Angular

一：Angular 的基本知识:

1. 要遍历这些预定义的商品列表，请使用 \*[ngFor](https://www.angular.cn/api/common/NgForOf) 指令。
2. 并使用属性绑定语法 []把该链接的 title 设置成该商品的名字，
3. 插值表达式 {{}} 允许你把属性值渲染为文本；而属性绑定语法 [] 则允许你在模板表达式中使用属性值。
4. 添加商品说明。在 p 标签上，使用 \*[ngIf](https://www.angular.cn/api/common/NgIf) 指令，这样才能在当前商品有描述信息的情况下创建这个 p 元素。
5. 事件绑定是通过把事件名称包裹在圆括号 () 中完成的.

Angular 模板语法的五个常用特性：

* \*[ngFor](https://www.angular.cn/api/common/NgForOf)
* \*[ngIf](https://www.angular.cn/api/common/NgIf)
* 插值表达式 {{}}
* 属性绑定 []
* 事件绑定 ()
* React

React:用于构建用户界面的JavaScript库。

React 特点：声明式、组件化、一次学习，随处编写。

**简单组件**

React 组件使用一个名为 render() 的方法，接收输入的数据并返回需要展示的内容。

有状态组件

除了使用外部数据（通过 this.props 访问）以外，组件还可以维护其内部的状态数据（通过 this.state 访问）。当组件的状态数据改变时，组件会再次调用 render() 方法重新渲染对应的标记。

**应用**

使用 props 和 state，我们可以创建一个简易的 Todo 应用。在示例中，我们使用 state 来保存现有的待办事项列表及用户的输入。

更新元素渲染

React 元素都是不可变的。当元素被创建之后，你是无法改变其内容或属性的。

目前更新界面的唯一办法是创建一个新的元素，然后将它传入 ReactDOM.render() 方法：

**React JSX**

React 使用 JSX 来替代常规的 JavaScript。

JSX 是一个看起来很像 XML 的 JavaScript 语法扩展。

我们不需要一定使用 JSX，但它有以下优点：

* JSX 执行更快，因为它在编译为 JavaScript 代码后进行了优化。
* 它是类型安全的，在编译过程中就能发现错误。
* 使用 JSX 编写模板更加简单快速。

**React Props**

state 和 props 主要的区别在于 props 是不可变的，而 state 可以根据与用户交互来改变。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 事件处理**

React 元素的事件处理和 DOM 元素类似。但是有一点语法上的不同:

* React 事件绑定属性的命名采用驼峰式写法，而不是小写。
* 如果采用 JSX 的语法你需要传入一个函数作为事件处理函数，而不是一个字符串(DOM 元素的写法)

React 条件渲染

在React中的条件渲染和JavaScript中的一样，使用JavaScript运算符if或者条件运算符去创建元素来表现当前的状态，然后React根据他们来更新UI。

列表 &key

key帮助react 识别哪些元素改变了，比如被添加或被删除。因此你应当给数组每一个元素赋予一个确定的标识。

一个元素的key最好是这个元素在列表中独一无二的字符串，通常我们使用数据id来作为元素的key。

表单

在react中，HTML表单元素的工作方式和其他的DOM元素有些不同，这是因为表单元素通常会保持一些内部的state。