

## 20-iOS开发的最佳学习路径是什么？

你好，我是戴铭。

我在专栏的第一篇文章中，就和你分享了我的 iOS 知识体系。通过前面“基础篇”内容的学习，你有没有总结出一套高效地构建自己的知识体系的路径呢？

今天这篇文章，我就重点和你分享一下，从一个新人到专家的学习路径，希望帮你少走一些弯路，快速成长。这条路径里面不仅有我的亲身经历，更有我观察到的、身边的那些优秀开发者的经历。所以，你大可放心地按照这条路径走下去，达到领域专家的水平一定是没有问题的。

在我看来，iOS开发者，可以分为这么三类：**刚跨入 iOS 领域的开发者、有一定 iOS 实际开发经验的开发者，以及 iOS 开发老手**。接下来，我就和你聊聊这三类人分别应该如何精进自己的iOS开发能力，成为这个领域的专家。

在我开始讲述这三类人的成长路径之前，我先和你分享一下完全0基础想要学习iOS开发的群体，可以参考什么样的资料来入门。

如果你喜欢通过书籍来学习的话，我推荐你去看看《iOS编程》（iOS Programming）这本书。这本书的内容，包括了开发语言、Cocoa 设计模式和控件使用、Xcode 技巧等，涉及 iOS 开发基础的方方面面。因此，它非常适合 iOS 编程0基础的人阅读，在 Quora 上被评为 iOS 最佳入门书。而且每次 iOS 系统和开发语言增加了新特性，这本书都会进行同步的版本更新。

如果你习惯于通过手把手的实例来学习的话，我推荐你看一下[APPCODA](#)网站。这里面每篇教程都非常简单易懂，而且详细，可以带着你一步一步地动手编写程序，非常适合初学者。同时，这个网站的更新也很频繁。

当然了，不排除以后还会有小学生朋友阅读这个专栏，那么如果你还在读小学或者初中，希望能够自己开发App的话，你可以看一下知乎里的这个问答“[12岁如何入门iOS编程？](#)”。这个问题下的被赞的最多的回答里，列出了很多孩子编程成功的事例，相信也可以让你信心满满。

接下来，我们就具体看一下刚跨入 iOS 领域的开发者、有一定 iOS 实际开发经验的开发者，以及 iOS 开发老手分别应该选择什么样的进阶路径吧。

### 不贪基础知识

是不是总是有人和你讲，不论做什么，都要先打好基础。但是，基础怎样才算打好了呢，却没有人可以替你做主。因为，基础知识涉及面实在是太广了，而且多偏理论，如不经过实践的检验，要想掌握好也不是件简单的事情。

但我这么说，并不是说基础就不重要了，必要的基础一定要掌握，而且还要掌握好。那么，对于 iOS 开发者来说，哪些基础是在开始就需要好好学的呢？

我先将新入 iOS 开发的人群进行一个细分，包括在校生、刚参加工作的应届生、从其他领域转向 iOS 开发的老手。

对于在校生来说，我推荐斯坦福大学在 iTunes U 上的 [App 开发课程](#)，网上还有[同步的中文字幕项目](#)。这个

课程中包含了开发一个 App 需要的、最基础的知识，同时还会现场一步一步带领你来开发一些简单的 App。

这个课程中会涉及 MVC 架构、iOS 开发语言基础知识、多点触摸、动画、ViewController 的生命周期、Scroll View、多线程、Auto Layout、拖拽、TableView、Collection View、Text Field、持续化存储、弹窗、通知、整个 App 生命周期、Storyboards、Core Motion、摄像等内容。

跟着斯坦福的这个课程学完，动手开发几个简单 App 上线是没什么问题的。但是，你不要就此满足，因为真实工作中光靠这些知识是远远不够的，甚至光靠这些知识连面试都过不了。基于类似这样简单应用层面的知识，掌握斯坦福的这个课程就可以了，不要一味地贪多，后面还有更多值得深入学习的知识在等着你。而且，应用层面的基础知识根据实际工作需要再去学习也不迟。

**对于刚参加工作的应届生**，我就不推荐斯坦福的这个课程了。不是说不适用，而是对于应届生来说，可能一上来手头就会有开发任务，怎么能够快速地上手完成任务才是最重要的。因为，公司是不会等到你掌握好了各种开发知识以后，才让你开始工作的；而且，不同公司的产品使用的技术侧重点也会有很大差异，也等不到你掌握好各种开发知识后才让你开始工作。所以，对于这个阶段的 iOS 开发者来说，如何快速上手完成任务才是最重要的。

针对应届生，我推荐苹果[官方 iOS 开发指南](#)，虽然内容不多，但却能够帮你快速掌握开发思路。实际工作中碰到所需要的知识，首先翻看[官方开发手册](#)，先在这里面找解决方法。官方代码示例都很规范，分类也很清晰，内容也更全。从大块上，可以分为 App Frameworks、图形、App 服务、媒体、开发工具、系统等。

- App Frameworks 里面主要是 Foundation 框架、UIKit、AppKit 这类文档资料。
- 图形分类里包含了 UIKit 更底层的 Core Graphics、Core Image、Core Animation，还有 ARKit、Metal、GameKit、SpriteKit 等也在这里面。
- App 服务里是苹果公司为开发者提供的服务套件，比如推送套件 PushKit、富文本套件 Core Text、方便集成机器学习模型到你 App 的 Core ML、车载应用的 CarPlay 等。JavaScript 引擎 JavaScriptCore 在 iOS 中应用接口的资料，你也可以在这个分类里找到。
- 媒体里主要包含了 AVFoundation、Core Audio Kit、Core Media 这些音视频的内容。
- 开发工具里有 Swift Playgrounds、XcodeKit、XCTest。
- 系统这个分类里的内容非常多而且实用，有网络基础 CFNetwork 和 Network、多核队列相关的 Dispatch、内核框架 Kernel、运行时库 Objective-C Runtime、安全 Security。

这份开发手册内容大而全，没必要在开始时就什么都学，不然耗费大量精力学到的东西却一时难以用上，会导致你的职业道路走得过慢。我觉得用时再看即可，只要你记着，工作中碰到 iOS 不清的知识，先到这里来找就对了。

**对于已经有多年其他领域开发经验的开发者来说**，通过几个示例代码，或者看看已有项目中的代码就能够快速上手了。其实，iOS 开发中的 Objective-C 语言实际上就是 C 语言的一个超集，有 C 和 C++ 基础和经验的开发者转行过来是很容易的，在开发思想上也有很多可以互相借鉴的地方。而 Swift 语言也融入了很多新语言的特性，比如闭包、多返回值、泛型、扩展、协议的结构体、函数编程模式等，你再学起来时也会有似曾相识的感觉。

对这个阶段的开发者，我推荐到 [RayWenderlich](#) 网站里翻翻它们的教程。这里的每一个教程都会详细指导你如何一步一步去实际完成一个项目，来达到掌握一个知识点的目的。RayWenderlich 里也有很详细的分类，

你可以根据实际工作需要来学习。我的建议同样是，这里知识虽好，但不要贪多。

关于掌握了一定的基础知识后，如何继续学习来提升自己的iOS 开发技能，在5年前唐巧写了篇博客“[iOS开发如何提高](#)”、limboy 写了篇“[自学 iOS 开发的一些经验](#)”，里面提到的这些提高自己开发能力的方法，拿到今天依然适用。不过，学习终究是需要实践来验证的。

在进入实战阶段之前，为了避免少走弯路，你需要一份 iOS 最佳实践。这里有一份一直在维护的[最佳实践指导](#)，里面包含了完整的 App 开发生命周期，从 IDE 搭建工程最佳使用方式、基础库选择、架构选择、存储方式、资源管理、代码规范、安全、静态分析、崩溃分析、构建，到部署，甚至是 IAP(In-App Purchases 应用内支付)都考虑到了。

实战过程中手册是最常用的，特别是 Swift 语言特性非常多，在还没能达到熟练的情况下，手册往往能够起到查漏补缺的效果。Swift 语言的手册就在它的官网，你可以点击[这个链接](#)查看；[中文版手册](#)，现在是由 SwiftGG 在维护，目前已经同步更新到了 Swift5。如果你想及时获得Swift的最新消息，可以订阅 [Swift 官网的博客](#)。

## 在实践中积累

基础知识不要贪多，但对于工作实践中涉及到的领域相关知识，我的建议是一定要贪多，而且越多越好。在实践中多多积累工作中涉及的相关知识，这种学习方法特别适合有了几年工作经验的开发者。此外，你还要时刻关注和你工作内容相关的领域知识的发展动向，并全面掌握，从而达到由量变到质变的效果，最终达到领域专家的水平。

举个例子吧。有一段时间，我的工作是和 App 性能相关的。这段时间，我会首先在网上收集一些其他公司在性能上做的事情，然后针对那些事情再去学习相关知识，平时还会通过订阅一些博客和技术团队的输出，持续关注性能这个领域。

等工作上做出了些成绩以后，我就会及时进行整理和总结。在这个过程中，再进行一些思考，多问问自己为什么这么做，还有没有更好的做法。最后再输出，看看其他人和团队的意见和建议。通过交流相互成长，独乐乐不如众乐乐，何乐而不为呢。

对于学习和积累什么样的知识，我的建议是，你一定要怀着一颗饥渴的心，查找资料时永远不要觉得够了。查到了很多资料后要多总结、多思考，这样才会有新的思路、新的方案出来。如果你细心观察 iOS 技术这几年的发展，就会发现很多方案刚开始都很简单，但是随着对底层的研究和思考后会出现更多更优解。

以内存监控方案为例。这个方案一开始是到 JetsamEvent 日志里找到那些由内存引起的系统强杀日志来做监控的，但是很难精确定位到问题。

后来，随着 fishhook 这种可以 Hook 系统 c 方法的库浮出水面，最终将其引入到了获取内存上限值的方案里。引入fishhook后的原理是，先找到分配内存都会执行的 c 函数 malloc\_logger，然后使用 fishhook 去截获，并记录到日志里，从而可以精确分析出内存过大的那个对象是在哪里分配的内存。

所以，我也会在专栏里面，分享一些我在实际工作中积累的领域知识，帮你解决这部分的痛点。我相信，这些领域知识也正是你所需要的。

## 殊途同归，深挖知识

在上面的内容中，我提到说很多方案，都是在不断地研究底层原理的基础上日趋完善的。由此看出，在基础知识掌握到一定程度、领域知识积累到一定程度后，我们需要再继续深挖知识，也就是众多计算机细分领域中的通用底层知识。

在我看来，底层知识是最值得深挖的，不管哪个领域，殊途同归，底层都是需要持续学习的。这里我推荐 Michael Ash 的 [“The Complete Friday Q&A”](#)。这本书里面涉及的内容，都是对一些知识点更深一层的探究，会让你了解更多内存、性能和系统内部的原理，帮你理解那些万变不离其宗的核心知识。

同样，我也会在专栏里面，通过5篇文章的篇幅，和你分享那些通用的底层知识，也就是系统内核XNU、AOP、内存管理和编译这些主题。

**当你 iOS 基础打牢了，也积累了很多实践经验，工作上也取得了一定成绩，那你也就成长为一名 iOS 开发老手了。**这个时候，你可以选择一个方向持续地深入学习。在我看来，最好的深入学习方式就是从头开始实现一个技术雏型。

如果你想对 LLVM 编译器能够有更加深刻的理解，那就去实现一个简单的编译器或解释器。比如，这个最小的 C 编译器 [OTCC](#) ( Obfuscated Tiny C Compiler ) 就是一个很好的实践。

如果你想更多地了解数据库的原理，除了看已有数据库的源码外，还可以自己动手开发一个简单的数据库项目。这里有个教程 [“Let’ s Build a Simple Database”](#)，可以一步步地教你如何开发一个简单的数据库。你可以照着这个教程，动手做起来。

甚至是，如果你想更多地了解操作系统，也可以学着从头创建一个。这样做，可以帮助你更深刻地理解分页、信号量、内存管理等知识。这里有个非常好的教程叫作 [“os-tutorial”](#)，你可以跟着动手做起来，去学习怎么开发一个麻雀虽小五脏俱全的操作系统，包括系统启动、中断处理、屏幕输出键盘输入、basic libc、内存管理、文件存储系统、简单的 shell、用户模式、文本编辑器、多进程和调度等。

对于 iOS 老手来说，你能够使用已经掌握的技术栈，触类旁通地去接触其他领域来拓宽自己的技术视野。以服务端开发为例，使用 [Perfect](#)就能够用 Swift 语言来做服务器的开发。

Perfect是由加拿大一个团队开发并维护的，这个团队的成员对技术都很有热情，热衷于用优秀的技术去做优秀的事情。所以，Perfect不仅功能丰富，而且紧跟最新技术的发展，对TensorFlow、NIO、MySQL、MongoDB、Ubuntu、Redis的支持做的都很到位。

这也是我们作为iOS开发老手，需要借鉴并学习的。

## 小结

今天这篇文章，我和你总结了一下 iOS 的学习路径。

随着工作时间的增加，为了将工作做得更好，我们会去学习更多的知识。但是，学得越多就会发现缺失的知识越多，那么接下来的学习之路该何去何从呢？有没有办法少走弯路呢？如何才能够系统化学习呢？

我觉得如果你到了这一步，有了这些困惑，就应该好好去梳理自己的知识体系了，查漏补缺让自己的知识更体系化。

我参与过日活千万和日活过亿的 App 项目，团队规模和代码规模都很大，攻坚过很多难解问题，对于性能



质量和开发效率也有过很多思考和实践，我都会在这个专栏中与你一一分享，希望能够对你有帮助。

现在，我们已经掌握了构建自己iOS开发知识体系的高效路径，接下来需要做的就是不断自我完善的过程了。

今天我留给你的课后思考题，就是请你来总结一下自己学习iOS开发的方法，并和我分享一下吧。

感谢你的收听，欢迎你在评论区给我留言分享你的观点，也欢迎把它分享给更多的朋友一起阅读。



# 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。

## 精选留言：

- 细腻的莽夫 2019-04-25 08:58:53  
铭哥果然能力惊人 一篇学习引导都是干货十足 [1赞]
- 黄昏 2019-04-25 12:19:06  
我在工作中遇到问题，就会把这个知识点看懂。但是没有进行相关知识点的扩充，是需要优化的。  
  
当然，针对具体知识点学习的时候，确实发现自己懂的太少了，听作者的话，好好梳理下自己的知识点去。
- Geek\_d7ad60 2019-04-25 10:20:08  
在早前就留意到您的小专栏没收费，觉得很良心，这些文章很干货。顺带想问您一个问题，假如英文水平低的人该如何学习编程呢？
- 尘归心 2019-04-25 09:48:03  
github，博客，看书，github--iOS-Weekly还是挺不错的,之前学习swift是在看的swiftGG上的手册，但是除了一些设计模式或者一部分常用的知识，其他的看完后面就不怎么记得了，如何解决工作中不常用的知识的记忆问题，如果说反复的看就又感觉很花时间
- Geek\_Charlie\_Z 2019-04-25 08:33:24  
戴老师，Vapor和Perfect都是Swift在服务端的框架，根据您的经验，能不能分享下您对服务端框架优缺点

的理解呢？

- 疯狂的丹顶鹤 2019-04-25 07:28:54

戴老师，能不能增加一些数据传输，安全防范方面的经验分享啊？