

关于后续更新

后续所有更新均在微信公众号统一发布：扫描下方二维码关注，不迷路😊



正式开始聊聊自学那些事儿！

这次我要跟大家聊聊自学那些事

在前言部分我也说过了，我认为的自学应该是一个人的主观意识，也就是这个学习过程是自己主动进行的，而不是被动进行的，可能有很多人都是想自学的，但是自学一段时间就会发现，自学真的是太不容易了，所以，有些人坚持不了多久就放弃了，被自学中的各种各样的问题给击溃了！

在跟别人聊到自学的时候，我经常收到的一个反馈就是，感觉自己学习很慢，没啥效率，很多东西学了跟没学一个样子，我想这个问题绝对不是个例，而是共性问题，每一个自学的人都会被这个问题给困扰，很多人也正是因为这个问题儿坚持不到最后，怎么出现这样的问题，你有没有考虑过为什么呢？

我在自学的时候也遇到过这些问题，而且把我折磨的不轻，所以我就想弄明白为什么会出现这样的问题，在多方试错总结之后，我得出了这么几条原因

1. 学习方法不对
2. 有效时间不多
3. 没有正向反馈

这个我在之前也提到过，方法在自学当中是至关重要的，方法对了，你会异常轻松，方法不对，那你就脑壳疼去吧！所以在自学过程中，这个方法是你一定要格外注意的！

我们这里抛开方法先不说，就自学编程这件事，靠谱吗？自学编程真的可以吗？会不会以后搞得个混吃等死啊？一个人刚开始自学编程的话也许会有这些疑惑，你要我说自学编程真的可以吗？我还真不敢回答，这个真的要看个人努力程度了，当然还要加上正确的方法，否则花费大力气却也得不到自己想要的想过。虽然我不能肯定的回答你自学编程可不可以，但是我可以告诉你的是，就目前的大环境而言或者整体的发展趋势，互联网行业是最容易逆袭的一个行业，它给了很多平凡人一个证明自己的机会，就看你自己是否认真努力了，

我还是想强调下这句话“互联网行业是最容易逆袭的一个行业，它给了很多平凡人一个证明自己的机会，就看你自己是否认真努力了”，觉得这句话很有思考价值。

所以说，自学编程绝对是一个有发展潜力的事情！

就拿我现在接触到的一些行业大佬来说吧，其实他们大部分都是自学编程走过来的，为什么会这样，我们一般觉得这些大佬都非常有主见有思想，这其实跟自学编程有很大关系，因为在自学的过程中，你会经常一个人思考总结很多问题，久而久之，思考能力提高了，解决问题的能力也上去了，所以很多大佬都是喜欢自学的！

可是自学又真的不是那么简单的一回事，就拿这些大佬来说，他们在当初自学的时候也会遇到很多的问题，其实也就是我们自学者遇到的共性问题！那接下来我们就来好好说说我们在自学的时候到底会遇到哪些共性问题呢？

首先第一个就是没人教没人问的尴尬境地了，其实我个人觉得这是自学路上最大的一个问题，举一个很简单的例子，我们平常遇到的编程问题，我们很可能想破头皮也解决不了，通常情况下既影响自己的自学进度，还很打击自己的自信心，而且还不知道该怎么办，总觉得这个坎迈不过去接下来的就无法进行。

如果这个时候能有个人问问可能你的问题很快就解决了，或者会给你提供一些新的思路方法，这都非常有助于你的自学，但是嘞，实际情况是大部分人在自学的过程中很难遇到这样的人，为啥，因为这样的人不是随随便便就能成为的，这要求他有一定的资历，而且在自学这块有过自己的感悟，最重要的一点是愿意跟你分享，愿意教导你，而这样的人你真的不容易遇到，即使有，限于你的圈子太小，你也接触不到，这个就很难受了，你就只能一个人硬着头皮往前闯了，这也是大多数人的一种状态。

在自学过程中有个好的导师，可以让你缩短一半的自学时间，达到事半功倍的效果，可是这样的机遇实在是太难得了。这其实是也是我想把自己的经验总结写成小册子的原因，因为在很大程度上，这本小册能成为你自学路上的小导师！

当然在自学的过程中绝对不仅仅会遇到这一个问题，还有一个问题就是你会时常感觉到自己的学习效率很低，也就是说你会经常出现的一个问题就是，学了东西很快就会忘记了，然后经过一段时间，感觉自己啥也没学到，之前学习的东西忘的差不多了，这其实对自学的我们打击很大的，我都那么努力去学了，结果没过多久忘的差不多了，难道自己真的有那么笨吗？

这其实是方法的问题，你只是一味的去学习，却不懂的如何把知识内化，所以才会导致这样的状况！

在自学的时候我们还常常会遇到的一个问题就是突然学着学着不知道该学什么了，就是突然变得迷茫了，不知道还要学习什么，没有方向了，这其实是一个自学疲劳期，很多人都会遇到，这个其实也是因为自己的学习没有方法，基本上是想到哪学到哪，在学之前就没有一个完整的路线，以至于想到的学完了就不知道该学什么了，处于一个知识的空档期，通常情况下，如果不及时解决这个问题，也会拖慢你的自学进度，降低你的自学效率！

还有一个问题大概就是自学的人往往会觉得要学的东西实在是太多了，有种有生之年学不完的节奏（其实还真学不完），感觉要学习的知识不仅多而且还杂，想想都脑壳疼的赶脚。

除此之外你可能还遇到不知道该看什么书，不知道哪个视频教程适合现在的你来看，如何做项目，学到什么程度可以找工作等等这些问题，其实我所说的这些只不过是自学者会遇到的一些共性问题，要是具体到你们个人，指不定你们还会遇到什么稀奇古怪的问题呢？

总之就是，自学不简单，你会遇到各种各样的问题，没有好的方法，问题得不到有效的解决，自学进度变慢，效率变得低下，久而久之，看不到希望，很多人也就放弃了。

如果你在自学的过程中没有一个好的可切实执行的方法作为支撑，那自学真的是一个痛苦的过程，相反，如果你能有一个不错的学习方法，不仅学着愉快，而且效率贼高，进度贼快，但是，这个方法不是那么容易习得的，而且靠你自己来总结归纳，你可能要花费大量的时间，而且还得不到有效的方法！

其实市面上目前不乏很多讲解自学编程的方法，但是我总觉得这些方法都很虚，什么意思呢？可能你会有这样的感受，就是看完别人说的这个方法，突然觉得，说的好有道理啊，简直太对了，自己怎么不早一点知道呢？可是嘞，真的让你按照别人说的这个方法去实际操作的话你就懵了，好像你并不知道该怎么去做，也就是说这些方法说的都很有道理，但是可操作性太低了！

所以在总结自己的自学编程方法体系的时候，我所遵循的理念就是不仅仅停留在理论，一定是要切实可行的！我想这点就是我与他人不一样的地方！

明天开始更新具体自学步骤第一步！敬请期待！

自学编程从搭建知识框架开始！

接下来我会尽最大的篇幅把我的方法论给大家讲清楚，每一个步骤会告诉你为什么这么做以及到底该如何做，也就是说我不仅告诉你这个道理，更会给你提供切实可行的方案，可以让你马上行动起来！

首先自学编程的第一步，经过我长时间的自学经验总结，最好的方式就是首先从搭建知识框架开始，这个知识框架就好比盖一座房子，它就相当于房屋的整个蓝图，能让你整体看到是个什么样子的，自己在心里也有个数，知道自己以后该往哪地方发力，如何去填补，这一点在初期自学编程的时候很重要！

往往我们刚开始自学编程，在没有人给予指导的情况下，我们的方向是模糊的，也就是自己不知道该怎么学，也不知道学习哪些东西，更不知道如何去获取相关的这些知识，如果你在初期想弄明白这些问题的话，你就要花费一些心思了，可以说是费时费力可能也达不到你的预期效果！

而我现在可以告诉你的是，刚开始自学编程，接触一个新的领域，比如初开始学习Java或者python，先从搭建知识框架开始，这将是你后续学习的一个蓝图，是你后续学习这门语言的基础，是个底子，这个相当重要，记住了：**自学编程第一步，先从搭建知识框架开始！**

可能听到这里你就会觉得这跟网上说的一些讲方法的不差不多吗？搭建知识框架这个方法很多人都在提啊？这就有点虚啊，可是嘞，我要告诉你的可不仅仅是这些，上面说的只是让你明白自学编程第一步的一个大前提那就是搭建知识框架，接下来还会更加详细的进行叙述！

首先为什么建议大家自学编程第一步从搭建知识框架开始呢？

我们在刚开始自学编程的时候，如果没有注重搭建知识框架的话，学到后期的话你就会遇到一个问题，就是会感觉到自己之前学习的知识很乱，不成系统，再加上越往后自己学习的知识会越来越多，而之前的学习又是零散的自己没有整理的，你就会感觉自己所掌握的这些知识是散乱的。

就好像很多点，线可能已经连接成了，但是就是组不成一个面，这样的话到了后期非常不利于你对之前旧知识的巩固和新知识的学习，就好比如果我们在一个抽屉里放东西，如果都是随意放，那物品与物品之间就会存在大量的空隙，这样就会浪费掉很多空间，相反，如果我们把物品整理整齐放在抽屉就会比随意放散乱的存放更多的物品。

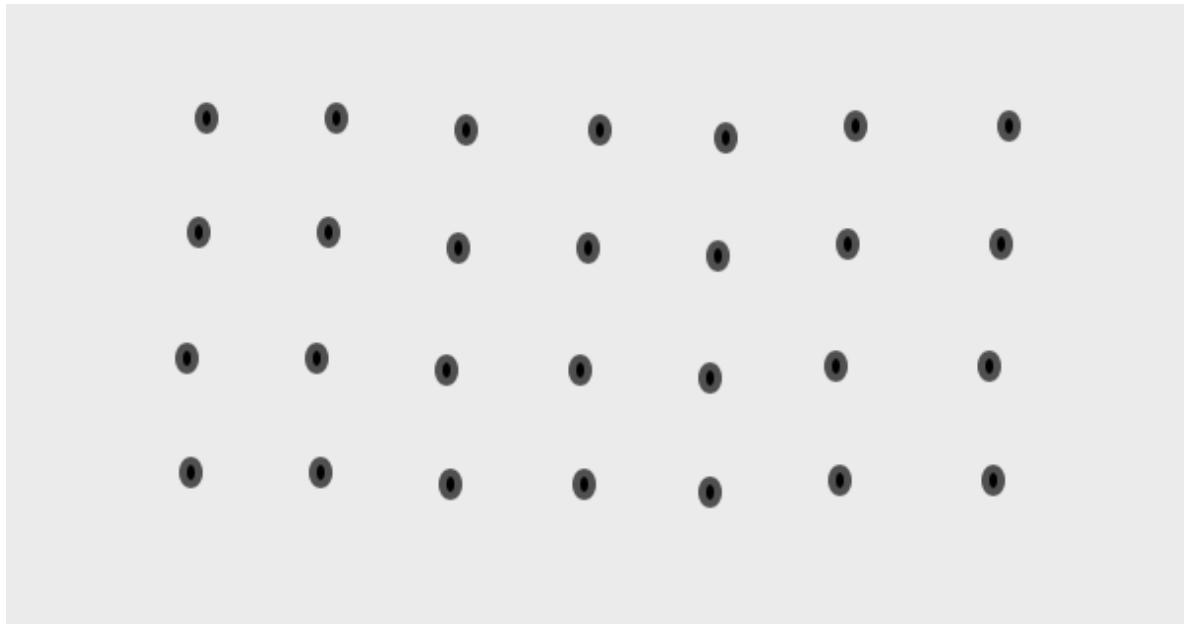
也就是**空间利用最大化了**，这样就会显得紧凑充实，放在知识这块的话，就会感到知识的系统化给你带来的充实感！

假如我们把所学习到的每一个知识都比做一个点的话，那么如果刚开始自学没有搭建知识框架的话，那就是这个样子的



也就是说你的**知识存储是散乱的**，越到后期由于所需知识量会逐渐增加，你就会感到这些知识点的混乱，这非常不利于你对旧知识的巩固和新知识的学习，而且由于你的知识点的散乱，不成体系，你在运用这些知识的时候，也不能随心所欲的进行调用。

但是如果我们在刚开始自学编程的时候就注意知识框架的搭建的话那就是这个样子的



其实自学编程就是一个点到线，线到面的过程，也就是说无论你学习什么，学到什么程度，都是在一个组成面的过程，就看大家这个面谁的更大，谁的更结实了，可想而知，，只有当每一个点都整齐排列才会更加形成线和面，这样面才会足够大足够结实，这其实就是搭建知识框架的重要性！

知识框架就像一座房屋的整体架构，大致模型如此，然后按照这个模型去添砖加瓦，至于最后出来的效果如何，就看你如何添砖加瓦了，以后如何装修了，但是无论怎样，这个整体架构必须有的，它是一个前提。

所以在刚开始自学编程的时候一定要先搭建知识框架！

以上我说明了为啥要在自学编程的第一步先从搭建知识框架开始，这个可以说是理论，那么具体的到底该如何搭建这个知识框架呢？真的让你去实打实的操作了，你该怎么去搭建这个知识框架呢？

这里我说两种方式，都是我认为非常好的方式，看你们个人喜好！

第一种：看视频教程来搭建知识框架

这种方式我觉得更加适合那些完全零基础的小白使用，当然，这是个通用的方式，这里我说一点，我发现在自学的朋友经常会犯这样的一个错误，就是对视频教程的欲望很强导致的**松鼠病**，什么意思呢？就是很多朋友到处找视频教程，然后各种囤积，顶多是这个看看，那个看看，感觉这个也好那个也好，于是这个也想看，那个也想看，导致的最终结果就是，视频教程倒是收集了不少，可是没有一个真正看完过，你说，你是不是就是这样的？

其实这就是自学中的一大禁忌，它会严重的欺骗你，让你觉得自己学的很多，实则不仅学的很少，而且还拖慢了你的自学进度！我告诉你啊，关于视频教程，找一套差不多的就行，就这市面上流传的一些培训班的视频教程，其实都大差不差，只要不是年久失修的那种都行，要知道这就是培训班的啊，所以用来搭建知识框架很合适，找一套就行了，多了没任何好处，现在听我的，看看你收藏的资料里面是不是有好几套同类型的视频教程，听我的，留一套，其他的都删除吧！

记住，只留一套用来搭建知识框架，只留一套，一套，就一套（如果你没有的话，找我，我给你）。

然后你就靠这套视频教程来搭建知识框架，那么有了这么一套视频教程之后，又该如何去做呢？

这个你要知道，一般这些视频教程都是按照知识点的从前到后，从浅到深的按阶段讲解的，所以你完全可以按照这个视频教程里讲解的顺序去搭建这个知识框架，在搭建知识框架这块，我之前听到别人说可以使用思维导图来搭建，我也试过这种方式，把各个知识点都整理到思维导图上。

可是呢？就是感觉不爽，总是感觉哪里不舒服，所以索性放弃这种方式，后来尝试使用笔记软件，以文件夹的形式来搭建，这简直叫一个爽啊，这里推荐有道云笔记软件，因为我们需要它的多层级目录，然后你就按照视频里面的讲解顺序去建立相应的文件夹，然后边学边记笔记。

这里要注意的是，这个阶段，你就认认真真的把这套视频教程看完，其他的啥也别管，把这套视频教程看完最重要，记得要记笔记等你看完了，你就会发现，你建立的这些文件夹还有你的笔记就是你的知识框架了，看着这些，你会有满满的自豪感以及清爽无比的感觉！而且对于这块的学习你也会清楚很多！

至此，看视频教程搭建知识框架的方式就是这样的了！

下面说说第二种方式：搜索相关知识或路径，自我组织知识框架

这种方式其实要求就比较高了，这要求你要有一定的搜索能力和组织能力，这个方式的核心就是自己去搜索所需要学习的知识点，然后整理组合，然后组织成一个知识框架。

这样说你可能不理解，我举一个例子吧，比如说我们学习Java，如果是按照看视频的方式搭建知识框架，那我们只需要找一套视频教程，然后从前往后看，根据所讲建立相应的文件夹就可以形成我们的知识框架。

但是这种方式不是这样的，因为没有了视频教程作为参考，我们就需要自己去搜索，比如去搜索“自学Java应该掌握哪些知识点？”，“Java需要学习什么”，“Java的学习路径是什么”，然后你就会得到很多的答案，就会有很多人告诉你先学什么，再学什么等，然后多找几个答案，你就会得出一个学习路线，然后按照这个路线建立相应的文件夹，之后再按照文件夹的顺序去逐个学习，可以找单独的视频教程看，也可以上网搜系列博客来学习，这个就自由的多了。

至此，这就是第二种搭建知识框架的方式了！

然后我必须强调几点：

1. 这个搭建知识框架可不是说你整理好要学习的知识顺序，然后建立相应的文件夹就行了，这所包含的知识你是要真的去学的！
2. 学习的时候你一定要记笔记，一定要记笔记，这个很重要
3. 最终你的知识框架应该是一个完善的层级文件夹，每个文件夹下有相应知识的学习笔记

这样你的知识框架才算真正的搭建完成！

恭喜你，可以进入第二个大步骤了！

自学编程第二步上阵杀敌--以项目为驱动

这里的上阵杀敌其实就是开始实战，也就是开始做项目。

你要知道，你以后进公司工作就是做项目的，所以实战是你必不可少的，可以说是你最常做的事情，可能有人会说，刚经历了第一步骤，刚把知识框架搭建完毕就要开始实战了吗？虽然说把知识大概都学了一遍，也记了相应的笔记，但是感觉很多知识理解的还是太浅，用的也不熟练啊，这样的情况下，做项目真的行吗？

你还别说，这真的行？我之前在自学的时候就犯了这样的错误，就是觉得刚刚把知识框架建立完备，虽然学习了大部分的知识点，但是很多知识点还是不熟练掌握的不好，做项目开始实战的话效果肯定不好，还是把一些重要的知识点，在项目中用到的知识点再单独的进行学习，之后再开始做项目吧！

后来才发现，这样的想法其实就是一个大坑，你要知道，**永远不要等状态，最好的状态永远不会到来，最好的方式就是马上开干！**

很多人都说，在公司实习一年顶得上你自学三年，这句话不是没道理的，当然我也很想知道为啥会是这样，所以为了探究这个问题，体验我们平常自学和在公司到底有什么区别，我特意找个时间去公司体验了一把，后来让我明白的就是，在公司完全就是项目驱动式的学习（当然不是全部）。

什么意思呢？也就是说，在公司，会给你分配任务，让你完成某某功能，而大部分情况下是这个功能所需要的知识点可能你会，但是你不熟练，想要完成这个功能还是不够的，你必须还要深入的去学习，因此，在短时间内，在压力和任务的推动下，你需要去继续深入学习这个知识并且把功能做好，这样一来，在短时间内你就深化了这部分知识的学习，项目功能也完成了，这就是后续不断的学习。

而我们平常的自学呢？虽然说也在深入学习某块知识，但是由于我们**人脑天生就是喜欢趋易避难**，也就是说**我们总喜欢做那些简单的事情去不愿意费脑子去思考那些比较难的问题**，因此，我们平常的自学因为没有足够的压力和任务，我们往往会陷入一个舒适区，经常去学那些简单的东西，稍微遇到点难得就感觉学不下去，然后下意识的开始逃避，开始去找一些简单的来做，这样看似自己也在不停的学习，但是实际上自己进步不大，而且还浪费了不少时间。

所以以项目驱动的学习是你必不可少的，因此，当你搭建完成知识框架之后，最好的学习就是开始实战了，通过实战是对你提升最好的方式，

当然，我知道你刚刚搭建完毕知识框架，虽然学了很多，肯定有很多不理解不会的，但是我想说的是，通过项目会让你对之前所学知识有个更深的理解之前不明白的说不定通过做项目也就明白了，毕竟有些知识你学了不用，也就不能更好的去理解它！

说到这里，我就不得不说一个自学中最大的一个拦路虎了，那就是**学了之后很容易忘记**，我相信会有很多自学的小伙伴被这个问题困扰到脑壳疼，学的知识倒是不少，但是经常性的没过多久就忘记了，为什么会出现这样的情况呢？大致有如下几条原因吧

1. 知识没有用起来
2. 没有经常复习
3. 只有输入没有输出

你放心，在这个系列教程中我提供的自学编程方法体系都已经把上述问题给解决了，首先这个知识没有用起来的问题，我们就通过实战做项目的方式把它解决了，要知道，我们学习那么多的知识就是为了在以后的项目中去用，也就是说这些知识是要应用到项目中的，一个知识不能够完全脱离项目而存在，与项目结合才能发挥这个知识的最大作用，所以，开始做项目吧！

那么，这一阶段，我们该如何去做项目呢？这里分享我的一些经验。

首先就是项目来源，这个我觉得主要就是来自视频教程，当然还可以买一些实战类的书籍或者网上找一些开源项目（个人觉得这个有点难度），我个人是比较倾向于看实战类的项目视频教程的，这里推荐下幕课实战，还是很不错的，另外还有其他的一些实战类视频教程都可以拿来学习。

那么该如何进行项目实战的学习呢？

这个要从两个阶段开始，第一个阶段就是模仿，第二个阶段就是修改！

我了解过很多过来人的项目学习经验，大部分都是从模仿开始的，模仿之后再慢慢的试着去修改，按照自己想要的效果去修改，这其中要强调的一点就是，我了解到有很大一部分人第一个阶段就完成不了，很多人在开始项目学习的时候，跟着视频都完成不了，因为学着学着发现自己太多不会的了，很多代码照着敲了也不知道啥意思，所以很多都是学着学着就不想学了，觉得没啥意思，自己不会的太多了。

这里我想说的是，这是很正常的现象，毕竟项目是一个整合了很多知识的东西，碰到自己不知道的再正常不过了，但是你可以想想，正是因为如此，学习项目才会有更大的提高啊，所以尽管初开始学习项目很难，但是无论如何请坚持下来。跟着视频教程把代码敲下来，然后多看几遍，多敲几遍，自然而然的你就懂了，这之后你就可以按照自己的想法去修改这个项目了。

以上就是做项目的一些方法，不过千万别以为做完一个项目就完事了，你要知道，做项目这件事是个长久的事情，除非你不干这行了，所以，一定要经常性的做项目，做各种有关的项目，这样你才能不断的提高，不至于被淘汰！

分水岭：知识的深度拓展

这个行业，大家刚开始的时候，差距可能还不大，但是越往后你就会发现，大家的差距开始逐渐增大，个人觉得三年是一个不小的分水岭，三年之后你会发现，有些人的脚步可能是你再也追不上的节奏，所以，头三年相当重要，那这个跟什么有关呢？（当然，影响你变牛X的因素很多）

第一个就是跟你的**项目经验**有关，也就是做的项目这块，你做的项目多了，自然而然编码能力，业务分析能力什么的就逐渐提高了，这就是你的工作经验，所以在平常多做项目还是对你有很大好处的，那么第二个关键点是什么呢？

那就是**知识的深度**了，大家刚开始自学的时候，可能学到的知识都是一样的，刚开始大家的理解也都差不多，可能有些知识点我不理解，其实你也不怎么理解，这个时候大家对一些知识点的掌握程度都差不多，但是越往后你就会越发现，对于同样一个知识点，你们却有着不同的理解，你们掌握的程度也大不相同，可能你还是停留在表面，而别人已经深入底层了，这样的话，平常的工作中倒是没有什么差别。但凡遇到一些技术瓶颈之类的，可能别人能解决的你就解决不了了，这个时候差距就出现了。

这就是知识深度的问题，那么如何解决这个知识深度的问题呢？

到这里你还记得我们之前讲的两个步骤是什么吗？

第一个大步骤：先从搭建知识框架开始

第二个大步骤：不停的做项目

那你想一下，通过以上这两个步骤之后，你会得到些什么？其实你能得到这么一种能力，那就是你会逐渐对一些重要的知识点异常清晰，也就是说经历了第一和第二阶段的学习之后，你会了解到，哪些是重点知识，哪些是高频率问题，这点很重要，为什么？这不就是你要拓展深度的那些知识点吗？

所以你看，自学编程有了方法之后，很多问题不再是问题，很多学习就会变得顺利成章了，当你了解了哪些是重点知识之后，下一步就要对这些知识点进行深度的拓展学习了，那么具体该怎么做呢？

这其实也有个具体的方法，在说这个具体的方法之前，我需要跟你们说说两个概念，那就是**碎片化和系统化**，这个相信大家都会感到陌生，大家可以明显感觉到的是现在我们的碎片化时间越来越多了，在此之前大家一直都在讨论什么碎片化学习，其实我个人觉得，**碎片化的时间真的不适合用来学习，起码不适合用来学习新的知识。**

碎片化的学习带来的都是零散的知识点，这些知识点大部分情况下都只会在我门脑子中存储不了多长时间，所以大部分时间我们碎片化的学习其实都是无效的，在我看来系统化的学习才是真正的有效学习，那么碎片化的时间该怎么办呢？其实**碎片化也可以进行系统化的学习！**

这是怎么回事呢？碎片化进行系统化学习，这不是自相矛盾吗？你还别说，这真的不矛盾，接下来我要告诉你的就是**如何利用碎片化进行系统化的学习来拓展知识深度！**

还记得我们之前说的学了就忘的三个原因吗？

1. 知识没有用起来
2. 没有经常复习
3. 只有输入没有输出

这次我们要解决的就是“只有输入没有输出”的问题，那该怎么做呢？那就是进行系统化的输出，我们之前因为第一步骤和第二步骤让我们知道了哪些是重点知识，这个时候我们就可以进行有目标的系统化输出了，举个例子，比如经过知识框架搭建和实战之后我发现在Java中jvm是个重点知识，那接下来我就要对jvm进行系统化输出了，在知识框架搭建阶段我们肯定也学习过jvm的知识，但是可以肯定的是，当时学习的是比较表面的知识，仅靠这些表面的知识我们是无法进行系统化的输出的。

因此我们还需要继续进行深入的学习，然后多方查找资料，学习相关深度讲解的视频教程，进行总结归纳试错，这就是一个输入的过程，而这个输入的过程我们完全可以放在碎片化的时间去完成，然后再将所学所悟总结成文，归纳整理，发布到博客上！其实你可以感受到，**这不仅仅是一个系统化的输出，更是一个系统化的输入**，只不过我们的输入是在碎片化时间完成的，可能整体下来花了十天时间将jvm大部分知识点进行详细讲解，发布成文，这不就是一个系统化的输出吗？

而这，不正是对知识进行了深度的拓展吗？

对于知识的深度拓展有两个关键点那就是**知识的输入和输出**，这里面分别对两点单独说说，首先是这个知识的输入，这块我建议你可以这么做，可以找一些深度讲解这个知识点的视频教程来看，网上搜一下深度讲解的博客文章，或者买本专门讲解这个知识点的书籍来学习，而输出这块其实就是写成文章分享出来，这里可以选择的平台有GitHub，csdn和微信公众号，当然，绝不仅限于这几个平台！

关于知识深度的拓展就先到这了！

“面试”是绝佳的学习方式！

日常面试？什么鬼？这跟你听到的大多数的自学方法貌似有很大的不同，甚至有点另类，面试这个东西还能安排进方法之中吗？其实，这是一个辅助高效自学编程的好方法，而且是会给你带来长久受益的一个方法！

之前有听别人说到这样的一个方法，就是应该学哪些知识才能更好的找工作，其中有这么个方法我觉得很有效，就比如你以后要找一个Java开发岗位，那么你就去各大招聘网站上去搜索Java相关的岗位，然后仔细研究他们给出的任职要求，这个任职要求一般就是会告诉你，你要胜任这个岗位的话你要掌握哪些技术，然后你要多找一些公司，看看他们对Java岗位的任职要求，看的多了，对比对比，你就很容易发现，你要是想以后找到一个不错的Java岗位，你应该注重哪些技术知识点的学习。

我觉得这个方式挺好，即是从根本上去解决问题，你需要什么，我则是把中心放在这个上面，可以说这就是**单点发力**，好钢用在刀刃上，**把注意力集中在最重要的事情上**，就会达到事半功倍的效果！

那么我们在自学编程的时候完全可以根据这样的方式提炼一个辅助高效自学的方式，那就是每天进行一场面试！

看到这里可能有人不理解了，每天进行一场面试，这个有点不现实吧，你别误会，我这里所说的每天进行一场面试不是让你真的去公司去面试，这样肯定是不现实的，我的意思是说你应该分担一部分注意力在“面试”这个事情上，你完全没有必要去实实在在的公司去面试，你的操作方式有很多，每天做一套面试题或者每天网上搜索看一篇别人的面试总结和技巧等，这就是我所说的“每天进行一场面试”

为什么要这样做呢？

说到这个，你就要知道，以后找工作的前提是啥，那一定是面试，无论去哪一家公司上班，肯定会有面试这个环节（当然，也不排除例外），而且你不得不承认的一个事实是，会做项目的人不一定会面试，如果是做笔试的话总会考到你的知识盲点，面试可以说是一个独立的过程了，所以大部分人在找工作之前，为了能够找到一份满意的工作，都会花一些固定的时间去准备面试，刷题，看一些面试技巧等等，反正就是做很多可以增加面试通过率的事情！

有的人提到面试的时候会很紧张，为什么会紧张，因为你的心里没底，生怕出错，也就是**你的准备不充足**，如果你本身对于一些知识的掌握就不好，而且还没有提前准备面试的话，那你就更怕了，一般这样面试的通过率就会很低，不过还有这样的情况，那就是有些人即使跟你一样本身对于一些知识的掌握也不好，但是人家做了充足的面试准备，那很大几率人家的面试效果就会比你好很多。这足以说明面试的作用有多大！

既然这样，我们为何不在自学之初就把面试这个重要的环节加上去呢？想一下，经过长时间的自学，**面试对你来说就是很平常的一件事**，对于一些面试常考问题，你也了然于胸，到时候真的去面试的时候你不会很紧张，可以说胸有成竹，这难道不好吗？

而且每一天都进行一场面试的好处还不仅仅如此，它还可以拓展你的知识广度，从而完善知识框架，方便你进行知识深度的拓展，而且它还会解决你学了就忘的问题，还记得学了就忘的几个原因吗？

1. 知识没有用起来
2. 没有及时复习
3. 只有输入没有输出

想一下，我们通过“每天进行一场面试”可以解决哪些问题，这就要考虑到面试本身解决了哪些问题了，面试一般就是考察你对一些常用知识点的掌握情况（这里的面试单纯指技术面试）。

所以如果你经常进行模拟面试，那么你就会把一些常用知识给用起来，而且这样也起到了对知识的及时复习，这个可以有效的帮助你减少学了就忘的情况，之前也说了，这也可以帮助你完善你的知识框架，拓展的你的知识广度从而方便你进行知识的深度拓展，这个又是怎么说呢？

因为知识是无穷无尽的，你不可能掌握的面面俱到，什么都会，这也不现实，而面试你永远不知道会遇到哪些知识盲点，因此非常有助于你查漏补缺，完善你的知识层面，因此，每天进行一场面试绝对是你日常自学不可缺少的一个环节！

那具体该如何进行每天的面试呢？首先你要知道，这里说的面试不是让你真的去公司去面试（当然能去更好，可以作为一个检验），而是通常情况下的模拟面试，这里主要就是以做面试题，网上搜索相关面试总结去看，另外还要注意些面试其他细节，意思也就是你要把面试作为你自学过程中的平常事，那这里简单归纳下，你应该如何在自学中进行面试，你应该注重以下几个方面

1. 面试题，包括笔试题（网上有资料）和别人总结的面试问题（这个就是面试中面试官问的问题了）
2. 简历的书写和面试技巧

你每天的面试就应该以面试题为主，简历的书写和面试技巧为辅了，所以这里我们经常做的应该就是每天做套面试题了，这些面试题网上能够找到好多资料，这里还要注意的就是在每天做面试题的过程中你会有如下两大明显感觉

1. 遇到一些从未了解或者很模糊的知识，记住把它完善到你的知识框架中
2. 经常遇到相同的问题，记住这就是重点知识，要对他进行知识的深度拓展

以上就是关于“日常面试”的方法步骤了！

方法好与不好，只有试过了才知道！

知识学了就忘怎么办？

对于温故知新这一块，我是想用它来着重强调我们在自学中遇到的一个大问题，那就是“**学了就忘**”，我相信这是每个自学者都会遇到的一个大问题，而且很多人还不知道如何解决，或许说有很多人没有思考过这个问题产生的原因！

当然当你看完了这篇文章，我想你以后再也不需要为这个问题困扰了，我将给出产生这个问题的原因以及应对的方法，你只需要照着做就可以了。

首先我给出出现这种情况的原因，然后再逐一进行说明：

1. 知识没有用起来
2. 没有及时复习
3. 只有输入没有输出

其实产生“学了就忘”的情况就是因为这三个原因，解决掉这三个问题可以有效帮助你减少“学了就忘”的情况，这三个原因可不是我随口说说而已，而是经过我的**不断试错验证总结**得出的结论，接下来我们就这三个问题进行探讨一下。

首先就是第一个：知识没有用起来

在如今这个**碎片化的时代**，各种信息真的是太多了，我们在碎片化时间去获取这些信息，通常情况下都是在做无用功，只不过打发掉碎片化时间而已，因为这样的一个环境会导致大家对知识的学习也存在一个碎片化，什么意思呢？

也就是大家虽然学习了很多知识，但是却**没有把知识给内化掉**，你虽然学了，但是这个知识并不是真正的属于你的，你只是非常浅的去认识了这个知识，并不知道它有什么用，也就是虽然学了，但是没有把知识真正的去用起来，只有当你把知识真的用起来了，你才会对它有更深的理解，你才会更加懂得它的价值，这样你才会记忆的更加深刻！

所以，学到的知识没有用起来，这是导致你“学了就忘”的一个重要原因之一！

第二个原因就是：没有及时复习

我一直觉得**复习是有效抵抗遗忘的方法**，记不住那就多看几遍，不是有很多人都在说吗？记不住是因为你看的少，看的多了你就记住了，这其实并不无道理，所以及时复习对于知识的记忆非常有帮助。

其实自学的时间久了你就会发现，**学习是最简单，难得是记忆和运用**，你学到了多少新知识并不重要，关键这些知识你是否能够记得，是否能够在需要的时候调用他们，这个才是最重要的，很多人都是学习了新知识，学完然后就放到一边去了，根本没有什么复习可言，时间久了，相当于白学，当初所花费的时间也是属于无效时间，这就是无用功了！

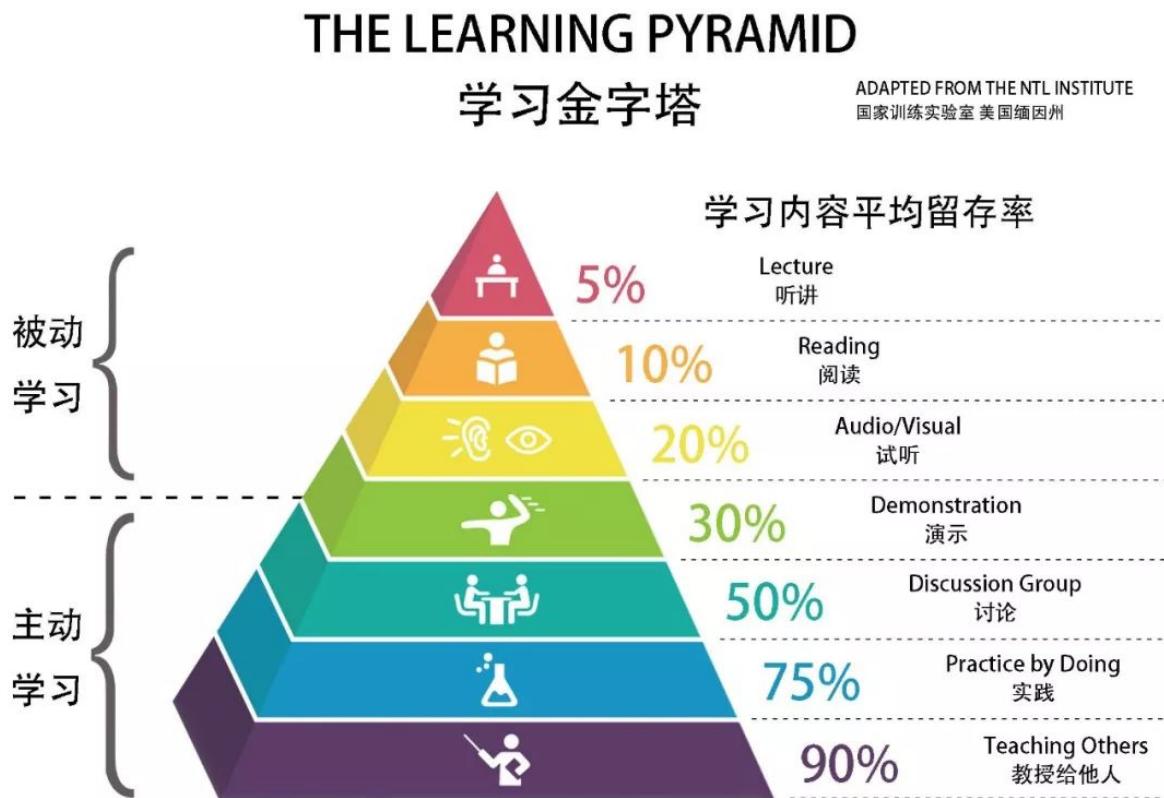
所以对于知识的学习其实是非常有必要进行及时的复习的，不过这有个前提，那就是你得有可复习的东西，那是啥呢？其实就是笔记，我是一个方法论患者外加笔记重度使用者，**好记性不如烂笔头**说的还是很对的，所以我强烈建议每个自学的人都要有记笔记的好习惯，这也是我在自学第一大步骤中说的搭建知识框架一定要

记笔记的原因，只有这样，你才会有可复习的东西啊，时间长了，这些笔记就是你的财富啊！

所以一定要记笔记，然后进行知识的及时复习，这个可以有效帮助你抵抗遗忘！

最后的一个原因其实也就是：只有输入没有输出了

我们首先来看一个学习模型，就是“学习金字塔”模型



在说这个学习金字塔之前我们要知道这么一个问题，那就是我们说的学了就忘，实质上就是指得学习内容的留存率太低，这样我们就能很容易看明白这个学习金字塔要表达的意思了，我们直接看金字塔底端，也就是学习内容平均留存率最高的方式就是教授给他人，那这个具体该怎么做呢？

你可以写博客，做教程等等把知识以自己的理解分享给他人，这会使得你对知识有更深的理解，记忆的自然会更加深刻，有了这个过程其实也就完成了一个闭环的学习模式。也就是你的学习不仅有输入，同时也有输出了，形成了一个单点上的学习闭环，你之前之所以会学了就忘，有很大一部分原因就是因为你只有知识的输入却没有输出，所以对于知识的学习，不仅要有输入还要有输出，要注意学习闭环的形成。

其实看到这里你就会发现，我们这里所提倡的跟在之前说的步骤三中的拓展知识深度已经关联上了，拓展知识深度的最终表现其实也就是输出，所以从现在开始不仅要做知识的输入，更要注重知识的输出，因此，可以试着开始写博客！

系统化的学习才是有效的学习！

到了这一步，其实整体的自学编程方法体系就已经完成了，这里只不过我想强调几个比较重要的点！

首先是关于自学大家都会遇到的两个大问题：

第一：就是总是感觉自我学习效率不高

出现这样的情况，原因有如下几点

1. 方法不对
2. 有效时间不多
3. 没有正向反馈

其实这三个原因中最主要的就是第一个方法不对的问题，也就是没有办法，所以才会导致一系列的问题，就比如有效时间不多，因为你的方法不对，所以很多时候你做的都是无用功，自然而然就会导致你的有效时间不多了，而没有正向反馈这块可能你有点不理解，我们人呐，总是喜欢听好听的，总是喜欢尝点甜头，所以对于学习其实也是一样的。

自学那么久，看不到什么效果，学了那么多，过不了多久就忘的差不多，这样很打击你的信心，时间长了，会感觉没有一点成就感，也就会觉得没有意思，可以说你得到的都是负面反馈了，正向反馈基本上没有，那什么是正面反馈呢？比如某人因为你的一篇文章解决了他很长的疑惑，然后特意评论感谢你，甚至联系你感谢你，你做的很多事得到认可等等，这些都是正面反馈，这样才会给你提供长久自学的动力，这是一种成就感，会推动你不断前行。

可是你想一下，如果没有一个好的方法做支撑，你要想得到正向反馈也不是那么容易的！

第二：就是学了就忘

这个是很多自学者的共性问题，它有种让人头秃的赶脚，产生它的原因无外乎这么几个

1. 知识没有用起来
2. 没有及时复习
3. 只有输入没有输出

关于这个问题我在步骤五中已经详细的介绍并且给出了解决方案！

其实到了现在，你只要是跟着我的这个教程顺序读下来你就会发现，无论是“自学效率不高”还是“学了就忘”的问题，我给出的自学方法步骤中都已经解决了这些问题，对于学了就忘的问题对应的方法有实战，拓展知识深度以及日常面试，这都很好的解决了学了就忘的问题，而这些组合起来，按照我的步骤，那就是一个完整的可执行的自学编程方法，自然就帮你解决了自学效率不高的问题。

在这里我还要跟大家分享一个重要的理念，那就是系统化的学习才是最有效的学习，一定要记住“系统化”，只要你掌握了系统化的学习，那碎片化时间你也可以进行系统化的学习（前面的文章有提到过），为了让你更好的理解，我举个简单的例子

假如说你要学习一大块内容，比如学习Java这门编程语言，那你就要在很长的一段时间按照我的这个方法步骤去学习Java，期间不要再去看什么python或者PHP之类的。

假如说你要学习一小块内容，比如学习jvm（这其实可以归结为知识的深度拓展），那你就要在一段时间内除了每日的本职工作，其他时间都要放在jvm的学习上，比如查资料等等，然后完成一个从输入再到输出的系统化学习闭环，而这个期间就不要再去学习其他的知识点！

这其实就是我比较提倡的“系统化学习”！

掌握这些，自学效率嗖嗖的！

这次给大家分享一些自学中的小技巧，配合之前的自学编程方法体系来自学，效率会大大增加！

1、关于记笔记以及如何复习笔记

我是强烈建议大家记笔记的，俗话说得好“好记性不如烂笔头”，记笔记的话我推荐使用电子笔记，这样方便同步存储，也方便自己的查看，关于电子笔记其实有很多软件可以选择，我这里推荐使用有道云笔记，主要是它的多层级目录是我们搭建知识框架所必须的，也方便知识的整理，大的知识点和小的知识点都可以建立相应的文件夹进行单点发力学习。

我之前曾强调过一定要对知识进行及时的复习，这个很重要，关于对知识的复习也是有方法可循的，我这里分享下我的方法！

比如准备复习jvm这块的知识，你之前肯定学习过，可能也深度拓展过，因此你的笔记软件中会有相对较为完善的jvm知识点，这个知识现不要去直接看笔记复习巩固知识点，正确的做法是，找一张A4纸，拿一支笔，然后去回想，是尽力回想，在自己的大脑中去搜索关于jvm的知识点都有哪些，把它写下来，然后针对每个知识点去回忆，去搜索，比如你写下的知识点有jvm的垃圾回收，这个时候你就要去想，什么是jvm的垃圾回收以及如何进行垃圾回收，这其中用过哪些算法，然后去回想，去写写画画，然后再去看自己记得笔记，哪些地方说错了，哪些知识点遗漏了，然后可以进行google不断完善，这样的复习效果很好，你们可以试试！

2、如何打造学习闭环

我个人非常提倡系统化的学习，其中在系统化的学习中很重要的一点就是要把握一个学习闭环的形成，其实也就是不仅要有输入还要有输出，根据学习金字塔，这样的方式，我们对知识的留存率才会尽可能最大化，而具体怎么做其实也很简单，可以写博客做文章输出，当然也可以进行教程的创作，这都很不错，关键是把握住，不仅要有输入还要有输出，这样也有利于形成正面反馈，对自学帮助是很大的！

不过这个个人一定要坚持，不要写个几天就不写了，要知道，难在坚持，贵在坚持，成在坚持，败在坚持！

3、关于学习资源的获取

我分享的这套学习方法，其实需要用到很多的学习资源，比如视频教程，好的博客，好的面试精选题库等等，尤其是视频教程这块，是一个最主要的学习资料，也是非常适合自学者们，但是很多人会苦于资料的获取渠道，其实很多公众号都会做资源的分享，以前我也做，现在不做了，太累，说实话这些资源很多，你多多少少都能获取到，但是我个人觉得太乱太杂，需要花费时间整理。

这里提供一个获取资源的好去处，那就是某鱼，当然你也可以联系我，我给你提供针对性的视频教程，会免去你很多麻烦，节省你很多时间！

4、自学者必备的一个思想

如果你选择了自学，我希望你有这样的思想改变，那就是把“不懂就问”变成“不懂就查”，自学的人到最后都会形成一个比那些不自学的人强的一项能力，那就是自主解决问题的能力，因此，在自学的过程中，你要学会慢慢的去加强锻炼自己的自主解决问题的能力，这将是你以后的一个巨大的优势，很多自学者都会存在的一个毛病就是，遇到问题，自己根本就没有试着自己先去解决，自己就没有上网搜索过解决办法就去问别人，这种习惯非常不好，一定要改掉！

遇到问题首先要自己解决，而碰到那些实在解决不了的问题有两种解决方案

1. 忽略它，继续进行后面的学习，之所以解决不了，是因为你还没有达到那个层次，这就好比你上幼儿园的时候你会疑惑为什么一加一等于二，而现

在我想你就不会有这样的疑惑！

- 不是上面那种情况，这个问题解决不了无法进行后续工作，自己花费一个小时左右还解决不了，果断求助别人！

5、最后一点

道理千千万，践行者寥寥无几，即使明白了再多的道理，可是依然过不好自己的一生，方法虽好，如果你不去执行的话，那也是没有任何价值，要知道，纸上得来终觉浅，绝知此事要躬行，有的时候我们就是喜欢瞎想，想个啥呢？你倒是行动起来啊！

当你找到了适合自己的努力方法，你就如同万千牛人一般，走在了光彩夺目和掌声阵阵的道路上！

成功无捷径，学习有方法！

生死看淡，不服就干！

学习方法分享 | 从JVM说起，聊聊Java的学习和复习！

我们在学习的时候，多多少少都会遇到这么一个问题：

“学了就忘”

之前也聊过这个问题，在我看来，这个问题是个共性问题，尤其初学者，更是不知所措，倍受打击，那么，为啥会学了就忘呢？

我根据自己的学习经验，大致总结以下三点原因：

1. 知识没有用起来
2. 没有及时复习
3. 没有反馈输出

当然，导致我们学了就忘的原因不止如此，而且根据每个人的学习状况不同，可能远远不止如此，但是正如这个问题是个共性问题一样，我们在这里只说普遍性原因，可能很多人会因为出现上面三个原因导致学习效果不理想。

在此之前，你可能听过别人跟你说，学习编程要多实践，不能只是看书或者听视频，要实际行动起来，动动手，敲敲代码，很多初学者也都知道，你看明白或者听明白跟你能上机自己敲出来完全是两码事，到了后期可能又会有跟你说，多多实战，其实这些都是再告诉你：

“学了知识，如果不及时用起来，时间久了就会忘得差不多了，想要记得牢固，关键要把知识用起来”

说到这里，可能有人会说了，这个我知道，比如说像框架类的知识，就拿Spring来说吧，这个框架很重要，我学了之后知道要实战一下才会学的更好，事实上我也正是这么做的，不过像一些其他的知识好像就不这么回事，比如说Java虚拟机吧，这个属于Java高阶知识吧，初学者比较难啃，而且这类知识点找工作面试是常客啊，这个真的是学了忘，忘了学，我学的时候也想能不能找个实战项目练习下，这样学的更加牢固，可是这些知识初学感觉很多都是概念性的东西，找不到实际应用啊。

确实有这方面的问题，那么像这样的知识该怎么学呢？学了总忘怎么办呢？

我也有这样的困惑，但是通过我自己这样的做法，我发现实际情况改善了很多，什么方法呢？

先说下背景，我学过JVM，但是时间长了，自己感觉跟没学一样，忘得差不多了，相信很多人都有这样的感受，那该怎么做呢？

首先，我会新建一个空白思维导图，然后就开始想，使劲的去回想关于Java虚拟机的知识，因为之前学过这方面的知识，虽然猛地一想，感觉自己忘得差不多了，但是冷静下来，细细回想还是能想起来很多的关键词的，于是，我开始了思维导图的创建。

就这样一步一步的回想，首先从最简单的，遇到Java虚拟机，那就是首先要搞明白什么是Java虚拟机了，好，新建一个节点，写上“什么是JVM？”，有了这个问题之后，我就试着去回答这个问题，在去回答这个我问题的过程中，我突然又想到了Java的跨平台的原理，紧接着又想到当时在解决这块时，自己对编译和解释比较困惑，这些都是知识点，统统新建节点记下来。

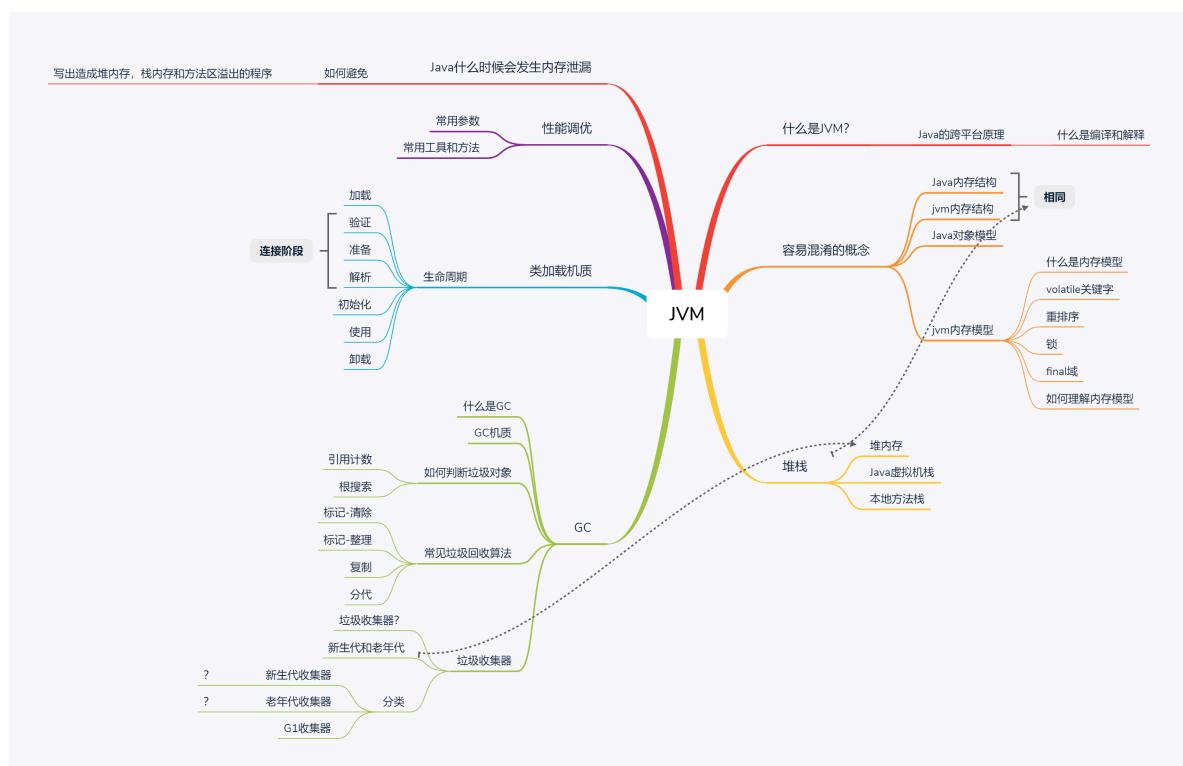
然后我继续回想，记得之前自己对这几个概念很模糊：

- 1、Java内存结构
- 2、Java对象模型
- 3、jvm内存结构
- 4、jvm内存模型

于是又记下新的节点，对这几个自己比较有印象的是jvm内存模型，当时花了很多力气学习，经过自己的不断努力，自己又想到了关于内存模型的一些知识点，比如说volatile关键字，重排序和锁之类的。

就顺着这个思路，一个知识点引起另外一个知识点，发现自己能够回想起来的越来越多了，比如后续的堆栈问题，还有很重要的垃圾回收机制，也就是GC，GC又会牵涉出算法和收集器这些知识，然后还有类加载的问题，对，还有虚拟机的性能调优等等。

就这样，经过自己的不断回想，慢慢的，一张思维导图就出现了



看到这个导图，我自己都惊讶了，本来觉得自己忘得差不多了，没想到还能想起来那么多，当然，这些只是关于Java虚拟机知识的冰山一角，不过，这些对我学习来说就足够了，看着这个图，我再试着去回想每一个知识点或者问题的答案，发现有些自己还能说出来个一二三四，但是有些就比较模糊了，于是乎，对于这块的知识点，自己就发现了属于自己的盲点或者难点。

人呐，最怕的就是不知道自己哪里出了问题，因为不知道所以即使想努力也不知道朝哪个方向使劲啊，既然现在知道自己在JVM这块的薄弱之处，那就花时间去弥补，去学习吧。

当然，自己这样回想出来的肯定是缺胳膊少腿的，不过我们可以找一些专业书籍来对照，或者网上搜搜相关面试题可别人的总结，把那些是jvm的重点但是咱们没想起来的添加上去，然后再重点去学习这些之前没有想起来的。

这既是查漏补缺，也是强化记忆！

当你完成了上面的学习之后你就可以歇歇了，等过个几天，有空了，再新建一个思维导图，继续来，你要相信，你回想的思维导图会越来越好的。

这样的方法好吗？实际上现在很大一部分人的做法是觉得自己忘了，就拿出自己之前的笔记，翻翻看看，这就是很多人的复习，也算是学习吧！

倒也不是说这样不好，只是效果偏差，在《知识大迁移》和《学习之道》中都有提过，不要过多的在书上标记内容，因为很多人都是标记了重点，然后回过头来看这些标记复习，经研究发现，这样的效果并不好，在《学习之道》中花了很大的篇幅去解释一个概念“**提取练习**”，它的核心就是推荐我们去回想内容，这样的学习效率是比较高的，感兴趣的可以去看看这两本书。

以上就是我现在对Java的学习和复习采用的一个方法，灵感大多来源《学习之道》，其实不单单是Java虚拟机这类的知识，我发现很多知识的学习和复习都可以这样来干，如果你仔细想想，你会发现，它很好的间接的解决了以下的问题：

- 1、知识没有及时复习
- 2、没有反馈输出
- 3、知识没有用起来

可能有些人不明白了，这怎么解决“没有反馈输出”这个问题了，当然啦，当你了解到自己的盲点和难点，你肯定会长时间去学习，然后会自己思考，会记笔记，说不定还会写一些博客分享出去，这不就是你的输出嘛。

那有些人又不明白了，“知识没有用起来”呢？

大家可以看我回想的导图左上角有一个关于Java内存泄漏的，有让写出会造成堆内存，栈内存和方法区溢出的程序，这个不就可以敲起来了嘛，然后就会牵涉到性能调优，像类加载这块，也可以敲起来啊，这不都是“用”？

那么，你再想，这不就是解决了我们“学了就忘”这个大问题嘛？

最后再说一句，其实有的时候你会发现，学习不难，你只是缺少一个好的方法和你的实际行动！

好了，就到这了，个人的一点小经验，希望对你有用！转发分享都是真爱！