而且React能够批处理虚拟DOM的刷新，在一个事件循环（Event Loop）内的两次数据变化会被合并，例如你连续的先将节点内容从A变成B，然后又从B变成A，React会认为UI不发生任何变化，而如果通过手动控制，这种逻辑通常是极其复杂的。尽管每一次都需要构造完整的虚拟DOM树，但是因为虚拟DOM是内存数据，性能是极高的，而对实际DOM进行操作的仅仅是Diff部分，因而能达到提高性能的目的。这样，在保证性能的同时，开发者将不再需要关注某个数据的变化如何更新到一个或多个具体的DOM元素，而只需要关心在任意一个数据状态下，整个界面是如何Render的。

如果你像在90年代那样写过服务器端Render的纯Web页面那么应该知道，服务器端所要做的就是根据数据Render出HTML送到浏览器端。如果这时因为用户的一个点击需要改变某个状态文字，那么也是通过刷新整个页面来完成的。服务器端并不需要知道是哪一小段HTML发生了变化，而只需要根据数据刷新整个页面。换句话说，任何UI的变化都是通过整体刷新来完成的。而React将这种开发模式以高性能的方式带到了前端，每做一点界面的更新，你都可以认为刷新了整个页面。至于如何进行局部更新以保证性能，则是React框架要完成的事情。

虚拟DOM(virtual-dom)不仅带来了简单的UI开发逻辑，同时也带来了组件化开发的思想，所谓组件，即封装起来的具有独立功能的UI部件。React推荐以组件的方式去重新思考UI构成，将UI上每一个功能相对独立的模块定义成组件，然后将小的组件通过组合或者嵌套的方式构成大的组件，最终完成整体UI的构建。

这里大家可能会奇怪，为什么script的type是text/jsx,这是因为 React 独有的 JSX 语法，跟 JavaScript 不兼容。凡是使用 JSX 的地方，都要加上 type="text/jsx" 。 其次，React 提供两个库： react.js 和 JSXTransformer.js ，它们必须首先加载。其中，JSXTransformer.js 的作用是将 JSX 语法转为 JavaScript 语法。这一步很消耗时间，实际上线的时候，应该将它放到服务器完成。

React.render 是 React 的最基本方法，用于将模板转为 HTML 语言，并插入指定的 DOM 节点。

React.render(

<h1>Hello,world!</h1>,

document.getElementById(‘container’);

);

这里需要注意的是，react并不依赖jQuery，当然我们可以使用jQuery，但是render里面第二个参数必须使用JavaScript原生的getElementByID方法，不能使用jQuery来选取DOM节点。

前面说了，ReactJS是基于组件化的开发，React.createClass 方法就用于生成一个组件类。

不过这里有几点需要注意：

1）获取属性的值用的是this.props.属性名

2）创建的组件名称首字母必须大写。

3）为元素添加css的class时，要用className。

1. 组件的style属性的设置方式也值得注意，要写成style={{width: this.state.witdh}}。

3、组件的生命周期

组件的生命周期分成三个状态：

Mounting：已插入真实 DOM

Updating：正在被重新渲染

Unmounting：已移出真实 DOM

React 为每个状态都提供了两种处理函数，will 函数在进入状态之前调用，did 函数在进入状态之后调用，三种状态共计五种处理函数。

componentWillMount()

componentDidMount()

componentWillUpdate(object nextProps, object nextState)

componentDidUpdate(object prevProps, object prevState)

componentWillUnmount()

此外，React 还提供两种特殊状态的处理函数。

componentWillReceiveProps(object nextProps)：已加载组件收到新的参数时调用

shouldComponentUpdate(object nextProps, object nextState)：组件判断是否重新渲染时调用

六、ReactJS小结

关于ReactJS今天就先学习到这里了，下面来总结一下，主要有以下几点：

1、ReactJs是基于组件化的开发，所以最终你的页面应该是由若干个小组件组成的大组件。

2、可以通过属性，将值传递到组件内部，同理也可以通过属性将内部的结果传递到父级组件(留给大家研究)；要对某些值的变化做DOM操作的，要把这些值放到state中。

3、为组件添加外部css样式时，类名应该写成className而不是class;添加内部样式时，应该是style={{opacity: this.state.opacity}}而不是style="opacity:{this.state.opacity};"。

4、组件名称首字母必须大写。

5、变量名用{}包裹，且不能加双引号。