

三年前端面试经验加感悟

19年年底的时候从工作了两年半的公司离职，用了半个月的时看面试题及刷算法，半个月后开始投递简历，分别投了阿里、百度、快手和字节跳动，除了阿里一面匆匆忙忙电话面试把我卡掉了之外，百度、快手是现场面，字节跳动视频面，三家offer全都拿到，最终选择了字节跳动

三家面试，都是三轮技术面加一轮hr面，实际上，无论是哪家的面试，问的东西都是很相似的，都可以在网上搜到答案，都一定存在于某篇面试题汇总文章内，因为技术就是那个技术，就算是问出花来，本质或原理还是一样的，想看详细的真实面试题，掘金、牛客网、知乎都有很多，随便找下都看不完

授人以鱼不如授人以渔，这篇文章还有其他我觉得更值得分享的东西，所以为了避免篇幅过长，我就不详细一一列举我在这些公司中都碰到过什么题目了，每家各提两句我印象比较深刻的

百度

百度是我被阿里卡了之后面试的第一家公司，年前北京还是可以现场面试的，所以我也是去百度现场面

一面问的基本上都是基本功了，值得一提的是，百度是三家中最——一个问我 CSS 问题的，大概就是面试官现场从他正在做的业务中拿了一个布局场景，让我写样式做好布局（不是圣杯布局或者双飞翼布局之类的），主要考察对 CSS 的灵活运用，需要稍微动点脑筋

二面也问了一点基础，主要还是考察了一些场景解决方案或设计方案，算法的话，就问了一个原理是二叉树后序遍历的题目，比较简单，没费什么事就做出来了

三面应该是经理面，就问了一个场景设计题，然后就开始谈人生理想，为什么离职啊，有什么规划啊，在之前团队做过什么有贡献的事情啊，碰到过什么难题啊最后又是怎么解决的啊，加入百度后你想做什么啊，等等，反正我感觉我回答这些问题比我回答技术题难多了，因为有些问题太抽象了，我都不知道想问什么，但又不好一直追问他到底想问什么，我只能根据我的理解来回答，一边回答一边还要察言观色，直到面试官感觉差不多了才能停，说得我口干舌燥，我宁愿多问我几个技术问题

总结一下，百度的这场面试，更倾向于考察实际做事能力，问的都是很实际的问题，如果恰好曾经在实际工作中碰到过这些问题，或者看过相关的文章，回答起来基本是问题不大的

百度的部门是百度APP，算是主航道吧，也挺想去的，但最后因为一些原因还是拒了

快手

一面同样还是考察基础，面试官的工作技术栈是 React，我的是 Vue，面试官也没有为难我，就从 Vue 问起，问了一些面试题常见的什么响应式什么 diff 算法什么观察者模式之类的，因为我之前系统性地看过 Vue 源码，所以回答起来也没什么难度，并且回答完了之后，顺便给他额外讲了下 Vue 的执行流程和相关源码原理之类的，说得比较多，面试官还是挺满意的

二面问了原型链等比较高级抽象的东西，然后问的其中一个算法是洗牌算法，我感觉问这个题目其实不太好，因为洗牌算法这个算法有点太独立了，不是什么回溯啊动态规划啊分治啊这些，是无法举一反三的一个题目，如果你恰好看过洗牌算法的实现，就那几行代码，你闭着眼睛都能写出来，如果你没看过，可能你想破脑袋都不知道该怎么写，当然，换个角度，运气也是实力的一部分

三面应该是团队的前端负责人吧，和百度三面问得差不多的问题，都是先来一道场景设计题考察一下技能运用能力，然后开始问规划之类的人生理想

总结一下，快手的这场面试考察得更加全面，同时和三面的前端负责人挺能聊得来，感觉挺合适的，而且快手也是我主动投递的最后一家公司，面完之后第二天我就回家过年了，就等着年后入职了，只不过还是没去成，因为我跳车字节跳动了

字节跳动

其实我没打算投字节跳动的，是过年在家玩耍的时候字节的 hr 把我从简历库捞出来给我打电话让我投一下，说可以年后视频面试，我想了下，投就投吧，反正也没事，万一快手那边出了什么岔子我还有后路可走，然后就约了视频面试，结果没成想就过了

三面面试官就是我现在 leader，聊完后觉得也很对眼，再加上我一年前和半年前其实投过字节跳动两次，第一次过了，但因为某些原因我没去，第二次没过，现在是第三次，过了，比较纠结，因为已经接了快手的电子 offer，反悔不太好，但又想了下，可能这就是缘分吧，我感觉就算我这次不进字节，将来早晚也要进，晚进不如早进，所以就只好对不住快手那边了

成功面试的几个经验

经过这些面试，加上我看过的一些面经，我稍微总结了一下大厂面试成功的几个经验

学历

这很好理解，无论你工作了几年，hr或面试官看简历第一眼想看的就是学历，这是一个很直观地给简历打分的点，好的学历不一定能让你通过面试，但却可以让你的简历通过初筛，很多人可能就因为学历这一关，导致连面试流程都进不去，可能空有一身本事却无法施展

但这也是可以理解的，大厂简历太多了，hr不可能给所有人面试机会的，否则面试官其他的啥也不干加上24小时连轴转也面试不过来啊，学历是一个很直观的筛简历的方式

工作经历

学历这种事情既然发生了就没法改变了，只能认了，但学历最大的作用其实还是让简历通过初筛，如果你的学历不足以让你的简历通过筛选，那么你还可以将希望放在一个好的工作经历上

你的学历不够让你立刻进大厂，但可以先进一些二三线厂或者小型的独角兽啊，那这可选择的范围就很多了，在这个层次的公司老老实实待个两年，期间别忘了修炼好基本功，别荒废了

两年后你的标签就是一个学历不是太好，但工作经历还不错，并且比较沉得住气的一个候选人，一般情况下，大厂给你个面试机会肯定还是没问题的，进入面试流程后，那就是你真正凭本事的时候了

这里有三点需要注意：

1. 作为跳板的公司不能太小了

最起码是二三线或者独角兽，公司太小了也不可能给你太好的项目经历，另外，面试官和hr大概率得听说过你这个公司

2. 不要频繁跳槽

如果你是从腾讯跳到阿里再跳到微软谷歌，哪怕你半年一跳都没有任何问题，但如果你是从一个不知名小厂跳到另外一个不知名小厂，这种一年一跳hr都嫌你跳得太频繁了，觉得你不稳定，半年一跳就没法看了，如果你在小厂，然后将来想去大厂，那么就要稳住，别把简历搞花了

3. 不要闷头搬砖

无论在哪工作，肯定都是以写业务代码居多，也就是搬砖了，但既然你打定了主意将来要去大厂，那么就绝对不能真的局限于搬砖，要把砖搬出花来

比如发现项目中有个结构不太合理，但凑合着用问题也不是太大，那么你就得想我不能凑合，我得把它整舒服了才行，然后说干就干直到真的把它搞定，比如你觉得工作流程中有些步骤是重复工作，那你可以考虑下是不是可以造个轮子解决这个问题

我经历的面试，很多情况下就是让你给出解决方案，考得就是技术的实际运用能力，几乎每一轮面试我都会碰到这种题目，这就要求你在平时工作中要有意识的积累，光搬砖你是说不出来完整的解决方案的，当然，一些常见的基础知识比如原型链、闭包、diff算法等也都是会问的，所以你不能因为平时搬砖很少用到这些就不管了，有些知识点你可能知道是怎么回事，但就是说不明白，那也是不行的，一定要真的弄明白了

总而言之，就是要有自己的思考，不能光顾着搬砖，因为搬砖谁都会，大家都一样，都是搬砖的，凭什么你能去大厂我不能去？所以你就得比他们强才行，没条件创造条件也要上

个人修炼

如果你因为实在太倒霉，去的公司接连倒闭或者领导们都太xx，你没办法不频繁跳槽，自然也没有可以拿出手的好项目，然后把简历搞花了，也不是就无法补救了的，外界条件靠不住，你还可以纯靠自己啊

比如写博客，记录自己平日的技术思考，倒腾出的解决方案，创建或者参与开源项目，把自己的博客和 Github 整得有模有样，各种分析、源码、解决方案、开源项目，面试官点进去看了后笑容满面，你还愁你没有面试机会吗？

个人修炼也有助于我上面提到过的面试官喜欢问的场景设计题

比如，现在需要你重构百度app首页的feed流，feed流的展示的每一条数据，它的功能、样式会根据这条数据不同归属有不同的效果，比如广告、百家号、小视频等，点击数据流会有一些操作，比如打开一个modal或者播放一个小视频或者跳转一个页面，那么请你说下当你接到这个需求时你想要怎么设计？

这是一种考查知识点很多，并且问的问题范围比较大的题目？不会涉及到具体的问题点，考察的就是候选人的知识广度和深度，平时的积累，考虑问题的维度等方面，上面那个题，是没有明确答案的

这个时候就是考验候选人个人修炼的时候了，有的人可能一脸懵逼随便说几句就不知道该继续说什么了，有的人则可以跟你说上几个小时还意犹未尽，如果我是候选人，我可能会从长列表的优化、跨域/跨页面通信、配置化（可维护性）、数据预加载、数据缓存、页面/组件无缝切换/转场动画、离线包/ssr等角度来回答这个问题，如果没有这些知识储备，是不可能想到这些的，场景设计题是一个可以让面试官全方位认知你，同时也是你自由发挥的一类题目，所以一般大厂面试都会问到这类题目

然后也有几点需要注意：

- 高质量博客

博客既然是反映你自己技术水平的一个地方，那么首先就要保证原创，确实是你的东西，如果全是搬运的 API 文档或者其他人的东西或者小白教学的无脑文，那么还不如没有，面试官看了后只会觉得你难道就这点水平？所以，博客的质量一定要保证

- 持之以恒

不能三天打鱼两天晒网，一定要坚持下去，形成一个连贯的体验，一是让人一看就觉得你这人确实是认真在做这件事情，二是只有这样你才能有足够多的积累，几十篇文章肯定比寥寥几篇文章更能让面试官全方面了解你

亮点

这其实是一个在很多时候可以一招制胜的关键

关于亮点这个东西，我在 [前面一篇文章的最后](#) 其实已经说过了，这里我就知道照抄过来了：

最近参加了几场面试，发现面试官们都很喜欢问你有哪些亮点，不管是业务层面还是技术层面，并会按照你给出的答案深入下去，看看你这个亮点到底有多亮

一个追问你亮点的面试官，其实是比较愿意给你机会的，技术的范围太广，可能他问的你恰好不熟悉，这是很常见的事情，比如你熟悉 vue，他团队内用的都是 React，他追着你问 React 可能很难问出结果来，另外一方面，你也无法保证在每场面试中都保持最佳状态，万一你跟面试官根本不在同一个频道上，你们之间相互听不懂对方在说什么，还怎么继续？所以把选择权交给你，给你机会让你自己选，那么这就引出另外一个问题，如果你真的没做过什么有亮点的事情怎么办？给你机会你都抓不住，这可怪不到别人了

所以，如果你有一个较高的追求，那么在平时的工作中，哪怕是天天写业务代码，你也要有自己的思考，这个组件可不可以换一种写法，那个需求是不是可以简化一下，项目里的webpack需不需要升级到最新版，这个问题可不可以造个轮子来一劳永逸地搞定它？

无关问题大小，都可以引发思考，实际上，一般情况下也不太可能有什么大问题等着你去解决，大部分情况下都是小问题，但问题再小，解决得多了那也是一种可观的积累，通过这种积累，在团队内部，你就有了可以拿出来说的输出贡献，离开了团队，你也可以以此抓住面试官给你的机会

有时候，这种亮点比你背面试题刷算法还好用，毕竟，面试题或者算法题会就是会，不会就是不会，但是亮点这种东西可没有标准答案，能说的可多了去了

需要注意的是，你的这个亮点一定要足够亮，不能就亮一秒就没了，容易闪着眼，必须要足够坚挺

也就是你必须要在你这个亮点上有足够的深入和见解，上下游和细枝末节都要搞清楚，足以应对面试官展开而来的提问，这样面试官才会觉得你确实是擅长这方面

软素质

- 简历用点心

简历是招聘流程开始的第一步，首先最起码找个好点的模板，网上很多

然后就是内容要好好写，切忌罗列毫无意义的技术点，什么jq/vue/vuex/react/redux/http/cdn/vim/git，不是不能写，而是你别占用太多篇幅煞有其事地介绍这些，一句话完事就行了，要突出重点

不要写**精通**什么什么东西，精通这个东西每个人理解得都不一样，你觉得精通就是能在任意场景下熟悉运用。面试官可能觉得你既然精通 **js**，怎么 **vue** 是 **Evan You** 写的而不是你呢？搞不好要为难你一把，所以尽量避免这些可能引起争论的东西

重点写你做过的项目，**项目用到的技术栈，大概功能是什么，你在其中扮演了什么角色**，写重点就行了，别写得跟文档一样又多又长，细枝末节的东西等到面试的时候会问你的，三年以内工作经验的简历最好不要超过两页

如果你博客或 github上有点东西，最好把链接带上，如果就是一个空壳子，没什么内容，就别放上去了

另外，个人建议，简历上最好放张照片，相当于给简历增添特定的主语，有明确的个人属性，跟长得好不好看关系不大，除非你实在是觉得自己长得有点对不住全国观众那就算了

- 一定要会表现自己

面试官的初衷肯定是想尽可能多地了解你，这样才能更好地评估你的能力，但问的问题可能就一两句话，那么作为候选人，你不能就因为面试官只问你 $1+1$ 等于几然后你回答个2就完事了，这种回答虽然合格，但不是面试官最想得到的答案

给出了标准答案后，你应该根据主动面试官问的问题继续发散开来，比如为什么等于2，有什么实际运用的场景，以往有没有等于3的情况？

具体点话，比如面试官给你出了个原型链的题目，那么你要做的就不仅仅是把这个道题做出来，除此之外，最好还要主动说一下原型链的相关原理，最好现场画个关系图出来，然后说下有哪些应用场景，比如原型链继承，如果你说了这个，那么其实还可以继续往继承上延伸（尽管面试官根本没问你这个），总之就是要把你的知识点全都体现出来，尽可能多地给面试官透露你的能力

这不是偷奸耍滑，这恰恰是面试官想要看到的，他不需要绞尽脑汁地刻意引导，你就自己主动坦白了，面试官肯定高兴啊，最起码说明你懂得确实多，因为知识点掌握的不咋滴的人，他也说不出这么多来

你通过在某个知识上深入或发散的表现，一定程度上，也可以触发**晕轮效应**，某一方面的突出会给面试官留下深刻的印象，就像是月亮的光晕，导致面试官下意识地就忽略了你其他方面的不足

很多的程序猿可能就是不善于言辞，但这又不是让你搞社交，技术方面的东西我觉得还是可以开口多说两句的

- 第一印象

面试有时候也要看眼缘，有的面试官，在你刚看到他的时候，你就知道这场面试肯定过不了，同样的，有的面试官，在刚看到你的时候，就决定不给你过了

所以第一印象很重要，程序猿不需要西装革履，但也别太懒散，特别是个人的精气神方面，决不能弱了，一定要有精神，不要因为担心面试过不了而一副诚惶诚恐的样子，你越这样反而越过不了

面试你的人，基本是就是你将来的同事、leader或者隔壁组的同事，换做是你，你肯定也不想招个看着就邋遢，性格太负面的人进来当同事，所以一定要自信，需要注意的是，自信和狂妄是两回事，自信和谦虚也并不冲突

小结

作为技术人，最重要的就是自己的技术能力，很多人在刚毕业的时候一腔热血，每天打鸡血一般通宵工作也不嫌累，但是后来被社会毒打的次数多了，很容易就疲倦了，可能有的人喊着喊着就放弃了，开始安于现状，这是很危险的事情，技术这条路不进则退，只有持续学习才能在技术这条路上越走越远

而另外一方面，客观的困难确实又是存在的，比如工作量太大没时间，前端技术发展太快学不完等，这就需要你找到一条可持续发展的道路，比如每周仔细研读一篇高质量技术文章，或者定期地进行总结，不要给自己太大的压力，但又必须要保持学习和进步

无论你选择的方法是什么，最终最重要的还是要落到实处，收藏了一大堆优秀文章的链接放在收藏夹吃灰是没用的