

讲堂 左耳听风 文章详情

高效学习：源头、原理和知识地图

2018-08-30 陈皓



高效学习：源头、原理和知识地图

朗读人：柴巍 10'44" | 4.92M

有了上一篇文章中分享的那些观点，我们来看看应该怎么做。下面是我觉得比较不错的一些学习的方法，或者说对我来说最有效的学习方法。我相信，只要你和我一样，做到的话，你的学习效率一定能够提升很快。

挑选知识和信息源

还是我在《程序员练级攻略 2018》中说的那样，学习英文是是否能够成长的关键，尤其对于计算机知识来说。如果你觉得用百度搜中文关键词就可以找到自己想要的知识，那么你一定远远落后于这个时代了。如果你用 Google 英文关键词可以找到自己想要的知识，那么你算是能跟得上这个时代。如果你能在社区里跟社区里的大牛交流得到答案，那么你算是领先于这个时代了。

所以，我认为你的信息源要有下面几个特质。

- 应该是第一手资料，不是被别人理解过、消化过的二手资料。尤其对于知识性的东西来说，更是这样。应该是原汁原味的，不应该是被添油加醋的。

- 应该是有佐证、有数据、有引用的，或是有权威人士或大公司生产系统背书资料。应该是在时间和实践检验过的，或是小心求证过的，不是拍脑袋野路子或是道听途说出来的资料。
- 应该是加入了一些自己的经验和思考，可以引发人深思的，是所谓信息的密集很大的文章。

顺便说一句，我发现 Medium 上的文章质量比较高，很多文章都 Google 到了 Medium 上。

我在《程序员练级攻略 2018》后期的文章中罗列了很多文章资源，有的读者很不能理解，他们觉得我多少应该导读一下或是写上一些自己的想法，而不是只是简单地罗列出来。这里请允许我辩解一下，我之所以这样做，并不是因为偷懒，我完全可以把这些信息资料全部隐藏起来，翻译也好，搬运也好，导读也好，自己消化完后再写出来。那么，我可以写出多少个专栏来？

我觉得只要我有时间，极客时间上的所有专栏都不用写了，我一个人就 OK 了。我可以写得又快又好，而且超出所有的人，那我可以挣到很多钱。但我不想这样，我想把我读过的好的文章推荐给大家，就像推荐书一样。那些是信息源头，已经写得非常不错了，我不用再多废话。而且那些文章底部都有很多的引用，你可以一路点过去。

但我想通过这些简单链接的方式，为我的读者打开一个全新的世界，他们可以在这个世界中自己找食吃，而不需要依赖我，这才是我想给大家带来的。我不知道，我的那些推荐文章，有没有让你看到了一个很广阔的世界，在那里，每天都在产生很多最新、最酷、最有营养的一手信息，而不是被我或他人消化过的二手信息。

我问过几个在别的领域知识付费的专栏作者：你们写的这些东西，不就是卡内基的那些东西吗？不就是某某书里的那些东西吗？不就是某某英文资料里的那些内容吗？

他们告诉我，是的，我们就是搬运知识，有些国外的东西，国人理解不了，需要用他们喜欢的语言讲出来，而他们又不读书，英文水平也一般，但他们想速成，所以，才有了我们的市场。这有点像餐馆，他们不想自己做饭吃，那就我们来做，有的人还要别人喂到嘴边，甚至好些人都需要先帮他们嚼一遍，他们才能吃得下去。所以，我们这些学习能力强的人挣点他们的钱也是应该的。

对于上面这些言论，我不想做过多的评论。这里，我只想说，对于一个学习者来说，找到优质的信息源可以让你事半功倍。一方面，就像找到一本很好的武林秘籍一样，而不是被他人翻译过或消化过的，也不会有信息损失甚至有错误信息会让你走火入魔。另一方面，你需要的不只有知识和答案，更重要的是掌握学习的方法和技能。你要的是“渔”，而不是“鱼”。

注重基础和原理

我在很多的场合都提到过，基础知识和原理性的东西是无比重要的。这些基础知识就好像地基一样，只要足够扎实，就要可以盖出很高很高的楼。正所谓“勿在浮沙筑高台”。我说过，很多人并不是学得不够快，而他们的基础真的不行。基础不行，会影响你对事物的理解，甚至会让你不能理解为什么是这样。当你对事物的出现有不理解的东西时，通常来说，是因为你的基础知识没有跟上。

在《程序员练级攻略 2018》一文中，我用了很大的篇幅给出了学习基础技术的路径。只要你努力学习那些基础知识，了解了其中的原理，就会发现这世界上的很多东西是大同的。

举个例子，如果你学习过底层的 Socket 编程，了解多路复用和各种 I/O 模型的话（select, poll, epoll, aio, windows completion port, libevent 等），那么，对于 Node.js、Java NIO、Nginx、C++ 的 ACE 框架等这些中间件或是编程框架，你就会发现，无论表现形式是什么样的，其底层原理都是一个样的。

无论是 JVM 还是 Node，或者是 Python 解释器里干了什么，它都无法逾越底层操作系统 API 对“物理世界”的限制。而当你了解了这个底层物理世界以后，无论那些技术玩成什么花样，它们都无法超出你的掌控（这种感觉是很爽的）。

再举一个例子，当学了足够多的语言，并有了丰富的实践后，你开始对编程语言的各种编程范式或是控制流有了原理上的了解，这时再学一门新语言的话，你会发现自己学得飞快。

就像我 2010 年学习 Go 语言一样，除了那些每个语言都有的 if-else、for/while-loop、function 等东西以外，我重点在看的就是，出错处理是怎么玩的？内存管理是怎么玩的？数据封装和扩展怎么玩的？多态和泛型怎么搞的？运行时识别和反射机制是怎么玩的？并发编程怎样玩？.....

这些都是现代编程语言必需的东西，如果没有，那么这个语言的表达能力就很落后了。所以，当知道编程语言的本质和原理后，你学习一门新的语言是非常非常快的。而且可以直达其高级特性。

最最关键的是，这些基础知识和原理性的东西和技术，都是经历过长时间的考验的，所以，这些基础技术也有很多人类历史上的智慧结晶，会给你很多启示和帮助。比如：TCP 协议的状态机，可以让你明白，如果你要设计一个异步通信协议，状态机是一件多么重要的事，还有 TCP 拥塞控制中的方式，让你知道，设计一个以响应时间来限流的中件间是什么样的。

当学习算法和数据结构到一定程度的时候，你就会知道，算法不仅对于优化程序很重要，而且，会让你知道，该如何设计数据结构和算法来让程序变得更为健壮和优雅。

有时候，学习就像拉弓蓄力一样，学习基础知识感觉很枯燥很不实用，工作上用不到，然而学习这些知识是为了未来可以学得更快。基础打牢，学什么都快，而学得快就会学得多，学得多，就会思考得多，对比得多，结果是学得更快.....这种感觉，对于想速成的人来说，很难体会。

这里我想再次强调一下，请一定要注重基础知识和原理上的学习！

使用知识图

先讲一个故事，2000 年我从昆明到上海，开始沪飘的岁月。刚到上海，找不到好工作，只能大量地学习和看书，C/C++/Java，TCP/IP，Windows 编程，Unix 编程，等等。结果呢，书太多了，根本看不过来。我想要更多地掌握知识，结果我发现以死记硬背的方式根本就是在使蛮力学习，我很难在很短的时间内学习很多的知识。

于是我自己发明了一种叫“联想记忆法”的方法，比如，在学习 C++ 的时候，面对《C++ Primer》这种厚得不行的书，我就使用联想记忆法。

我把 C++ 分成三部分。

- 第一部分是 C++ 是用来解决 C 语言的问题的，那么 C 语言有什么问题呢？指针、宏、错误处理、数据拷贝..... C++ 用什么技术来解决这些问题呢？
- 第二部分是 C++ 的面向对象特性：封装、继承、多态。封装，让我想到了构造函数、析构函数等。构造函数让我想到了初始化列表，想到了默认构造函数，想到了拷贝构造函数，想到了 new..... 多态，让我想到了虚函数，想到了 RTTI，RTTI 让我想到了 dynamic_cast 和 typeid 等。
- 第三部分是 C++ 的泛型编程。我想到了 template，想到了操作符重载，想到了函数对象，想到 STL，想到数据容器，想到了 iterator，想到了通用算法，等等。

于是，我通过“顺藤摸瓜”的方式，从知识树的主干开始做广度或是深度遍历，于是我就得到了一整棵的知识树。这种“顺藤摸瓜”的记忆方式让我记住了很多知识。最重要的是，当出现一些我不知道的知识点时，我就会往这棵知识树上挂，而这样一来，也使得我的学习更为系统和全面。

这种画知识图的方式可以让你从一个技术最重要最主干的地方出发开始遍历所有的技术细节，也就是画地图的方式。如果你不想在知识的海洋中迷路，你需要有一份地图，所以，学习并不是为了要记忆那些知识点，而是为了要找到一个知识的地图，你在这个地图上能通过关键路径找到你想要的答案。

小结

总结一下今天的内容。首先，我强调了，挑选知识和信息源的重要性，因为优质的信息源可以让你事半功倍。其次，我认为，一定要注重基础和原理，基础打牢，学什么都快，而学得快就会学得多，学得多，就会思考得多，对比得多，结果是学得更快。

最后，我指出，学习时一定要使用知识图，学习并不是为了要记忆那些知识点，而是为了要找到一个知识的地图，你在这个地图上能通过关键路径找到你想要的答案。我相信，只要掌握了好的方法，你能做到的话，你的学习效率一定提升很快。

下篇文章中，我将接着介绍几个不错的学习方法。希望对你有帮助。

下面是《高效学习》系列文章的目录。

- [端正学习态度](#)
- [源头、原理和知识地图](#)
- 深度，归纳和坚持实践
- 如何学习和阅读代码

- 面对枯燥和量大的知识



版权归极客邦科技所有，未经许可不得转载

精选留言



slam

0

受益匪浅，感触比较深的是这两三年，为跟上技术的发展，去系统的接触微服务、ML、区块链等，才发现英语不行对查找并获得最有价值的内容的阻碍，经常是筛选的好内容原来只是翻译，而且还只是翻译一部分，另外也发现阻碍学习的不是工作忙时间不够，反而是基础不扎实，学习新系统新框架新语言不得不一次一次去翻查操作系统原理、网络协议、语言特性，尤其痛苦学习ML，完全暴露数学没基础，唯独有安慰的一点也是如耗子哥所说的，面对知识的海洋，要有知识的地图，自己现在也是这么做，把一个大的领域放入思维导图，梳理的知识往图上挂，至少感觉可以沉下来学了

2018-08-30



还有这种操作

0

这几篇给自己一个方向。谢谢老师

2018-08-30



知行合一

0

非常感谢老师的分享。深深的理解到老师提到的基础原理基础知识的重要性。就像上大学时对C指针理解了就对java传递引用变量就有深入的理解；从汇编层面理解了C函数的调用机制就理解了递归机制。我一朋友是搞逆向工程的，他对我说，如果逆向搞得好，其他高级语言，框架也不会差。我想请问下老师，他说的这个道理和老师说的基础知识的重要性本质上是一样的吗？如果从逆向角度去学习基础，老师觉得这个方法可行吗？谢谢老师

2018-08-30



Michael

学习基本原理确实重要~

2018-08-30

0



知行合一

非常感谢老师的分享！深深感受到老师提到的基础知识的重要性，就像上大学时理解了C指针后，对JAVA 传递引用变量就有深入的理解；从汇编层面理解了C函数的调用原理，就理解了递归机制~我一朋友是搞逆向工程的，十来年了，他对我说如果逆向搞得好的，其他高级语言，那些框架也不会差，我想请问老师，他说的这个道理是不是和老师说的基础原理的重要性，本质上是一样的吗？如果从逆向工程这个角度去学习基础知识，老师你觉得可行吗？谢谢老师

2018-08-30

0



dilei

读耗子叔的文章，能读到对读者得尊重，能读到老师对学生的谆谆教诲，在这个复制拷贝的年代，遇到这样的作者，是我们的幸运，谢谢老师。

2018-08-30

0



宁静致远

我16开始转行做的前端，当时主要是被人工智能弄的焦虑不安，几乎是零基础学习编程，只是为了补充我对世界的认识，这么想是不是太狂了。另外工资水平还行，就转了

2018-08-30

0



Gotta

耗叔，Linux系统上有没有好的画图工具推荐，找了挺多都不太满意...

2018-08-30

0



Lu | SSEINFO

知识图是和思维导图一样的吗？还是有其他的具体形式？

2018-08-30

0



TinsonChan

c++的三部分，我第一次看到这样分好像是在《effective c++》这本书，这里是我第二次看到。其次，关于知识图，应该和知识体系是一个概念，只有建立自己的知识体系才能把各个知识点联结成网，更利于自己的理解。耗子叔这里更上一层，提出深度学习，这估计也是为啥我这么努力却只是一个普通程序员的原因，紧跟随组织，深度学习

2018-08-30

0



小薛薛

谢谢，调整好心态，继续上路。

2018-08-30

0

