1. 后端与部署（P2）
2. 用过的koa2中间件、koa-body的原理、自己写过的中间件

koa-bodyparser怎么来解析request。koa原理，为什么要用koa

1. Pm2、master挂了pm2怎么处理. pm2怎么做进程管理，进程挂掉怎么处理
2. Node端的事件循环机制和浏览器端的区别，每个事件发生在何处
3. mongodb特点
4. node的事件方法及特性适合处理什么场景，
5. vue和react和函数式编程（P1）
6. 路由的动态加载模块，路由history。前端怎么控制管理路由

单页应用中游览器url跳转变化了，但没发请求是什么原理，怎么实现的（说了spa路由及原理）

1. 介绍纯函数
2. 介绍JSX，高阶组件，vdom原理，为什么性能好， diff算法。最后也要用dom的api，直接操作不效果更好为什么还要用v-dom。树的遍历，为什么复杂度是O(n)
3. Emit事件。getter和setter发生在哪个阶段
4. 渲染的时候key给什么值，可以使用index吗，用id好还是index好

vuex是做什么的？缺点？

1. spa应用
2. react单向数据流的理解

三、计网、web安全（P3）

2.常见http头、搜索请求中文如何请求、http2.0、通过什么做到并发请求、http1.1时如何复用tcp连接, HTTP状态码

3. cookie放哪里，cookie能做的事情和存在的价值

cookie和token都存放在header里面，为什么只劫持前者

cookie和session有哪些方面的区别\*\*

后端怎么存储session,讲一下登录验证全过程

1. tcp3次握手，tcp。http缓存机制。画出OSI七层，TCP/IP四层

TCP的可靠性如何实现；TCP重传机制怎么实现，TCP拥塞控制有哪些；能不能用udp实现tcp

1. 介绍xss，xsrf。点击劫持、canvas破解验证码，前端sql注入
2. 介绍HTTPS, HTTPS怎么建立安全通道
3. RESTful常用的Method。get，put区别，什么是幂等

12.301和302的区别

1. 如何避免301跳转https(在response中header)
2. TCP和UDP的区别，优缺点，适用场景。然后问TCP三次握手，四次挥手，怎样保证数据有序？怎样保证可靠性？HTTP协议有哪些方法
3. 基础问题（p1）

1.画三角形，css3对溢出的处理，css的选择器和优先级

2. promise、async有什么区别。Promise，异常捕获

Async里面有多个await请求，可以怎么优化（请求是否有依赖）

1. 介绍css3中position:sticky，position属性包括CSS3新增
2. JS实现倒计时

5. 介绍事件代理以及优缺点，事件模型，捕获与冒泡，哪些事件没有冒泡，没有冒泡的原因是什么

8.html语义化含义，优点

10. event对象封装，就是个观察者模式

11. Set和Map的区别，常用来做什么，底层实现

五、项目构建（p1）

1.用过哪些webpack的plugin和loader，怎么实现的

2.dev-server怎么跑起来的，抽取公共文件怎么配置

3. 使用import时，webpack对node\_modules里的依赖会做什么

4. webpack生命周期

5. 介绍单页面应用和多页面应用

6.webpack3升级到4为什么会提升速度？

7.webpack优化有哪些？

六、项目中问题(p2)

1. 文件上传如何做断点续传

2. 前端性能提升大致分几类

3. 安卓4.0到6.0过程中WebView对js兼容性的变化

WebView和原生是如何通信

1. base64为什么能提升性能，缺点，webp
2. 扫码登录业务设计。登录端除了websocket，还可以如何？轮询。
3. linux查看目录大小命令。linux的nginx配置。vi编辑器到末尾的命令
4. 单元测试

七、面向对象、设计模式、软件功能(p2)

1. 单例、工厂、观察者、中介者模式。观察者和订阅发布模式的区别(2)及其使用场景

2. 介绍AST

3.resuful的API设计

八、浏览器相关(p2)

1. 浏览器事件流向

3. 进程和线程的区别，线程的那些资源共享，那些资源不共享，进程间的通信

4. 重绘和回流和优化

九、新技术(p4)

1. 介绍service worker，PWA

十、数据结构和算法(p2)

1. 冒泡排序，选择排序，冒泡排序如何优化。快排

2. 介绍二叉搜索树的特点

4.栈和堆的区别，垃圾回收时栈和堆的区别具体怎么存储

5.二叉树层序遍历(面试官提醒)

6.JS的全排列(10分钟)