# **React**

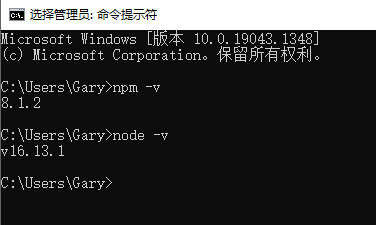
React 官网： https://reactjs.org/

## 安装

1. Node Js 安装

Download from <https://nodejs.org/en/download/> , 安装成功后，通过

npm -v 和 node -v 命令去检查是否安装完成

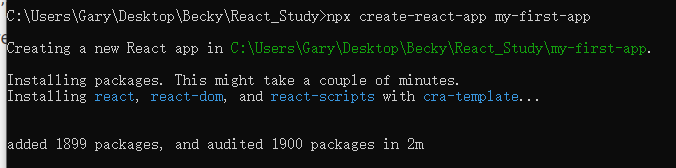


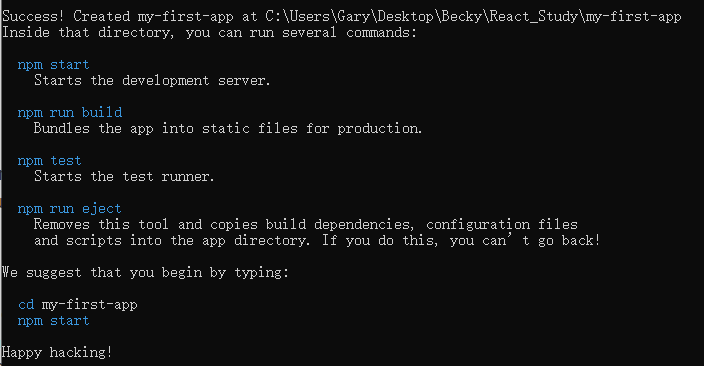
1. 通过create-react-app 去build a new single-page application

Cmd is :

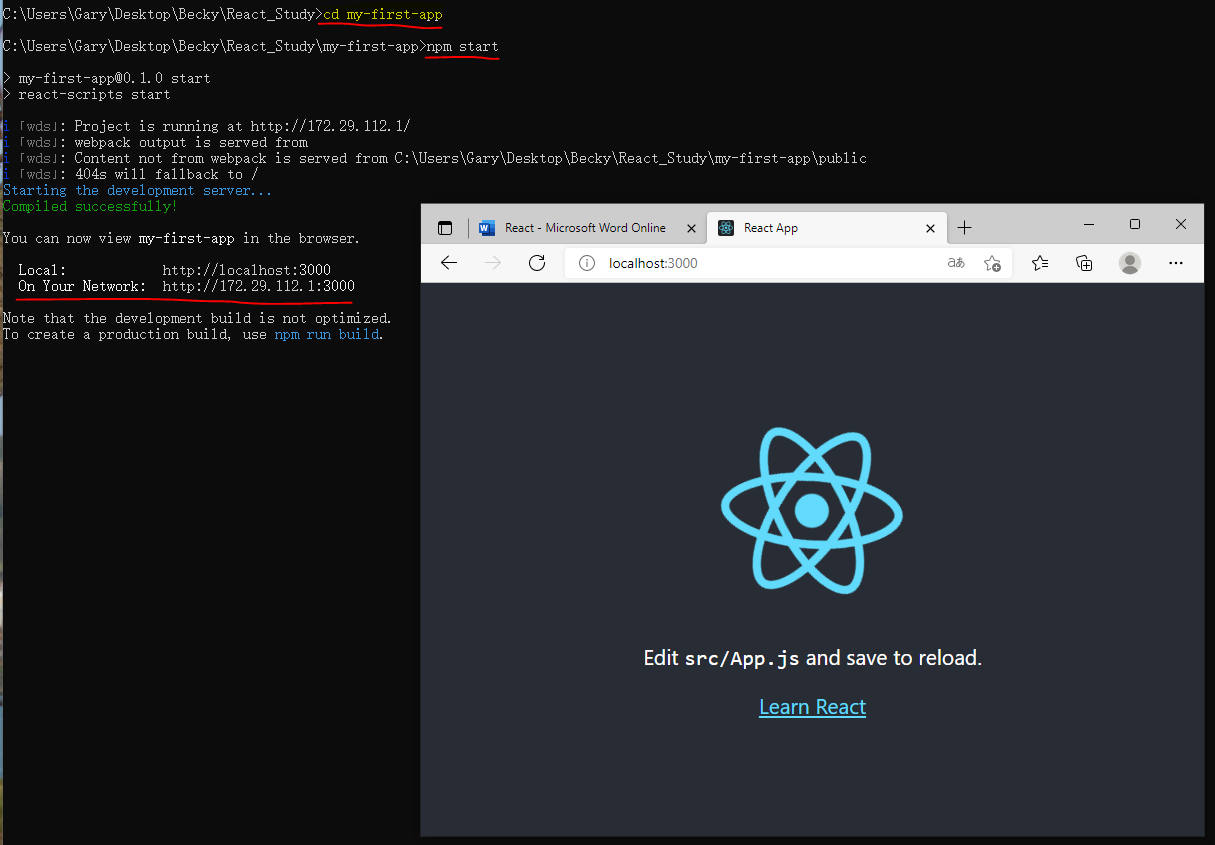
(1) cd target\_folder

(2) npx create-react-app new\_application\_name



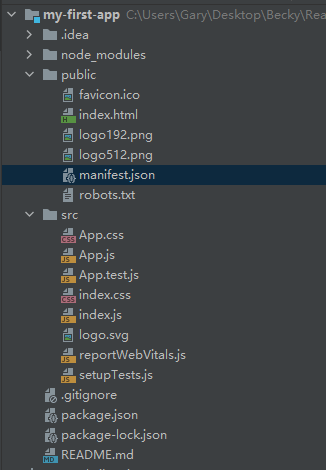


1. 启动应用

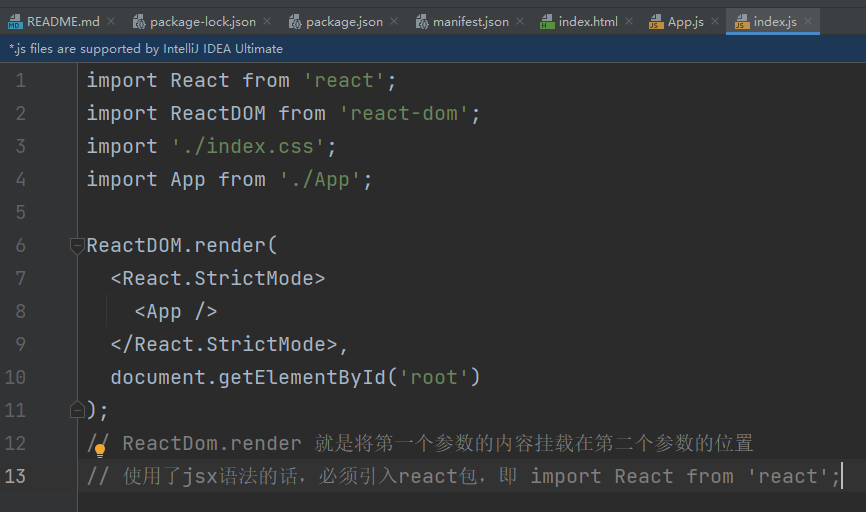


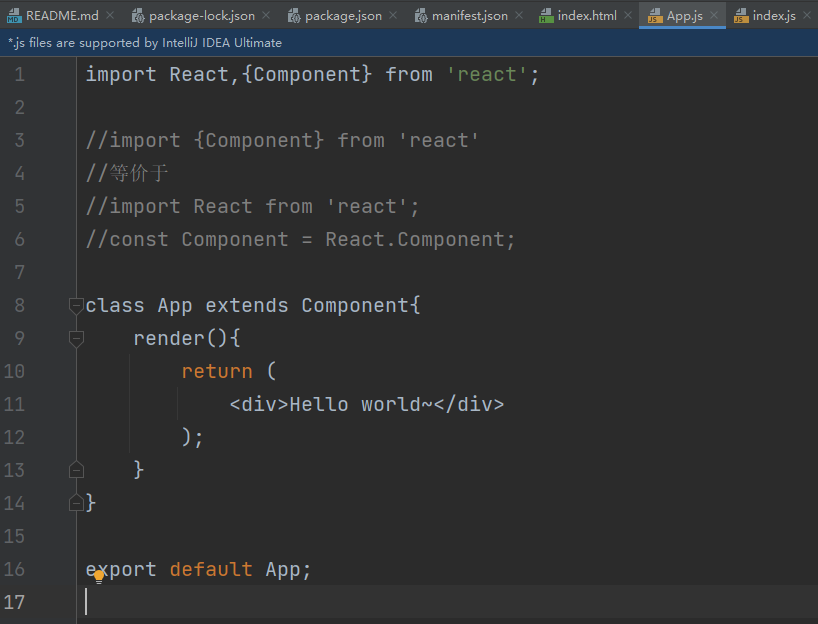
React 工程目录介绍：

* package.json 文件是node的包文件，可以让项目变成node的包
* node\_modules 存放我们以来的包，是脚手架的第三方模块依赖
* public/manifest.json 指定了开始页面 index.html(start\_url)，一切的开始都从这里开始
* public/index.html 通常是网页入口
* src目录 存放项目的源代码，src/index.js是整个程序运行的入口文件



最简单的组件





# JSX 语法

1. 菜鸟教程： https://www.runoob.com/react/react-jsx.html







* JSX 语法中使用js的语法或者变量，需要使用{}包含

<input value={this.state.inputValue} />

* <Fragment> 是JSX里面的组件，JSX 里面以大写字母开头的通常是组件，小写字母开头的通常是普通元素
* JSX 的注释 格式是 {/\*comment.....\*/} 或者

{

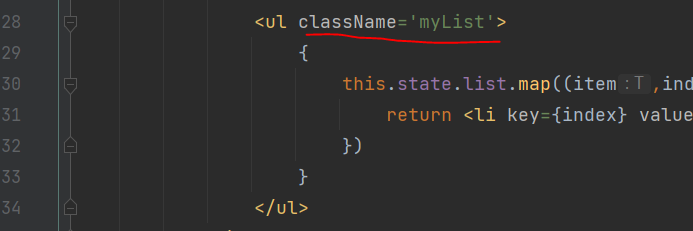
//

}

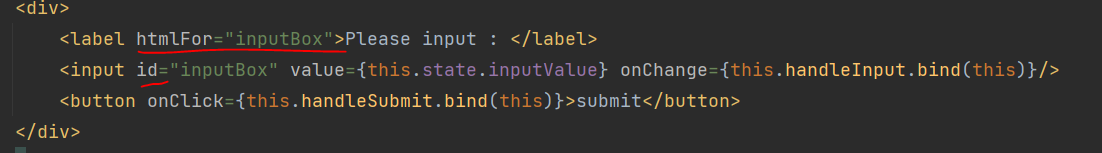
注释不显示在页面上,也不会变成任何html 元素



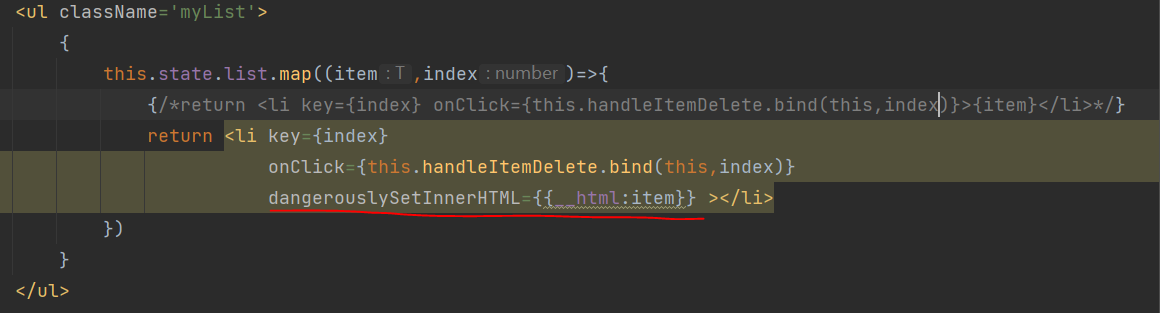
* 在JSX 中要制定样式时，需要使用className=’xxx’ 属性绑定，并非class(因为这是关键字)



* 如果要使用html的for属性，需要使用htmlFor=’xxx’ 属性进行绑定，并非for(for 是关键字)



* 如果在jsx中想显示的内容不被转移的话，就设置 dangerouslySetInnerHTML 属性，外层{} 表示的是只想着一个JSX 对象，里层{} 表示这是一个数据对象



# 组件

#### 自定义组件定义

import ***React***,{Component,***Fragment***} from 'react';

//必须导入React 跟父类Component  
  
class 类名(即组件名) extends Component{

//继承父类Component

render(){

//return 内容写在括号内，并且需要用容器元素包含，如div，span

// Fragment 是一个容器元素，相当于占位符，并不会在源代码中输出任何html元素  
 return (….)

}  
  
}  
  
export default ToDoListSimple;

//将新组建导出，让其他地方可以应用得到



#### 构造函数 constructor()

1. 构造函数优于其他函数会被最先执行,
2. 构造函数必须接受props参数输入
3. 构造函数必须调用父类的构造函数
4. 组件的数据保存在react的状态中，即this.state



#### React state(状态)

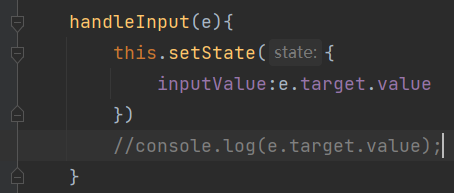
1. 如果需要改变状态数据，需要通过setState方法，不能直接this.state去更改

#### 组件自定义方法

方法名(e){

// e.target 可以获取得到事件触发元素，e.target.value 就是input 这类元素的value属性

}



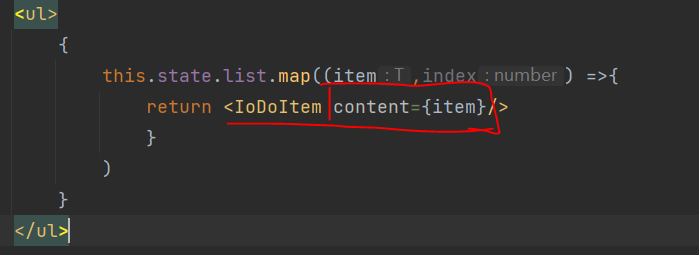
#### 事件绑定

React的事件绑定，js 中的onchange 要变成onChange(方法名变成驼峰规则)。事件绑定是可以通过bind将this（react组件）对象传进去更改方法内部的this指向, 需要的参数也是通过bind 方法传到方法里面。通常方法.bind(this)的绑定放在构造函数内



* 父组件向子组件传递数据

1. 父组件向子组件可通过自定义的属性进行传递
2. 子组件通过this.props.属性名 获取数据



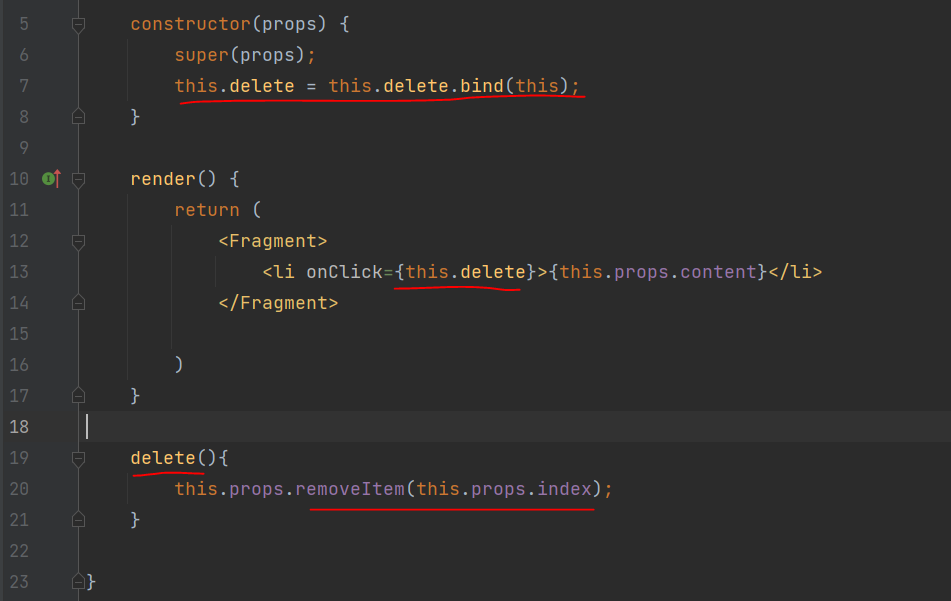


* 子组件调用父组件方法改变父组件数据

1. 在父类中，将父类方法通过子类控件属性并绑定this将方法传递给子控件



1. 在子类中，通过属性名(即方法名)调用父类方法



结构赋值

函数体