React是前端主流框架之一，起源于Facebook内部项目，

## 用react脚手架快速搭建搭建react项目

1. 全局安装create-react-app

Npm install -g create-react-app

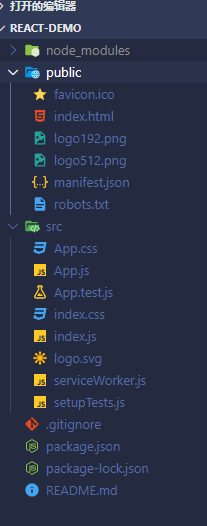
Create-react-app xxx



1. 或直接使用npx安装（[Node >= 8.10 和 npm >= 5.6](https://nodejs.org/en/" \t "_blank)）

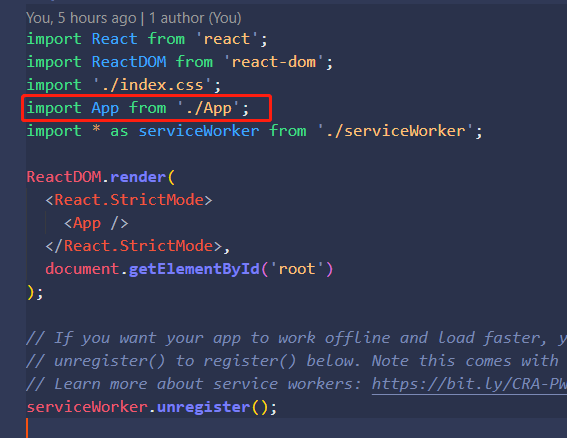
npx create-react-app my-app

项目目录：



App.js全局组件入口文件类似vue的app.vue

Index.js js入口文件类似类似vue的main.js



引入组件时 组件名首字母必须大写

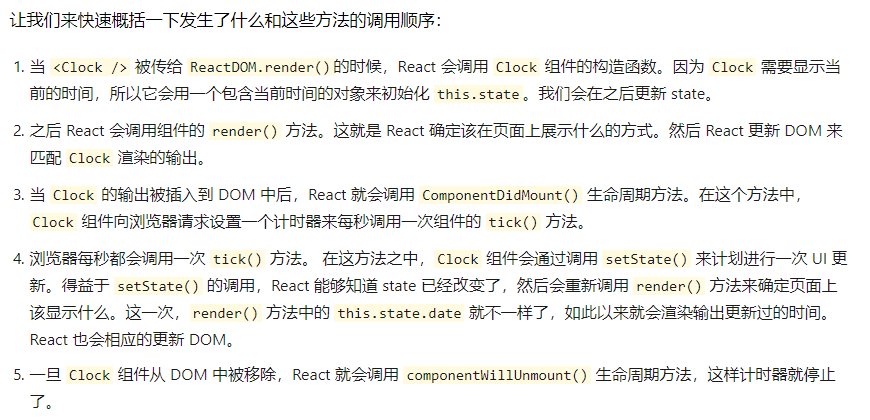
组件中的标签必须包含在一个大的dom元素中：和vue模板中的根div一样；

Fragment空白占位dom

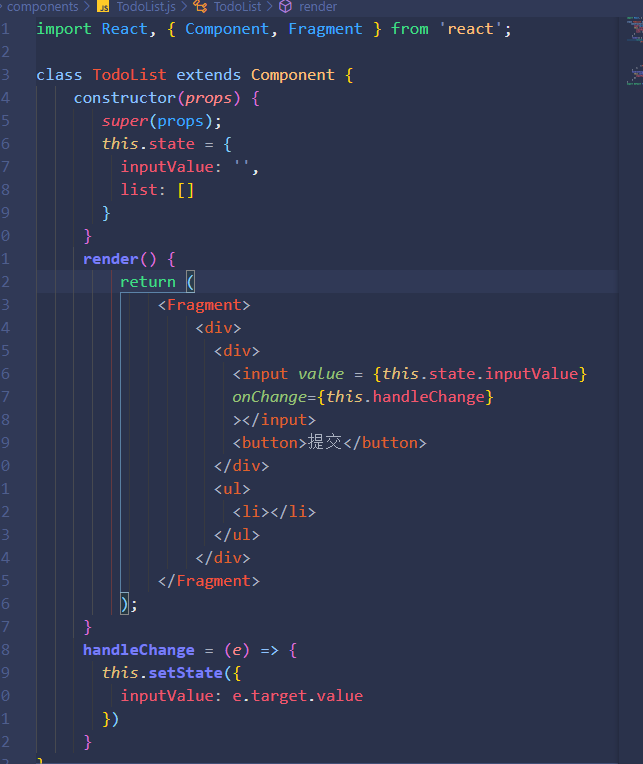


React 元素是[不可变对象](https://en.wikipedia.org/wiki/Immutable_object" \t "_blank)。一旦被创建，你就无法更改它的子元素或者属性。一个元素就像电影的单帧：它代表了某个特定时刻的 UI。

根据我们已有的知识，更新 UI 唯一的方式是创建一个全新的元素，并将其传入 [ReactDOM.render()](https://zh-hans.reactjs.org/docs/react-dom.html" \l "render)。

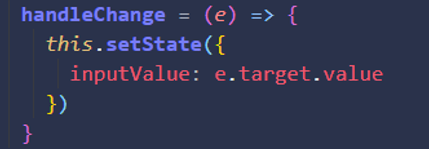
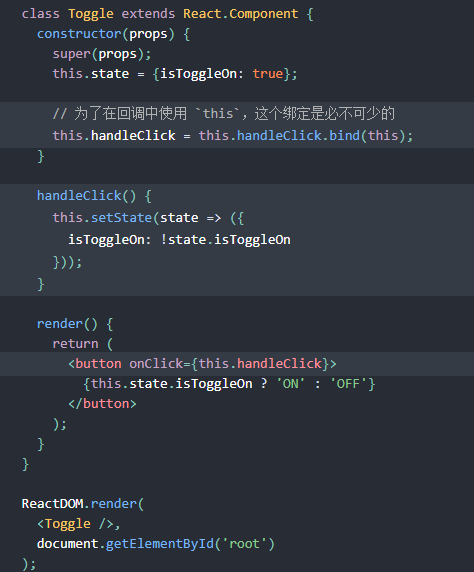


**State与事件处理：**

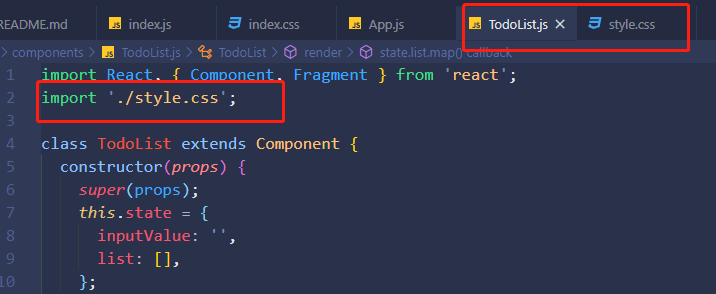


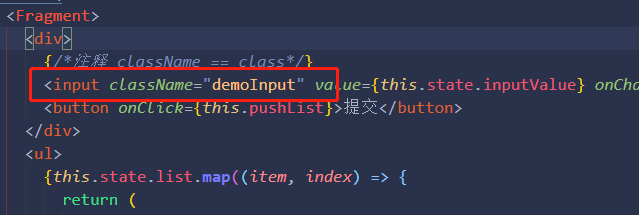
只能通过setState来改变state中的值；

事件函数执行时会有this指向问题，目前有三种方式来解决：（事件是小驼峰）

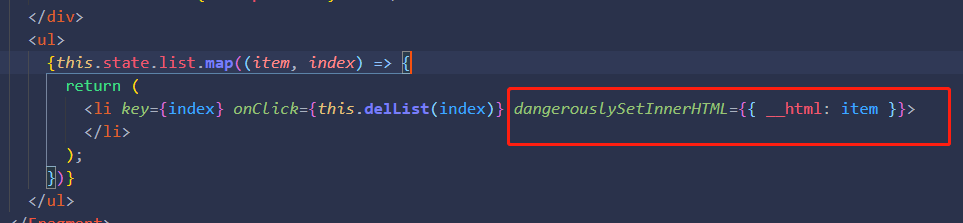
1. 
2. 
3. 
4. 

样式文件的引入与className：



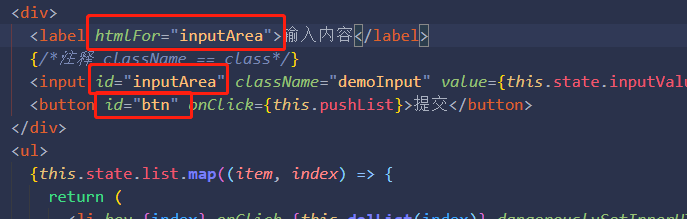


小知识：



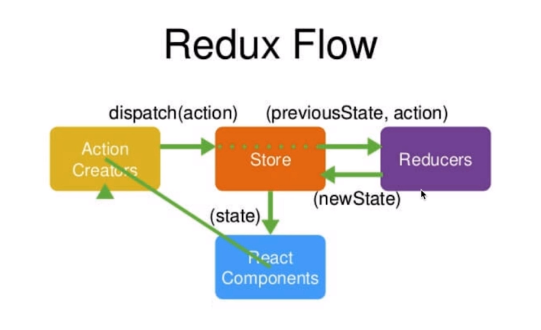
可保留输入的html标签；

htmlFor：事件转移



**列表与Key**

使用map()函数



Action creators（动作，指令）

Store（仓库）

Reducers（处理数据更新逻辑）

Components（组件）

1：创建action

2：定义reducers

3：创建store（reducer）传入reducer

4：ReactDOM.render(

<Provider store={store}>

<App />

</Provider>,

document.getElementById('root')

)

使所有组件都能访问store；

React-Redux 提供Provider组件，可以让容器组件拿到state。上面代码中，Provider在根组件外面包了一层，这样一来，App的所有子组件就默认都可以拿到state了。

五：mapStateToProps；将state映射到组件的props

function mapStateToProps(state) {

return {

value: state.count

}

}

六：// 将action映射到组件的props

function mapDispatchToProps(dispatch) {

return {

onIncreaseClick: () => dispatch(increaseAction)

}

}

七：export default connect(

mapStateToProps,

mapDispatchToProps

)(Counter)