

**译者按：**原文作者乔纳森·丹尼可（Jonathan Danylko）是一位自由职业的web架构师和程序员，编程经验已超过20年，涉足领域有电子商务、生物技术、房地产、医疗、保险和公用事业。正如乔纳森在文中所言，本文适合刚毕业的大学生和刚入门的程序员。如果你已是高级开发人员，或许你能在本文中看到自己的身影。

从11岁时，我就一直在编程，并且一直都很喜欢技术和编程。这些年来，我积累了一些艰难又容易的经验。作为一名程序员，你或许还没这些经验，但我会把它们献给那些想从中学到更多的朋友。

虽然我想我会持续更新这些经验，可能还会有更多的感想，但就我这20年来看，我想下面这个列表中基本不需要增添额外的东西了。下面就是我至今最难忘的经验：

**1. 估算解决问题所需要的时间。**不要怕，承认吧！我曾见过一些程序员为了解决一个特殊问题而坐在显示器前面8小时。为自己定一个时间限制吧，1小时、30分钟或甚至15分钟。如果在这期间你不能解决问题，那就去寻求帮助，或到网上找答案，而不是尝试去做“超级堆码员”。

**2. 编程语言是一种语言，只是一种语言。**随着时光推移，只要你理解了一种语言的原理，你会发现各种语言之间的相似之处。你所选择的语言，你应该觉得“舒服”，并且能够写出有效（而且简洁）的代码。最重要的，让语言去适应项目，反之亦然。

**3. 不要过于注重程序的“设计模式”。**有时候，写一个简单的算法，要比引入某种模式更容易。在多数情况下，程序代码应是简单易懂，甚至清洁工也能看懂。

**4. 经常备份代码。**在我年轻时，我就有过因硬盘故障而丢了大量代码的经历，这经历很恐怖的。只要你一次没有备份，就应当像有着严格的期限，客户明天就需要。此时就该源码/版本控制软件大显身手了。

**5. 承认自己并不是最顶尖的程序员 – 知不足。**我常想，我对编程了解已足够多，但是总有其他人比你优秀。正所谓，“一山总比一山高”。所以，向他们看齐吧！

**6. 学习再学习。**正如第5点所说，我经常会在手里拿一本计算机或编程相关的杂志或书（不信，可以问我的朋友）。诚然，总有很多你不知道的技术，你可以从中学习以保持不落后。如果你有一种灵巧的方式来获取你需要的新技术，那你每天都应该坚持学习。



**7. 永恒的变化。**你对待技术/编程知识，就应像你对待股票一样：多样化。不要在某一特定技术上自我感觉良好。如果那种技术或语言已经没有足够支持，那你还不如现在就开始更新你的简历，并启动培训新计划。我能保持前行的主要原则是什么呢？至少了解两到三种语言，所以，如果某种语言过时了，你在学习新技术的时候还可以依靠另一种语言。

**8. 提携新人。**协助并且培养初级/入门的开发人员学习优秀的编程方法和技巧。也许你还不知道，在帮助他们向更高一层前进时，你自己也在向更高一层提升，你会更加自信。

**9. 简化算法。**代码如恶魔，在你完成编码后，应回头并且优化它。从长远来看，这里或那里一些改进，会让后来的支持人员更加轻松。

**10. 编写文档。**无论是Web服务的API，还是一个简单的类，你尽量编写相应文档。我曾经引以为豪的代码注释，因过度注释而有人指责。给三行代码加一行注释，只需要你几秒时间。如果那是一个比较难以理解的技术，千万别担心过多注释。如果你能很好做好自己的工作，大多数架构师、后备程序员、支持组都会感激你。

**11. 测试、测试再测试。**我是一名黑盒测试粉丝。当你完成编码后，你“被认可”的时候就开始了。如果你们公司有QA部门，如果你的代码中有错误，那你得到的评论，会比项目经理还多。如果你不彻底测试自己的代码，那恐怕你开发的就不只是代码，可能还会声名狼藉。

**12. 庆祝每一次成功。**我见过很多程序员在解决编程技术难题后，会和同伴握手、击掌或甚至手舞足蹈。每个人在生命中都会碰到“顿悟”。如果一个程序员高兴地跑来叫你看他的非凡代码，也许你已经看过这样的代码100遍了，但你也应该为了这个家伙而庆祝第101次。（编者注：《[庆祝成功的九种方式](#)》。）

**13. 经常检查代码。**在公司，你的代码要经常检查（包括自查和其他同事检查）。不要把别人的检查，看成是对代码风格的苛求。应该把它们看作是有建设性的批评。对个人来说，经常检查你的代码并且自问，“我怎样才能写得更好呢？”这会加速你的成长，让你成为一个更优秀的程序员。

**14. 回顾你的代码。**在看到自己以前的代码时，通常会有两种方式：“难以至信，这代码是我写的”和“难以至信，这代码是我写的”。第一种往往是厌恶的语气，并在想如何改进它。你也许会惊叹，旧代码也能复活成为一种更好的程序，甚至是一个完整的产品。第二种通常带着惊奇和成就感。开发人员应该一到两个自己完成的项目成果，能让众人不禁而立并注目而观的项目。同样，基于你优越的编程能力，你可以把过去的程序或项目拿出来，把它们更新为更加优秀的产品或想法。

**15. 幽默是不可缺的。**在我20年的开发生涯中，我还没有碰到哪位程序员是没有幽默感的。实际上，干我们这行，幽默是一项必备品。

**16. 谨防那些无所不知的程序员，不愿分享的程序员，还有经验不足的程序员。**当你遇到这几种程序员时，你自己要谦虚。无所不知的程序员，更想当一个英雄而不是团队成员；保守的程序员则是在编写着他们独享的代码；而经验不足的程序员则会每十分钟就来问你一下，当代码完成后，代码已经是你的，而不是他们。

**17. 任何项目都不会那么简单。**朋友、家人和同事曾请求我仓促做一些事情，仓促做一个程序或者网站。对于这样的事，应该从双方做计划，才能做出令双方都会满意的东西。如果某人起初只需要一个使用 Microsoft Access的、只有有3个页面的网站，但来就很可能变成一个有15个页面的网站，并使用SQL Server，有一个论坛，还有一个定制的CMS（内容管理系统）。

**18. 任何时候不要想当然。**假如你承接一个简单的项目，你可能会认为某个部分可以轻松完成。千万别这样想！除非你有一个类、组件、或者一段已经写好的代码，并且在现有的项目已经测试通过。不要认为这将是很容易的。

**19. 从来没有已经做完的软件。**曾经有一位程序员告诉我，没有软件是已经完成的，它只是“暂时完成了”。这是明智的忠告。如果客户还在使用你写的程序，并经受了时间的考验。如果有机会，你仍在更新它，这并不是什么坏事，这让你不断地前行。

**20. 耐心是一种美德。**当客户、朋友或家庭成员用电脑的时候，他们也许会受挫，进而想砸电脑，或气冲冲地离开。我一直在告诉他们，“是你掌控电脑，不是电脑掌控你。”对于用作编程的电脑，你要有一定的耐心。一旦程序员知道问题所在后，他们就会站在电脑的角度看问题，并且说“哦，这就是为什么它是这样做。”