

如何成为一名程序员面霸

程序员面试，这对于不同程度的程序员或程序媛来讲都会有对应层级的问题，如何将自己想说的明白地表达出来，把自己已经很熟练的技术说得很理所当然，把自己会一点儿的技术如何理性客观地表达出来，这是一门技术。

程序员，大家似乎都公认的一个事实就是，程序员寡言少语，且有时候词不达意，因此造成了很多的大小悲剧。

面试找工作，这是一件特别神圣的事情，如何做好面试准备，如何合理地用程序员这个身份去应对面试官，如何站在心理角度去思考自己的表现以及面试官的反应，其实，很多时候，都是有套路的。

面试不是百分百中的，但我们若是做好准备，提高通过率确是百分百的事实。

所以我们做好面试准备是十分有必要的，面试准备主要有三点，心态、简历、技术知识。

心态，关键是要自信

心态决定一切，当然这并不是说，你心态好，你就一定能面试通过。我们去面试前，一定要做好相关准备，若是时间来不及，可以抓紧时间做一些面试题，因为只有硬件条件足够了，我们才有足够的底气。

当然，保持良好的精神风貌也是必须的，面试官若是看到一个萎靡不正的求职者，那我想，这场面试多半会黄了。

在面试之前也要做好一些调查工作，准备充足，在战场上才不会没有安全感。去面试

面试中有很多确定因素，但是也有很多不确定因素。对于确定因素，你会的，稳扎稳打地回答，不会的，也不要紧张，金无足赤，人无完人，保持平和的心态，搜索大脑里的知识，若觉得有探讨的必要，可以提出来，若无，则直接告知对方，若对方明确地表示他没有时间或者不愿意回答，你不能表现出一些怪异的脸色；当然，在面试中也会出现很多意外，面对意外，不要慌了神，理性地对待这个意外，控制自己害怕的心理，不断地告知自己，这种意外会有很多，要习惯，因为我们不能总是要求生活是一帆风顺的。

自信可以通过我们掌握扎实的基础知识来提高，也需要我们做好一些外在准备来排除影响我们自信的因素。时间安排好，例如，不要迟到，迟到了的话，不仅面试官会给你减分，而且自己心里也会因为迟到而害怕或紧张，从而影响自己的发挥。穿着得体，程序员的着装不需要华丽，当然肯定是不能邋遢的。合适的装扮不仅能让面试官觉察到你是一个朴实、踏实的人，也能提高自己自信心。穿着太暴露，爆人眼球，面试官肯定在第一眼就给你一个差评；穿着太邋遢，会让人觉得一个对自己生活都如此不伤心的人怎么会在工作中尽心尽力呢。程序员一般短袖或衬衫套外套，至于具体搭配，大家可以去网上咨询下。

自信不是自负，我们一定要对自己有一个清晰的认知，不要好高骛远、眼高手低，以为自己是211或985就了不起，毕竟人家招的是要会做实事的程序员，而不是只会说大话的人。知之为知之，不知为不知，是知也，千万不能不懂装懂，面试官都是工作很多年的，而且肯定也经历过很多求职者的面试，你的回答支支吾吾，或者你自以为是地乱说一通，面试官用“胡搅蛮缠”就给你定位了，程序员的标配是：逻辑清晰，基础扎实，经验丰富，而不是，所有杂乱的知识点混搭在一起。

自信也不是你高谈阔论薪资的资本，薪资是个敏感话题，却也是不可避免的，毕竟，我们最终的目的都是赚钱，但是若你在面试过程中处处都提薪资待遇，这会让面试官觉得你是一个无心工作，只关心工资的人，这样的人肯定在工作中也不会全力以赴的。

一个刚毕业的大学生，开口就要一万，项目经理问他一些问题，根本答不上来，最后项目经理说，我们这里还差一个前端的，他立马就把自己的前端简历拿出来，告诉对方说，其实我也在找前端的工作。

事先你理所当然地要价一万，你确定你做过自我评估了吗，觉得自己值得起这个价位吗，在经不住项目经理面试之后，又把自己的另一份简历拿来，给人的感觉是怕人家不

简历，关键是要简单明了

程序员不需要在简历上多花功夫，关键是简历的内容，毕竟大家公认的事实是程序员就是简单明了、朴实无华。

面试官一般会在求职者做自我介绍的时候大致浏览完简历上的内容，所以求职者在做自我介绍的时候，要语速适宜，心态平和，关注面试官大致把简历浏览到哪一块，然后自己做适当调整，切不可在做自我介绍的时候把很多不相关的一些经历说出来。

我们的简历，一般应有这样几个内容：

- 基本信息：我们是谁，联系方式是什么，专业，学历等。
- 技能验证：我们会什么，会搭建什么框架，会用什么技术。
- 工作经验：我们做什么，在哪些地方做过什么样的项目。
- 成果：项目最终达到了什么样的效果。

在简历上说自己会什么的时候，切忌说精通，你若是精通了，就不会是别人来面试你了。我想，没有多少人可以真正地拍着胸膛说，自己精通Java技术。

我们在写技能这一块儿的时候，多与即将要去面试的公司的要求搭勾，这样会提高通过率。

在简历上描述项目经验的时候，通常写上项目名称、项目的功能有哪些、项目所涉及到的框架或技术、你的职责、项目成果。

重点说一下你自己做过的自认为很不错的项目，你在项目中的职责，完成的主要模块，以及用到了那些亮点技术，效果怎么样，得到怎样的反响。

如果在项目中遇到了一些有价值的问题，也可以在描述该项目的时候告诉面试官自己是如何解决的。若是独立解决的，可以体现出自己是一个喜欢探究的程序员，有独立解决问题的能力，若是和团队成员一起解决的，也可以体现自己的团队合作意识，这些都是

除了这些内容，简历上还应有一些与工作相关的证书，例如英语四六级，软件设计师，信息系统项目管理师等。

一般成体系的公司的面试官都是很有礼貌的，很尊重人，而且也会综合考察求职者能力，所以，一般会在最后问一些看起来与项目技术无关的问题，例如，你还有问题问我吗，你平常干什么之类的。

对于这类问题，一定要在脑海里快速想一想，面试官可能会考察你什么能力，是团队合作，还是随机应变能力，还是看你是否主动积极。

技术知识，关键是要牢记

这是硬性条件了，不可避免的。

但是要在短时间内记住大学几年的知识，着实有点难，所以借着有效的方法才能事半功倍。

花些时间，把所有的知识搭成一根线，从逻辑上连起来，这样有条理，有逻辑因果关系，记忆起来也会方便些，而且在答题的时候也会由此及彼，不会陷入无话可说的尴尬地步。

Java中涉及到很多知识点，不能把它们放在每个都独立的小空间中去记忆，这样死记硬背是很痛苦的，而且刻在脑海中的时间也很短暂。

由一个知识点，引发一条线的知识点，记忆起来丝毫不费力，因为给人的感觉是顺其自然，因为这件事情发生，所以那件事情就应该是那样。

例如，面试官问ajax的工作原理，你可以从ajax的背景、诞生的意义、它的影响力、再到它的原理、它的用法、它的用处，最后来一句：所以它有简单、方便、减轻服务器的压力、局部刷新、提高客户体验之类的好处。

我们在回答面试官的问题的时候，切忌吐字不清晰，要不紧不慢、匀速地把自己的答案说出来。

- Java中安全机制是什么。
- 在Java中的运行时的四级安全性保障机制什么。
- Java中的异常机制框架图是怎样的。
- String类的修饰符是什么，为什么是它。
- 重写和重载的区别和意义。
- Final、Finally和Finalize的区别。
- Static代码块、普通代码块和构造代码块之间的调用顺序，以及一些常用场景。
- StringBuidler和StringBuffer的区别和联系。
- 文件的增删改查是否熟练。
- 绘制容器继承关系图。
- Collections和Collection的区别和联系。
- MySql数据库索引是怎么实现的。
- ArrayList和Vector的区别和联系。
- Set和Map有联系吗。
- HashMap和HashTable的区别和联系。
- 类型擦除是什么意思。
- HashSet和TreeSet的原理。
- 数组和字符串谁有Length方法，谁有Length属性。
- JDK1.5新增了哪些常用的概念或功能。
- 能比较清楚地简述各个集合类的特点及适用场合。
- IO操作（多媒体、对象、文件、文本）。
- 多线程和进程的概念。
- 线程的几个状态。
- 扩展知识：
 - POI、文档API、PDFAPI等。
 - TCP、UDP如何实现。
 - Java网络通信。
- Socket编程。
- Java加密和解密。

2. 中级知识

- 前后台数据通信方式。
- Ajax的原理。
- JQuery和操作原生DOM元素之间的关系。
- 浏览器对象框架图。
- 常用设计模式有哪些。
- 常用的组件开发，例如：文件上传、文档解析都有哪些技术。

3. 高级知识

- MVC思想是怎样的。
- Struts框架图能否描述清楚。
- Struts框架搭建是否熟练。
- OGNL与Struts的关系。
- Struts中的值栈是如何工作的。
- Struts.xml中的一般配置有哪些。
- Struts是如何实现线程安全的，能否真正意义上实现线程安全。
- 为什么Struts2逐渐没落。
- 如果让你对Struts2进行开源开发，让它重新走入市场，你会如何哪里入手。
- ThreadLocal在Struts中的关键是什么。
- Hibernate的二级缓存。
- Hibernate一般和那些插件缓存组合
- Hibernate的工作流程
- Hibernate的配置过程
- Hibernate与MyBatis的对比 Hibernate的特点
- 连接池
- 一对多和多对多关联关系
- Hibernate的优化机制
- Hibernate中的对象都有哪些状态，以及状态之间如何转
- Hibernate与Spring，Struts整合原理
- SSH究竟指的什么
- Hibernate中如何管理事务
- Hibernate的五个核心类是什么

- MyBatis中的缓存方式有哪些。
- MyBatis如何实现关联查询。
- 在MyBatis中一对一、一对多、多对多如何实现。
- 在MyBatis中是否存在延迟加载的概念，如何实现。
- MyBatis中的Sql语句与对象参数、对象返回值如何绑定。
- MyBatis如何实现事务管理。
- MyBatis如何实现session的管理。
- Spring的框架图。
- Spring的实现原理。
- Spring如何给bean赋值，有哪些方式。
- Spring如何管理事物。
- Spring注解存在的意义和原理。
- Spring的AOP的概念是什么，如何实现。
- 向上委托机制是什么原理。
- SpringMVC如何实现文件上传等。
- 过滤器、拦截器、监听器、连接池的原理分别是什么，如何实现。
- SSH流程。
- SSM整合原理。
- 爬虫：可以有Java爬虫，Python爬虫，自己可以简述爬虫的原理、背景和意义、爬虫的流程，做过的爬虫项目。
- POI、word的API、PDF的API，这些工具类的使用，可以很方面我们在项目中扩展功能，若能熟悉其核心API，也可以让面试官觉得我们是一个喜欢探索的人，阅读源码，喜欢探究。
- XML解析方法有哪些，DOM、SAX、JDOM、DOM4J几种方法的优点和缺点，以及适用的场景

若能清晰明了地讲清楚这几种方式中的核心方法，在面试过程中，流畅地回答可以提高我们的信心。

4. 高级研发

Log4j架构：Log4j将写日志功能抽象成7个核心类或者接口：Logger、LoggerRepository、

程序员的成长体系

人生有三个阶段。

1. 昨夜西风凋碧树，独上高楼望尽天涯路
2. 衣带渐宽终不悔，为伊消得人憔悴
3. 众里寻他千百度，蓦然回首，那人却在灯火阑珊处

在网上看到有人关于在阅读源码这一块对程序员的成长做了总结

1. 只关注项目本身，不懂就baidu一下
2. 除了做好项目，还会阅读和项目有关的技术书籍，看weikipedia
3. 除了阅读和项目相关的书外，还会阅读IT行业的书，比如学Java时，还会去了解函数语言，如LISP
4. 找一些开源项目看看，大量试用第三方框架，还会写写demo
5. 阅读基础框架、J2EE规范、Debug服务器内核。

在菜鸟阶段，有人总结到程序员成长历程的四个阶段

1. 不知道自己不知道 (Unconscious incompetence)。
表现：假自信、过度承诺、满口我能行没问题、看不起其他程序员...
2. 知道自己不知道 (Conscious incompetence)。
表现：有点心虚、虚心学习、偷偷努力、羡慕牛人...
3. 知道自己知道 (Conscious competence)。
表现：自信、比较客观、真正承诺、有点自我...
4. 不知道自己知道 (Unconscious competence)
表现：开放、主动、好学、谦虚...

当我们对自身职业的成长体系有了大致的了解，接着对号入座，看自己在哪一个阶段，如何能达到更高的阶段，当自己到了一个新的阶段，是否有新的体会。

拥抱源码，从源码中汲取思想，学会举一反三，我们刚开始学习JDBC的时候，自己会封装一个DBUtil的工具类，当自己封装好了，是否去Apache中寻找类似的工具进行对比，看权威是如何实现的，连接池底层究竟是如何实现的，缓存又怎样实现的，当我们站在三界之外的時候，会觉得一切都是那么地神秘，但是当我们走进来看到底层，不过感叹原来如此。

学会总结反思，孔子云，学而不思则罔，思而不学则殆，学无止境，新的东西之所以新不是因为它多么地高大上，关键在于，它利用平凡普通的基石能够实现非一般的功能。

写技术博客、维护个人网站系统、做技术demo分享，实现自我品牌效应。

GitChat