



链滴

我想对所有新手程序员说些肺腑之言

作者: [gentoo666](#)

原文链接: <https://ld246.com/article/1513576239167>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)

欢迎来编程世界。

编程真的很难****

无论你是多么擅长编程，能拖慢你脚步的唯一因素是你学习和解决问题的能力。随着你的编程愈发提，你更能体会我这句话了。编程还有个有趣的地方，永远不会觉得枯燥。如果你觉得枯燥，那估计你什么地方做错了，你应当对在写的代码，做一下抽象，或者优化。

编程同样也会是最让你沮丧的事情之一****

正常的编程经历应该是这样的：尝试解决问题 → 受挫沮丧 → 再努力尝试 → 回退一步 → 灵光乍现 → 最终搞定，程序正常运行。感觉受挫，这实在太正常不过了，不可能完全没有受挫。（随着你的技艺高，你会挑战更难的问题，这过程还是会遇到受挫。）挫折感让提高解决问题带来的满足感。你得适挫折感。

不管挫折感有多大，总会有解决方案的****

随着你的技艺提高，对摆脱困境，你会有更强更大的自信。挫折感越大，推倒你解决更难的问题。（果问题太简单，那就太没趣啦。）

当我们还是小孩时，很多事情都做的很糟糕，只是我们没注意到。****

成年后，把事搞得很糟糕的机会并不多。拥抱这种感受，完全正常。你做的事情，并不是人人都有能去做的。

你还得知道，练习编程的第 1 个小时，和第 10 个小时是截然不同的****

与第 100 个小时更不一样了。Dan Plan 的博主曾写过一篇文章，说他在练习 1 万小时后，他的高尔极大提升。编程也是同样的道理。（伯乐在线补充：谷歌大牛 Peter Norvig 的观点也类似，请看他文章《[自学编程，十年磨一剑](#)》。）

你是一个魔术师！****

编程也是一种魔术。想想看，我们努力学习，从魔法书中汲取知识，然后制作符咒（从计算机书和互联网学习，然后编写程序）。魔术师可以教会魔术师如何施展魔法。我们程序员可以在现实也能做到这，并且还能创造人人都能用的神奇设备，比如：互联网和移动电话。如果没有我们的技能，不可能有这样的设备。

善用互联网****

在你学习编程的第一年中，你遇到的每个问题，很多在你之前的程序员，早就碰到了。并且网上已有录好的答案了。你使用搜索引擎的效率越高，你会更高兴。**警告**：如果你对网上搜到的代码段不理解那就不要随便就用。

不管何时编程，都要随身带着「百宝囊」****

这些工具是我们干活的诀窍。你要学的第一批东西（变量、循环等），就好比工具箱中的螺丝刀和锤。在编程生涯中，这些工具是日常必需品。随着经验积累提升后，你开始学习使用更高级的工具（比：树、A*、valgrind、调试器等），也许这些东西平常用的不多，但在解决特定问题时特别得劲。大

数编程领域（游戏开发、Web开发、数据库）都有各自领域问题及对应的解决方案。因此，你先前解过的问题，以后遇到时，可以更加轻松应对了。

每个程序都是由代码组成****

你可以在自己电脑上几乎阅读、修改、捣鼓所有东西，还有其他人已写好的程序。说到电脑，它基本就是一个超级大国，啥都有。

提高编程能力的最佳方式，就是多写代码****

保持做东西、写代码！作为一个老师，我会尽我所能，把你带入门。一旦你技艺达到一定程度后，假时日，你能解决所有碰到的难题。坚持写代码写下去吧。毕竟李白都说过了：只要功夫深，铁杵磨成！

综合前面 2 点，除了写大量代码，还要读大量代码****

天外有天，人外有人，这世上比咱们聪明的程序员多的是。他们已经写了很多代码了，或许你就在 GitHub 上就能看到。

推荐：《[阅读源码，推荐 10 个 C 语言开源项目](#)》

据说程序员学的第 2 门编程语言是最难的****

你在学习第 1 门编程语言时，塑造了你的编程思维方式（看我的严肃脸.jpg）。当你在学第 2 门语言，你必须重新思考之前被定型的东西。（不能打破改变你思维模式的语言，不值得学习。）当你掌握第 2 门语言后，掌握其他新语言会更得心应手。任何一种必要的语言，我用一个周末基本就可以学的差不多。

没有最好的编程语言！****

在解决某些特定问题时，某些语言和工具的确优于其他的。当你学习了其他不同的语言时，那就不要陈旧的编程思维，移植到新系统上。相反，要习惯用新思维来编程。

多玩，多犯傻踩坑！****

编程好比一个巨型乐高模型，充满了待解决的问题。多花些心思和时间，在这上面添砖加瓦，做些「具」程序。这比分派给你的任务会有趣多了。

工作之外，优秀的程序员也会为自己编程****

如果你真的擅长这一点，你就永远不会发现自己失业了。

遇到问题卡住时，试试在纸上写代码****

我是认真在说，不是开玩笑。这种方式很神奇，并且是编程竞赛中的标准实践。为什么这种方法有效，我认为当不用考虑语法问题后，我们有更多脑力去解决实际问题。

我确信自己时不时地会冒出其他建议。但正如大多数建议，如果是真的很重要的建议，你自己也能想的。

编程愉快 :D