

Learning iOS Programming



/// **iOS**

编程指南

[美] *Alasdair Allan* 著
周亮 徐可 译

O'REILLY®

 人民邮电出版社
POSTS & TELECOM PRESS

O'REILLY®

iOS 编程指南

[美] Alasdair Allan 著
周亮徐可译

人民邮电出版社

北京

图书在版编目 (CIP) 数据

iOS编程指南 / (美) 艾伦 (Allan, A.) 著 ; 周亮, 徐可译. — 北京 : 人民邮电出版社, 2014. 5
ISBN 978-7-115-34664-3

I. ①i… II. ①艾… ②周… ③徐… III. ①移动终端—应用程序—程序设计—指南 IV. ①TN929.53-62

中国版本图书馆CIP数据核字(2014)第037843号

版 权 声 明

Copyright© 2013 by O'Reilly Media, Inc.

Simplified Chinese Edition, jointly published by O'Reilly Media, Inc. and Posts & Telecom Press, 2014. Authorized translation of the English edition, 2013 O'Reilly Media, Inc., the owner of all rights to publish and sell the same.

All rights reserved including the rights of reproduction in whole or in part in any form.

本书中文简体版由 O'Reilly Media, Inc. 授权人民邮电出版社出版。未经出版者书面许可, 对本书的任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

内 容 提 要

本书是经典的 iOS 编程指南, 针对 iOS 6 和 Xcode 4, 详细介绍了 iOS 平台程序开发的各方面主题。

全书共分 14 章, 包括 iOS 编程预备和基础知识、Objective-C 编程基础、表视图和其他视图应用、网络编程、数据处理、传感器应用、地图和定位、iCloud 介绍、系统应用集成、应用发布等各个方面。

本书主要面向有一定编程经验的开发者。如果你想为 iPhone 开发应用程序, 却是第一次接触 Mac 平台, 或者你之前从事 Mac 桌面应用开发, 但现在想转向 iOS 应用开发, 这本书就非常适合你。

◆ 著 [美] Alasdair Allan
译 周亮 徐可
责任编辑 陈冀康
责任印制 程彦红 焦志炜

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路 11 号
邮编 100164 电子邮件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
三河市海波印务有限公司印刷

◆ 开本: 787×1000 1/16

印张: 24.5

字数: 355 千字

印数: 1-3 000 册

2014 年 5 月第 1 版

2014 年 5 月河北第 1 次印刷



著作权合同登记号 图字: 01-2013-5580 号

定价: 69.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第 0021 号

前言

iPhone 的出现改变了很多事情，往最小的方面说，它改变了移动平台的软件开发方向，实际上这个改变也足够大了。iPhone 的出现甚至还催生出一个“模仿 iPhone”的时代，导致整个数十亿美元的产业萎缩。这一切都是小到可以放在口袋里的 iPhone 所带来的。

关于第 3 版

iOS 5 到 iOS 6 的变化显然要比 iOS 3 到 iOS 5 的变化小多了，而且随着 Xcode 4 逐渐变得成熟，苹果对 Xcode 的更新也变慢了。因此，本书第 3 版的改动比第 2 版的改动也少一些。不管怎么样，你现在拿在手里的这本书使用了苹果为支持 iPhone 5 和 iOS 6 而发布 Xcode 的新版本，除此以外，本书还对 iOS 6 中引入的一些有意思的技术以及相关的改变做了简单地介绍。

第 2 版的改动

本书第 2 版的改动主要针对第 1 版发布以来 iOS 应用开发领域发生的重要变化，这些变化包括：iPad 的发布、Xcode 新版本发布、iOS 系统的两个版本升级以及苹果的 iCloud 云服务的推出等。为了介绍这些基础性质的改变，本书第 2 版对内容进行了大幅翻新。另外，本书中的示例代码全部用 iOS 5 中引入的 ARC 技术并基于 Xcode 4 进行了重写。

谁应该读这本书

本书主要面向的读者是有一定的编程经验的开发者。如果你想为 iPhone 开发应用程序，却是第一次接触 Mac 平台；或者你之前从事 Mac 桌面应用开发，但现在想转向 iOS 应用开发，这本书就非常适合你。本书会快速地介绍为 iPhone、iPod touch 以及 iPad 平台开发应用所必需的基本知识。

阅读本书之前你需要做的准备

本书假设你已经具有 C 语言的基础知识，或者了解基于 C 语言的其他编程语言。另外，如果你有一点面向对象编程的知识就更好了。

你会学到什么

本书会一步步地指导你如何为 iOS 平台开发你的第一款应用，从打开 Xcode 到提交应用到 App Store 并进行审核。你可以在跟着示例进行实际编码的过程中学习到 Objective-C 的基础知识以及 iOS 开发所需要的核心框架。本书也会提供给你一个开发 iOS 应用的基本程序框架。

本书的内容组织

第 1 章，为何选择本地化应用

本章讨论了为什么要选择开发本地化应用而不是基于 Web 的应用，并且对比了开发本地化应用和开发基于 Web 的应用的差别。

第 2 章，成为一名开发者

本章手把手地教你如何注册成为一名 iOS 开发者以及如何搭建开发环境：从安装 Xcode 和 iOS SDK 到生成开发者证书，这个开发者证书是构建应用并部署到 iPhone，iPod touch 以及 iPad 上所必须的。

第 3 章，你的第一个 iOS 应用

本章教你一步步地构建你的第一个 Hello World 应用，使你能尽可能快地上手 iOS 应用开发，本章也会教你如何将你的应用部署到 iPhone，iPod touch 或者 iPad 上并运行。

第 4 章，Objective-C 编程

本章会快速地教你一些关于 Objective-C 语言的基础知识。如果你熟悉其他的基于 C 语言的编程语言（如果还了解面向对象编程就更好了），读完本章后，你就会熟悉 Objective-C 语言和 Cocoa Touch 框架了。

第 5 章，基于表视图应用

UITableView 及其相关的类可能是 iOS 应用中最经常用于构建界面的类了。由于 iPhone 的特点，这些类可以解决大部分的问题，因此，这些类无处不在。在本章中，我们会对表视图相关的类进行深入地探讨。

第 6 章，其他视图控制器

在详细讲解了表视图控制器后，我们在本章中将讲解其他一些在创建 iOS 应用也常用的视图控制器和类，包括：简单的两个视图的应用、基于标签栏的应用、模态视

图控制器以及用于选择视频和图像的视图控制器。

第 7 章，访问网络

本章讨论了如何连接到互联网，通过内嵌网页视图来浏览网页、发送电子邮件以及通过网络来获取数据。

第 8 章，处理数据

本章详细讨论了如何获取用户输入的数据，包括用户的输入以及程序的输入。本章也讨论了如何解析 XML 文档和 JSON 文档，以及如何将数据存储到纯文本文件或 SQLite 数据库中。

第 9 章，使用传感器

本章讨论了如何判定特定硬件是否可用，并且演示了如何从 iOS 设备上配备的传感器获取数据，这些传感器包括：加速度计、磁力计、照相机和 GPS。

第 10 章，地图和定位

本章通过示例讲解了如何在应用中使用 Core Location 框架以及 MapKit 框架。

第 11 章，iCloud 介绍

本章简要介绍了苹果公司新推出的 iCloud 服务，以及如何将其整合到你的应用中。iCloud 服务可以帮助你有多台设备之间同步数据和文档。

第 12 章，与系统应用集成

本章介绍了一些将你的应用和 iOS 软件体系整合在一起的技巧，包括如何在 Settings Bundles 中加入你的应用的设置项；如何使用自定义 Scheme 来启动应用。本章还讲解了如何使用媒体播放器来播放媒体资源以及如何使用通信录。

第 13 章，发布你的应用

本章讲解了如何完成应用发布前的最后的完善工作，并且一步一步地教你构建用于发布用的版本，包括通过 Ad hoc 发布方式以及通过 App Store 发布方式。

第 14 章，未来之路

本章列举了更多更深入学习 iOS 开发所必须了解的要点和资料，包括在本书中提到过的知识点，也包括因为篇幅的原因在本书中未做介绍的知识点。

本书使用的书写约定

本书使用了这些书写约定。

斜体 (*Italic*)

用于表示相关术语、URL 链接、电子邮件地址、文件名以及文件的后缀名

等宽字体 (**Constant width**)

用于表示代码，通常出现在一段文字中，用来表示变量名、函数名、数据库名、数据类型、环境变量、语句或者编程语言的关键字等

等宽粗体 (**Constant width bold**)

用于表示命令或者需要用户自己输入的内容

等宽斜体 (*Constant width italic*)

用于表示需要用户根据当前上下文进行替换的内容



这个图标表示提示、建议或者一般性的注解。



这个图标表示的是提醒或者警告。

使用本书的示例代码

本书的目的是帮你把事情做好。一般来说，你无需特地联系我们就可以在你的程序和文档里任意使用本书的代码，除非你要把其中的关键代码以你的名义重新发布。举例来说，你的程序使用到了书中的几段代码，不需要获得许可；出售或者发布 O'Reilly 的随书光盘，则需要获得许可；引用本书的内容和示例代码去解答一个问题，不需要获得许可；把本书中的大量代码合并到你的产品文档里时，需要获得许可。

我们会感谢，但是不要求你写上代码所属权的声明。这个所有权声明一般包括书名、作者、出版商和 ISBN，比如：“Learning iOS Programming(3rded) by Alasdair Allan. Copyright 2013 Alasdair Allan, 978-1-44935-934-8 (O'Reilly).”

如果你觉得你对示例代码的使用超过了正当适用的范围或者如上所述的授权范围，请跟我们联系：permissions@oreilly.com。

如何联系我们

对于本书的评论或问题请联系出版商：

美国：

O'Reilly Media, Inc.

1005 Gravenstein Highway North

Sebastopol, CA 95472

中国：

北京市西城区西直门南大街 2 号成铭大厦 C 座 807 室 (100035)

奥莱利技术咨询（北京）有限公司

我们为本书提供了一个网页用来提供勘误表、示例以及其他信息，网页地址是：<http://oreil.ly/learn-ios-programming-3e>。您也可以通过 <http://www.learningiosprogramming.com/> 来获取关于本书的补充材料。

如有关于本书的建议或者技术问题，请发送电子邮件到：bookquestions@oreilly.com。

想知道 O'Reilly 更多有关书籍、会议、资源中心和 O'Reilly Network 的更多信息，请登录 O'Reilly 网站：<http://www.oreilly.com>。

在 Facebook 上关注我们：<http://facebook.com/oreilly>

在 Twitter 上关注我们：<http://twitter.com/oreillymedia>

在 YouTube 上关注我们：<http://www.youtube.com/oreillymedia>

Safari 图书在线



Safari 图书在线是一个随需而变的提供专业内容的电子图书馆，包括世界领先的技术和商业出版商出版的图书、视频。众多的专业技术人员、软件开发人员、网页设计师和商业与创意专业人士将 Safari 图书在线作为他们进行研究、解决问题、学习和认证培训的主要资源。

Safari 图书在线为公司组织、政府机构和个人提供了一系列的组合产品以及价格优惠。用户可以通过专业的数据库系统检索来自于 O'Reilly Media, Prentice Hall Professional, Addison-Wesley Professional, Microsoft Press, Sams, Que, Peachpit

Press, Focal Press, Cisco Press, JohnWiley & Sons, Syngress, Morgan Kaufmann, IBM Redbooks, Packt, Adobe Press, FTPress, Apress, Manning, New Riders, McGraw-Hill, Jones & Bartlett, Course Technology 等众多出版商出版的成千上万的书藉、培训视频以及预发布手稿。想获得更多关于 Safari 图书在线的信息, 请访问其网站。

致谢

书不会自己写, 特别是对于一本书的第 3 版, 而且尤其是你的第一本书的第 3 版。修正了第 2 版中你感觉错误的地方之后, 或者至少你自己已经满意之后, 你还必须为了跟上技术的发展要全部再重新修订一遍。这是一个恐怖的场景, 我感觉比坐下来从头开始写一本新书还要难。因此, 我要感谢我的编辑 Brian Jepson 和 Shawn Wallace, 感谢他们一次又一次地催促我提起笔来开始写作。我还要感谢我的妻子 Gemma Hobson, 长久以来她都在背后默默地承受, 我一点儿也不知道她为什么没有阻止我继续写, 但是肯定不是为了那点儿版税。最后, 我要将本书献给我的儿子 Alex, 他现在还小, 虽然从本书的第 1 版出版到现在, 他进步很大, 已经能认些字了, 但仍然完全不懂他的父亲正在做的事情。对于他, 我感觉很抱歉, 在追求事业的过程中没有很多的时间来陪他, 反而对他有点暴躁。

目录

第 1 章 为何选择本地化应用	1
1.1 优点和缺点	1
为何要开发本地化应用?	2
1.2 发行周期	3
1.3 做好了用户自然会来	4
第 2 章 成为一名开发者	6
2.1 注册成为 iOS 开发者	6
2.2 加入 iOS 开发者计划	9
2.3 Mac 开发者计划	10
2.4 安装 iOS SDK	10
关于 Beta 版本	14
2.5 准备你的 iOS 设备	14
2.5.1 创建开发证书	15
2.5.2 获取开发设备的 UDID	17
2.5.3 创建 App ID	18
2.5.4 创建移动描述文件	19
2.5.5 让你的设备支持开发	20
第 3 章 你的第一个 iOS 应用	22
3.1 Objective-C 基础	22
3.1.1 面向对象编程	22
3.1.2 Objective-C 的对象模型	25
3.1.3 Objective-C 的基本语法	26
3.2 创建工程	27
3.2.1 浏览 Xcode 工程	31
3.2.2 Interface Builder	38
3.2.3 设计用户界面	39
3.2.4 关联 UI 控件和代码	41
3.2.5 在模拟器上运行	44

3.2.6	在 iPhone 上运行	46
第 4 章	Objective-C 编程	47
4.1	声明和定义类	47
4.1.1	声明类的接口	47
4.1.2	定义类的实现	49
4.1.3	强类型与弱类型	49
4.1.4	属性	50
4.1.5	合成 (Synthesizing) 属性	51
4.1.6	点句式 (Dot Syntax)	52
4.1.7	方法声明	52
4.1.8	方法调用	53
4.1.9	调用 nil 对象的方法	54
4.2	内存管理	54
4.2.1	创建对象	54
4.2.2	自动回收池 (Autorelease Pool)	55
4.2.3	alloc, retain, copy 和 release 周期	55
4.2.4	自动引用计数	56
4.2.5	dealloc 方法	57
4.2.6	处理内存警告 (Memory Warnings)	57
4.3	常用 iOS 设计模式	58
4.3.1	MVC 模式	58
4.3.2	视图和视图控制器	59
4.3.3	委托和数据源模式	60
4.4	总结	60
第 5 章	基于表视图应用	62
5.1	创建工程	62
5.2	创建表视图	63
运行代码		68
5.3	为表视图填充内容	68
5.4	创建模型	70
5.4.1	添加城市	73
5.4.2	添加图片	75
5.5	关联控制器和模型	77

用 Alert 窗口测试功能	79
5.6 添加导航控制器 (Navigation Control)	80
5.7 添加城市视图	82
5.8 编辑模式	91
5.8.1 删除城市	95
5.8.2 添加城市	96
5.8.3 添加新城市的界面	99
5.8.4 保存城市数据	106
第 6 章 其他视图控制器	111
6.1 工具类应用 (Utility Application)	111
电量监控应用	112
6.2 标签栏应用 (Tab Bar Application)	123
添加标签视图	124
6.3 联合使用多种视图控制器	128
6.4 模态视图控制器	134
修改 City Guide 应用	134
6.5 图片选择视图控制器	143
在 City Guide 应用中加入图片选择器	143
6.6 主从视图 (Master-Detail) 应用	152
创建一个通用 (Universal) 应用	153
6.7 Popover 控制器	159
第 7 章 访问网络	160
7.1 监测网络状态	160
Reachability 类	160
7.2 在应用中嵌入浏览器	173
7.2.1 网页视图控制器	173
7.2.2 显示静态 HTML 文件	182
7.2.3 从 UIWebView 中获取数据	183
7.3 发送邮件	183
7.4 从互联网获取数据	189
7.4.1 同步请求	189
7.4.2 异步请求	190
7.4.3 使用 Web 服务	191

第 8 章 处理数据	207
8.1 数据输入	207
8.1.1 UITextField 及其委托	207
8.1.2 UITextView 及其委托	209
8.2 解析 XML	211
8.2.1 使用 libxml2 解析 XML	212
8.2.2 使用 NSXMLParser 解析 XML	213
8.3 解析 JSON	214
8.3.1 NSJSONSerialization	215
8.3.2 JSON Framework	217
8.3.3 抓取 Twitter 热门话题 (Trending)	218
8.3.4 使用 Social 框架	219
8.3.5 Twitter Trends 应用	220
8.4 正则表达式	228
8.5 存储数据	233
8.5.1 使用纯文本文件保存数据	233
8.5.2 使用 SQL 数据库保存信息	234
8.5.3 Core Data 框架	242
第 9 章 使用传感器	243
9.1 硬件支持	243
9.1.1 网络可用性	243
9.1.2 相机可用性	245
9.1.3 音频输入可用性	245
9.1.4 GPS 可用性	245
9.1.5 磁力计可用性	246
9.2 设置应用的硬件需求	246
9.2.1 WiFi 长连接	248
9.2.2 后台运行模式	248
9.3 iPhone 和 iPad 的不同	249
9.4 使用 Camera	249
9.5 Core Motion 框架	250
9.5.1 轮询运动数据	251
9.5.2 订阅运动数据	252

9.5.3	加速度计	253
9.5.4	陀螺仪	257
9.5.5	磁力计	259
9.6	使用近距离感应器	262
9.7	使用震动器	264
第 10 章	地图和定位	265
10.1	Core Location 框架	265
	设备的朝向	267
10.2	基于位置的天气	269
10.2.1	反向地理编码	270
10.2.2	正向地理编码	270
10.2.3	CLPlacemark 对象	271
10.2.4	完善天气应用	271
10.3	定位与 MapKit	279
10.4	地图标记	286
第 11 章	iCloud 介绍	294
11.1	怎样使用 iCloud?	294
11.1.1	iCloud 备份	294
11.1.2	获取授权描述文件	295
11.2	使用键值对存储	298
11.3	总结	305
第 12 章	与系统应用集成	306
12.1	应用选项	306
12.2	Accounts 框架	317
12.3	Social 框架	318
12.3.1	发送 Twitter 信息	318
12.3.2	发送 Facebook 信息	320
12.4	自定义 URL Scheme	320
12.4.1	使用自定义 URL Scheme	320
12.4.2	注册自定义 URL Scheme	321
12.5	媒体播放	327
12.6	使用地址簿	331

12.6.1	使用交互方式选择联系人	332
12.6.2	使用编程方式选择联系人	336
12.7	发送短信	336
第 13 章	发布你的应用	338
13.1	添加还缺少的功能	338
13.1.1	添加图标	338
13.1.2	添加启动图片	342
13.1.3	修改显示名称	345
13.1.4	启用自动旋转	346
13.2	编译与签名	348
13.2.1	Ad hoc 发布	348
13.2.2	Developer-to-Developer 发布	353
13.2.3	App Store 发布	354
13.3	提交到 App Store	355
13.3.1	为 App Store 发布构建你的应用	360
13.3.2	App Store 资源中心	364
13.4	审核被拒的原因	364
第 14 章	未来之路	367
14.1	Cocoa 和 Objective-C	367
14.2	Web 应用	368
14.3	PhoneGap	368
14.4	Core Data	369
14.5	应用内购买	369
14.6	Core Animation	370
14.7	Game Kit	370
14.8	编写游戏	371
14.9	界面和交互	371
14.10	配件	372

为何选择本地化应用

早在 2007 年发布 iPhone 的时候，苹果并没有一同发布本地化 SDK，苹果公司声称不需要本地化 SDK，鼓励大家使用 JavaScript，CSS 和 HTML 开发 Web 应用。但接下来剧情并没有按照苹果设计的那样发展，也许是苹果的思想太超前，大多数开发者并不吃这一套，他们希望自己开发的应用和 iOS 系统内置的应用一样，可以直接访问硬件并和系统高度集成。

于是在 iPhone 发布后短短几个月，开源社区就完成了一系列苹果原本认为不可能的事情。苹果本来试图通过技术手段把设备封锁起来，可是道高一尺魔高一丈，黑客们还是破门而入了，他们通过逆向工程搞出了“山寨版”的 SDK，进而开发了一整套用于构建本地化应用的开源工具集。据估计，当时市场上存在超过三分之一越狱设备，而越狱目的非常单纯，只是安装未经认证的第三方本地化应用。

不过本书的内容并不是向你介绍这些黑客们搞出来的“山寨”SDK，因为 iPhone 发布不到一年后的 2008 年 3 月，苹果公司就公开表示要改变策略，并且向开发者社区发布了期待已久的官方 SDK。这也许是苹果对于开源社区所做出的努力的回应吧。不过苹果一贯遮遮掩掩，让人捉摸不透，要说这是苹果本来的计划也有可能。到底是怎样，以前只有乔布斯和上帝知道，现在只有上帝知道了。

1.1 优点和缺点

对于苹果发布的 SDK，也不是所有人都买账，在有些人看来这简直是技术的退步，他们认为 Web 应用已经足够好了。因为用 Objective-C 专门针对 iPhone 开发的应用很难移植到其他平台，而 Web 应用只需要开发一套新的 CSS 模板就可以了。

但用户并不同意这样的观点，他们才不管开发起来难度有多大，他们只关心好不好用。不过说起来 Web 应用移植起来也没有想象中那样容易。各个平台的应用外观不尽相同，在做到平台兼容的同时还要实现外观的本地化，其实是一件非常困难的事。如果把 Windows 平台的应用没有经过重新设计直接移植到 Mac 系统上来，就会让人感觉很奇怪，不光是外观，连操作也会和用户的预期不一样。

这里并不是在诋毁 HTML、CSS 和 JavaScript。Web 技术也是很成熟很稳定的，而且作为开发者，这种跨平台的技术框架绝对是有巨大吸引力的。但是对于用户，就不见得如此了。大部分情况下，提出使用 Web 技术或者其他非本地化框架的人往往是从开发者角度来考虑的。这当然不无道理，不过或许你更应该考虑一下用户的感受，他们唯一关心的是可用性而非用什么技术来实现。

用户是很挑剔的，他们拒绝购买你的应用往往只是因为你还支持了其他的平台。这听起来很滑稽，其实很好理解。他们只是希望一个应用的风格看起来和系统里其他应用是一样的。

如果你的应用放在 iOS 生态圈里，更有效地利用硬件资源，并且针对设备优化你的用户界面，这样用户体验会大幅度提高。当然要实现这些目标，非本地化应用大部分也是可以做到的，但这仅仅是理论上可行，实际操作起来困难重重，更别提还要做好几个平台的适配了。

为何要开发本地化应用？

本地化应用最明显的优势在于可以实现 Web 技术无法完成的事情。举一个很典型的例子，第一代现实增强类应用就依赖多种硬件设备，比如 GPS 芯片、加速度传感器、指南针和摄像头等。

尽管 iOS 上的 Safari 浏览器支持 HTML5 提供的地理位置的功能，也还是无法完整地解决问题。鉴于不同平台和不同硬件所带来的差异，HTML5 提供的统一接口会有更大的不确定性。所以你几乎见不到基于 Web 的现实增强类应用。



如果你有 Web 开发的背景，你也许会对跨平台的 PhoneGap 框架感兴趣。这个框架提供了封装了本地化功能的接口，让你可以基于 HTML 和 JavaScript 构建本地化应用，并且可以兼容多个移动平台，iOS 就是其中之一。

其实很多时候，新的技术并不是来帮你实现原本无法实现的功能，而只是带来了更快的速度体验和更好的容错体验。事实上苹果的 iTunes 和 App Store 这两款 iOS 应用和 PhoneGap 应用就很类似，它们本质上也是 Web 应用。就像 Mac 桌面版的 iTunes