JSX

1，介绍jsx是什么

2，如何使用jsx

3，在jsx中使用变量

4，在jsx中使用样式

5，在jsx中循环输出

6，在jsx中使用三元表达式

7，在jsx中使用if-else和switch-case

8，在jsx中绑定事件

9，编写一个demo来应用学到的知识

Jsx是什么，如何使用jsx

HTML 语言直接写在 JavaScript 语言之中，不加任何引号，这就是 JSX 的语法，它允许 HTML 与 JavaScript 的混写。

**React.render(**

**<div>**<h1>Hello, world!</h1>**</div>,**

**document.getElementById('example')**

**);**

**在render函数中使用，即可**



在jsx中使用变量

如：

**var names = ‘小明’;**

**React.render(**

**<div>{names}</div>,**

**document.getElementById('example')**

**);**

**变量中的内容出现在了页面上**

****

上面代码体现了 JSX 的基本语法规则：遇到 HTML 标签（以 < 开头），就用 HTML 规则解析；遇到代码块（以 { 开头），就用 JavaScript 规则解析。JSX 允许直接在模板插入 JavaScript 变量。如果这个变量是一个数组，则会展开这个数组的所有成员:

**var arr = [**

**<h1>Hello world!</h1>,**

**<h2>React JSX</h2>,**

**];**

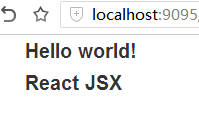
**React.render(**

**<div>{arr}</div>,**

**document.getElementById('example')**

**);**

**运行将得到**

****

**数组里的内容被自动展开了**

在jsx中使用样式

在jsx中，样式必须以对象的形式存在,并且被包裹在{{}}中

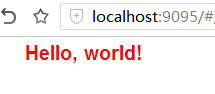
如：

**React.render(**

**<div>**<h1 style={{color:”red”}}>Hello, world!</h1>**</div>,**

**document.getElementById('example')**

**);**

****

**字体变成了红色**

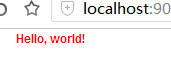
**如果是多个样式，则用逗号分隔,属性名不可带有- 如有-符号应该用驼峰格式**

**React.render(**

**<div>**<h1 style={{color:”red”,fontSize:”10px”}}>Hello, world!</h1>**</div>,**

**document.getElementById('example')**

**);**

****

**字体变红且变小了**

**除此之外，jsx也可以使用外部样式， 选择器规则如普通html一样**

**唯一不同是 jsx中不许使用class 因为class是js的关键字，应该用className 代替**

**<div className=“name”></div>**

在jsx中循环输出

**var names = ['Alice', 'Emily', 'Kate'];**

**React.render(**

**<div>**

**{**

**names.map(function (name) {**

**return <div>Hello, {name}!</div>**

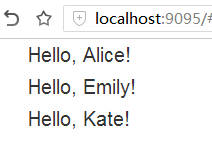
**})**

**}**

**</div>,**

**document.getElementById('example')**

**);**

****

**通过es6提供的map方法进行循环， 也可以用传统的for循环**

在jsx中使用三元表达式

Jsx中可以使用三元表达式来做简单的判断

**Var a = true;**

**React.render(**

**<div className={a ? ‘class1’: ‘class2’}>**

**{a ? ‘name’ : ‘title’}**

**</div>,**

**document.getElementById('example')**

**a = true 所以我们将会输出 name**

**className 则会等于class1**

在jsx中使用if-else和switch-case

在较为复杂的判断中，我们则需要用到if else 或switch或case

If不能直接写在｛｝中

需要写在外面 然后用变量来储存值，如果写在里面，需要用匿名函数来包裹

**var loginButton;**

**Var loggedIn = true;**

**if (loggedIn) {**

**loginButton = <span>提交</span>;**

**} else {**

**loginButton = <span>取消</span>;}**

**return (**

**<div>**

**{loginButton}**

**</div>**

**);**

**因为loggedIn 为true 我们将会得到 提交按钮**

**Var color = ‘red’**

**<p>**

**{(() => {**

**switch (color) {**

**case "red": return "#FF0000";**

**case "green": return "#00FF00";**

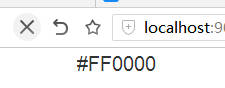
**case "blue": return "#0000FF";**

**default: return "#FFFFFF";**

**}**

**})()}**

**</p>**

**我们得到了ff000**

在jsx中绑定事件

在jsx中绑定事件和传统方式一样

**Test(){**

**Alert(1);**

**}**

**return (**

**<div>**

**<input type=”button” onClick={this.test.bind(this)} />**

**</div>**

**);**

**这样即可绑定事件**

**需要在事件上绑定当前的this .bind（this） 即可**

编写一个demo来应用学到的知识

使用jsx 定义一个 老师的列表 每一个老师下面又有一个学生的列表 并且循环输出他们