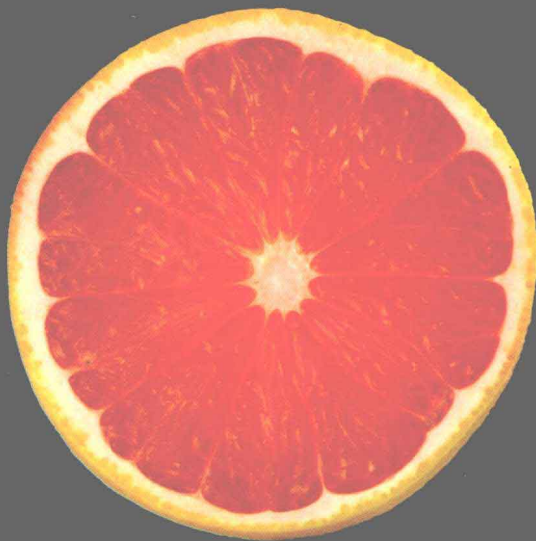


- 创造销售奇迹的经典著作最新版
- 针对iOS 5 SDK新特性全面升级
- 引领你跨入移动应用开发的大门



Beginning iOS 5 Development
Exploring the iOS SDK

iOS 5基础教程

Dave Mark

[美] Jack Nutting 著

Jeff LaMarche

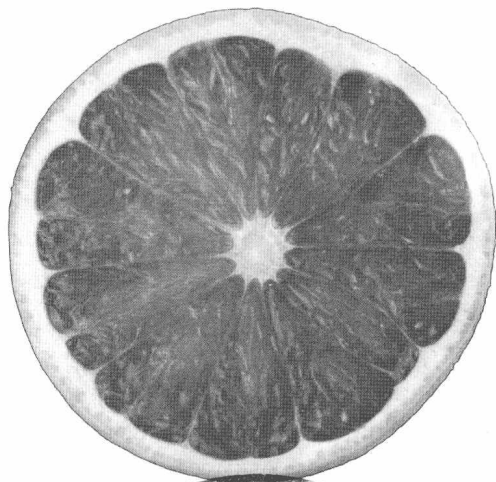
毛姝雯 漆振 杨越 孙文磊 等译



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书 移动开发系列



Beginning iOS 5 Development
Exploring the iOS SDK

iOS 5基础教程

Dave Mark
[美] Jack Nutting
Jeff LaMarche

毛姝雯 漆振 杨越 孙文磊 等译



人民邮电出版社
北京

图书在版编目 (C I P) 数据

iOS 5基础教程 / (美) 马克 (Mark, D.), (美) 纳丁 (Nutting, J.), (美) 拉马赫 (LaMarche, J.) 著; 毛姝雯等译. — 北京: 人民邮电出版社, 2012. 9

(图灵程序设计丛书)

书名原文: Beginning iOS 5 Development:
Exploring the iOS SDK
ISBN 978-7-115-29099-1

I. ①i… II. ①马… ②纳… ③拉… ④毛… III. ①
移动电话机—游戏程序—程序设计—教材 IV.
①TN929.53②TP311.5

中国版本图书馆CIP数据核字(2012)第188334号

内 容 提 要

iOS 是如今炙手可热的移动平台, 苹果公司为其推出了强大的软件开发工具包 iOS SDK。本书是一部关于 iOS SDK 开发的基础教程, 内容翔实、语言生动。几位作者结合消费类设备上的常见实例, 循序渐进地讲解了适用于 iPhone 4S、iPad 2 及 iPod touch 开发的基本流程。新版介绍强大的 iOS 5 操作系统, 涵盖 Xcode 4 以来的新功能, 其中最为引人注目的便是 storyboard 和 iCloud, 书中将以全新章节详细介绍。全书所有项目均使用 Xcode 4.2 重新创建, 让开发者全面感受 Xcode 4 带来的振奋人心的新变化。

本书具有较强的通用性, 编程领域的各层次读者都能通过本书快速学习 iOS 开发, 提高相关技能。

图灵程序设计丛书

iOS 5基础教程

-
- ◆ 著 [美] Dave Mark Jack Nutting Jeff LaMarche
译 毛姝雯 漆 振 杨 越 孙文磊 等
责任编辑 朱 巍
执行编辑 刘美英
 - ◆ 人民邮电出版社出版发行 北京市崇文区夕照寺街14号
邮编 100061 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: 800×1000 1/16
印张: 37.25
字数: 880千字 2012年9月第1版
印数: 1-5 000册 2012年9月北京第1次印刷

著作权合同登记号 图字: 01-2012-2975 号

ISBN 978-7-115-29099-1

定价: 99.00元

读者服务热线: (010)51095186转604 印装质量热线: (010)67129223

反盗版热线: (010)67171154

版 权 声 明

Original English language edition, entitled *Beginning iOS 5 Development: Exploring the iOS SDK* by Dave Mark, Jack Nutting and Jeff LaMarche, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2011 by Dave Mark, Jack Nutting and Jeff LaMarche. Simplified Chinese-language edition copyright © 2012 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。

前言

真是难以置信，现在你手里拿的（或在屏幕上看到的）已经是本书的第4版了。开始这趟旅程的这些年来，我们在本书中倾注了大量的心血、汗水和泪水，Cocoa Touch开发非常奇妙，我们力求竭尽所能地将其特性和魅力展现给开发者。一路走来，我们乐在其中，希望你也一样。

本书进行了全新改版，涵盖了Xcode 4带来的振奋人心的新变化。苹果公司在从Xcode 3过渡到Xcode 4时，重新设计了Xcode的绝大部分内容，同样，随着Xcode的更新（编写本书时为Xcode 4.2），我们也跟着作了修改。本书中的每个项目都从头创建，使用了Xcode 4.2出色的新技术。

当然，正如本书书名所示，所有项目都以iOS 5为中心设计，能在iOS 5系统上正常运行。iOS SDK在iOS 5中变化显著。你应该想到了，项目模板发生了不少变化，对于你经常要做的那些事情又有了许多新方法。当然，还有大量新技术需要掌握。我们新增了两章，分别介绍storyboard和iCloud；新版还涵盖了处理表视图的新策略，使用自动引用计数（ARC）特性重新创建每个示例项目，以简化内存管理。

总之，新版无疑是至今为止内容最为丰富和充实的版本。无论你是iOS开发的初学者还是老手，我们相信你都会喜欢这个版本中所涵盖的新内容。如果你没有读过之前的版本，或者仍然感到有些无所适从，或者你只想支持一下我们，那么一定要购买第4版。我们非常感谢你的支持。记得访问本书的官方社区论坛<http://iphonedevbook.com>^①，告诉我们你开发的新应用。我们期待在论坛上看到你。祝编程愉快！

Dave、Jack和Jeff

① 中文版请访问图灵社区本书网页<http://www.it-ebooks.com.cn/book/948>。——编者注

致 谢

没有我们善良、能干又聪明的家人、朋友和同伴的支持，本书是不可能完成的。首先，感谢Terry、Weronica和Deneen对我们的宽容，他们为我们专心写书提供了非常好的环境。这个项目耗费了相当长的时间，但是他们从未抱怨过。我们很幸运！

另外，还要感谢Apress出版社的工作人员，没有他们，本书不可能出版。Clay Andres策划了本书，并将我们带到了Apress出版社。Dominic Shakeshaft始终笑容可掬地处理我们的抱怨，并总能找到合适的解决方法，使本书更趋完善。本书出色而和蔼的流程编辑Kelly Moritz对我们原本缓慢的写书进程起到了强有力的推动作用。Tom Welsh，我们的开发编辑一路上帮我们解决了相当多的棘手问题。他们让本书的编写能够有条不紊地进行，并为我们指明了正确的方向。Marilyn Smith是极为出色的文字编辑，很荣幸能和他一起工作！Jeffrey Pepper、Frank McGuckin和Brigid Duffy等组成的生产团队把零碎的稿件整合成书。Dylan Wooters征集宣传素材，策划推出一系列营销活动。我们对Apress出版社的所有工作人员表示由衷的感谢！

特别感谢优秀的技术审稿人Mark Dalrymple。除了提供具有独到见解的反馈之外，Mark还测试了书中的所有代码，帮助我们保证了本书的正确性。感谢他！

最后，感谢我们的孩子，他们在父亲努力工作时表现出非常好的耐心。本书是送给他们的：Maddie、Gwynnie、Ian、Kai、Henrietta、Dorotea、Daniel、Kelley和Ryan。

目 录

第 1 章 欢迎来到 iOS 世界	1	第 3 章 处理基本交互	35
1.1 关于本书	1	3.1 MVC 范型	35
1.2 必要条件	1	3.2 创建项目	36
1.2.1 开发者的选择	3	3.3 查看视图控制器	37
1.2.2 必备知识	4	3.3.1 理解输出口和操作	38
1.3 编写 iOS 应用程序有何不同	5	3.3.2 清理视图控制器	41
1.3.1 只有一个应用程序正在运行	5	3.3.3 设计用户界面	41
1.3.2 只有一个窗口	5	3.3.4 测试项目	51
1.3.3 访问受限	5	3.4 理解应用程序委托	51
1.3.4 有限的响应时间	6	3.5 小结	55
1.3.5 有限的屏幕大小	6		
1.3.6 有限的系统资源	6	第 4 章 更丰富的用户界面	56
1.3.7 不支持垃圾收集	7	4.1 满是控件的屏幕	56
1.3.8 新功能	7	4.2 活动、静态和被动控件	58
1.3.9 与众不同的方法	7	4.3 创建应用程序	59
1.4 本书内容	7	4.4 实现图像视图和文本字段	59
1.5 本次更新的内容	9	4.4.1 添加图像视图	59
1.6 准备开始吧	9	4.4.2 调整图像视图	61
		4.4.3 设置视图属性	62
第 2 章 创建基本项目	10	4.4.4 添加文本字段	65
2.1 在 Xcode 中设置项目	10	4.4.5 创建和连接输出口	70
2.1.1 Xcode 工作区窗口	14	4.5 关闭键盘	71
2.1.2 深入研究项目	21	4.5.1 完成输入后关闭键盘	72
2.2 Interface Builder 简介	23	4.5.2 通过触摸背景关闭键盘	73
2.2.1 nib 文件的构成	24	4.5.3 添加滑块和标签	75
2.2.2 库	25	4.5.4 连接操作和输出口	77
2.2.3 在视图中添加标签	26	4.5.5 实现操作方法	77
2.2.4 属性修改	29	4.6 实现开关、按钮和分段控件	78
2.3 美化 iPhone 应用	30	4.6.1 添加两个带标签的开关	79
2.4 小结	34	4.6.2 连接开关输出口和操作	80
		4.6.3 实现开关的操作方法	80

4.6.4 添加按钮	81	6.3.3 修改 BIDSwitchView Controller.h	116
4.6.5 为按钮创建并关联输出口和 操作	82	6.3.4 添加视图控制器	116
4.6.6 实现分段控件的操作方法	82	6.3.5 构建包含工具栏的视图	117
4.7 实现操作表和警报	82	6.3.6 编写根视图控制器	119
4.7.1 遵从操作表委托方法	83	6.3.7 实现内容视图	123
4.7.2 显示操作表	83	6.3.8 制作转换动画	126
4.8 美化按钮	86	6.4 小结	128
4.8.1 viewDidLoad 方法	87	第 7 章 标签栏与选取器	129
4.8.2 控件状态	87	7.1 Pickers 应用程序	130
4.8.3 可拉伸图像	88	7.2 委托和数据源	132
4.9 小结	88	7.3 建立标签栏框架	132
第 5 章 自动旋转和自动调整大小	89	7.3.1 创建文件	133
5.1 自动旋转机制	89	7.3.2 添加根视图控制器	134
5.1.1 点、像素和 Retina 显示屏	90	7.3.3 创建 TabBarController.xib	135
5.1.2 自动转屏方法	91	7.3.4 连接输出口, 然后运行	140
5.2 使用自动调整属性处理旋转	91	7.4 实现日期选取器	141
5.2.1 配置应用支持的方向	91	7.5 实现单组件选取器	144
5.2.2 指定旋转支持	92	7.5.1 声明输出口和操作	144
5.2.3 使用自动调整属性设计界面	93	7.5.2 构建视图	145
5.2.4 大小检查器的自动调整属性	94	7.5.3 将控制器实现为数据源 和委托	146
5.2.5 设置按钮的自动调整属性	96	7.6 实现多组件选取器	149
5.3 在旋转时重构视图	97	7.6.1 声明输出口和操作	150
5.3.1 创建和连接输出口	98	7.6.2 构建视图	150
5.3.2 在旋转时移动按钮	99	7.6.3 实现控制器	150
5.4 切换视图	100	7.7 实现依赖组件	153
5.4.1 设计两个视图	101	7.8 使用自定义选取器创建简单游戏	160
5.4.2 实现交换	102	7.8.1 编写控制器头文件	160
5.4.3 修改输出口集合	104	7.8.2 构建视图	160
5.5 小结	104	7.8.3 添加图像资源	161
第 6 章 多视图应用程序	106	7.8.4 实现控制器	161
6.1 多视图应用程序的常见类型	106	7.8.5 最后的细节	166
6.2 多视图应用程序的体系结构	109	7.8.6 链接 Audio Toolbox 框架	170
6.2.1 根控制器	110	7.9 小结	171
6.2.2 内容视图剖析	111	第 8 章 表视图简介	172
6.3 构建 View Switcher	111	8.1 表视图基础	172
6.3.1 创建视图控制器和 nib 文件	112	8.1.1 表视图和表视图单元	173
6.3.2 修改应用程序委托	114	8.1.2 分组表和无格式表	174

8.2 实现一个简单的表	175	9.2.7 第四个子控制器: 可移动的行	252
8.2.1 设计视图	175	9.2.8 第五个子控制器: 可删除的行	257
8.2.2 编写控制器	176	9.2.9 第六个子控制器: 可编辑的详细窗格	262
8.2.3 添加一个图像	179	9.2.10 其他内容	280
8.2.4 表视图单元样式	181	9.3 小结	282
8.2.5 设置缩进级别	182		
8.2.6 处理行的选择	183	第 10 章 storyboard	284
8.2.7 更改字体大小和行高	185	10.1 创建一个简单的 storyboard	285
8.3 定制表视图单元	186	10.2 动态原型单元	287
8.3.1 向表视图单元添加子视图	186	10.2.1 使用 storyboard 的动态表内容	288
8.3.2 创建 UITableViewCell 子类	187	10.2.2 编辑原型单元	289
8.3.3 从 nib 文件加载 UITableViewCell	192	10.2.3 实现表视图数据源	290
8.4 分组分区和索引分区	197	10.2.4 它会加载吗	292
8.4.1 构建视图	197	10.3 静态单元	293
8.4.2 导入数据	197	10.3.1 实现静态单元	293
8.4.3 实现控制器	198	10.3.2 实现表视图数据源	294
8.4.4 添加索引	201	10.4 大话 segue	296
8.5 实现搜索栏	202	10.4.1 创建 segue 导航	296
8.5.1 重新考虑设计	203	10.4.2 设计 storyboard	297
8.5.2 深层可变副本	203	10.4.3 第一个 segue	299
8.5.3 更新控制器头文件	205	10.4.4 更为实用的任务列表	299
8.5.4 修改视图	206	10.4.5 查看任务详情	300
8.5.5 修改控制器实现	210	10.4.6 设置更多 segue	301
8.6 小结	221	10.4.7 从列表中传递任务	301
第 9 章 导航控制器和表视图	222	10.4.8 处理任务细节	303
9.1 导航控制器	222	10.4.9 回传详细信息	304
9.1.1 栈的性质	222	10.4.10 让列表获取详细信息	305
9.1.2 控制器栈	223	10.4.11 小结	306
9.2 由 6 个部分组成的分层应用程序: Nav	224	第 11 章 iPad 开发注意事项	307
9.2.1 子控制器	225	11.1 分割视图和浮动窗口	307
9.2.2 Nav 应用程序的骨架	228	11.1.1 创建 SplitView 项目	309
9.2.3 向项目中添加图形	234	11.1.2 在 storyboard 中定义结构	310
9.2.4 第一个子控制器: 展示按钮视图	235	11.1.3 代码定义功能	311
9.2.5 第二个子控制器: 校验表	242	11.2 显示总统信息	318
9.2.6 第三个子控制器: 表行上的控件	246	11.3 创建浮动窗口	324

11.4 小结	329	第 14 章 iCloud 之旅	399
第 12 章 应用程序设置和用户默认设置	330	14.1 使用 UIDocument 管理文档存储	399
12.1 设置束	330	14.1.1 构建 TinyPix	400
12.2 AppSettings 应用程序	331	14.1.2 创建 BIDTinyPixDocument	401
12.2.1 创建项目	333	14.1.3 主代码	404
12.2.2 使用设置束	334	14.1.4 初始化 storyboard	410
12.2.3 读取应用程序中的设置	346	14.1.5 创建 BIDTinyPixView	412
12.2.4 注册默认值	350	14.1.6 storyboard 设计	416
12.2.5 更改应用程序中的默认设置	351	14.2 添加 iCloud 支持	419
12.2.6 实现逼真效果	354	14.2.1 创建 provisioning profile	420
12.3 小结	357	14.2.2 启用 iCloud 授权	420
第 13 章 保存数据	358	14.2.3 如何查询	421
13.1 应用程序的沙盒	358	14.2.4 保存在哪里	423
13.1.1 获取 Documents 目录	360	14.2.5 在 iCloud 上存储首选项	423
13.1.2 获取 tmp 目录	360	14.3 小结	424
13.2 文件保存策略	361	第 15 章 Grand Central Dispatch、后台处理及其应用	426
13.2.1 单个文件持久性	361	15.1 Grand Central Dispatch	426
13.2.2 多个文件持久性	361	15.2 SlowWorker 简介	427
13.3 属性列表	361	15.3 线程基础知识	430
13.3.1 属性列表序列化	362	15.4 工作单元	430
13.3.2 持久性应用程序的第一个版本	363	15.5 GCD: 低级队列	431
13.4 对模型对象进行归档	368	15.5.1 傻瓜式操作	431
13.4.1 符合 NSCoder	368	15.5.2 改进 SlowWorker	432
13.4.2 实现 NSCopying	369	15.6 后台处理	438
13.4.3 对数据对象进行归档和取消归档	370	15.6.1 应用程序生命周期	439
13.4.4 归档应用程序	371	15.6.2 状态更改通知	439
13.5 使用 iOS 的嵌入式 SQLite3	374	15.6.3 创建 State Lab	441
13.5.1 创建或打开数据库	375	15.6.4 执行状态	442
13.5.2 绑定变量	376	15.6.5 利用执行状态更改	444
13.5.3 SQLite3 应用程序	377	15.6.6 处理不活动状态	444
13.6 使用 Core Data	383	15.6.7 处理后台状态	449
13.6.1 实体和托管对象	385	15.7 小结	457
13.6.2 Core Data 应用程序	388	第 16 章 使用 Quartz 和 OpenGL 绘图	458
13.7 小结	398	16.1 图形世界的两个视图	458
		16.2 Quartz 2D 绘图方法	459

16.2.1 Quartz 2D 的图形上下文.....	459	第 18 章 Core Location 定位功能.....	517
16.2.2 坐标系.....	460	18.1 位置管理器.....	517
16.2.3 指定颜色.....	461	18.1.1 设置所需的精度.....	518
16.2.4 在上下文中绘制图像.....	463	18.1.2 设置距离筛选器.....	518
16.2.5 绘制形状: 多边形、直线和 曲线.....	463	18.1.3 启动位置管理器.....	519
16.2.6 Quartz 2D 工具采样器: 模 式、梯度、虚线模式.....	464	18.1.4 更明智地使用位置管理器.....	519
16.3 QuartzFun 应用程序.....	465	18.2 位置管理器委托.....	519
16.3.1 构建 QuartzFun 应用程序.....	465	18.2.1 获取位置更新.....	519
16.3.2 添加 Quartz Drawing 代码.....	474	18.2.2 使用 CLLocation 获取纬度 和经度.....	519
16.3.3 优化 QuartzFun 应用程序.....	478	18.2.3 错误通知.....	521
16.4 GLFun 应用程序.....	481	18.3 尝试使用 Core Location.....	522
16.4.1 构建 GLFun 应用程序.....	482	18.3.1 更新位置管理器.....	525
16.4.2 创建 BIDGLFunView.....	482	18.3.2 确定移动距离.....	526
16.4.3 更新 BIDViewController.....	489	18.4 小结.....	527
16.4.4 更新 nib.....	490	第 19 章 陀螺仪和加速计.....	528
16.4.5 完成 GLFun.....	490	19.1 加速计物理学.....	528
16.5 小结.....	490	19.2 不要忘记旋转.....	529
第 17 章 轻击、触摸和手势.....	491	19.3 Core Motion 和动作管理器.....	529
17.1 多点触控术语.....	491	19.3.1 基于事件的动作.....	530
17.2 响应者链.....	492	19.3.2 主动动作访问.....	535
17.2.1 响应事件.....	492	19.3.3 加速计结果.....	537
17.2.2 转发事件: 保持响应者链的 活动状态.....	493	19.4 检测摇动.....	537
17.3 多点触控体系结构.....	494	19.4.1 Baked-In 摇动.....	538
17.4 4 个手势通知方法.....	494	19.4.2 摇动与击碎.....	539
17.5 检测触摸.....	495	19.5 将加速计用做方向控制器.....	544
17.6 检测轻扫.....	498	19.5.1 滚弹珠程序.....	545
17.6.1 使用自动手势识别.....	502	19.5.2 编写 Ball View.....	547
17.6.2 实现多个轻扫动作.....	503	19.5.3 计算小球运动.....	549
17.7 检测多次轻击.....	505	19.6 小结.....	552
17.8 检测捏合操作.....	509	第 20 章 iPhone 照相机和照片库.....	553
17.9 创建和使用自定义手势.....	512	20.1 使用图像选取器和 UIImagePickerController.....	553
17.9.1 CheckPlease 应用程序.....	512	20.2 实现图像选取器控制器委托.....	555
17.9.2 CheckPlease 触摸方法.....	514	20.3 实际测试照相机和库.....	556
17.10 小结.....	516	20.3.1 设计界面.....	557
		20.3.2 实现照相机视图控制器.....	558

20.4 小结	562	21.3.6 本地化应用程序显示名称	577
第 21 章 应用程序本地化	563	21.4 小结	578
21.1 本地化体系结构	563	第 22 章 未来之路	579
21.2 字符串文件	564	22.1 苹果公司的文档	579
21.2.1 字符串文件里面是什么	565	22.2 邮件列表	579
21.2.2 本地化的字符串宏	565	22.3 论坛	580
21.3 现实中的 iOS：本地化应用程序	566	22.4 网站	580
21.3.1 创建 LocalizeMe	567	22.5 博客	581
21.3.2 测试 LocalizeMe	569	22.6 会议	582
21.3.3 本地化 nib 文件	570	22.7 作者	582
21.3.4 本地化图像	573	22.8 再会	583
21.3.5 生成和本地化字符串文件	575		

第 1 章

欢迎来到iOS世界



你想编写iPhone、iPod touch和iPad应用程序？哦，这事也怪不得你。事实上，所有这些设备最核心的软件——iOS，恐怕要算是长久以来最吸引人的新平台了，自2007年发布以来，iOS发展异常迅速。移动应用平台的崛起意味着人们无时无刻不在使用各种软件。随着iOS 5以及最新iOS软件开发工具包（SDK）的发布，iOS应用开发变得更加高效和有趣。

1.1 关于本书

本书将带你走上创建iOS应用程序的大道。我们的目标是让你通过初步学习，理解iOS应用程序的运行和构建方式。

在学习过程中，你将创建一系列小型应用程序，每个应用程序都会突出某些iOS特性，展示如何控制这些特性或与其交互。如果你扎实地掌握了本书中的基本知识，充分发挥自己的创造力，并且坚定不移，同时借助苹果公司大量翔实的文档，你就完全可以创建出专业级的iPhone和iPad应用程序。

说明 Dave、Jack和Jeff为本书创办了一个论坛。志同道合的开发人员可以在此相互交流，搞懂一些问题，并且还可以回答别人提出的问题。论坛地址为<http://iphonedevbook.com>。一定要访问此论坛哦！

1.2 必要条件

在开始编写iOS软件之前，需要做一些准备工作。初学者需要一台运行Lion（OS X 10.7或更高版本）的基于Intel架构的Macintosh计算机。任何最近上市的基于Intel架构的Macintosh计算机（不管是笔记本电脑还是台式机）应该都符合要求。

你还需要注册成为iOS开发人员。只有完成这一步，苹果公司才允许下载iOS SDK。

注册请访问<http://developer.apple.com/ios/>，这会打开如图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。如果你还没有Apple ID，请单击Create

Apple ID按钮创建一个, 然后再登录。登录之后, 将进入iOS开发主页面。其中不仅有SDK的下载链接, 还有各类文档、视频和示例代码等的链接, iOS应用开发人员可通过这些内容获得详尽的指导。



图1-1 苹果公司的iOS开发中心网站

这个页面上一个最重要的工具是Xcode，它是苹果公司的IDE（集成开发环境），必须下载。Xcode提供了各种实用工具，用于创建和调试源代码、编译应用程序以及调优应用程序性能。

完成注册后就能在<http://developer.apple.com/ios/>页面上找到Xcode的下载链接。也可以从Macintosh的App Store下载Xcode（可以通过Mac的Apple菜单来访问App Store）。

示例所用的SDK版本和源代码

随着SDK和Xcode版本的不断发展，下载它们的机制也将会改变。有时SDK和Xcode需要分开下载，有时它们可以合在一起下载。基本上，你应该下载最新发布的Xcode和iOS SDK版本（非beta版）。

本书是针对最新的SDK版本编写的。在某些地方，我们选择使用iOS 5中引入的新函数或方法，它们可能与早期的SDK版本不兼容，出现这些情况时我们一定会指出。

请从本书网站<http://iphonedevbook.com>或者Apress（<http://apress.com>）本书页面上下载最新的源代码。在发布SDK新版本时，我们将相应地更新代码，所以一定要定期查看该网站。

1.2.1 开发者的选择

这个可免费下载的SDK还包含一个模拟器，它支持在Mac上创建和运行iPhone与iPad应用程序。这对于学习编写iOS程序极其有用。但是，模拟器不支持依赖于硬件的某些特性，如加速计或摄像功能。同时模拟器还不支持将应用程序下载到实际的iPhone或其他设备中。此外，它也不能在苹果公司的App Store上分发应用程序。要实现这些功能，需要注册并下载使用另外两个方案中的一个，它们不是免费的。

- 标准方案的价格为99美元/年，它提供了大量开发工具、资源和技术支持，支持通过苹果公司的App Store分发应用程序。最重要的是，标准方案支持在iOS设备上（而不只是在模拟器上）测试和调试代码。
- 企业方案的价格为299美元/年，面向开发专用的、内部的iOS应用程序的企业，以及为苹果公司的App Store开发应用程序且拥有参与该项目的多名开发人员的企业。

有关这两种方案的详细信息，请访问<http://developer.apple.com/programs/ios>，以及<http://developer.apple.com/programs/ios/enterprise>。

由于iOS设备是一种始终连网的移动设备，并且使用的是其他公司的无线基础设施，因此苹果公司对iOS开发人员的限制比对Mac开发人员严格得多（Mac开发人员无需经过苹果公司的审查或批准就能够编写和分发程序）。尽管iPod touch和那种只能使用Wi-Fi的iPad不允许使用其他的基础设施，它们仍然会面临同样的限制。

苹果公司添加这些限制，主要是为了尽量避免分发恶意或效率低下的程序，因为这类程序可能降低共享网络的性能。开发iOS应用程序似乎有很高的门槛，但苹果公司在简化开发过程方面也付出了巨大努力。还应该提及的是，99美元的价格比微软公司的软件开发IDE（比如像Visual Studio）的价格低得多。

另外,很明显,你还需要一部iPhone、iPod touch或iPad。虽然大部分代码都可以通过iOS模拟器进行测试,但并非所有程序都是如此。模拟器上运行的一些应用程序需要在实际的设备上全面测试,然后才能分发给公众。

说明 如果已决定注册标准版或企业版方案,应该立即注册。批准过程可能需要一些时间,并且批准之后才能在真实的设备上运行应用程序。但是不必担心,前几章中的所有项目以及本书中的大多数应用程序,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。你应该理解面向对象编程的基础知识,例如,了解对象、循环和变量的含义,还应该熟悉Objective-C编程语言。SDK中的Cocoa Touch是本书使用的主要工具,它使用的是最新版的Objective-C,包含了之前的版本中都没有出现过的多种新特性。但是如果不了解Objective-C的新增特性也没有关系。我们将重点介绍要使用的Objective-C的新语言特性,并解释其工作原理和使用它的原因。

作为用户,你还应该熟悉iOS系统本身。就像在任何其他平台中编写应用程序一样,你需要熟悉iPhone、iPad或iPod touch的各种特性,并了解iOS界面以及iPhone和iPad应用程序的外观。

Objective-C的学习资源

如果你从未使用Objective-C编写过程序,那么以下资源有助于你了解该语言。

- 阅读Mac编程专家Mark Dalrymple和Scott Knaster撰写的《Objective-C基础教程》^①,该书浅显易懂,是学习Objective-C基础知识的优秀图书。网址如下所示。<http://www.apress.com/book/view/9781430218159>
- 参考Apple公司在*Learning Objective-C: A Primer*中对该语言的介绍:
http://developer.apple.com/library/ios/#referencelibrary/GettingStarted/Learning_Objective-C_A_Primer
- 阅读*Objective-C Programming Language*,其中对该语言的介绍非常详尽全面,是一个上乘的参考指南,从以下网站可以获取该书。

<http://developer.apple.com/library/ios/#documentation/Cocoa/Conceptual/ObjectiveC>

最后一个资源可以从iPhone、iPod touch或iPad的iBooks免费下载。有了这些资源就完美了,可以随时随地阅读了!苹果公司已经发布了此种形式的好几种开发资料,希望还有更多惊喜!在iBooks中搜索“Apple developer publications”可以找到这些内容。

^① 请访问<http://www.it-ebooks.com.cn/book/303>了解本书信息。——编者注

1.3 编写 iOS 应用程序有何不同

如果从未使用过Cocoa或它的前期产品NeXTSTEP和OpenStep,那么你可能会发现Cocoa Touch(用于编写iOS应用程序的应用程序框架)稍显另类。它与其他常用应用程序框架(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。你起初可能会有点不知所措,但不必担心,只要勤加练习,就可以掌握其中的规律。

如果你具备使用Cocoa或NeXTSTEP编程的经验,就会发现iOS SDK中有许多熟悉的身影。其中的许多类都是从用于Mac OS X开发的版本中原样借鉴过来的,一些类即便不同,也遵循相同的基本原则,并使用与旧版本类似的设计模式。但是,Cocoa和Cocoa Touch之间却存在一些差异。

无论你的知识背景如何,都需要谨记iOS开发与桌面应用程序开发之间的重要差异。后续几节将讨论这些差异。

1.3.1 只有一个应用程序正在运行

在iOS上,每一时间段内只能激活一个应用程序并在屏幕上显示。从iOS 4开始,在用户按下home按钮后,应用程序可以继续后台运行,但是这种情况也只限于特定的场合,而且必须为此编写特定的代码。

当应用程序未激活或运行于后台时,它不会占用任何CPU资源,这将导致网络连接断开以及其他问题。iOS 5在后台处理能力方面已经有了大幅提升,但要使应用程序在这种情况下仍运行良好,还需要你自身多加努力。

1.3.2 只有一个窗口

在台式及笔记本电脑操作系统中,多个程序可以同时运行,并且可以分别创建和控制多个窗口。而iOS则有所不同,它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,而且这个窗口的大小就是iOS设备屏幕的大小,是固定的。

1.3.3 访问受限

计算机上的程序可以访问其用户能够访问的任何内容,而iOS则严格限制了应用程序的访问权限。

你只能在iOS为应用程序创建的文件系统中读写文件。此区域称为应用程序的沙盒,应用程序在其中存储文档、首选项等需要存储的各种数据。

应用程序还存在其他方面的限制。举例来说,你不能访问iOS上端口号较小的网络端口,也不能执行台式计算机中需要有根用户或管理员权限才能执行的操作。