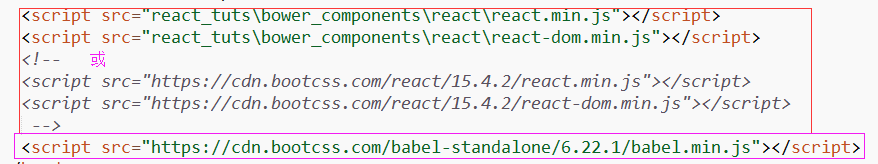
**第一个react网站：**

1. **引入react.js react-dom.js bowerser.js 三个js文件**



1. **自己的js文件：**

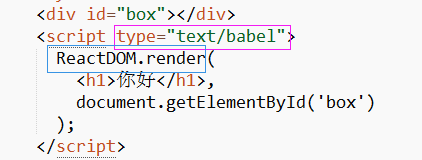
**<script type="text/bable">**

**<!-- 标签的 type 属性为 text/babel 。**

**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/babel" 。 -->**

**</script>**



## 安装cnpm

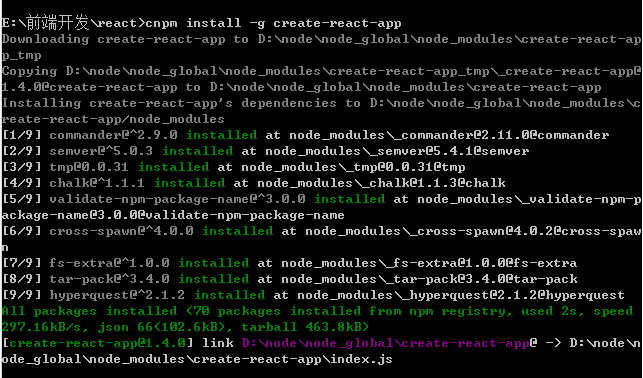
$ npm install -g cnpm --registry=https://registry.npm.taobao.org

$ npm config set registry https://registry.npm.taobao.org



## 构建 React 开发环境

$ cnpm install -g create-react-app



$ create-react-app my-app

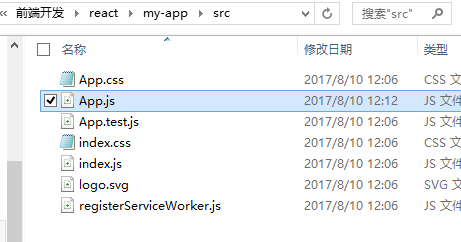




$ cd my-app/

$ npm start

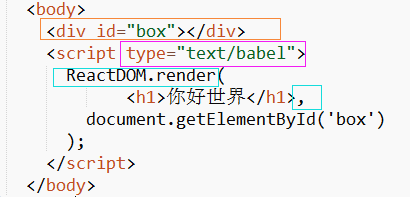
修改my-app\src\App.js



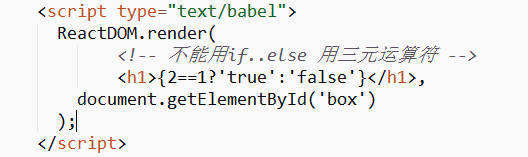


## JSX

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ReactDOM.render()\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*三元运算符\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

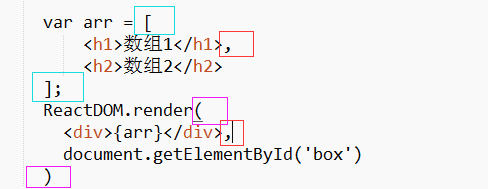


\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*样式\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*数组\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



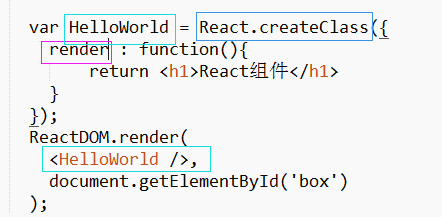
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*注释 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

{/\*注释部分\*/}

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*html标签 React组件\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*



## React组件



## React Props(不可变的状态)

state 和 props 主要的区别：

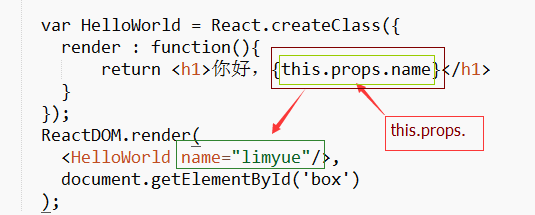
props 是不可变的，

state 可以根据与用户交互来改变。（下面紧接着有state用法）

这就是为什么有些容器组件需要定义 state 来更新和修改数据。

而子组件只能通过 props 来传递数据。

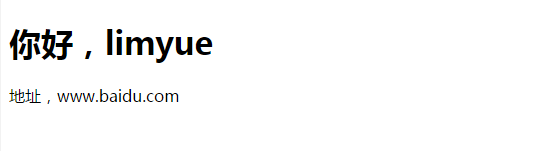
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*





\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

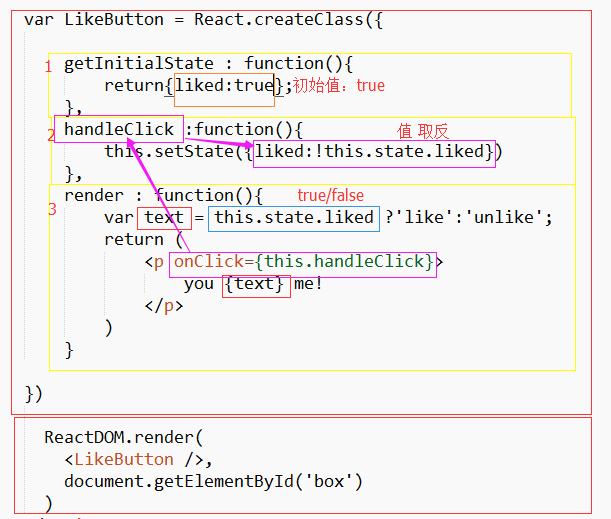


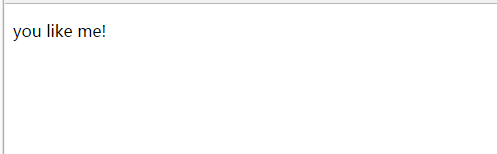


## React State(状态)

React 把组件看成是一个状态机（State Machines）。通过与用户的交互，实现不同状态，然后渲染 UI，让用户界面和数据保持一致。

React 里，只需更新组件的 state，然后根据新的 state 重新渲染用户界面（不要操作 DOM）。 this.state







## props验证

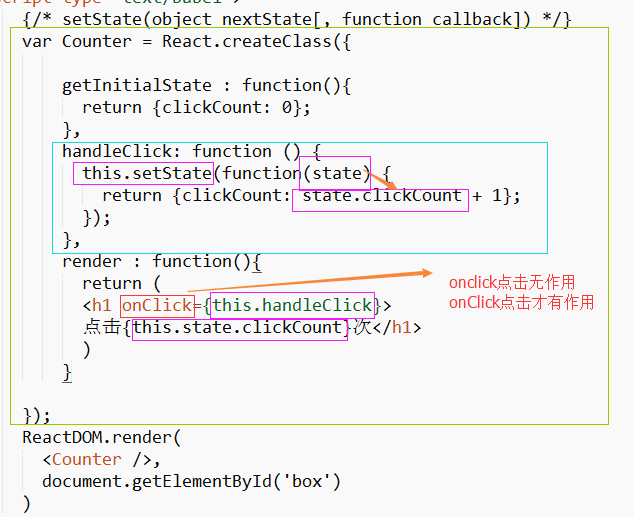
随着应用不断变大，保证组件被正确使用变得非常有用。为此我们引入propTypes

<http://www.runoob.com/react/react-props.html>

## React 组件 API

### 设置状态：setState

setState(object nextState[, function callback])



### 替换状态：replaceState

### 设置属性：setProps

### 替换属性：replaceProps

### 强制更新：forceUpdate

### 获取DOM节点：findDOMNode

### 判断组件挂载状态：isMounted

## 组件生命周期（3个）

### Mounting：已插入真实 DOM

### Updating：正在被重新渲染

### Unmounting：已移出真实 DOM

<http://www.runoob.com/react/react-component-life-cycle.html>

