

1. 基础

这个就很简单了比如

最基本的数据结构、算法

以 Java 为例的一些基础知识：JVM、集合、框架

计算机操作系统、计算机网络

通用的一些中间件：netty、nginx、redis、MySQL 等

这些是基本上都要知道的，尤其是一些面试常问的必须要数量掌握，尽管有人认为这是背书，但是连背书都不背的说明态度有问题，肯定是不是被公司接受的。

当然以上说的太笼统了，具体的复习路线可以参考其他大佬的作品。

2. 能力

这方面主要是围绕实习、项目来展开的基本上=技术硬实力+能力软实力

比如：有实习的

在实习期间做了哪些事能够证明自己的能力

如何在开发过程中优雅地书写代码？（其实一些知识譬如设计模式大家都会背，但是能够真正将这个知识落到实地的又有几个呢？

项目开发过程中如何一步步提升自己的技术能力、业务水平？

如何快速的理解业务？适应环境？

对于实习 OR 项目的思考总结？

要明白一点：工作了的人很喜欢问一些 case，尤其是一些领导特别喜欢问，哪怕这个项目技术再牛，那么它是如何落地的呢？

他的场景是什么？为了解决什么问题？使用了什么方法 OR 工具？达到的效果如何？最终能满足预期吗？

比如：大家都会背一些 Spring 源码，设计模式，但是能自己将这二者结合起来吗？比如结合 Spring 源码+设计模式开发？

对面试官而言，他们已经听吐了这些背的东西，如果自己能讲的让面试官眼前一亮，那么即时面试问题回答的不太好 最起码也能证明自己 对于技术是有追求、有思考的，而不是一个背题机器

对于一个技术而言，它在这个公司存在的意义就是为了变业务为服务 技术服务于业务，用业务创造价值

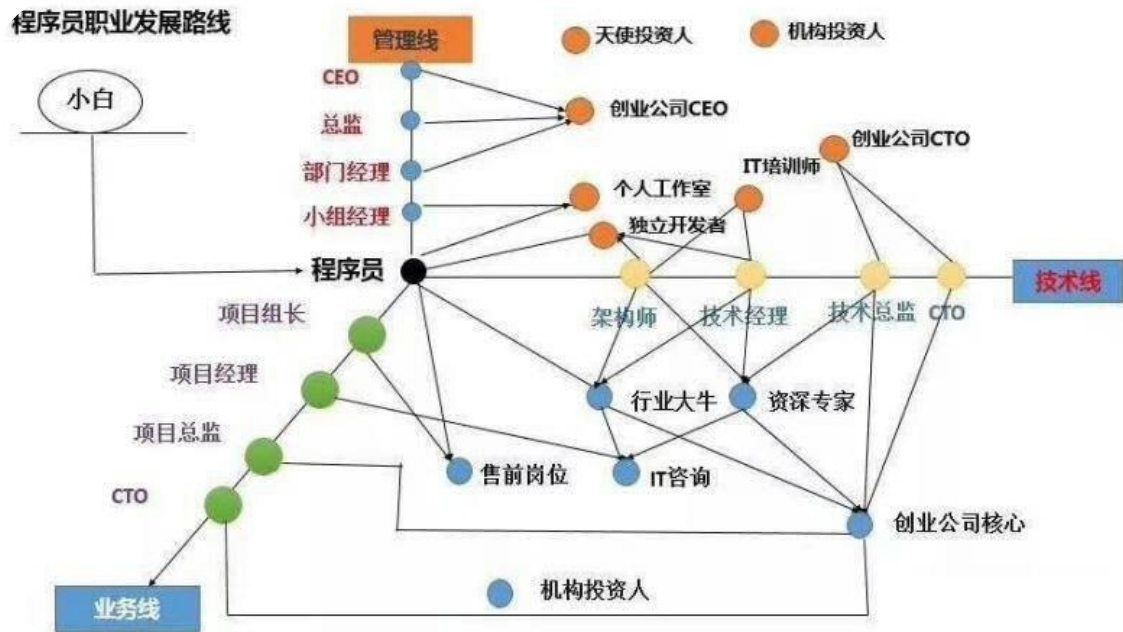
个人认为学生状态过渡到工作状态就是理论转化为实践动力的过程 有些人总觉得校招生身上有一股“书生气” 大概就是：理论的东西多而幼稚，有些不切实际的想法，能不能落地呢？

3. 规划

这点其实发现面试总结里面很少有人去谈，但是看到脉脉上，十个 HR，九个都会喜欢听到候选人这方面的一些思考、总结

先说路线规划

举个例子比如：会不会提前去规划自己的人生路线？



这是从网上找的一张图，可以看到有很多路线，究竟哪一条适合自己？可能很多人压根没思考过这个问题？另外自己适合哪一行呢？教育？广告？地图？还是纯技术？

隔行如隔山，因为我从去年五月份就来实习，也换过俩部门，所以对于这些事情比其他人要深刻一些

只有清楚知道自己喜欢的才有动力去做好他 不喜欢的只是为了生活被动的产出

不是所有人都适合做纯技术、也有的人压根就不适合搞业务，究竟那条大路通向自己心中的罗马，只有自己清楚

而这些事很多人都这么回复的：先有了工作再说

看过很多人 尤其是 90 后 工作半年跳槽、一年的也有 反正给我的感觉就是：不踏实 对于公司而言就是不忠实：谁能保证你从上一家公司跳又能干多长时间又跳走了呢？公司培养一个人的代价又有谁去承担呢？

在一个公司呆够超过五年 最起码可以说明 这个人在这个行业、公司 站住脚了

有的人入职以后才发现自己不太适合这个行业、这个方向，于是就跳槽，简历就花了。

简历一花，找工作就更不好找。别的公司不知道，jd 这边有的部门五二原则卡的很重

这还是次要，更重要的是自己能确保这个新的方向就自己适合或者喜欢吗？于是又开始跳、跳、跳

人生往往是一步错、步步错

别人已经在自己合适的方向上工作很久了 而你还在思考自己要做什么

男怕入错行，女怕嫁错郎，说的大概就是这个意思。

再说时间规划

大家都知道程序猿 35 是一道坎，到时候要么转管理，要么成为 CTO，自己对于自己的成长路线是一个什么规划呢？

管理路线：3 年能够处理日常开发当中的任何问题；五年能够在技术上达到自己一个比较理想的状态；七年能够成为小组 leader；十年能够成为部门 leader

技术路线：3 年能够在技术上能够有较大提升 五年成为架构师 七年成为资深架构 十年成为总监等

能够表达出这些，最起码能够说明自己 比较踏实，是一个有规划、有思想的人

面试经验

这块真的就只能实战找感觉了，每个人都有自己的一个表达方式，不过套路都是差不多：

1. 取长避短

像楼主本人学校不好但是一直在 jd 实习，那么自我介绍的时候可以说自己实习时间比较长等

学校比较好但是导师不放实习的可以着重说下自己的学校经历，paper、竞赛情况，都没有的可以说能够凸显自己能力的地方

2. 适当的往自己会的方向引导面试官

有的面试官会自己电脑前放一个题库，但是大部分不会，会根据简历上，自己脑海中搜索问题

比如问你 MySQL 调优，自己知道那些就说哪些，比如你知道索引这块哪些自己知道原理就说那些：（is null 判断可能会导致放弃索引 尽量避免使用判断等）再往下往往会问原理，因为你知道最左前缀原则、MySQL 优化器的索引代价分析、选择过程，你就可以轻松回答上来

但是你不太懂索引相关的原理就不要胡言乱语，瞎往自己不擅长的领域引导面试官，可以从设计规范方面谈起（使用 varchar 而不是 char 等等）因为面试官一般会问：为什么？自己在学习知识的过程中也要经常问自己一句：为什么 比如都知道函数表达式操作会导致索引失效，那么原理呢？

3. 面试充满了不确定性 你又不是 RMB 不会所有人都喜欢你

这个恐怕很多人是深有体会：我面试面的挺好的啊，问题都回答上来了，怎么还是挂了？放宽心态，该佛系的时候佛系一点啦~

面试充满了不确定性，能和面试官聊得来，即便问题回答的不怎么样，面试官也会放你一马；有的人跟面试官聊不来，甚至面试过程中发生了争执，那肯定就是不给过了呗

4. 面试是一个双向选择的过程

你被面试官面试的时候其实你也在考量这个部门、这个面试官技术水平、人品以及是否愿意引导新人等

一般面试自己的都会和自己入职后的工作关系是在一起的，一般是自己的同事、直属领导。

如果面试官为人和善、愿意引导你解答出问题来，那么入职之后你的成长速度也会更快的

如果面试官技术问题问的很模糊其辞，不够专业，那么面试多了你也会感受出来

你不是非我不要，我也不是非你们部门、公司不选，面试就是一个双向选择的过程

5. 烂大街的项目尽量不要写

6. 学历差的自信一点

大家的学历大多数都比我这个双非渣本菜鸟学历高得多，这也不是意味着学历差就一定不行，我这不是也上岸了百度、京东了么

衷心劝诫 22 届及以后毕业学历较差的盆友 学历差不代表一切

但正是因为学历的问题 就需要我们比别人付出更多 别人不会的可以通过学校、paper 补过来，但是我们一无所有只能冲

确实像 bat 这大厂对于学历的要求不是特别高

7. 回答 offer 情况

回答面试官

8. PS

都在问啥项目比较好，我从个人角度谈一下：有实习的话肯定是公司的项目比较好 无论是体量还是专业程度

没实习的话可以做个：仿制 dubbo、netty、tomcat、简单的 ioc 容器啦 或者一些实实在在能落地的项目，因为一个项目落没落地面试官一问就知道

前者会更好的走完一个底层的流程，从原理搞懂这些中间件 会比直接背书强
真正走完一个流程的项目远远优于那些网上的项目 没有实际的背景 那些开发中的情况面试官一问就破（以上为个人观点）

补充

1. 我觉得后端（尤其是 Java）竞争压力太大了，我想转前端可不可以呢？

我的建议是不要转

实际例子：我认识的一哥们后端转前端，最后去了 WXX 但是和我聊天的时候还是说 有点后悔

职业规划：别的不说，以 jd 为例，这边的高 T 基本都是后端程序猿， 前端、测试能到 T7 的寥寥无几，这是真实存在的现象

职业发展：个人感觉只有后端才能真正接触到核心的业务，能够接触到这些东西的，无论是业务发展、还是转管理，个人感觉都是很不错的

2. 我觉得现在竞争压力大，非科班的我要不要再坚持还是直接放弃互联网呢？

我的建议始终是：冲就完事了

非科班的确实要比科班的要学很多东西，他们能在课堂上、实验室里能学到的，自己要花很多时间去学习；

或许程序员就是这样 要有一股钻牛角尖的精神才能钻透那些枯燥无味的源码、底层原理，这个行业这么卷，找实习 OR 校招机会的期间养成的良好的学习习惯也会使你受用终身；