顶级程序员的成功秘诀

在正式学习Python语言的使用之前，我们先来了解一下顶级程序员的成功秘诀。

是什么呢？是头悬梁锥刺股的钻研精神？还是无限YY的超出边际的想象力？

其实都不是，简单来说只有三点。

1.兴趣

如果对编程开发没兴趣，那么一个一个月开发经验的程序员和20年开发经验的老司机本质上没有任何区别。

至于兴趣如何来，其实很随机。比如玩一款游戏很过瘾，就想开发自己的游戏。比如用某个APP或网站很好玩，就想自己弄一个。又比如cryptokitty这种区块链养猫很有意思，于是自己也想搞一个。比如比特币的发明者中本聪竟然跻身全球富豪之列很羡慕，于是自己也想研发一个新的区块链数字火币。比如觉得跟柯洁对战的AI超级NB，自己也想研究下。。。

有了兴趣，才能迈出第一步。

2.目标

很多人对编程开发感兴趣，但是三天打鱼两天晒网，不到几个月就放弃的大有人在。可能有人觉得是意志力不足的问题，可是这些看似意志力不足的家伙在其它事情上似乎又能坚持的比较久。

个人认为，是缺乏明确的目标和方向。

兴趣毕竟只是一个引子，要想持续的前进，就必须有坚定的目标和方向指引。这样即便中间遇到很多困难和障碍，但是至少知道今后的路在哪儿，而不至于拔剑四顾心茫然。

3.方法

在这三点里面我没有提到天赋，因为所谓的天赋不是自己后天可以改变的。而对于编程开发来说，相比艺术创作和科学研究，天赋并不是最重要的。

但方法其实真的很重要。

对于编程开发，有很多方法和技巧，但是个人认为作为一门应用学科，基于实战项目去学习和探索是最最重要的技巧。

除此之外，如果想要成为一个顶级程序员，至少就目前而言，英语一定要过关。至少能毫无障碍的看懂技术文档，开发讨论社区中的发言和回复，英文的音视频开发教程等等。

此外，学会如何提问和如何搜索也非常重要。

那么，究竟应该如何提问呢？

1.给问题起一个好的标题

具体来说，如果是求助的问题，应该在标题前面加上[Help]这个标签。

标题要尽量清晰易懂，比如下面的这个：

[Help]Answer not being shown on screen when running

2.问题的内容要详细易懂

具体来说，应该包含以下要素：

（1）所观察到的现象

比如对刚才的问题，这里可以写到，The answer is not shown on the screen when running my Python scripts.

（2）所期待的结果

比如对刚才的问题，这里可以写到，The calculated result should show on the screen when running my Python scripts

（3）代码链接

推荐大家将代码片段上传到gist上，<https://gist.github.com>

部分情况下可能需要将整个工程源代码上传到github或云盘作为参考

（4）问题出现的可能原因

这里要描述的是，自己所推测的问题的可能原因。当然，其实这一部分很多时候都没法写，因为提问者多半并不知道问题出在哪里。

（5）为了解决该问题所尝试过的方法和努力

这里要描述的是，为了解决该问题，自己做了哪些尝试，结果是怎样的。

（6）所出现的错误和警告

这里要提供所出现的错误和警告提示。

（7）软件版本号和开发环境配置

根据问题的不同，这里可能要提供软件和编程语言的版本，开发环境的具体配置，甚至包括硬件环境的描述。

（8）问题的截图

部分情况下我们需要提供问题的截图

3.问题解决后要更改标签并表示感谢

如果你需要别人后面仍然愿意帮助你，那么就需要遵从编码界的基本礼仪。

在问题得以解决后，一定要更改标签为[Solved]，同时要对提供解决方案的人表示自己的谢意。

如果是自己解决的，那么也一样要更改标签，同时在问题的描述下面补充上自己的具体解决方案。

除了提问，善于搜索也是这个时代程序员应该具备的基本技能。

应充分利用相关领域的开发者论坛和社区，以及开源项目等。

具体来说，包括以下：  
[github.com](http://github.com)

[stackoverflow.com](http://stackoverflow.com)

[google.com](http://google.com)

[zhihu.com](http://zhihu.com)

但是需要提醒大家的是，在复制粘贴代码横行的时代，不要忘了思考。

貌似是傅盛说过一句话，不要用战术上的勤奋来掩盖战略上的懒惰。

在编程开发领域，不要用疯狂的复制粘贴来完全替代思考。

不然，再过十年，恐怕你就要被AI彻底取代了~