

程序员练级攻略（2018）：开篇词

2018-05-29 陈皓，杨爽



程序员练级攻略（2018）：开篇词

朗读人：柴巍 11'41" | 5.36M

2011 年，我在 [CoolShell](#) 上发表了《[程序员技术练级攻略](#)》一文，得到了很多人的好评。并且陆续收到了一些人的反馈，说跟着这篇文章找到了不错的工作。几年过去，也收到了一些邮件，希望我把这篇文章更新一下，因为他们觉得有点落伍了。

是的，老实说，抛开这几年技术的更新迭代不说，那篇文章写得也不算特别系统，同时标准也有点低，当时是给一个想要入门的朋友写的。所以，非常有必要从头更新一下《程序员练级攻略》这一主题。

前言导读

升级版的《程序员练级攻略》会比 Coolshell 上的内容更多，也更专业。这篇文章有【入门篇】、【修养篇】、【专业基础篇】、【软件设计篇】、【高手成长篇】五大篇章。它们会帮助你从零开始，一步步地，系统地，完成从陌生到熟悉，到理解掌握，从编码到设计再到架构，从码农到程序员再到工程师再到架构师的进阶，实现从普通到精通到卓越的完美转身.....

在入门篇中，我先推荐的是 Python 和 JavaScript 作为入门语言，并给出了相应的学习资源和方法。Python 语法比较简单，有大量的库和语法糖，是零基础的人学习编程的不二之选。而 JavaScript 是前端语言，更容易让你获得编程的成就感。

随后，我们需要学习使用操作系统 Linux、编程工具 Visual Studio Code 等入门必学内容。Web 互联网作为第三次工业革命信息化浪潮中最大的发明，也是每个程序员都不能错过的。而学习编程还是要多多动手，因此我给出了 Web 编程入门的学习要点，并给出了一个实践项目，帮助你理解和巩固所学的内容。

如果你跟着我的这个教程走过来，并能自己去解决遇到的问题，那么，我相信你能够做一点东西了，而且你还可能会对编程非常感兴趣了。但是你千万不要以为自己已经入门了。我只是用这些内容给你一些成就感，并激发你持续学习兴趣的。

正式入门，我推荐的语言是 Java，因为我认为，它是所有语言里综合实力最强的。随后，推荐了更为专业实用的编程工具，如编程的 IDE、版本管理工具 Git、调试前端程序和数据库设计工具等，并且给出了一个实践项目。我同时设置了业务和技术两方面的需求，跟着做一遍，相信你对学习编程会有更多的理解和感悟。

接下来，我要带你进入更为专业更为复杂的编程世界中。进入之前，我们需要树立正确的三观和心态，这对于程序员是至关重要。这就好像民工建筑队和专业的工程队的区别，就好像小作坊和工厂的差别，他们并不仅仅是差别在技能和技术上，更是差别在做事的心态和三观上。

因此，在学习专业的软件开发知识之前，我们来谈谈程序员修养。它看似与程序员练级关系不大，实际上却能反映出程序员的工程师特质和价值观，决定了这条路你到底能走多远，是精髓所在。有修养的程序员才可能成长为真正的工程师和架构师，而没有修养的程序员只能沦为码农，这是码农和工程师的关键区分点。

在“修养篇”，我给出了一些相关的学习资料，并指出了我认为比较重要的几个方面：英文能力、问问题的能力、写代码的修养、安全防范意识、软件工程和上线规范、编程规范等。这些能力的训练和培养将为后续的学习和发展夯实基础。

此时，相信你已经迫不及待地想进入专业基础篇了。这部分内容主要涵盖编程语言、理论学科和系统知识三方面知识。在编程语言方面，推荐学习 C、C++ 和 Java 这三个工业级的编程语言。理论学科方面，需要学习算法、数据结构、网络模型、计算机原理等内容。系统知识方面会讲述 Unix/Linux、TCP/IP、C10K 挑战等专业的系统知识。最后给出了你可以选择的几大从业方向。

- 如果你对操作系统、文件系统、数据库、网络等比较感兴趣，那么可以考虑从事底层方面的工作。
- 如果对分布式系统架构、微服务、DevOps、Cloud Native 等有热情，那么可以从事架构方面的工作。

- 如果是对大数据、机器学习、人工智能等比较关注，那么数据领域可以成为你一展身手的地方。
- 如果你对用户体验或者交互等更感兴趣，那么前端工程师也是个不错的选择。
- 此外，安全开发、运维开发、嵌入式开发等几大方向中，也为你提供了丰富多彩的发展空间。

以我之见，该如何选择应该完全遵从于你的本心，你更愿意在哪个领域里持续奋斗和学习。这个答案，在你的手中，在你的心中。这里我只想和你说两观点：各种技术方向不是鱼和熊掌，是可以兼得的；很多技术是相通的，关键是你是学在表面还是深入本质。

软件设计能力是每个程序员都需要具备的基本素质。我结合各主流语言讲述了泛型编程、函数式编程、面向对象编程等多种编程范式，分享了 DRY- 避免重复原则、KISS- 简单原则、迪米特法则（又称“最少知识原则”）、面向对象的 S.O.L.I.D 原则等等多个经典的软件设计原则。

同时，给出了软件设计领域的一些重要的学习资料。软件设计是工程师非常重要的能力，这里描述了软件工程自发展以来的各种设计方法，这是从工程师通往架构师的必备技能。

登峰造极，是每个武林高手都渴望达到的境界，对于每个有理想有追求的程序员也是如此。因此，我特意在《程序员练级攻略（2018）》这一系列内容的最后设置了高手成长篇。

相较前面的内容，这部分内容相当全面和丰富，涵盖系统、数据库、分布式架构、微服务、容器化和自动化运维、机器学习、前端方向和技术论文等几方面内容，而且深度一下子拔高了好几个数量级。

同时，这也是我留给你的再一次做选择的机会，平凡还是卓越？自在悠闲，还是猛啃书本，“不破楼兰终不还”？还是遵循你内心的选择吧。偷偷地告诉你，我选的是后者。

你应该不难看出这一系列文章比我在 CoolShell 上的那一篇更为专业，标准也会更高，当然，难度也会更大。但是，也会让你有更坚固的技术基础，并能有更高更广泛的提高。

通过这一系列文章，我主要想回答以下几个问题。

- 理论和现实的差距。你是否觉得自己从学校毕业的时候只做过小玩具一样的程序？走入职场后哪怕没有什么经验也可以把文中提到的这些课外练习走一遍。学校课程总是从理论出发，作业项目都看不出有什么实际作用，到了工作上发现自己什么也不会干。
- 技术能力的瓶颈。你又是是否觉得，在工作当中需要的技术只不过是不断地堆业务功能，完全没有什么技术含量。而你工作一段时间后，自己都感觉得非常地迷茫和彷徨，感觉到达了提高的瓶颈，完全不知道怎么提升了。
- 技术太多学不过来。你是否又觉得，要学的技术多得都不行了，完全不知道怎么学？感觉完全跟不上。有没有什么速成的方法？

对此，我有如下的一些解释，以端正一下你的态度。

- 并不是理论和现实的差距大，而是你还没有找到相关的场景，来感受到那些学院派的知识的强大威力。算法与数据结构、操作系统原理、编译原理、数据库原理、计算机原理.....这些原理上的东西，是你想要成为一个专家必需要学的东西。这就是“工人”和“工程师”的差别，是“建筑工人”和“建筑架构师”的差别。如果你觉得这些理论上的东西无用，那么只能说明，你只不过在从事工人的工作，而不是工程师的工作。
- 技术能力的瓶颈，以及技术太多学不过来，只不过是你要为自己的能力不足或是懒惰找的借口罢了。技术的东西都是死的，这些死的知识只要努力就是可以学会的。只不过聪明的人花得时间少，笨点的人花得时间多点罢了。这其中的时间差距主要是由学习方法的不同，基础知识储备的不同造成的。只要你的方法得当，多花点时间在基础知识上，会让你未来学习应用知识的时间大大缩短。以绝大多数人努力的程度，和为自己不努力找借口的程度为参考，只要你坚持正常的学习就可以超过大多数人了。
- 这里没有学习技术的速成的方法，真正的牛人不是能够培训出来的，一切都是要靠你自己去努力和持续地付出。如果你觉得自己不是一个能坚持的人，也不是一个想努力的人，而是一个想找捷径的人，那么，这篇文章并不适合你。这篇文章中的成长路径是需要思考、精力和相关的经验的，这都需要时间，而且是不短的时间。你先问问自己有没有花十年磨一剑的决心，如果没有，那这篇文章对你没有任何作用。

这里有一篇传世之文《[Teach Yourself Programming in Ten Years](#)》（[中英对照版](#)）。还有在我 Coosllhell 上的这篇《[程序员的荒谬之言还是至理名言？](#)》。

我希望你在学习编程前先读一读这两篇文章。如果你觉得可以坚持的话，那么，我这一系列文章会对你很有帮助。否则，我相信你只要大致浏览一下目录及其中的某些章节，就会选择放弃走这条路的。是的，这个系列的文内容也会让一些想入行但又不愿意付出努力的同学早点放弃。

最后，给出我的几点学习建议。

- 一定要坚持，要保持长时间学习，甚至终生学习的态度。
- 一定要动手，不管例子多么简单，建议至少自己动手敲一遍看看是否理解了里头的细枝末节。
- 一定要学会思考，思考为什么要这样，而不是那样。还要举一反三地思考。
- 不要乱买书，不要乱追新技术新名词，基础的东西经过很长时间积累，会在未来至少 10 年通用。
- 回顾一下历史，看看历史时间线上技术的发展，你才能明白明天会是什么样的。

另外，这篇文章的标准会非常高。希望不会把你吓坏了。《易经》有云：“取法其上，得乎其中，取法其中，得乎其下，取法其下，法不得也”。所以，我这里会给你立个比较高标准，你要

努力达到。相信我，就算是达不到，也会比你一开始期望的要高很多.....

下面是《程序员练级攻略（2018）》系列文章的目录（持续更新中）。

- [开篇词](#)
- 入门篇
 - [零基础启蒙](#)
 - [正式入门](#)
- 修养篇
 - [程序员修养](#)
- 专业基础篇
 - 编程语言
 - 理论学科
 - 系统知识
- 软件设计篇
- 高手成长篇



左耳听风

洞悉技术的本质
享受科技的乐趣

极客时间
重拾极客精神·提升技术认知

陈皓

资深技术专家
骨灰级程序员

扫码订阅

版权归极客邦科技所有，未经许可不得转载

精选留言



RZ_diversity

终于等到耗哥更新练级攻略了，已经摩拳擦掌准备大干一场了！

2018-05-29

👍 30



dancer

👍 18

买了10个专栏了，耗子叔的专栏真的是干货最多、诚意最多的！

2018-05-29

作者回复

真能买

2018-05-29



夏洛克的救赎

👍 6

编程背后的数学原理呢？

2018-05-29



云学

👍 5

耗子叔真是一个宝库呀，不断释放干货

2018-05-29



ADC之父

👍 3

形成自己的知识体系很重要。没有所谓的速成，从基础知识抓起，一步步夯实这才是捷径，别无他法

2018-05-31



小马

👍 2

最好的专栏，没有之一

2018-05-30



极品混混

👍 2

十年磨一剑，进军芝加哥。

2018-05-29



期待

👍 2

向着高级工程师迈进

2018-05-29



Tobian

👍 2

迫不及待

2018-05-29



右耳朵猫咪

👍 1

为什么耗子叔的文笔这么好 我也想这样

2018-06-06

作者回复

你的昵称是故意的么.....

2018-06-07





Hua

👍 1

懒惰找借口是一方面。第二方面是的确人老了记忆力下降头脑不灵光，而技术是在演变的。很多技术你学了但没有在实际工作中用到，哪怕你练了书上的example没多久就忘。现在时代不同，10年前你懂个60%就可以当senior，好公司呆呆几年后就可以当架构师，多数负责整体设计。而如今年代苦活累活都是senior在干，加班也多。人的时间有限，毕竟不是机器，需要休息从而保证第二天做事有效率。现在可惜发展这么快，按你说的把所有都自学掌握我觉得还是有很大难度。

2018-06-05



108度

👍 1

什么十年磨一剑，花多少时间都不在乎，遇到一个名师，愿终身跟随

2018-05-30



网虫子

👍 1

终身学习。

2018-05-29



WTF

👍 1

软硬技能兼顾，系统化体系，由浅入深，同时提供了learn do，自己再演进一下，learn teach给你周边的人，不成高手都难！学贵有恒，除了给耗叔打call，最重要的是自己要能坚持打卡！

2018-05-29



宋桓公

👍 0

感觉值了，你好陈叔

2018-06-10



mengtanx

👍 0

专门为了看程序员练级攻略买的专栏，趁涨价前赶紧买了（没想法真的要涨价了，没见过还能半路涨价的，只能说明专栏太赞干货太多）

2018-06-09



棍子

👍 0

我也陆陆续续买了很多专栏，感觉自己的老毛病又犯了，就像自己喜欢买很多的电子书和纸质的书，看不看先放着，放着放着过时了！还是回头认真看自己买的第一份专栏吧！感觉还是耗子哥干货多！当然这里面讲得更像是武功心法，其他的专栏更像是某一种功夫！我还是慢慢来吧！

2018-06-07



右耳朵猫咪

👍 0

为什么耗子叔的文笔这么好 我也想这样

2018-06-06





大脚一巴掌

站在巨人的肩膀上

2018-06-05

👍 0



厚子

买的第二专栏，看来自己还是挺有眼光的，哈哈，感谢

2018-06-05

👍 0