

用户故事-我是如何学习这个专栏的？

你好，我是戴铭。

专栏上线后，有些同学觉得学起来太吃力，也有些同学觉得学起来很顺利，那么这中间是因为学习方法的问题吗？有道是，方法比努力更重要，只有找到适合自己的学习方法，才能达到事半功倍的效果。

因此，我今天特意邀请了4位同学来分享他们学习专栏的方法，这不仅是要给各位同学提供一个分享自己故事的平台，更是希望这些同学们分享的学习方法，能够让其他同学受益，从而帮助你可以更高效地学习专栏内容。

小万叔叔

你好，我是小万叔叔，目前在某鱼带领 iOS 架构组团队做一些工程优化方面的工作。我希望通过自己不断地学习、积累，可以把握团队的技术方向，进而把架构组发展成一流的技术团队。

其实，我在很早之前就知道戴铭老师，也看过他写的很多文章。这些文章，给我的总体感觉是有深度，有独特的见解。因此，当我在微信里面看到有分享订阅的时候，就第一时间订阅了。

说到专栏文章的学习时机，我一般是在文章更新后，当天早晨的上班路上去听。因为我对大前端技术非常感兴趣，所以听音频的时候自然也就比较专注，可以比较轻松地吸收文章中好的建议和思路。

说到学习专栏文章的方法，我的学习方式可能和其他同学不太一样。大部分同学应该会跟踪一个知识点，学习完专栏文章然后落地到实践，就类似看了专栏文章，然后做课后学习的方式。而**我的学习方法是，会看很多文章，然后把这些知识点汇聚到一起，落地到一个项目中。也就是说，我更希望的是我所学的知识能构成一个体系。**

比如说，包大小优化就是一个明显的案例，我在做包大小优化时用到的方案中，汇聚了头条团队的一些技术分享，图片格式的存储算法、识别算法，源码格式的二进制内容，CocoaPods 打包的方式等等，这些所有的知识点才能构成整个项目的推进树。

目前来看，这个专栏课程中提到的很多技术，我们都在实践，并且在持续优化，希望能做得更好。比如，在专栏的[第4篇文章](#)中，戴铭老师谈到了架构如何设计得更合理，也是我们目前在做全量组件化痛并快乐的一个过程，从最开始组件之间的严重耦合，到划分好相关的层级，业务组件按一定粒度进行划分，然后通过路由中间件进行解耦调用，都和我们全量组件化一路走来过程极其类似。也许，这就是大型业务从堆积到细分，以适应未来业务变迁必须要经历的过程。

针对已经更新的三十多篇文章，我的感触是从架构、性能监控、测试，到代码规范，涉及到了iOS开发的方方面面。与我而言，很多内容都没有作者看得深，所以在工程实践的过程中，我也在不断地扩充自己的知识面，去更深入地理解背后原理。也正因为如此，这个专栏确实让我受益匪浅。

如果说让我感触最深的文章的话，就是讲[App包瘦身](#)的那篇文章。我所在公司的业务体量庞大，而且由于前期的业务堆积没有处理好App包的问题，导致App安装包的大小一度高达270MB。经过我们的努力，现在的包大小在170MB左右，而且仍在继续瘦身中。

总结我们的实践经验，再匹配这篇文章的内容，我们主要是从两个方面做出了努力：

1. 图片优化方面，包括图片格式、Assert 分包、无用图片等方面的优化，为我们缩减了近 50% 的包体积。
2. 在二进制代码瘦身这块，我们做的更多的是，静态库和动态库的平衡、无用代码的删除精简、多套功能同化组件的合并统一。专栏文章里面提到的运行时检查类的方法，是我以前没有想到的。所以，我也会在接下来的二进制减包工作中，继续落地实践这个思路。

对于还在犹豫是否要购买《iOS开发高手课》的同学，我的建议是：不管是否是初学者，里面涵盖的信息量和思考问题的方式，都值得你去购买。

谭鹏

你好，我是谭鹏，是河南省郑州市的一名iOS开发者。从2011年底的第一份工作开始到现在，我已经工作快8年了。

在订阅《iOS开发高手课》专栏之前，其实我已经通过工作实践、阅读相关技术资料，积累了不少iOS开发知识。或许和我不擅于总结有关，这些知识都是零散的，没能构成属于我自己的知识体系。

比如说，我了解iOS的底层原理，对RunLoop、Runtime、KVC、KVO、多线程、内存管理这些都很熟悉了，但也仅限于记忆，并没有进行深入挖掘进而形成自己的输出，更没能用这些知识储备去优化App性能。

因为我清楚自己的短板在哪里，在看到这个专栏的目录以及开篇词之后，发现这个专栏包含的知识非常系统，加之我目前的工作重心就是性能优化、架构，所以我就毫不犹豫的订阅了。等真正开始学习的时候，我发现专栏文章，大多是实战经验的总结，无论从知识的广度还是深度来看，内容都相当不错。

在我看来，学习专栏文章的过程，是和工作相结合的过程，也是扩展自己知识边界的过程。比如，我自己做原生开发的时候使用过Promise，但是从来没想过要使用Promise的思想去解决一些问题。学完专栏的[第25篇文章](#)后，我就马上在项目中引入了PromiseKit，解决了多个网络请求串行的问题。

在学习完专栏文章之后，可以快速在工作中应用实践，实践的结果又会反过来扩充自己的知识体系，并增加学习的成就感、激情。这是一个正向反馈，可以让我不断地提高自己。

使用极客时间的这段时间，我的最大感触就是，**高效学习，需要先搭建知识的金字塔结构。有了完善的知识结构，我们才能看到完整的知识地图，学习起来才能做到有的放矢。**

所以，当我们在选择学习资料的时候，还需要结合自己的实际情况，以尽量完善自己的知识结构为目标。比如说，iOS开发者要想提升自己，想要成为一名高手的话，就不能只盯着具体的语言（比如，Objective-C）和业务逻辑，还要在iOS底层原理、跨平台开发、网络协议、数据结构和算法等方面多下功夫。

因为，只有具备了完整的知识结构，我们才能走得更远。

大太阳

你好，我是温超，是一个北漂三年有余的iOS开发工程师。你也可以叫我大太阳，因为我喜欢阳光，也很乐观，所以就用了这样一个昵称。现在，我在一家做教育的互联网创业公司工作。在这之前，我还做过棋牌游戏，所以现在的工作也算是跨界啦。

其实，我以前也听说过极客时间这个平台，但是真正接触还是从戴铭老师的《iOS开发高手课》开始。让我订阅这个专栏的原因，在于课程目录，以及开篇词中的“我们程序员都是手艺人”这个观点，这些内容真的

是都说到我的心坎里了。

百度百科是这么解释“手艺人”的：以手工技能或其他技艺为业的人，包括陶工、瓦工、铁匠、织工、木匠、厨子等等，也包括米开朗基罗等文化艺术巨匠。在我看来，能创造、会精心打磨作品，正是广大开发工程师的标签。

千里之行始于足下，订阅课程是我们迈出学习之路的第一步，那接下来的路要怎么走呢？在此，我就和你分享一下我的学习方法吧。

在我看来，**当选择一种资料去学习一门知识时，最重要的是要弄清楚自己要收获什么**。这也很好理解，只有当我们知道自己要什么以后，才能有足够的动力去学习。可能同学们接触iOS开发的程度不同，对自己需要学习的iOS知识的认知也不同。针对我自己的情况来说，在我还没开始学习之前，我就清楚自己的弱点，以及想要理解、学习的知识点。比如，我的iOS基础薄弱，对Runtime、RunLoop的原理掌握得不够透彻，也没有过大型项目经验等等。

现在，这个专栏每更新出一篇文章后，我都会第一时间去收听音频，先对这篇文章涉及了哪些知识点、这些知识点的难度如何、我需要花多少时间去消化等问题，有个大概的了解。接下来，我会再去看对应的文稿，根据自己的情况进行针对性的学习，查找些相关的技术资料，并且一定要动手去做。正如戴铭老师在开篇词中所说，对咱们手艺人来说，不动手都是空谈，只看不练是不会有进步的。

对有些同学来说，专栏更新的速度可能有些快，前面的知识还没吃透，这时候要怎么办呢？我的建议是，不要着急，踏踏实实地学习，能学明白一个就是一个，学到就是赚到。

但是，当我们遇到一个难啃的知识点时，尤其是你已经在这上面花了很长时间了，那我的建议是不要一直钻牛角尖，你可以先去学其他的知识，等你觉得自己又思路了再回过头来继续钻研的效率会更高，这也是一种曲线救国的思想吧。

因为，很多知识都是相通的，就和链表一样，有的知识点是单向链表，有的是双向链表，而有的是循环链表等，你可能在学习其中的某一环，当你了解到它的上一个节点的知识点之后，你就能更快地学会这个知识点啦。

之所以这么说，是因为曾经在一个项目开发的初期，我因为语音播放的问题钻了牛角尖，导致项目工期延后了三天。其实，在我第一天卡在这个问题上的时候，技术老大就劝我说可以先写其他部分代码，然后再回头来解决这个问题。但当时我坚持己见，一直到把这个问题解决完才继续写其他部分的代码。等到项目上线后，我才反应过来，当时就应该先把这个问题放一放，等有思路了再去完成这部分代码，就不会有延误工期的事情了。

上面这些内容，就是我学习这个专栏的方法和情况了。接下来，我想再和你聊聊戴老师的《iOS开发高手课》。

说实话，大家在订阅一门课程的时候，看到的只有简介、目录和开篇词，很难去判断值不值得我们为此付费。那么，关于这个专栏，我可以肯定的和你说，它值得。

就目前已经更新的三十多篇文章来说，其中涉及的知识点，需要我下很大功夫才能掌握。毕竟，这些内容与我来说都是干货，而且涉及的这些知识点，是我们在iOS开发过程中真真切切会遇到的，需要也值得我花时间去思考、去学习、去实践。

对我来说，从我做iOS开发到现在，一直没有在一个像样的、大规模的公司工作过，而且基本上都是独立开发，遇到问题基本上都是自己解决。这个过程，就要求我必须不断地去了解、去学习相关的知识。

不是说通过这种方式学习的知识不深刻，而是会很零散，没有系统的知识脉络。在这之前，我也通过其他平台，尝试了很多iOS开发相关的课程、学习资料，但效果一般，没有沟通和交流，感觉我就是在孤立、被动地学习。

相比之下，在极客时间这个平台上，同学们可以在评论区各抒己见，分享自己遇到的问题，以及对应的解决方法，作者还会帮我们解答那些我们自己解决不了的问题，这让我感觉收获非常大。我可以看到自己的问题，也可以通过其他同学的分享去避免自己犯类似的错误。所以，每篇文章后面的留言我都会认真看，除了收获知识外，我也更加坚定自己还有很大的进步空间。

在我看来，这才是一个正常的学习交流模式。

除此之外，我还想对广大的对iOS开发有兴趣的同学们说几句，只有你去做，才会知道自己的能力强，自己的不足在哪里，自己有需要提高些什么。学习是一个持续的过程，知识是不断积累的，如果你想深入了解iOS开发，你想成为一名优秀的iOS开发者，或者是想要多学习与iOS相关的课程的话，那就和我们一起学习这个专栏吧。

冷秋

你好，我是冷秋，从事 iOS 开发有3年了，现在瑞幸咖啡厦门总部负责 iOS 平台架构工作。

作为开发者，业务大多是重复性工作，你并不会因为多两年经验而有多么明显的竞争力。你要想比别人优秀，就必须跳出业务圈子，学习业务之外的知识，这样才能获得更高、更快的成长。所以，我每天都会抽出两到三个小时的时间，去学习，去丰富自己的知识体系。

在通过网站来学习iOS开发知识这件事儿上，我总结得出，入门级的内容超过90%都是重复的，而进阶级的内容你又要花时间去确认正确性的问题。而我们要利用好有限的时间去高效学习的前提，就是要学会挑选适合自己的学习资料。

所以，当我在看到戴铭老师的《iOS开发高手课》专栏时，就毫不犹豫地订阅了。当然，在这之前我就知道戴铭，看过他很多的技术分享，干货十足，而且尤其喜欢他PPT的风格，佩服他能把事情做到极致的匠人精神。

说到我是如何学习专栏文章，或者其他资料去完善知识体系的，我觉得最重要的是要找到适合自己的方法，并养成一个良好的习惯持续学习。

对于任何一个开发人员来说，他的成长路径都是从 0 开始。那，对于 0 基础的入门同学，我总结了一个**由浅入深的学习步骤：会基本使用->会完全使用->会自己改造->会自己创造。这些步骤的谓语都是一个库或者是一个工具**。而在这整个的学习过程中，都离不开一个词，也就是**举一反三**。

接下来，我以一个发送网络请求的问题为例，和你分享我是怎么用这种方法去学习的。

在日常开发中，我们经常会遇到问题，而这些问题就是学习的起点。遇到问题后，我们通常会在网上寻找解决方案，而解决方案可能就是一个库，或者一个工具。假设我需要发送一个网络请求，搜到了AFNetworking，于是就开始了对网络框架的学习。

第一个阶段，会基本使用。通常，我们遇到的问题大概率是紧急的，所以找到一个解决方案时，看一些基本的使用文档，然后复制粘贴地开始使用了。这对新手来说是没问题的，但我建议你多去思考一层，而不是简单地去copy。

比如说，你可以举一反三。除了 AFNetworking 外，还有别的库也可以解决这个问题吗？这时，你需要在网络上进行深入搜索，查看更多的类似分享，然后就会发现还有很多网络库可以发请求。你可以先粗略看一下这些库的接口差异，选择最适合自己的网络库，而不是一定要用 AFNetworking。

第二个阶段是，会完全使用。解决完发送网络请求的问题后，你又遇到了更多的网络问题，依旧需要靠 AFNetworking 来解决。但前提是，你需要对这个库有更深入的理解。

而深入理解一个三方库的最好办法，就是阅读官方示例以及文档，甚至阅读源码。因为，我们从其他技术文章里面得到的大多是基础用法，距离你了解这个库的所有功能，进而去更优雅地解决问题还差很远。

到这个阶段，你还可以举一反三，针对 AFNetworking 这个网络库去思考更深入的问题。比如说，除了网络请求，AFNetworking 还能帮我解决什么问题吗？

通过阅读 AFNetworking 的源码，你可以了解到 AFNetworking 还有很多很强大的功能。比如，AFNetworking 不仅能支持传递明文的 JSON 数据，同时也能支持各种类型的加解密；除此之外，还能帮我监控当前的网络环境，是离线还是 4G 还是 WiFi，可以让你在即将有大流量消耗的操作前提示用户是否继续。

第三个阶段是，会自己改造。随着业务发展，现有的网络库可能没办法支撑你的业务需求，你需要考虑改造现有的网络库。此时，你可以根据之前阅读的源码，掌握大佬的设计思路，并按照这个思路去修改和扩展 AFNetworking。

在此基础上，你可以继续举一反三。你通过搜索发现，另一个网络库可以解决你的问题，这时你可否考虑手动修改现有的网络库，来增加其他网络库的功能呢。

第四个阶段，会自己创造。你在 App 开发过程中，会用到大量的第三方库，可以通过阅读源码去了解它们的设计思想。这样，当你遇到现有的第三库不能解决自己的问题时，就可以通过这些设计思想去封装一个库来使用，甚至开源出来供其他人使用。

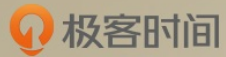
我认为，创造是学习的最后一步。当你开始创造自己的网络库时，所考虑的就不仅仅是接口调用了，还要考虑接口设计的易扩展性等，甚至还要去深入学习 HTTP 传输协议，以及其他平台的网络库的设计思路。相比于会基本使用，这时你学习到的知识将会更加全面。

这时，你可以再举一反三。在开发 App 过程中，你为解决网络问题开发了一个库。你可以将这个库的接口设计得更完整，将架构设计得更灵活易扩展，使得它可以被用于公司的其他 App。然后，你可以将这个库从 App 中独立出来，变为一个大家都可以用的私有库，甚至将它放到 GitHub 上，供更多的开发者使用、完善其功能。

这就是我这几年来一直在坚持的学习方法了，而关于学习时间，我习惯在每天的早晨和临睡前，花上 2~3 个小时，去阅读些技术资料来提升自己。我坚持每天利用好好这两三个小时的时间，直到现在这已经形成了我的习惯。

希望我的分享，可以对你有所帮助。

在最后，我特别感谢这4位同学的分享。如果你也有自己的故事想要分享，欢迎在留言区分享。我会在发布结束语的时候，选出留言点赞数最高的那位同学，送出一份“学习加油礼包”，期待你的故事。



iOS 开发高手课

从原理到实战，带你解决 80% 的开发难题

戴铭

前滴滴出行技术专家



新版升级：点击「👤请朋友读」，20位好友免费读，邀请订阅更有**现金**奖励。