

面

面考察点

- 基面
 - 基技点
 - 和框架
- 面
 - 做什目，目中解决了什核心？
 - 目流程，前后端多个角色如何配合，gitflow等
 - 多人如何合作？
 - 的工作做的任何思考，目//自身 各方面都可以。
- 思路面
 - 框架 用的景
 - 某个技景的解决方案
- 面数
 - 第一 直接老大面，主要注合能力
 - 第二 技面，注算法思路和架思路
 - 第三 CTO面，和差不多，不会力
 - 第四 HR面，公司的好，了解的性格和做事方式

什是秀工程？

- 持成
 - 好奇心
 - 学能力
 - 分析解决的方式和能力
- 具体表
 - 利用技解决生活中遇到的，是只是上班写代糊口（好奇心）
 - 有自己的小作品（好奇心，学方法）
 - 有自己的或者博客，学或者目（做事方式）
 - 新技保持敏感，投入大量，注源社区（学能力，做事度）
 - 子主 5

各位技展方向（从0到1）

- 服务端
 - 看基法和api->写demo->引入学各和框架->写的->深入了解一些偏底的机制->写一个真正的上用->捕上性能上保障->工程化可持展多人->服化->服治理
- 前端
 - 看犀牛->看写->模仿一些特效 -> 学流行的框架->用框架小用或者一些流行的系的模仿 -> 真目，累 -> 深入了解各框架解决的核心 -> 解决多人，工程化 -> 深入js底，了解各框架的核心机制-> 架

我上次的

- 不会列所有目（太多了）
- 强特殊技能（管理，跨端，全）
- 近期做的重要目，核心解决了什
- 附加：github和网站，app等
- 面前准
 - 将最近自己做的几个比重要的事情梳理下，能清楚出整体架和思考改
 - 外将前端界常的一些一下，不必全面，但要有核心争力
 - 准一个重要：最后要面官的
- 面程
 - 冷静着，不要太多的猜面官的意，就事事
 - 掌握主，面官听，而不是一直提

面技巧

- 掌握主
- 梳理工作模式，自己的
- 很多并没有准答案，所以不要担心回答了
- 考官的考察点可能并不是本身
- 提前想好可能会的，个次的面都会有一些常

回答

- 目如何体？
 - 不是做了多少目（技用，做事方式）
- 面注意什
 - 价，加班
 - 工
 - 探是否急缺
 - 不要太于下
 - 准一个重要：最后要面官的
- 感自己能力不差，学能力强，不了面
 - 明面官怎不差？
 - 不要靠感
- 面候人不合，是否当面告知？
 - 一般比差的会，或者心情好的候会，但是会予指，不完全看面官格
- 自己以后有什打算的，如何回答
 - 工程成方向
 - 自我
- 生没有目
 - 打好基
 - 了解社区解决方案
 - 自我践

基面

- js
 - ES6的特性介，越多越好
 - 正
 - 包和作用域
 - 遍
- css
 - 盒模型
 - 画
 - 浮
 - flexbox
- 泛化
 - 列常的 http status code
 - 器起求之后生了什
 - 客端和内嵌之H5如何通信？
 - 跨域，有何解决之法？
 - mvvm具体是怎一个分？述 个及其角色

参考料

- <http://www.html-js.com/article/Large-search-front-team-column%20296>
- <https://zhuanlan.zhihu.com/p/21707924>
- 我的小密圈：http://t.cn/Ri2cBT0（全成日）