



一个菜鸡前端的大厂逆袭攻略(下)

五、面试篇

5.1 做好充足的知识储备

5.2 提前在 leetcode 上刷算法题

5.3 简历准备

5.4 保持耐心，及时反馈总结

5.5 使用一定的技巧进行问题陈述

5.6 展现自己的思考和研究能力

六、乘风破浪，道阻且长

## 一个菜鸡前端的大厂逆袭攻略(下)

### 五、面试篇

#### 5.1 做好充足的知识储备

有了前面学习和项目的基础，接下来就到了面试环节。在投简历之前，我反复梳理了前端的各项基础知识点，包括但不限于：

- 【CSS】BFC/IFC、居中原理、三列布局，左右定宽中间自适应、flex/grid
- 【JS】原型链与继承方式（如寄生组合继承）、new 操作符、执行上下文与闭包、深浅拷贝、内存管理、event loop、this 指向、es6 新特性（map、set、weakset、weakmap、proxy）、手写 promise.all、手写防抖节流
- 【TS】访问修饰符、interface 和 type、泛型及 pick/partial/record 等操作符
- 【node】多进程模式，libuv 原理，express 与 koa 等框架及中间件原理
- 【网络】DNS 解析流程、CDN 原理、TCP/UDP 协议、HTTP 各版本差异、HTTPS 原理、状态码、cookie/session
- 【浏览器】DOM tokenizer、CSS tokenizer/parser、layout/paint/composite 原理
- 【性能优化】首屏渲染时间如何获取，常见优化手段，performance API、SSR 原理
- 【缓存】强缓存和协商缓存，如何让缓存强制失效
- 【安全】XSS/CSRF/SQL 注入，原理及前端防御手段
- 【跨域】成因及常见解决方案
- 【react】virtual dom 结构及 diff 算法原理、fiber 原理、hooks 原理
- 【vue】双向绑定原理、模板编译过程、数组如何实现监听、computed/nextTick/keep-alive 原理
- 【工程化】webpack 原理，如何手写一个 plugin 和 loader，热模块更新原理、tree shaking、webpack5 新特性、打包策略及优化方案、rollup/vite/snowpack
- 【PWA】serviceworker 使用场景、原理，和 webworker 区别 上面这些是前端应该需要具备的通用能力，大家可以对照着参考一下。实际上我还针对我自身情况，额外准备了以下内容的，大家根据自己实际情况来就好：
- 【跨端开发】flutter、weex、rn、ionic、小程序、快应用、webview 和 JS bridge
- 【图形学】bezier 曲线/曲面、b 样条曲线/曲面、phong 光照模型、three.js 和 WebGL、css/canvas 矩阵变换
- 【AI】SVM（包括 linear SVM 和 kernel SVM）、特征提取（各个场景不一样，我主要研究了图像识别领域，所以特意关注了 HOG）、卷积神经网络、空间金字塔池化
- 【后端】Java、C++、php
- 【数据库】SQL、mongoDB、redis

不打无把握的仗，准备好了咱再上。

#### 5.2 提前在 leetcode 上刷算法题

现在无论大厂小厂的面试，基本都绕不开算法这一块（没错说的就是你，字节跳动）。所以在准备面试之前，我提前三个月开始，每天坚持去 leetcode 上刷算法题，并同时学习回顾数据结构的知识。

在准备的这段时间内，我刷完了 leetcode 前 200 题中所有的简单和中等题，甚至困难题也啃下了几个。事实证明虽然刚开始刷算法的时候很痛苦，但当我拥有了丰富的刷题经验以后，后面的做题效率越来越高。在面试过程中再遇到算法题，轻车熟路两下解决。

#### 5.3 简历准备

接下来的环节就是准备简历，这一环节的重要性不用多说，大家都懂。我在此过程中主要注意了以下几个点：

- 1) 履历真实，大厂都有背调的，想蒙混过关基本不可能
- 2) 突出技能和项目中亮点，非亮点的一笔带过，不要占据宝贵的篇幅
- 3) 所有写在简历上的知识点和项目经历，100%经得起考察
- 4) 展现出工作业绩，但不过分夸大，能数据说话就不用文字

写简历的过程其实也蛮痛苦的（就像我写这篇文章一样），需要不停修改甚至推倒重来。我在老班指导下前前后后也小改了半个月，最后才拿出一份还算满意的结果。

#### 5.4 保持耐心，及时反馈总结

当简历投递出去，正式进入面试阶段以后，要做的事情就是耐心了。

虽然期待着进入大厂，但在一开始没有着充足把握的前提下，面对招聘网站上各路大厂面试官的沟通，我一律选择了无视。毕竟如果没做好准备就投递简历去大厂，得到的结果大概率不会太好。

所以最开始大半个月的时间都在面试一些中小公司，一来是锻炼自己的口头表达总结能力，二来是通过面试审视自己知识体系上的缺失。这里很重要的点是，在面试结束以后，一定要立刻趁着自己记忆尚在，赶紧把问题记录下来，留待后续复习审视。

#### 5.5 使用一定的技巧进行问题陈述

针对一些开放性的问题，比如问到项目的时候，我会尽量在表述过程中使用一些小技巧，STAR 法则便是最常被用到的一条。

通过情境(situation)、任务(task)、行动(action)、结果(result)这样的描述流程，既准确的描述了经历，又展现了解决问题的能力。同时也会让面试官觉得我是一个很有条理的人，这在面试场合里非常加分。

#### 5.6 展现自己的思考和研究能力

任何面试官都想要招一个对技术有追求，善于思考和研究的人，面试过程中也需要尽量体现这一点。

换位思考一下，假设你是面试官，你希望听到对方说【我用过 XX 框架】，还是希望听到说【我在做 X X 项目的时候，预研过 A 和 B 框架，最终因为 XX 等原因，我选择了 A 框架。在此过程中，我遇到了问题 XX，为此去看了 A 框架的源码，了解到该框架是基于 XX 思想。后续在研究后尝试用 XX 方法对其中一个模块进行优化改写，以解决项目问题，完成后写了篇总结文章发布到 XX 网站。甚至尝试给原团队提了个 PR 以推进解决此问题。】没错，后一种回答方式就是我在面试中经常用到的。

### 六、乘风破浪，道阻且长

虽然成长的过程是痛苦的，但通过这两年的努力最终拿到了一个满意的结果，写下这篇文章也算是对过去的一个总结。未来路还长，进入大厂只是一个开始，今后除了技术还需要锻炼自己的综合能力，要面临的挑战和提升都会更大。也希望看到这里的小伙伴们能乘风破浪，实现自己的目标，诸君一起共勉！