首先，欢迎来到程序员的世界。在这个世界上，不是有很多人想创造软件并解决问题。你是一名hacker，属于那些愿意做一些有挑战性的事情的人。

　　“当你不创造东西时，你只会根据自己的感觉而不是能力去看待问题。” – WhyTheLuckyStiff

　　对于下面的文字你不必完全接受，所有这些来自一个其貌不扬的程序员。我喜欢把事情做到最好，而不是对原来的东西修修补补。

　　仅仅是因为爱好开始做一些创新，这是一个很好的开始！如果你说“我要先学习一下再开始做”那么你永远不会真正开始。每个人都需要从某个地方开始，所以现在打开你的编辑器开始写代码吧。

　 　下面是一些很重要的建议，虽然有人并不认同，但我保证你看完我的分析以后会站在我这一边。一开始，先不要管算法和数据结构。大多数简单的程序不需要用到 算法和数据结构，所以当你真正需要时再去学习。编程一段时间以后，你就会知道在哪些地方用到他们。这时知道算法的名字并了解它们的功能，然后找一些相关的 论文去理解算法并动手编程实践。如果没有现成的函数库（其他程序员提供的可重用代码），你用自己喜欢的编程语言来实现它。

**0. 选择一门好的编程语言，一门你自认为可以用它快速地做出一些有用东西的语言。**

　　不要选择C语言作为你第一门 编程语言。也许选择C语言会给你一种过时的满足感。虽然它比早期的汇编语言提供了更好语法描述，但是今天C（或者C++）不是能够快速编写软件的语言。毫 不偏袒的说，我建议你学习一种动态语言。选择一种你觉得语法（文档）让你使用起来更舒服的编程。为此，你可能需要花费一些时间来在几种不同的语言之间进行 比较。这么做目的不是为了让你感觉更好或者让编程更容易。更快地完成自己的作品并且能够看到成果，才是激励你编程的源动力。不要选择一门需要某种重量级的 IDE（IDE是一种能够帮助你编写并运行代码的工具）才能轻松编程的语言。你所需要的仅仅是一个文本编辑器。（注：原文并没有标号，是我们加上的，请大 家别忘记了哦，我们是从零开始数数的。:) ）

**1. 选择一个好的编辑器。**

　　编辑器对程序员而言，就是像是弓对弓箭手一样重要。这里有一些推荐的编辑器：

* SublimeText 2 – 如果你刚开始编程，推荐使用SublimeText 2
* Emacs – 学习曲线陡峭，快捷键复杂。如果你想定制自己的使用习惯，需要学习Emacs Lisp.
* Vim – 不仅操作简洁而且它默认包含在linux的发行版中，所以深受大家欢迎。我开始使用了2年Emacs。由于Emacs的快捷键操作过于复杂，我的双手开始 经常疼痛，这迫使我转向了Vim阵营。了解Vim的快捷键是非常有必要的。当你在远程的服务器上编辑代码时，你唯一不需要安装的编辑器就是Vim。

　　注意！Emacs和Vim可能已经成为成为古董了。但是，它们都具备一些现代编辑器多不具备的功能。

**2. 选择一个你可以从中学到东西的操作系统**

　 　Windows不会教会你任何东西。使用Windows你唯一学会的就是打开一个exe（可执行）文件来安装软件和使用它。可能刚开始的时候你会觉得很 酷，但是长远来看，如果你立志成为一个web开发人员，可能还是要具备基本的linux知识，尤其是当你要发布程序的时候。Linux也允许你根据自己的 要求来定制程序。Macs也很酷，但是我认为你现在经济上还负担不起。

**3. 不要通过复制粘贴来做备份**

　 　一个新手程序员通常会把文件复制粘贴到一个临时目录下面来进行备份。这也许也是他们唯一知道的方法。不要那么做！你需要使用版本控制软件。我强烈推荐使 用Git，它是个目前很流行的版本控制工具，而且使用起来非常简单。对一个Git新手，有一个良好的社区和很多参考资源来学习。（除了Git之外，还有 mercurial, darcs, fossil等等。但是我还是建议从Git开始，我有很多的理由来支持Git，这里就不多费口舌了。）

**4. 知道去哪里寻求帮助**

　 　参加一个和你相关的（你使用编程工具相关的）程序员社区。比如StackOverflow就是程序员的Facebook。那里没有消息状态和回复，取而 代之的是问题和答案。除此之外还要学会使用IRC。IRC是一种老式的聊天室，现在还在被大多数开发人员用来分享技术信息和帮助解决问题。

**5. 培养你的礼仪**

　 　知道什么时候问问题。你遇到的大多数问题都是别人碰到过并且在Internet上能够找到解答的。在IRC或者任何论坛里发问之前，先搜索一下 google（或者blekko）来看看是不是有现成的答案。在IRC上问问题需要有耐心。永远记住，人们只是处于好心来免费帮助你。有时你发问以后需要 几个小时才能有回复。所以，耐心等待。除此之外，记得问问题的时候要礼貌。世界很小，你怎么对待别人，别人就会怎么对待你。

**6. 结交朋友，技术书籍只能教你解决常见的问题（他们说：书是死的）**

　 　当你在捣鼓一些程序或者从程序的作者那里学习，你能学到一些书本上没有的东西。当你散步的时候，请和遇到的朋友打招呼。你肯定不是唯一的程序员。和其他 的程序员一起交朋友一起工作。你会注意到，当一群技术爱好者在一起的时候，不论话题是从什么地方开始，最后总是以技术话题收尾。这是必然的现象。所以你可 以尽情参与。在你的黄金年龄努力编程，我能告诉你的是，我在开始6年前寻找朋友并开始讨论编程，从那时起才学到那些书本和文章不会我的东西。所以我总是 说，我的编程经验有6年，因为只有那个时候开始，我才开始与人交流并开始感觉真正地开始学习编程。

**7. 参加开源项目**

　 　为开源项目编写代码能带来回报。这不仅仅是帮助别人，你留下的代码会被其他人使用或（可能）改进。当别人给你的代码添加功能或者提出修改意见时，也是在 提高你的编程水平。开源软件项目不一定要是一个大工程，编写一个下载youtube视频的小程序也是很有用的。更重要的是，你会惊奇地发现你编写的代码会 成为与他人有效沟通的桥梁。

　　最后，若干年之后如果本文的内容曾经对你有一点点的帮助，作为回报请写一封类似的信给希望开始编程并不知如何入手的朋友，有可能的话请纠正本文中过时的内容。

　　智者说过，掌握某个东西需要10年或10000个小时，也就是汉语中的“十年磨一剑”，所以不用着急。

　　英文原文：[An open letter to those who want to start programming](http://blog.akash.im/an-open-letter-to-those-who-want-to-start)