



<> Code

Issues

Pull requests 2

Actions

Projects

Security

Ir



master



advanced-programmer / docs / proverbs

/ 这些年在互联网的经历-送你一些不错的建议.md



Snailclimb 结合自己这些年在互联网的经历,送你一些不错的建议

5 years ago



140 lines (74 loc) · 11.9 KB

Preview

Code

Blame

Raw



来源: <https://www.zhihu.com/question/312019918/answer/652749310>

作者: [Tamic](#)

本人普通二本毕业去的一家小公司,半年后去了某知名大厂,后来继续在大厂之间跳槽。如今已娶妻生子买房定居。这几年经历过面试,买房,学车,结婚,生子,职业沉淀。也算是真正领悟到了编程到底有什么意义!编程的意义这个话题留到以后再谈。

每年都有无数年轻的程序员,加入软件行业。他们在学校里学过编程,但是对这个行业的现实一无所知,我今天介绍这个行业的一些实际情况。这几年自己在技术博客沉淀很多技术。其他方面也算成功,至少和同龄人和自己起点一样本科生来说,作为过来人有点说一下:

关于跳槽的我在专栏写过文章: [Tamic: 年后离职跳槽指南](#)

前面四点主要针对三年经验以内的。后面的几点写给3年到8年的同行,最后一段写给所有的同行。

## 1.不要迷信“大牛”

看到你问题中有什么踩坑的,技术人唯一的不好就是对非技术盲目崇拜,尤其对非技术网红。就放前面说说:网上的“大牛”多半不靠谱。

其一是真正的大牛没时间混迹在网上做人生导师,天天讲道理,整天出现在你Timeline里的人要么半吊子水货,要么另有所图(赚钱智商税)(我黑起人来连自己也不放过)

其二相比较入门的你比就算是大牛,他的经验对你未必有用。大牛说我学 Python 也就翻了个基本语法,当天就上项目开干了。(大神说我觉得没有顺手的语言,就发明了 Python)大牛说我从不看基础知识,因为大牛天天忙着公司架构哪来机会看基础。

## 2. 少说废话，多写代码

废话有很多种，程序员届最多的大概就是整理哪个语言/框架/平台/技术好，准程序员则喜欢问，我是不是应该学Java Python php.....学了是不是有用之类。我学习什么技术好？技术讨论并不完全无意义，但在你打好计算机基础、数据结构，熟练掌握至少一门语言、开发过一个能看的项目之前，这些讨论对你来说就是浪费时间的废话，而不是今天跟风学这个，明天又听人说那个好。不要等毕业的时候说出XX技术学校没教过，所以我不会这种话。

有这时间多写点代码，刷刷算法题，找点项目做做，找不到就从自己日常身边找点需求去用代码实现。不要说你觉得没有什么可写的，如果是这样，那你要反思下自己适不适合编程这条路。

我能告诉你的是 接下来几年内不会算法是不行的。不要说我进去公司用不到这些话 如果你拒绝学 那没必要做程序员了。

华为，头条面试对算法要求越来越严格，或许几年后全民算法。不代表我本人对算法有什么兴趣，只是是市场需要。

## 3. 做一个程序员，不是码农。

码农的工作和搬砖没什么区别，就是把一些模块拼到一起，有问题了就东修西补，程序能跑通就不错，只会用教过的一两种技术。做个码农很容易，天天改改Bug 水水群，编程5的分钟 扯淡两小时。就是说的是码农。但程序员是个长期的职业，你在学校学习的是编程的思想、解决问题的逻辑、持续学习的能力。重视基础课程，深入学习一两门语言和技术，理解透彻了，以后学习别的新技术自然触类旁通。没事看看就行，别去跟风凑热闹。你上的是大学，不是技能速成培训班。

总结一句，码农喜欢说颠覆 喜欢说取代 更喜欢扯布道者。天天喊着新技术。

程序员专注基础能力，谈业界影响，说贡献，谈价值。

## 4. 短期不要高估自己，长期不要低估自己

绝大多数人都会犯的错误就是“短期高估自己，长期低估自己”。除非你天资聪慧，否则别想着大一就捣鼓出什么牛逼程序来一鸣惊人。学编程的过程会经历数个瓶颈期，中间你也许会数次感到万分绝望，甚至开始怀疑人生。

但你要相信，只要坚持写下去，瓶颈总会突破，成为一个超越平均水准的职业程序员是没有什么大问题的。

毕业三年拿到15到20k 的人不少，不代表你牛逼，因为行业曲线就这样。但是你想6年拿60k 确实没几个人，毕竟50w年薪是大部人的天花板，所以也别灰心。

## 5, 最后相信自己，尊重自己的职业

- 不要自称「xx 狗」，尊重自己

- 不要为了谈恋爱而谈恋爱，不要相信「程序员找不到女朋友」
- 除了计算机知识，也要学好英语 + 数学
- 基础学扎实，学算法，不要追求热门
- 不要迷恋语言，不要陷入语言的战争
- 多写有质量的代码 别瞎折腾实践

## 6. 90%的编程工作来自内部软件

计算机专业的学生，可能有一种错觉，觉得大部分程序员，都在编写公开出售的软件或者通用软件。

这种看法是不对的。大部分程序员，实际上编写的是不公开的企业内部软件，这就是所谓的外包。比如企业内部聊天工具，追踪费用的软件、优化装运成本的软件、帮助记账的软件、设计新部件的软件、计算保单价格的软件、识别恶意订单的软件等等。

各种各样的商业公司，开发内部软件，解决它们自己的问题。市场上对程序员的大部分需求来源于此，只有极少数程序员直接编写面向外部用户的软件。

内部软件的开发(OA B端)，通常非常乏味，令人厌倦。因为它们的技术复杂性低、技术决策非常保守、预算很少、缺乏长远考虑。但是，世界上大部分编程工作都是这种。如果你想入行程序员，你就要有心理准备去接受这样的工作。

## 7 别人雇你的目的，是让你创造利润，不是让你编程

商业公司最关心的（或者说唯一在乎的）事情，就是增加收入、降低成本。因此，它们实际上需要的不是程序员，而是能够帮助它们增加收入、降低成本的人。

开发优美的软件，解决技术难题，编写没有 bug 的代码，这些都不是商业公司的目的。它们雇佣你，是为了让你帮它们完成某个可以增加收入、降低成本的项目，而不是为了让你追求个人的软件成就。

你对于公司的唯一价值，就在于能多大程度上为它们增加收入、降低成本。

这就是为什么阿里p8拿不到业务就是滚蛋的情况，在没有业务就没有产出，没产出就没有利润，那么技术好坏 只成为你不容易被裁的几率。

## 8 好的程序员专注于为公司带来价值

很多公司的开发经理不懂技术，在他们心目中，程序员就是一群高成本的劳动力，只会在一台复杂的机器上干一些他们难懂的事情。

如果你自称为“码农”，只会写代码。当公司需要压缩成本的时候，某些经理首先就会想到解雇你，因为你的工资高。

有一家公司叫 Salesforce，口号是“没有软件”，意思就是如果经理们购买了他们的服务，就不再需要别的软件管理销售业务了，也就是说，不再需要自己雇佣程序员了。

正确的做法是，你应该把自己描述成与增加收入、降低成本有关系的人，比如“xx产品的开发者”或“改进者”。有一个 Google AdSense 程序员的自我介绍，是这样写的：“Google 公司97%的收入，与我的代码有关。”

## 9 不要给自己设限

年轻学生经常会问，应该选择哪种语言或平台？Java 是不是比 python 容易找工作？

过分强调某一种语言或平台，都是不必要的。如果你把自己限定为 Java 程序员或 .NET 程序员，你就已经输了，因为首先你不应该自称为码农（理由见上一点），其次这种限定使得你自动被排除在世界上大多数编程工作之外。

现实生活中，学会一种新语言，只需要一两个月，然后再过半年到一年，你就会变成老手。那时，根本没人在乎你以前用什么语言。

天才程序员是很少的，可是需要天才程序员的工作机会却很多很多，大多数场合都是需求远远大于供给。

这意味着，即使你不是天才程序员，只要你是一个优秀工程师，那些招聘公司也会立刻录用你，因为它们知道招聘到天才程序员的机会微乎其微。（重复一遍，所谓“优秀工程师”，就是你的履历上有一连串可以为公司增加收入、降低成本的记录。）

某些公司的人事部门，会根据某个关键词（比如 Java 或 .NET）过滤简历。虽然这样的公司根本不值得去，但是如果你真的想过这一关，也很容易：投入几个晚上和周末，在你当前的项目中设法用到这个关键词，然后再把它写进简历就行了。

你要知道 语言只是工具，公司为了利润，在任何时候都可能让你转语言。

## 10 如何提高求职时的谈判能力？

(1) 记住你不是在求职，不是在展示编程技巧，而是在推销某种商业问题（增加收入或降低成本）的解决方案。

(2) 面试时，要有自信，要平等的对话。你要的是一个互利的录用合同，不要每次对方提出要求，你都说Yes。

(3) 雇主可能会问“你的上一份工资是多少”，他们其实在说“给我一个理由，压低你的报酬”。你要想好如何适当地回答这个问题。

(4) 要还价。这里不仅仅指钱，还指其它你关心的方面。如果你无法要求更高的薪水，那就试着要求更多的假期。

## 11 创业公司是否适合应届毕业生？

如果你一毕业就加入创业公司，最可能的结果是，接下来几年你都工作得非常辛苦，然后公司悲惨地失败了，你失业了，不得不又去另一家创业公司工作。

如果你真的想去创业公司，应该首先找一家大公司干上几年，攒一点钱，积累一些经验，然后精心挑选一家创业公司，再去实现自己的梦想。

在创业公司工作，一般情况下，你遇到的都是创业者。他们大多数人没有能力在两年后带你；而在大公司工作，你遇到的都是其他大公司的人，他们中很多人将来有能力带你或者帮你介绍工作。

## 12 沟通技能是最重要的职业技能

前面说过，工程师被雇佣，不是因为会编程，而是因为能够创造商业价值。所以，你得让人们相信你能创造价值，这是帮助你找到工作的最重要的能力。这种能力与你真的能创造多少价值，实际上联系不是很紧密。很多情况下，你向别人展示的是对你能力的一种期望，而不是你当前的能力。

所以不要时刻说技术，因为技术只是工作的一种手段，价值才是你的武器。

最后关注技术的同时不要忘了关注自己自身能力的提高。这一行都应该有面对时刻失业的准备，因为大部分的危机来自于体制外的工作当成了体制内。

## 13. 最后有几点道理

1 普通人为了达成某个目标，需要舍弃一些东西，舍得舍得，有舍才有得。

2 职业生涯早期有机会去大公司大平台，还是要去大公司见见世面。很多事情不是一蹴而就的，如果能力不够可以采取曲线救国，只要最终能达到目的就好。

3 技术是根本，没有技术你就没有钱。技术又不是全部，因为你很难搞一辈子技术。

4 眼光和机遇很重要，机遇可遇不可求，但是机遇来了，眼光不够没有做出相对准确的预测，做错了选择，那也是很伤。

5 学历和起点不是问题，其实能力和眼界才是程序员核心竞争力，大厂不一定是铁饭碗。

6 人都需要学习，并能不断敢于挑战自己，才能离成功几率越大。