复习:

见思维导图

面试题:三大框架中数据绑定实现上有何绑定?

1.React的第二种使用方法 —— 脚手架方式

①前提: Node.js > 8.10

②下载全局脚手架工具

npm i -g create-react-app

③运行全局脚手架工具,创建空白项目

create-react-app 项目名

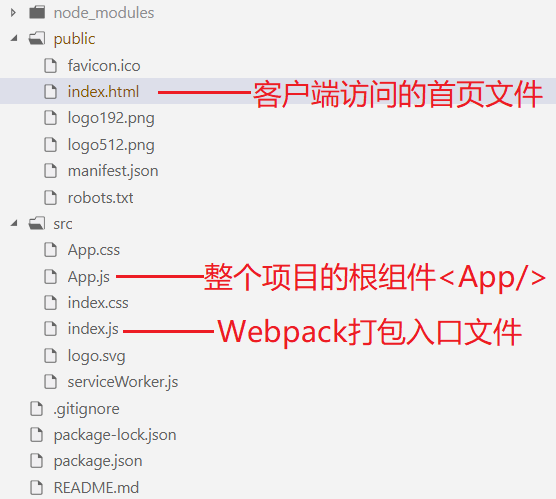
④进入空白项目,运行开发服务器

cd 项目名

npm start

⑤客户端访问测试

http://127.0.0.1:3000



练习: 创建自定义组件 MyC01BuyCount.js (购物计数器), 在根组件App.js中使用该组件.

2.React中的数据绑定

①内容绑定 —— <p>{表达式}</p>

②属性绑定 —— <p title={表达式}></p>

③指令绑定 —— React中没有指令

④事件绑定 —— 与其它框架区别较大！

⑤双向数据绑定 ——与其它框架区别较大！

React中的事件句柄不同于HTML中的事件句柄(handler)：

onclick —— onClick

ondblclick —— onDoubleClick

onmouseover —— onMouseOver

onblur —— onBlur

......

|  |  |  |
| --- | --- | --- |
| Vue.js中的事件绑定  <p @click="f2">  <p @click="f2()"> | Angular中的事件绑定  <p (click)="f2()"> | React中的事件绑定  <p onClick={this.f2}> |

总结：React中事件绑定需要注意：

①事件句柄名称必须用“小驼峰命名法”

②事件处理函数不能用引号

③事件处理方法前必须加 { }

④当前class内部的方法做事件处理方法，则必须加 this

⑤事件处理方法名后不能加 ( )

面试题1：React中事件处理方法中为什么this默认指向undefined？

JSX中，事件处理方法赋值时不是函数调用，而是函数的引用赋值 —— this指向不再指向当前组件对象！而是指向当前ES6模块中的全局this——值为undefined

面试题2：如何解决React事件处理方法中无法使用this的问题？——用三种方法

**①用匿名函数封装立即调用的函数**

f2( ){

console.log(this)

}

<p onClick={ ( )=>{ this.f2( ) } } >

**②用箭头匿名函数代替function匿名函数**

f2 = ( )=>{

console.log(this)

}

<p onClick={ this.f2 } >

**③在构造方法中固定this的指向**

constructor( ){

super( )

this.f2 = this.f2.bind( this )

}

f2( ){

}

<p onClick={ this.f2 } >

3.React中组件里的Model数据

React中class组件可以声明两种形式的Model数据：

**①class属性数据 —— 可以一次性绑定到模板，数据再变则不会再次修改视图模板**

class MyC02{

count =3

}

**②组件状态数据 —— 可以绑定到模板，未来Model变则View变**

使用步骤：

1)创建状态数据——只能在构造方法中

constructor(){

super()

this.state = { age: 20 }

}

2)绑定状态数据

<p>{this.state.age}</p>

3)修改状态数据

this.setState({

age: 21

})

注意：①setState()会修改虚拟DOM树 ②setState()不会清空已有的状态数据，只会把新修改的数据合并到已有数据中 ③setState()是异步操作，不会立即执行——底层会把连续的多次setState合并为一次操作，以减少虚拟DOM/真实DOM的修改；如果希望查看修改后的值，可以使用setState({ }, callback)中的回调函数 ④不推荐直接在setState中使用之前的this.state数值，使用临时变量代替

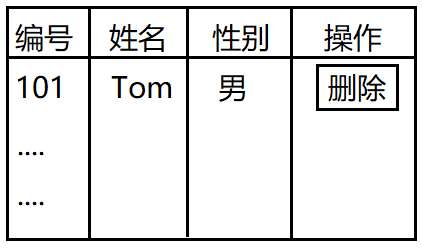
练习：创建一个组件MyC04，其中有状态数据：empList = [

{eid: 101, ename: 'Tom', sex:1},

{eid: 102, ename: 'Mary', sex:1},

{eid: 103, ename: 'Jerry', sex:1},

]，把这些数据绑定到一个TABLE中，点击删除可以删除该记录，如下所示：



4.React中的双向数据绑定

|  |  |  |
| --- | --- | --- |
| Vue.js中的双向数据绑定 | Angular中的双向数据绑定 | React中的双向数据绑定 |
| v-model | [(ngModel)] | 受控组件 |

提示：双向数据绑定只能用于“表单元素”，如input、textarea、select等

**方向1：Model => View(value)**

constructor(){

this.state = { uname: '' }

}

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

至此，输入框中会呈现状态数据，但是无法再修改——受控组件

**方向2：View(value) => Model**

<input onChange={this.doChange}/>

doChange = (event)=>{

this.setState({

uname: event.target.value

})

}

|  |
| --- |
| 受控组件：一旦将输入元素的value绑定为状态数据，用户在其中的输入都会被重置为状态数据的当前值（感觉像是只读输入框一样）—— 只有通过setState修改了状态数据，输入框的内容才能改变；  接下来只需要监听输入框的onChange事件，捕获到事件源当前的最新值(用户的最新输入)，修改为状态数据即可。 |

课后任务：

(1)总结比较三大框架中数据绑定、事件绑定、双向数据绑定实现上的异同

(2)完成“待办事项列表”应用：

