天前

Thoughtful valiant. 前端青瓜蛋子
买了一段时间才开始看，顺便也买了一本《深入浅出React和Redux》支持一波~
▲ 0 收起评论 15天前

> 程墨 Hulu
> 感谢支持
> 14天前
>
> 评论审核通过后显示 评论

xhhcocos2d
随着框架的升级，我想问下会更新吗？
▲ 0 评论 17天前

我贵姓张
买一本 Michele Bertoli 的 react设计模式和最佳实践 电子书版也才 29 啊
▲ 0 收起评论 1月前

> 程墨 Hulu
> 你是觉得这个小册定价太贵了吗？
> 1月前
>
> ImCat
> 老铁，你买的那本书你觉得内容如何？
> 6天前
>
> 评论审核通过后显示 评论

小汪大汪旺旺 前端工程师 @ 湖南云迈科技
191s
▲ 0 评论 1月前

三郎本尊
之前学习react，程大神写的书帮助不少，必属精品，买了
▲ 1 评论 1月前

起风了8866 前端工程师
看网上说propType是必须的，但考虑到增加工作量而且实际效果却只是提供一个报错功能。感觉鸡肋，所以没用
▲ 0 收起评论 1月前

> 程墨 Hulu
> propType不是必须的，完全可以用TypeScript或者Flow代替，所以也不是React核心package的一部分。
> 1月前
>
> 程序界霍建华 前端开发工程师 @ 湾区
> 是的，之前项目没有引入 TS，propType 大多数时候起的作用都很小
> 1月前
>
> 评论审核通过后显示 评论

zhangyanling77 前端开发 @ 成都
刚看完深入浅出react和redux。
赶紧来支持一波！
▲ 0 收起评论 1月前

> 程墨 Hulu
> 感谢支持
> 1月前 回复
>
> 评论审核通过后显示 评论

Xaasu developer @ undefined
上文提到的propTypes在这章没有用上，不是很明白propTypes的具体作用
▲ 0 收起评论 1月前

> 程墨 Hulu
> PropTypes最初属于React的一部分，顾名思义就是“属性类型检查”，但是，功能上没有PropTypes也依然可行，而且还有TypeScript之类工具实现一样的功能，所以后来React就把PropTypes放在一个单独的包prop-types里了。
> 1月前
>
> 起风了8866 前端工程师
> 看网上说propType是必须的，但考虑到增加工作量而且实际效果却只是提供一个报错功能。感觉鸡肋，所以没用
> 1月前
>
> 起风了8866 前端工程师
> 回复 程墨：看网上说propType是必须的，但考虑到增加工作量而且实际效果却只是提供一个报错功能。感觉鸡肋，所以没用
> 1月前
>
> 评论审核通过后显示 评论

Glee 前端开发
请问作者还有优惠码吗？想买一份
▲ 0 评论 1月前

Paradise不想说话
看到作者是程墨就立马来阅读了 果然不会让人失望
▲ 0 收起评论 1月前

> 程墨 Hulu
> 谢谢支持，您太客气了。
> 1月前
>
> 胸怀文若谷 前端 @ 国际庄某养猪场
> 哈哈，有点意思。
> 1月前
>
> 评论审核通过后显示 评论

Stanley酱
无法返回上一级？
▲ 0 评论 1月前

花狗是我 前端 @ E园
react 调用子组件的方法，有什么优雅的方式吗？不通过 给子组件绑定 ref
▲ 0 收起评论 1月前

> 程墨 Hulu
> 最好不要直接调用子组件的方法，如我所说，组件之间的“语言”是props，应该用props来传递消息。
> 1月前
>
> 花狗是我 前端 @ E园
> 回复 程墨：嗯，了解了
> 1月前
>
> zhangyanling77 前端开发 @ 成都
> 回复 程墨：但是比如我的一个子组件内部的状态我需要一个函数来收集，然后供父组件调用。这个时候是没有触发动作的。加载完毕就需要获取子组件的这些状态，那么我也只能使用ref吧。不知道大佬有没有好的办法
> 1月前
>
> casablanca
> 回复 zhangyanling77：用props, 事件， redux应该都可以
> 1月前 回复
>
> 评论审核通过后显示 评论

Farris 前端工程师
言简意赅，前排点赞~~
▲ 0 收起评论 1月前

> 程墨 Hulu
> 谢谢
> 1月前
>
> 评论审核通过后显示 评论

酱菜家
通俗易懂
▲ 0 收起评论 1月前

> 程墨 Hulu
> 感谢评价
> 1月前
>
> 评论审核通过后显示 评论