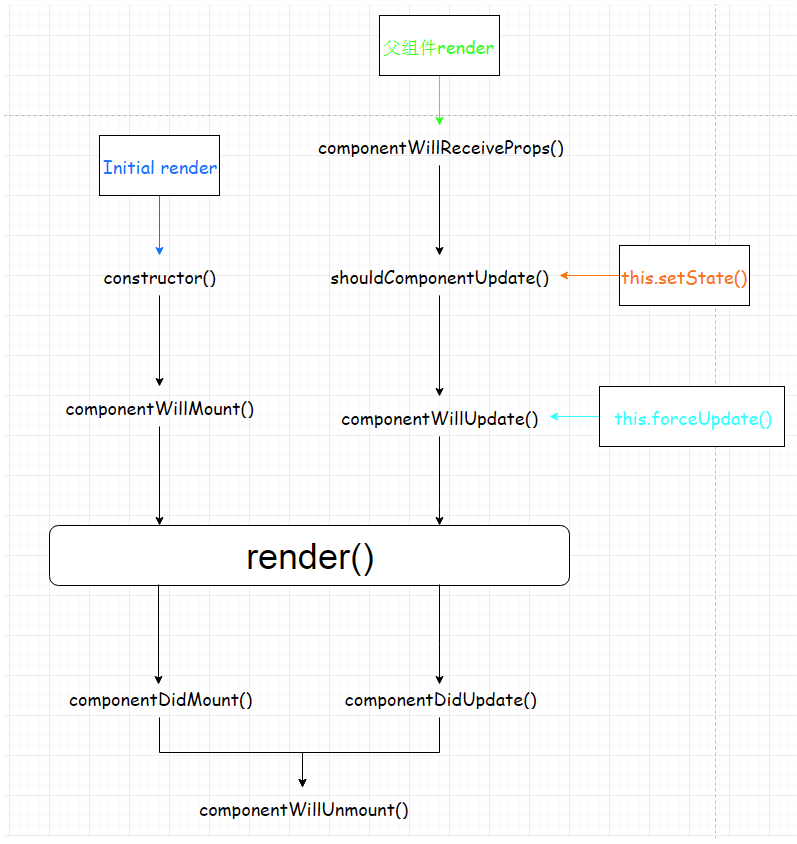
1. react有哪几个生命周期？（8个）
2. 

只执行一次： constructor、componentWillMount、componentDidMount

执行多次：render 、子组件的componentWillReceiveProps、componentWillUpdate、componentDidUpdate

有条件的执行：componentWillUnmount（页面离开，组件销毁时）

不执行的：根组件（ReactDOM.render在DOM上的组件）的componentWillReceiveProps（因为压根没有父组件给传递props）

1. 在哪个生命周期做优化？ shouldComponentUpdate
2. react怎么从虚拟dom中拿出真实dom?(ref）

这问题？

1. react的diff算法是怎么完成的？(基础原理）

dirty检查机制

1. React中的props和state的用法
2. Redux的流程 中间件（thunks saga logger dvtools的理解，用法）
3. 比较redux和vuex的区别

Redux：state reducer action

Vuex：state mutition aciton getter moudle

1. react router3是否用过，router4是否用过，3到4有什么改变
2. react是什么层面上的框架，数据流是单向流还是双向绑定

单向数据流

1. webpack

entry、output、plugin、loader

难易程度 等级（!,!!,!!!）

1. js的垃圾回收机制是什么原理（js）!!

标记清除法、引入计数

2.经常遇到的解决js兼容性问题列举至少三个（js）!

1. 不设置寛高的div，用至少两种方法，让它相对于页面水平垂直居中。（css）!

flex、position

**4.react-redux的实现原理（react框架）!!**

5.git经常用到的命令，至少五个，并且要说出每个命令是干什么的。（git add .）（git commit -m “”）（git push origin \*\*）（git clone）（git branch）（git checkout）!

6.react-router的实现原理 !

7.react的生命周期（必须知道并且知道每个的作用）!

8.数组的方法（至少说出6个）（js）!

9.闭包的原理（js）!

1. **伪数组转成真数组的方法（至少两种方法） （js）!!**

##### **常见的伪数组**

1.函数的内置对象arguments，它是所有实参组成的伪数组。  
2.DOM对象组成的伪数组，通过document.querySelectorAll等获得的dom对象列表。  
3.jQuery对象组成的伪数组，通过$('选择器')获取到的包含dom对象列表和jQuery方法的jQuery对象。

**1、Array.form**

1. rgba和opacity的区别（css）!

opacity属性的值，可以被其子元素继承，给父级div设置opacity属性，那么所有子元素都会继承这个属性，而RGBA设置的元素的后代不会继承该属性及属性值。

12.link和import的区别 （js）!

1. **哪些操作会造成内存泄露（至少三个）（js）!!!**

**1、闭包，**值得注意的是闭包本身不会造成内存泄漏，但闭包过多很容易导致内存泄漏。闭包会造成对象引用的生命周期脱离当前函数的上下文，如果闭包如果使用不当，可以导致环形引用（circular reference），类似于死锁，只能避免，无法发生之后解决，即使有垃圾回收也还是会内存泄露。

#### 2、意外的全局变量引起的内存泄露

Fun() {

a = 123 // 等价于 window.a = 3导致不能被回收

}

3.未清理的DOM元素引用

var a = document.getElementById('id');

document.body.removeChild(a);

不能回收，因为存在变量a对它的引用。虽然我们用removeChild移除了，但是还在对象里保存着#的引用，即DOM元素还在内存里面。

解决方法： a = null;

4、定时器setInterval或者setTimeout在不需要使用的时候，没有被clear，导致定时器的回调函数及其内部依赖的变量都不能被回收，这就会造成内存泄漏。

5、控制台日志记录对总体内存内置文件的影响，也是个重大的问题，同时也是容易被忽略的。记录错误的对象，可以将大量的数据保留在内存中。传递给console.log的对象是不能被垃圾回收，所以没有去掉console.log可能会存在内存泄漏~

14.react虚拟DOM实现原理 !!

**15.call，apply，bind的区别（js）!!!**

16.ES6有哪些新特性 !

17.H5新特性（canvas，video，webstorage，语义化标签，表单控件 等） !

1. HTTP请求返回状态码（至少说出常见的200,304,404）!

200，302，304，400，401，403，404， 502

19.关系型数据库和非关系型数据库的区别和好处 !!!

1. web网络请求的过程 !!!

web请求的处理流程如下：

1、客户发起请求到服务器网卡；  
2、服务器网卡接受到请求后转交给内核处理；  
3、内核根据请求对应的套接字，将请求交给工作在用户空间的Web服务器进程  
4、Web服务器进程根据用户请求，向内核进行系统调用，申请获取相应资源（如index.html）  
5、内核发现web服务器进程请求的是一个存放在硬盘上的资源，因此通过驱动程序连接磁盘  
6、内核调度磁盘，获取需要的资源  
7、内核将资源存放在自己的缓冲区中，并通知Web服务器进程  
8、Web服务器进程通过系统调用取得资源，并将其复制到自己进程的缓冲区中  
9、Web服务器进程形成响应，通过系统调用再次发给内核以响应用户请求  
10、内核将响应发送至网卡  
11、网卡发送响应给用户

通过这样的一个复杂过程，一次请求就完成了。  
简单来说就是：用户请求– 送达到用户空间– 系统调用– 内核空间– 内核到磁盘上读取网页资源- 返回到用户空间- 响应给用户。上述简单的说明了一下，客户端向Web服务请求过程，在这个过程中，有两个I/O过程，一个就是客户端请求的网络I/O，另一个就是Web服务器请求页面的磁盘I/O。