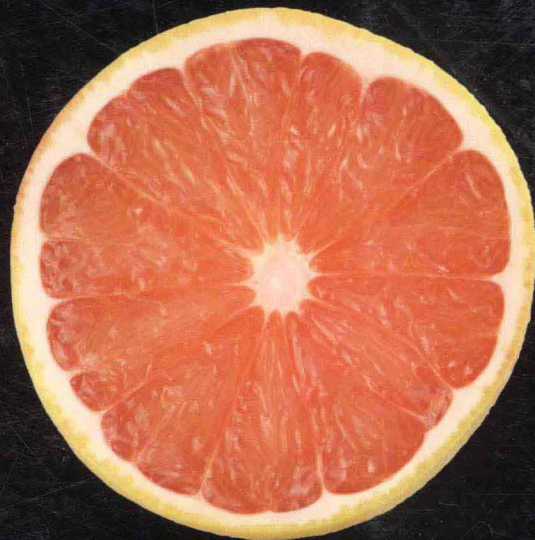


TURING

图灵程序设计丛书 移动开发系列

Apress®

- 2009年第1版面世以来，累计 **销量超过5万册**
- 全球iOS开发者交口称赞的iOS开发圣经，新版本经过重新翻译审校
- 资深技术专家担纲撰写，iOS开发必备



Beginning iOS 6 Development Exploring the iOS SDK

# 精通iOS开发

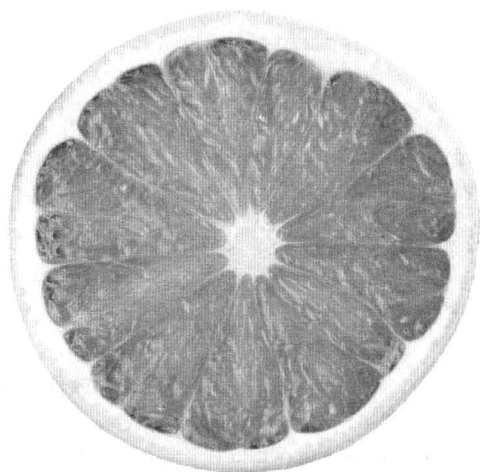
(第5版)

[美] David Mark [瑞典] Jack Nutting 著  
[美] Jeff LaMarche [法] Fredrik Olsson  
邓强 武海峰 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 6 Development Exploring the iOS SDK

# 精通iOS开发

(第5版)

[美] David Mark [瑞典] Jack Nutting 著  
[美] Jeff LaMarche [法] Fredrik Olsson  
邓强 武海峰 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5版 / (美) 马克 (Mark, D.) 等著 ;  
邓强, 武海峰译. -- 北京 : 人民邮电出版社, 2013.9

(图灵程序设计丛书)

书名原文: Beginning ios 6 development:  
exploring the ios sdk

ISBN 978-7-115-32761-1

I. ①精… II. ①马… ②邓… ③武… III. ①移动电  
话机—应用程序—程序设计 IV. ①TN929.53

中国版本图书馆CIP数据核字(2013)第184303号

## 内 容 提 要

iOS 是如今最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 应用开发的基础教程, 内容翔实、语言生动。几位作者结合大量实例, 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 6 操作系统, 涵盖了 Xcode 4 以来的新功能, 其中最值得注意的便是集合视图, 本书将以全新章节详细讲述。全书所有项目均使用 Xcode 4 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, iOS 开发新手可通过学习本书快速入门进阶, 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  |
|                | [美] Jeff LaMarche   | [法] Fredrik Olsson |
| 译              | 邓 强 武海峰   |                    |
| 责任编辑           | 刘美英   |                    |
| 执行编辑           | 李 静 毛倩倩   |                    |
| 责任印制           | 焦志炜   |                    |
| ◆ 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             | <a href="http://www.ptpress.com.cn">http://www.ptpress.com.cn</a> |                    |
|                | 三河市海波印务有限公司印刷   |                    |
| ◆ 开本: 800×1000 | 1/16  |                    |
| 印张: 37.5       |   |                    |
| 字数: 886千字      | 2013年9月第1版  |                    |
| 印数: 1-4 000册   | 2013年9月河北第1次印刷  |                    |
|                | 著作权合同登记号  | 图字: 01-2013-3866号  |
- 

定价: 109.00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

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 iOS 6 Development: Exploring the iOS SDK* by David Mark, Jack Nutting, Jeff LaMarche, Fredrik Olsson, published by Apress, 2855 Telegraph Avenue, Suite 600, Berkeley, CA 94705 USA.

Copyright © 2013 by David Mark, Jack Nutting, Jeff LaMarche, Fredrik Olsson. Simplified Chinese-language edition copyright © 2013 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。

# 献 词

谨以此书怀念史蒂夫·乔布斯。他的精神和远见将继续鼓舞我们。

——David Mark和Jack Nutting

献给我的父母，感谢你们送我的第一台计算机。

——Fredrik Olsson

欢迎加入

**图灵社区** ituring.com.cn

## ——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。

**优惠提示：**现在购买电子书，读者将获赠书款20%的社区银子，可用于兑换纸质样书。

## ——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的梦想。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作文章、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、乐译、评选等多种活动，赢取积分和银子，积累个人声望。

# 目 录

第 1 章 欢迎来到 iOS 世界	1	第 3 章 实现基本交互	35
1.1 关于本书	1	3.1 MVC 方法	36
1.2 必要条件	1	3.2 创建项目	36
1.2.1 开发者选项	3	3.3 视图控制器	37
1.2.2 必备知识	3	3.3.1 输出口和动作	38
1.3 编写 iOS 应用程序有何不同	4	3.3.2 清理视图控制器	41
1.3.1 只能激活一个应用	4	3.3.3 设计用户界面	41
1.3.2 只有一个窗口	5	3.3.4 运行应用	50
1.3.3 有限的访问权限	5	3.3.5 样式文本	50
1.3.4 有限的响应时间	5	3.4 应用程序委托	52
1.3.5 有限的屏幕大小	5	3.5 小结	55
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 不支持垃圾回收	6	4.1 满是控件的屏幕	56
1.3.8 新功能	6	4.2 活动控件、静态控件和被动控件	58
1.3.9 与众不同的交互方法	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 这一版的新内容	8	4.4.1 添加图像视图	59
1.6 准备开始吧	9	4.4.2 调整图像视图的大小	62
第 2 章 创建项目	10	4.4.3 设置视图属性	63
2.1 在 Xcode 中创建项目	10	4.4.4 添加文本字段	65
2.1.1 Xcode 工作区窗口	14	4.4.5 创建和关联输出口	70
2.1.2 深入研究项目	21	4.5 关闭键盘	72
2.2 Interface Builder 简介	23	4.5.1 输入完成后关闭键盘	73
2.2.1 nib 文件的构成	24	4.5.2 通过触摸背景关闭键盘	74
2.2.2 库	25	4.5.3 添加滑块和标签	76
2.2.3 在视图中添加标签	26	4.5.4 添加顶部约束	78
2.2.4 属性修改	29	4.5.5 创建并关联动作方法和输出口	78
2.3 画龙点睛——美化 iPhone 应用	30	4.5.6 实现动作方法	79
2.4 小结	34		

4.6 实现开关、按钮和分段控件	80	6.3.4 添加视图控制器	124
4.7 实现分段控件的动作方法	84	6.3.5 构建带有工具栏的视图	126
4.8 实现操作表单和警告视图	85	6.3.6 编写根视图控制器	128
4.8.1 遵从操作表单委托方法	85	6.3.7 实现内容视图	131
4.8.2 显示操作表单	85	6.3.8 转换过程的动画效果	134
4.9 美化按钮	88	6.4 小结	136
4.9.1 viewDidLoad 方法	89	<b>第 7 章 标签栏与选取器</b>	137
4.9.2 控件状态	90	7.1 Pickers 应用程序	138
4.9.3 可拉伸图像	90	7.2 委托和数据源	140
4.10 小结	91	7.3 建立标签栏框架	141
<b>第 5 章 自动旋转和自动调整大小</b>	92	7.3.1 创建文件	141
5.1 自动旋转机制	93	7.3.2 添加根视图控制器	142
5.1.1 点、像素和 Retina 显示屏	93	7.3.3 创建 TabBarController.xib	143
5.1.2 自动旋转的实现方式	94	7.3.4 初次运行	149
5.2 选择视图支持的方向	95	7.4 实现日期选取器	150
5.2.1 应用级支持的方向	95	7.5 实现单组件选取器	153
5.2.2 单个控制器的旋转支持	96	7.5.1 声明输出和动作方法	153
5.3 使用约束设计界面	98	7.5.2 构建视图	153
5.3.1 覆盖默认的约束	100	7.5.3 将控制器实现为数据源和委托	155
5.3.2 与屏幕等宽的按钮	101	7.6 实现多组件选取器	158
5.4 旋转时重构视图	103	7.6.1 声明输出和动作方法	158
5.4.1 创建并关联输出	104	7.6.2 构建视图	159
5.4.2 旋转时移动按钮	105	7.6.3 实现控制器	159
5.5 切换视图	106	7.7 实现依赖组件	162
5.5.1 设计两个视图	108	7.8 使用自定义选取器创建一个简单游戏	168
5.5.2 实现视图切换	109	7.8.1 编写控制器头文件	168
5.5.3 修改输出集合	111	7.8.2 构建视图	169
5.6 小结	112	7.8.3 添加图像资源	170
<b>第 6 章 多视图应用程序</b>	113	7.8.4 实现控制器	170
6.1 多视图应用程序的常见类型	113	7.8.5 最后的细节	173
6.2 多视图应用程序的体系结构	117	7.8.6 链接 Audio Toolbox 框架	177
6.2.1 根控制器	119	7.9 小结	178
6.2.2 内容视图剖析	119	<b>第 8 章 表视图简介</b>	179
6.3 UINavigationController	120	8.1 表视图基础	179
6.3.1 创建视图控制器和 nib 文件	121	8.1.1 表视图和表视图单元	180
6.3.2 修改应用程序委托	123	8.1.2 分组表和无格式表	180
6.3.3 修改 BIDSwitchView-Controller.h	124		



8.2 实现一个简单表	181	9.2.10 其他内容	272
8.2.1 设计视图	182	9.3 小结	273
8.2.2 编写控制器	182	<b>第 10 章 故事板</b>	274
8.2.3 添加一个图像	186	10.1 创建一个简单的故事板	274
8.2.4 表视图单元样式	187	10.2 动态原型单元	277
8.2.5 设置缩进级别	189	10.2.1 使用故事板的动态表内容	277
8.2.6 处理行的选择	190	10.2.2 编辑原型单元	278
8.2.7 更改字体大小和行高	192	10.2.3 实现表视图数据源	279
8.3 定制表视图单元	193	10.2.4 它会加载吗	281
8.3.1 向表视图单元添加子视图	193	10.3 静态单元	282
8.3.2 创建 UITableViewCell 子类	194	10.3.1 实现静态单元	282
8.3.3 从 nib 文件加载 UITableViewCell	199	10.3.2 实现表视图数据源	283
8.4 分组分区和索引分区	202	10.4 联线	285
8.4.1 构建视图	202	10.4.1 创建联线导航	285
8.4.2 导入数据	203	10.4.2 设计故事板	286
8.4.3 实现控制器	204	10.4.3 第一个视图转换	287
8.4.4 添加索引	207	10.4.4 更为实用的任务列表	288
8.5 实现搜索栏	208	10.4.5 查看任务详情	289
8.6 小结	215	10.4.6 设置更多联线	290
<b>第 9 章 导航控制器和表视图</b>	216	10.4.7 从列表中传递任务	291
9.1 导航控制器	216	10.4.8 处理任务细节	292
9.1.1 栈的性质	216	10.4.9 回传详细信息	293
9.1.2 控制器栈	217	10.4.10 让列表获取详细信息	294
9.2 由 6 个部分组成的分层应用程序:		10.4.11 小结	295
Nav	218	<b>第 11 章 iPad 开发注意事项</b>	296
9.2.1 子控制器	219	11.1 分割视图和浮动窗口	296
9.2.2 Nav 应用程序骨架	223	11.1.1 创建 SplitView 项目	298
9.2.3 向项目中添加图像	229	11.1.2 在故事板中定义结构	300
9.2.4 第一个子控制器: 展开按钮 视图	229	11.1.3 使用代码定义功能	301
9.2.5 第二个子控制器: 检查表	237	11.2 显示总统信息	307
9.2.6 第三个子控制器: 表行上的 控件	242	11.3 创建浮动窗口	313
9.2.7 第四个子控制器: 可移动的 行	248	11.4 小结	319
9.2.8 第五个子控制器: 可删除的 行	252	<b>第 12 章 应用程序设置及用户默认   设置</b>	320
9.2.9 第六个子控制器: 可编辑的 详细面板	257	12.1 设置捆绑包入门	320
		12.2 AppSettings 应用程序	321
		12.2.1 创建项目	324
		12.2.2 使用设置捆绑包	325

12.2.3	读取应用程序中的设置	338	14.1.5	创建 BIDTinyPixView 类	400
12.2.4	在应用程序中修改默认设置	342	14.1.6	设计详细视图	403
12.2.5	注册默认值	344	14.2	添加 iCloud 支持	407
12.2.6	保证设置有效	345	14.2.1	创建 provisioning profile	408
12.3	小结	348	14.2.2	启用 iCloud 授权	409
第 13 章	数据持久化基础知识	349	14.2.3	如何查询	409
13.1	应用程序的沙盒	349	14.2.4	保存在哪里	411
13.1.1	获取 Documents 目录	350	14.2.5	将首选项保存到 iCloud	411
13.1.2	获取 tmp 目录	351	14.3	小结	412
13.2	文件保存策略	351	第 15 章	Grand Central Dispatch 和后台处理	414
13.2.1	单文件持久化	352	15.1	Grand Central Dispatch	414
13.2.2	多文件持久化	352	15.2	SlowWorker 简介	415
13.3	属性列表	352	15.3	线程基础知识	418
13.3.1	属性列表序列化	352	15.4	工作单元	419
13.3.2	Persistence 应用程序的第一个版本	353	15.5	GCD: 低级队列	419
13.4	对模型对象进行归档	358	15.5.1	傻瓜式操作	420
13.4.1	遵循 NSCoder 协议	358	15.5.2	改进 SlowWorker	421
13.4.2	实现 NSCopying 协议	360	15.6	后台处理	426
13.4.3	对数据对象进行归档和取消归档	360	15.6.1	应用程序生命周期	427
13.4.4	归档应用程序	361	15.6.2	状态更改通知	427
13.5	使用 iOS 内嵌的 SQLite3	364	15.6.3	创建 State Lab 项目	428
13.5.1	创建或打开数据库	365	15.6.4	执行状态的变化	430
13.5.2	绑定变量	366	15.6.5	利用执行状态更改	431
13.5.3	SQLite3 应用程序	367	15.6.6	处理不活动状态	432
13.6	使用 Core Data	373	15.6.7	处理后台状态	436
13.6.1	实体和托管对象	374	15.7	小结	443
13.6.2	Core Data 应用程序	378	第 16 章	使用 Quartz 和 OpenGL 绘图	444
13.7	小结	386	16.1	图形世界的两个视图	444
第 14 章	iCloud 之旅	387	16.2	Quartz 2D 绘图方法	445
14.1	使用 UIDocument 管理文档存储	387	16.2.1	Quartz 2D 的图形上下文	445
14.1.1	构建 TinyPix	388	16.2.2	坐标系	446
14.1.2	创建 BIDTinyPix-Document 类	388	16.2.3	指定颜色	447
14.1.3	主代码	391	16.2.4	在上下文中绘制图像	449
14.1.4	初始故事板	397	16.2.5	绘制形状: 多边形、直线和曲线	449
			16.2.6	Quartz 2D 工具采样器: 模式、梯度、虚线模式	450

16.3 QuartzFun 应用程序.....451	18.2.1 获取位置更新.....508
16.3.1 建立 QuartzFun 应用程序.....451	18.2.2 使用 CLLocation 获取纬度和经度.....508
16.3.2 添加 Quartz 2D 的绘图代码.....460	18.2.3 错误通知.....510
16.3.3 优化 QuartzFun 应用程序.....465	18.3 试用 Core Location.....511
16.4 GLFun 应用程序.....468	18.4 将移动路线展现在地图上.....516
16.4.1 建立 GLFun 应用程序.....468	18.5 小结.....519
16.4.2 创建 BIDGLFunView 类.....469	
16.4.3 更新 BIDView-Controller 类.....475	<b>第 19 章 陀螺仪和加速计.....520</b>
16.4.4 更新 nib 文件.....476	19.1 加速计物理学.....520
16.4.5 完成 GLFun 应用程序.....476	19.2 不要忘记旋转.....521
16.5 小结.....476	19.3 Core Motion 和动作管理器.....521
<b>第 17 章 轻击、触摸和手势.....477</b>	19.3.1 基于事件的动作.....522
17.1 多点触控术语.....477	19.3.2 主动动作访问.....527
17.2 响应者链.....478	19.3.3 加速计结果.....529
17.2.1 响应事件.....478	19.4 检测摇动.....530
17.2.2 转发事件：保持响应者链的活动状态.....479	19.4.1 内嵌的摇动检测.....531
17.3 多点触控体系结构.....480	19.4.2 摇动与击碎.....531
17.4 4 个手势通知方法.....480	19.5 将加速计用做方向控制器.....536
17.5 TouchExplorer 应用程序.....481	19.5.1 滚弹珠程序.....537
17.6 Swipes 应用程序.....485	19.5.2 实现 BIDBallView 类.....539
17.6.1 自动手势识别.....488	19.5.3 计算弹珠运动.....542
17.6.2 实现多指滑动.....489	19.6 小结.....544
17.7 检测多次轻击.....491	<b>第 20 章 照相机和照片库.....545</b>
17.8 检测捏合和旋转.....496	20.1 图像选取器和 UIImagePickerController-Controller.....545
17.9 自定义手势.....500	20.2 实现图像选取器控制器委托.....547
17.9.1 CheckPlease 应用程序.....501	20.3 实际测试照相机和照片库.....548
17.9.2 CheckPlease 的触摸方法.....503	20.3.1 设计界面.....549
17.10 小结.....505	20.3.2 实现照相机视图控制器.....550
<b>第 18 章 Core Location 和 Map Kit.....506</b>	20.4 小结.....553
18.1 位置管理器.....506	<b>第 21 章 集合视图.....554</b>
18.1.1 设置精度.....507	21.1 创建 DialogViewer 项目.....554
18.1.2 设置距离筛选器.....507	21.2 视图控制器类.....555
18.1.3 启动位置管理器.....508	21.3 自定义单元.....555
18.1.4 合理使用位置管理器.....508	21.4 配置视图控制器.....558
18.2 位置管理器委托.....508	21.5 内容单元.....559
	21.6 实现流式布局.....560
	21.7 分区标题视图.....562

21.8 小结 .....	563	22.3.2 测试 LocalizeMe .....	570
<b>第 22 章 应用程序本地化</b> .....	<b>564</b>	22.3.3 本地化 nib 文件 .....	571
22.1 本地化体系结构 .....	564	22.3.4 本地化图像 .....	575
22.2 字符串文件 .....	565	22.3.5 生成并本地化字符串文件 .....	577
22.2.1 字符串文件 .....	566	22.3.6 Base Internationalization .....	579
22.2.2 本地化的字符串宏 .....	566	22.3.7 应用显示名称的本地化 .....	582
22.3 现实中的 iOS: 本地化应用程序 .....	567	22.4 小结 .....	583
22.3.1 创建 LocalizeMe .....	568	<b>附录 A 未来之路</b> .....	<b>584</b>

# 欢迎来到iOS世界



这么说，你想编写iPhone、iPod touch和iPad应用程序？哦，当然可以。作为所有苹果设备的核心软件，iOS从2007年发布以来发展异常迅速，是个极具吸引力的平台。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 6以及最新iOS软件开发工具包（SDK）的发布，开发iOS应用正变得更加高效、更加有趣。

## 1.1 关于本书

本书是iOS应用编程的入门指南，旨在帮助你克服入门的困难，帮助你理解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，如何与其交互。如果你扎实地掌握了本书的基础知识，充分发挥自己的创造力，并且持之以恒，再借助大量条理清晰的苹果文档，你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。志趣相投者可会聚于此，相互答疑解惑。一定要访问这个论坛哦！地址是<http://forum.learncocoa.org>。

---

## 1.2 必要条件

开始编写iOS应用程序之前，需要做一些准备工作。初学者需要一台安装了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（台式机或笔记本）均可。

读者还需要注册成为iOS开发者。只有完成这一步，苹果才允许下载iOS SDK。

如果要注册成为开发者，可以访问<http://developer.apple.com/ios/>。然后会看到类似图1-1所示的页面。

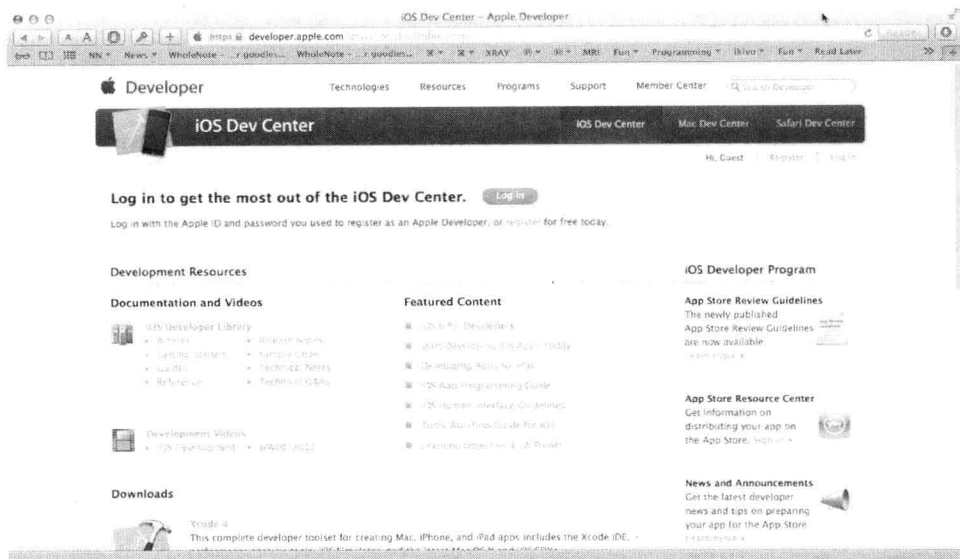


图1-1 苹果的iOS开发中心网站

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Join now创建一个，然后再登录。登录之后就进入了iOS开发中心的主页面。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，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，它是苹果的集成开发环境（IDE）。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。

可以从Mac App Store下载Xcode（可以通过Mac的Apple菜单访问App Store）。

### 书中示例使用的SDK版本和源代码

随着SDK和Xcode版本的不断更新，它们的下载方式也在发生变化。从Xcode 4.3开始，苹果就开始把当前“稳定”版本的Xcode和iOS SDK放在Mac App Store中了，同时也会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（非beta版），应该使用Mac App Store。

本书面向最新版SDK。在一些地方，我们会使用iOS 6中引入的新函数或方法，它们可能与旧版SDK不兼容。当然，出现这些情况时我们会特别指出。

请从<http://iphonedevbook.com>或者本书论坛<http://forum.learncooca.org>上下载最新的源代码<sup>①</sup>。当有新版SDK发布时，我们会及时更新代码，所以你应该定期登录这些网站，看看是否有更新内容。

<sup>①</sup> 本书源代码也可从图灵社区本书页面“随书下载”部分下载，网址是<http://www.ituring.com.cn/book/1120>。

### 1.2.1 开发者选项

免费下载的SDK中包含一个模拟器，使用这个模拟器，就可以在Mac上创建并运行iPhone和iPad应用。这个模拟器对于学习编写iOS程序极其有用。但是，模拟器不支持那些依赖于硬件的特性，比如加速计和摄像头。另外，免费下载的SDK不支持把应用程序下载到实际的iPhone等设备中，也不能在苹果的App Store上出售应用。如果想要这些功能，需要注册一个付费选项。

- 标准版计划的价格为99美元/年。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）测试和调试代码。
- 企业版计划的价格为299美元/年。如果需要开发公司内部使用的私有iOS应用程序，可以选择这个计划。

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）来查看这两个计划的详细信息。

由于iOS支持始终联网的移动设备（比如iPhone），这种设备使用的是其他公司的无线基础设施，因此苹果对iOS开发人员的限制比对Mac开发人员严格得多（目前，Mac开发人员无需经过苹果的审查或批准就可以编写并销售程序）。虽然iPod touch和那种只支持Wi-Fi的iPad不使用其他公司的基础设施，但是它们也受到同样的限制。

苹果添加这些限制，主要是为了尽量避免发布恶意程序和不良程序，因为这类程序可能会降低共享网络的性能。开发iOS应用似乎有很高的门槛，但是苹果也为简化开发过程做出了巨大努力。值得一提的是，99美元比Visual Studio（微软公司的软件开发IDE）的价格低得多。

显然，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，最好马上注册。因为审批过程可能需要一些时间，通过审批之后才能在真实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。我们假设你理解面向对象编程的基础知识（例如，知道对象、循环、变量这些基本概念）。你还应该熟悉Objective-C编程语言。本书大部分章节都需要用到Cocoa Touch（它是SDK的一部分），Cocoa Touch使用的是最新版的Objective-C，这个版本有一些新特性。不过不了解Objective-C的新特性也没有关系。本书在用到Objective-C的新特性时会特别指出，并解释它们的工作原理和使用原因。

作为用户，你还应该熟悉iOS系统本身。就像在其他平台中编写应用程序一样，你需要熟悉iPhone、iPad或iPod touch的一些细微差别和怪异模式。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编写程序，下面这些资源可以作为入门读物。

- 《Objective-C基础教程（第2版）》一书浅显易懂，是非常优秀的Objective-C入门读物，作者是Mac编程专家Scott Knaster、Waqar Malik和Mark Dalrymple。
- 苹果文档中介绍该语言的部分“Learning Objective-C: A Primer”。
- *The Objective-C Programming Language*，对Objective-C的介绍非常详尽全面，是非常好的参考指南。

后两个资源可以在<http://developer.apple.com/library/ios/navigation/>上找到。最后一个还可以使用iPhone、iPod touch或者iPad从iBooks中免费下载。下载后就可以随时随地阅读了！苹果已经用这种方式发布了很多开发者学习资料，希望还会有更多惊喜！在iBooks中搜索“Apple developer publications”就可以找到这些内容。

## 1.3 编写 iOS 应用程序有何不同

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（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。起初你可能会觉得有点不得要领，不过不必担心，只要多加练习，很快就能对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。接下来几小节讨论这些差异。

### 1.3.1 只能激活一个应用

在iOS中，任意时刻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，但这也只限于少数情况，而且必须专门为此编写代码。

当应用程序不处于活动状态也不是在后台运行时，它不会占用任何CPU资源，因此也会断开网络连接。iOS允许在后台进行处理，但要做到这一点，开发者需要多做些努力。



### 1.3.2 只有一个窗口

在桌面计算机及便携式计算机操作系统中，多个程序可以同时运行，每个程序可以创建并控制多个窗口。而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 有限的访问权限

计算机上的程序可以访问属主用户（启动这个程序的用户）能够访问的任何内容。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，这块区域称为沙盒，每个应用只能对自己沙盒内的文件进行读写。沙盒就是应用用于存储文档、首选项等任何必要数据的地方。

应用程序还会受到其他方面的限制。比如，不能访问iOS中端口号较小的网络端口，也不能做那些在桌面计算机中需要有根用户权限或管理员权限才能进行的操作。

### 1.3.4 有限的响应时间

由于使用方式特殊，iOS需要能够快速响应各种事件，你的应用程序也应如此。启动应用程序时，需要先打开它，载入首选项和数据，并尽快把主视图显示到屏幕上，这些过程要在几秒钟内全部完成。

在应用程序运行中的任何时刻，都可以通过双击Home键查看最近使用的应用列表。如果用户按Home键，iOS就会返回主屏幕，应用必须快速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，无论是否已经保存完成，应用程序进程都会被终止。

请注意，从iOS 5开始，这种情况因为一种新API的出现而有所改善。它使应用程序在终止前可以申请多一些的时间，完成必要的工作。

### 1.3.5 有限的屏幕大小

iPhone的屏幕显示效果非常出色，在相当长的一段时间里，iPhone一直都是分辨率最高的消费电子设备。

但是，iPhone的显示空间并不大，与现代计算机相比，在iPhone上能使用的屏幕空间要小很多。最初几代的iPhone屏幕分辨率只有320×480，后来，从iPhone 4的视网膜屏幕开始，分辨率增加到了640×960。iPhone 5的分辨率进一步提高到了640×1136。像素数量有了极大的提高，但是屏幕的显示空间却没有大的变化，因此还是不能在屏幕上摆放更多的控件之类。小屏幕极大地影响了iPhone能提供的应用种类和交互性。

iPad的分辨率是1024×768，显示空间也增加了，但也不是非常大。下面来做一个有趣的对比，写本书时，苹果最便宜的iMac的分辨率是1920×1080，最便宜的笔记本电脑（11寸的MacBook