

SAMS
**Teach
Yourself**

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- “Read Less, Do More”（精读多练）的教学理念
- 以示例引导读者完成最常见的任务

每章内容针对初学者精心设计，**1**小时轻松阅读学习，
24小时彻底掌握关键知识

每章**案例与练习题**助你轻松完成常见任务，
通过**实践**提高应用技能，巩固所学知识

iOS 9 应用开发

入门经典

（第7版）

[美] John Ray 著
袁国忠 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

1000 户用开发

IOS 9 应用开发

[美] John Ray 著
袁国忠 译



人民邮电出版社

北京

图书在版编目 (C I P) 数据

iOS 9应用开发入门经典 : 第7版 / (美) 雷
(Ray, J.) 著 ; 袁国忠译. — 北京 : 人民邮电出版社,
2016. 6

ISBN 978-7-115-42178-4

I. ①i… II. ①雷… ②袁… 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(2016)第085715号

版 权 声 明

John Ray: Sams Teach Yourself iOS 9 Application Development in 24 Hours(7th Edition)

ISBN: 0672337673

Copyright © 2016 by Pearson Education, Inc.

Authorized translation from the English languages edition published by Pearson Education, Inc.

All rights reserved.

本书中文简体字版由美国 Pearson 公司授权人民邮电出版社出版。未经出版者书面许可, 对
本书任何部分不得以任何方式复制或抄袭。

版权所有, 侵权必究。

-
- ◆ 著 [美] John Ray
译 袁国忠
责任编辑 傅道坤
责任印制 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>
北京艺辉印刷有限公司印刷
- ◆ 开本: 787×1092 1/16
印张: 43.75
字数: 1101 千字 2016 年 6 月第 1 版
印数: 1—2 500 册 2016 年 6 月北京第 1 次印刷
- 著作权合同登记号 图字: 01-2016-2944 号
-

定价: 89.00 元

读者服务热线: (010)81055410 印装质量热线: (010)81055316
反盗版热线: (010)81055315

内容提要

介绍本书

本书基于 Apple 最新发布的 iOS 9.1 和 Xcode 7 编写，循序渐进地介绍了从事 iOS 开发所需的基本知识，包括使用 Xcode、Swift 和 Cocoa Touch 等开发工具，使用 iOS 试验场测试代码，设计及美化用户界面，使用切割工具创建按钮模板，利用多场景故事板、切换、弹出框、导航控制器和选项卡栏控制器，使用表视图和分割视图导航结构化数据，使用堆叠视图和 3D Touch，读写和显示数据，使用自动布局和尺寸类创建自适应用户界面，播放和录制多媒体，使用地图、地理编码和定位功能，使用加速计和陀螺仪检测运动和朝向，创建通用应用程序，编写支持后台处理的应用程序，跟踪和调试应用程序等主题。

本书通过简洁的语言和详细的步骤，帮助读者迅速掌握开发 iOS 应用程序所需的基本知识，适合没有任何编程经验的新手阅读，也可供有志于从事 iOS 开发的人员参考。

作者简介

要對容內

John Ray 当前是俄亥俄州立大学科研信息系统办公室（Office of Research Information Systems）主任。他在 Macmillan、Sams 和 Que 出版了大量图书，其中包括 *Using TCP/IP: Special Edition*、*Teach Yourself Dreamweaver MX in 21 Days*、*Mac OS X Unleashed*、*My OS X – El Capitan Edition* 和 *Teach Yourself iOS 8 Development in 24 Hours*。作为一名从 1984 年起就开始使用 Macintosh 的用户，他努力确保在每个项目中都会以应有的深度涉及 Macintosh。即使在编写技术性图书（如 *Using TCP/IP: Special Edition*）时，也在其中包含大量有关 Macintosh 及其应用程序的信息。他的写作手法简单明了，即使对初、中级读者来说也简单易懂，因此深受好评。

读者可访问网站 <http://teachyourselfios.com> 或关注 Twitter 账号 @johnemeryray 或 #iOSIn24 与他联系。

致谢

感谢 Sams Publishing 的团队成员 Laura Norman、Keith Cline 和 Mark Renfrow，还要感谢技术编辑 Anne Groves，是他们帮助我再次顺利地升级了本书。Swift 和 Xcode 工具的发展迅猛，这样的速度让开发人员有点难以跟上步伐。在确保我有条不紊、实事求是而又不偏离目标方面，Pearson 的编辑团队做得非常出色。

前言

当您拿起 iOS 设备并开始使用时，您会感觉非常流畅。无论是 iPad、iPhone 还是 iPod，其界面都犹如与您的手指连接在了一起：流畅、舒适、引人入胜。其他与之竞争的设备也提供了类似的功能，有的甚至装备了诸如全景屏幕和跟踪板等运动小部件，但用户体验却无法与 iOS 媲美。

最近几年，iOS 和相关开发工具发生了翻天覆地的变化。iOS 7 引入了使用半透明控件的用户界面，让用户能够将注意力放在内容上，同时又不迷失方向。iOS 8 支持一种全新的应用程序开发语言——Swift，让所有的人都大跌眼镜。随着 iOS 9 的推出，Swift 也变成了开源语言，进一步巩固了其 Apple 平台开发语言的地位。

Swift 是 iOS 和 OS X 开发历史中的里程碑。通过推出 Swift，Apple 几乎放弃了在 Apple 和 NeXT 平台上使用了 25 多年的 Objective-C 语言。Swift 具备更多现代语言的特征，有望成为更友好的开发平台。本书出版时，Apple 开发 Swift 语言的历史已有 4 年之久，但它成为公开的编程语言才一年多。

不幸的是，介绍一门年青的语言有些需要注意的地方。Swift 处于快速发展中，在每个 Xcode 版本中都不同——针对一个 Xcode 版本编写的代码，可能在下一个版本中不再管用。例如，在 iOS 9 自带的 Swift 版本中，依然采用了以前创建循环的语法，但其用途完全不同了。一夜之间，几个月前编写的代码就不管用了。

Swift 带来了挑战，但您也将发现，使用它进行编程既有趣又直观。

在打造 Swift 和 iOS 开发平台时，Apple 考虑到了应用程序的整个生命周期，进而在其中集成了所需的一切——从界面设计工具到使界面能够正常运行并显示所需的代码，并确保它们能够无缝地协同工作。作为开发人员，这是否意味着您需要遵守某些规则呢？确实如此，但通过遵守这些规则，您打造的应用程序将是深得用户喜爱的交互式艺术品，而非安装后就被遗忘的软件。

通过 App Store，Apple 向开发人员提供了一种最佳的数字发布系统。任何年龄和派别的程序员都可将其应用程序提交到 App Store，且只需要支付少量的开发人员年费。人们开发了针对各种领域的游戏、实用程序和应用程序，其范围涵盖了从学前教育到退休生活的所有阶段。鉴于 iPhone、iPod Touch 和 iPad 用户群庞大，因此不管什么内容都能找到合适的用户。

本书旨在向新一代开发人员介绍 iOS 开发相关的知识，并以循序渐进的方式提高开发人员的开发技能。读者只需通过 24 章内容的学习，就能掌握所有基本知识——从安装开发工具和向 Apple 注册设备，到调试应用程序，再到将应用程序提交到 App Store。

谁能成为 iOS 开发人员

只要有学习兴趣，有时间探索和使用 Apple 开发工具，并拥有一台运行 Yosemite、El Capitan 或更高版本的 Intel Macintosh 计算机，便可开始 iOS 开发了。更重要的是，从 Xcode 7 起，即便您没有加入开发人员计划，也可以在自己的设备上运行您开发的应用程序。

开发人员不可能在一夜之间就开发出 iOS 应用程序，但是只要多加练习，完全可以在几天之内编写出您的第一款应用程序。在 Apple 开发工具上花费的时间越多，创建出激动人心的应用程序的可能性就越大。

进行 iOS 开发时，要以创建自己想用的软件为宗旨，而不是创建您认为其他人想用的软件。如果只想着一夜暴富，您很可能会失望。虽然 App Store 空间很大，但它也是一个拥挤的市场，销售排行榜的竞争非常激烈。然而，如果将重点放在创建有用而独特的应用程序上，您的劳动成果得到用户赏识的可能性将大很多。

本书适合的读者

本书是为从未进行 iOS 开发，但使用过 Macintosh 平台的读者编写的，读者不需要有 Swift、Cocoa 和 Apple 开发工具方面的经验。当然，读者如果有一定的开发经验，将更容易掌握这些工具和技术。

虽然如此，本书对读者还是有一定的要求。具体地说，读者必须愿意花时间学习。如果读者只是阅读每章的内容，而不完成其中的项目，很可能错过一些重要概念。另外，读者还需花时间阅读 Apple 开发文档，并研究本书介绍的主题。有关 iOS 开发的信息浩如烟海，但本书的篇幅有限，只能为您打下坚实的 iOS 开发基础。

本书的内容

本书是基于 iOS 9.1 和 Xcode 7.1 编写的，大部分内容都适用于所有 iOS 版本，但也介绍了较新的 iOS 版本新增的一些功能，如手势识别器、支持 3D Touch 和 AirPlay 的嵌入式视频播放、Core Image、社交网络、多任务、通用应用程序（适用于 iPhone 和 iPad）、自动布局、尺寸类等。

遗憾的是，本书并非完整的 iOS 应用程序编程接口（API）参考手册，也未涉及 Apple TV 和 Apple Watch 开发，因为本书的篇幅无法满足有些主题的需求。虽然如此，本书深入地介绍了各种工具和技巧，足以帮助任何 iOS 开发人员取得成功；另外，您将在第 1 章下载的免费工具包含 Apple 开发文档。很多章都包含名为“进一步探索”的一节，指出了您可能感兴趣的其他相关主题。这里需要重申的是，探索精神是成功的开发人员必须具备的重要品质。

涉及编码的每章都有配套的项目文件，其中包含编译并测试示例所需的一切；但更佳的做法是根据介绍自己创建应用程序。请务必从本书的配套网站 <http://teachyourselfios.com> 下载项目文件。如果您在使用这些项目时遇到问题，请查看该网站的帖子，看看是否有解决方案。

目 录

第 1 章 为开发准备好系统和 iOS 设备 1

1.1 欢迎进入 iOS 平台 1

1.1.1 iOS 设备 1

1.1.2 显示屏和图形 2

1.1.3 应用程序资源约束 4

1.1.4 连接性 4

1.1.5 输入和反馈 4

1.2 成为 iOS 开发人员 5

1.2.1 Apple 开发人员计划 5

1.2.2 加入付费的开发人员计划 6

1.2.3 安装 Xcode 7

1.3 运行 iOS 应用 9

1.3.1 什么是开发供应配置文件 9

1.3.2 配置用于开发的设备 9

1.3.3 启动应用 11

1.4 开发技术概述 14

1.4.1 Apple 开发工具 14

1.4.2 Swift 15

1.4.3 CoCoa Touch 15

1.4.4 模型-视图-控制器 15

1.5 进一步探索 15

1.6 小结 15

1.7 问与答 16

1.8 作业 16

1.8.1 测验 16

1.8.2 答案 18

1.8.3 练习 18

第 2 章 Xcode 和 iOS 模拟器简介 19

2.1 使用 Xcode 19

2.1.1 创建和管理项目 20

2.1.2 编辑和导航代码 29

2.1.3 生成应用程序 35

2.1.4 管理项目属性 38

2.1.5 设置支持的设备朝向 39

2.1.6 应用程序图标、启动图像和
启动屏幕 40

2.2 使用 iOS 模拟器 43

2.2.1 启动应用程序 44

2.2.2 模拟多点触摸事件 45

2.2.3 旋转模拟的设备 46

2.2.4 模拟其他情况 46

2.2.5 添加其他要模拟的设备 47

2.3 进一步探索 49

2.4 小结 49

2.5 问与答 49

2.6 作业 49

2.6.1 测验 49

2.6.2 答案 51

2.6.3 练习 51

第 3 章 探索 Swift 和 iOS 试验场 52

3.1 面向对象编程和 Swift 52

3.2 面向对象编程术语 54

3.3 探索 Swift 文件结构 55

3.3.1 类文件 56

3.3.2 自动生成的结构 60

3.4 Swift 编程基础 61

3.4.1 声明变量和常量 61

3.4.2 使用方法 70

3.4.3 表达式和决策	75	5.2.1 对象库	121
3.5 内存管理和自动引用计数	81	5.2.2 将对象加入到视图中	121
3.6 iOS 试验场简介	82	5.2.3 使用 IB 编辑工具	122
3.6.1 新建试验场	82	5.2.4 自动布局系统	126
3.6.2 使用试验场	83	5.3 定制界面外观	128
3.6.3 生成并查看输出	84	5.3.1 使用属性检查器	128
3.7 进一步探索	87	5.3.2 设置辅助功能属性	129
3.8 小结	87	5.3.3 预览界面	131
3.9 问与答	88	5.4 连接到代码	133
3.10 作业	88	5.4.1 打开项目	133
3.10.1 测验	88	5.4.2 实现概述	134
3.10.2 答案	90	5.4.3 输出口和操作	135
3.10.3 练习	90	5.4.4 对象身份	139
第 4 章 Cocoa Touch 内幕	91	5.5 进一步探索	140
4.1 Cocoa Touch 是什么	91	5.6 小结	141
4.1.1 冷静面对大量的功能	92	5.7 问与答	141
4.1.2 少年老成	92	5.8 作业	141
4.2 探索 iOS 技术层	93	5.8.1 测验	141
4.2.1 Cocoa Touch 层	93	5.8.2 答案	143
4.2.2 多媒体层	94	5.8.3 练习	143
4.2.3 核心服务层	96	第 6 章 模型-视图-控制器应用	
4.2.4 核心 OS 层	98	程序设计	144
4.3 跟踪 iOS 应用程序的生命周期	98	6.1 了解设计模式 MVC	144
4.4 Cocoa 基础	99	6.1.1 制作意大利面条式代码	145
4.4.1 核心应用程序类	100	6.1.2 使用 MVC 将应用程序设计	
4.4.2 数据类型类	101	结构化	145
4.4.3 界面类	103	6.2 Xcode 如何实现 MVC	146
4.5 使用 Xcode 探索 iOS 框架	106	6.2.1 视图	146
4.5.1 Xcode 文档	106	6.2.2 视图控制器	146
4.5.2 快速帮助	110	6.2.3 数据模型	148
4.6 进一步探索	111	6.3 使用模板 Single View	
4.7 小结	112	Application	149
4.8 问与答	112	6.3.1 实现概述	149
4.9 作业	112	6.3.2 创建项目	149
4.9.1 测验	112	6.3.3 规划变量和连接	153
4.9.2 答案	114	6.3.4 设计界面	155
4.9.3 练习	114	6.3.5 创建并连接输出口和操作	158
第 5 章 探索 Xcode Interface Builder	115	6.3.6 实现应用程序逻辑	161
5.1 了解 Interface Builder	115	6.3.7 生成应用程序	161
5.1.1 IB 采用的方法	116	6.4 进一步探索	162
5.1.2 剖析 IB 故事板	116	6.5 小结	162
5.2 创建用户界面	120	6.6 问与答	162
		6.7 作业	163

6.7.1 测验	163
6.7.2 答案	164
6.7.3 练习	165
第7章 使用文本、键盘和按钮	166
7.1 基本用户输入和输出	166
7.1.1 按钮	166
7.1.2 文本框和文本视图	167
7.1.3 标签	167
7.2 使用文本框、文本视图和按钮	168
7.2.1 实现概述	168
7.2.2 创建项目	168
7.2.3 使用切割技术制作按钮模板	169
7.2.4 设计界面	173
7.2.5 创建并连接输出和操作	182
7.2.6 隐藏键盘	184
7.2.7 实现应用程序逻辑	187
7.2.8 生成应用程序	189
7.3 进一步探索	189
7.4 小结	190
7.5 问与答	190
7.6 作业	190
7.6.1 测验	190
7.6.2 答案	192
7.6.3 练习	192

第8章 处理图像、动画、滑块和步进控件	193
8.1 用户输入和输出	193
8.1.1 滑块	194
8.1.2 步进控件	194
8.1.3 图像视图	194
8.1.4 视觉效果视图	195
8.2 创建并管理图像动画、滑块和步进控件	195
8.2.1 实现概述	195
8.2.2 创建项目	196
8.2.3 设计界面	197
8.2.4 创建并连接到输出和操作	204
8.2.5 实现应用程序逻辑	207
8.2.6 生成应用程序	213
8.3 进一步探索	213
8.4 小结	214

8.5 问与答	214
8.6 作业	214
8.6.1 测验	214
8.6.2 答案	216
8.6.3 练习	216
第9章 使用高级界面对象和视图	217
9.1 再谈用户输入和输出	217
9.1.1 开关	218
9.1.2 分段控件	218
9.1.3 Web 视图	219
9.1.4 可滚动的视图	222
9.1.5 堆叠视图	222
9.2 使用开关、分段控件和 Web 视图	222
9.2.1 实现概述	222
9.2.2 创建项目	223
9.2.3 设计界面	224
9.2.4 创建并连接输出和操作	228
9.2.5 实现应用程序逻辑	231
9.2.6 生成应用程序	235
9.3 使用可滚动视图和堆叠视图	236
9.3.1 实现概述	236
9.3.2 创建项目	237
9.3.3 设计界面	237
9.3.4 创建并连接输出和操作	241
9.3.5 实现应用程序逻辑	241
9.3.6 生成应用程序	242
9.4 进一步探索	242
9.5 小结	243
9.6 问与答	243
9.7 作业	244
9.7.1 测验	244
9.7.2 答案	245
9.7.3 练习	246

第10章 引起用户注意	247
10.1 提醒用户	247
10.1.1 提醒控制器	248
10.1.2 系统声音服务	254
10.2 探索提醒用户的方法	255
10.2.1 实现概述	256
10.2.2 创建项目	256
10.2.3 设计界面	257
10.2.4 创建并连接输出和操作	258

10.2.5 实现提醒	259	第 12 章 使用工具栏和选择器	
10.2.6 实现操作表	264	做出选择	310
10.2.7 实现提醒音和震动	266	12.1 了解工具栏	310
10.3 进一步探索	267	12.2 探索选择器	313
10.4 小结	268	12.2.1 日期选择器	313
10.5 问与答	268	12.2.2 选择器视图	314
10.6 作业	268	12.3 使用日期选择器	318
10.6.1 测验	268	12.3.1 实现概述	319
10.6.2 答案	270	12.3.2 创建项目	319
10.6.3 练习	270	12.3.3 设计界面	321
第 11 章 实现多场景和弹出框	271	12.3.4 创建切换	322
11.1 多场景故事板简介	271	12.3.5 创建并连接输出口和操作	322
11.1.1 术语	272	12.3.6 实现日期计算逻辑	323
11.1.2 创建多场景项目	273	12.3.7 实现场景切换逻辑	328
11.1.3 添加提供支持的视图		12.3.8 生成应用程序	329
控制器子类	275	12.4 实现自定义选择器	329
11.1.4 创建切换	277	12.4.1 实现概述	329
11.1.5 配置模态切换	279	12.4.2 创建项目	330
11.1.6 配置弹出切换	280	12.4.3 设计界面	331
11.1.7 手工控制模态切换	283	12.4.4 创建切换	333
11.1.8 使用出口和回退切换	284	12.4.5 创建并连接输出口和操作	333
11.1.9 以编程方式创建切换	285	12.4.6 实现自定义选择器视图	334
11.1.10 弹出切换的独特之处	287	12.4.7 实现场景切换逻辑	339
11.1.11 在场景之间传递数据	290	12.4.8 生成应用程序	340
11.2 使用切换	292	12.5 进一步探索	341
11.2.1 实现概述	292	12.6 小结	341
11.2.2 创建项目	292	12.7 问与答	342
11.2.3 设计界面	295	12.8 作业	342
11.2.4 创建模态切换	298	12.8.1 测验	342
11.2.5 回退到初始场景	299	12.8.2 答案	344
11.2.6 创建并连接输出口和操作	299	12.8.3 练习	344
11.2.7 实现应用程序逻辑	300	第 13 章 使用导航控制器和选项卡栏	
11.2.8 生成应用程序	302	控制器	345
11.3 弹出框、通用应用程序		13.1 高级视图控制器	345
和 iPhone	302	13.2 探索导航控制器	347
11.3.1 配置弹出切换	302	13.2.1 导航栏、导航项和栏按钮项	347
11.3.2 改为通用应用程序	303	13.2.2 在故事板中使用导航控制器	348
11.4 进一步探索	306	13.2.3 在导航场景之间共享数据	350
11.5 小结	306	13.3 了解选项卡栏控制器	350
11.6 问与答	307	13.3.1 选项卡栏和选项卡栏项	351
11.7 作业	307	13.3.2 在故事板中使用选项卡栏	
11.7.1 测验	307	控制器	351
11.7.2 答案	309		
11.7.3 练习	309		

13.3.3 在选项卡栏控制器管理的 场景之间共享数据	354
13.4 使用导航控制器	355
13.4.1 实现概述	355
13.4.2 创建项目	356
13.4.3 创建显示切换	358
13.4.4 设计界面	359
13.4.5 创建并连接输出口和操作	360
13.4.6 实现应用程序逻辑	361
13.4.7 生成应用程序	362
13.5 使用选项卡栏控制器	362
13.5.1 实现概述	363
13.5.2 创建项目	363
13.5.3 创建选项卡栏关系	365
13.5.4 设计界面	365
13.5.5 创建并连接输出口和操作	367
13.5.6 实现应用程序逻辑	367
13.5.7 生成应用程序	370
13.6 进一步探索	370
13.7 小结	370
13.8 问与答	371
13.9 作业	371
13.9.1 测验	371
13.9.2 答案	373
13.9.3 练习	373

第 14 章 使用表视图和分割视图控制器

导航数据	374
14.1 了解表视图	374
14.1.1 表视图的外观	375
14.1.2 表单元格	375
14.1.3 添加表视图	376
14.2 探索分割视图控制器	381
14.2.1 实现分割视图控制器	382
14.2.2 模板 Master-Detail Application	383
14.3 一个简单的表视图应用程序	384
14.3.1 实现概述	384
14.3.2 创建项目	385
14.3.3 设计界面	385
14.3.4 连接输出口 delegate 和 dataSource	387
14.3.5 实现应用程序逻辑	387
14.3.6 生成应用程序	391
14.4 创建基于主-从视图的 应用程序	391

14.4.1 实现概述	392
14.4.2 创建项目	392
14.4.3 调整界面	393
14.4.4 实现应用程序数据源	395
14.4.5 实现主视图控制器	398
14.4.6 实现细节视图控制器	401
14.4.7 生成应用程序	403
14.5 进一步探索	403
14.6 小结	403
14.7 问与答	404
14.8 作业	404
14.8.1 测验	404
14.8.2 答案	406
14.8.3 练习	406

第 15 章 读写应用程序数据

15.1 iOS 应用程序和数据存储	407
15.2 数据存储方式	409
15.2.1 用户默认设置	410
15.2.2 设置束	411
15.2.3 直接访问文件系统	412
15.3 创建隐式首选项	416
15.3.1 实现概述	417
15.3.2 创建项目	417
15.3.3 创建界面	417
15.3.4 创建并连接输出口和操作	418
15.3.5 实现应用程序逻辑	419
15.3.6 生成应用程序	421
15.4 实现系统设置	422
15.4.1 实现概述	422
15.4.2 创建项目	422
15.4.3 设计界面	423
15.4.4 创建并连接输出口	424
15.4.5 创建设置束	425
15.4.6 实现应用程序逻辑	430
15.4.7 生成应用程序	431
15.5 实现文件系统存储	432
15.5.1 实现概述	432
15.5.2 创建项目	432
15.5.3 设计界面	432
15.5.4 创建并连接输出口和操作	433
15.5.5 实现应用程序逻辑	435
15.6 进一步探索	438
15.7 小结	439
15.8 问与答	439

15.9 作业	439
15.9.1 测验	439
15.9.2 答案	441
15.9.3 练习	441
第 16 章 创建自适应用户界面	442
16.1 自适应界面	442
16.1.1 启用界面旋转	443
16.1.2 设计可旋转和调整大小 的界面	444
16.2 使用自动布局	446
16.2.1 自动布局的语言和工具	446
16.2.2 在约束对象中导航	448
16.2.3 手工定义约束	454
16.2.4 示例 1: 居中约束	456
16.2.5 示例 2: 增大控件	458
16.2.6 示例 3: 统一调整大小	462
16.3 以编程方式定义界面	464
16.3.1 实现概述	464
16.3.2 创建项目	465
16.3.3 界面编程	466
16.3.4 生成应用程序	470
16.4 进一步探索	470
16.5 小结	470
16.6 问与答	470
16.7 作业	471
16.7.1 测验	471
16.7.2 答案	472
16.7.3 练习	472
第 17 章 使用复杂的触摸和手势	473
17.1 多点触摸手势识别	473
17.2 添加手势识别器	474
17.3 3D Touch 预览和打开	475
17.4 使用手势识别器	477
17.4.1 实现概述	478
17.4.2 创建项目	479
17.4.3 设计界面	480
17.4.4 给视图添加手势识别器	481
17.4.5 创建并连接输出口和操作	484
17.4.6 实现应用程序逻辑	486
17.4.7 生成应用程序	494
17.5 实现 3D Touch 手势	494
17.5.1 实现概述	494

17.5.2 修改 ModalEditor	495
17.5.3 修改 FlowerDetail	496
17.6 进一步探索	498
17.7 小结	498
17.8 问与答	499
17.9 作业	499
17.9.1 测验	499
17.9.2 答案	500
17.9.3 练习	501
第 18 章 检测朝向和移动	502
18.1 理解运动硬件	502
18.1.1 加速计	503
18.1.2 陀螺仪	504
18.2 访问朝向和运动数据	505
18.2.1 通过 UIDevice 请求朝向通知	505
18.2.2 使用 Core Motion 读取加速度、 旋转速度和姿态	506
18.3 检测朝向	508
18.3.1 实现概述	508
18.3.2 创建项目	508
18.3.3 设计界面	508
18.3.4 创建并连接输出口	509
18.3.5 实现应用程序逻辑	509
18.3.6 生成应用程序	511
18.4 检测加速、倾斜和旋转	512
18.4.1 实现概述	512
18.4.2 创建项目	513
18.4.3 设计界面	513
18.4.4 创建并连接输出口和操作	514
18.4.5 实现应用程序逻辑	515
18.4.6 生成应用程序	520
18.5 进一步探索	521
18.6 小结	521
18.7 问与答	521
18.8 作业	522
18.8.1 测验	522
18.8.2 答案	523
18.8.3 练习	524
第 19 章 使用多媒体	525
19.1 探索多媒体	525
19.1.1 MediaPlayer 框架	526
19.1.2 AVFoundation 和 AVKit 框架	529

19.1.3 图像选择器	535
19.1.4 Core Image 框架	536
19.2 创建用于练习使用多媒体 的应用程序	538
19.2.1 实现概述	538
19.2.2 创建项目	538
19.2.3 设计界面	539
19.2.4 使用容器视图嵌入 AVPlayerViewController	540
19.2.5 创建并连接输出和端口	541
19.2.6 实现视频播放器	543
19.2.7 实现音频录制和播放	545
19.2.8 使用照片库和相机	549
19.2.9 实现 Core Image 滤镜	552
19.2.10 访问并播放音乐库	554
19.3 进一步探索	559
19.4 小结	559
19.5 问与答	560
19.6 作业	560
19.6.1 测验	560
19.6.2 答案	562
19.6.3 练习	562
第 20 章 与其他 iOS 服务交互	563
20.1 集成 iOS 服务	563
20.1.1 通讯录	563
20.1.2 电子邮件	567
20.1.3 在社交网站上发表信息	569
20.1.4 访问 Web 浏览器 Safari	570
20.1.5 地图功能	571
20.2 使用通讯录、电子邮件、 社交网络、Safari 和地图	577
20.2.1 实现概述	578
20.2.2 创建项目	578
20.2.3 设计界面	578
20.2.4 创建并连接输出和端口	580
20.2.5 实现通讯录逻辑	581
20.2.6 实现地图逻辑	583
20.2.7 实现电子邮件逻辑	585
20.2.8 实现社交网络逻辑	587
20.2.9 实现 Safari 视图控制器	588
20.2.10 将状态栏设置为白色 (可选)	589
20.2.11 生成应用程序	589
20.3 进一步探索	590
20.4 小结	590

20.5 问与答	590
20.6 作业	590
20.6.1 测验	590
20.6.2 答案	592
20.6.3 练习	592
第 21 章 实现定位服务	594
21.1 理解 Core Location	594
21.1.1 获取位置	594
21.1.2 获取航向	599
21.2 创建支持定位的应用程序	600
21.2.1 实现概述	600
21.2.2 创建项目	600
21.2.3 设计视图	601
21.2.4 创建并连接输出和端口	603
21.2.5 实现应用程序逻辑	603
21.2.6 生成应用程序	607
21.3 理解磁性指南针	608
21.3.1 实现概述	608
21.3.2 创建项目	609
21.3.3 修改用户界面	609
21.3.4 创建并连接输出和端口	610
21.3.5 修改应用程序逻辑	610
21.3.6 生成应用程序	615
21.4 进一步探索	616
21.5 小结	616
21.6 问与答	616
21.7 作业	617
21.7.1 测验	617
21.7.2 答案	618
21.7.3 练习	619

第 22 章 创建支持后台处理的 应用程序	620
22.1 理解 iOS 后台处理	620
22.1.1 后台处理类型	621
22.1.2 支持后台处理的应用程序 的生命周期	625
22.2 禁用后台处理	627
22.3 处理后台挂起	628
22.4 实现本地通知	629
22.4.1 请求用户允许使用通知	629
22.4.2 常用的通知属性	630
22.4.3 创建和调度通知	630

22.5 使用任务特定的后台处理.....	632	22.12.3 练习.....	650
22.5.1 修改应用程序 Cupertion 以 支持在后台播放音频.....	632	第 23 章 创建通用应用程序.....	651
22.5.2 使用声音指示前往库珀蒂诺 的方向.....	633	23.1 开发通用应用程序.....	651
22.5.3 添加后台模式.....	636	23.1.1 将项目配置为通用的.....	652
22.6 完成长时间运行的后台任务.....	636	23.1.2 通用应用程序开发工具 和技巧.....	654
22.6.1 实现概述.....	636	23.2 尺寸类.....	654
22.6.2 创建项目.....	637	23.2.1 使用尺寸类工具.....	656
22.6.3 设计界面.....	637	23.2.2 使用尺寸类创建故事板.....	661
22.6.4 创建并连接输出.....	638	23.3 进一步探索.....	665
22.6.5 实现应用程序逻辑.....	638	23.4 小结.....	666
22.6.6 启用后台任务处理.....	639	23.5 问与答.....	666
22.6.7 生成应用程序.....	641	23.6 作业.....	666
22.7 执行后台获取.....	641	23.6.1 测验.....	666
22.7.1 实现概述.....	641	23.6.2 答案.....	667
22.7.2 创建项目.....	641	23.6.3 练习.....	667
22.7.3 设计界面.....	642	第 24 章 应用程序跟踪、监视和调试.....	668
22.7.4 创建并连接输出.....	642	24.1 使用 NSLog 提供即时反馈.....	668
22.7.5 实现应用程序逻辑.....	642	24.1.1 使用 NSLog.....	669
22.7.6 添加后台获取模式.....	644	24.1.2 查看 NSLog 输出.....	670
22.7.7 生成应用程序.....	644	24.2 使用 Xcode 调试器.....	671
22.8 添加 3D Touch 快速操作 菜单.....	644	24.2.1 设置断点及单步执行代码.....	672
22.8.1 添加快速操作图标.....	645	24.2.2 使用调试导航器.....	680
22.8.2 定义快速操作.....	645	24.3 进一步探索.....	684
22.8.3 实现快速操作逻辑.....	646	24.4 小结.....	684
22.8.4 生成应用程序.....	647	24.5 问与答.....	684
22.9 进一步探索.....	647	24.6 作业.....	684
22.10 小结.....	648	24.6.1 测验.....	684
22.11 问与答.....	648	24.6.2 答案.....	686
22.12 作业.....	648	24.6.3 练习.....	686
22.12.1 测验.....	648		
22.12.2 答案.....	650		

第1章

为开发准备好系统和 iOS 设备

本章将介绍：

- 您面临的 iOS 硬件局限性；
- 到哪里获取 iOS 开发工具；
- 如何加入 iOS 开发人员计划（Developer Program）；
- 创建并使用供应配置文件（provisioning profile）；
- iOS 开发技术概述。

iOS 设备向开发人员展示了一个全新的世界——多点触摸界面、可始终在线、视频以及众多内置的传感器，这些传感器可用于创建从游戏到提高生产率的应用程序等各种软件。信不信由您，作为新开发人员，您有一个优势：您是白纸一张，不受以前知道的移动应用程序概念的羁绊。您的高见很可能变成 Apple App Store 的优秀作品。

本章将让您为开发第一个项目做好准备。您即将踏上成为 iOS 开发人员的道路，但在开始编码前还需要做些准备工作。

1.1 欢迎进入 iOS 平台

您阅读本书说明您可能拥有 iOS 设备，这意味着您知道如何使用其界面：清晰的图形、响应迅速、多点触摸和数以十万计的应用程序，这只是冰山一角。然而，作为开发人员，您需要习惯应对一个迫使您以不同方式思考的平台。

1.1.1 iOS 设备

当前，iOS 平台家族成员包括 iPhone、iPad 和 iPod Touch。Apple Watch（watchOS）和

Apple TV (tvOS) 是类似的平台, 但使用不同的用户界面, 要求使用不同的开发方法。在阅读本书时您将发现, 很多屏幕截图都基于以 iPhone (具体地说是 4.7 英寸的屏幕) 为中心的项目, 这并非是不喜欢大设备, 而是因为 iPad 和 iPhone Plus 界面很大, 难以在屏幕截图中显示。好消息是, 如果您要开发针对 iPad 的项目, 这样去做就是了! 同样, 如果您要开发针对 iPhone 6s+ 的项目, 去做好了。为这两种设备进行开发时, 编码过程几乎相同。对于不同的情况, 我将确保您明白设备之间的差别 (及其原因), 但这样的情况很少。您还将发现, 每个示例都提供了 iPhone 版本和 iPad 版本, 这些代码可从本书的配套网站 (<http://teachyourselfios.com>) 下载。因此, 不管您使用哪种设备, 按本书介绍的做都将创建适用于该设备的应用程序。

By the Way

注意:

与 Apple 开发工具和文档一样, 本书不区分 iPhone 和 iPod Touch。几乎在所有情况下, 为这些设备开发的方式都相同, 虽然有些功能在较早的 iPod Touch 版本上没有, 但较早的 iPhone 和 iPad 版本亦如此。

1.1.2 显示屏和图形

iOS 设备的分辨率各不相同, 但 iOS 提供了一种考虑分辨率的简单方式。例如, 在 iPhone 5 和 iPhone 6 之前, iPhone 的屏幕大小为 320×480 点。请注意, 这里说的是“点”而不是像素。iPhone 4 采用了 Retina 屏幕, 在此之前, iPhone 的屏幕分辨率为 320×480 像素; 现在, iOS 设备的实际分辨率为上述分辨率与缩放因子的乘积。这意味着虽然在小设备 (如 iPhone 4s) 上对元素进行定位时, 使用的是数字 320×480 , 但像素数可能更多。例如, iPhone 4 (s)、5 (s) 和 6 的缩放因子为 2, 这意味着 iPhone 4s 的实际分辨率为 $(320 \times 2) \times (480 \times 2)$, 即 640×960 像素。iPhone 5 稍有不同, 其屏幕更大, 为 320×568 点, 即 640×1136 像素; 而 iPhone 6 的屏幕 375×667 点, 即 750×1334 像素, 如图 1.1 所示。

那么庞大的 iPhone 6+/6s+ 呢? 它更与众不同——缩放因子为 3, 这意味着屏幕上的每个点都表示 9 个像素。以点为单位时, 屏幕尺寸是多少呢? 414×736 点。在缩放因子为 3 的情况下, 这相当于 1242×2208 像素, 对吧? 不完全对。iPhone Plus 的分辨率为 1080×1920 , 但“内部”分辨率为 1242×2208 。实际上, 您可以这么认为: 内部分辨率更高, 但自动缩小内容, 以适合 1080×1920 的屏幕。

像素如此之多, 屏幕空间好像很大 (即便是 iPhone 5 亦如此), 但别忘了, 所有这些像素都将显示在对角线只比无名指长一两英寸的屏幕上。

iPad 系列产品的情况类似。iPad 2 和第一代 iPad mini 的屏幕分辨率为 768×1024 点, 但其缩放因子为 1。采用 Retina 屏幕的 iPad Air 和 iPad Mini 的屏幕分辨率仍为 768×1024 点, 但缩放因子为 2, 因此以像素为单位时, 分辨率为 1536×2048 。iPad Pro 配置了 1024×1366 点的超大屏幕, 其缩放因子为 2, 因此以像素为单位时, 分辨率为 2048×2732 。

本书后面介绍如何将对象放置到屏幕上时, 将更详细地介绍缩放因子的工作原理。您要知道的要点是, 当您创建应用程序时, iOS 将考虑缩放因子, 以最大可能的分辨率显示应用程序及其界面, 而您几乎无需为此做任何工作。