1、最后一个script的type为text/label，因为React独有的JSX语法和javascript不兼容，凡是有JSX的地方都需加上text/babel。

2、三个库必须先加载：react.js是React的核心库、react-dom.js提供与dom相关的功能、Browser.js将JSX语法转为JavaScript语法，这一步很耗时间，实际上线的时候应该将它放到服务器上完成。

3，分号、逗号不能乱加

4、核心思想：封装组件

5、要有闭合标签

基本方法

1. ReactDOM.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。只能有一个顶级标签。Dom节点需要添加自定义属性以data-前缀。

二、JSX语法

HTML语言直接写在Javascript语言中，不加任何引号，这就是JSX语法，它允许HTML与JavaScript的混写。

语法规则：遇到HTML标签（以<开头），就用HTML规则解析；遇到代码块（以{开头），就用JavaScript规则解析。可以在 JSX 中使用 JavaScript 表达式。表达式写在花括号**{}**中。JSX 中不能使用**if else**语句，单可以使用**conditional (三元运算)**表达式来替代

JSX允许直接在模板插入JavaScript变量。如果变量是一个数组则会展开这个数组的所有成员。

1. 组件

组件中存在很多方法，各个方法的具体使用详情请查看组件API

React允许将代码封装成组件，然后像插入普通HTML标签一样，在网页中插入这个组件。React.createClass方法就用于生产一个组件类。所有的组件类都必须有自己的render方法，用于输出组件，需要有返回值。

组件类的第一个字母必须大写，组件类只能包含一个顶层标签（用一个最外层的标签将子元素包起来，否则会报错）。

添加组件属性，有一个地方需要注意，就是 class 属性需要写成 className ，for 属性需要写成 htmlFor ，这是因为 class 和 for 是 JavaScript 的保留字。

this.props.attr获取组件的属性

四、this.props.children组件的所有子节点（组件相关）

this.props.children 的值有三种可能：如果当前组件没有子节点，它就是 undefined ;如果有一个子节点，数据类型是 object ；如果有多个子节点，数据类型就是 array 。所以，处理 this.props.children 的时候要小心。

React 提供一个工具方法 React.Children 来处理 this.props.children 。我们可以用 React.Children.map 来遍历子节点，而不用担心 this.props.children 的数据类型是 undefined 还是 object。

五、PropTypes（组件相关）

组件的属性可以接受任意值，字符串、对象、函数等等都可以。有时，我们需要一种机制，验证别人使用组件时，提供的参数是否符合要求，当向 props 传入无效数据时，JavaScript 控制台会抛出警告

组件类的PropTypes属性，就是用来验证组件实例的属性是否符合要求，相关验证器查看API

六、获取真实的dom节点

组件并不是真实的dom节点，而是存在于内存之中别的一种数据结构，叫做虚拟的DOM。只有当它插入文档以后，才会变成真实的DOM。根据React的设计，所有的DOM变动，都现在虚拟的DOM上发生，然后在将实际发生变动的部分，反映在真实DOM上，这种算法叫做DOMdiff，它可以极大提高网页的性能表现。但是有时需要从组件获取真实DOM的节点，这时就要用到ref属性。This.refs.ref的属性值就会返回这个真是的DOM节点

注意：This.refs.ref的属性值获取的是真实的节点，所以必须等到虚拟节点插入文档以后才能使用这个属性，

七、this.state

组件免不了要与用户互动，React的一大创新就是将组件看成是一个状态机，一开始有一个初始状态，然后与用户互动，导致状态变化，从而触发重新渲染UI。

getInitialState方法用于定义初始状态，也就是一个对象，这个对象可以通过this.state属性读取。当用户点击组件，导致状态变化，this.setState方法就修改状态值，每次修改以后，自动调用this.render方法，再次渲染组件

由于this.props和this.state都用于描述组件的特性，可能会产生混淆，一个简单的区分方法是，this.props表示那些一旦定义就不在改变的特性，而this.state是会随着用户的互动而产生变化的特性。

组合使用 state 和 props 。我们可以在父组件中设置 state， 并通过在子组件上使用 props 将其传递到子组件上

getDefaultProps 方法可以用来设置组件属性的默认值。

getInitialState 方法用于定义初始状态

八、表单

用户在表单中填入内容，属于用户和组件的互动，所以不能用props读取，而是要定义一个onChange事件的回调函数，event为回调函数的参数，通过event.target.value读取用户输入的值。Textarea元素，select元素、radio元素都属于这种情况。

可以使用onChage时间进行监听input的变化

九、组件的生命周期

组件的周期分为三个状态：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();在渲染前调用，在客户端也在服务端

componentWillUpdate(object,nextProps,object,nextState);

**componentDidMount();**在第一次渲染后调用，只在客户端。之后组件已经生成了对应的DOM结构，可以通过this.getDOMNode()来进行访问。 如果你想和其他JavaScript框架一起使用，可以在这个方法中调用setTimeout, setInterval或者发送AJAX请求等操作(防止异部操作阻塞UI)。

componentDidUpdate(object,nextProps,object,nextState);

componentWillUnmount();

十、Ajax

组件的数据来源，通常是通过 Ajax 请求从服务器获取，可以使用 componentDidMount 方法设置 Ajax 请求，等到请求成功，再用 this.setState 方法重新渲染 UI

十一、相关方法简介

setState设置组件状态：

不能在组件内部通过this.state修改状态，因为该状态会在调用setState()后背替换掉，

setState()并不会立即改变this.state，二十创建一个即将处理的state，setState（）会触发一次组件的重绘。

SetProps设置组件属性

Prop相当于组件的数据流，会从父组件向下传递至所有的子组件中，