

## 31-iOS开发学习资料和书单推荐

你好，我是戴铭。

在更新专栏的这段时间里，我收到很多同学的反馈，希望我能推荐些iOS开发的优秀学习资料和图书，来帮助自己高效学习。确实，现在各种学习资料非常丰富，但这些资料一方面质量参差不齐，另一方面搜索出适合自己的内容也需要花费不少时间。

快速找到经过检验的、适合自己的学习资料，不仅可以提升我们的学习效率，还能帮助我们快速解决现阶段遇到的问题。所以，今天我就来跟你分享一些我觉得不错的学习资料和图书。

### 学习资料

iOS 开发往往会涉及界面交互，[iOS Examples](#)和[Cocoa Controls](#)这两个网站收集了大量的开源组件和库，并且进行了非常细致的分类，你能够非常方便地找到适合自己的“轮子”。

如果你希望通过完整的例子来系统学习 App 开发，我推荐你查看一下GitHub上的[Open-Source iOS Apps](#)项目。作者在这个项目中收录了很多优秀的、完整的开源 iOS App，并做了详细分类，还专门标出了上架了 App Store 的开源 iOS App。

AFNetworking 和 Alamofire 的作者 Mattt 维护着一个 [NSHipster](#) 的网站，主要关注的是一些不常用的 iOS 知识点。如果你想扩展自己的iOS 知识面，这个网站里的内容会非常适合你。

[Awesome iOS](#) 也是一个值得推荐的网站，里面包含了 iOS 开发的方方面面，而且内容都是经过人工筛选、分类的。我觉得，**你遇到任何 iOS 的问题，都应该先到这个网站看看。**

Awesome iOS 最大的特点就是大而全，囊括了从开发、调试到发布 App Store 的各种学习资料，也包含了博客、书籍、教程、邮件订阅、播客的推荐。同时，这个网站还包括了 iOS 开发更细分的 Awesome 推荐，比如关于 ARKit 的 [Awesome ARKit](#)，关于面试问题收集的 [Awesome iOS Interview](#) question list 等等。

你还可以通过关注一些知名开发者动态的方式，来为自己学习方向的判断做输入。

这里有份列表，列出了 iOS 领域那些[知名开发者](#)，你可以通过关注他们的博客、Twitter、GitHub，来了解走在 iOS 领域前沿开发者的视野和 iOS 最新的动向。除了关注知名开发者外，你还可以关注下[开源项目团队](#)的列表，如果你正在使用他们的开源项目，通过关注他们的动向，随时了解这些开源项目的最新进展。

### 图书推荐

通过上面我和你推荐的学习资料，你可以去分析并解决在开发中遇到的问题，也可以通过知名开发者和优秀开源项目的团队动态，来了解iOS开发的技术动向。但是，如果你想要透彻地掌握某领域的专业知识，还是需要静下心来慢慢学习。而，阅读相关书籍，就是一种很好的学习方式。

那么，接下来我再跟你推荐一些我觉得还不错的书籍，希望能够对你的学习有所帮助。

[Raywenderlich](#) 出版的图书质量都非常不错，可以一步一步教你掌握一些开发知识，内容非常实用，而且这些图书的涉及面广。比如，这些图书包括有 [ARKit](#)、Swift 服务端的 [Vapor](#) 和 [Kitura](#)、[Metal](#)、[数据结构和算法的 Swift 版](#)、[设计模式](#)、[Core Data](#)、[iOS 动画](#)、[Apple 调试和逆向工程](#)、[RxSwift](#)、[Realm](#)、[2D](#) 和 [3D 游](#)

戏开发等各个方面。

另外，[objc.io](#)家的图书会从原理和源代码实现的角度来讲解知识点，也非常不错，内容比 Raywenderlich 出版的图书更深入，适合有一定 iOS 开发经验的人阅读。

Raywenderlich 和 objc.io 的书基本都是 Swift 来写的。如果你想更深入地理解 Objective-C 的话，我推荐 [《Objective-C高级编程》](#) 这本书。这本书里的知识点并不多，主要讲的是内存管理、Block、GCD (Grand Central Dispatch)。

这三个知识点对Objective-C来说非常重要，如果使用不当将会置你的工程于风险之中。正是因为涉及的知识点不多，所以全书能基于苹果公司公开的源码，集中讲清楚这三个知识点。这，非常难得。因此，如果你对内存管理、Block、GCD 了解地不是很透彻，我建议你仔细阅读这本书。

如果你想要了解系统工作原理的话，我推荐阅读 [《程序员的自我修养 - 链接、装载与库》](#)。这本书详细且深入地讲解了硬件、操作系统、线程的知识。

阅读这本书之前，你需要先掌握 CPU、计算机原理、汇编、编译原理、C 语言、C++语言等计算机学科的基本知识。掌握了这些知识后再阅读这本书，它能帮你把知识串起来，帮你从代码编译、链接到运行时内存虚拟空间和物理空间映射的角度，了解一个程序从编写到运行时的底层知识的方方面面。

现在编程技术不断推陈出新，不断通过添加中间层将底层埋住，新一代开发人员也越来越不重视底层知识，所以当他们的上层知识被更新替代以后就会感叹赶不上技术更新脚步，知识焦虑感越来越严重。

而读完这本书，你就会发现，有些知识是不会变的，不管上层如何变化，只要抓住这些知识就能够抓住核心，掌握技术的走向。

《程序员的自我修养 - 链接、装载与库》耗时30年才被出版，期间作者不断优化其中的内容，最终成为一本经典图书。正如其名，程序员的自我修养，就是来自对程序运行背后机制的学习，而不是一味地追新。除了内容地道以外，这本书的作者们精益求精的精神也非常值得我们学习。

当你有了大量的编程经验，需要考虑工程架构合理性的时候，我推荐你看看 [《架构整洁之道》](#) 这本书。架构设计的思想也不会过时，并适用于所有的知识领域。

这本书详细分析了架构的目标价值、编程范式、架构设计原则、组件拆分和边界处理。其中，编程范式介绍了结构化编程、面向对象编程、函数式编程等。设计原则包含了开闭原则 OCP、单一职责原则 SRP、里氏替换原则 LSP、接口隔离原则 ISP、依赖反转原则 DIP 等等，内容十分丰富。熟练掌握这些架构设计原则，会让你对架构的理解更深，也可以从更多方面去思考。

值得一提的是，这本书还通过实践案例把所讲知识都串了起来，让你更容易理解架构设计的知识。

## 小结

在今天这篇文章中，我和你分享了很多关于iOS开发的学习资料，这其中有优秀的开源项目，也有一些经典的图书。尤其是《程序员的自我修养 - 链接、装载与库》和《架构整洁之道》这两本书，值得你反复阅读。在不同阶段去阅读这两本书，你会有不同的体会，也会有不同的收获。

如果你还想阅读更多计算机编程相关的经典书，可以再看看《编码》、[《代码整洁之道》](#)、《代码大全》、

《算法》。

每个人的学习时间都是有限的，上班时要争分夺秒的完成任务，下班放假还要放松休息。与其无止境的寻找资料，还不如静下心来阅读经典和多一些思考。所以，希望我今天的这篇文章可以帮到你。

## 课后作业

你在读过的书里，哪一本对你帮助最大呢？

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。