

中文版累计销量逾60000册

Swift和Objective-C双语版

全球数百万开发者交口称赞的iOS开发圣经



# 精通iOS开发

## (第7版)

【美】David Mark 【瑞典】Jack Nutting 【英】Kim Topley 著  
【法】Fredrik Olsson 【美】Jeff LaMarche  
周庆成 邓强 武海峰 蒋振华 译



中国工信出版集团



人民邮电出版社  
POSTS & TELECOM PRESS

## 作者简介

### David Mark

经验丰富的Mac技术开发专家，技术图书作者，撰写过许多Mac平台和iOS开发方面的畅销书。David还是iOS/Android开发公司MartianCraft的联合创始人之一。

### Jack Nutting

资深苹果移动开发技术专家，首批Cocoa用户，技术图书作者。Jack使用Cocoa为许多行业开发过各种应用程序。  
个人博客：[www.nuthole.com](http://www.nuthole.com)。

### Kim Topley

技术图书作者，具有丰富的iOS开发经验。Kim现任StreamingEdge公司高级软件工程师，拥有30多年从业经历。

### Fredrik Olsson

具有丰富的Cocoa和iPhone开发经验，从业经历丰富，从实时汇编到企业版Java开发。Objective-C和Cocoa框架的坚定拥护者。

### Jeff LaMarche

资深Mac与iOS开发人员，拥有20多年开发经验。Jeff是MacTech Magazine和苹果公司开发人员网站的专栏作家。Jeff关于iOS开发的博文广受欢迎，地址为[www.iphonedevdevelopment.blogspot.com](http://www.iphonedevdevelopment.blogspot.com)。



# 数字版权声明

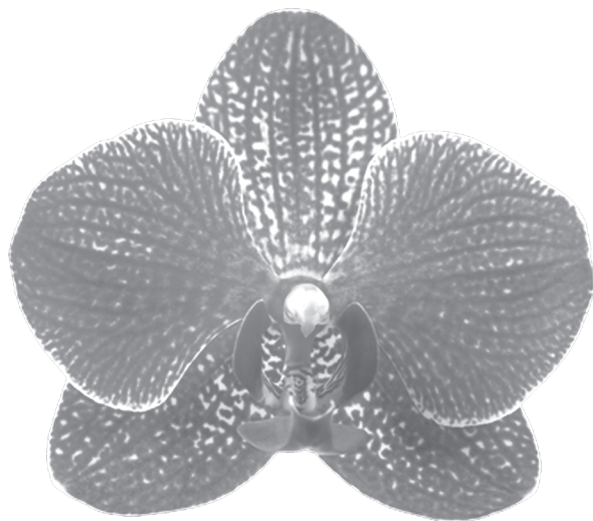
图灵社区的电子书没有采用专有客户端，您可以在任意设备上，用自己喜欢的浏览器和PDF阅读器进行阅读。

但您购买的电子书仅供您个人使用，未经授权，不得进行传播。

我们愿意相信读者具有这样的良知和觉悟，与我们共同保护知识产权。

如果购买者有侵权行为，我们可能对该用户实施包括但不限于关闭该帐号等维权措施，并可能追究法律责任。

**TURING** 图灵程序设计丛书



# 精通iOS开发

## (第7版)

【美】David Mark 【瑞典】Jack Nutting 【英】Kim Topley 著  
【法】Fredrik Olsson 【美】Jeff LaMarche  
周庆成 邓强 武海峰 蒋振华 译

人民邮电出版社  
北京



## 图书在版编目 (C I P) 数据

精通iOS开发 : 第7版 / (美) 马克 (Mark, D.) 等著;  
周庆成等译. -- 北京 : 人民邮电出版社, 2015. 9  
(图灵程序设计丛书)  
ISBN 978-7-115-40111-3

I. ①精… II. ①马… ②周… III. ①移动终端—应  
用程序—程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字 (2015) 第180911号

## 内 容 提 要

本书是 iOS 应用开发基础教程, 内容翔实, 语言生动。作者结合大量实例, 使用 Swift 和 Objective-C 两种语言循序渐进地讲解了适用于 iPhone/iPad 开发的基本流程。新版介绍强大的 iOS 8 操作系统, 涵盖 Xcode 6 的新功能, 书中所有案例全部重新编写。

本书具有较强通用性, Swift 语言的 iOS 发新手可通过本书快速入门进阶, 经验丰富的 iOS 开发人员也能从中找到令人耳目一新的内容。

- 
- ◆ 著 [美] David Mark [瑞典] Jack Nutting  
[英] Kim Topley [法] Fredrik Olsson  
[美] Jeff LaMarche  
译 周庆成 邓 强 武海峰 蒋振华  
责任编辑 朱 巍  
执行编辑 杨 琳  
责任印制 杨林杰
- ◆ 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>  
北京 印刷
- ◆ 开本: 880×1230 1/16  
印张: 36.5  
字数: 1231千字 2015年9月第1版  
印数: 1-4 000册 2015年9月北京第1次印刷  
著作权合同登记号 图字: 01-2015-1057号
- 

定价: 118.00元

读者服务热线: (010)51095186转600 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

# 版权声明

Original English language edition, entitled *Beginning iPhone Development with Swift: Exploring the iOS SDK*, *Beginning iPhone Development: Exploring the iOS SDK (Seventh Edition)* by David Mark, Jack Nutting, Kim Topley, Fredrik Olsson, and Jeff LaMarche, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2014 by David Mark, Jack Nutting, Kim Topley, Fredrik Olsson, and Jeff LaMarche. Simplified Chinese-language edition copyright © 2015 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由Apress L.P.授权人民邮电出版社独家出版。未经出版者书面许可，不得以任何方式复制或抄袭本书内容。

版权所有，侵权必究。

# 译者序

在 2014 年的 WWDC (WorldWide Developers Conference, 苹果全球开发者大会) 上, 苹果公司发布了新的 OS X 操作系统 Yosemite 和移动操作系统 iOS 8。iOS 8 系统是 iOS 7 的升级版, 延续了扁平化界面设计风格并增强了系统的稳定性。此外, iOS 8 系统还开放了更多的 API 接口, 以增强未来应用程序的功能并减少开发人员的工作量。

在此次开发者大会中, 还有一件尤为引人注目的事情: 苹果推出了全新的开发语言 Swift, 它可以与 Objective-C 共同运行于 OS X 和 iOS 平台, 用于搭建基于苹果平台的应用程序。

在此之前, 苹果平台推荐的开发语言一直是 Objective-C。它是基于 C 语言的扩展, 由苹果公司负责维护的面向对象语言, 出现时间比 C++ 还要早。随着 OS X 系统与 iOS 平台的不断发展, 之后越来越多的移动开发者开始学习这门语言, 因而其市场份额不断增大, 排名也一度超越 C++、C#、Javascript 等主流语言。在 2011 年与 2012 年, Objective-C 凭借快速上升的使用率连续两次赢得了 TIOBE 的年度编程语言大奖。

如今苹果公司又推出了 Swift 语言, 必然有其重大意义。尽管 Objective-C 是一种非常神奇的语言, 甚至比互联网的年龄还要长, 不过这个已为苹果贡献多时的编程语言, 想要上手却并不容易, 对那些新人程序员来说更是如此。初学者难以掌握它的主要原因在于, Objective-C 的语法风格独特, 看起来和其他的 C 系语言大相径庭。

为了方便在苹果平台上的学习, 降低开发应用的负担, 苹果公司推出了全新的 Swift 语言。Swift 语言的主要目标是简单易学、便捷高效, 其执行速度比 Python 和 Objective-C 程序更快。Swift 还与 Xcode 6 的 Playground 功能配合, 可以像脚本语言一样, 具有实时预览功能。开发者在输入代码后, 右边屏幕会实时显示代码效果。此外, 开发者还可以看到应用程序运行过程中的代码效果, 从而使测试过程更加方便。

Swift 语言一经推出, 便立刻引起了全世界开发者的关注。Swift 汲取了许多语言的特点并加以改进。对于编程老手来说, 适应这种新的语言非常容易; 而对于新人来说, Swift 的风格更加友好。开发者问答网站 Stack Overflow 的调查显示, 苹果公司的 Swift 语言以极大的优势成为了最受开发者欢迎的编程语言。

然而苹果公司对于 Swift 语言的未来发展仍有很高的期待。仅仅一年之后, 在 2015 年的 WWDC 上, 苹果公司又给了所有人一个惊喜: 发布升级版的 Swift 2, 并宣布 Swift 语言开源, 同时兼容 iOS、OS X 和 Linux 操作系统。苹果公司向广大开发者伸出了橄榄枝, 这一重要举措引起了业界的一致好评。它意味着开发者将更容易掌握这门强大的编程语言, 并且拓展了更广泛的实用方向。在未来, 将会有更多的人去接受这一语言, 全球开发者也将会越来越重视苹果平台上的应用开发。

本书中的内容讲解和示例代码同时提供了 Objective-C 和 Swift 语言两种版本<sup>①</sup>, 并针对各自的不同进行了详细的解释, 其价值远胜于同时阅读两本书。读者可以根据自身情况并比较两种语言的特点, 选择更加擅长的方向。通过书中层层递进的讲解, 通过一个个富有代表性的示例应用, 读者能够逐步掌握 iOS 的软件开发。无论你倾向于经典成熟的 Objective-C 还是灵巧轻便的 Swift, 它们都将能够帮助你开发出同样优秀的 iOS 应用程序。

本书的原作者是几位经验极为丰富的开发人员, 从 iPhone OS 2.0 时代就开始编写苹果平台开发的书籍, 到现在已经出版了 iOS 系列书籍的第 7 版。这一系列的书籍在国内外的销量一直长盛不衰, 有越来越多的程序员依靠其步入了 iOS 开发的殿堂。当前这一版的内容极为丰富, 全面涵盖了 iOS 系统的许多功能, 并进行了深入的探索。如此庞大厚重的一本书堪称 iOS 开发学习的圣经。这本书适用于所有对设计 iOS 应用程序感兴趣的人, 无

---

<sup>①</sup> 本书中的代码段部分, 左侧统一放 Swift 代码, 右侧统一放 Objective-C 代码。——编者注



论你拥有多年的开发经验还是第一次学习编程，都可以藉由此书领略到 iOS 的魅力。希望你能够通过本书的学习开启通往熟练驾驭 iOS 应用设计的旅程。

在翻译本书的过程中遇到了许多困难，在此要感谢所有给予过我帮助的人。感谢我的家人给予我精神上的支持，感谢那些替我分担了压力的朋友，也要感谢图灵公司参与了本书编辑与校对等工作的每个人，正是由于你们的努力才能保证它的质量。尽管如此，书中依然难免有疏漏之处，希望读者能够包涵并向我们提出宝贵的建议，也希望你们的应用能够早日登上 App Store。

周庆成

# 目 录

第 1 章 欢迎来到 iOS 和 Swift 世界 .....	1	3.3 视图控制器 .....	34
1.1 关于本书 .....	1	3.3.1 输出接口和操作方法 .....	35
1.2 必备条件 .....	1	3.3.2 精简视图控制器代码 .....	37
1.2.1 选择开发者计划 .....	2	3.3.3 设计用户界面 .....	38
1.2.2 必备知识 .....	3	3.3.4 运行应用 .....	47
1.3 iOS 应用程序的特点 .....	4	3.3.5 布局预览 .....	52
1.3.1 只能运行一个应用 .....	4	3.3.6 添加字体类型 .....	54
1.3.2 只能使用一个窗口 .....	4	3.4 应用程序委托 .....	55
1.3.3 有限的访问权限 .....	4	3.5 小结 .....	58
1.3.4 有限的响应时间 .....	4	第 4 章 更丰富的用户界面 .....	59
1.3.5 有限的屏幕尺寸 .....	5	4.1 满是控件的屏幕 .....	59
1.3.6 有限的系统资源 .....	5	4.2 动态控件、静态控件和被动控件 .....	61
1.3.7 不支持垃圾回收 .....	6	4.3 创建应用程序 .....	62
1.3.8 一些新功能 .....	6	4.4 实现图像视图和文本框 .....	62
1.3.9 与众不同的交互方法 .....	6	4.4.1 添加图像视图 .....	62
1.4 本书内容 .....	6	4.4.2 调整图像视图的大小 .....	64
1.5 新版增加内容 .....	8	4.4.3 设置视图属性 .....	65
1.6 Swift 和 Xcode 版本 .....	8	4.4.4 添加文本框 .....	67
1.7 准备好了吗 .....	8	4.4.5 添加约束 .....	71
第 2 章 创建第一个项目 .....	9	4.4.6 创建并关联输出接口 .....	72
2.1 在 Xcode 中创建项目 .....	9	4.5 关闭键盘 .....	74
2.1.1 Xcode 项目窗口 .....	12	4.5.1 按下 Done 按钮关闭键盘 .....	74
2.1.2 深入了解项目 .....	18	4.5.2 触摸背景关闭键盘 .....	75
2.2 Interface Builder 简介 .....	19	4.5.3 添加滑动条和标签 .....	76
2.2.1 文件格式 .....	20	4.5.4 添加更多约束 .....	78
2.2.2 storyboard .....	21	4.5.5 创建并关联操作方法和输出接口 .....	78
2.2.3 资源库 .....	22	4.5.6 实现操作方法 .....	78
2.2.4 在视图中添加标签 .....	23	4.6 实现开关、按钮和分段控件 .....	79
2.2.5 属性修改 .....	27	4.6.1 添加两个带标签的开关 .....	80
2.3 画龙点睛——美化 iPhone 应用程序 .....	28	4.6.2 为开关创建并关联输出接口和操作方法 .....	81
2.4 启动界面 .....	30	4.6.3 实现开关的操作方法 .....	81
2.5 小结 .....	31	4.7 美化按钮 .....	83
第 3 章 实现基本交互 .....	32	4.7.1 可拉伸图像 .....	83
3.1 MVC 模式 .....	32	4.7.2 控件状态 .....	84
3.2 创建项目 .....	33		

4.7.3 为按钮创建并关联输出接口和操作 方法	84	7.2 委托和数据源	139
4.8 实现分段控件的操作方法	85	7.3 创建 Pickers 应用程序	139
4.9 实现操作表单和警告视图	85	7.3.1 创建视图控制器	140
4.9.1 显示操作表单	86	7.3.2 创建分页栏控制器	140
4.9.2 显示警告视图	88	7.3.3 首次运行	143
4.10 小结	90	7.4 实现日期选取器	143
第 5 章 自动旋转和自动调整大小	91	7.5 实现单滚轮选取器	146
5.1 自动旋转机制	91	7.5.1 构建视图	146
5.1.1 点、像素和 Retina 显示屏	92	7.5.2 将控制器实现为数据源和委托	147
5.1.2 控制旋转	92	7.6 实现多滚轮选取器	150
5.2 选择视图支持的方向	93	7.6.1 声明输出接口和操作方法	150
5.2.1 应用级支持的方向	93	7.6.2 构建视图	150
5.2.2 独立控制器的旋转支持	94	7.6.3 实现控制器	151
5.3 使用约束设计界面	95	7.7 滚轮内容根据环境变化	153
5.3.1 覆盖默认的约束	98	7.8 使用自定义选取器创建一个简单游戏	159
5.3.2 与屏幕等宽的标签	99	7.8.1 编写控制器头文件	159
5.4 创建自适应布局	101	7.8.2 构建视图	159
5.4.1 创建 Restructure 应用程序	101	7.8.3 实现控制器	160
5.4.2 尺寸分类	105	7.8.4 最后的细节	163
5.4.3 尺寸分类和 storyboard	106	7.9 小结	165
5.4.4 创建 iPhone 横屏布局	108	第 8 章 表视图简介	166
5.4.5 创建 iPad 布局	112	8.1 表视图基础	166
5.5 小结	114	8.1.1 表视图和表视图单元	167
第 6 章 多视图应用	115	8.1.2 分组表和无格式表	167
6.1 多视图应用的常见类型	115	8.2 实现一个简单表	168
6.2 多视图应用的体系结构	118	8.2.1 设计视图	168
6.2.1 根控制器	119	8.2.2 编写控制器	170
6.2.2 内容视图剖析	119	8.2.3 添加一个图像	174
6.3 构建 View Switcher 项目	120	8.2.4 表视图单元样式	175
6.3.1 重命名视图控制器	120	8.2.5 设置缩进级别	177
6.3.2 添加内容视图控制器	123	8.2.6 处理行的选择	178
6.3.3 修改 SwitchingViewController 实现 文件	123	8.2.7 更改字体大小和行高	179
6.3.4 创建拥有工具栏的视图	124	8.3 定制表视图单元	181
6.3.5 连接工具栏按钮和视图控制器	126	8.3.1 向表视图单元添加子视图	181
6.3.6 编写根视图控制器	127	8.3.2 创建 UITableViewCell 子类	182
6.3.7 实现内容视图	130	8.3.3 从 nib 文件加载 UITableViewCell	186
6.3.8 转场过程的动画效果	133	8.4 分组分区和索引分区	190
6.4 小结	135	8.4.1 构建视图	190
第 7 章 分页栏与选取器	136	8.4.2 导入数据	191
7.1 Pickers 应用程序	136	8.4.3 实现控制器	191
		8.4.4 添加索引	195
		8.5 实现搜索栏	195
		8.6 视图调试器	202
		8.7 小结	204



第 9 章 导航控制器和表视图 .....	205	11.4 iPhone 上的分割视图 .....	266
9.1 导航控制器 .....	205	11.4.1 iPhone 6 Plus 上的分割视图 .....	267
9.1.1 栈的概念 .....	205	11.4.2 在所有 iPhone 上支持 iPhone 6 Plus 的行为 .....	268
9.1.2 控制器栈 .....	206	11.5 自定义分割视图 .....	271
9.2 简单的字体浏览器: Fonts .....	207	11.6 小结 .....	273
9.2.1 子控制器 .....	207	第 12 章 应用设置和用户默认设置 .....	274
9.2.2 Font 应用的基础框架 .....	209	12.1 设置捆绑包入门 .....	274
9.3 创建根视图控制器 .....	214	12.2 应用: Bridge Control .....	275
9.4 初始化 storyboard .....	217	12.2.1 创建项目 .....	278
9.5 第一个子控制器: 字体列表视图 .....	218	12.2.2 使用设置捆绑包 .....	279
9.5.1 设定字体列表的 storyboard .....	219	12.2.3 读取应用中的设置 .....	291
9.5.2 对根视图控制器的转场进行设置 .....	220	12.2.4 在应用中修改默认设置 .....	294
9.6 创建字体尺寸视图控制器 .....	221	12.2.5 注册默认值 .....	296
9.6.1 设计字体尺寸视图控制器的 storyboard .....	223	12.2.6 保证设置有效 .....	297
9.6.2 对字体列表视图控制器的转场进行 设置 .....	223	12.3 切换到“设置”应用程序 .....	299
9.7 创建字体信息视图控制器 .....	223	12.4 小结 .....	300
9.7.1 设计字体信息视图控制器的 storyboard .....	225	第 13 章 数据持久化基础知识 .....	301
9.7.2 设置约束 .....	226	13.1 应用的沙盒 .....	301
9.7.3 调整字体列表视图控制器的转场 .....	227	13.1.1 获取 Documents 目录 .....	303
9.7.4 我的收藏字体 .....	227	13.1.2 获取 tmp 目录 .....	304
9.8 改善表视图 .....	228	13.2 文件保存方案 .....	304
9.8.1 实现轻扫删除 .....	228	13.2.1 单文件持久化 .....	304
9.8.2 实现拖动排序 .....	230	13.2.2 多文件持久化 .....	304
9.9 小结 .....	232	13.3 属性列表 .....	304
第 10 章 集合视图 .....	233	13.3.1 属性列表序列化 .....	305
10.1 创建 DialogViewer 项目 .....	233	13.3.2 Persistence 应用的第一个版本 .....	306
10.2 修补视图控制器类 .....	234	13.4 对模型对象进行归档 .....	310
10.3 自定义单元 .....	234	13.4.1 遵循 NSCoding 协议 .....	310
10.4 配置视图控制器 .....	237	13.4.2 实现 NSCopying 协议 .....	312
10.5 内容单元 .....	238	13.4.3 对数据对象进行归档和取消归档 .....	313
10.6 实现流式布局 .....	239	13.4.4 归档应用 .....	314
10.7 分区标题视图 .....	241	13.5 使用 iOS 内嵌的 SQLite3 .....	317
10.8 小结 .....	242	13.5.1 创建或打开数据库 .....	317
第 11 章 分割视图和浮动窗口 .....	243	13.5.2 绑定变量 .....	319
11.1 创建分割视图项目 .....	245	13.5.3 SQLite3 应用 .....	319
11.1.1 在 storyboard 中定义结构 .....	247	13.6 使用 Core Data .....	324
11.1.2 使用代码定义功能 .....	248	13.6.1 实体和托管对象 .....	326
11.1.3 Master-Detail 模板应用程序的工作 原理 .....	253	13.6.2 Core Data 应用 .....	328
11.2 显示总统信息 .....	255	13.7 小结 .....	335
11.3 创建浮动窗口 .....	261	第 14 章 文档和 iCloud .....	336
		14.1 使用 UIDocument 管理文档存储 .....	336
		14.1.1 构建 TinyPix .....	336

14.1.2 创建 TinyPixDocument 类	337	第 17 章 Sprite Kit 游戏框架	411
14.1.3 主控制器代码	340	17.1 基础入门	411
14.1.4 初始 storyboard	346	17.2 玩家动作	420
14.1.5 创建 TinyPixView 类	348	17.2.1 向场景中插入玩家	421
14.1.6 设计 storyboard	352	17.2.2 触摸处理: 玩家移动	421
14.2 添加 iCloud 支持	356	17.2.3 几何运算	422
14.2.1 创建授权文件	356	17.2.4 轻微摆动	423
14.2.2 如何查询	357	17.3 创建敌人	424
14.2.3 保存在哪里	359	17.4 在场景中放入敌人	425
14.2.4 将首选项保存到 iCloud	360	17.5 开始射击	426
14.3 小结	362	17.5.1 定义物理类别	426
第 15 章 Grand Central Dispatch 和后台处理	363	17.5.2 创建 BulletNode 类	427
15.1 Grand Central Dispatch	363	17.5.3 应用物理知识	428
15.2 SlowWorker 简介	363	17.5.4 在场景中添加导弹	429
15.3 线程基础知识	367	17.6 利用物理引擎攻击敌人	430
15.4 工作单元	367	17.7 实现关卡	431
15.5 GCD: 底层队列	368	17.7.1 注意敌人	432
15.5.1 傻瓜式操作	368	17.7.2 进入下一关卡	432
15.5.2 改进 SlowWorker	369	17.8 自定义碰撞	433
15.6 后台处理	374	17.8.1 为 SKNode 添加类扩展/类别	434
15.6.1 应用生命周期	375	17.8.2 向敌人添加自定义碰撞行为	436
15.6.2 状态更改通知	375	17.8.3 显示玩家正确的生命值	436
15.6.3 创建 State Lab 项目	376	17.9 粒子系统	438
15.6.4 执行状态的变化	377	17.9.1 第一个粒子	438
15.6.5 利用执行状态更改	379	17.9.2 向场景中加入粒子	440
15.6.6 处理不活跃状态	379	17.10 游戏结束	441
15.6.7 处理后台状态	382	17.11 创建开始场景	443
15.7 小结	388	17.12 播放音乐	445
第 16 章 使用 Core Graphics 绘图	389	17.13 添加力场	446
16.1 Quartz 2D 基础概念	389	17.14 小结	449
16.2 Quartz 2D 绘图方法	389	第 18 章 轻点、触摸和手势	450
16.2.1 Quartz 2D 图形环境	389	18.1 多点触控术语	450
16.2.2 坐标系统	390	18.2 响应者链	451
16.2.3 指定颜色	391	18.2.1 响应事件	451
16.2.4 在环境中绘制图像	393	18.2.2 转发事件: 保持响应者链的活动状态	452
16.2.5 绘制形状: 多边形、直线和曲线	393	18.3 多点触控体系结构	452
16.2.6 Quartz 2D 样例: 图案、渐变色、虚线图	394	18.4 4 个手势通知方法	452
16.3 QuartzFun 应用程序	395	18.5 TouchExplorer 应用	453
16.3.1 构建 QuartzFun 应用程序	395	18.6 Swipes 应用	456
16.3.2 添加 Quartz 2D 绘制代码	403	18.6.1 使用触摸事件检测轻扫	457
16.3.3 优化 QuartzFun 应用程序	408	18.6.2 自动手势识别	459
16.4 小结	410	18.6.3 实现多指轻扫	460
		18.7 检测多次轻点	462

18.8 检测捏合和旋转	466	20.4.2 摇动与击碎	498
18.9 自定义手势	469	20.5 将加速计用作方向控制器	500
18.9.1 CheckPlease 应用	470	20.5.1 滚弹珠程序	501
18.9.2 CheckPlease 的触摸方法	471	20.5.2 实现 BallView 类	503
18.10 小结	473	20.5.3 计算弹珠运动	506
第 19 章 Core Location 和 Map Kit	474	20.6 小结	508
19.1 位置管理器	474	第 21 章 摄像头和照片库	509
19.1.1 设置精度	475	21.1 图像选取器和 UIImagePickerController	509
19.1.2 设置距离筛选器	475	21.2 图像选取器控制器	509
19.1.3 获取使用定位服务的权限	475	21.3 实际测试摄像头和照片库	511
19.1.4 启动位置管理器	475	21.3.1 设计界面	512
19.1.5 合理使用位置管理器	476	21.3.2 实现摄像头视图控制器	515
19.2 位置管理器委托	476	21.4 小结	518
19.2.1 获取位置更新	476	第 22 章 应用本地化	519
19.2.2 使用 CLLocation 获取纬度和经度	476	22.1 本地化体系结构	519
19.2.3 错误通知	478	22.2 字符串文件	520
19.3 开始构建 Core Location	478	22.2.1 字符串文件的格式	520
19.4 将移动路线展现在地图上	484	22.2.2 本地化的字符串函数	521
19.5 小结	488	22.3 现实中的 iOS 本地化应用	521
第 20 章 陀螺仪和加速计	489	22.3.1 创建 LocalizeMe	522
20.1 加速计物理特性	489	22.3.2 测试 LocalizeMe	525
20.2 陀螺仪旋转特性	489	22.3.3 本地化项目	525
20.3 Core Motion 和动作管理器	490	22.3.4 本地化 storyboard	527
20.3.1 基于事件的动作	490	22.3.5 创建并本地化字符串文件	529
20.3.2 主动动作访问	494	22.4 在 Xcode 中预览本地化	533
20.3.3 陀螺仪和方位结果	496	22.5 添加其他本地化	534
20.3.4 加速计结果	497	22.6 小结	535
20.4 检测摇动	497	附录 Swift 简介	536
20.4.1 内嵌的摇动检测	498		





你想开发iPhone、iPod touch和iPad应用程序？关于这一点我们完全理解。这些设备的核心系统iOS是一个令人赞叹的平台，自从2007年面世以来发展势头不减。移动应用平台的兴起意味着人们不论走到哪里都可以使用应用软件。随着iOS 8、Xcode 6和最新版本iOS SDK（Software Development Kit，软件开发工具集）的发布，一切都变得更好、更加有趣。

## 1.1 关于本书

本书是一本入门指导，可以帮助你开发出自己的iOS应用，旨在帮助你入门，理解iOS应用程序的运行和构建方式。

在学习的过程中，你将会创建一些简单的应用程序。每个应用程序都会涵盖某些iOS特性，并展示如何使用这些特性。如果你扎实地掌握了书中的基础知识，加上自己的创造力和恒心，再借助条理清晰的苹果公司官方文档来扩大知识面，就可以创建出专业的iPhone和iPad应用。

---

**提示** Dave、Jack、Jeff和Fredrik为本书创办了一个论坛：<http://forum.learncocoa.org>。可以在这里遇到志趣相投的伙伴，相互答疑解惑。请务必来看看！

---

## 1.2 必备条件

开始编写iOS应用程序之前，需要做一些准备工作。初学者需要一台基于Intel架构的Macintosh计算机，并安装有Mavericks（OS X 10.9）或Yosemite（OS X 10.10）或更高的版本。任何最近上市的基于Intel架构的Macintosh计算机（台式机或笔记本）均可。

如果想使用苹果公司最强大的全新开发工具，需要注册成为iOS开发者。只需访问<http://developer.apple.com/ios/>就可以创建自己的开发者账号。你会看到如图1-1所示的页面。

首先，点击Log In按钮，页面将提示你输入Apple ID。如果没有Apple ID，就点击Register创建，然后登录。登录之后会进入iOS开发中心的主页面，其中有各类文档、视频和示例代码等的链接。所有这些都可以帮助你更好地进行iOS应用开发。

iOS开发中最重要的工具是Xcode，它是苹果公司的IDE（Integrated Development Environment，集成开发环境）。Xcode提供了一些实用工具，用于创建和调试源代码，编译应用程序，以及对应用程序进行性能调优。

可以通过Mac的苹果图标菜单访问Mac App Store，从中下载Xcode。

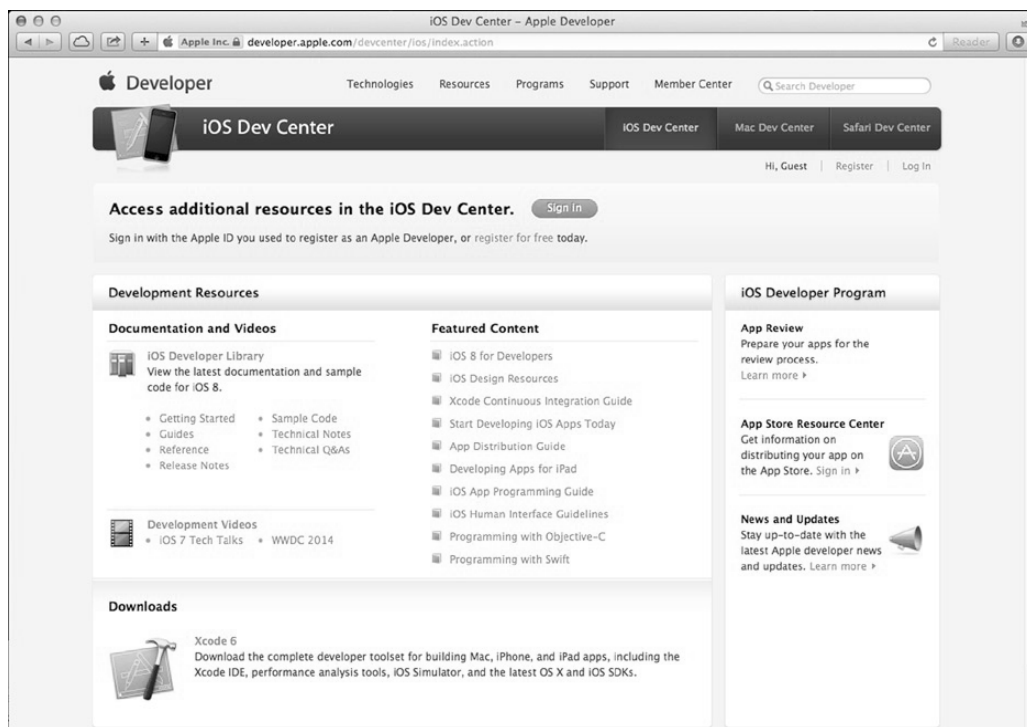


图1-1 苹果公司的iOS开发中心网站

### 本书采用的SDK版本以及源代码示例

随着版本的不断更新，SDK和Xcode的下载方式也在发生变化。在最近几年内，苹果已经开始将当前较“稳定”版本的Xcode和iOS SDK放在Mac App Store中，同时在开发者网站上提供预览版供开发者下载。总之，如果想下载Xcode和iOS SDK的最新版本（即非beta测试版），应该使用Mac App Store。

本书面向当前最新版的Xcode和SDK。我们会在某些地方使用iOS 8中新引入的函数和方法，它们可能会与旧版SDK不兼容。当然，我们会在相应的地方特别指出。

请务必从<http://apress.com>下载最新的源代码文档<sup>①</sup>。每当有新版SDK发布时，我们会及时更新代码，你可以时常过来看看是否有变化。

## 1.2.1 选择开发者计划

免费下载的Xcode中包含一个模拟器，通过这个模拟器，就可以在Mac上创建并运行iPhone和iPad应用。这对于学习编写iOS程序极有帮助。不过，模拟器不支持那些需要依赖硬件的特性，比如加速计和摄像头。此外，免费下载的SDK不支持把应用程序安装到iPhone或其他真机设备中，也不能在App Store上出售应用。如果想获取这些功能，需要从以下两个付费的开发者计划中选择一个。

- ❑ 标准版计划的价格为每年99美元。它提供了大量的开发工具和资源，以及技术支持，还可以通过苹果公司的App Store出售应用。最重要的是，它允许在iOS设备上（而不只是在模拟器上）测试并调试代码。
- ❑ 企业版计划的价格为每年299美元。如果一些公司需要开发只在企业内部使用的iOS应用程序，可以选择这个计划。

<sup>①</sup> 本书源代码也可以从图灵社区本书页面“随书下载”部分下载，网址是<http://www.it-ebooks.com.cn/book/1526>。——编者注

想要了解关于这些计划的详细信息，可以访问<http://developer.apple.com/programs/ios>（标准版计划）和<http://developer.apple.com/programs/ios/enterprise>（企业版计划）来查看并比较。

由于iOS支持的移动设备（比如iPhone）经常连接到其他公司的无线网络设施，所以苹果对iOS开发人员的限制比Mac开发人员严格得多（至少目前Mac开发人员完全不需要经过苹果的审查和批准就可以编写并发布程序）。虽然iPod touch和仅支持Wi-Fi的iPad不使用其他公司的无线设施，但是它们也受到同样的限制。

苹果之所以添加这些限制，主要是为了尽量避免发布恶意程序和蹩脚程序，因为这类程序可能会在共享网络中降低性能。开发iOS应用似乎有很高的门槛，不过苹果已经为简化开发过程作出了巨大的努力。值得一提的是，99美元比任何付费版本Visual Studio（微软公司的软件开发IDE）的价格低得多。

不用说，你还需要一部iPhone、iPod touch或iPad。虽然很多代码都可以通过iOS模拟器进行测试，但并非所有程序都是如此。即便是那些可以在模拟器上运行的应用程序，也需要在真实设备上进行全面测试后才能对外发布。

---

**注意** 如果你准备注册标准版计划或企业版计划，最好马上申请。审批过程可能需要一些时间，通过审批之后才能在真机设备上运行应用程序。不过不必担心，前几章中的所有项目以及本书的大多数应用程序都可以在iOS模拟器上运行。

---

## 1.2.2 必备知识

学习本书应该具备一定的编程知识。我们认为你已经理解了基本编程，尤其是面向对象编程的基础知识（例如，类、对象、循环、变量这些基本概念）。如果你想用Objective-C进行开发的话，应该熟悉Objective-C编程语言。Cocoa Touch（本书大部分章节都需要用到Cocoa Touch，它是SDK的一部分）采用的是最新版的Objective-C，有一些新特性。不过不了解也没有关系，本书在用到Objective-C的新特性时会特别指出，并解释其工作原理和使用目的。你也可以使用Swift来开发，不过你也许还不熟悉苹果公司的新编程语言Swift。本书的附录会介绍Swift和Xcode中全新的Playground（游乐园）特性，它可以简单地展示出新语言的特色。如果还想了解更多关于Swift的内容，可以通过以下方式来获取。

- ❑ *The Swift Programming Language*是苹果公司官方的Swift指导手册。你可以从iBooks store或从iOS开发者网站[https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift\\_Programming\\_Language/index.html](https://developer.apple.com/library/ios/documentation/Swift/Conceptual/Swift_Programming_Language/index.html)下载。
- ❑ Matthew Campbell撰写的*Swift Quick Syntax Reference*（Apress，2014）是一本Swift的代码和语法手册。浏览[www.apress.com/9781484204405](http://www.apress.com/9781484204405)可以获取更多详细信息。
- ❑ 如果你之前使用过Objective-C，可以通过阅读Scott Gardner撰写的*Transitioning to Swift*（Apress，2014）过渡到新语言。
- ❑ 假如你是一位Android开发者，想了解新世界的情况，Sean Liao撰写的*Migrating to Swift from Android*（Apress，2014）会很有帮助。浏览[www.apress.com/9781484204375](http://www.apress.com/9781484204375)可以了解更多详细信息。

你还应该像用户一样熟悉iOS系统本身。由于要基于各种平台来编写应用程序，你需要熟悉iPhone、iPad和iPod touch的一些细微差别和特有功能。花点时间去熟悉iOS界面以及iPhone/iPad应用的风格吧。

### Objective-C的学习资源

如果你从未使用Objective-C编写过程序，下列资源可以作为入门读物。

《Objective-C基础教程（第2版）》<sup>①</sup>一书浅显易懂，是非常优秀的Objective-C入门读物，作者是Mac编程专

---

<sup>①</sup> 中文版《Objective-C基础教程（第2版）》已由人民邮电出版社图灵公司出版，ISBN：978-7-115-31458-1。——编者注

家Scott Knaster、Waqar Malik和Mark Dalrymple。你可以通过[www.apress.com/book/view/9781430241881](http://www.apress.com/book/view/9781430241881)了解到更多信息。

*Programming with Objective-C*是苹果官方的Objective-C语言入门书。你可以在<https://developer.apple.com/library/mac/documentation/cocoa/conceptual/ProgrammingWithObjectiveC>上找到更多的内容。

---

## 1.3 iOS 应用程序的特点

如果从未使用Cocoa（或者它的前辈NeXTSTEP和OpenStep）写过程序，你可能会发现Cocoa Touch（用于编写iOS应用程序的应用程序框架）比较新奇。它与其他常用应用程序框架（比如用于构建.NET或Java应用程序的框架）之间存在一些根本差异。起初你可能有点不得要领，不过不必担心，只要多加练习，很快就能够对Cocoa Touch运用自如了。

如果曾经使用Cocoa或NeXTSTEP写过程序，你会发现iOS SDK中有许多熟悉的身影。有很多类都是从Mac OS X版本的Cocoa中原样移植过来的。即便是那些不同的类，也遵循相同的基本原则和相似的设计模式。但是，Cocoa和Cocoa Touch之间还是有一些不同的。

无论你的知识背景如何，都需要时刻牢记iOS开发与桌面应用程序开发之间的重要差异。接下来的几个小节将讨论这些差异。

### 1.3.1 只能运行一个应用

在iOS中，一次只能有一个应用处于活动状态并显示在屏幕上。从iOS 4开始，用户按下Home键后，应用程序可以在后台继续运行，但这只限于少数情况，而且必须专门为此编写代码（第15章将告诉你如何做到）。

不处于活动状态或者在后台运行时，应用程序不会占用任何CPU资源，这会严重干扰其与开放式网络的连接。iOS允许后台处理，不过要使应用程序在此状况下良好运行则需要开发者自己的努力。

### 1.3.2 只能使用一个窗口

在台式机和笔记本的操作系统中，多个程序可以同时运行，每个程序还可以创建并控制多个窗口。然而，除非应用程序经过特别编码，并且连接另一个屏幕或者使用Airplay镜像，否则iOS只允许应用程序操作一个窗口。应用程序与用户的所有交互都在这个窗口中完成，而这个窗口的大小就是iOS设备屏幕的固定大小。

### 1.3.3 有限的访问权限

通常，用户能够访问的内容，台式机和笔记本上的程序也可以访问。然而，iOS严格限制了应用程序的访问权限。

iOS的文件系统会为每个应用分配一块独立的区域，称为沙盒。每个应用只能读写自己沙盒内的文件。沙盒就是应用程序用于存储文档、偏好设置等任何有效数据的地方。

应用程序还会受到其他方面的限制。比如不能通过iOS访问端口号较小的网络，也不能进行在台式机中需要根用户权限或管理员权限的操作。

### 1.3.4 有限的响应时间

由于使用方式特殊，iOS需要快速响应各种事件，你的应用程序也应如此。启动应用程序时，要立即打开它，载入偏好设置和数据，并把主视图显示到屏幕上。这些过程要在几秒钟内全部完成。

应用程序在运行的时候，随时都有可能面对突然退入后台的情况。如果用户按下Home键，iOS就会返回主屏幕，应用必须尽快保存一切内容并退出。如果没有在5秒之内保存必要的数据并放弃对系统资源的控制，无论是

否已经完全保存，应用程序进程都会被终止。有一个API可以在应用程序终止前请求多一些时间来完成必要的工作，你需要知道如何使用它。

### 1.3.5  有限的屏幕尺寸

iPhone的屏幕显示效果非常出色。刚进入市场时，iPhone是当时分辨率最高的手持电子设备。不过，iPhone的显示空间并不大，比现代计算机的屏幕空间要小很多。最初几代iPhone的屏幕分辨率只有320像素×480像素，从iPhone 4的Retina屏幕开始，分辨率增加到了640像素×960像素。目前最大iPhone（即iPhone 6 Plus）的屏幕分辨率达到了1080像素×1920像素。这听起来像是个很平常的数字，不过要记住如此高密度的像素点（苹果公司用术语Retina来表示）被塞入了非常小的单位中，这对于iPhone和iPad上各种应用程序以及交互体验有重大的影响。表1-1列出了在编写本书时iOS 8支持的所有设备的屏幕尺寸。

表1-1  iOS设备屏幕尺寸

设    备	硬件尺寸	软件尺寸	精密程度
iPhone 4s	640×960	320×480	2x
iPhone 5和5s	640×1136	320×568	2x
iPhone 6	750×1334	375×667	2x
iPhone 6 Plus	1080×1920	414×736	3x
iPad 2和iPad mini	768×1024	768×1024	1x
iPad Air、iPad Retina和iPad mini Retina	1536×2048	768×1024	2x

硬件尺寸指的是屏幕的实际物理尺寸，以像素为单位。不过在编写软件时只需要注意软件尺寸一栏的数字。如你所见，软件尺寸大都是实际硬件尺寸的一半。这种情况是苹果公司引入第一代Retina设备后开始出现的，其像素数量在长度和宽度上都是上一代的两倍。如果苹果公司不作任何处理的话，所有现有的应用程序在新的Retina屏幕上都会只占据一半的尺寸，无法正常使用。因此苹果公司选择将应用程序绘制的所有内容都在内部乘以2，这样不需要作任何代码改动就可以将其铺满新屏幕。所有使用Retina屏幕的设备都采用了内部乘以2的机制，但iPhone 6 Plus有些例外，这是因为它拥有更高精密度的屏幕，需要乘以3。通常来说，你无需担心应用程序实际放大的倍数，只需要根据软件屏幕尺寸进行开发，剩下的事情交给iOS来做就行了。

唯一不适用于这个规则的是位图。由于位图自身的尺寸是固定的，你无法让同一张图片在Retina屏幕和非Retina屏幕上拥有一样好的效果。如果尝试这样做，会看到iOS为Retina屏幕设备将图片放大，从而产生了模糊的感觉。你可以通过为每张图片补充2倍和3倍尺寸的副本来解决这个问题，iOS会根据应用程序运行的设备屏幕来选取合适的那张。

**注意** 如果仔细观察表1-1，会发现第四列的精密程度就是硬件尺寸除以软件尺寸的比例。比如，iPhone 6的硬件宽度是750像素，软件宽度是375像素，比例是2:1。细心一点的读者会发现iPhone 6 Plus有些特殊，其硬件宽度与软件宽度的比例是1080/414，也就是2.608:1，高度也同样如此。就硬件而言，iPhone 6 Plus实际上不是3倍的Retina屏幕，而就软件而言则是3倍大小。这意味着，应用程序使用的软件屏幕尺寸414像素×736像素会按照规则首先映射为1242像素×2208像素的实际屏幕尺寸，并最终缩小到适应1080像素×1920像素的实际硬件尺寸。好在你不需要作任何特殊处理，iOS会负责这些细节。

### 1.3.6  有限的系统资源

提到一部至少拥有512MB内存和16GB存储空间的机器资源有限，很多资深程序员恐怕要发笑了，可事实确实是如此。或许开发iOS应用程序并不像是在内存为48KB的机器上编写复杂的电子表格应用，但是由于iOS具备的

图形特性和多种功能，很容易耗光内存。

目前市场上iOS设备的物理内存要么是512MB( iPhone 4s、iPad 2、一代iPad mini、iPod touch 5 ), 要么是1024MB( iPhone 5c、iPhone 5s、iPhone 6、iPhone 6 Plus、iPad Air、 iPad mini Retina ), 而且以后还会不断加大<sup>①</sup>。很大一部分内存被用于屏幕缓冲区和一些系统进程。一般，只有不到一半的内存留给应用程序使用( 实际可用内存可能更少，尤其是因为其他应用可能在后台运行 )。

虽然这些内存对于小型移动设备来说可能已经足够了，但谈到iOS的内存时，还要考虑另一个因素：现代的计算机操作系统( 比如OS X )会将未使用的内存块写到磁盘的交换文件中。这样，当应用程序请求的内存超过计算机的实际可用内存时，它仍然可以正常运行。但是，iOS并不会将易失性内存( 比如应用程序数据 )写到交换文件中。因此，应用程序的可用内存大小受限于iOS设备中未使用的物理内存空间。

Cocoa Touch提供了一种内置机制，可以在内存不足时通知应用程序。出现这种情况时，应用程序必须释放不必要的内存空间，否则就可能被强制退出。

### 1.3.7 不支持垃圾回收

之前提过，Cocoa Touch使用的是Objective-C，但是iOS并不支持Objective-C早在本世纪初就已经具有的一个关键特性：垃圾回收。是的，Cocoa Touch不支持垃圾回收。为iOS编写程序时需要手动管理内存，许多刚刚接触这个平台( 尤其是那些从支持垃圾回收的语言转来 )的开发者，还真有点不太适应。

但是，最新版本iOS支持的Objective-C基本解决了这个问题。这要归功于ARC( Automatic Reference Counting，自动引用计数 )功能，它解决了手动管理Objective-C对象占用内存的问题。ARC并不仅仅是垃圾回收的替代品，在很多方面实际上更优秀。因此，从OS X 10.8开始，ARC就成为了针对Mac应用程序的默认内存管理机制，垃圾回收则遭到了废弃。当然，ARC也是iOS应用程序的默认内存管理机制。第3章会详细介绍ARC。

### 1.3.8 一些新功能

前面提到过，Cocoa Touch缺少Cocoa的一些特性，但iOS SDK中也有一些功能是Cocoa目前没有的，至少不是在每一部Mac上都可用。

- ❑ iOS SDK中的Core Location框架可以帮助应用程序确定iOS设备的当前地理坐标。
- ❑ 大部分iOS设备都有内置的摄像头和照片库，SDK允许应用程序访问它们。
- ❑ iOS设备还有一个内置的运动传感器，用于检测设备的握持和移动方式。

### 1.3.9 与众不同的交互方法

iOS设备没有物理键盘和鼠标，这意味着iOS应用与用户的交互方式跟通用计算机完全不同。幸好，大部分交互都由iOS系统替你完成。例如，如果应用中用到了文本框，iOS系统就会在用户单击这个文本框时调出软键盘，不需要开发者为此额外编写代码。

---

**注意** 所有的设备都支持通过蓝牙连接外置键盘，这提供了一种不错的键盘体验，并且节省了屏幕空间。不过现在依然无法连接鼠标。

---

## 1.4 本书内容

下面是本书其余各章的简要概述。

---

<sup>①</sup> 事实上，iPad Air 2的内存已经达到了2GB。

- ❑ 第2章：讲述如何使用Xcode的搭档Interface Builder创建简单的界面，并在屏幕上显示文本。
- ❑ 第3章：展示与用户的交互，构建一个简单的应用程序，用于在运行时根据用户按下的按钮动态更新显示的文本。
- ❑ 第4章：以第3章为基础，介绍更多的标准iOS用户界面控件。此外，还将介绍如何使用警告框和操作表单提醒用户作出决策，或者通知用户发生了一些异常事件。
- ❑ 第5章：解释如何处理自动旋转和自动改变大小属性，以及使iOS应用程序同时支持纵向和横向运行的机制。
- ❑ 第6章：分析更多高级用户界面，并讲述如何创建多视图应用程序。还会介绍如何在运行时改变展示给用户的视图，从而创造出更强大的应用。
- ❑ 第7章：讨论如何实现标签栏和选取器等界面元素，它们是标准iOS用户界面的一部分。
- ❑ 第8章：介绍表视图。表视图是向用户提供数据列表的主要方法，并且是基于分层导航的应用程序的基础。这一章还会介绍如何让用户搜索应用程序数据。
- ❑ 第9章：说明如何实现分层列表这一标准的界面类型。分层列表是最常用的iOS应用程序界面之一，可以通过它查看更多详细的数据。
- ❑ 第10章：展示集合视图的使用。从一开始，各种iOS应用程序就使用了表视图来展示动态、垂直滚动的组件列表。在最近一段时间里，苹果引入了UICollectionView类，用于实现集合视图。集合视图在展示动态、垂直滚动的组件列表方面功能更加强大。使用集合视图，开发者可以更灵活地处理视觉组件的布局。
- ❑ 第11章：展示如何构建master-detail应用程序。它展示了一个列表（例如收件箱中的邮件），能让用户任意浏览某一项的详细信息。你将看到如何使用iOS控件来完成这一工作。这个功能原先是为iPad设计的，现在也可以用在iPhone上。
- ❑ 第12章：展示如何实现应用程序设置。iOS中的这种机制允许用户设置应用程序级的偏好设置。
- ❑ 第13章：介绍iOS中的数据管理，讨论如何创建对象来保存应用程序数据，以及如何将这些数据持久化到iOS文件系统中。这一章还会介绍使用Core Data的基础知识，使用Core Data可以很方便地保存和检索数据。
- ❑ 第14章：解释如何使用iCloud。iCloud是iOS 5引入的一项功能，可以将文档数据保存到iCloud中，这样就能在不同设备之间同步应用数据。
- ❑ 第15章：iOS开发人员可以使用一个强大的库来简化多线程开发，名为Grand Central Dispatch（GCD）。这一章将介绍Grand Central Dispatch，并展示在特定情况下如何使用iOS的特性让应用程序在后台运行。
- ❑ 第16章：人们都喜欢画画，这一章介绍自定义绘图。我们将为你讲解Core Graphics的绘图机制。
- ❑ 第17章：在iOS 7中，苹果引入了一个叫作Sprite Kit的新框架，用来创建2D游戏。它包含物理引擎和动画系统，也可以用来创建OS X系统上的游戏。在第17章中，你将会看到如何使用Sprite Kit创建一个简单的游戏。
- ❑ 第18章：iOS设备的多点触摸屏幕可以接受用户的各种手势输入。这一章讲述如何检测基本的手势，如双指捏合和单指滑动，还将介绍应该何时以及如何自定义新手势。
- ❑ 第19章：iOS可以通过Core Location确定设备所处的纬度和经度。本章会编写一些使用Core Location的代码，用于计算设备所在位置，并且使用位置信息来实现我们让世界连通的理想。
- ❑ 第20章：讨论如何与iOS的加速计和陀螺仪进行交互。通过它们可以确定设备的握持方式、运动速度，以及移动方向、设备所在位置。我们也会研究应用使用这类信息可以做哪些有趣的事情。
- ❑ 第21章：每台iOS设备都有自己的摄像头和照片库，获得用户允许之后，你的应用程序就可以访问摄像头和照片库。这一章介绍如何在自己的应用程序中使用它们。
- ❑ 第22章：iOS设备现已遍及90多个国家。本章会介绍如何把应用的各个部分恰当地翻译为其他语言，从而有助于发掘应用的潜在用户。
- ❑ 附录：介绍Swift编程语言，涵盖了理解本书示例代码所需的所有特性。



## 1.5 新版增加内容

自本书第1版上市以来，iOS开发社区取得了巨大发展。苹果不断更新SDK，使其不断发展完善。当然，我们也很忙。iOS 8和Xcode 6都增加了许多新的功能。我们竭尽全力更新了本书内容，涵盖了iOS 8和Xcode 6中关于开始编写iOS应用程序时需要注意的新技术。每个项目都是我们从头开始构建的，以确保项目的代码不仅能在最新版的Xcode和iOS SDK下编译，而且还能够充分利用Cocoa Touch提供的强大新功能。这一版的内容进行了大量的细微调整，所有的屏幕截图也都进行了更新。

## 1.6 Swift 和 Xcode 版本

Swift是一个仍在不断变动的新语言。在编写本书时，虽然iOS 8已经正式发布，但苹果公司依然在对Swift引入iOS API的方式进行修改。因此，用经过测试的Xcode编译并运行的示例代码，也许在你阅读本书时已经不再有效了。值得注意的是，苹果公司保证，对于现在编写的应用程序，其通过编译后的二进制文件能够在将来版本的iOS上运行，但是没有保证其源代码依然能正常编译。本书的代码是在Xcode 6.1中测试的。如果你发现同样的代码无法在当前使用的Xcode版本中编译，那么请浏览本书在Apress.com的页面并下载最新的源代码。如果仍然有问题，请在Apress.com提交勘误以提醒我们注意。

## 1.7 准备好了吗

iOS是一个不可思议的计算平台，是令人兴奋的新领域，让开发充满乐趣。和其他平台相比，编写iOS程序将成为一种全新的体验。所有看似熟悉的功能都具有其独特之处，但只要深入理解本书中的代码，就能把这些概念紧密联系起来，融会贯通。

应该谨记，完成本书中的所有练习并不能立即成为iOS开发专家。在进行下一个项目之前，请确保你已经完全理解了当前的内容。不要害怕修改代码，熟悉Cocoa Touch这种编程环境的最佳方法就是不断试验并观察结果。

如果你已经安装了iOS SDK，请继续阅读本书；如果还没有，请立即安装。明白了吗？好，那就开始动手吧！

## 关注图灵教育 关注图灵社区

# iTuring.cn

在线出版 电子书《码农》杂志 图灵访谈 ……



QQ联系我们

图灵读者官方群I: 218139230

图灵读者官方群II: 164939616



微博联系我们

官方账号: @图灵教育 @图灵社区 @图灵新知

市场合作: @图灵袁野

写作本版书: @图灵小花 @图灵张霞

翻译英文书: @朱巍ituring @楼伟珊

翻译日文书或文章: @图灵乐馨

翻译韩文书: @图灵陈曦

电子书合作: @hi\_jeanne

图灵访谈/《码农》杂志: @李盼ituring

加入我们: @王子是好人



微信联系我们



图灵教育  
turingbooks



图灵访谈  
ituring\_interview

## 亚马逊读者评论

“这本书简直太棒了！是我买过的性价比最高的一本书。作者非常清楚地解释了作出每种选择的原因以及每个iOS组件的特性，从而教你一种思维方式，而不是简单地完成每项任务。”

## 豆瓣读者评论

“这本书主要针对零基础读者，讲解得很详细。读完这本书再看代码会理解很多以前看不懂的东西。每个章节都是一个实例，很实用。我个人非常喜欢这种面面俱到的书，可以让你少走一些弯路。”

“（这本书是）学习iPhone开发必读的一本书，内容详细，条理清晰，例子丰富、有针对性。”

Beginning iPhone Development with Swift: Exploring the iOS SDK

# 精通iOS开发（第7版）

还在iPhone和iPad应用开发的大门徘徊？还是已经投身iOS应用开发阵营，但希望迅速提升自己的功力？选择本书绝对能令你惊喜连连。五位作者均是资深移动开发专家，具有丰富的Mac、iOS、Cocoa及Objective-C开发经验。作者将多年的实战经验与智慧感悟汇集成本书，旨在帮助没有经验的读者顺利叩开iOS应用开发的大门，帮助有经验的读者迅速提升功力，从而在iOS开发的道路上所向披靡。

本书自问世以来就受到读者的交口赞誉，被奉为学习iOS平台开发的不二之选。中文版累计销量已超过6万册。新版做了大幅修订，以Swift为主，使用双语讲解，力求使新老读者都能有最大收获。另外，作者重写了所有项目代码，使之兼容新旧SDK，并对原有的各章内容进行更新，增加了介绍Swift特性的附录，从而反映出技术的最新发展动态。

还等什么？立即展卷阅读，加入iOS开发的行列吧！

图灵iOS相关图书阅读路线图



Apress®

图灵社区: iTuring.cn

热线: (010)51095186转600

分类建议 计算机/移动开发

人民邮电出版社网址: www.ptpress.com.cn

ISBN 978-7-115-40111-3



9 787115 401113 >

ISBN 978-7-115-40111-3

定价: 118.00元

# 欢迎加入 图灵社区

## 最前沿的IT类电子书发售平台

电子出版的时代已经来临。在许多出版界同行还在犹豫彷徨的时候，图灵社区已经采取实际行动拥抱这个出版业巨变。作为国内第一家发售电子图书的IT类出版商，图灵社区目前为读者提供两种DRM-free的阅读体验：在线阅读和PDF。

相比纸质书，电子书具有许多明显的优势。它不仅发布快，更新容易，而且尽可能采用了彩色图片（即使有的书纸质版是黑白印刷的）。读者还可以方便地进行搜索、剪贴、复制和打印。

图灵社区进一步把传统出版流程与电子书出版业务紧密结合，目前已实现作译者网上交稿、编辑网上审稿、按章发布的电子出版模式。这种新的出版模式，我们称之为“敏捷出版”，它可以让读者以较快的速度了解到国外最新技术图书的内容，弥补以往翻译版技术书“出版即过时”的缺憾。同时，敏捷出版使得作、译、编、读的交流更为方便，可以提前消灭书稿中的错误，最大程度地保证图书出版的质量。

## 最方便的开放出版平台

图灵社区向读者开放在线写作功能，协助你实现自出版和开源出版梦想。利用“合集”功能，你就能联合二三好友共同创作一部技术参考书，以免费或收费的形式提供给读者。（收费形式须经过图灵社区立项评审。）这极大地降低了出版的门槛。只要有写作的意愿，图灵社区就能帮助你实现这个梦想。成熟的书稿，有机会入选出版计划，同时出版纸质书。

图灵社区引进出版的外文图书，都将在立项后马上在社区公布。如果你有意翻译哪本图书，欢迎你来社区申请。只要你通过试译的考验，即可签约成为图灵的译者。当然，要想成功地完成一本书的翻译工作，是需要有坚强的毅力的。

## 最直接的读者交流平台

在图灵社区，你可以十分方便地写文章、提交勘误、发表评论，以各种方式与作译者、编辑人员和其他读者进行交流互动。提交勘误还能够获赠社区银子。

你可以积极参与社区经常开展的访谈、审读、评选等多种活动，赢取积分和银子，积累个人声望。

ituring.com.cn