# React的事件系统

事件系统，react基于virtual dom实现了一个syntheticEvent（合成事件）层，我们所定义的事件处理器会接收受到一个syntheticEvent对象的实例，他完全符合w3c标准，不会存在任何ie标准的兼容性问题。并且与原生的浏览器事件一样拥有同样的接口，同样支持事件冒泡机制，我们可以阻止冒泡和阻止默认行为。

1. 合成事件

合成事件的绑定方式<button onClick={this.handleClick}>Test</button>

1. 合成事件的实现机制

在react底层，主要对合成事件做了两件事，事件委派和自动绑定。

1. 事件委派

事件委派不会把事件处理函数直接绑定到真实节点上，而是把所有事件绑定到结构的最外层，使用一个统一的事件监听器，这个事件监听器上维持了一个映射来保存所有组件内的事件监听和处理函数。 当事件发生时，首先被这个统一的事件监听器处理，然后映射里找到真正的事件处理函数并调用。这样做简化了事件处理和回收机制，效率也有很大提升。

1. 自动绑定

组件中每个方法的上下文都会指向该组件的实例，即自动绑定this为当前组件。而react还会对这种引用进行缓存，以达到cpu和内存的最优化。

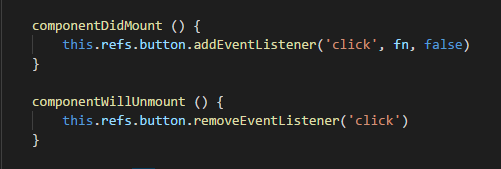
在使用es6 classes或者纯函数时，这种自动绑定就不复存在了，我们需要手动实现this的绑定。

四种方法：双冒号（stage-0），bind，构造器内声明，箭头函数

Ps：除箭头函数外的方法后加（）调用，是立即调用。

1. 在react中使用原生事件

react提供了合成事件，并不是意味着不能使用原生事件，在componentDidMount中存在真实DOM中可完成原生事件绑定



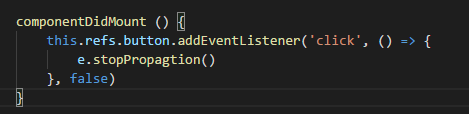
Ps：在react中使用dom原生事件时，一定要在组件卸载时手动移除，否则可能出现内存泄漏的问题。而使用合成事件系统则不需要，因为react以及帮我们妥善处理了。

四、合成事件和原生事件混用

React合成事件系统好处多，但是很多场景下只能借助原生事件才能完成，比如我们无法将事件绑定到body上，因为body时在组件范围外的，只能使用原生事件。

尽量避免在react中混用合成事件和原生DOM事件。

阻止react事件冒泡的行为只能用于react合成事件系统中，没办法阻止原生事件的冒泡。如：



反之，在原生事件中阻止冒泡行为，却可以阻止react合成事件的传播。

1. 对比react合成事件和javascript原生事件
   1. 事件传播与组织事件传播
   2. 事件类型
   3. 事件绑定方式
   4. 事件对象