1. 闭包，原型作用域链，继承，面向对象,

使用闭包主要是为了设计私有的方法和变量。闭包的优点是可以避免全局变量的污染，缺点是闭包会常驻内存，会增大内存使用量，使用不当很容易造成内存泄露。在js中，函数即闭包，只有函数才会产生作用域的概念。

闭包有三个特性：

1.函数嵌套函数

2.函数内部可以引用外部的参数和变量

3.参数和变量不会被垃圾回收机制回收

2.indow.onload与document.ready的区别？

1.执行时间

window.onload必须等到页面内包括图片的所有元素加载完毕后才能执行。

$(document).ready()是DOM结构绘制完毕后就执行，不必等到加载完毕。

2.编写个数不同

window.onload不能同时编写多个，如果有多个window.onload方法，只会执行一个

$(document).ready()可以同时编写多个，并且都可以得到执行

3.简化写法

window.onload没有简化写法

$(document).ready(function(){})可以简写成$(function(){});

3.this的理解，如何使用？

this代表函数运行时，自动生成的一个内部对象，只能在函数内部使用，随着函数使用场合的不同，this的值会发生变化。但有一个总的原则就是，谁调用它，它就指向谁。一般在这几种情况下会用到它，1.单纯的函数调用；2.作为对象方法的调用；3.作为构造函数调用；4.apply调用，apply（）是函数对象的一个方法，它的作用是改变函数的调用对象，它的第一个参数就表示改变后的调用这个函数的对象，当它的参数为空时，默认调用全局对象

4.react的优势以及特点

4.react 中如何获取一个真实的dom节点

使用ref属性获取Dom元素后，再使用原生javascript获取内容

5.react组件有哪些阶段：Mounted，Update,Unmount

6.react 可以在哪个生命周期中做性能优化

shouldComponentUpdate：做判断，判断当前state跟下一个state是否相同来决定是否渲染

7.redux 理解？

1. 8.props与state的区别？

==props：==一般用于父组件向子组件通信，在组件之间通信使用。 ==state：==一般用于组件内部的状态维护，更新组建内部的数据，状态，更新子组件的props等。

8.map与each方法的区别？

map()方法主要用来遍历操作数组和对象，each()主要用于遍历jquery对象。

each()返回的是原来的数组，并不会新创建一个数组。 map()方法会返回一个新的数组。如果在没有必要的情况下使用map，则有可能造成内存浪费。

9.你了解node吗？

npm koa框架

10.react组件之间如何传值？（直接关系组件，间接关系组件，无关系组件）

**一、【父组件】向【子组件】传值**

初步使用

这个是相当容易的，在使用 React 开发的过程中经常会使用到，主要是利用 props 来进行交流

**二、【子组件】向【父组件】传值**

接下来，我们介绍【子组件】控制自己的 state 然后告诉【父组件】的点击状态，然后在【父组件】中展示出来。因此，我们添加一个 change 事件来做交互。

**三、没有任何嵌套关系的组件之间传值(flux,reaux)**

11.你项目中是否用ES6的语法？如果是，请说出你经常用到的es6

12.css3的定位？让一个容器水平垂直居中？

* · [**方法一：*position*加*margin***](http://www.cnblogs.com/xianyulaodi/p/5863305.html%22%20%5Cl%20%22_label0%22%20%5Ct%20%22http://www.tuicool.com/articles/_blank)
* · [**方法二： *diaplay:table-cell***](http://www.cnblogs.com/xianyulaodi/p/5863305.html%22%20%5Cl%20%22_label1%22%20%5Ct%20%22http://www.tuicool.com/articles/_blank)
* · [**方法三：*position*加*transform***](http://www.cnblogs.com/xianyulaodi/p/5863305.html%22%20%5Cl%20%22_label2%22%20%5Ct%20%22http://www.tuicool.com/articles/_blank)
* · [**方法四：*flex;align-items: center;justify-content: center***](http://www.cnblogs.com/xianyulaodi/p/5863305.html%22%20%5Cl%20%22_label3%22%20%5Ct%20%22http://www.tuicool.com/articles/_blank)
* · [**方法五：*display:flex;margin:auto***](http://www.cnblogs.com/xianyulaodi/p/5863305.html%22%20%5Cl%20%22_label4%22%20%5Ct%20%22http://www.tuicool.com/articles/_blank)
* · [**方法六：纯*position***](http://www.cnblogs.com/xianyulaodi/p/5863305.html%22%20%5Cl%20%22_label5%22%20%5Ct%20%22http://www.tuicool.com/articles/_blank)
* · [**方法七：兼容低版本浏览器，不固定宽高**](http://www.cnblogs.com/xianyulaodi/p/5863305.html%22%20%5Cl%20%22_label6%22%20%5Ct%20%22http://www.tuicool.com/articles/_blank)

13.如何跨域？

**1、通过jsonp跨域**

**2、通过修改document.domain来跨子域**

**3、使用window.name来进行跨域**

**4、使用HTML5中新引进的window.postMessage方法来跨域传送数据（**但是缺点是IE6、IE7不支持**）**

14.对后台技术了解多少？

15.doctype这个标签是干嘛的？

16.讲一下你对盒模型的了解

1. 17.React项目遇到的问题

在react项目中使用Echarts的时候，在componentDidMount中执行绘制图表的函数echarts。Init（#id），提示获取不到宽高，因此，绘制出来的图标呈现出压扁的状态或者没有绘制

解决办法：给compoentDidMount中的drawCharts方法加上一个延时器

19.你有什么想问我的吗

20.版本怎么迭代

20.怎么缓存

21.position定位

22.网页优化

23. rem原理与em区别

24. 框架这一块问了一些很仔细的问题，还有es6那一块

25. 跨域机制，如何解决，为什么这样做，for of 的循环的原理，什么样的数据类型可以使用。generator的使用

26. 解构的原理，react和vue的钩子函数

27. 主要是angular的各模块原理

28. 以及js的原理

于明皓总结题库

1. react有哪几个生命周期？（8个）
2. 在哪个生命周期做优化？
3. react怎么从虚拟dom中拿出真实dom?(ref）
4. react的diff算法是怎么完成的？(基础原理）
5. React中的props和state的用法
6. Redux的流程 中间件（thunks saga logger devtools的理解，用法）
7. 比较redux和vuex的区别
8. react router3是否用过，router4是否用过，3到4有什么改变
9. react是什么层面上的框架，数据流是单向流还是双向绑定
10. webpack
11. render函数是发生在哪个生命周期
12. 正确答案应该是在willMount和didMount之间
13. 升云面试题总结
14. 数据为什么一定要在DidMount里获取更新
15. 总结一下不建议在constructor和componentWillMount里写的原因是会阻碍组件的实例化，阻碍组件的渲染，如果用setState，在componentWillMount里面触发setState不会重新渲染
16. 是请讲一下 后端给的数据是列表 前端实现的过程描述一下
17. 请讲一下 实际应用中 Set的使用场景
18. mobx的使用 （因为我用的mobx 也可能问redux的使用）
19. 路由都是怎么配置 怎么实现的
20. 如果要获取路由中的参数 你是怎么实现的
21. 路由如何实现异步加载
22. 1.react-router 如何获取路由参数；
23. 2.react生命周期，redux单向数据流；
24. 3.实际项目中遇到的棘手问题，如何解决；
25. 4.面向对象的理解；
26. 5.类组件和函数组件；
27. 6.ajax在哪个生命周期进行，为什么；
28. 8.然后主要想让你从实际项目出发 谈一下技术 或者 遇到的困难 以及如何解决的；
29. 1.箭头函数，普通函数（this）
30. 2.项目技术栈
31. 3.React按需加载
32. 4.React生命钩子
33. 5.Redux的理解及数据流
34. 6.项目组人员分布（前端、后端、测试）
35. 7.遇到的问题及解决方法（2个左右）
36. 8.网络安全及解决方法
37. 9.fetch和axios的区别 用fetch时有啥缺点
38. 10.vue和react的区别
39. 11.项目中遇到的难点怎么解决的
40. 12.es6用到过啥
41. 13.继承用过什么方法
42. 14.移动端布局
43. 15.vue的SPA开发性能优化

东淮朱君总结

react生命周期， 改变this的方法， redux流程， flex相关的，输入网址到呈现网页，发生的过程，http 状态码，

东淮罗童总结：笔试：第一个是flex布局的，用flex布局实现一个轮播图效果，只写css样式，

第二个是react父子组件传值的，比较简单

王雪总结：

1. 一个需求一个分支进行开发的时候，一个新的紧急需求加入进来，当前编辑的需求不想提交，如何进行下一个分支的编辑又不会造成污染？

2. React性能优化

3. 如何看待react？

4. Git命令

5. 如何做一个项目？

1,js常用语东西，promise,跨域，数组。react问的也不深

2,生命周期钩子和他们的作用

3,闭包，变量提升

4,es6新增

蚂蚁保险项目王超总结

蚂蚁前端问题

1.redux原理

2.thunk原理

3.provider原理

4.antd使用遇到什么问题

5.函数式组件和容器组件的使用

6.常用的生命周期钩子函数

1.怎么设置border设置四种颜色

Border-color: red green blue pink;

1. 浏览器的事件机制

事件捕获阶段---处于目标阶段 --- 事件的冒泡阶段---- 事件的默认阶段

1. redux的底层实现原理
2. 什么是flexx布局及一些属性
3. Target的原理
4. 事件流
5. React的生命周期
6. vue路由的原理
7. 盒模型和怪异盒模型
8. H5的新特性
9. Es6 的Promise方法原理
10. 怎么配置Webpack
11. Vuex怎么运行的
12. 项目中用到的Es6方法
13. Css3的动画
14. 对工程化的理解
15. 对node的理解
16. commenJs和jsx

伊鹏程总结题库

1. CORS 跨域
2. Css3 过度
3. Css3动画
4. 事件委托
5. On绑定时间和delegate绑定时间的区别
6. Axios请求数据的过程
7. Axios和ajax的区别
8. Git命令
9. Promise
10. React生命周期
11. 少量vue
12. 弹性盒
13. 笔试题:

1.假设一个对象数据结构为

1. ```
2. {
3. 'page5': {
4. a:1,
5. b:2 ...
6. },
7. 'page11': {
8. a:1,
9. b:2 ...
10. },
11. 'page2': {
12. a:1,
13. b:2 ...
14. },
15. 'page3': {
16. a:1,
17. b:2 ...
18. }
19. ...
20. }
21. ```
22. 请编写一个 js 函数，传入如上对象后将返回的数组中的对象 pagen 升序排序
24. 2. 用 reactjs 编写一个应用 <App /> 组件并挂载到 id 为 root 的 div 结点上，
25. <App /> 包含两个子组件 <Button /> （请一并实现）,
26. 点击哪个 <Button /> 时当前 <Button/> 的背景颜色变为红色，其他取消（不要直接操作dom)
27. 答：
28. 蚂蚁金服王英阁总结
29. 1.webpack配置项
30. 2.css3动画(实现进度条，背景图从白色变为蓝色)
31. 3.移动端优化(网络条件不同wifi 4g等)
32. 4.项目里遇到问题
33. 5.让一个div垂直居中
34. 6.创建一个对象方式（5种）
35. 7.项目里用px还是rem，px和rem对比，用rem比较爽的地方
36. 8.pc端和移动端需要注意的地方对比