面试题：

1)前端三大框架的区别有哪些？

2)前端可用的异步请求方案有哪些？

|  |
| --- |
| 响应消息的Content-Type：  text/html  text/css  application/javascript  application/json  image/jpeg  audio/mpeg3  ..... |
| 请求消息的Content-Type：  text/plain：请求数据是未经编码的普通文本  application/x-www-form-urlencoded：请求数据是经过编码的数据  multipart/form-data：请求数据包含文件上传内容  application/json：请求数据是JSON格式的字符串 |

1.React框架概述

官网：https://reactjs.org/ 最新版V16.10

中文网：https://zh-hans.reactjs.org/

中文社区网：https://react-1251415695.cos-website.ap-chengdu.myqcloud.com/

由Facebook维护的MVVM框架，官方定义自己是一个“用于构建UI的JS库”——核心概念很少；但生态圈扩展非常广泛，如

React-Router：路由、

Redux： 状态保持、

ReactVR/360： 虚拟现实、

ReactNative：使用JS做App原生开发...

|  |
| --- |
| 项目中使用Vue.js中的方法：  ①SCRIPT标签直接引入  ②脚手架方式 |
| 项目中使用Angular中的方法：  ②脚手架方式 |
| 项目中使用React中的方法：  ①SCRIPT标签直接引入  ②脚手架方式 |

2.HTML元素属性(Attribute) vs JS DOM对象的属性(Property)

任何一个HTML元素都对应一个JSDOM对象，有两套属性系统

|  |  |
| --- | --- |
| **HTML元素属性** | **JS DOM对象的属性** |
| Attribute | Property |
| <img src="" id="" title=""> | img.src = ""  img.id = ""  img.title = "" |
| <img class=""> | img.className = "" |
| <label for=""> | label.htmlFor = "" |
| <td colspan=""> | 没有对应项 |
| 没有对应项 | div.innerHTML = ""  div.innerText = "" |

3.使用SCRIPT方式引入React

<div id="box"></div>

<script src="js/react.js"></script> 提供React对象

<script src="js/react-dom.js"></script> 提供ReactDOM对象

<script>

let el = React.createElement(标签名, 属性列表, 内容/子元素)

ReactDOM.render( el, box )

</script>

练习：使用React在#box中添加如下的DOM结构：

<div class="login">

<label for="uname">用户名</label>

<input type="text" id="uname"></input>

</div>

4.JSX —— 简化元素的创建过程 (取代React.createElement)

JavaScript XML**：形式上是XML，本质是JS对象**——可以看做是一种JS的变种（类似于TS）——浏览器无法理解JSX语法，必须使用编译器转化为JS才能被浏览器执行

JSX语法规范：

①JSX不是字符串！最外侧不能有引号！

②JSX形式上不是HTML语法，而是XML语法；故<br>必须写作<br/>；属性值必须用引号；有且只能有一个根元素

③JSX中可以使用HTML标签，但严格区分大小写——任何HTML标签都必须全小写，自定义组件名必须使用大驼峰法则，如<MyHeader></MyHeader>

④因为JSX中都是JS对象，所以属性都要使用DOM属性，而不是HTML属性，例如：<p className="..">

⑤JSX片段中还可以出现JSX表达式： { }

⑥JSX中的注释为： {/\* .... \*/}

练习：使用React在#box中添加如下的DOM结构——使用JSX语法代替React.create( )

<div class="login">

<label for="uname">用户名</label>

<input type="text" id="uname">

</div>

|  |
| --- |
| Babel：是一个第三方提供的JS变种编译器，可以把ES/TS/JSX转化为标准的JS代码，使用方法：  <script src="js/babel.js"></script>  <script type="text/babel">  //此处的代码将由Babel编译器处理  </script> |

5.JSX中的{ }表达式

提示：在JSX中，还可以使用{ }进行数据绑定或者运算，可以使用在两种情形下：

① <p>{表达式}</p> —— React中的内容绑定

② <p title={表达式}></p> —— React中的属性绑定！注意与Angular中的不同

测试：JSX表达式中可以出现哪些语句：

1)算术运算：可以

2)比较运算：可以，但true/false都不显示

3)逻辑运算：可以，但true/false都不显示

4)三目运算：可以

5)调用函数：可以，对象成员函数或全局函数或匿名自调函数都可以

6)创建对象：只要是可以转换为字符串的对象就可以——Angular中完全禁止

7)调用全局对象方法(JSON.stringify())：可以——Angular中不可以

6.React中如何实现各种数据绑定

①内容绑定：<p>{内容}</p>

②属性绑定：<img src={imgUrl} title={myTitle} />

③指令绑定：React中没有指令的概念

④事件绑定：

⑤双向数据绑定：

注意：因为React中没有指令，所以类似于ngFor/ngIf的实现有些特别——很重要！具体见手册“核心概念>第7/8章”

|  |
| --- |
| **React中的条件渲染：**  function check(){  if() return JSX  else return JSX  }  let element = <div>{ check( ) }</div> |
| **React中的列表渲染：**  注意：{ 数组 } 绑定语句会把数组中的每个元素展开添加为当前元素的子节点  let list = [....]  let element = <ul>  {  list.map((e, i)=>{  return <li key={i}>{e}</li>  })  }  </ul> |

练习：创建一个变量表示用户当前的积分，若积分大于0，可以显示一个视频播放区(<video>)，否则显示一段警告消息(<p>积分不足无法观看</p>)

练习：创建一个数组，保存如下员工信息：

[

{eid: 101, ename: 'dingding', salary: 5000, sex: 1},

{eid: 102, ename: 'doudou, salary: 6000, sex: 1},

{eid: 103, ename: 'yaya', salary: 7000, sex: 0},

]

把上述数据渲染在一个TABLE中，每个员工信息对应一个TR，内部包含四个TD

7.React中的自定义组件

概念：组件是一段可以反复使用的页面片段

组件 = 模板 + 脚本(数据) + 样式

React提供了两种自定义组件的方法：

**①function式组件——一般只用于创建简单组件(只有模板)**

function MyC01( ){

return ( JSX )

}

<MyC01></MyC01>

**②class式组件——适合于创建复杂组件(模板+脚本数据)**

class MyC02 extends React.Component {

render( ){

return ( JSX )

}

}

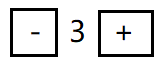
<MyC02></MyC02>

提示：React中要求组件名必须使用“大驼峰命名法”！

课后任务：

①阅读React官网手册“核心概念”第1~8章内容，学习事件绑定的方法和模型数据的声明方法——大约需要2小时

②创建自定义的class组件BuyCount，其中的模板内容如下：



要求实现点击减号按钮购买数量-1，点击加号按钮购买数量+1