



iOS开发进阶

唐巧 著



iOS开发进阶

唐巧 著

電子工業出版社
Publishing House of Electronics Industry

内 容 简 介

本书分工具、实践、理论三大部分。第一部分介绍 iOS 开发的常用工具，第二部分介绍 iOS 开发中的一些常见的实践经验，第三部分介绍 iOS 开发中涉及的原理。

如果把成为 iOS 开发高手的过程比作武侠小说中的修炼过程的话，工具、实践和理论的学习就分别对应兵器、招式和内功的修炼。本书希望通过这三方面的综合讲解，全面提高开发者的技能水平。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目 (CIP) 数据

iOS 开发进阶 / 唐巧著. — 北京: 电子工业出版社, 2015.1

ISBN 978-7-121-24745-3

I. ① i…II. ① 唐…III. ① 移动终端—应用程序—程序设计 IV. ① TN929.53

中国版本图书馆 CIP 数据核字 (2014) 第 268564 号

责任编辑: 徐津平

印 刷: 北京丰源印刷厂

装 订: 三河市鹏成印业有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱 邮编 100036

开 本: 787×980 1/16 印张: 16 字数: 328 千字

版 次: 2015 年 1 月第 1 版

印 次: 2015 年 1 月第 1 次印刷

定 价: 65.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888。

质量投诉请发邮件至 zlt@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线: (010) 88258888。

人生如长跑，成长如进阶

曾经有很多次想提笔写一本纯粹的技术图书，有几次甚至已经把目录做完了，但是看到“层峦叠嶂”的目录结构，我总是心生恐惧。一想到要耗费一年多的业余时间去写那些枯燥无味的技术文字，并逐一核对各种技术细节，我仿佛置身幽暗的森林而无法自拔。计划还未开始，就已经结束了。

总有人会去做这些费时费力并能够惠及大众的事情，他们像一个个沉默的跑者，在奔跑的同时，不停地标出路线，记录经验，传播技巧，并吸引更多的人加入跑步的队伍一起成长。

唐巧就是其中之一。

我很早就认识了唐巧，那时他还是一个初入江湖的“小球”，高高瘦瘦，正在网易有道做云笔记的开发。短短几年之内，唐巧获得了飞速的成长，他不仅是一个优秀的 iOS 开发者，而且成为 iOS 领域的布道者。他从 2011 年开始构建独立博客“唐巧的技术博客”，撰写了大量的 iOS 开发相关的文章，成为知名 iOS 博主。2012 年唐巧离开网易有道，加入猿题库，成为一个创业者。期间他组建了由 iOS 开发领域顶尖开发者构成的“真·iOS 开发”QQ 群，大家一边在群里讨论技术问题，一边通过各自的博客把成果分享出去。我有幸参与其中，可以说，这是 iOS 领域最专业的技术群体之一。2013 年春节，唐巧开通了微信公众平台“iOS 开发”（iOSDevTips），开始基于微信公众号传播自己的开发理念和相关技术，迄今为止已经积累了一万多名专业订阅者。

在承担日常研发任务的同时，还能做这么多的技术写作和内容分享，他让我这个老程序员亦十分佩服。

2013 年的一天，我在一次技术会议上碰到了唐巧，谈起了技术写作的事情。现在国内 iOS 开发领域的技术图书良莠不齐，很多是“编著”，摘抄或翻译自官方文档，而且大家很喜欢去写入门类的图书，从编程语言、MVC、协议、策略到 Xcode 的使用，到第一个 App 项目的

构建，等等，高级一点的进阶内容就少有涉及，很多 iOS 开发者让我推荐一本面向中高级开发者的中文书时，我总是遍寻不着。那时候我正准备出版自己的《MacTalk·人生元编程》，于是也希望唐巧写一本 iOS 进阶方面的书。我对唐巧说，你的博客上已经产出了那么多高质量的 iOS 开发进阶内容，为什么不再增加一些内容，集集成册梳理成书呢？他若有所思地回答，我正准备这么干！

经年以后，唐巧再也没和我提起这件事，我也在忙忙碌碌中把他的书忘得一干二净。上周突然收到唐巧的一份书稿，打开一看，《iOS 开发进阶》静静地躺在邮箱里，已然成书。唐巧用了一年的时间，在博客文章的基础上进行完善、修改、增补，用清晰的“丝线”把散落各处的“珍珠”串起，形成脉络完整的知识体系，然后又新增了超过成书一半的内容，最终完成了这本关于 iOS 开发的进阶图书。

书中从 iOS 开发实战最常用的工具讲起，一直写到底层开发原理。内容分成三块：工具、实践和原理，每个章节都独立成文，读者可以从头细细品读，也可以随时翻阅其中一节，利用碎片时间阅读学习。就我个人而言，非常喜欢实践和原理两个部分，比如处理多核编程的 GCD 技术、应用程序安全技术、CoreText 的排版引擎、Objective-C 对象模型、block 对象模型等，这些内容要么能帮助我解决实际问题，要么能够答疑解惑，深入细节。作为一个仍然在一线开发的技术作者，唐巧采用了图文并茂的方式去阐述问题，每个技术专题都配备了恰当的代码，力求将复杂的技术以最为简洁的方式呈现出来，让读者一目了然，读来深入浅出，并不枯燥。

写书永远有遗憾。由于出版时间的要求，这本书的原理部分略显单薄，另外，全书并未涉及苹果的最新语言 Swift 相关的内容，相关的代码示例都是基于 Objective-C 的。希望唐巧的下一本书能够涉及更多这部分内容，这也是我的一个心愿。

在这样一个原创匮乏的年代，我能做到的就是向更多的人去推荐这样一本书。人生如长跑，成长如进阶，如果你是一个 iOS 开发者，这本书一定不要错过！

池建强

《MacTalk·人生元编程》作者

微信平台 MacTalk 出品人

2014 年，冬

目录

推荐序	iii
第 1 章 引言	1
1.1 我为什么要写本书	1
1.1.1 我的构思	1
1.1.2 内容导读	1
1.1.3 目标读者	2
1.1.4 随书示例代码和勘误	2
1.1.5 致谢	2
1.2 如何提高 iOS 开发技能	3
1.2.1 阅读博客	3
1.2.2 读书	3
1.2.3 看 WWDC 视频	4
1.2.4 看苹果的官方文档	4
1.2.5 看开源项目的代码	4
1.2.6 多写代码，多思考	4
1.2.7 多和同行交流	5
1.2.8 分享	5
I 第一部分：iOS 开发工具	
第 2 章 使用 CocoaPods 做依赖管理	9
2.1 CocoaPods 简介	9
2.2 CocoaPods 的安装和使用	10
2.2.1 CocoaPods 的安装	10

2.2.2	使用 CocoaPods 的镜像索引	10
2.2.3	使用 CocoaPods	11
2.2.4	查找第三方库	11
2.3	注意事项	12
2.3.1	关于.gitignore	12
2.3.2	为自己的项目创建 podspec 文件	12
2.3.3	使用私有的 pods	12
2.3.4	不更新 podspec	12
2.3.5	生成第三方库的帮助文档	13
2.3.6	原理	13
第 3 章	网络封包分析工具 Charles	15
3.1	Charles 简介	15
3.2	Charles 的安装和使用	16
3.2.1	安装 Charles	16
3.2.2	安装 SSL 证书	16
3.2.3	将 Charles 设置成系统代理	17
3.2.4	Charles 主界面介绍	18
3.2.5	过滤网络请求	18
3.3	使用 Charles 协助 iOS 开发	19
3.3.1	截取 iPhone 上的网络封包	19
3.3.2	模拟慢速网络	21
3.4	高级功能	22
3.4.1	截取 SSL 信息	22
3.4.2	修改网络请求内容	23
3.4.3	修改服务器返回内容	25
3.5	Map 功能	25
3.6	Rewrite 功能	27
3.7	Breakpoints 功能	29
3.8	总结	30
3.9	参考链接	30
第 4 章	界面调试工具 Reveal	31
4.1	Reveal 简介	31
4.2	Reveal 的使用	32
4.2.1	用 Reveal 连接模拟器调试	32
4.2.2	用 Reveal 连接真机调试	34
4.2.3	用 Reveal 调试其他应用界面	34

4.2.4	总结	35
4.2.5	参考资料	36
第 5 章	移动统计工具 Flurry	37
5.1	Flurry 简介	37
5.2	Flurry 的基本使用	38
5.2.1	注册和下载对应 SDK	38
5.2.2	集成 SDK	40
5.2.3	自定义统计项	42
5.2.4	查看统计结果	43
5.2.5	统计 Crashlog	44
5.3	对比和总结	45
5.3.1	和其他统计分析平台的对比	45
5.3.2	总结	45
第 6 章	崩溃日志记录工具 Crashlytics	47
6.1	Crashlytics 简介	47
6.2	Crashlytics 的使用	49
6.3	实现原理和使用体会	52
第 7 章	App Store 统计工具 App Annie	55
7.1	App Annie 简介	55
7.2	App Annie 的使用	56
7.3	App Annie 账号的注册及配置	59
7.4	和其他工具的对比	60
7.4.1	官方的命令行工具	60
7.4.2	其他类似服务	61
7.4.3	功能对比	61
第 8 章	Xcode 插件	63
8.1	Xcode 插件管理工具 Alcatraz	63
8.1.1	简介	63
8.1.2	安装和删除	64
8.1.3	使用	64
8.1.4	插件路径	66
8.2	关于 Xcode 的插件机制	66

8.3 常用 Xcode 插件	66
8.3.1 KSIImageNamed	66
8.3.2 Xvim	67
8.3.3 FuzzyAutocompletePlugin	67
8.3.4 XToDo	67
8.3.5 BBUDebuggerTuckAway	68
8.3.6 SCXcodeSwitchExpander	68
8.3.7 deriveddata-exterminator	68
8.3.8 VVDocumenter	69
8.3.9 ClangFormat	69
8.3.10 ColorSense	69
8.3.11 XcodeBoost	70

第 9 章 其他工具介绍 71

9.1 取色工具：数码测色计 (DigitalColor Meter)	71
9.1.1 前言	71
9.1.2 使用介绍	71
9.1.3 其他类似工具：xScope	72
9.2 其他图形工具	73
9.2.1 ImageOptim	73
9.2.2 马克鳗	74
9.2.3 Dash	74
9.2.4 蒲公英	75
9.3 命令行工具	75
9.3.1 nomad	75
9.3.2 xctool	76
9.3.3 appledoc	76

II 第二部分：iOS 开发实践

第 10 章 理解内存管理 81

10.1 引用计数	81
10.1.1 什么是引用计数，原理是什么	81
10.1.2 我们为什么需要引用计数	83
10.1.3 不要向已经释放的对象发送消息	85
10.1.4 循环引用 (reference cycles) 问题	85
10.1.5 使用 Xcode 检测循环引用	88

10.2	使用 ARC	90
10.2.1	Automatic Reference Count	90
10.2.2	Core Foundation 对象的内存管理	92
第 11 章	掌握 GCD	95
11.1	GCD 简介	95
11.2	使用 GCD	97
11.2.1	block 的定义	97
11.2.2	系统提供的 dispatch 方法	98
11.2.3	修改 block 之外的变量	99
11.2.4	后台运行	99
11.2.5	总结	100
第 12 章	使用 UIWindow	101
12.1	UIWindow 简介	101
12.2	为 UIWindow 增加 UIView	102
12.3	系统对 UIWindow 的使用	102
12.3.1	WindowLevel	104
12.3.2	手工创建 UIWindow	105
12.3.3	不要滥用 UIWindow	108
12.3.4	参考资料	109
第 13 章	动态下载系统提供的多种中文字体	111
13.1	功能简介	111
13.1.1	前言	111
13.1.2	功能介绍	112
13.1.3	字体列表	112
13.2	使用教程	113
13.2.1	相关 API 介绍	113
13.2.2	总结	115
第 14 章	使用应用内支付	117
14.1	后台设置	117
14.2	iOS 端开发	119
14.3	服务端开发	121
14.4	注意事项	122

第 15 章 基于 UIWebView 的混合编程	123
15.1 混合编程简介	123
15.2 使用模板引擎渲染 HTML 界面	124
15.3 Objective-C 语言和 JavaScript 语言相互调用	126
15.4 如何传递参数	129
15.5 同步和异步	129
15.6 注意事项	130
15.6.1 线程阻塞问题	130
15.6.2 主线程的问题	130
15.6.3 键盘控制	130
15.6.4 CommonJS 规范	130
15.7 使用 Safari 进行调试	131
第 16 章 安全性问题	135
16.1 前言	135
16.2 网络安全	135
16.2.1 安全地传输用户密码	135
16.2.2 防止通讯协议被轻易破解	137
16.2.3 验证应用内支付的凭证	138
16.3 本地文件和数据安全	138
16.3.1 程序文件的安全	138
16.3.2 本地数据安全	140
16.4 源代码安全	140
16.5 总结	142
第 17 章 基于 CoreText 的排版引擎	143
17.1 CoreText 简介	143
17.2 基于 CoreText 的基础排版引擎	145
第 18 章 实战技巧	181
18.1 App Store 与审核	181
18.1.1 撤销正在审核的应用	181
18.1.2 申请加急审核	181
18.1.3 应用在市场的名字	182
18.1.4 测试设备数的限制	182
18.1.5 如何将应用下架	183
18.1.6 如何举报别的应用侵权	183
18.1.7 iTunes Connect 后台操作出错	184

18.1.8	Metadata Reject	184
18.2	开发技巧	184
18.2.1	UILabel 内容模糊	184
18.2.2	收起键盘	184
18.2.3	NSJSONSerialization 比 NSKeyedArchiver 更好	185
18.2.4	设置应用内的系统控件语言	185
18.2.5	为什么 viewDidLoad 被废弃	188
18.2.6	多人协作慎用 Storyboard	189
18.2.7	避免滥用 block	190
18.2.8	合并工程文件的冲突	192
18.2.9	忽略编译警告	193
18.3	Xcode 使用技巧	193
18.3.1	Xcode 快捷键	193
18.3.2	查找技巧	195
18.3.3	JavaScript 文件设置调整	195
18.3.4	清除 DerivedData	196
18.3.5	target 信息异常	197
18.3.6	下载 Xcode	197
18.4	调试技巧	197
18.4.1	模拟器快捷键	197
18.4.2	覆盖安装注意事项	197
18.4.3	给模拟器相册增加图片	198
18.4.4	获得模拟器中的程序数据	198
18.4.5	安装旧版本的模拟器	199
18.4.6	模拟慢速网络	199
18.4.7	异常断点与符号断点	199
18.5	ipa 文件格式	199
18.5.1	查看 ipa 的内容	199
18.5.2	查看 ipa 中的图片	200
18.6	为工程增加 Daily Build	201
18.6.1	前言	201
18.6.2	步骤	201
18.6.3	遇到的问题	206
18.6.4	总结	206
18.7	使用脚本提高开发效率	206
18.7.1	删除未使用的图片资源	206
18.7.2	用脚本自动生成小尺寸的图片	207
18.7.3	检查图片	208

18.8	管理代码片段	209
18.8.1	代码片段介绍	209
18.8.2	定义自己的代码片段	210
18.8.3	使用 Git 管理代码片段	211
18.8.4	其他代码片段管理工具	211

III 第三部分：iOS 开发底层原理

第 19 章	Objective-C 对象模型	215
19.1	简介	215
19.2	isa 指针	215
19.3	类的成员变量	217
19.4	对象模型的应用	220
19.4.1	动态创建对象	220
19.4.2	系统相关 API 及应用	222
19.4.3	参考文献	226
第 20 章	Tagged Pointer 对象	227
20.1	原有系统的问题	227
20.2	Tagged Pointer 介绍	228
20.2.1	Tagged Pointer	228
20.2.2	特点	230
20.3	注意事项和实现细节	231
20.3.1	isa 指针	231
20.3.2	64 位下的 isa 指针优化	231
20.3.3	isa 的 bit 位含义	232
20.3.4	总结	233
20.3.5	参考文献	233
第 21 章	block 对象模型	235
21.1	block 的内部数据结构定义	235
21.2	用 clang 分析 block 实现	237
21.2.1	NSConcreteMallocBlock 类型的 block 的实现	242
21.2.2	变量的复制	243
21.3	注意事项	243
21.3.1	避免循环引用	243
21.3.2	ARC 对 block 类型的影响	244

1.1 我为什么要写本书

1.1.1 我的构思

还记得几年前，我在学习 iOS 开发入门后，发现同行间的交流环境比较差，自己提高的过程很慢。在出版物方面，我也一直苦于没有找到一本 iOS 开发进阶方面的图书。

随着移动互联网的快速发展，移动开发的人才也慢慢增多，现在同行间的交流环境相比以前好了很多。但是在国内，仍然没有一本原创的 iOS 进阶方面的图书出现。因此，我萌生了写作的想法。

写作最开始是构思的过程，我回顾自己维护多年的 iOS 开发博客 (<http://www.devtang.com>)，从里面 100 多篇关于 iOS 开发的原创文章中，整理出涉及 iOS 开发进阶提高的三个主要的方向：工具、实践、理论。于是，我围绕着上述的三个方向，把以前的文章作为基础，再进一步完善相关内容，让零散的知识点能够衔接起来，成为一个完整的体系。整个写作的过程是痛并快乐着的，为了保证知识的完整，最终书稿中有超过 50% 是新增加的内容。

从知识的难度上，工具、实践、理论这三大部分的难度逐步提高，以保证读者能慢慢适应。而三大部分内容其实各自独立，有经验的读者也可以按兴趣点或具体需求，跳跃性地阅读本书。

1.1.2 内容导读

本书分工具、实践、理论三大部分。第一部分介绍 iOS 开发的常用工具，第二部分介绍 iOS 开发中的一些常见的实践经验，第三部分介绍 iOS 开发中涉及的原理。

如果把成为 iOS 开发高手的过程比作武侠小说中的修炼过程的话，工具、实践和理论的学习

就分别对应兵器、招式和内功的修炼。本书希望通过这三方面的综合讲解，全面提高开发者的技能水平。

好的工具可以使得开发效率成倍增长。本书第一部分介绍了 iOS 开发中的各种工具或服务，使用它们可以极大地方便我们的日常开发和维护，提高开发效率。

一个入门之后的 iOS 开发者，需要的是更深入的实际开发经验。本书第二部分选择了 iOS 开发中几个常见的实际场景，进一步介绍各种 iOS 开发进阶的实战技巧。对于一些比较零散的知识技巧，我也在实战小技巧集锦中进行了介绍。

如果把前两部分比作工具和招式的修炼的话，本书第三部分则是内功的修炼。在第三部分中，本书详细分析了 iOS 开发涉及的语言对象模型，从而能够帮助读者深入理解语言的各种特性和限制。

1.1.3 目标读者

本书定位于帮助那些 iOS 开发刚刚入门的同行快速提高自己的水平，适用于有三个月以上 iOS 开发经验的读者，不适合没有任何 iOS 开发基础的读者阅读。

对于 iOS 熟练开发者，本书也能补充其知识点的可能的盲区，使其相关知识体系更为完善。

1.1.4 随书示例代码和勘误

为了节省篇幅，本书中的示例工程代码只选取了关键的部分，完整的工程代码在与本书对应的 github 项目里，项目地址是：<https://github.com/tangqiaoboy/iOS-Pro>。

本书花费了我大量精力，但就像几乎没有毫无漏洞的代码一样，本书中难免会有一些错误，所以本书的勘误也会一同更新在该 github 项目中。

1.1.5 致谢

首先，感谢我的老婆和岳母，在我集中写作的这一年内，她们承担了大部分的家务，给了我足够安静的写作环境。然后，感谢我建的“真·iOS 开发”群里的每一个朋友，在我眼中这是全中国最牛的 iOS 开发者聚集地，我每天从大家的讨论中收获很多。最后，感谢电子工业出版社的张春雨老师、高丽阳老师，是他们的鼓励和协助，让本书能够快速出版。

在本书的写作过程中，我的宝宝也顺利降生，也把本书献给她，愿她能够健康快乐地成长。

1.2 如何提高 iOS 开发技能

许多人在博客上咨询我 iOS 开发技能如何提高，本书就是一本旨在帮助读者提高 iOS 开发技能的书，除了书中专门介绍的方法外，我也介绍其他一些方法。

1.2.1 阅读博客

在现在这个碎片化阅读流行的年代，博客的风头早已被微博盖过。而我却坚持写博客，并且大量地阅读同行的 iOS 开发博客。博客的文章长度通常在 3000 字左右，许多 iOS 开发知识都至少需要这样的篇幅才能完整地讲解清楚。博客相对于书籍来说，并没有较长的出版发行时间，所以阅读博客对于获取最新的 iOS 开发知识有着非常良好的效果。

我自己精心整理了国内 40 多位 iOS 开发博主的博客地址列表：<https://github.com/tangqiaoboy/iOSBlogCN>，希望大家都能培养起阅读博客的习惯。

国外也有很多优秀的 iOS 开发博客，它们整体质量比中文的博客更高，以下是一些值得推荐的博客地址。

博客名	博客地址
objc.io	http://www.objc.io/
Ray Wenderlich	http://www.raywenderlich.com
iOS Developer Tips	http://iosdevelopertips.com/
iOS Dev Weekly	http://iosdevweekly.com/
NShipster	http://nshipster.com/
Bartosz Ciechanowski	http://ciechanowski.me
Big Nerd Ranch Blog	http://blog.bignerdranch.com
Nils Hayat	http://nilsou.com/

另外，使用博客 RSS 聚合工具（例如 Feedly：<http://www.feedly.com/>）可以获得更好的博客阅读体验。手机上也有很多优秀的博客阅读工具（我使用的是 Newsfy）。合理地使用这些工具也可以将你在地铁上、睡觉前等碎片时间充分利用起来。

1.2.2 读书

博客的内容通常只能详细讲解一个知识点，而图书则能成体系地介绍整个知识树。与国外相比，中国的图书售价相当便宜，所以读书其实是一个非常划算的提高方式。建议大家每年至少坚持读完一本高质量的 iOS 开发图书。

1.2.3 看 WWDC 视频

由于 iOS 开发在快速发展，每年苹果都会给我们带来很多新的知识。而对于这些知识，第一手的资料就是 WWDC 的视频。

通常情况下，iOS 开发的新知识首先会在 WWDC 上被苹果公开，三个月左右之后，会有国内外的博客介绍这些知识，再过半年左右，会有国外的图书介绍这些知识。所以如果想尽早地了解这些知识，只有通过 WWDC 的视频。

现在每年的 WWDC 视频都会在会议过程中逐步放出，重要的视频会带有英文字幕。坚持阅读这些视频不但可以获得最新的 iOS 开发知识，还可以提高英文听力水平。

1.2.4 看苹果的官方文档

苹果的官方文档相当详尽，对于不熟悉的 API，阅读官方文档也是最直接有效的解决方式。

1.2.5 看开源项目的代码

大家一定有这样的感受，很多时候用文字讲解半天，还不如写几行代码来得直观。阅读优秀的开源项目代码，不但可以学习到 iOS 开发本身的基本知识，还能学习到设计模式等软件架构上的知识。

如果读者能够参与到开源项目的开发中，则能进一步提高自己的能力。

1.2.6 多写代码，多思考

知识的积累离不开实践和总结，我认为写过的 iOS 代码如果没有超过 10 万行，是不能称得上熟悉 iOS 开发的。某些在校的学生，仅仅做了几个 C++ 的大作业，就在求职简历里面写上“精通 C++”，真是让人哭笑不得。

在多写代码的同时，我们也要注意不要“重复造轮子”，尽量保证每次写的代码都能具有复用性。在代码结构因为业务需求需要变更时，及时重构，在不要留下技术债的同时，我们也要多思考如何设计应用架构，才能够保证满足灵活多变的产品需求。

在多次重构和思考的过程中，我们会慢慢积累出一类问题的最佳解决方式，成为自己宝贵的经验。