第一天

1、let的新特性请一一列出

不存在变量提升

暂时性死区

不允许重复声明

块级作用域

2、const新特性--列出

不存在变量提升

暂时性死区

不允许重复声明

块级作用域

声明的变量不允许重新赋值

3、let和var的区别

4、对{ foo: "aaa", bar: "bbb" }进行解构赋值写出代码

Let {foo,bar} = {foo: "aaa", bar: "bbb" }

1. 请使用ES6 Class创建一个类，包含name和sex属性、fn()方法，在方法console.log()输出类中的name

Class Student {

Constructor(){

This.name = “jack”,

This.age = 20

}

Fn(){

Console.log(this.name)

}

}

Let s = new Student()

S.fn()

6、请使用ES6 Class创建两个类（父：parent，子：child），child继承parent，请写出代码

class parent {

constructor(num) {

this.money = num;

}

}

class child extends parent {

constructor (num,a){

super(num)

alert(a)

}

};

let c = new child(2000,100);

alert(c.money)

7、在第6题基础上完成子类重写父类上的属性、方法，写出代码

8、箭头函数和function区别或新增了哪些特性

9、set数据结构有哪些常用的属性和方法

1. map数据结构有哪些常用的属性和方法

11、数据类型都要哪些？ES6中新增的数据类型是什么并介绍

12、基本数据类型和引用数据类型的区别

13、数组有哪些扩展的常用方法

14、介绍一下Promise是什么？分别阐述Promise有什么特点、缺点？

15、Promise常用方法有哪些及作用是什么？

第二天

1. 请描述一下你对webpack的理解

Webpack 可以看作是一个模块的打包机，它做的事情就是分析你的项目结构，找到了模块以及其它的一些浏览器不能使用的拓展语言。并将其转换和打包为合适的格式提供浏览器使用

1. webpack的四个核心概念是什么

四个核心概念：入口(entry)、输出(output)、loader(预处理文件)、插件(plugins)

1. 为什么要用webpack

Webpack 可以将多种静态资源 js、css、less 转换成一个静态文件，

减少了页面的请求

1. 分别介绍什么是loader、plugin

loader是文件加载器，能够加载资源文件，并对这些文件进行一些处理，诸如编译、压缩等，最终一起打包到指定的文件中

在webpack运行的生命周期中会广播出许多事件，plugin可以监听这些事件，在合适的时机通过webpack提供的API改变输出结果

5、 loader 和 plugin 不同

区别：

loader 用于加载某些资源文件。

plugin 用于扩展webpack的功能

6、什么是webpack和grunt和gulp有什么不同

webpack 是前端的一个项目构建工具，它是基于 Node.js 开发出来的一个前端工具；

区别

Gulp/Grunt是一种能够优化前端的开发流程的工具 是基于 task 任务的

而WebPack是一种模块化的解决方案 是基于整个项目进行构建的

不过Webpack的优点使得Webpack在很多场景下可以替代Gulp/Grunt类的工具

1. 说一下webpack的构建流程

从启动webpack构建到输出结果经历了一系列过程，它们是：

1.解析webpack配置参数，合并从shell传入和webpack.config.js文件里配置的参数，生产最后的配置结果。

2.注册所有配置的插件，好让插件监听webpack构建生命周期的事件节点，以做出对应的反应。

3.从配置的entry入口文件开始解析文件构建AST语法树，找出每个文件所依赖的文件，递归下去。

4.在解析文件递归的过程中根据文件类型和loader配置找出合适的loader用来对文件进行转换。

5.递归完后得到每个文件的最终结果，根据entry配置生成代码块chunk。

6.输出所有chunk到文件系统。

8、create-react-app 如何使用

使用create-react-app主要用来创建项目

创建项目之前需要安装create-react-app

安装命令： npm install -g create-react-app

第三天

1. 请描述一下你对react的理解

react是一个用于构建用户界面的JAVACRIPT库，很多人认为react是MVC中的V（视图）

1. react的特点

（它使用虚拟DOM 而不是真正的DOM。 它可以进行服务器端渲染。 它遵循单向数据流或数据绑定

1. 列出React的一些主要优点

它提高了应用的性能 可以方便地在客户端和服务器端使用 由于 JSX，代码的可读性很好 React 很容易与 Meteor，Angular 等其他框架集成 使用React，编写UI测试用例变得非常容易

1. **什么是JSX**

JSX 是J avaScript XML 的简写

JSX是javascript高级语法糖, 最终执行时还是会被转成原生js, 通过babel等方式

1. 什么是组件

一个应用/版块/页面中用于实现某个局部的功能(包括html, js, css等)

把这些局部功能组装到一起，就形成组件

1. 在react中如何创建组件

react创建组件的方式：

React.createClass()、ES6 class 和无状态函数。

1. 区分Real DOM和Virtual DOM
2. 你了解 Virtual DOM 吗？解释一下它的工作原理

React将DOM抽象为虚拟DOM，虚拟DOM其实就是用一个对象来描述DOM，通过对比前后两个对象的差异，最终只把变化的部分重新渲染，提高渲染的效率

原理：1） 用 JS对象结构表示 DOM 树的结构，然后用这个树构建一个真正的 DOM 树，插到文档当中。

2）当状态变更的时候，重新构造一棵新的对象树。然后用新的树和旧的树进行比较，记录两棵树差异。

3） 把记录的差异应用到步骤1所构建的真正的DOM树上，视图就更新了。

1. 怎样解释 React 中 render() 的作用

每个React组件强制要求必须有一个 render()。它返回一个 React 元素，是原生 DOM 组件的表示。

1. 你怎样理解“在React中，一切都是组件”这句话

组件是 React 应用 UI 的构建块。这些组件将整个 UI 分成小的独立并可重用的部分。每个组件彼此独立，而不会影响 UI 的其余部分

1. **如何在React组件中绑定一个事件**

<div onClick={this.show}>Click Me!</div>

1. 你对 React 的 refs 有什么了解

Refs 是 React 提供给我们的安全访问 DOM 元素或者某个组件实例的句柄。我们可以为元素添加 ref 属性然后在回调函数中接受该元素在 DOM 树中的句柄，该值会作为回调函数的第一个参数返回。 用于对 render() 返回的特定元素或组件的引用。当需要进行 DOM 测量或向组件添加方法时，它们会派上用场

第四天

1. 在 React 中，何为 state

状态是 React 组件的核心，是数据的来源，必须尽可能简单。基本上状态是确定组件呈现和行为的对象。

1. 如何给state设置初始值

this.state = {}

1. 如何修改状态

可以用 this.setState()更新组件的状态。

1. 组件的状态(state)和属性(props)之间有何不同

props一般用于接收外部传入的数据; 而state则用于记录组件内部数据, 而且是需要经常改变的数据。

1. 类组件(Class component)和函数式组件(Functional component)之间有何不同

区别

1）构造函数创建的组件叫：无状态组件；

2）class关键字创建的组件叫：有状态组件；

3）有状态组件与无状态组件的本质区别在于是否有state（状态）属性。

1. 何为受控组件(controlled component)

受控组件：表单元素（input.textarea,selection）的值由React组件控制。

1. 在构造函数中调用 super(props) 的目的是什么

接收父组件传递的参数

1. 如何改变事件处理方法的this指向

箭头函数

通过bind绑定this的指向

1. React 中有三种构建组件的方式

React.createClass()、ES6 class 和无状态函数

1. props怎么设置初始值

getDefaultProps 设置props默认值

第五天

1. react组件的生命周期分为哪三个阶段

初始化阶段

更新阶段

卸载阶段

1. React组件生命周期的阶段是什么

初始化阶段

更新阶段

卸载阶段

错误处理阶段

3、详细解释 React 组件的生命周期方法

ComponentWillMount 编译前。渲染前调用

Render() 渲染时调用

componentDidMount 编译完成，渲染后调用

componentWillReceiveProps 组件接受props之前

shouldComponentUpdate 组件state被调用 必须返回一个布尔值，true false

componentWillUpdate 组件state调用后 将要更新时，但还没有render调用

componentDidUpdate 在组件完成更新后立即被调用

componentWillUnmount 在组件从DOM中移除的时候被调用

1. componentWillUpdate阶段什么时候使用

数据更新前使用

5、适合请求网络获取数据的生命周期是哪个阶段

componentDidMount（）{} 组件渲染完成时盗用

1. componentWillUnmount经常做什么操作

卸载阶段是调用，清除程序

第六天

1. react中父组件如何向子组件传值
2. 子组件如何向父组件传值
3. 兄弟组件如何传值
4. 子组件向父组件传值方法的this指向如何改变
5. 如何获取到表单输入的值
6. Axios请求数据的方法有哪些
7. Axios怎样配置默认属性
8. Axios拦截器的作用是什么

第七天

1. 什么是React-router？
2. 如何安装react路由，执行什么命令？
3. 嵌套路由怎么实现？
4. React-router 共几种模式？
5. 单页面应用的特点？
6. Redirect的使用方法？
7. Link和 NavlLink的区别？
8. BrowserRouter和HashRouter分别是什么？
9. 如何实现路由传值？
10. react路由js进行跳转？

第八天

1. Ant Design框架如何在项目中引用？
2. 如何在react项目中安装Ant Design？
3. 蚂蚁金服UI是一个什么样的框架？
4. 请写出你熟知的UI框架？
5. 如何在react中引入Ant Design样式库？
6. 如何实现按需引入Ant Design？
7. Ant Design如何使用Table组件？
8. Table组件实现分页，有什么属性？

第九天

1. Redux 安装？
2. Redux的设计思想？
3. Redux的核心概念？
4. Redux的优点？
5. Redux的工作流程?
6. Redux的基本概念？
7. redux 中间件redux-thunk的作用？
8. combinereducers的作用是什么？
9. Subscribe方法什么作用？
10. dispatch的作用是什么？

第十天

1. React-redux如何安装？
2. Provider组件的使用？
3. connect的作用是什么？
4. mapStateToProps方法的作用？
5. mapDispatchToProps方法的作用？
6. React-redux的基本概念？