

# iOS

## 开发之美

——和凌志 著——

| Storyboard

Delegate

TableView

GO



源代码+视频教程



电子工业出版社  
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY  
<http://www.phei.com.cn>

# iOS

## 开发之美

——和凌志 著——

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

## 内 容 简 介

基于 iOS 平台开发 App 是一个复杂的工程,如何从众多的知识点中理顺思路,把握 iOS 开发的核心是 iOS 初学者面对的难题。

本书是一本初学者都看得懂的图书,作者从 iOS 开发的三个主线——Storyboard、Delegate 和 TableView 入手,以自己开发过的 App 为基础,循序渐进地讲述了在 iOS 开发过程中面对的技术问题。

本书配有光盘,包含了程序的源代码和相关内容的视频教程,可以帮助读者更快地掌握 iOS 的开发。

本书适合从事 iOS 开发以及对 iOS 感兴趣的读者阅读,也可作为相关专业和培训机构的教材。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。  
版权所有,侵权必究。

### 图书在版编目(CIP)数据

iOS 开发之美/和凌志著. —北京:电子工业出版社,2014.11  
ISBN 978-7-121-24513-8

I. ①i… II. ①和… 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(2014)第 236436 号

责任编辑:田宏峰

印 刷:北京天来印务有限公司

装 订:北京天来印务有限公司

出版发行:电子工业出版社

北京市海淀区万寿路 173 信箱 邮编:100036

开 本:720 × 1000 1/16 印张:14.25 字数:210 千字

版 次:2014 年 11 月第 1 版

印 次:2014 年 11 月第 1 次印刷

定 价:58.00 元(含 DVD 光盘 1 张)

凡所购买电子工业出版社图书有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系,联系及邮购电话:(010)88254888。

质量投诉请发邮件至 zlt@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线:(010)88258888。

# 前 言

之所以写这本书，还得从我的个人经历说起。

在过去的十几年中，我一直在手机行业摸爬滚打。对手机软件平台的认识，源自我在 Siemens Mobile 担任软件架构师期间的工作和学习。也正是基于在手机软件行业的积累，才出版了《手机软件平台架构解析》一书。

在我看来，手机软件平台的水分岭是在 2008 年前后。在此之前，各个主流手机厂商都是在打造自己的手机平台，也曾出现过一些小公司所打造的第三方平台。打造一个可用的手机软件平台，绝非一件容易的事，尤其是在智能机时代。仅仅是解决一个触摸屏问题，就让很多手机厂商伤透了脑筋。如何打造一个手机软件平台，让第三方开发者能够在此之上轻松开发出一款 App，这是所有手机软件架构师梦寐以求的向往。直到 iPhone 手机的出现，手机行业才为之一振。我们所期待的智能手机，就应该是这个样子。随着 2008 年 10 月 iOS SDK 的开放，开发者才意识到，原来 App 开发还可以这么玩！

## 为什么将本书命名为“iOS 开发之美”？

时至今日，业内还有多个手机软件平台。我心中很清楚，仅仅以 iOS 为美，肯定会招来各种非议。就我个人而言，我和我的团队在 iOS 平台上做开发，已经

有六个年头了。这里，我只是将自己的工作心得写出来，供大家学习参考。

称“iOS 开发之美”绝非哗众取宠，并不意味着其他平台就是不美。因为我并没有刻意强调 iOS 为最美。只是在 iOS 平台上做开发，确实能体验到一种编程之美。

iOS 架构之美。但凡有编程思想者，在设计软件架构时，都想应用那些设计模式。在这种以 UI 展示为主的 App 开发中，更希望应用 MVC 模式。在我们的开发过程中，深切感受到 MVC 模式在 iOS 平台上发挥得淋漓尽致。没有哪一个手机软件平台能够比 iOS 的图形化编辑器更能接近真实的产品。开发一款产品原型，仅仅需要几个简单的拖拽，便跃然成型。这不能不说是 iOS 平台所独有的 Storyboard 技术成全了 App 开发之美。

iOS 代码之美。在编写 iOS 代码中，我们能切实感受到，完成一项功能，只需编写短短的代码量。这奉行了“简单即是美”的原则。在斯坦福大学的 iOS 公开课上，我们看到的是那种行云流水般的代码编写，如同敲击钢琴键盘一样流畅，这不得不说是一种美的享受。

iOS 适配之美。对 App 开发者来说，最担心的是那些琳琅满目的机型适配。开发一款 App 已经让我们筋疲力尽，如果再适配那么多的机型，让我们情何以堪。好在 iOS 平台已经为适配考虑得很是周全，一个 Autolayout 技术解决了开发者的后顾之忧。在 Autolayout 上做适配，如同在 Word 上排版一样轻松自如，无须编写大量的代码，永远是拖拽，拖拽！与其说是一种简单的操作，毋宁说是一种轻松愉悦的享受！

## 写一本让 iOS 初学者看得懂的书

如同编写剧本一样，场景的顺序是多么的重要。在 iOS 浩瀚的知识海洋中，如何寻到一条主线，将 iOS 关键知识点贯穿起来，编写一本让 iOS 初学者看得懂的书，这是我过去几年一直努力完成的工作。

在我的网易博客上，曾有一篇博文引起了读者的强烈反响，那就是“写给 iOS 初学者”，而这篇博文的灵感，来源于自己创建的 QQ 群“iOS 之美”。初次接触 iOS 开发的群友经常问起如何入手，推荐哪一本教程。iOS 入门之所以让人感觉到一头雾水，是因为在过去的几年中，iOS 开发平台更新得太快了，可用“频繁”两个字来形容。苹果公司是追求完美主义者的代名词，在过去的几年，iOS 开发平台一直在改进，不断地在自我革命，日趋完美。

这本书没有刻意追求面面俱到，而是沿着一条主线，讲述了四个故事：Objective-C 编程语言、Storyboard 框架、Delegate 应用和 TableView 数据展示。

(1) Objective-C 语言。开发 iOS App，离不开 Objective-C 编程语言。单要完整讲述 Objective-C，就得需要大量的篇章。这里，我们着重讲述了几个关键对象的应用。例如，property、NSString、NSArray、NSDictionary 等，这些都是与数据处理密切相关的概念，所有的 App，都是建立在数据处理基础之上的。掌握了数据管理机制，就为数据在 UI 层面的展示打下了基础。

(2) Storyboard 技术的应用。iOS 开发平台之所以强大，正是源于它拥有一个神器——Storyboard 框架。试想一下，所有的 App，从用户体验的角度看，不就是在多个页面之间跳来跳去么？为实现页面之间的跳转而编写大量的代码，有人愿意这么做吗？大多平台，不得已而为之。而苹果公司发布的 XCode 4.2 版本，通过引入 Storyboard 技术，成功地解决了这个问题。花大量时间，来实现页面之间的跳转逻辑，对程序员来说，这是一种简单而重复的劳动。Storyboard 技术的推出，使得 iOS 开发者可以快速开发出一款 App。在 Storyboard 篇，我们从多个维度讲述了 Storyboard 的引擎——Segue 的应用。

(3) Delegate 的应用。Delegate 是委托、代理的意思。它是一种设计模式。因为 Delegate 是一个抽象的概念，不容易理解。在我的博客上，曾有一篇“Delegate 应用五步曲”，深受用户的点赞。为此，我重新设计了一个更加贴切的实例，并作为一道上机编程的面试题，呈现给读者。在这一章节，我们详细讲述了对 Delegate 的疑惑，Delegate 的应用场景，其目的是为了弄清楚一个概念：难

道非用 Delegate 不可吗？我们的回答是，在某些特定的场景下，选用 Delegate 就是最优的方案。

(4) TableView 的应用。TableView 又称为表视图。表视图是 iOS App 中一个常用的 UI 对象。可以说，表视图是处理数据的神器。App 中，那些可上下滑动的列表，就是表视图。iPhone 手机中的通讯录，就是一个典型的表视图。表视图不仅可以显示文本，也可以显示图像，还可以内嵌按钮、输入框等。只要是涉及数据处理的 App，都会用到 TableView。正是因为它的强悍，TableView 的应用也变得复杂起来，熟练掌握 TableView 需要一定的基础。这也是我们为什么把 TableView 放到最后一章的原因。

## 书中的每个实例是怎样讲述的？

对于书中出现的每个知识点，都辅以相关的代码实例。每个篇章中的实例都不是独立的，而是延续从易到难的线索。拿 TableView 中的实例来讲，从创建最基本的工程开始，到逐步丰富这个实例的功能，中间设有六个台阶。每新增一项功能，所需添加的代码都在几行之内，以免读者一下子被大量的代码所晕倒。TableView 系列教程，是我最为满意的一个章节。它涉猎了 iOS 的多个知识点，不管是 NSArray、NSDictionary、Delegate，还是 Storyboard，都在这里得到了充分的应用。

值得一提的是，在 TableView 实例讲解中，我们之所以将电台作为素材，是因为我对网络电台的情结之深。诞生于移动流媒体平台的“悠悠电台”，不管是过去还是现在，一直都是我们引以为豪的 AppStore 杰作。

## 致谢

在本书成稿的过程中，我得到了很多人的指点和帮助，客套话不再讲太多。这里，我非常感谢曾经拜读过多遍的斯坦福大学 iOS 公开课。书中的部分实例，有它的影子。每次赏析公开课的视频，都能得到一次心灵的升华。

参与本书编写的还有林志红、尹陆军、袁芳、张俊、王方、马钧君、和凌群、刘晓波、王娟、张爱华。

## 关于本书配备的光盘

在 Storyboard 上开发 App，离不开 UI 界面的拖拽操作。为了更详尽地讲述操作手法，特录制了部分视频。每个实例对应的工程源码，都可以在光盘中找到对应的文件。

编程是一个脑力活儿，也是一个体力活儿。我坚定地认为：“任何技术都必须通过模仿和实践来掌握。”

## 作者交流方式

作者的 QQ 是 2385911707；作者的微信号是 Leopard2385911707；QQ 群（iOS 之美）是 238303969。

和凌志  
2014 年 9 月



# 写给 iOS 初学者

编者按：“写给 iOS 初学者”是作者为 iOS 初学者量身定制的技术路线，深受网友好评，在此，以飨读者。

谈到 iOS 的学习，一下子将我和我的团队带到六年前的时光。

记得那是 2007 年，我们拿到了传说中的 iPhone，做的第一件事情就是“越狱”。那时的 iPhone，不越狱是打不了电话的。越狱的过程充满了提心吊胆，唯恐一不小心把 iPhone 变成“砖头”。

多少年来，我一直在手机软件行业摸爬滚打。作为一名软件工程师，自然想在 iPhone 上开发一款属于自己的 App。终于等到 2008 年 10 月，Apple 开放了 SDK，自此，我们开始走向了正轨，加足马力，先后发布了几款极具商业价值的 App，并打造了属于自己的移动互联网平台。

历经六年的迅猛发展，iOS 开发如日中天。这一方面得益于国内广泛的 iOS (iPhone/iPad) 用户基础；另一方面，iOS 开发平台本身，也为开发者提供了一个先天的优势。这里绝不是厚此薄彼，早在开发 iOS 之前，我们也曾在 Nokia 塞班平台上开发过软件，所经历的痛苦，可用一句话形容——“谁用谁知道”。至于 Android 平台，我们不想谈及太多。为对比几大平台的优劣，而引发口水战，大可不必。相信你读到这里时，对选择哪个开发平台，你已经有了自己的主见。

很多人在看了我们的 App 之后，经常问：你们最初是怎么学 iOS 的？

谈起我们初学 iOS，忍不住有很多话要吐槽。这个过程依然是很艰难的。iOS

开发的门槛较高，一个硬件设备，比如苹果电脑，足以把很多跃跃欲试者挡在了门外。我们也尝试过虚拟机的安装，也就是业内所说的“黑苹果”。对于“黑苹果”，我的感悟是，玩玩还行，若真的要下决心做事，还得 MacBook。最初接触 iOS 开发时，我用过“黑苹果”。所谓“黑苹果”，就是在 Windows 操作系统上装一个 Mac OS 虚拟机。当时购买苹果电脑(MacBook)，是一笔不小的开销。后来决定从事 iOS 开发时，最终还是痛下决心，购置了 MacBook。学习 iOS，常被开发环境所困扰。为装一台“黑苹果”，折腾一周算是正常，这还得运气好。在开发环境的搭建上，浪费太多的时间，得不偿失。对任何人来讲，时间才是最宝贵的财富，每个人都应该把时间投入到最有意义、最有影响的地方去。没有投入，就没有产出。如果你狠心买了 MacBook，想必会坚持学好 iOS，正所谓“舍不得孩子套不着狼”。

在 iOS 开发的旅途中，曾记录了我们多次瞬间的喜悦与感动：

- 当成功注册为 iPhone 开发会员 (iPhone Developer Program, IDP) 那一瞬间；
- 当 iPhone SDK 开发环境配置成功，显现“Hello World”的那一瞬间；
- 当一款自己精心设计的应用程序得以实现的那一瞬间；
- 当应用程序成功签名并在 iPhone 手机上随手指的触动而飘逸的那一瞬间；
- 当应用程序成功发布到 AppStore 上，收到来自 Apple 的 E-mail，赫然标识“Ready for sale”的那一瞬间。

作为 iOS 初学者的你，相信在未来的开发之旅中，也会经历一些困惑与喜悦。这里，先送上一句鼓励的话：“含泪播种，必将微笑收割。”

## 我经常被问到的第二个问题是：iOS 开发从何入手？

iOS 开发没有什么神秘的地方，就开发一个 App 来说，这个平台已经足够强大了。iOS 的版本发布虽然很频繁，但从技术框架的角度来看，iOS 里程碑式的发布，主要体现在 iOS5 SDK 的发布，对应的 XCode 版本是 4.2，此版本引入 Storyboard 技术。以我之拙见，Storyboard 技术才是 iOS 开发框架的分水岭。至于其他版本的发布，对 iPhone 手机用户来讲，有多个亮点和创新，但与 iOS 开发者关系不大。

这里重点吹捧下 Storybord 技术。在我看来，Storybord 技术的出鞘，让其他任何手机平台难以望其项背。试想一下，所有的 App，从用户体验的角度看，不就是在多个页面之间跳来跳去么？为实现页面之间的跳转而编写大量的代码，有人愿意这么做吗？大多平台，不得已而为之。而苹果公司发布的 XCode 4.2 版本，通过引入 Storyboard 技术，成功地解决了这个问题。花大量时间，来实现页面之间的跳转逻辑，对程序员来说，这是一种简单而重复的劳动。只有工作量，缺乏技术含量。Storyboard 技术的推出，使得 iOS 开发者能够以最快的速度，以最低的成本开发出一款 App，这本身不就是强有力的竞争力么？

说了这么多，其实就为了表达一句话：学习 iOS 开发，一定要善用 Storyboard 技术。

iOS 平台所涵盖的知识点浩瀚无垠，如何找到一条主线，将这些知识点贯穿起来？不妨，我们还是从 App 本身的特点来寻找答案吧。几乎所有的 App，无非解决两个问题——UI 展示和数据处理。

我们先来看看数据的展示。移动互联网应用的核心在于用户体验上，而绝妙的用户体验正是集中体现在如何流畅地展示数据。iPhone 手机最为经典的数据展示页面，莫过于电话本。几千个电话号码，通过上下滑动而一览无余。为此，iOS SDK 推出了表视图 (TableView) 技术，有效地解决了数据展示的问题。

接下来，再看看数据的交互。我们知道，每个 App 都是由多个页面组成的，用户在使用时，会触摸页面上的对象，页面之间也会前后跳转。其实，页面之间的跳转必将引发数据的交互。如何解决好数据的交互问题呢？iOS SDK 正是借助 Delegate 这个设计模式，有效地解决了这一问题。

最后，又回到了 UI 的实现上来，正是为实现页面的跳转和页面之间的数据交互，Apple 才引入了 Storyboard 技术。Storyboard 技术才是真正意义上的王者归来。Storyboard 让整个 App 开发的世界，变得如此简单而有趣味。

至此，或许你已经清楚了这本教程的主线，那就是 Storyboard + Delegate + Ta-

bleView。我们所讲述的所有实例，都是在剥茧抽丝，从多个维度来阐述这条主线的。

学 iOS 开发，还需要了解很多相关的知识。你需要注册 Apple ID；如果想发布到 AppStore，还需要注册苹果开发者账号，还要熟悉发布过程。诚然，了解和掌握这些知识点，是很重要的。但是，这些点都是对 iOS 开发流程相关的理解，涉及的技术知识点不多。如果将 iOS 周边知识点纳入到这本书中，将占大量的篇幅，就有点儿喧宾夺主了。

说一千道一万，还是那句老话，教程本身不是万能的。每个人的基础不一样，对同一个教程的领悟也不相同。我们所做的，就是帮你在 iOS 学习的旅途上找回信心；让你在遇到问题时，不再感到孤独无助。当你自己精心开发的 App 发布到 AppStore 的那一天，蓦然回首，你会发现，过去付出的一切，都是值得的！

# 目 录

## 第 1 篇 Objective-C 语法篇

第 1 章 iOS 开发环境的搭建 .....	3
1.1 硬件环境 .....	4
1.2 软件环境 .....	4
1.2.1 注册 Apple ID .....	4
1.2.2 XCode 下载与安装 .....	5
1.2.3 iOS 开发语言: Objective-C .....	6
1.3 分享一个小故事 .....	7
1.4 总结与启发 .....	8
第 2 章 Objective-C 编程语言特性 .....	9
2.1 Objective-C 有什么奇特的地方 .....	10
2.2 如何声明实例变量(浅谈@ property 的使用) .....	12
2.3 NSString 的应用 .....	14
2.4 如何判断两个字符串是否相等 .....	14
2.5 对象类型与基本数据类型的混合使用 .....	16
2.6 对象类型与基本数据类型的转换(NSNumber 与 NSInteger) .....	17

2.7	不可变数组 (NSArray) 与可变数组 (NSMutableArray) .....	19
2.7.1	不可变数组 (NSArray) 的特征 .....	19
2.7.2	可变数组 (NSMutableArray) 的特征 .....	20
2.7.3	如何遍历数组中的对象 .....	21
2.8	不可变字典 (NSDictionary) 与可变字典 (NSMutableDictionary) 的应用 .....	23
2.8.1	不可变字典 (NSDictionary) .....	23
2.8.2	可变字典 (NSMutableDictionary) .....	24
2.8.3	如何遍历字典中的对象 .....	25
2.9	一种最简单的永久数据存储方式 (NSUserDefaults 的应用) .....	26
2.10	创建类的对象的两种方法 .....	28
2.11	一种最简单的调试方法 (NSLog 的使用) .....	29
2.12	忍不住也来说说 iOS 的设计模式 (MVC 的使用) .....	29
2.13	总结与启发 .....	31

## 第 2 篇 Storyboard 技术篇

第 3 章	iOS 常用开发控件 .....	35
3.1	XIB App 与 Storyboard App 之争 .....	36
3.2	Storyboard 开发环境 .....	36
3.3	创建一个新的工程 .....	37
3.4	熟悉 XCode 开发环境 .....	39
3.5	运行 HelloWorld .....	40
3.6	按钮 (UIButton) 与标签 (UILabel) .....	41
3.6.1	创建 UIButton 的 IBAction .....	42

3.6.2	创建 UILabel 的 IBOutlet	44
3.6.3	实现 Button 被触发后的方法	45
3.7	如何删除对象的 IBOutlet 或 IBAction	45
3.8	让键盘消失的四种方法	46
3.8.1	键盘消失方法一: Delegate 应用三步法	48
3.8.2	键盘消失方法二: 事件响应的调用	51
3.8.3	键盘消失方法三: 将 UIView 改为 UIControl	53
3.8.4	键盘消失方法四: 巧用 UIButton	55
3.9	UITextField 更多用法	56
3.10	getter 与 setter 的应用场景	57
3.11	总结与启发	59
第 4 章	视图 (UIView) 与视图控制器 (UIViewController)	61
4.1	如何创建一个基本的视图	62
4.2	透过 UICollectionViewController 看 Delegate 的应用	64
4.3	视图与视图控制器的生命周期	67
4.4	滚动视图 (UIScrollView) 的应用	69
4.5	总结与启发	71
第 5 章	Storyboard 知多少	73
5.1	Storyboard 编辑页面	76
5.2	Storyboard 利器: Segue(连线)	77
5.3	创建和触发 Segue 的两种方法	78
5.4	Segue 对象的属性设置	81
5.5	触发 Segue 的两种方法	82
5.6	Segue 触发瞬间都要做些什么	83
5.7	巧用 Segue	84
5.8	Segue 的创建	86
5.9	嵌入 Navigation Controller	87

5.10	活用 prepareForSegue;sender:	90
5.11	重写对象的 setter 和 getter	93
5.12	UIScrollView 应用三步法	98
5.13	如何实现图片的缩放	100
5.14	通过异步加载获取网络图片	102
5.15	如何实现一个等待提示	104
5.16	总结与启发	107

## 第 3 篇 Delegate 应用篇

第 6 章	Storyboard 场景跳转方法	111
6.1	场景之间的正向跳转	112
6.1.1	直接跳转法(控件直接拖拽)	112
6.1.2	条件跳转法(performSegueWithIdentifier)	113
6.1.3	手动代码跳转法	116
6.1.4	三种方法的比较	118
6.2	场景之间的逆向跳转	118
6.2.1	Unwind Segue 连线法	119
6.2.2	关于 Unwind Segue 的更多用法	120
6.2.3	通过编码方式实现逆向跳转	121
6.3	总结与启发	121
第 7 章	Delegate 应用五步曲	123
7.1	学习 Delegate 的困惑	124
7.2	从一道经典的面试题说起	124
7.3	学习 Delegate 常出现的几个误区	125



7.4	Delegate 技术难点在哪里 .....	126
7.5	数据逆向传送一定要通过 Delegate 吗 .....	127
7.6	Delegate 应用五步曲 .....	128
7.7	总结与启发 .....	133

## 第 4 篇 表视图( TableView) 应用篇

第 8 章	创建一个简单的表视图 .....	137
8.1	表视图与表视图单元格( TableView 与 TableViewCell) .....	138
8.2	UITableView 的代理机制( Data Source 和 Delegate) .....	139
8.2.1	UITableViewDataSource 的应用 .....	140
8.2.2	UITableViewDelegate 的应用 .....	142
8.3	创建一个表视图工程 .....	143
8.4	视图控制器添加三步法 .....	144
8.5	为表视图填充数据 .....	147
8.6	如何确定列表的行数 .....	148
8.7	如何显示列表每一行对应的内容 .....	149
8.8	如何在表视图中添加缩略图 .....	152
8.9	总结与启发 .....	155
第 9 章	如何实现表视图的定制化单元格 .....	157
9.1	如何让每一行显示特定的缩略图 .....	158
9.2	如何定制化单元格( UITableViewCell) .....	161
9.3	为定制化单元格创建一个类 .....	166
9.4	总结与启发 .....	169