

<<iOS 5基础教程>>

图书基本信息

书名：<<iOS 5基础教程>>

13位ISBN编号：9787115290991

10位ISBN编号：7115290997

出版时间：2012-9

出版时间：人民邮电出版社

作者：[美] Dave Mark,[美] Jack Nutting,[美] Jeff LaMarche

页数：583

字数：880000

译者：毛姝雯,漆 振,杨 越,孙文磊

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

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，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

<<iOS 5基础教程>>

内容概要

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。

《iOS 5基础教程》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。

《iOS 5基础教程》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。

作者简介

Dave Mark 经验丰富的Mac技术开发专家，作家。

他撰写过许多Mac平台和iOS开发方面的畅销书，比如《iPad开发基础教程》（中文版已由人民邮电出版社出版）。

iOS/Android开发公司MartianCraft创始人。

Jack Nutting 资深苹果移动开发技术专家，作家。

Objective-C和Cocoa框架的坚定拥护者。

《iPad开发基础教程》合著者。

个人博客www.nuthole.com。

Jeff LaMarche 资深Mac与iPhone开发人员，拥有20多年开发经验。

他是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，其博客地址为www.iphonedevdevelopment.blogspot.com。

<<iOS 5基础教程>>

书籍目录

目 录

第1章 欢迎来到iOS世界	1
1.1 关于本书	1
1.2 必要条件	1
1.2.1 开发者的选择	3
1.2.2 必备知识	4
1.3 编写iOS应用程序有何不同	5
1.3.1 只有一个应用程序正在运行	5
1.3.2 只有一个窗口	5
1.3.3 访问受限	5
1.3.4 有限的响应时间	6
1.3.5 有限的屏幕大小	6
1.3.6 有限的系统资源	6
1.3.7 不支持垃圾收集	7
1.3.8 新功能	7
1.3.9 与众不同的方法	7
1.4 本书内容	7
1.5 本次更新的内容	9
1.6 准备开始吧	9
第2章 创建基本项目	10
2.1 在Xcode中设置项目	10
2.1.1 Xcode工作区窗口	14
2.1.2 深入研究项目	21
2.2 Interface Builder简介	23
2.2.1 nib文件的构成	24
2.2.2 库	25
2.2.3 在视图添加标签	26
2.2.4 属性修改	29
2.3 美化iPhone应用	30
2.4 小结	34
第3章 处理基本交互	35
3.1 MVC范型	35
3.2 创建项目	36
3.3 查看视图控制器	37
3.3.1 理解输出口和操作	38
3.3.2 清理视图控制器	41
3.3.3 设计用户界面	41
3.3.4 测试项目	51
3.4 理解应用程序委托	51
3.5 小结	55
第4章 更丰富的用户界面	56
4.1 满是控件的屏幕	56
4.2 活动、静态和被动控件	58
4.3 创建应用程序	59

<<iOS 5基础教程>>

- 4.4 实现图像视图和文本字段 59
 - 4.4.1 添加图像视图 59
 - 4.4.2 调整图像视图 61
 - 4.4.3 设置视图属性 62
 - 4.4.4 添加文本字段 65
 - 4.4.5 创建和连接输出口 70
- 4.5 关闭键盘 71
 - 4.5.1 完成输入后关闭键盘 72
 - 4.5.2 通过触摸背景关闭键盘 73
 - 4.5.3 添加滑块和标签 75
 - 4.5.4 连接操作和输出口 77
 - 4.5.5 实现操作方法 77
- 4.6 实现开关、按钮和分段控件 78
 - 4.6.1 添加两个带标签的开关 79
 - 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
 - 4.6.5 为按钮创建并关联输出口和操作 82
 - 4.6.6 实现分段控件的操作方法 82
- 4.7 实现操作表和警报 82
 - 4.7.1 遵从操作表委托方法 83
 - 4.7.2 显示操作表 83
- 4.8 美化按钮 86
 - 4.8.1 viewDidLoad方法 87
 - 4.8.2 控件状态 87
 - 4.8.3 可拉伸图像 88
- 4.9 小结 88
- 第5章 自动旋转和自动调整大小 89
 - 5.1 自动旋转机制 89
 - 5.1.1 点、像素和Retina显示屏 90
 - 5.1.2 自动转屏方法 91
 - 5.2 使用自动调整属性处理旋转 91
 - 5.2.1 配置应用支持的方向 91
 - 5.2.2 指定旋转支持 92
 - 5.2.3 使用自动调整属性设计界面 93
 - 5.2.4 大小检查器的自动调整属性 94
 - 5.2.5 设置按钮的自动调整属性 96
 - 5.3 在旋转时重构视图 97
 - 5.3.1 创建和连接输出口 98
 - 5.3.2 在旋转时移动按钮 99
 - 5.4 切换视图 100
 - 5.4.1 设计两个视图 101
 - 5.4.2 实现交换 102
 - 5.4.3 修改输出口集合 104
 - 5.5 小结 104
- 第6章 多视图应用程序 106
 - 6.1 多视图应用程序的常见类型 106

<<iOS 5基础教程>>

- 6.2 多视图应用程序的体系结构 109
 - 6.2.1 根控制器 110
 - 6.2.2 内容视图剖析 111
- 6.3 构建View Switcher 111
 - 6.3.1 创建视图控制器和nib文件 112
 - 6.3.2 修改应用程序委托 114
 - 6.3.3 修改BIDSwitchView Controller.h 116
 - 6.3.4 添加视图控制器 116
 - 6.3.5 构建包含工具栏的视图 117
 - 6.3.6 编写根视图控制器 119
 - 6.3.7 实现内容视图 123
 - 6.3.8 制作转换动画 126
- 6.4 小结 128
- 第7章 标签栏与选取器 129
 - 7.1 Pickers应用程序 130
 - 7.2 委托和数据源 132
 - 7.3 建立标签栏框架 132
 - 7.3.1 创建文件 133
 - 7.3.2 添加根视图控制器 134
 - 7.3.3 创建TabBarController.xib 135
 - 7.3.4 连接输出口, 然后运行 140
 - 7.4 实现日期选取器 141
 - 7.5 实现单组件选取器 144
 - 7.5.1 声明输出口和操作 144
 - 7.5.2 构建视图 145
 - 7.5.3 将控制器实现为数据源和委托 146
 - 7.6 实现多组件选取器 149
 - 7.6.1 声明输出口和操作 150
 - 7.6.2 构建视图 150
 - 7.6.3 实现控制器 150
 - 7.7 实现依赖组件 153
 - 7.8 使用自定义选取器创建简单游戏 160
 - 7.8.1 编写控制器头文件 160
 - 7.8.2 构建视图 160
 - 7.8.3 添加图像资源 161
 - 7.8.4 实现控制器 161
 - 7.8.5 最后的细节 166
 - 7.8.6 链接Audio Toolbox框架 170
 - 7.9 小结 171
- 第8章 表视图简介 172
 - 8.1 表视图基础 172
 - 8.1.1 表视图和表视图单元 173
 - 8.1.2 分组表和无格式表 174
 - 8.2 实现一个简单的表 175
 - 8.2.1 设计视图 175
 - 8.2.2 编写控制器 176
 - 8.2.3 添加一个图像 179

<<iOS 5基础教程>>

- 8.2.4 表视图单元样式 181
- 8.2.5 设置缩进级别 182
- 8.2.6 处理行的选择 183
- 8.2.7 更改字体大小和行高 185
- 8.3 定制表视图单元 186
 - 8.3.1 向表视图单元添加子视图 186
 - 8.3.2 创建UITableViewCell子类 187
 - 8.3.3 从nib文件加载UITableViewCell 192
- 8.4 分组分区和索引分区 197
 - 8.4.1 构建视图 197
 - 8.4.2 导入数据 197
 - 8.4.3 实现控制器 198
 - 8.4.4 添加索引 201
- 8.5 实现搜索栏 202
 - 8.5.1 重新考虑设计 203
 - 8.5.2 深层可变副本 203
 - 8.5.3 更新控制器头文件 205
 - 8.5.4 修改视图 206
 - 8.5.5 修改控制器实现 210
- 8.6 小结 221
- 第9章 导航控制器和表视图 222
 - 9.1 导航控制器 222
 - 9.1.1 栈的性质 222
 - 9.1.2 控制器栈 223
 - 9.2 由6个部分组成的分层应用程序：Nav 224
 - 9.2.1 子控制器 225
 - 9.2.2 Nav应用程序的骨架 228
 - 9.2.3 向项目中添加图形 234
 - 9.2.4 第一个子控制器：展示按钮视图 235
 - 9.2.5 第二个子控制器：校验表 242
 - 9.2.6 第三个子控制器：表行上的控件 246
 - 9.2.7 第四个子控制器：可移动的行 252
 - 9.2.8 第五个子控制器：可删除的行 257
 - 9.2.9 第六个子控制器：可编辑的详细窗格 262
 - 9.2.10 其他内容 280
 - 9.3 小结 282
- 第10章 storyboard 284
 - 10.1 创建一个简单的storyboard 285
 - 10.2 动态原型单元 287
 - 10.2.1 使用storyboard的动态表内容 288
 - 10.2.2 编辑原型单元 289
 - 10.2.3 实现表视图数据源 290
 - 10.2.4 它会加载吗 292
 - 10.3 静态单元 293
 - 10.3.1 实现静态单元 293
 - 10.3.2 实现表视图数据源 294
 - 10.4 大话segue 296

<<iOS 5基础教程>>

- 10.4.1 创建segue导航 296
- 10.4.2 设计storyboard 297
- 10.4.3 第一个segue 299
- 10.4.4 更为实用的任务列表 299
- 10.4.5 查看任务详情 300
- 10.4.6 设置更多segue 301
- 10.4.7 从列表中传递任务 301
- 10.4.8 处理任务细节 303
- 10.4.9 回传详细信息 304
- 10.4.10 让列表获取详细信息 305
- 10.4.11 小结 306
- 第11章 iPad开发注意事项 307
 - 11.1 分割视图和浮动窗口 307
 - 11.1.1 创建SplitView项目 309
 - 11.1.2 在storyboard中定义结构 310
 - 11.1.3 代码定义功能 311
 - 11.2 显示总统信息 318
 - 11.3 创建浮动窗口 324
 - 11.4 小结 329
- 第12章 应用程序设置和用户默认设置 330
 - 12.1 设置束 330
 - 12.2 AppSettings应用程序 331
 - 12.2.1 创建项目 333
 - 12.2.2 使用设置束 334
 - 12.2.3 读取应用程序中的设置 346
 - 12.2.4 注册默认值 350
 - 12.2.5 更改应用程序中的默认设置 351
 - 12.2.6 实现逼真效果 354
 - 12.3 小结 357
- 第13章 保存数据 358
 - 13.1 应用程序的沙盒 358
 - 13.1.1 获取Documents目录 360
 - 13.1.2 获取tmp目录 360
 - 13.2 文件保存策略 361
 - 13.2.1 单个文件持久性 361
 - 13.2.2 多个文件持久性 361
 - 13.3 属性列表 361
 - 13.3.1 属性列表序列化 362
 - 13.3.2 持久性应用程序的第一个版本 363
 - 13.4 对模型对象进行归档 368
 - 13.4.1 符合NSCoding 368
 - 13.4.2 实现NSCopying 369
 - 13.4.3 对数据对象进行归档和取消归档 370
 - 13.4.4 归档应用程序 371
 - 13.5 使用iOS的嵌入式SQLite3 374
 - 13.5.1 创建或打开数据库 375
 - 13.5.2 绑定变量 376

<<iOS 5基础教程>>

- 13.5.3 SQLite3应用程序 377
- 13.6 使用Core Data 383
 - 13.6.1 实体和托管对象 385
 - 13.6.2 Core Data应用程序 388
- 13.7 小结 398
- 第14章 iCloud之旅 399
 - 14.1 使用UIDocument管理文档存储 399
 - 14.1.1 构建TinyPix 400
 - 14.1.2 创建BIDTinyPixDocument 401
 - 14.1.3 主代码 404
 - 14.1.4 初始化storyboard 410
 - 14.1.5 创建BIDTinyPixView 412
 - 14.1.6 storyboard设计 416
 - 14.2 添加iCloud支持 419
 - 14.2.1 创建provisioning profile 420
 - 14.2.2 启用iCloud授权 420
 - 14.2.3 如何查询 421
 - 14.2.4 保存在哪里 423
 - 14.2.5 在iCloud上存储首选项 423
 - 14.3 小结 424
- 第15章 Grand Central Dispatch、后台处理及其应用 426
 - 15.1 Grand Central Dispatch 426
 - 15.2 SlowWorker简介 427
 - 15.3 线程基础知识 430
 - 15.4 工作单元 430
 - 15.5 GCD : 低级队列 431
 - 15.5.1 傻瓜式操作 431
 - 15.5.2 改进SlowWorker 432
 - 15.6 后台处理 438
 - 15.6.1 应用程序生命周期 439
 - 15.6.2 状态更改通知 439
 - 15.6.3 创建State Lab 441
 - 15.6.4 执行状态 442
 - 15.6.5 利用执行状态更改 444
 - 15.6.6 处理不活动状态 444
 - 15.6.7 处理后台状态 449
 - 15.7 小结 457
- 第16章 使用Quartz和OpenGL绘图 458
 - 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
 - 16.2.2 坐标系 460
 - 16.2.3 指定颜色 461
 - 16.2.4 在上下文中绘制图像 463
 - 16.2.5 绘制形状: 多边形、直线和曲线 463
 - 16.2.6 Quartz 2D工具采样器: 模式、梯度、虚线模式 464
 - 16.3 QuartzFun应用程序 465

<<iOS 5基础教程>>

- 16.3.1 构建QuartzFun应用程序 465
- 16.3.2 添加Quartz Drawing代码 474
- 16.3.3 优化QuartzFun应用程序 478
- 16.4 GLFun应用程序 481
 - 16.4.1 构建GLFun应用程序 482
 - 16.4.2 创建BIDGLFunView 482
 - 16.4.3 更新BIDViewController 489
 - 16.4.4 更新nib 490
 - 16.4.5 完成GLFun 490
- 16.5 小结 490
- 第17章 轻击、触摸和手势 491
 - 17.1 多点触控术语 491
 - 17.2 响应者链 492
 - 17.2.1 响应事件 492
 - 17.2.2 转发事件：保持响应者链的活动状态 493
 - 17.3 多点触控体系结构 494
 - 17.4 4个手势通知方法 494
 - 17.5 检测触摸 495
 - 17.6 检测轻扫 498
 - 17.6.1 使用自动手势识别 502
 - 17.6.2 实现多个轻扫动作 503
 - 17.7 检测多次轻击 505
 - 17.8 检测捏合操作 509
 - 17.9 创建和使用自定义手势 512
 - 17.9.1 CheckPlease应用程序 512
 - 17.9.2 CheckPlease触摸方法 514
 - 17.10 小结 516
- 第18章 Core Location定位功能 517
 - 18.1 位置管理器 517
 - 18.1.1 设置所需的精度 518
 - 18.1.2 设置距离筛选器 518
 - 18.1.3 启动位置管理器 519
 - 18.1.4 更明智地使用位置管理器 519
 - 18.2 位置管理器委托 519
 - 18.2.1 获取位置更新 519
 - 18.2.2 使用CLLocation获取纬度和经度 519
 - 18.2.3 错误通知 521
 - 18.3 尝试使用Core Location 522
 - 18.3.1 更新位置管理器 525
 - 18.3.2 确定移动距离 526
 - 18.4 小结 527
- 第19章 陀螺仪和加速计 528
 - 19.1 加速计物理学 528
 - 19.2 不要忘记旋转 529
 - 19.3 Core Motion和动作管理器 529
 - 19.3.1 基于事件的动作 530
 - 19.3.2 主动动作访问 535

<<iOS 5基础教程>>

- 19.3.3 加速计结果 537
- 19.4 检测摇动 537
 - 19.4.1 Baked-In摇动 538
 - 19.4.2 摇动与击碎 539
- 19.5 将加速计用做方向控制器 544
 - 19.5.1 滚弹珠程序 545
 - 19.5.2 编写Ball View 547
 - 19.5.3 计算小球运动 549
- 19.6 小结 552
- 第20章 iPhone相机和照片库 553
 - 20.1 使用图像选取器和UIImagePickerController 553
 - 20.2 实现图像选取器控制器委托 555
 - 20.3 实际测试相机和库 556
 - 20.3.1 设计界面 557
 - 20.3.2 实现相机视图控制器 558
 - 20.4 小结 562
- 第21章 应用程序本地化 563
 - 21.1 本地化体系结构 563
 - 21.2 字符串文件 564
 - 21.2.1 字符串文件里面是什么 565
 - 21.2.2 本地化的字符串宏 565
 - 21.3 现实中的iOS：本地化应用程序 566
 - 21.3.1 创建LocalizeMe 567
 - 21.3.2 测试LocalizeMe 569
 - 21.3.3 本地化nib文件 570
 - 21.3.4 本地化图像 573
 - 21.3.5 生成和本地化字符串文件 575
 - 21.3.6 本地化应用程序显示名称 577
 - 21.4 小结 578
- 第22章 未来之路 579
 - 22.1 苹果公司的文档 579
 - 22.2 邮件列表 579
 - 22.3 论坛 580
 - 22.4 网站 580
 - 22.5 博客 581
 - 22.6 会议 582
 - 22.7 作者 582
 - 22.8 再会 583

章节摘录

版权页：插图：保存更改，先试运行一下这个版本的Persistence。

更改的东西不多。

首先我们指定了一个新的文件名，以避免应用加载旧的属性列表作为归档。

我们还定义了一个新的常量，作为编码和解码的键 / 值。

然后我们重新编写加载和保存代码，使用BIDFourLines保存数据，并且使用NSCodin9的方法完成实际的加载和保存工作。

GUI与上一个版本完全相同。

新版本需要比属性列表序列化多实现几行代码，因此你可能想知道使用归档是否比使用序列化属性列表更有优势。

对于该应用程序，答案非常简单：实际上并非如此。

但是回想第9章中的最后一个示例，在该示例中，我们允许用户编辑总统列表，每个总统共有4个不同的可以编辑的字段。

要使用属性列表处理对总统列表的归档，需要涉及迭代总统列表、为每个总统创建一个NSDictionary实例、将每个字段中的值复制到NSDictionary实例、将该实例添加到另一个数组，然后将该数组写入属性列表文件。

当然，这是假设我们将自己局限于只使用可序列化的属性。

否则，如果不做大量转换工作就根本不能使用属性列表序列化。

另一方面，如果我们拥有一个可归档对象的数组（如我们刚才构建的BIDFourLines类），则可以通过对数组实例本身进行归档来归档整个数组。

归档集合类（如NSArray）时，也归档其包含的所有对象。

只要放入数组或字典中的每个对象都符合NSCodin9，你就可以归档数组或字典并还原它。

另外，当你对其进行归档时，其中的所有对象都将位于已还原数组或字典中。

换句话说，该方法可以适当伸缩（至少对于代码多少而言），因为无论你添加多少对象，将这些对象写入磁盘的方式（假设你使用单个文件持久性）都完全相同。

但使用扁平列表，工作量会随着添加的每个对象而增加。

13.5 使用iOS的嵌入式SQLite3 我们将要讨论的第三个持久性选项是iOS的嵌入式SQL数据库，名为SQLite3。

SQLite3在存储和检索大量数据方面非常有效。

它能够对数据进行复杂的聚合，与使用对象执行这些操作相比，获得结果的速度更快。

例如，如果应用程序需要计算应用程序中所有对象中特殊字段的总和，或者如果需要只符合特定条件的对象的总和，SQLite3将可以执行该操作，而不需要将每个对象加载到内存中。

从SQLite3获取聚合比将所有对象加载到内存中，然后计算它们值的总和要快几个数量级。

作为一个羽翼丰满的嵌入式数据库，SQLite3包含使其速度更快（例如，通过创建可以加快查询速度的表索引）的工具。

媒体关注与评论

“我是iOS开发平台的铁杆粉丝，读过相当多iOS开发方面的图书，强烈为初学者推荐本书，绝对不会让你失望！”

——StackOverflow评论

编辑推荐

《iOS 5基础教程》英文原版自第1版问世以来在亚马逊计算机图书排行榜持续雄踞榜首，总排名一度在20名左右，被奉为经典。

而《iOS 5基础教程》中文版出版后也迅速荣登各大网店排行榜榜首，并创造了不到半年就印刷四次的销售奇迹。

美国Dave Mark、Jack Nutting、Jeff LaMarche编著的《iOS 5基础教程》从到苹果公司网站注册账号、下载和安装相关工具开始,清晰透彻地讲述了创建iOS应用程序的全过程。

作者在探讨基本概念和各个关键特性时，提供了丰富的实例。

更难得的是，《iOS 5基础教程》始终强调iOS开发中的各种最佳实践,即使是有经验的开发人员,也会受益匪浅。

《iOS 5基础教程》新版针对iOS5和最新发布的SDK做了大幅修订,力求使新老读者都觉得物有所值。

作者使用Xcode 4.2重写了所有项目代码,使之兼容新IBSDK，还专设两章介绍iOS5新特性storyboard和iCloud,这些新特性大大增强了iOS对开发人员的吸引力。

名人推荐

“教程类的书，必须要抓住重点，最好还能给出有针对性的示例和练习题。我看过很多关于iOS开发的书，只有这本书符合我对教程的期望！”

”——David Thompson，iOS专家，Reticulated Ventures公司CEO

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>