



TURING

智取 程序员面试

160 Questions and Answers for Success

【英】Edward Guiness 著 石宗尧 译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

智取程序员面试 / (英) 吉尼斯 (Guiness, E.) 著 ;
石宗尧译. — 北京 : 人民邮电出版社, 2015. 7
ISBN 978-7-115-39617-4

I. ①智… II. ①吉… ②石… III. ①程序设计—工
程技术人员—资格考试—自学参考资料 IV. ①TP311. 1

中国版本图书馆CIP数据核字(2015)第131405号

内 容 提 要

作者总结了自己几十年间作为应聘者和面试官的经验，分 12 章介绍了程序员顺利通过面试需要注意的方方面面，涉及电话面试、面试前的准备、面谈具体注意事项、待遇的沟通、具体技术问题等。本书用 160 个问题引发读者对编程的思考并且给出答案详细分析，是一本全面的程序员面试指导书。

本书适合所有程序员。

-
- ◆ 著 [英] Edward Guiness
 - 译 石宗尧
 - 责任编辑 朱 巍
 - 执行编辑 杨 琳
 - 责任印制 杨林杰
 - ◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号
 - 邮编 100164 电子邮件 315@ptpress.com.cn
 - 网址 <http://www.ptpress.com.cn>
 - 北京 印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 20
 - 字数: 473千字 2015年7月第1版
 - 印数: 1~4 000册 2015年7月北京第1次印刷
 - 著作权合同登记号 图字: 01-2013-8799号
-

定价: 59.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

目 录

第 1 章 招聘程序员：内幕	1
1.1 招聘的原因	2
1.1.1 公司计划扩张	2
1.1.2 特定的项目	4
1.1.3 有员工离职	5
1.2 同经理谈话	5
1.2.1 技术对话——不要有所保留	6
1.2.2 使用比喻	6
1.3 准备简历	6
1.3.1 囊括相关关键词，注意上下文	7
1.3.2 文笔要好	7
1.3.3 对工作经验作出解释	7
1.3.4 不要听信“简历不能超过两页”的建议	8
1.3.5 着重强调招聘广告中的技能	8
1.3.6 工作经历间不要留有情况不明的空白期	8
1.3.7 “读书、听音乐、看电影”	8
1.3.8 条理清晰	9
1.3.9 应届生简历	9
1.3.10 工作经验过多的简历	9
1.3.11 回归职场的简历	10
1.3.12 简历的常见错误	10
1.4 使用招聘网站	11
1.5 招聘中介	12
1.6 自己搜索职位	14
1.6.1 内向者的 关系网	14
1.6.2 寻找雇主	15
1.6.3 接近雇主	16
1.6.4 坚持不懈	17
1.6.5 把握时机	17
1.7 其他途径	17
1.7.1 Twitter	17
1.7.2 Facebook	18
1.7.3 LinkedIn	19
1.7.4 博客	19
1.7.5 Stack Overflow	20
1.7.6 Skills Matter 项目：“寻找你的师傅”	21
第 2 章 自信地应对电话面试	22
2.1 有的放矢	23
2.1.1 准备你的“小抄”	24
2.1.2 联系相关经历	25
2.1.3 处理难题	25
2.1.4 学会提问	26
2.1.5 电话面试准备清单	27
2.1.6 使用电话面试准备清单	27
第 3 章 现场面试	29
3.1 准备面试	29
3.1.1 了解充分	29
3.1.2 做足功课	30
3.1.3 着装得体	31
3.1.4 搞定不同类型问题	32

2 ► 目 录

3.2 最重要的事.....	34	5.1.1 十六进制转换为二进制.....	48
3.2.1 建立默契.....	35	5.1.2 Unicode.....	49
3.2.2 其他努力.....	35	5.2 数据结构	51
3.3 同样重要的事.....	36	5.2.1 数组	51
3.3.1 表达要清晰.....	36	5.2.2 散列表	51
3.3.2 掌控面试时间	36	5.2.3 队列和栈.....	52
3.3.3 用事实说话	37	5.2.4 树	52
3.4 有效交流	37	5.2.5 图	53
3.4.1 用热情战胜紧张	37	5.2.6 图的遍历.....	54
3.4.2 使用手势	37	5.3 排序	54
3.4.3 放慢语速	38	5.4 递归	56
3.4.4 开始和结尾要清晰.....	38	5.5 面向对象编程.....	57
3.4.5 重复主要观点	38	5.5.1 类和对象.....	57
3.4.6 熟能生巧	38	5.5.2 继承和复合.....	58
第4章 合同谈判.....	39	5.5.3 多态	59
4.1 了解招聘市场	39	5.5.4 用封装实现的数据隐藏.....	60
4.2 算算账	40	5.6 像函数式程序员一样思考	60
4.2.1 考虑整体待遇	40	5.7 SQL	61
4.2.2 必须有、应该有、最好有	41	5.7.1 什么是 ACID	61
4.3 招聘中介的作用	42	5.7.2 基于集合的思考方式	61
4.4 开个好头	42	5.8 全栈 Web 开发	61
4.4.1 避免过分让步	42	5.9 解密正则表达式	62
4.4.2 理想和现实	43	5.9.1 用锚定和单词边界来查询内容 ..	64
4.5 衡量合同条款	43	5.9.2 匹配字符集	65
4.5.1 知识产权	43	5.9.3 用限定符约束的匹配	67
4.5.2 不竞争条款	44	5.9.4 组和捕获	68
4.5.3 不招揽条款	44	5.9.5 不要想当然	69
4.6 如何应对不利状况	44	5.9.6 延伸阅读	70
4.6.1 “这是一份标准合同”	44	5.10 辨认难题	71
4.6.2 沉默回应	45	5.11 问题	71
4.6.3 谈判结果恶化	45	5.12 答案	73
4.7 谈判技巧总结	45	第6章 代码质量	85
第5章 编程基础.....	46	6.1 保持清晰	86
5.1 二进制、八进制、十六进制	47	6.2 富于表达能力	87
		6.3 效率和性能评估	87

6.3.1 大 O 表示法	88
6.3.2 性能评估	92
6.3.3 注意应用情境	92
6.3.4 目标明确	92
6.3.5 多次评估，取平均值	92
6.3.6 分治策略	93
6.3.7 先简后繁	93
6.3.8 使用分析器	93
6.4 理解“模块化”的含义	93
6.5 理解 SOLID 原则	94
6.5.1 单一职责原则	95
6.5.2 开放封闭原则	96
6.5.3 里氏替换原则	97
6.5.4 接口分离原则	97
6.5.5 依赖倒置原则	98
6.6 避免代码重复	99
6.7 问题	101
6.8 答案	106
第 7 章 常见问题	123
7.1 并发编程	124
7.1.1 竞态条件	125
7.1.2 锁	126
7.1.3 死锁	130
7.1.4 活锁	131
7.2 关系数据库	131
7.2.1 数据库设计	132
7.2.2 规范化	132
7.2.3 反规范化	133
7.2.4 填充规范化数据库	134
7.3 指针	134
7.3.1 接收值类型参数的函数	136
7.3.2 数组的处理	136
7.3.3 值传递和引用传递	137
7.4 设计问题	139
7.4.1 YAGNI 不是走捷径的借口	140
7.4.2 设计要考虑性能	140
7.4.3 不要只关注理论	140
7.5 不良编码习惯	141
7.5.1 错误的异常处理	141
7.5.2 不够谨慎	142
7.5.3 过于迷信	143
7.5.4 和团队对着干	143
7.5.5 太多的复制粘贴	144
7.5.6 预加载	144
7.6 问题	145
7.7 答案	146
第 8 章 编程语言的特性	151
8.1 二进制小数和浮点数	151
8.2 JavaScript	152
8.3 C#	152
8.4 Java	153
8.5 Perl	153
8.6 Ruby	154
8.7 Transact-SQL	154
8.8 问题	154
8.8.1 二进制小数和浮点数	154
8.8.2 JavaScript	155
8.8.3 C#	157
8.8.4 Java	158
8.8.5 Perl	160
8.8.6 Ruby	162
8.8.7 Transact-SQL	163
8.9 答案	164
第 9 章 软件测试：不只是测试人员的工作	194
9.1 单元测试	194
9.2 测试驱动开发	195
9.2.1 行为驱动开发	195
9.2.2 红、绿、重构	195
9.3 写出优秀的单元测试	196

9.3.1 运行速度快	196	10.7 答案	218
9.3.2 尽量简单	196		
9.3.3 目的明确	196		
9.3.4 具有指导性	196		
9.3.5 具有独立性	196		
9.4 测试运行缓慢的模块	197		
9.5 单元测试框架	197		
9.6 模拟对象	199		
9.7 问题	201		
9.8 答案	203		
第 10 章 选择合适工具	210		
10.1 Visual Studio	210	11.1 快速估算	240
10.2 命令行工具	210	11.2 脑筋急转弯	241
10.3 PowerShell	211	11.3 概率问题	241
10.4 Sysinternals 工具	212	11.4 并发处理	241
10.5 管理源代码	212	11.5 位操作技巧	241
10.5.1 Team Foundation Server	212	11.6 使用递归算法	242
10.5.2 Subversion	212	11.7 理解逻辑门	242
10.5.3 Git	212	11.8 编写代码	243
10.6 问题	213	11.9 问题	244
10.6.1 Visual Studio	213	11.9.1 快速估算	244
10.6.2 命令行工具	215	11.9.2 脑筋急转弯	244
10.6.3 PowerShell	216	11.9.3 概率问题	245
10.6.4 Sysinternals 工具	217	11.9.4 并发处理	246
10.6.5 管理源代码	217	11.9.5 位操作技巧	247
		11.9.6 使用递归算法	247
		11.9.7 理解逻辑门	249
		11.9.8 编写代码	250
		11.10 答案	251
第 12 章 编程智慧	278		
12.1 问题	278		
12.2 答案	282		
附录 准备小抄	309		

第 1 章

招聘程序员：内幕

小时候交朋友似乎很简单。由于我是看“巨蟒剧团”^①的喜剧长大的，因此，我交朋友的开场白往往是说一句：“我是一名骑士，哇呀呀呀，嘿！”我觉得这让人忍俊不禁。用这种方式，我结交到了几位密友（有一个一直交往了30年），但也失败过很多次。实际上，大部分情况下都失败了，极少能成功。有时，我的开场白会让对方不快。我对这点甚是不解。

那时的我没有意识到：并非每个人都喜欢这种滑稽的表达方式。不是所有小孩都看过“巨蟒剧团”的喜剧；即使看过，也不是每个人都像我一样喜欢。我用我自己的“巨蟒”视角去看整个世界，并没有意识到每个小孩的童年都是不同的，不是所有人都和自己志趣相投。

和我天真的行为一样，很多人事经理在招聘的时候也会犯相同性质的错误。由于他们在某些领域有一些宝贵的经验，所以会理所应当地认为，在相同领域工作过的其他人看待问题的方式会和他们一样，甚至可能会主观地认为彼此的思考过程都是类似的。在面试中，人事经理可能会突然说一些奇怪的话，谈话方式和我的“巨蟒”式开场白如出一辙：

“很高兴认识你！现在请你描述一个情境，在这个情境下，把数据库的表归一化为BC（Boyce-Codd）范式关系模式并不合理。”

如果你喜欢“巨蟒剧团”，而面试官刚好引用了他们剧中的台词，简直就是美梦成真了。如果真的是这样，那就祝你一切顺利，只需注意一些细节即可。比如，当面试官对你说“请这边（这么）走”^②的时候，不要理解岔了。

不过对于其他人而言，为了与面试官进行轻松随意的交流，是需要花些时间和精力来作准备的。

你可能会觉得我在说需要默契。默契是个绝佳的词儿，它表示一种和谐、融洽、一拍即合的感觉。我在用这个词的时候非常犹豫，因为最近它总是被误读，被认为有伪善的一面，就好像挂

^① 英文名为Monty Python，是英国的超现实幽默表演团体。下文楷体的两句话均为其剧中台词。——译者注

^② 原文“Please, walk this way”是一句双关，既可以表示“请走这条路”，也可以表示“请用这种方式走”。——译者注

着假笑的菜鸟推销员一样。

然而，如果想展示良好的交流能力，和面试官处在同一立场，并且高效通过面试，那么你确实需要和面试官建立默契。

从面试官的角度思考问题，是建立默契最简单有效的方法之一。如果你知道面试官的动机，就可以建立共通点，很容易就能恰当地回应。不论是从积极还是消极方面来解读这一点，你都可以快速洞察到他所期望得到的答复。

本章会大体介绍寻找编程相关工作的流程，包括以下内容：

- 什么能激起人事经理的兴趣，以及如何恰当地调整你的表达方式；
- 怎样的简历可以帮你得到一个面试机会；
- 如何使用招聘网站；
- 了解招聘中介的职能、运作方式，以及如何同他们打交道；
- 在不利用招聘中介的情况下，如何找到工作（这是完全可以做到的）。

让我们先来看看，公司雇用程序员的常见原因是什么。

1.1 招聘的原因

毫无疑问，想要雇用程序员的公司都有各自的理由。如果你知道其原因和动机，就可以相应地调整自己的策略。

1.1.1 公司计划扩张

为了跟上业务成长的步伐，很多公司都会制定中长期发展计划，从而需要招聘更多的程序员。随着经济复苏，这类现象会越来越普遍。

1. 面试官的动机和策略

因为招聘的职位是公司长期计划的一部分，所以面试官不会急于完成招聘。准备充分的面试官会有一份职位描述，可能还会有一份适岗者描述，用来与求职者的信息作比较。由于时间很充裕，他们不大可能降低事先定好的招聘标准。虽然他们在现实中可能并不期望面试者能完美符合所有要求，但也不会考虑那些主要条件完全不符的求职者。

2. 你的对策

你应当着重强调自己与职位描述相符的技能和经验，这点很容易做到。

如果技能和职位描述不太符合怎么办？比如，你想要应聘一个.NET程序员职位。很明显，在面试的过程中，面试官会期望求职者具备使用.NET组件库的相关经验，而你恰恰缺乏这方面的经验。当你的职业背景和职位需求不完全一致的时候，你有如下三个选择。

● 弱化差距

第一个选择是弱化技能上的认知差距，包括用其他经验来替代。如果你决定用这个方法，可以这么说：

“根据我之前的经验，学习一个新组件库的基本原理并不会花费太多时间。作为程序员，我们每天都会面对无数来自开源社区和软件供应商的新组件和新框架。”

你也可以从“学习是工作的一部分”这方面来解释：

“我热爱编程的原因之一就是有机会学习新的技术和平台，这正是这份工作的持续魅力所在。”

用这种方式谈话的最大风险是，你可能会给人含糊其辞的感觉，因此不要过头了。毕竟，招聘条件并不是面试官随便选择的。如果你太过轻视职位需求，可能会无意间陷入和面试官的争论之中，请务必避免这种情况。

● 正视差距

第二种选择是正视差距，如实回答。以“发展机遇”作为阐述重点，简要说明你是如何获得其他类似技能或经验的。

如果你决定正视差距，就需要同意面试官的看法，同时展示学习新事物的热情：

“我并没有那种技术的相关经验，但我真的很乐于学习它。”

如果面试官固执己见，你可以转移话题，问问团队内的其他开发者是如何学习新技能的：

“可以大体谈下目前我们团队里的开发者是如何学习新技能的吗？”

面试官举的每个例子对你来说都是机会，你可以借此说明，如果自己加入团队，当然也能采用相同的方法来掌握必备技能。

● 理解职位需求

第三种选择是挖掘面试官的动机，逐步探究职位需求之下的根本目的。

弄清楚职位需求的深层动机之后，你就获得了一个展示自己强项的绝佳机会。不过，使用这个方法的前提是，你必须已经和面试官建立了一定程度的默契，否则你看起来就像是在强词夺理，这是很危险的。这个方法的基本思路是让面试官觉得，尽管你缺乏特定的技能或经验，但是仍能理解并且满足职位的潜在需求。

例如，对于特定的IoC容器（比如微软的Unity），你可能缺乏使用经验。这时候可以问问选择Unity作为IoC容器实现的原因是什么：

“我知道使用控制反转模式的一些优点。能不能告诉我在你的工作中，微软Unity框架是如何有利于开发的？”

如果回答是促进组件间的松耦合，这就给了你一个机会去谈论自己对依赖注入原则的理解。如果回答是促进更有利于支持单元测试的代码组织方式，你就可以谈谈自己对于重构旧代码来添加单元测试的经验，或者谈谈你对于非IoC容器实现的依赖注入是如何理解的。

如果面试官问了一个很尖锐的问题，或者直接对你的相关工作经验提出质疑，你的回答应该有礼貌并且直率。在很多情况下，当面试官的表达方式很直接时，他希望求职者用同样的方式回应。

挑战：“我在你的简历中没看到任何有关微软Unity的经验。我们要使用这项技术，所以你不符合我们的工作要求。”

回应：“这是非有不可的吗？”

在这种情况下，回应的原则是把面试官的行为反射回去（参考第3章有关建立默契的内容，其中讨论了这种强大的技巧）。

不论你用哪种方式，请记住，不要总是想着自己不符合某些条件。尤其注意你的回答，尽量保持简明扼要。利用这种表达方式，你谈论得越多，面试官在之后的考量中就越对你有好感。如果面试官一直在强调你不符合条件的地方，而你基本上无能为力，就只能让你的回应保持简洁，并且尽量避免继续在这个话题上纠缠。

1.1.2 特定的项目

公司一旦找到新的市场机遇，可能会迅速组建一个开发团队，推出产品来填补市场空白。争取第一个进入市场，是公司迫在眉睫的任务，商业机会可能稍纵即逝。

1. 面试官的动机和策略

面试官期望应聘者在工作中的抗压能力强，能善始善终地完成任务。有时，招聘很急迫，面试官会放宽招聘标准，并不会要求你的工作经验和职位描述完全相同。尽管如此，你还是不要抱有这种侥幸心理。

2. 你的对策

请注意，虽然你需要尽力展示技能和经验，以便让面试官相信你十分符合公司的职位要求，但面试官也会考虑特定项目的需求。公司很可能会根据项目来调整职位需求，但面试官往往会使⽤同一份标准的职位描述，然后在面试中再去寻找“亮点”。

你要展示出比平时更大的适应能力。比起那些死抠职位描述的应聘者，你若能迅速了解应聘项目的特点，无疑可以给你带来优势。

很多软件咨询和服务公司常把若干新开发团队整合到一起来响应客户需求。在这种情况下，招聘公司会从特定项目的角度来进行招聘。如果你能展示相关的能力，被录用的机会就会高很多。如果你不确定公司招聘的动机，不妨直接问一下：

“请问，贵公司为什么要招聘新员工？是因为某个特定的项目吗？”

1.1.3 有员工离职

招聘程序员的第三个常见原因，是有员工要离职，需要补充人力。

1. 面试官的动机和策略

面试官的动机和公司计划扩张情境下的动机类似。他们可能有一定的时限，但又不如为特定项目招聘那样紧急。

这种情形要注意一点：该职位以前曾经有个员工，因此面试官脑中可能会有一份清单（也有可能是书面清单），列出他的优缺点，来确认接替者是否有类似情况。比如，那位员工可能是个细心缜密的人，那么面试官也想知道你是不是也比较细心。

2. 你的对策

很明显，你不太可能事先知道面试官对前任职员的看法，以及他喜欢哪些方面，不喜欢哪些方面。你在面试中能做的，是问问这个职位有什么特殊的挑战，需要具备哪些条件才能胜任，等等。

“请问，相对于其他程序开发工作，这个职位会面临哪些不一样的挑战？”

有经验的面试官不会主动提供前任职员的信息，但他们会让你一些暗示，来告诉你哪些品质比较重要，比如“团队精神”。如果你问得恰到好处，就会从面试官的回答中得到一些重要线索，继而懂得接下来该如何强调自己的经验和能力：

“冒昧问一下，您这样强调团队精神，是否因为团队以前的合作出现过问题？”

1.2 同经理谈话

业内流传这样的故事：一个聪明能干的程序员，给所有人都留下很好的印象，结果被提拔成团队领导，当了经理；新的经理再用他出色的交际能力招聘来了更多优秀的程序员。

然而事实往往并非如此。很多在其他方面都很优秀的程序员单单缺乏出色的交际能力，而这些人往往会直接或间接地面试你。

这到底是好还是坏？要看具体情况。有可能意味着你会有一次可怕的面试经历——如果真的是这样，你应该庆幸不必为一个笨嘴拙舌的经理工作。不过，你也有可能从中收获颇丰。不妨这样想：人际交往中，需要找到彼此之间的共同点；共同语言越多，在交流中就越能游刃有余，享受同他人的对话。

那么，程序员（即使经理现在已经不是程序员了）在对话时，都有哪些共同语言？简直太多了。是否曾经耗费几小时甚至几天的时间来追踪一个顽固的bug？你一定有这种经历，几乎可以肯定，对方也有类似经历。你喜欢某种编程语言的哪一点？经理有可能给出相同的答案。你经常访问程序员的网站吗？经理有可能也一样。你有最喜欢的XKCD漫画吗？你访问技术博客

<http://thedailywtf.com>吗？你最喜欢的编程类书籍是什么？你在闲聊中也顺口说出过技术术语 nullable 之类的吧？关于IDE，什么最让你抓狂？看看，你们有许多话题可以讨论。

1.2.1 技术对话——不要有所保留

在面试过程中，你可能会发现面试官不如你想象的那么擅长技术。比如，他可能是做项目管理或者产品销售出身。那么你应该如何应对？不要错误地认为你需要降低对话的专业度，否则面试官将你同其他面试者进行比较的时候，这会置你于不利之地。他可能并不理解你说的每句话，但他对你的印象会被你的用语所影响。举个例子，如果你始终含糊其辞，面试官就很可能会认为你不如其他面试者懂这门技术，因为其他面试者在谈话中使用了简练的技术语言和专业词汇。

如果非技术背景的面试官问了你一个技术问题，那么他期望的回答就是技术性的。面试官的手中可能有一张“小抄”，列出了所有问题和对应的答案。

如果同技术面试官打交道，更不要有所保留，尽可能地仔细回答问题。如果非技术的面试官想让你从非技术角度解释一些东西，你可以用比喻的方法（参阅1.2.2节）。请记住一条原则：你的回答一定要建立在现实的基础上，使用真实名称和恰当术语来作比喻。

1.2.2 使用比喻

有时候，非技术面试官会评估你用非技术语言解释技术问题的能力。对于这种情况，使用比喻是最佳方案。

例如，面试官要求你用非技术语言解释什么是IP地址。维基百科对IP地址的定义如下：

网络协议地址（IP地址）是每个设备（计算机、打印机等）在计算机网络中所分配的一组数字标签，使用网络协议进行通信。

你想到的第一种比喻应该是IP地址很像通信地址。送信需要通信地址，而传递数据包需要有IP地址，因此这种比喻是说得通的。但是，不同国家的通信地址在格式和风格上有所不同，而IP地址则严格遵循全世界通用的一套标准。或许对IP地址更好的比喻是由经度和纬度构成的地理坐标？

比喻能够帮助非技术人员理解一些技术问题，但这种方式并不完美，存在很多局限。因此，使用比喻不要无限度。例如，如果用经度和纬度去比喻IP地址，那么要知道，你并不是想说IP地址是由计算机或网络设备的实际地址所决定的。

1.3 准备简历

一份好的简历能让你通过招聘中介和人事部门的筛选，它就像一块敲门砖，让你有机会与面

试官谈论职位职责。不过简历本身并不会帮你获得一份工作，换句话说，人事经理永远不会只凭简历而雇用一个人。

1.3.1 囊括相关关键词，注意上下文

众所周知，大多数非技术性招聘中介会根据关键词来筛选简历，因此你需要确保简历中的关键词和招聘广告中的关键词有所联系。懂技术的招聘经理可能明白有ECMAScript标准工作经验的人对JavaScript肯定也会有深入的理解，但是普通的招聘人员看不到这种联系，除非你在简历中明确写上关键词“JavaScript”。当然，你也要尽量避免胡乱地在简历中填满关键词，请确保所有的关键词都出现得恰到好处。

1.3.2 文笔要好

文笔差会给你带来巨大的劣势。所写内容一定要简明扼要，你自己（或者请朋友）务必反复校对文字，遇到表述不清晰或文字冗余的情况，要重写或干脆删除。

不知道如何组织文字的时候，我会尝试把想表达的内容讲给朋友听——这非常有效。如果你没有朋友，那就讲给自己听，然后把用到的词汇记下来。不要在意初稿不够好，因为下一步就是修改。行文中展现一些个性不是坏事，但也别做过头了。要记住你觉得有意思的事情，别人可未必这么认为，不要自作聪明地把那些东西全写进简历里，而是在面试时和面试官建立默契之后，再去表达你的个性。如果你把握不好这个分寸，那就用简单直白的写作方式，这比较稳妥。

有些经理不太注意拼写和语法错误，但也有人（比如我）会非常在意。我之所以厌恶这种错误，是因为在多年的代码审查经历中，看到过太多充满拼写错误的代码。文本编辑器都有拼写检查功能，因此，不要忽略那些有红色波浪下划线的文字！

想要写出好的文字需要耗费一些精力，因此不要急于求成，多拿点时间去写作和修改。

1.3.3 对工作经验作出解释

如果你声称具备某种工作经验，或者精通某种技术，招聘经理需要知道你是如何获得的。如果简历中没有说明如何获得了“5年ASP.NET工作经验”，就会显得虚假。在你声称具有某种经验之后，面试官就会想了解你是在哪里获得该经验的。就我个人而言，就算没有看到这部分内容，也不一定会认为你在撒谎；但这毕竟对你非常不利，特别是我手里还有其他简历，而别人清晰说明了这些工作经验。

如果你声称擅长某件事情，就应在工作经历和教育经历中有所体现，尽量明确、详细。同时也要明白，在当今的社会，职位和头衔没有什么意义。“数据分析师”有多种含义，并不一定代表具备某种分析技能；“网络开发者”可能是具备编程技能的人，但也有可能是只会使用

Photoshop的人；“程序员”应该是会写代码的人，但我也见过只用下拉菜单做系统配置的人挂着这个头衔。你需要描述清楚自己掌握的技术，以及你是如何运用这项技术的。

1.3.4 不要听信“简历不能超过两页”的建议

只有当招聘人员专注度不高或是根本不关注简历细节的时候，才需要控制简历长度。不要盲从“简历一定要短”的建议。如果你有丰富的工作经验，就应该全部体现在简历上，作为一名招聘经理，我很乐意看到这些。更重要的是，你丰富的工作经历，应该引以为傲，不要羞于表达。

这并不是让你什么都往上写——别忘了简历要简明扼要，措辞和行文一定要尽量简洁高效，巧用关键词和总结。

1.3.5 着重强调招聘广告中的技能

着重强调招聘广告上的相关技能没有任何问题。作为一名求职者，我花了很长时间才明白这个道理：根据不同职位强调相应的技能是一个非常明智的做法。我曾经认为这是耍小聪明的手段，但现在再也不会这么想了。以招聘经理的角度来说，如果一份简历清晰地展示了符合职位描述的技能，这份简历就会比那些内容混乱的简历更具有吸引力。

换言之，强调相关技能对求职者和招聘人员都有利。

如果你在很多领域都有工作经验，就可以考虑多写几份简历，分别强调不同的工作经验。譬如说，你可以用一份简历突出数据库相关的工作，另一份简历则展示你的业务分析技能，根据招聘岗位投送不同的简历。这么做的好处是：看到招聘广告，你可以快速申请，而不必回头重新修改简历。

1.3.6 工作经历间不要留有情况不明的空白期

在简历中，不要让经历间留下空白期，特别是很长的空白期，这会让人觉得你心里有鬼。如果你在非洲做过志愿者，这非常棒；如果你曾经回学校重新读书，或者践行某个创业计划，甚至回家去带小孩，这都是非常好的事情，没有任何问题。你只需证明自己可以胜任这份工作。当然，在某些情况下，如果你有空白期，可能薪水会少些，但明智的雇主会把这种情况看成是一种机会，而不是什么麻烦，而且他也明白个人生活经历同样是有价值的。

1.3.7 “读书、听音乐、看电影”

在我翻阅过的数百份简历中，估计80%都有“个人爱好”这一项，而其中80%都还有这么几项：读书、听音乐、看电影。

如果你的爱好也是如此，真的不必在简历中加上这一项。如果你是投圆片游戏(Tiddlywinks)

的世界冠军，那才有点炫耀的资本。通常来说，不要在简历的犄角旮旯里写一堆没用的爱好来展示你的多才多艺。最好在简历的其他部分中，以个人经历的形式来透露个人爱好，或者在后续的面试中展示你的爱好。

假如你曾为某个非盈利组织或者慈善公会做过技术支持，那么这种经历应该写进简历。我认识一位曾经帮助过贫民区孩子的人，作为招聘经理，我对这类经历非常感兴趣。任何能展示热情、无私等崇高品质的经历都可以写进简历，但要避免过于枯燥。如果你觉得这部分内容过于牵强，那就不要写了。

1.3.8 条理清晰

通常来说，简历的结构一定要尽量简单、有条理，以便招聘者快速阅读。简历的主要作用之一，就是让其他人能够逐条清点你的哪些技能和经历符合职位需求。我建议你在简历的开头部分好好写一个概要，要包括你的主要技能和经历；接下来是主要工作经历部分，以时间顺序详细描述你的工作经历；再往下是其他经历部分，可以包括志愿者或公益活动经历；最后一部分应该是教育和培训经历。

简历的每一页都应该包含你的联系方式：姓名、电子邮箱和联系电话，可以放在页眉或者页脚。联系方式不一定要多么吸引眼球，不过要保证招聘人员在想要联系你时能够轻松地找到你的联系方式。

1.3.9 应届生简历

由于相关工作经验比较少，你需要列出做过的（在校和校外）项目以及自己的职责，以此来展示你的专业能力。如果你是项目带头人，起到了非常重要的作用，解决了非常困难的问题，或者设计了一个复杂的组件，那么请把这个经历当作一项成就放在简历显眼的位置。如果有人评估过你的工作，或是从中得到了帮助，加入他们的评价会使简历的可信度更高。

1.3.10 工作经验过多的简历

有工作经验是非常好的，没有什么比工作经验更有说服力。然而，工作经验过多，别人也会有不同的看法。如果你在一家公司工作20年不跳槽，别人可能会认为你没有遇到过职业挑战。但实际上，在这20年的时间里，你战胜了很多难以跨越的困难，这些应该在工作经验中着重强调。在每条经历的下面，还应该包括如下的副标题：

- 由你负责的重大项目；
- 承担的各种任务或主要职责；
- 曾经应对过的机构变动。

1.3.11 回归职场的简历

如果此前你曾经离职，去抚养孩子、旅行或深造，那么你完全没必要隐瞒这些。在简历上把当时离职的原因写清楚，一句话就足够了。在工作空白期，如果某些方面的技能有所退化，也要如实相告。

不论你是因为公司裁员而失业（公司因为经济问题而缩减人员），还是被老板开除，都应在简历中如实描述。你不必使用诸如“解雇”（如果有过这种情况）之类的字眼，但也不要隐瞒事实或者声称自己是主动离职的，招聘经理对这种小伎俩非常敏感。相信我，如果招聘经理觉得你弄虚作假，你是不会有任何就职机会的。

裁员经常发生，人们也偶尔会被解雇。在简历上如实地简单讲明，是这种情况下的最佳策略。

如果是被解雇而离职，可能你会失去一些机会，现实就是如此。在简历中避免使用“解雇”这类字眼，是我能给你的最实用的建议。当被问到为什么离职的时候，要有一份准备好的合理答案。好的答案大体是这样的：

“由于和团队领导意见不合，我被解雇了。我用了他不同意的一个方案，之后我们就彻底决裂了。这经历让人很痛苦，但我也从中学到了很多。”

说到被解雇，你可能会情绪比较激动，这是很自然的事情，但有件事一定要记住：不要对前任雇主有任何怨言。即使面试官主动提及前任雇主，也不要这么做。通过你叙述的事实，面试官可以自己得出结论。

从这个角度来想：如果你在面试中表现得镇定、稳重、理智，那么面试官自然会乐意相信你的故事；如果你很情绪化，口无遮拦、无所顾虑，面试官则会更倾向于同情你的前任雇主。

这种问题比较麻烦，请务必提前思考如何回答，它一定会在面试中出现的。

1.3.12 简历的常见错误

“千里之堤，溃于蚁穴”。招聘经理对你的第一印象是通过简历形成的。在一开始，你的简历会和其他简历摞在一起，大部分招聘经理会从这一大摞简历中筛选出一批简历。因此，他们必须有一些过滤方式。在其他条件相同的情况下，简历中细小的错误可能会导致被拒。接下来，我们来看一些你需要尽量避免的常见“小错误”。

1. 拼写和语法错误

多处拼写和语法错误可以毁掉一份其实很优秀的简历，这种情况经常发生。请确保你校对自己的简历，最好请一位细心的朋友帮你检查。如果英语不是你的母语，即使花钱也要请一位母语是英语的人来帮你审阅简历。

2. 成绩描述含糊不清

描述自己的成绩的时候，记住两个原则：第一，一定要具体；第二，要以他人能理解的方式来描述。例如，你发现并修复了一个框架组件的性能问题，但有些人并不熟悉这个组件（所使用的技术）。如果你能说清楚组件的性能究竟提升了多少，而不是你用了什么技术，就会更容易给别人留下深刻印象。与其描述如何实现一个排序算法，不如多谈谈“200%的系统性能提升幅度”。等到了面试，你还会有很多机会去展示你的专业水平。

3. 排版混乱

在所有的错误中，发现和修改排版问题最为简单。把你的简历拿起来（不用阅读内容），扫视页面，看能不能找到简历的重点。如果找不到，那就说明简历不是排得太密就是排得太空！第一眼看到的不是简历所表达的重点，说明简历的结构不够好。

在你想重点表达的文字旁边留出足够的空间。不论如何，都不要用小号字体来试图放入更多信息，那会让页面显得很拥挤。

4. 电子邮箱格式不规范

对于“jojo_2hot4u@hotmail.com”这类邮箱地址，不论你的朋友觉得多酷，都不要把它写在简历上。试想，陌生人看到这样一个邮箱地址会怎么想？要知道，他只能通过简历的表达方式来形成对你的初步印象。

除非你的名字拼写和热门域名一样，否则购买同名域名并不贵，空间的维护成本也不高。“名@姓.com”或者“@姓名.info”格式的邮箱地址都是可行的。

如果你不想花钱购买域名空间，可以考虑注册一个相对“专业”的电子邮箱。网络上有很多免费的电子邮箱供应商，比如谷歌和雅虎。

1.4 使用招聘网站

所有招聘网站的功能都类似：你能上传简历，让雇主（和招聘人员）可以搜索到；你可以浏览、搜索职位，订阅新发布职位的邮件通知，还可以在线申请职位。

这些网站对应聘者一直都是完全免费的，它们汇集了数量庞大的企业用户群体（招聘人员和雇主）。统计表明，相对于传统的简历投递方式，利用招聘网站，潜在雇主看到你简历的几率会高很多。

这些网站的搜索功能非常有用。比如，你可以添加工作地点、工资、职位级别等限制条件来进行搜索。

招聘网站上的简历浩如烟海，你的简历很有可能没有人注意。加入到寻找工作的洪流之中，想让自己的简历脱颖而出更是难上加难。投出去的简历石沉大海，你别提有多沮丧了。

不幸的是，稍加留意就能发现一个事实：招聘网站上几乎所有的招聘广告都是中介发布的。这意味着你几乎不可能直接同雇主打交道。有时候（这种情况相当少），如果中介对你的简历特别感兴趣，就可能为你量身定做一个求职方案。然而在大多数情况下，这些中介往往是横亘在你和面试官之间的一座大山。

另外，有些无良中介还经常伪造一些招聘广告发布到网站上，这在某种程度上影响了招聘网站信息的可信度。这些中介以此来搜集求职者的简历，然后把这些简历整理一下就转手卖给相关客户。这种情况可能并不常见，但也别天真地认为这永远不会发生。

要提高警惕，在你申请了一份“完美”的职位之后，如果长时间没有回应，就不要浪费太多精力了。这份工作可能压根就不存在，是中介伪造出来的。

主要招聘平台的比较

所有招聘网站的操作方法都差不多，优缺点也类似，因此可以依照个人喜好来选择使用哪个网站。通过表1-1和表1-2，你可以看到哪些网站发布的程序员职位比较多。我用的搜索关键字是“程序员或软件工程师”，职位发布时间限制在7天以内，搜索日期是在2012年6月。

表1-1 美国主要招聘网站

网 站	职位数量
CareerBuilder.com	1950
Monster.com	1000+
Jobserve.com	718
ComputerJobs.com	685
Dice.com	152

表1-2 英国主要招聘网站

网 站	职位数量
CWJobs.co.uk	3018
Reed.co.uk	2439
Jobserve.co.uk	930
Monster.co.uk	531

1.5 招聘中介

招聘中介的职责是帮雇主寻找合适的人选来填补职位空缺。最初，所有中介都会对你的职业前景表现出浓厚的兴趣，但他们到底能出多少力，则取决于你在招聘市场上抢不抢手。如果他们手里没有适合你的职位，那么你跟他们可能就只有一面之缘了。

几乎所有中介都是靠碰运气的方式来运作的。当推荐的人选被录用之后，他们会向客户（雇主）收取一定比例的酬金，而酬金多少跟被录用者的工资是有关系的。如果中介没能为雇主解决

职位需求问题，就不会得到任何报酬。不同招聘中介收取的费用差别很大，但通常是新职员第一年薪资的10%~30%。这看起来像是个工作量不大却很赚钱的差事，但实际上这个收入是非常不稳定的，何况雇主也愿意花这个钱，中介的要价自然有他的道理。

求职者通常不需要付钱给招聘中介。如果有人让你交钱，就一定要多加警惕。

为了得到酬金，中介会在很多招聘网站上发布招聘广告，也会利用自己的数据库来搜索与雇主主要求相匹配的求职者。大部分招聘人员也会通过自己的关系网来传播各类招聘信息，希望能藉此吸引到合适的求职者，然后引荐给自己的客户。

有趣的是，中介公司中负责招聘软件开发人员的人往往自己没有直接的软件开发经验。在雇主和程序员之间周旋，他们的角色完全是销售和市场拓展人员。当然，也有些资深中介懂一点基础的编程词汇，这让他们看起来不至于显得太外行；但你是程序员，同他们打交道的时候，你会发现总得向他们解释一些非常基本的技术常识。你需要记住的是，虽然中介可能不懂技术，但你对他们说的话会被原封不动地转达给潜在的雇主。为了避免中介胡言乱语，最好不要谈论太多有关工作经验的细节，把这些信息留到同雇主交谈的时候再去表达。好的中介会有选择地把与招聘职位相关的工作经验传达给雇主，他们知道自己缺乏技术常识，因此得到雇主反馈后，会记下有关技术的问题，然后原封不动地传达给你。如果你发现某个中介“瞎编”你的工作经验（有些真的会这么做！），那最好立刻和他们分道扬镳，他们不仅无法帮你找到合适的工作，甚至还会破坏你的名声。

在把你介绍给雇主之前，有些中介公司会对你提前面试。面试过程会非常有趣，他们一点也不懂你简历中的技术内容，却爱问那些技术相关的问题。也有些中介比较务实，知道自己没什么专业知识，于是就只问一些普通的问题，比如你的工作动机、个人理想、出勤率、病假记录等。值得注意的是，除了你的专业知识和工作经验，中介还想了解你的个性，以此来推测你和哪位雇主合得来。这是中介公司“挖掘亮点”的方式之一，但这种由主观判断形成的印象并不可靠。

如何有效地同中介打交道

中介是从雇主那里拿报酬的，因此他们帮助你的动机取决于他们是否看好你的前途。同中介打交道，不论他们提供的职位多么诱人，都不要带太多个人情绪。当然，不论中介公司看起来多么高效，都不要把自己束缚在一家中介公司上。

另外需要注意的一点是，中介的角色是求职者和雇主之间的中间人，你说的每一句话都会被传达到雇主那里。虽然你需要说服中介，让他相信你的证书和经验都是真的、有价值的，但也可以在不利于求职的信息上有所保留。你不应该回避一些明显的问题，比如换工作间隙的空白期，但也别主动展示负面或者“复杂”的信息，因为有可能会被中介曲解后传到雇主那里。

在等待反馈期间，如果中介迟迟不通知你，就不要把时间浪费在焦虑的等待上。如果他们有反馈，一定会马上通知你。要知道，如果你得不到工作，他们也别想得到报酬。另外，中介其实

也是任雇主摆布的，因此他们和你一样，在等待通知的期间也非常煎熬。同中介保持联系没坏处，打电话的频率保持在每周一次就行，来确认中介没忘掉你，同时也可以展示自己求职的热情；不过一直催促中介是毫无用处的。

1.6 自己搜索职位

如果不依靠中介公司求职，那么可以自己寻找职位、联系雇主。这件事并不如你想象中的那么难，毕竟大多数雇主恨不得全世界都知道自己的公司在招聘。很多公司的网站都会刊登招聘职位的广告；大公司可能和很多招聘中介有合作关系，或者人事部门有一支优秀的团队负责所有招聘事务。但我还是要再三强调，大多数招聘经理都欢迎合适的求职者毛遂自荐，只要注意联系时要体现出对人的尊重，内容也要尽量个性化。

如果你直接同招聘经理打交道，你的申请可能会被转移到人事部门处理。不必因此而沮丧，招聘经理很可能是出于公司“效率和公平”的规定而这么做的。除非你觉得自己不受欢迎，否则在人事部门处理申请的期间，也要同招聘经理保持联系。请记住，在你成为公司的正式员工之前，公司的规定并不适用于你。在沟通过程中要保持尊重和坦诚，不要因为公司的形式化流程而失去耐心。

有一点一定要注意：不要发送无意义的信件或电子邮件。对雇主来说，这种做法和那些恼人的中介没什么区别。这些中介总是发送各类“难得的机会”和“热门求职者”的垃圾邮件给雇主，因此雇主对这类邮件早就没感觉了。在给雇主写邮件的时候，你的内容应该尽量个性化，不要用“敬爱的某某”这种死板的信件格式，而是直接说出你的求职动机。如果你没有好的想法，可以问自己几个问题：我用过这个公司的产品吗？公司业务的发展潜力怎么样？我有没有从朋友或者新闻里听说过这个公司的趣事？

寻找雇主的最好方法是利用你的关系网。也许你够幸运，多年的经营已经有了很大的社会关系网；也许（更有可能）作为一个程序员，你的社会交际面比较窄。众所周知，人脉资源是巨大的财富，特别是在求职期间，作用巨大。但是我们中有多少人（包括我）具备这种财富呢？我们都愿意把时间和精力浪费在拓展和经营人际关系上。

如果你没有广阔的社会关系网，因为你是（我就直言不讳了）一个腼腆的人，也不要绝望，下面的内容就是专门为你除准备的。老实说，我真希望我年轻的时候就能遵循这些建议。

1.6.1 内向者的关系网

你会发现很多讲人际交往的书中大部分建议都有一个问题：都是外向者写给外向者看的。同很多其他程序员一样，如果你不是天生的外向者，那么这些建议都是没用的。更糟糕的是，如果你做不到书上那些“简单又有效”的事情，就会有负罪感，并且自尊心受挫。

不要自责，相反，作为内向者，要发挥你的优势。

内向者通常善于思考，喜欢花大量时间来思考问题，而不是谈论问题。准备和人沟通的时候，这个特质是内向者的最大优点。如果你想要和某人取得联系，准备一些深思熟虑的问题来问他。大多数人都喜欢谈论自己，并且同他人分享自己的见解。这并不是在批评人是自私的，而是源于人类心理学：分享个人经历是人际关系中的基本要素。在恰当的时间，提出有深度、开放式的问题，可以避免冷场，让人与人的相处有一个完美的起点。

内向者更喜欢同人一对一交流，不喜欢在众人面前说话，而从意见交换和关系建立的角度来说，一对一交流恰好是最有效的方式。寻找一个可以一对一交流的环境，即便在公共场合也是一样。出人意料的是，微博的公共空间是内向者进行一对一交流的良好环境。140字的限制对喜欢思考的人来说完全没问题，他们可以用简洁的语言来取代冗长、复杂的对话。拥有粉丝之后，你既可以直接回复他，也可以给他发私信。本章稍后会介绍这些，你可以学习如何使用微博和其他社交工具。

很多内向者对于发展关系网这件事都有点胆怯，这完全没有必要，就把它当作一个游戏就行。游戏的目的是同人打交道，建立有意义的合作关系。对于内向者来说，这个游戏最难的部分就是，“交际原则”是捉摸不透的，内向者非常想知道“具体怎么做”，“我是否正在建立关系网”。如果你性格内向，就会一直胡思乱想、心烦意乱：“我下一步到底该怎么做？”如果你是外向者，那么这些想法都不是问题：“我想怎么做就怎么做！”（外向的读者可以跳过这部分内容。想看一下也无妨。）对于内向者，下面的速成教程可以教会你如何建立关系网。

- 列出所有潜在联系人。
- 按照从“最想交往”到“最不想交往”进行排序。
- 从头逐一思考哪些问题对其来说很有趣，把问题列出来。
- 在合适的环境下，向5个联系人询问对应的问题（超过5个可能有点难，这取决于你的个人能力和精力）。
- 当对方有回应的时候，与之交流，建立联系。
- 持续添加联系人到你的列表中，保证一直有5个新的潜在联系人。
- 不断重复！

最后，如果你发现人际交往太分散你的精力，就要注意调整自己的步调。拿出时间去思考和总结社交过程中的信息，通过这种方式来准备下一次交际机会，这样才能建立并维护好你的关系网。这个过程就像马拉松赛跑一样，不要急于求成。

1.6.2 寻找雇主

潜在雇主其实到处都是，不幸的是，很多雇主都不假思索地把招聘任务完全委托给中介，或者（公平地说）他们只是根本不知道有什么好方法，因此只能依靠中介。

请从招聘经理的角度来看招聘这件事：一个合适的求职者直接同公司打交道，这难道不好吗？这难道不会节省很多时间和金钱吗？太正确了，就是这样！这里有一些渠道，可以帮助你把

潜在的雇主挖出来：

- 你的个人社交网络，包括Twitter、Facebook和LinkedIn；
- 本地的公司名册；
- 大型招聘会；
- 招聘网站，只搜索由雇主发布的广告；
- 用“职位”和“招聘公司名称”进行搜索；
- 专业性的杂志期刊。

找工作不要遮遮掩掩，把你的网络个人介绍改成求职中，说不定就有人会注意到。

1.6.3 接近雇主

对很多求职者来说，在完全陌生的状态下接近潜在雇主可能是最困难的事情。如果条件允许，请其他人（不是收费的中介）帮你介绍雇主认识是非常好的方法。如果你有LinkedIn，可以看看有没有人能帮上忙。当然，也要问问身边的人，利用到所有的社会关系。你可能听说过“六度空间”理论：每个人都可以通过约6个人认识到其他任何人。

如果没有关系渠道认识雇主，你就需要直接同招聘经理联系。不要害怕，这种方式没有任何问题。你要做的第一件事是查询尽量多的信息，包括公司、意向部门，以及影响和决定招聘的人物。请注意，我说的是“尽量多”，而不是花大量的时间调查公司底细。时间分配要合理，几个小时就足够了。在这期间，你可以得到所有有用的信息，对于公司的结构和决策者也会有一些深入的了解。

显然，通过频繁浏览公司网站，你能查到大部分需要的信息，至少能找到一个联系电话。直接打电话，不要隐瞒身份，开门见山地表明自己是求职者，并询问接线员能否帮个忙。你只需简单介绍自己，如果足够坦诚，大部分人都会乐于帮助你的。很多前台接线员和你（求职者）的立场都是一样的，对求职者有同情心；但是你要小心，如果你有任何欺骗或隐瞒的迹象，你们的对话时间就不会太长。我坐在接线员旁边的时候，见过太多这种情况：公司会接到销售人员和中介打来的没完没了的骚扰电话。对于这类电话，大多数接线员养成了一种非常敏锐的直觉。如果一位没有经验的销售人员请求接线员把电话转接到软件开发经理那边，原因是想谈一个“商业机会”，那么毫无疑问，接线员不会转他的电话，他的信息只会被记上便签，而且会被很快丢弃。不要让这种事情发生在你身上！

我不建议你通过电子邮件的方式和公司建立第一次联系。回忆一下，你第一次收到莫名其妙的邮件时是什么感觉？所以，你需要拿起电话，同相关人员直接对话。第一个电话的目的非常简单，你需要知道是谁负责招聘程序员，还有他的姓名和联系方式。如果够幸运，你还能了解到公司结构以及招聘需求。如果无法得到这类信息，至少要拿到招聘经理的姓名和邮箱地址。如果第一次通话顺利，你可以询问接线员能否帮你向招聘经理转达信息：我是个软件开发人员，正在求职，希望有机会与你交流。不必担心信息没有传达到，虽然确实有这个可能，但更有可能的是，

这条信息可以直接让招聘经理认识你，你不必费力去找关系。

1.6.4 坚持不懈

实习销售人员在接受培训的时候，会将一个理念深深印入脑海：不断坚持一定会有所回报。坚持是一把双刃剑，它确实可能带给你回报，但也可以坏了你的名声，给你贴上“狗皮膏药”的标签。坚持和纠缠只有一步之遥，但有一个基本原则：当你第一次遇到挫折的时候，不要放弃，这就是坚持。

这可能意味着你需要多打几个电话来得到想要的信息；可能意味着你需要发送几封后续邮件来确认情况，而不是胡思乱想；也可能意味着你的邮件被不小心归入了“垃圾邮件”，需要直接向招聘负责人邮寄一封信（不论是哪种情况，我都十分推荐这个方法）。最后，你必须评估自己的行为有没有触碰到“警戒线”。请记住，如果你一直保持礼貌、谦虚、坦诚，对方就没有任何理由视你为“狗皮膏药”。

1.6.5 把握时机

媒体会报道很多大公司的有关新闻。如果你想加入大公司，就应该多注意这些新闻，特别是有关公司扩张计划的新闻。例如，很多上市公司会在获得大合同之后大肆宣传。看到这些新闻之后，可以提交（或者重新提交）一份职位申请。这样，获得有利反馈的几率会上升。如果在其他时间提交简历，回复可能就是“目前无合适职位”了。

另外，大多数公司的扩张计划是与财年吻合的。如果你发现财年开始了（上市公司的这类信息都是公开的），公司的招聘流程也会随之启动。趁此期间递交职位申请，你被录用的几率就会更高。

1.7 其他途径

你若常上网，一定听说或使用过几个社交网站。下面会介绍一些流行的社交网站，告诉你如何利用社交网站去寻找一份程序员的工作。

1.7.1 Twitter

Twitter是一个供用户发布和交换信息的社交网站。这些信息被称作“状态”，字数限制在140字以内。Twitter声称有超过1.4亿的活跃用户，这些用户每天会发布3.4亿条状态。

当你在Twitter上寻找潜在雇主的时候，不要忘了大多数人使用Twitter的目的并不在此，而是为了同他人分享好玩的事情、有趣的想法、有用的信息。人云亦云和不带个人色彩的乏味内容，不会使你的Twitter账号有太多粉丝。为了让你的Twitter变得有趣，你得跟随潮流，多分享自己的

观点。不要在乎你的观点受不受欢迎，不要把那些不受欢迎的观点（如果有的话）藏起来。正相反，要多表达你的想法，在互相尊重的基础上，怎么想就怎么说。满世界给别人发送你账号的网址链接不会给你带来多少粉丝。

如果你是个Twitter菜鸟，先花几天时间多读读其他人的状态。观察哪类状态最流行，被转发次数最多，也要注意哪类状态很冷清，没人愿意看。

虽然花钱就能买一些粉丝，但我不建议你这么做。你买的粉丝对你根本没兴趣，你发的状态他们也不会看。也可以买刷粉丝的软件，这种软件可以自动添加关注和取消关注，这样你就能拥有大量的粉丝了。你可能觉得这个东西很有用，可以小小地满足一下虚荣心，但我同样不建议你这么做。自动操作的账户其实很好与真实用户区分：它们关注你之后，如果你没有关注它们，它就会很快会取消关注。Twitter中的人际关系和现实中的人际关系很像，没人喜欢跟机器人打交道。你真正想要的是同人们建立真实、有意义的联系。

我们来总结一下你应该怎么做。

正确做法：

- 参与到同他人的讨论中来，包括转发那些你认为有趣的状态；
- 更新你的个人资料，包括目前的求职状态；
- 在个人资料页面中添加一条简历的链接（更好的方法是公布你的个人网站网址，这样对方就能找到你的简历，同时也能看到你写的文章）。

错误做法：

- 发布不道德或者有攻击性的状态（包括转发这类状态）；
- 只发布和找工作相关的信息，毫不谈论有意思的话题；
- 关注非常明显的“垃圾账号”和“自动回粉账号”——是的，这能让你的粉丝数量飙升，但同时也让你的账号看起来非常不靠谱。

1.7.2 Facebook

Facebook大概是最出名的社交网站了。你可能看过《社交网络》这部电影，知道Facebook在全世界的普及程度。

Facebook的普及是一把双刃剑。某位雇主可能拥有Facebook账号，通过他的页面，你可以获得一些有关公司文化和理念的信息；但另一方面，雇主也能看到你的页面，因此你曾经发表的一些负面信息可能会对你有不良影响。你可能也听说过，美国联邦贸易协会指控过Facebook“欺骗用户”：Facebook声称会保护用户的隐私，但却不断地跟人分享并公开用户数据。他们最近达成了谅解协议，你可以认为这会更好地保护Facebook用户的隐私；也可以持怀疑态度，认为除非有外部条例的约束，否则Facebook不可能重视用户的隐私保护。

Facebook的隐私保护还有另外一个问题。据媒体报道，有些雇主和研究机构与Facebook有合

作关系，会请求Facebook给他们访问用户数据的权限，以此来寻找合适的员工和学生。这种行为的对错不好判断，即使合法，也是说不通的。很多求职者都很不喜欢这种行为，至少很多用户在发布信息的时候，为了保护隐私，都会有所保留。

总之，Facebook是个非常棒的社交网站，但并不太适合求职者。

1.7.3 LinkedIn

如果你只能选择一个网站寻找编程相关工作，最好的选择是LinkedIn。和其他社交网站不同，LinkedIn的定位非常明确，专攻职场和招聘领域。毫无疑问，LinkedIn是专业人士之间交流的平台。

LinkedIn 的任务是让全世界的专业人士都能建立联系，让工作更加高效，更加成功。

使用LinkedIn的理由一目了然。你可以同其他人建立联系，加入感兴趣的讨论组，及时更新自己的工作经历，偶尔发布一下最新状态。你还可以通过发送InMail的方式，直接同其他LinkedIn用户交流。这种方式非常方便，不知道对方其他联系方式的情况下也可以完成。

LinkedIn还是个展示推荐信的好地方，来自同事和商业伙伴的推荐信都可以显示（或隐藏）在你的个人资料里。

如果每个月缴纳一定的费用，求职者还可以享受LinkedIn提供的更多附加服务，包括：

- 可以发送5个InMail，对象没有限制，可以是任何LinkedIn用户（一般只能发送给人脉圈里的人）；
- 可以看到谁查看过你的资料（一般只能看到一部分）；
- 你的资料上会有“求职者”徽章，这是可选的；
- 通过LinkedIn申请职位的时候，你会被作为“特殊申请者”对待，你的简历会排在申请列表的前面，因此职位发布者可以优先看到你的资料；
- 你的个人页面上会显示“OpenLink”图标（可选），招聘人员可以（事实上，任何LinkedIn用户都可以）通过它给你发送信息。

1.7.4 博客

如果你想和潜在的雇主建立联系，写博客是个相当有效的方法。如果你写得不错，并且内容有趣，就会有很多人关注你的博客、订阅你博客的RSS。久而久之，你就能拥有一批固定的读者，这对求职者来说是巨大的优势。

当然，这是理想的状况。

而现实则是，很多博客一开始规划得很好，但在一阵热情之后就陷入沉寂了。为什么会这样？因为写作需要花费大量时间，积累读者也需要很长时间，拥有数量庞大的粉丝群需要数月甚至数

年的时间。

如果你经常使用Twitter等社交网站，那么写不写博客就没那么重要了，因为你在社交网站中发布的那些内容完全可以取代博客。如果没有，就得问问自己有没有时间去经营一个博客。写博客文章需要用心，否则不会拥有太多读者。

开通博客非常简单，很多网站提供免费的博客服务，但我不推荐你这么做。最好使用自己的域名，空间也从收费供应商那里购买，这会让你的博客显得非常专业。更重要的是，这可以让你对博客的管理和内容拥有完全的控制权。如果你使用免费的博客服务，就会给人以“不专业”的印象。比如，有些免费博客网站会为你的博客强制添加一些不相关的广告。类似地，如果你在知名社交网站上写博客，对于内容的控制权就没那么高，因为在你注册的时候，网站有协议（你必须同意才能注册，记得吧）限制你页面的内容。最极端的情况是你压根无法写任何内容。

如果你决定开通一个博客，可参照下面这些建议。

- 为博客选择一个与职业相关的主题。程序员们都很幸运，因为有太多的主题可以选择。
不要让博客充满不相关的内容，一个专注的博客才更有可能吸引忠实的读者。
- 使用收费的网站空间，购买你喜欢的域名。
- 为你的写作制定一个合理的计划，要有规律。这样你才可以长期保持写作的习惯。
- 写作计划应该尽量灵活。如果有特殊情况（比如休假或生病），要给自己留出缓冲的时间，暂停写作。
- 使用博客软件。如果你不知道哪款博客软件好用，我推荐你先用WordPress。
- 允许读者评论你的博客文章。是的，这会带来很多垃圾信息，但让读者参与讨论的好处远远大于过滤垃圾信息要花费的时间。人们都喜欢被倾听，因此不要忽略评论。

1.7.5 Stack Overflow

遇到难题的时候，大多数程序员喜欢通过网络搜索答案。很多人都发现，答案可以在Stack Overflow上找到。Stack Overflow是个较新的网站，但已经成为了程序员聚集的圣地，具有丰富的资源。大多数程序员应该都听说过Stack Overflow，而且已经访问过很多次了。

很多人可能并不知道，Stack Overflow最初是从programming Q&A网站分离出去的，而且也为雇主和求职者提供招聘服务：<http://careers.stackoverflow.com/>。

这个网站的使用方法和其他招聘网站类似：你可以提交个人资料，这样雇主和招聘者就能通过关键字和所处地区搜索到你的信息。<http://careers.stackoverflow.com>的独特之处在于，其用户数据和Stack Overflow是共享的。如果你回答过Stack Overflow上的问题，你的个人资料就能显示出很多相关信息，包括“最佳回答”以及各项技术讨论区下的得分。

如果你在Stack Overflow上参与过讨论，这能不能给你的求职加分呢？这取决于雇主的态度。懂技术的雇主可能会很看重这一点，有的甚至会浏览你的发言记录（包括你提过的问题），这样

就能知道你的交流风格和表达能力。还有一些雇主并不关心你在Stack Overflow上的发言记录，他们关心其他东西。比如，相对于工作时间，你花在网站上的时间有多少。如果你花的时间非常多，雇主就可能认为这是一个问题。如果有后续的电话或者个人面试，你必须提前准备如何应对这个问题。如果你是利用离职时间泡在Stack Overflow上，这可能就没关系了，但最好还是考虑一下这个问题，保证被问到的时候能够从容回答。

1.7.6 Skills Matter项目：“寻找你的师傅”

程序员求职的方法有很多种，除了那些常规的方法，还有一个有趣的方法，那就是位于伦敦的培训公司Skills Matter的项目：“寻找你的师傅”。

基本流程是这样的：雇主花钱为公司预约一个交流会，观众可以免费参加，并且全部都是程序员。

这种活动看起来和传统的招聘会差不多，但存在一个本质区别。由于观众基本上都是在Skills Matter上听说这个活动的资深程序员，会议内容会更加专业。雇主不会宣传企业的丰功伟绩，而是更关注于技术和公司独特的编码环境。讲座的人一般是公司的技术总监，而不是人事部门的员工。

根本上来说，“寻找你的师傅”和其他类似项目都是为有上进心、有能力的程序员服务的。如果你认为自己是这种人，并且离伦敦不远，那就可以考虑参加这类活动。

如果你的住址远离伦敦，可以联系自家或目标公司附近的程序员培训公司，看看他们有没有类似的活动。