

爱上 Swift

Getting Started  
with Swift



从零开始到 AppStore 上架  
基础知识与实例分步精讲  
含图片素材和代码包下载链接

专注实战开发的 Swift 语言系列书

3

书内  
附赠

无限互联

500 元代金卡

可抵培训班学费

学 App 开发

# Swift 语言 实战精讲

刘明洋 / 著 汪鸿俊、任小雷 / 审



人民邮电出版社  
POSTS & TELECOM PRESS

---

本书仅提供部分阅读，如需完整版，请联系QQ: 461573687

提供各种书籍pdf下载，如有需要，请联系 QQ: 461573687

PDF制作说明：

本人可以提供各种PDF电子书资料，计算机类，文学，艺术，设计，医学，理学，经济，金融，等等。质量都很清晰，而且每本100%都带书签和目录，方便读者阅读观看，只要您提供给我书的相关信息，一般我都能找到，如果您有需求，请联系我 QQ: 461573687, 或者 QQ: 2404062482。

本人已经帮助了上万人找到了他们需要的PDF，其实网上有很多PDF,大家如果在网上不到的话，可以联系我QQ。因PDF电子书都有版权，请不要随意传播，最近pdf也越来越难做了，希望大家尊重下个人劳动，谢谢！

**备用QQ:2404062482**

爱上 Swift

Getting Started  
with Swift

学 App 开发

# Swift 语言 实战精讲

刘明洋 / 著 汪鸿俊、任小蕾 / 审

人民邮电出版社  
北京

## 图书在版编目(CIP)数据

Swift语言实战精讲 / 刘明洋著. — 北京: 人民邮电出版社, 2015.4  
(爱上Swift)  
ISBN 978-7-115-34703-9

I. ①S… II. ①刘… III. ①程序语言—程序设计  
IV. ①TP312

中国版本图书馆CIP数据核字(2015)第039412号

## 内 容 提 要

《Swift语言实战精讲》通过案例精细讲解用Swift语言开发App的实战技巧。容易理解,并能马上应用与实战。真正实现从零基础到App Store上架,让你成为Swift开发的领航者。本书共分为两大部分。第一部分是对基础知识的快速讲解,语法入门快人一步。第二部分包含14个章节,是实战技巧的重头戏,主要通过一个App的实例来介绍iOS图形用户界面的开发、数据存储、触摸和手势、动画、多媒体、网络通信设备朝向和移动、定位与地图、地址簿、短信、邮件、电话、本地化、推送通知、真机调试与iOS程序发布等高级开发技巧,一本书搞定商业化的移动开发技能。

- 
- ◆ 著 刘明洋  
审 汪鸿俊 任小蕾  
责任编辑 紫 镜  
执行编辑 马 涵  
责任印制 周昇亮
- ◆ 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: 800×1000 1/16  
印张: 35.25 2015年4月第1版  
字数: 820千字 2015年4月北京第1次印刷
- 

定价: 79.00 元

读者服务热线: (010)81055339 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京崇工商广字第0021号

# 推荐序 1

北京时间 2014 年 6 月 3 日凌晨，注定是个被 IT 界记住的日子。苹果公司在 WWDC 上发布了全新的编程语言 Swift，志在让更多的人轻而易举的做出自己的 App。苹果公司早在 2010 年就启动了这项计划，经过 4 年的酝酿，终于在 2014 年夏天发布。因为苹果公司的保密工作做得很棒，在 WWDC 之前几乎没有人知道会有全新的语言发布。

Swift 语言仅发布一月，就已跃至 TIOBE 排行榜 16 位，苹果发布 Swift 的目标就是让更多的人学会 iOS 开发，Swift 语言的发展前景要比 C 语言、Objective-C 语言、Java 等语言更好，势头会更迅猛。我们可以想象一下，几年之后，Swift 开发者越来越多，Objective-C 会逐渐减少，甚至很多人都不熟悉 Objective-C。所以开发者能越早使用 Swift 进行编程，那么优势就会越明显，不管是哪个行业，先进去的只有两种可能，要么败得一塌糊涂，要么赚得盆满钵满，后来者连汤都喝不到。当然这些事情都不会在一夜之间发生，你可以从现在开始做好准备。

目前国内关于用 Swift 语言来开发 App 的资料甚少，市面上也仅仅有几本以 Swift 语法为主的学习资料和个别 Demo 介绍，而在实际工作中，仅仅会语法会一两个 Demo 是远远不够的。而刘明洋写的这本《Swift 语言实战精讲》恰好给大家提供这种便利，可以为大家掌握 Swift 语言起到帮助作用。无论你之前有没有 Objective-C 基础，都可以使用 Swift 语言，轻松快速地开启 iOS 开发之门。

本书除了讲解 Swift 语言基础知识外，还添加了许多 Demo 和商业项目用法。全书共 16 个章节，覆盖面广，讲述了多媒体、动画、网络通信、地理定位、手势、系统服务（邮件、短信、地址簿等）、真机调试、推送通知等内容。另外，Swift 语言可以与之前的 C、Objective-C 语言混编，这样也解决了 Swift 语言目前无可开源框架的问题，可以直接用来开发商业 App。刘明洋的这本《Swift 语言实战精讲》从培训角度全面讲解 Swift 语言，带领大家从实际着手，深入了解 iOS 移动开发的每一个环节。一本书，一次到位，让你可以直接对企业说“我能做”。

——无限互联 CEO 熊彪

---

以机构名称首字母顺序排序，排名不分先后

# 推荐序 2

曾几何时，三个苹果改变世界的趣闻传遍大街小巷，从夏娃摘下苹果的那一刻起，人类进入一个新的篇章。牛顿的故事不论真假，都在人类探索世界的道路上留下了漂亮的一笔。然后是乔布斯的苹果，这个完美主义者，给我们设计了一个全新的感知世界，无数人为之疯狂。

很幸运，我们见证了乔布斯苹果一路的发展和成长，我们一次次地等待苹果产品的发布。等待，让我们学会珍惜，获得，让我们欣喜若狂，为之奋斗，我们勇往直前。我们期待改变，期待新的生活方式，苹果满足了我们的心理需求。每一次，我们都不会失望。在刚刚过去的 WWDC2014 大会上，苹果公司新发布了一种编程语言 Swift，此语言继承了 C 语言以及 Objective-C 的特性，且克服了 C 语言的兼容性问题，对于广大开发者来说，这无疑是一剂难以抗拒的兴奋剂。

市面上不乏关于 Swift 语言基础知识的书籍，但当我看到明洋的这本书稿时，我的第一感觉是，终于有这么一本系统全面地从实战的角度讲解 Swift 语言进行移动 App 开发的书籍了。他向广大 Swift 语言学习者和开发者分享了 iOS 商业开发方面的实战经验，帮助更多开发者快速学到优化产品性能的方法，以提升产品的用户体验。

改变，从这一刻开始，希望这本书籍能够对广大 Swift 开发者有所帮助，也祝大家创作更多优秀的产品为我们的生活带来更多的色彩和惊喜。

——西湖创客汇理事长、杭州汇文董事长 戴立

# 推荐序 3

尽管挑战苹果 iOS 体系的安卓手机厂商不会少于 1000 家，但到目前为止，iOS 的用户体验依然难以超越，特别是在易用性方面。

在 WWDC 2014 发布了新的编程语言 Swift 之后，苹果又把这种易用性送到了开发者手中。比起 Objective-C 来说，Swift 无疑是更加简洁和易用的。

掌握一门新的开发语言，经历一段时间去学习是不可避免的过程。而让这个过程缩短最有效的方法，就是结合实例来学习实战开发的技巧，明洋的这本书就具备了这样的特点。

《Swift 语言实战精讲》内容强调实战性，涵盖了 Swift 语法、UI 开发与控件、读写应用程序数据、多媒体应用、OC 汇编与迁移、真机调试和发布等多个方面的内容。这是一本“端到端”的开发教程，从创建工程开始，一直覆盖到让 App 成功在 App Store 上架，可以说是手把手地与读者一起完成用 Swift 进行 App 开发的过程了。相信这本书能够对广大 iOS 开发者有所帮助，可以做到一书在手，Swift 编程不愁。

说到编程，一般人对它的感觉可能是枯燥。但是如果深入到大脑的工作层面，其实对一个程序员来说，编程的过程和音乐的创作过程比较类似，也是一种身无旁骛的灵感与技术的紧密结合。

希望诸位读者，能够通过掌握一门简单的开发语言，用最简洁、最精确的代码，搭架一个零 bug 的自由王国，让自己的作品成为无数人愿意倾听的艺术。

——至顶网 CEO 高飞

# 前言

自从苹果公司在 2014 年的 WWDC 大会上发布了全新的编程语言 Swift 语言，本人就一直在关注和研究 Swift 语言。对于一门新语言来说，开荒阶段的探索自然是激动人心的，但很多时候资料的缺失和细节的隐藏也让人着实苦恼。虽然 Swift 相对于 Objective-C 可以说除了语法上比较特别之外，更简洁容易。然而就在这漂亮的语法背后其实隐藏了很多细节和功能。我们如果无法理解这些细节的实现，就很难明白这门新语言在设计上的考虑。在实际开发中会遇到各种各样的问题导致编译不过。

对于任何一门语言来说，语法书都是漫天盖地的，但真正适合读者的书实在寥寥。目前市面上已经出现了各种相关书籍，但多数都是基本语法讲解，只能算语法级入门，算不上开发入门。本人通过发表一些博客试图通过自己的分享向大家提供一些帮助，通过博客也发现有很多喜欢 Swift 语言的同学，不管是自学的还是专业的，我都能感受到大家的热情。也是一种机缘，无限互联的汪老师找到我，问我是否可以写本书来分享更多的经验。回想自己多年编程经历，也积累了一些自己的看法和讲解技巧，因此有兴趣一试。本人出于对编程的爱好，平时大多通过自学来补给知识，所以对于什么样的开发入门书能够使读者用起来更得心应手有一点心得。我希望能够写出一本书，让新手、老手学起来简单，看完之后能真正做出点东西来，能自己感觉到真正学到了一些知识，甚至能开发实际的项目，熟悉整个移动开发的所有流程。本着这些想法，本人历时 3 个月写了这本书，运用 Swift 语言从入门到实战，引入 App 实例来演示从创建 Project 到最终发布程序的 Swift 编程方法。希望这本书能够实现我写这本书的初衷，让大家读起来觉得好用。

Swift 语言极有可能代替 Objective-C 语言。而目前国内 iOS 平台下拥有庞大的 Objective-C 开发群体，更有大批想加入苹果开发的人群。相信不论是对于 Objective-C 开发者还是对于新手来说，大家都很关心到底如何开启 Swift 之旅、如何从 Objective-C 转向 Swift 阵营。希望读者通过阅读此书可以真正地了解移动开发的所有流程、掌握使用移动设备特有的一些功能。期盼我的经验能让大家少走弯路，也期待各位能从我的经验中吸取有价值的东西，更快地跃过这扇编程之门，正式步入 Swift 编程的世界。

## 本书特点：

- 容易上手，通过案例精细讲解 Swift 语言的实战技巧。使读者容易理解，并能马上学以致用。对于每一部分具体内容，都精心设计了相应的示例程序，一方面可以帮助读者加深理解，另一方面也可以逐步培养读者的程序设计能力。
- 技术实用，本书针对 Swift 开发中常用的一些功能进行了详细讲解。
- 内容全面，本书不仅讲解了基本控件的使用，而且介绍了很多使用高级功能的方法。
- 版本最新，本书以 Swift 1.1 版本为基础进行讲解，介绍了新版项目配置和最新的技术实现。



## 本书内容：

本书一共分为两个部分。

第一部分分为 2 个章节，快速讲解基础知识，主要讲述 Swift 语言的基本语法、Swift 与 Cocoa。通过简单清晰的例子，让你学到所有语法基础，语法入门快人一步。

- 第 1 章主要介绍了 Swift 语言的基本语法、表达式与控制流程语句以及苹果帮助文档的使用。
- 第 2 章介绍了字符串、集合的常用处理方法以及类的定义，包含了 Swift 语言的特性，诸如闭包、Playground 等。与此同时，还包括了多线程的处理方式。

第二部分分为 14 个章节，重点介绍了 iOS 图形用户界面的开发、动画、多媒体、定位与地图、网络通信、本地化、推送通知、真机调试与 iOS 程序发布等高级开发。

- 第 3 章介绍了苹果开发环境（IDE），详细解释了“自动布局”和“Size Classes”等用法。
- 第 4 章详细讨论了 iOS 常用基本控件的使用以及 MVC 的设计模式。
- 第 5 章讨论了 iOS 开发中数据的持久性的解决方案，如文件操作、对象归档、SQLite 数据的使用、Core Data 和 iCloud 的使用等。
- 第 6 章介绍了 iOS 系统事件响应的处理和手势识别器的使用等。
- 第 7 章介绍了音频、视频的常用处理方法以及访问相册和调用摄像头的处理方式。
- 第 8 章重点讨论了动画处理的解决方案。
- 第 9 章详细介绍了网络通信。如，判断网络是否连接、解析 JSON 数据、常用请求方式介绍以及 NSURLSession 的常用处理方法等。
- 第 10 章介绍了加速计、陀螺仪以及设备横竖屏幕的处理方式。
- 第 11 章介绍了定位与地图的处理。
- 第 12 章重点论述了 iOS 地址簿、邮件、电话和短信的处理方式。
- 第 13 章介绍了本地化的应用。
- 第 14 章探讨了 Swift 与 Objective-C 语言的混编和将 Objective-C 语言迁移至 Swift 语言。
- 第 15 章介绍了 iOS 如何真机调试和应用程序如何发布等。
- 第 16 章介绍了苹果的推送服务通知技术（APNS）。

## 读者对象

本书既可以作为大、中院校 iOS 专业教材用书，也可以作为想进入或者已进入 iOS 开发队伍的人员使用的案头参考书。具体目标读者定位为：希望快速了解 Swift 语言的读者、想要快速使用 Swift 进行实际项目开发的读者、相关培训机构的老师和学员、程序测试及维护人员、编程爱好者、准备做毕业设计的学生、参加实习的初级程序员、大中专院校的老师和学生、初中级程序开发人员。通过学习本书的内容，

能够帮助读者提升实际的工作能力，提升收入水平至 8K 以上。

## 致谢

非常感谢人民邮电出版社对我的信任，感谢人民邮电出版社的编辑对本书的帮助和支持。

十分感谢无限互联 CEO 熊彪、西湖创客汇理事长戴立、至顶网 CEO 高飞对本书的大力支持。

感谢无限互联 iOS 教学总监、国内著名的 iOS 培训专家、学院 iOS 培训课程研发领航者——汪鸿俊老师，以及国内著名高级 iOS 开发工程师、实力派苹果 iOS 开发工程师、资深 iOS 研发工程师——任小蕾老师对本书的审核。

另外也非常感谢以下各位对本书的大力帮助和支持，他们分别是：李志河、江艳、王丹、吴利光、王莎、薛富荣、张欣杰、李晓兰、杨宪杰。

## 联系方式

每一个用心写出的 App 都是一件艺术品，希望通过本书帮助大家使用 Swift 语言去创建属于自己的艺术品，由于时间仓促，本书难免有疏忽、纰漏，不完美之处敬请老师和读者对我们的工作提出建议和指正，谢谢。

请通过下面的方式联系我们：

Swift 领航者 QQ 群：248728032

微信公共账号：Swift 课堂（右侧二维码）

加入 Swift 领航者 QQ 群或微信公共账号，一起探讨、一起探究、一起成为 Swift 的领航者。若书中夹带的卡片出现遗漏、使用问题的也可联系该群管理员。



Swift 课堂  
微信公共账号二维码

您也可以在无限互联网站中获取更多的学习资料——[www.iphonetrain.com](http://www.iphonetrain.com)，或加入以下 QQ 群学习。

学习视频等资料下载地址请通过 QQ 联系汪老师：308148266

无限互联 iOS 开发学习 QQ 群：378597540

无限互联 iPhone 开发学习 QQ 群：391805984

——刘明洋

或根据本书封面上的二维码扫描进入无限互联的微信和网站进行掌上学习。

## 代码下载地址

<http://swift.leadingdo.com>

# 目 录

## 第一部分 入 门

第 1 章 Swift 开发入门	2
1.1 Swift 概述	2
1.2 搭建开发环境	3
1.2.1 开发环境的要求	3
1.2.2 系统环境搭建	3
1.2.3 Xcode 开发工具搭建	14
1.3 编写第一个程序	16
1.4 Swift 代码分析	22
1.4.1 文件构成	22
1.4.2 标识符	23
1.4.3 关键字	24
1.4.4 注释和分号	25
1.4.5 代码的构成	25
1.5 使用帮助文档	26
1.5.1 简介	26
1.5.2 下载安装	26
1.5.3 使用	27
1.5.4 Quick Help	32
1.6 Swift 基础语法	33

1.6.1 变量和常量 .....	33
1.6.2 基本数据类型及转换 .....	34
1.6.3 运算符和表达式 .....	42
1.6.4 控制流 .....	52
1.6.5 枚举和结构体 .....	57
1.6.6 宏定义 .....	59
1.6.7 代码规范 .....	60
<b>第2章 Swift 与 Cocoa .....</b>	<b>62</b>
2.1 字符串 .....	62
2.1.1 初始化字符串 .....	62
2.1.2 字符串常用方法介绍 .....	63
2.2 集合类型 .....	68
2.2.1 数组 .....	68
2.2.2 字典 .....	71
2.2.3 集合的可变性 .....	74
2.3 解包 .....	75
2.4 类 .....	75
2.4.1 类和对象 .....	75
2.4.2 属性 .....	76
2.4.3 方法 .....	78
2.4.4 下标脚本 .....	80
2.5 函数和闭包 .....	81
2.5.1 函数 .....	82
2.5.2 闭包 .....	84
2.6 初始化与反初始化 .....	86
2.7 id 兼容性 .....	91
2.8 nil 对象 .....	92
2.9 继承 .....	93

2.9.1 继承	93
2.9.2 重写	94
2.10 扩展	95
2.10.1 扩展的定义	95
2.10.2 扩展属性	96
2.10.3 扩展构造器	97
2.10.4 扩展方法	97
2.10.5 扩展下标脚本	98
2.10.6 扩展嵌套类型	99
2.11 协议	100
2.12 自动引用计数 (ARC)	107
2.12.1 自动引用计数的工作机制	107
2.12.2 类实例之间的循环强引用及解决办法	108
2.12.3 闭包引起的循环强引用及解决办法	113
2.13 多线程	116
2.13.1 NSThread 的使用	116
2.13.2 NSOperation 和 NSOperationQueue 的使用	120
2.13.3 Grand Central Dispatch (GCD) 的使用	121
2.14 新特性 Playground	130

## 第二部分 实 战

第3章 UI 基础	138
3.1 iOS 开发平台限制和方式	138
3.2 iOS SDK 介绍	141
3.3 Xcode 6 详细介绍	144
3.3.1 Xcode 介绍	144
3.3.2 Interface Builder 工具	151

3.3.3 模拟器 .....	160
3.4 Auto Layout 和 Size Classes .....	166
3.4.1 Auto Layout .....	166
3.4.2 Size Classes .....	176
<b>第 4 章 UI 开发与控件 .....</b>	<b>182</b>
4.1 窗口与视图 .....	182
4.1.1 UIScreen、UIWindow 和 UIView .....	182
4.1.2 界面旋转和大小处理 .....	188
4.2 图片资源的使用 .....	191
4.3 UILabel .....	201
4.4 UITextField .....	206
4.5 UIButton .....	216
4.6 UITableView .....	221
4.6.1 UITableView 基本使用 .....	221
4.6.2 UITableView 高级使用 .....	233
4.6.3 自定义 UITableViewCell .....	240
4.6.4 创建分组表 .....	244
4.7 UITextView .....	249
4.8 UISegmentedControl .....	259
4.9 UISlider .....	262
4.10 UISwitch 和 UIStepper .....	265
4.11 UIActivityIndicatorView 和 UIProgressView .....	266
4.12 UIAlertView 和 UIActionSheet .....	269
4.13 UIImageView .....	275
4.14 UIScrollView 和 UIPageControl .....	278
4.15 UIWebView .....	284
4.16 UIDatePicker 和 UIPickerView .....	290

4.17	UIToolBar 和 UISearchBar	298
4.18	UINavigationController	306
4.19	UITabBarController	314
4.20	MVC 模式	319
4.20.1	MVC 介绍	319
4.20.2	Xcode 实现 MVC	320
<b>第 5 章</b>	<b>读写应用程序数据</b>	<b>328</b>
5.1	NSUserDefaults 和对象归档	328
5.2	文件操作	335
5.3	SQLite3 存储和读取数据	343
5.4	Core Data	362
5.5	iCloud 存储	373
<b>第 6 章</b>	<b>触摸和手势</b>	<b>375</b>
6.1	触摸	375
6.2	手势 (UIGestureRecognizer)	378
<b>第 7 章</b>	<b>多媒体应用</b>	<b>385</b>
7.1	照片选择	385
7.1.1	相册图片选择	387
7.1.2	拍照	391
7.2	AV Foundation 框架	393
7.2.1	系统声音	393
7.2.2	AV 音频播放器	396
7.2.3	AV 录音机	401
7.3	Media Player 框架	405
<b>第 8 章</b>	<b>动画</b>	<b>410</b>
8.1	UIView 动画	410
8.2	CATransition 动画	414

8.3	transform 属性实现动画 .....	416
8.4	CABasicAnimation .....	419
8.5	CAKeyframeAnimation .....	420
<b>第 9 章</b>	<b>网络通信 .....</b>	<b>340</b>
9.1	判断网络连接情况 .....	423
9.2	网络请求 JSON 解析 .....	426
9.3	Get、Post 方式 .....	433
9.4	NSURLSession 加载数据、下载上传 .....	438
<b>第 10 章</b>	<b>检测设备朝向和移动 .....</b>	<b>444</b>
10.1	加速计和陀螺仪 .....	444
10.2	判断设备方向和摇晃 .....	447
<b>第 11 章</b>	<b>定位处理与地图 .....</b>	<b>451</b>
11.1	CoreLocation .....	451
11.2	MapKit .....	461
<b>第 12 章</b>	<b>地址簿、短信、电话、邮件 .....</b>	<b>469</b>
12.1	地址簿 .....	469
12.2	短信、电话、邮件 .....	483
<b>第 13 章</b>	<b>iOS 应用本地化 .....</b>	<b>488</b>
13.1	本地化字符串 .....	488
13.2	本地化图像 .....	494
13.3	本地化 Xib 文件和应用名称 .....	496
<b>第 14 章</b>	<b>OC 汇编与迁移 .....</b>	<b>498</b>
14.1	Swift、OC 互调 .....	500
14.2	将 Objective-C 代码迁至 Swift .....	502
<b>第 15 章</b>	<b>真机调试和发布 .....</b>	<b>504</b>
15.1	真机调试 .....	504



15.2 发布 .....	519
15.2.1 创建证书和描述文件 .....	519
15.2.2 上传提交审核 .....	524
<b>第 16 章 推送通知 .....</b>	<b>534</b>
16.1 本地推送 .....	534
16.2 服务器推送 .....	538
<b>总结 .....</b>	<b>546</b>

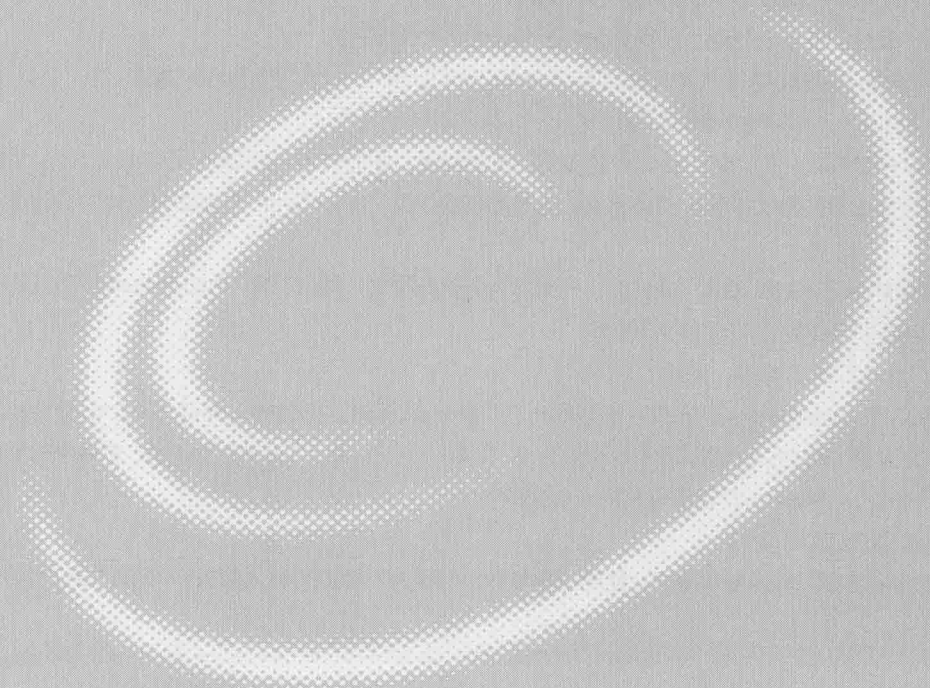
## 第一部分

---

# 入 门

### 学习目标

- 一、了解 Swift 语言
- 二、了解开发环境搭建
- 三、了解 Swift 基础语法
- 四、掌握 Swift 与 Cocoa 的交互



### 1.1 Swift 概述

#### ● 什么是 Swift

Swift 是 Apple 在 WWDC2014 所发布的一种全新的编程语言,用来开发 iOS 和 Mac OS X 应用。Swift 兼容 Objective-C 语言,并从 Python 和 JavaScript 中借鉴了一些技术,使得 Swift 更加简单、易读。可在 Mac OS X 10.9.4 及其以上版本、Xcode 6 及其以上版本中使用。它的文件扩展名是\*.swift。

Swift 语言已存在多年,Apple 在设计 Swift 时,就有意使其和 Objective-C 共存,两者可以实现混编。但有理由相信在不久的将来,Swift 会完全取代 Objective-C,成为 iOS 8、OS X Yosemite 的新宠。

#### ● Swift 的特点

1. 建立在 C 和 Objective-C 语言之上,保留了 C 与 Objective-C 的优点,它最出色的一大亮点就是摆脱了 C 的束缚,克服了 C 语言的兼容性问题。

2. 快速、现代、安全、互动,而且全面优于 Objective-C 语言。

3. 其简洁的代码风格与成熟的框架 Cocoa、Cocoa Touch 相结合,拥有了一个强大的 Framework Library,并采用安全的编程模式。

4. 增加了一些现代化的功能,使其更易使用、更加灵活、更为有趣。Xcode Playgrounds 功能是 Swift 为开发工具带来的最大创新,该功能提供强大的互动效果,能让 Swift 语言编码过程中实时显示其运行结果。

5. Apple 基于已有的编译器、调试器、框架作为其基础架构。通过 ARC (Automatic Reference Counting, 自动引用计数) 来简化内存管理。

#### ● 和 Objective-C 共存、混编

Swift 采用了 Objective-C 的命名参数和动态对象模型,提供了无缝接入现有 Cocoa 框架以及 Objective-C 混编的能力。但是不同于 Objective-C 和 C++或者 C 在同一个.mm 文件中的混编,Swift 代码不能和 Objective-C 代码写在同一个文件中。

#### ● Swift 适合任何人学习

Swift 语言的主要目标是简单易学、使用便捷高效。苹果并不强迫所有人都去学习新的语言。但也会

吸引更多的开发者来到苹果的平台。过去，每当他们带来一些巨大的技术变革（就像是自动引用计数，Automatic Reference Counting）时，苹果总会给开发者留一些适应时间，让大家继续用原有的方式开发，当大家逐渐发现新技术的易用性和优点以后，便会主动接纳新技术。

Objective-C 开发者会感到 Swift 的似曾相识。Swift 采用了 Objective-C 的命名参数和动态对象模型。相比 Objective-C，Swift 算得上是极致简洁。很多内容写在一行，却仍然简单易读。当然，用 Objective-C 也能实现不少功能，但是内容会很长，而且不方便阅读。有些新的功能实现，用 Objective-C 编写起来往往会让人十分头疼，但是用 Swift 却是一种快乐的享受。相信大家很快会喜欢上 Swift 语言。

Swift 对新的程序员也是友好的。它是工业级品质的系统编程语言，却又像脚本语言一样的友好。它支持 playground，允许程序员试验一段 Swift 代码功能并立即看到结果，而无需麻烦的构建和运行一个应用。

对程序员来说，熟悉 Swift 语法需要一天时间足矣。关键是如何使用 Swift 语言开发一个 App，本书将带领读者熟悉 Swift 所有基础知识，注重实践了解 App 产品开发的每个功能和环节，以使更多的人对 Swift 语言有所了解，让更多的人成为第一批 Swift 语言专家。

## 1.2 搭建开发环境

由于 Swift 开发环境需要在 Mac OS X 系统中运行，因此其环境的搭建将不同于 Windows 环境。下面介绍 Swift 开发环境的搭建方法，使读者能更好地使用各种开发工具和语言进行快速应用开发。

### 1.2.1 开发环境的要求

- 硬件

1. 苹果电脑（MacBook Pro、MacBook Air、iMac、Mac mini、Mac Pro）。
2. iPhone、iPod Touch 或 iPad 设备，主要用来测试编写好的程序（如果没有此条件，可以使用 Xcode 模拟器运行）。

- 软件

1. 苹果系统 Mac OS X（电脑系统必须升级到 OS X 10.9.4 及以上）。
2. Xcode 开发工具（Xcode6.0 及以上）。

### 1.2.2 系统环境搭建

- 现有环境升级

现有系统升级到最新 Mac OS X 的版本需要 10.9.4 及其以上版本。如果低于这个版本，我们在打

开下载的 Xcode 包 (dmg 文件) 时, 会发现 Xcode 带有禁止安装符号, 如图 1-1 所示。双击 Xcode 会看到安装对于系统版本的要求提示, 如图 1-2 所示。

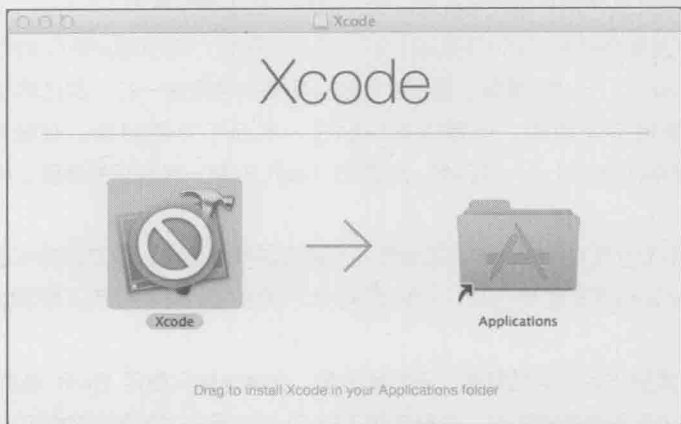


图 1-1 带有禁止符号的 Xcode 安装包

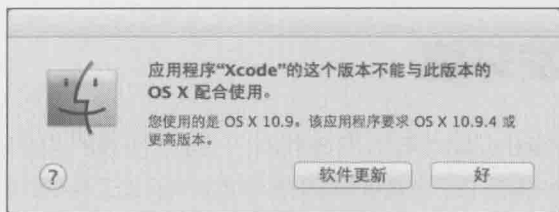


图 1-2 对系统要求的提示信息

这时我们只需要将系统升级到最新版本即可。打开“窗口左上角的苹果图标→软件更新...”, 如图 1-3 所示。我们会看到 AppStore 软件的更新界面, 如图 1-4 所示。点击全部更新即可, 系统经过重启安装之后, 就会更新到最新系统环境。

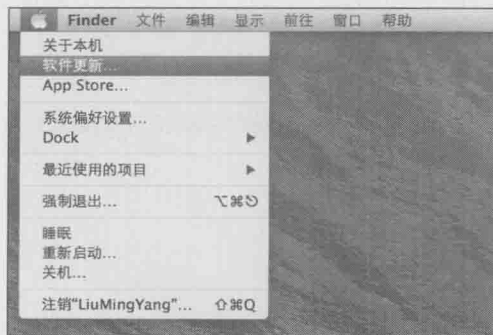


图 1-3 升级系统操作指示图

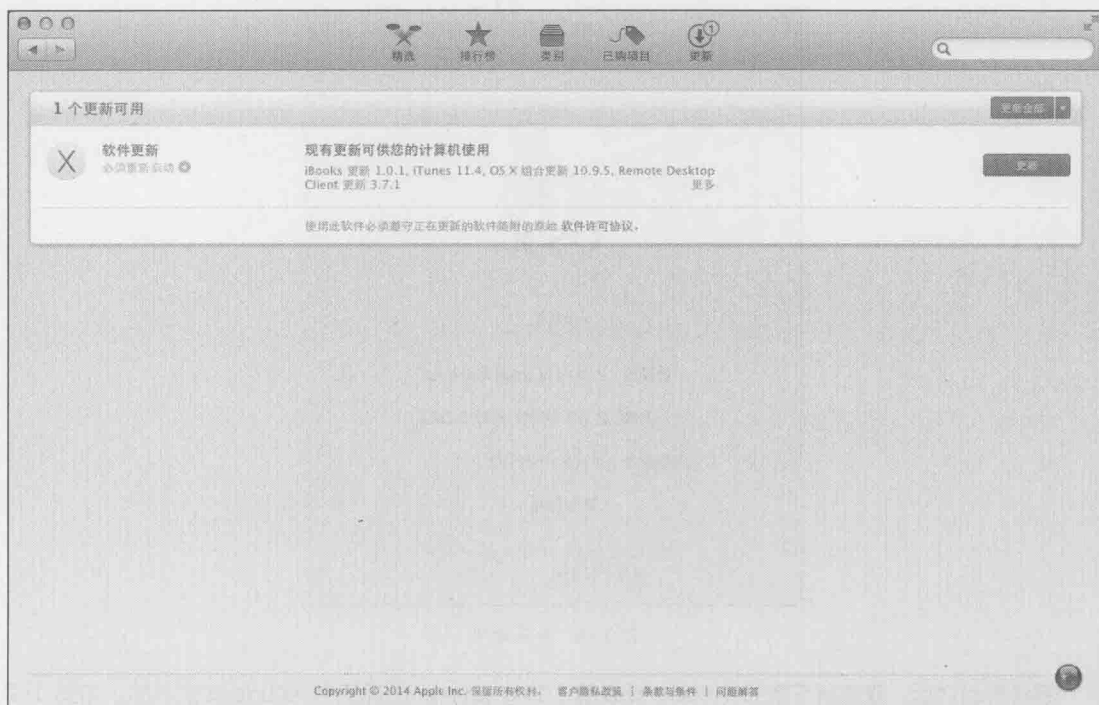


图 1-4 AppStore 软件更新界面

升级之后，可以点击“窗口左上角的苹果图标→关于本机”，如图 1-5 所示。在打开的关于本机界面会显示系统版本号，如图 1-6 所示。



图 1-5 关于本机操作指示图



图 1-6 关于本机

系统升级之后，双击所下载的 Xcode 安装包（dmg 文件），出现正常 Xcode 安装界面，如图 1-7 所示。

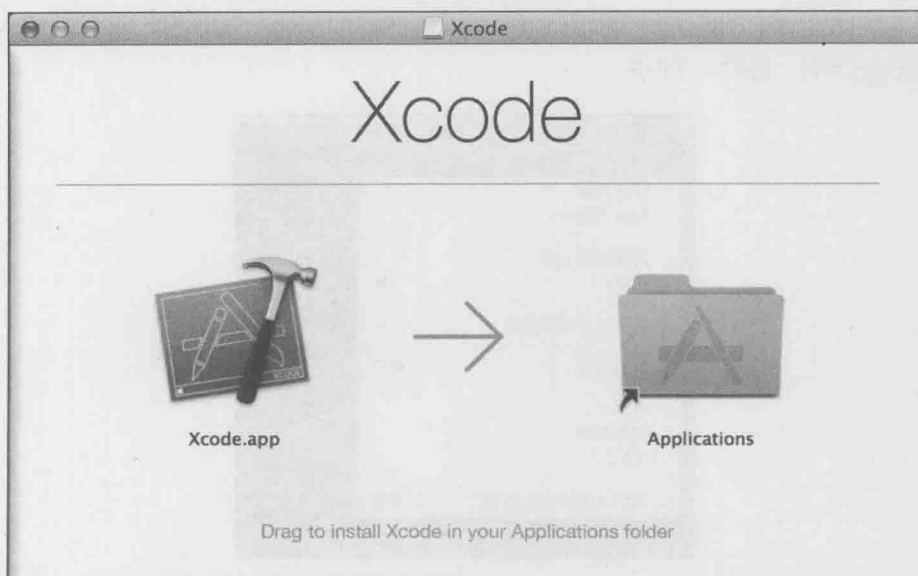


图 1-7 正常 Xcode 安装界面

下载 Mac OS X Yosemite ( 10.10 及以上版本 ), OS X 目前是免费系统, 无需序列号与 Apple ID 激活。具体下载方法如下。

1. 通过官方网站申请试用用户, Mac 开发官网地址 <https://developer.apple.com/devcenter/mac/index.action>。

2. 可通过各大网盘搜索下载安装文件。

如果通过 App Store 下载, 下载完成后它将出现在应用程序文件夹内, 如图 1-8 所示。



图 1-8 应用程序内新下载的系统示意图

双击“安装 OS X Yosemite Developer Preview.app”可以直接安装系统, 为了防止数据丢失, 安装系统前可以使用应用程序下的 Time Machine.app 进行备份。

### ● 制作 U 盘安装

制作 U 盘安装, 需要先激活系统显示隐藏文件的功能。打开“应用程序-实用工具-终端”输入命令“defaults write com.apple.finder AppleShowAllFiles YES”回车, 如图 1-9 所示。退出终端, 重新启动 Finder, 鼠标单击“窗口左上角的苹果标志→强制退出→Finder→重新启动”。

右击“安装 OS X Yosemite Developer Preview.app”显示包含内容, 如图 1-10 所示。



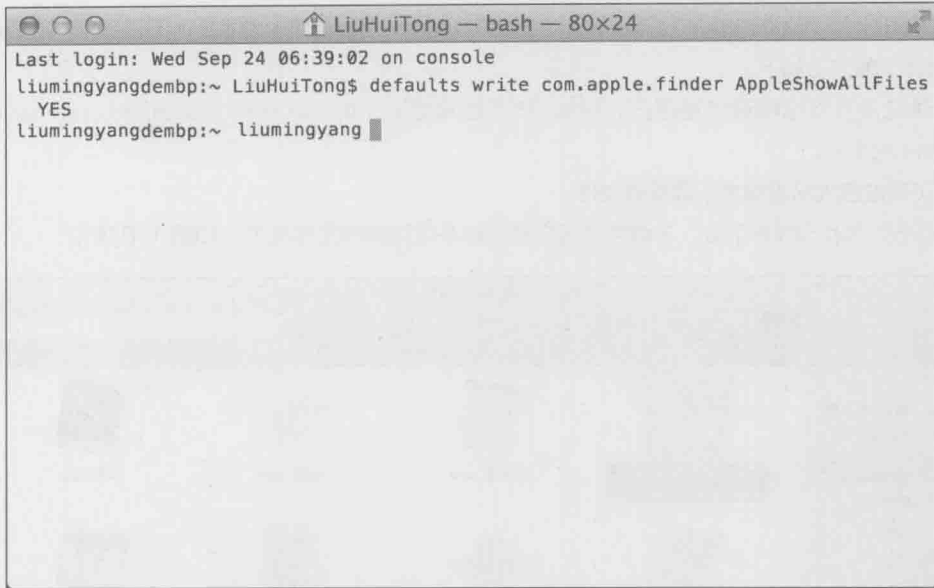


图 1-9 显示系统隐藏文件命令行操作示意图



图 1-10 查看 OS X Yosemite Developer Preview.app 内文件内容

进入目录 Content→SharedSupport 双击 InstallESD.dmg 打开后,可以看到隐藏文件 Base System.dmg, 如图 1-11 所示。

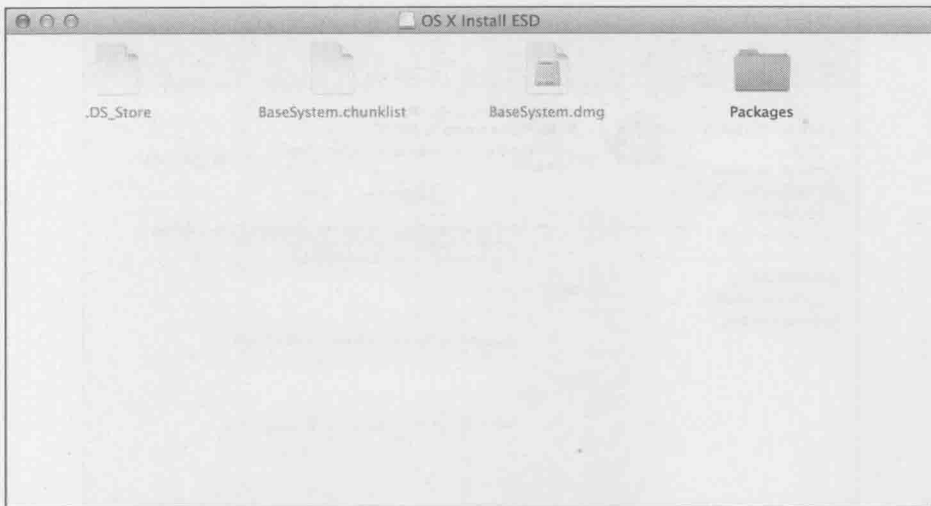


图 1-11 InstallESD 目录

打开应用程序→实用工具→磁盘工具，选择要恢复使用的 U 盘设备，点击恢复，将 Base System.dmg 文件拖到源磁盘位置，将 U 盘拖到目标磁盘位置，点击右下角恢复按钮，如图 1-12 所示。

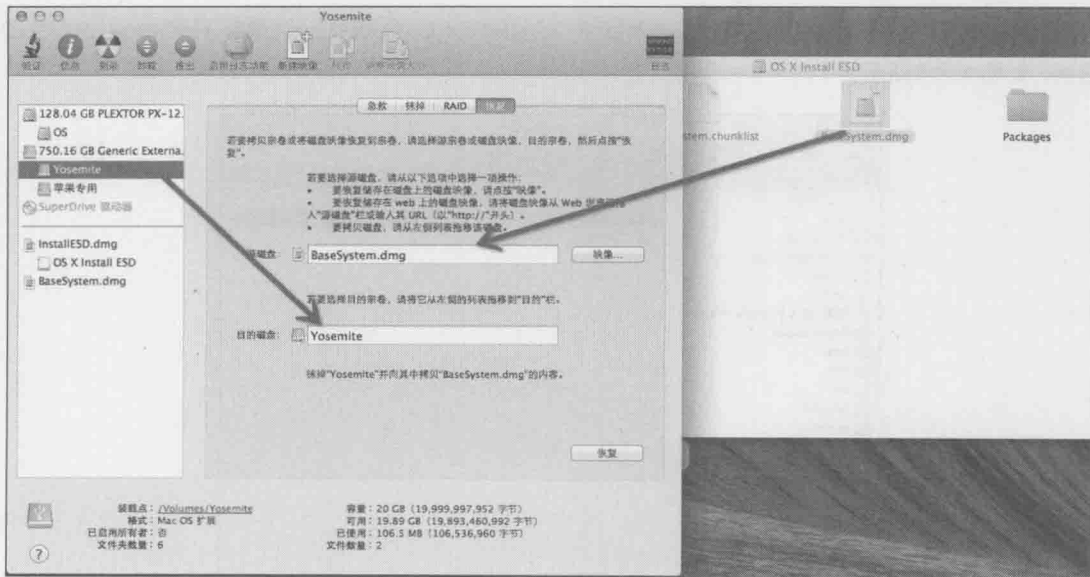


图 1-12 恢复 BaseSystem.dmg 内容到 U 盘内

要注意恢复过程将会抹掉磁盘上的原有数据，如图 1-13 所示。

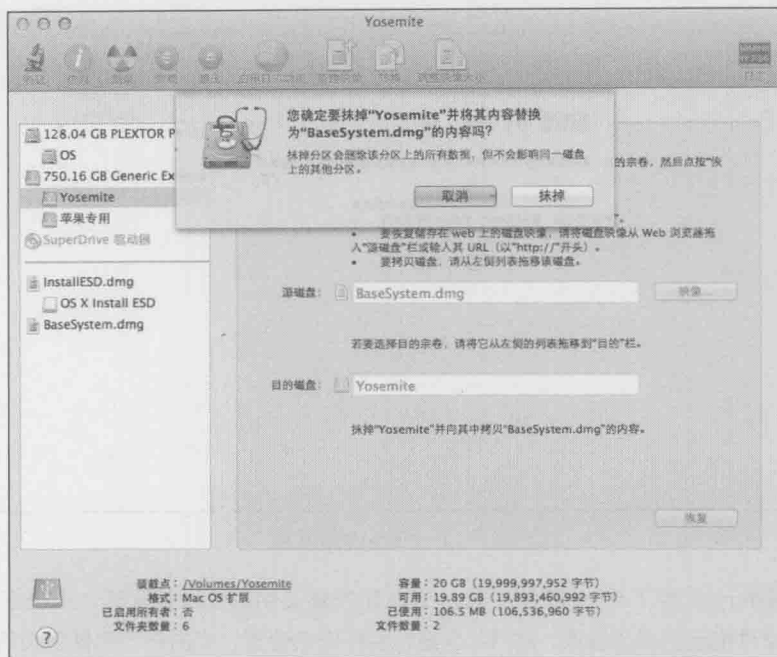


图 1-13 替换 U 盘内容提示

恢复完成后，打开 U 盘上的文件，进入目录“System-Installation”，先删除 Package 这个快捷方式，如图 1-14 所示。

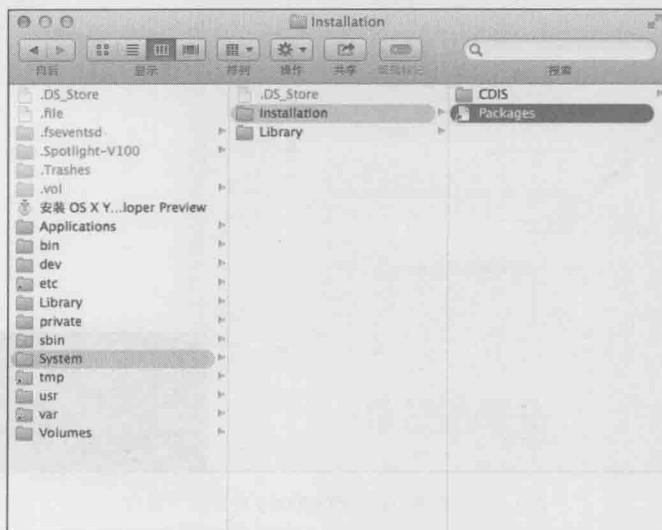


图 1-14 恢复成功之后 U 盘内容

将原安装文件夹 OS X Install ESD 内的 Packages 文件夹复制到“System – Installation”目录内，如图 1-15 所示。

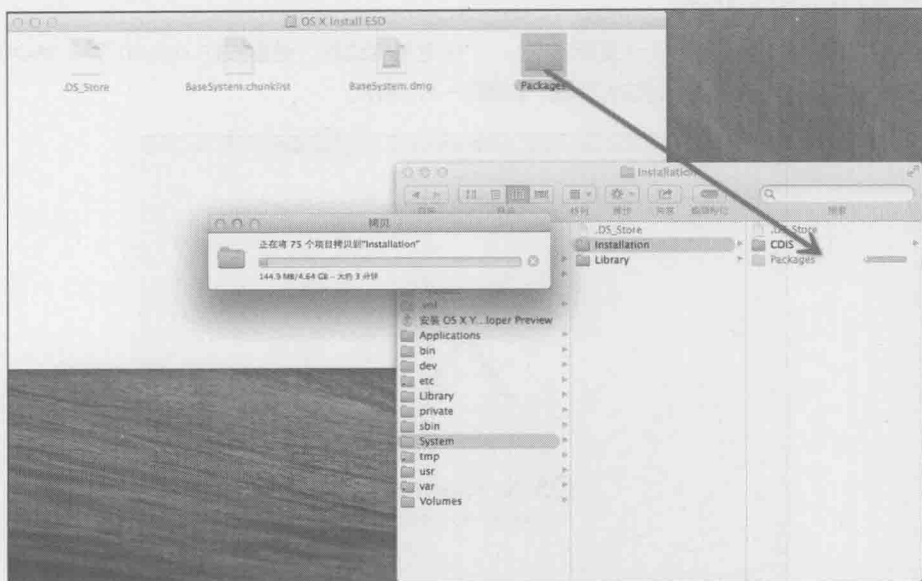


图 1-15 复制 Packages 文件夹

再将 OS X Install ESD .dmg 文件内的 BaseSystem.dmg 与 Basesystem.chunklist 复制到 U 盘的根目录内，如图 1-16 所示。

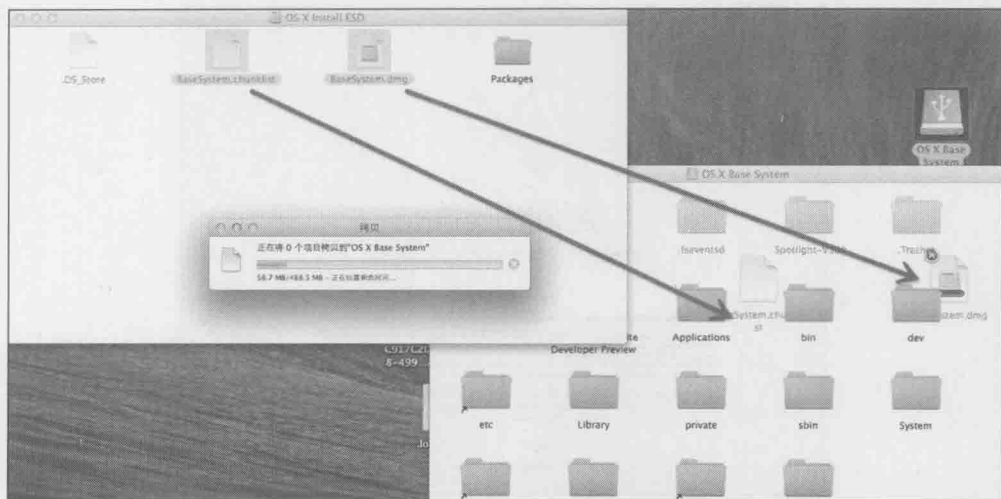


图 1-16 OS X Install ESD dmg 内文件复制到 U 盘根目录

拷贝完成后，再隐藏系统隐藏文件即可。这个 USB 系统盘也就做好了。隐藏系统隐藏文件的命令是“defaults write com.apple.finder AppleShowAllFiles NO”。

### ● 安装 Mac OS X 10.10

打开“窗口左上角的苹果图标→重新启动...”，在电脑启动后，键盘按住 option 键，就会出现启动盘选择界面，选择 U 盘，就进入到装机界面，如图 1-17 所示。

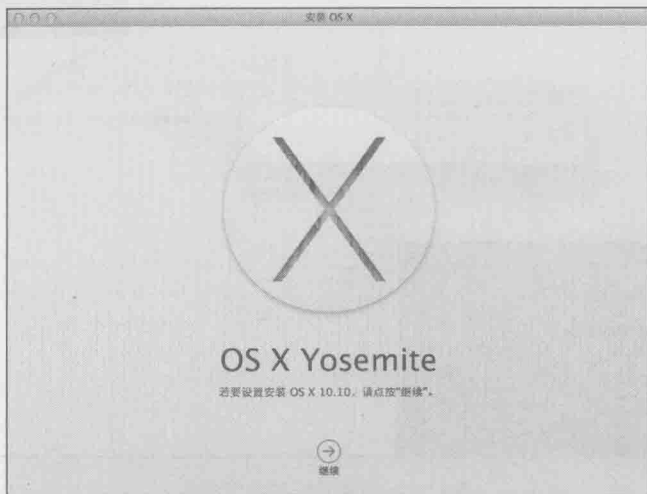


图 1-17 OS X Yosemite 安装启动界面

点击继续，进入到许可协议界面，如图 1-18 所示。

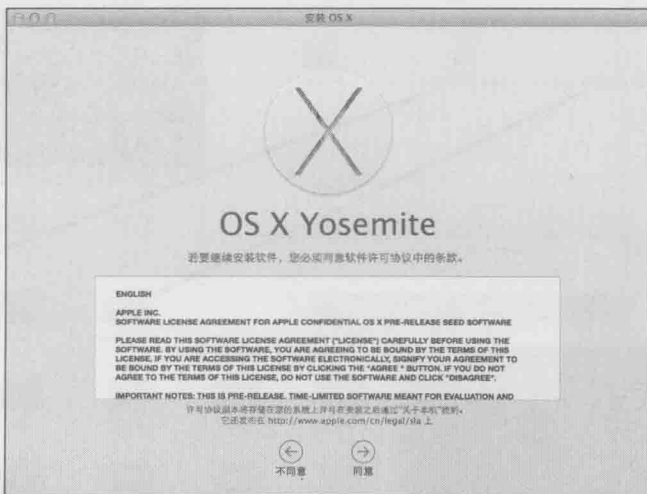


图 1-18 OS X Yosemite 安装许可协议界面

同意许可协议，进入到选择系统盘界面，如图 1-19 所示。

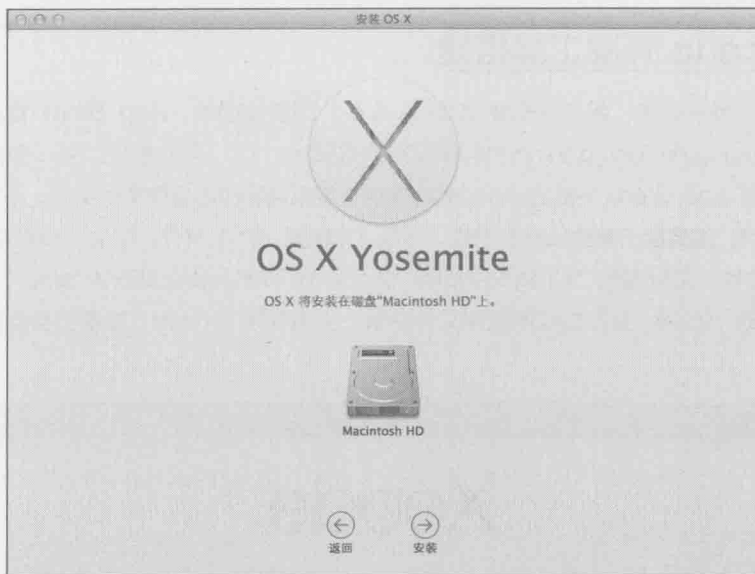


图 1-19 OS X Yosemite 安装系统盘选择界面

选择要安装到的系统盘，点击安装，进入到安装界面，如图 1-20 所示。



图 1-20 安装进度界面

经过十几分钟的安装，系统重启之后，就顺利安装上了 OS X Yosemite 系统。

### 1.2.3 Xcode 开发工具搭建

如果满足以上系统要求，就可以安装 Xcode 6.0.1 或更高版本。App Store 官网下载地址是：<https://itunes.apple.com/cn/app/xcode/id497799835?mt=12>，另外也可以去一些网盘下载，下载速度会更快。通过 App Store 下载的 Xcode 将直接下载到电脑应用程序文件夹内，下载成功之后，只需要双击运行即可，如果自己单独从网上下载，下载下来的是 dmg 文件。Mac 上软件的安装很简单，我们打开 dmg 文件，可以看到“Drag to install Xcode in your Applications folder”，如图 1-21 所示，意思就是拖动 Xcode 到右边应用程序文件夹里，实现安装 Xcode，接着会自动进行拷贝，如图 1-22 所示。

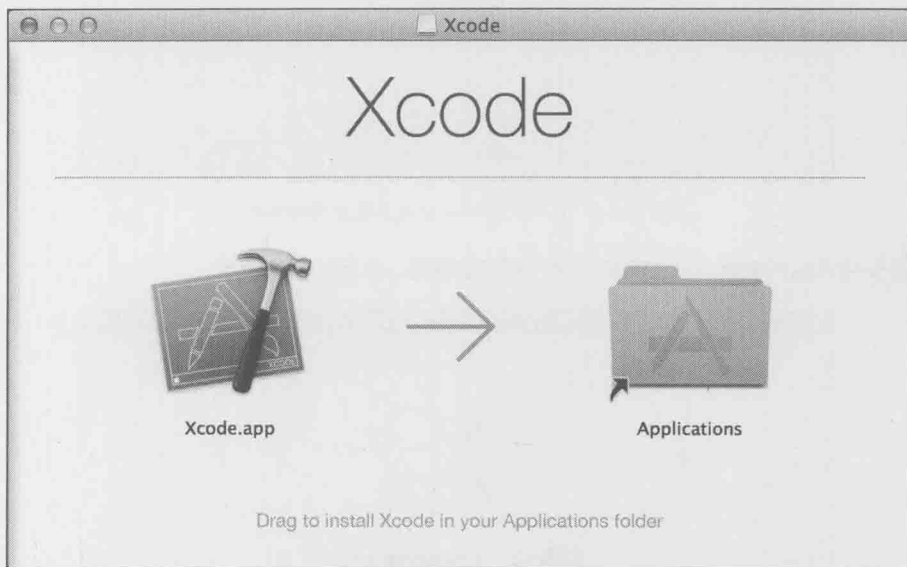


图 1-21 正常 Xcode 安装界面

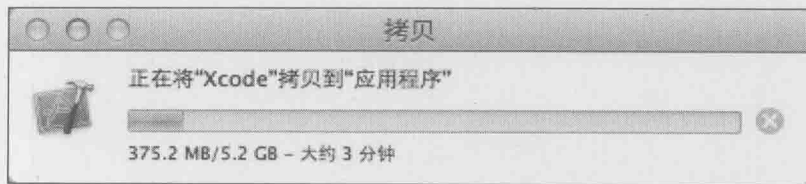


图 1-22 Xcode 拷贝界面

拷贝完成之后，我们打开应用程序文件夹，就可以看到 Xcode，如图 1-23 所示。



图 1-23 应用程序目录

双击 Xcode，首次运行将会出现先授权合约，按下 Agree 后，会进行最后的原件安装动作。如图 1-24 和图 1-25 所示。



图 1-24 Xcode 安装许可协议界面

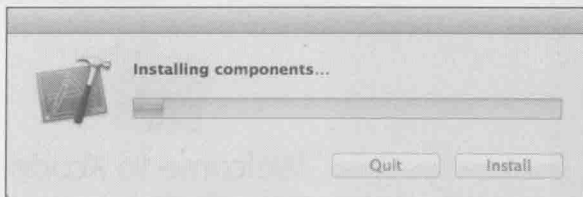


图 1-25 Xcode 原件安装界面



安装完成之后, 就可以看到 Welcome to Xcode 的界面, 如图 1-26 所示, 至此我们就可以开启 Swift 开发之旅了。

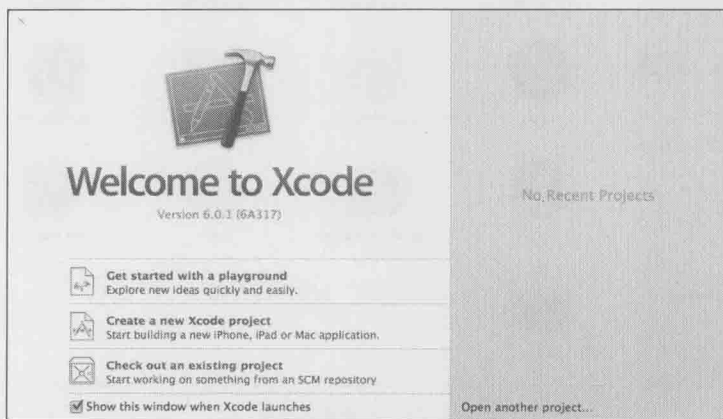


图 1-26 Xcode 欢迎界面

## 1.3 编写第一个程序

打开 Xcode, 会看到 Welcome to Xcode 界面, 如图 1-26 所示。Swift 语言可以开发 iOS 和 OS X 两个平台的程序, 如果有苹果开发者账号, 可以选择 iOS 平台, 因为可以在 iPhone 真机上面测试全部的功能。如果没有开发者账号, 可以考虑选择 OS X 平台, 通过输出台查看运行结果, 也可以选择 iOS 平台通过 iOS 模拟器来测试 Swift 程序。点击工具栏 File → New → Workspace 建立一个新工作区, 方便将后续学习的 Project 都放在一起, 如图 1-27 所示。



图 1-27 创建 Workspace 操作指示图

接着我们展开对话框，点击 New Folder，输入 SwiftClass，点击 Create 来创建一个主目录，如图 1-28 所示。

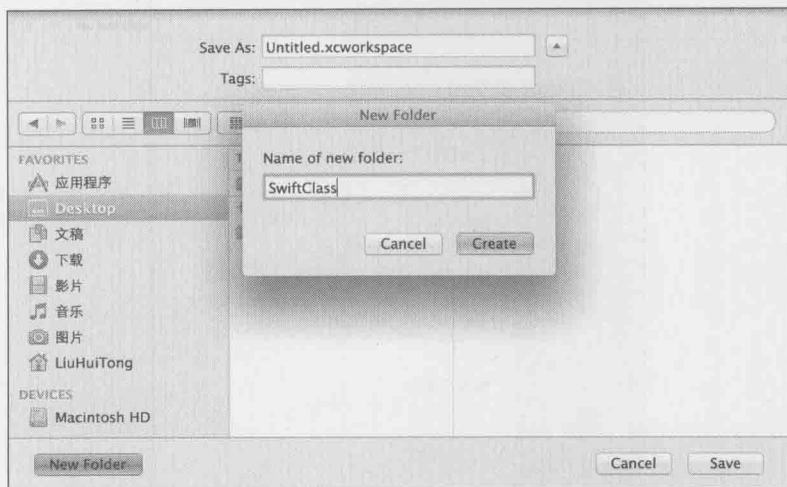


图 1-28 创建 Workspace 主目录

在 Save As 位置输入工作区名称为 SwiftClass，点击 Save 完成建立工作，如图 1-29 所示。

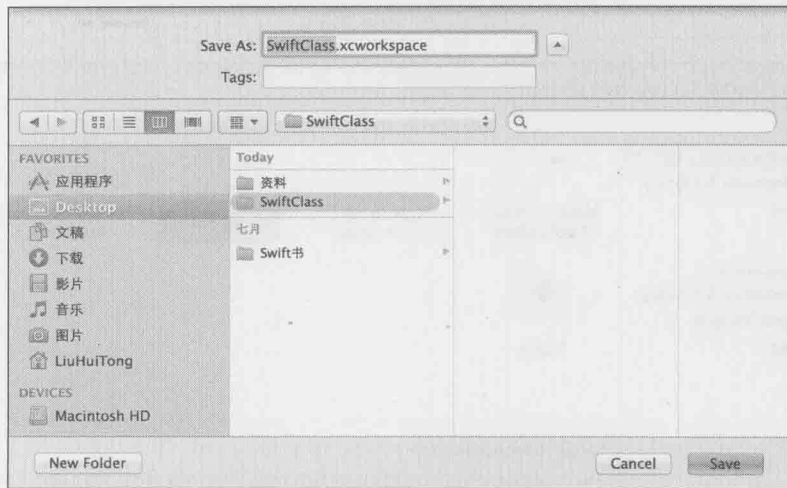


图 1-29 建立工作区

在打开的工作区内，左侧浏览区，点击鼠标右键新建一个工程 New Project，如图 1-30 所示。

选择 iOS 下的“Application”，然后在右侧工程模板列表选择一个，如“Single View Application”，如图 1-31 所示。

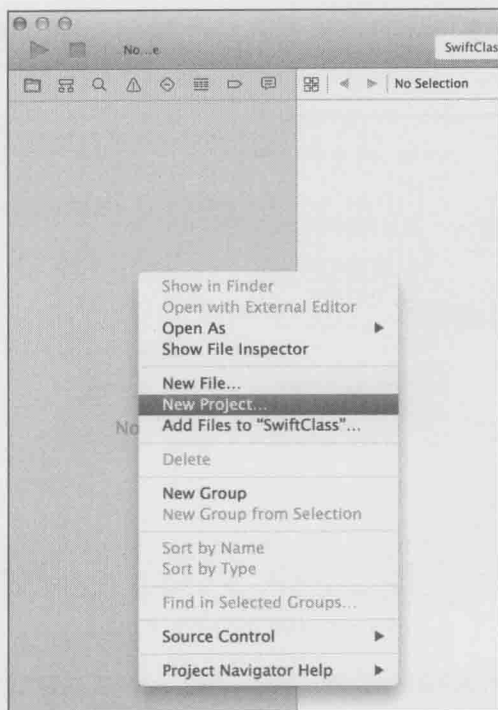


图 1-30 创建新项目

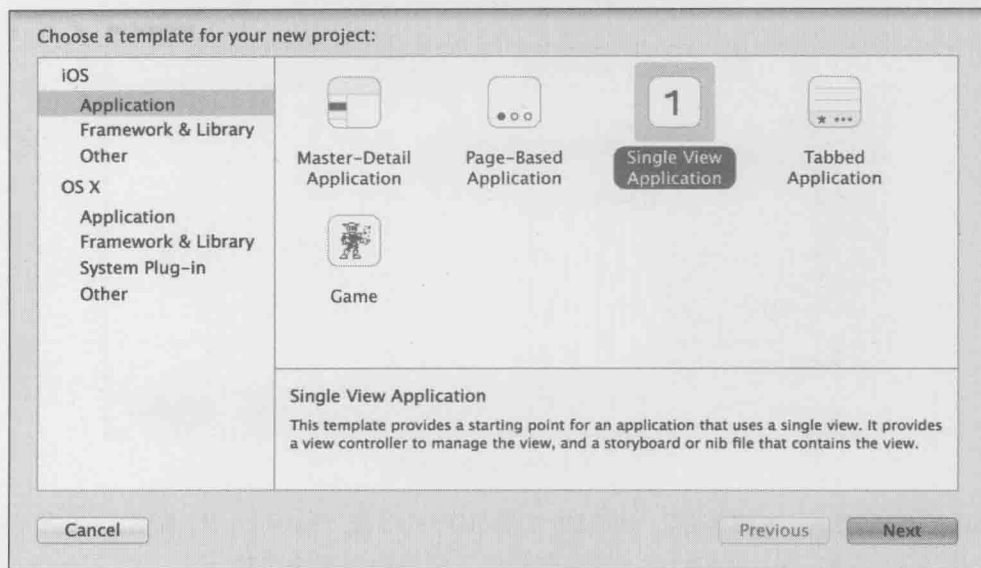


图 1-31 创建 Single View Application 项目

点击 Next 继续, 在 Product Name 位置输入 SwiftClassP1C1F3 (P1 代表第 1 部分, C1 代表第 1 章, F3 代表第 3 节, 这样后面每章节代码可以对应工程找到), Organization Name 位置输入自己的组织名称, Organization Identifier 位置输入组织标示符, 后期发布 App 时, 创建证书要与这里的 Organization Identifier 保持一致。Organization Identifier 可以在工程中重新修改。在 Language 下拉选择 Swift, 如图 1-32 所示。

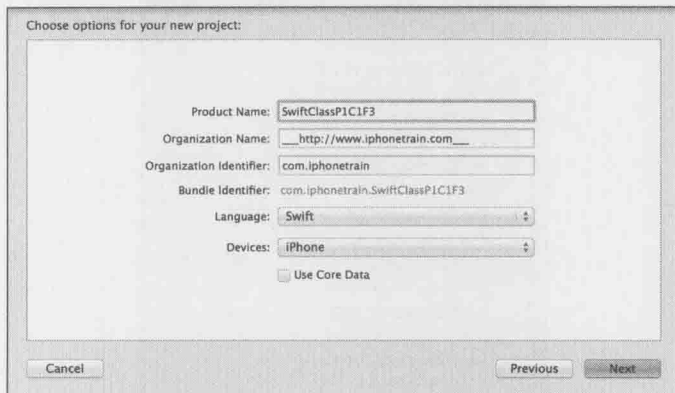


图 1-32 创建工程基本信息

点击 Next 后选择要放的那一层目录下, 我们让它与 SwiftClass 放在同一层即可, 如图 1-33 所示, 按下 Create 完成工程的创建。

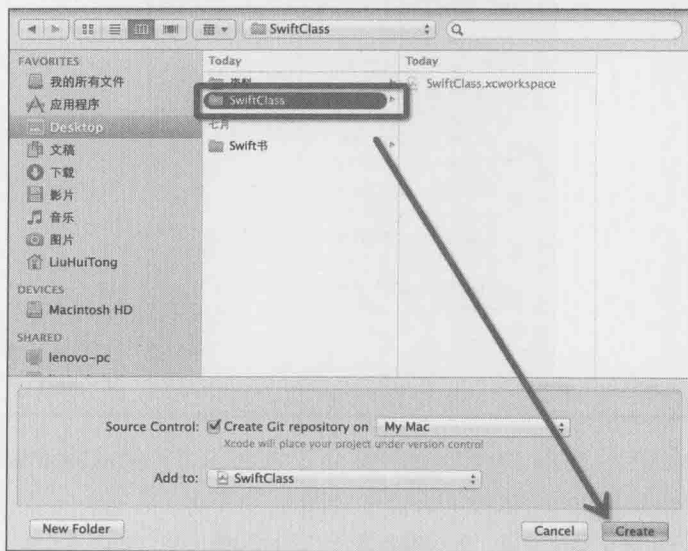


图 1-33 选择项目存储位置

在这里我们可以把后面章节的工程依次创建完成，如图 1-34 所示。

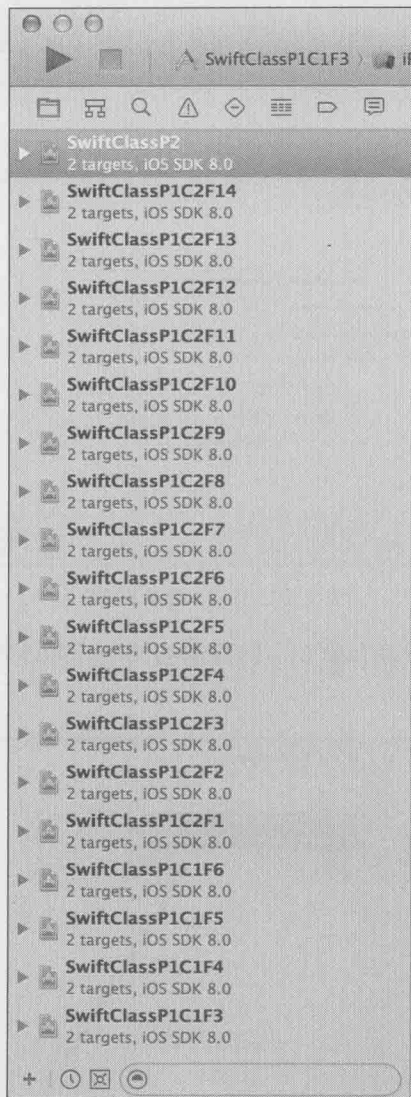


图 1-34 所有工程列表

创建项目成功之后，选择 `SwiftClassP1C1F3` 项目下的 `ViewController.swift` 文件，在 `viewDidLoad` 方法内，使用 `println` 语法书写代码，如图 1-35 所示。

在 Xcode 界面左上角工具栏位置，选择需要运行的工程和模拟器，如图 1-36 所示。这里我们选择 `SwiftClassP1C1F3` 工程和 `iPhone 6` 模拟器。

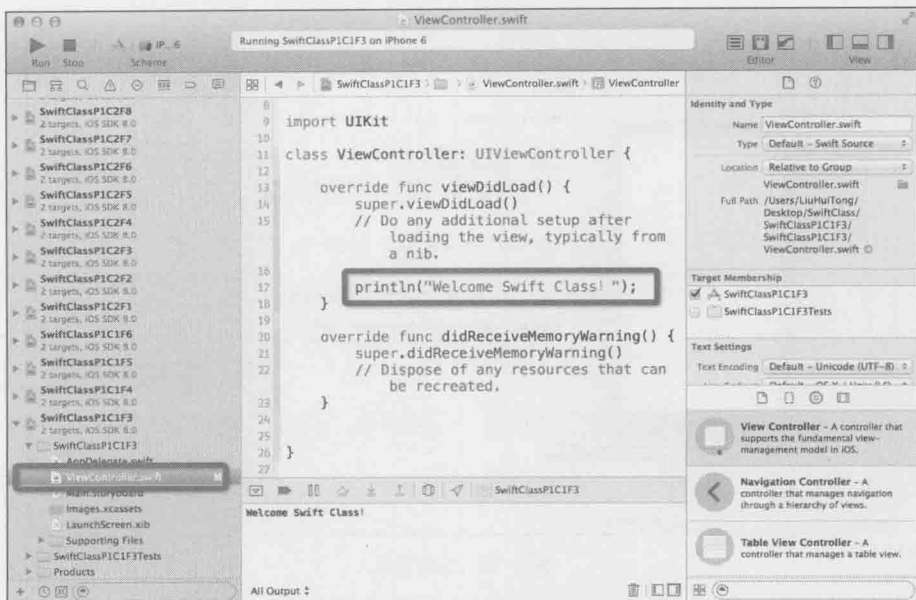


图 1-35 书写代码，创建第一个程序

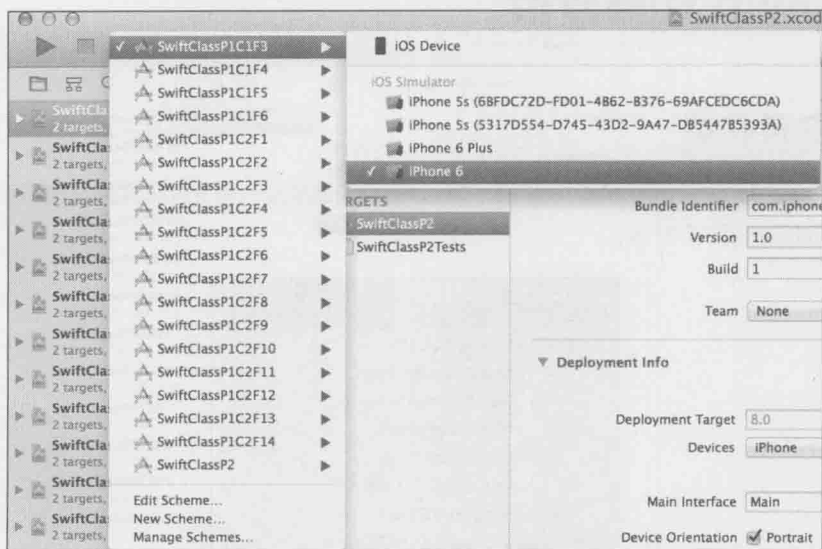


图 1-36 选择项目和模拟器

按下 `command⌘ + R` 或点击左上角的执行按钮运行程序，首先会启动 iOS 模拟器，然后在模拟器中会运行一个 iOS 程序，该程序并没有任何 UI，背景为白色。本书后面的章节会介绍如何使用 Swift 开发 iOS 应用。我们可以看到在输出区域打印出了 `Welcome Swift Class!` 字样，如图 1-37 所示。

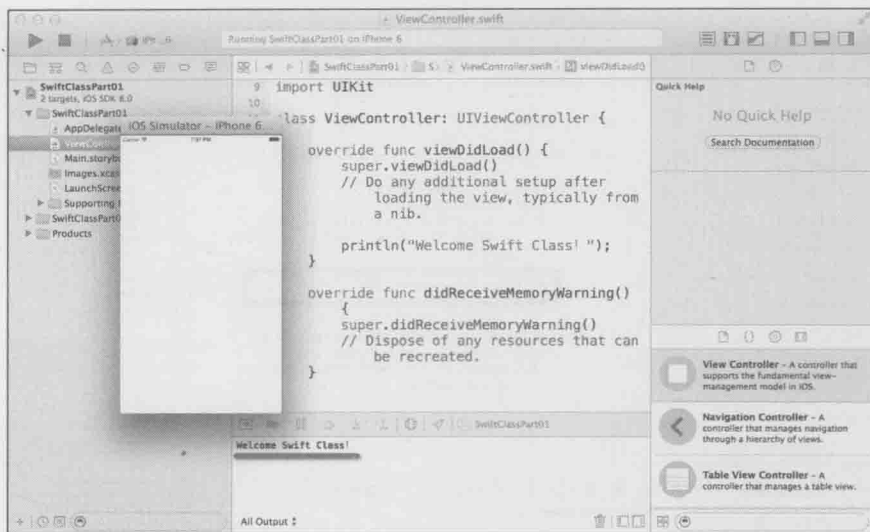


图 1-37 运行结果

## 1.4 Swift 代码分析

### 1.4.1 文件构成

创建“Single View Application”工程，Xcode 会自动生成一些文件，文件结构如图 1-38 所示（工程名为 SwiftClassP1C1F3）。

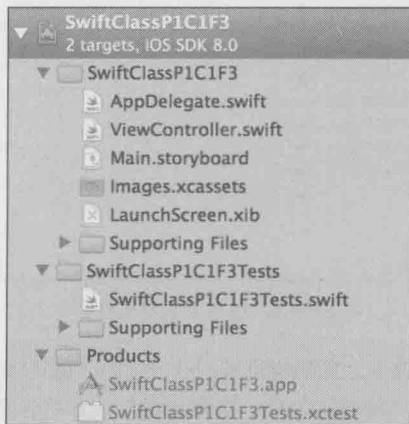


图 1-38 项目文件结构

- AppDelegate.swift 文件是应用程序的委托代理类，应用程序有时会回调该类下的一些方法。
- Main.storyboard 是应用程序的主窗口。
- ViewController.swift 是应用程序默认视图控制器，主窗口会显示该视图控制器的控件内容。
- Images.xcassets 是应用程序图片资源存放的目录，下面包含 AppIcon (APP 所有 Icon)。

可以创建其他图片项，每一项都可以设置 deployment target 以及尺寸大小。后面我们将继续介绍如何使用。

● LaunchScreen.xib 是 Xcode6/iOS 8 的新功能，应用的启动界面，但只支持 iOS 8 系统。iOS 8 中已经取消了原来的 LaunchImage, 变成一个 LaunchScreen.xib，这样比以前用一种固定图片更加灵活，可以实现一些复杂效果页面，关键还是方便适配各种尺寸的设备。在后面章节中，我们将会深入讲解 LaunchScreen.xib 的使用。

Swift 应用包含这些源文件，基本上每个源文件中都有构成该应用的函数、类和其他一些声明等。Swift 应用中的大多数源文件都是不需要按顺序访问的，即都是无顺序的，甚至可以在某个模块的最底部导入需要的源文件（虽然 Swift 不推荐这种编码风格）。我们可以根据自己的开发风格，来创建自己的目录和文件，如图 1-39 和图 1-40 所示。

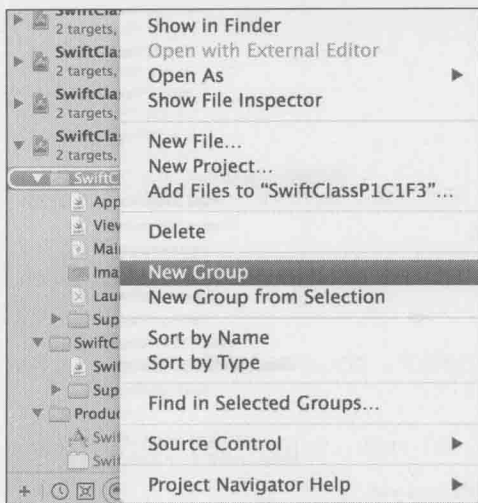


图 1-39 创建文件目录

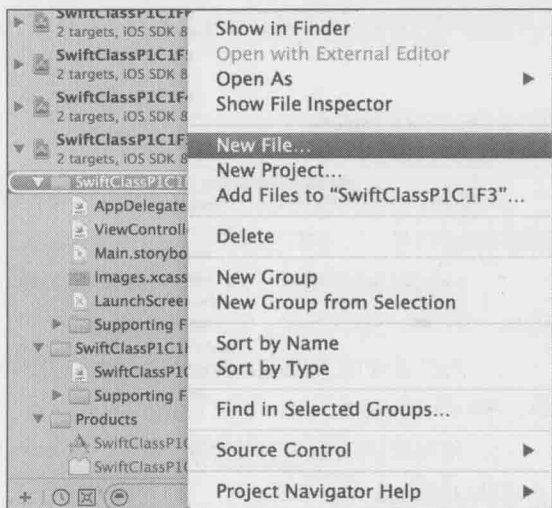


图 1-40 创建文件

## 1.4.2 标识符

标识符是开发过程中给变量、常量、方法、函数、枚举、结构体、类、协议等指定的名字。命名标识符需要有一定的规范，区分大小写，SwiftClass 与 swiftclass 是两个不同的标识符。

标识符 (identifiers) 可以由大写或小写的字母 A 到 Z、下划线\_、基本多语言面 (Basic





### 1.4.4 注释和分号

#### ● 注释

通过注释,可在代码中嵌入一些非执行文本,作为笔记或对自己的提示,以方便将来阅读。Swift 的编译器将会在编译代码时自动忽略注释的内容。

Swift 中的注释与 C 语言的注释非常相似。单行注释以双正斜杠作 (//) 为起始标记:

```
//这是一个注释
```

也可以进行多行注释,以左斜杠加星号 (/\*) 开头,并以星号加左斜杠 (\*/) 结尾。

```
/*这是一个多行注释
```

```
1. 第一行注释
```

```
2. 第二行注释
```

```
.....
```

```
.....
```

```
.....
```

```
*/
```

与 C 语言多行注释不同,Swift 的多行注释可以嵌套在其他的多行注释之中。先生成一个多行注释块,然后在这个注释块之中再嵌套成第二个多行注释:

```
/*第 1 个注释开头
```

```
/*第 2 个注释开头
```

```
第 2 个注释结尾*/
```

```
第 1 个注释结尾
```

```
*/
```

通过嵌套多行注释,可以方便地注释一大段代码,即使这段代码中已经含有了多行注释块。

#### ● 分号

与其他大部分编程语言不同,Swift 不强制要求在每条语句的结尾处使用分号(;),如果在同一行书写多条语句时,必须使用分号隔开:

```
Let `dog` = "🐶";println(dog)
```

```
//输出 "🐶"
```

### 1.4.5 代码的构成

Swift 代码一般由代码签名、头文件、执行部分组成,如图 1-41 所示。其中头文件是一个引用,在此文件中存放了特定的函数和方法定义,执行部分是开发者实现某一功能而编写的代码。

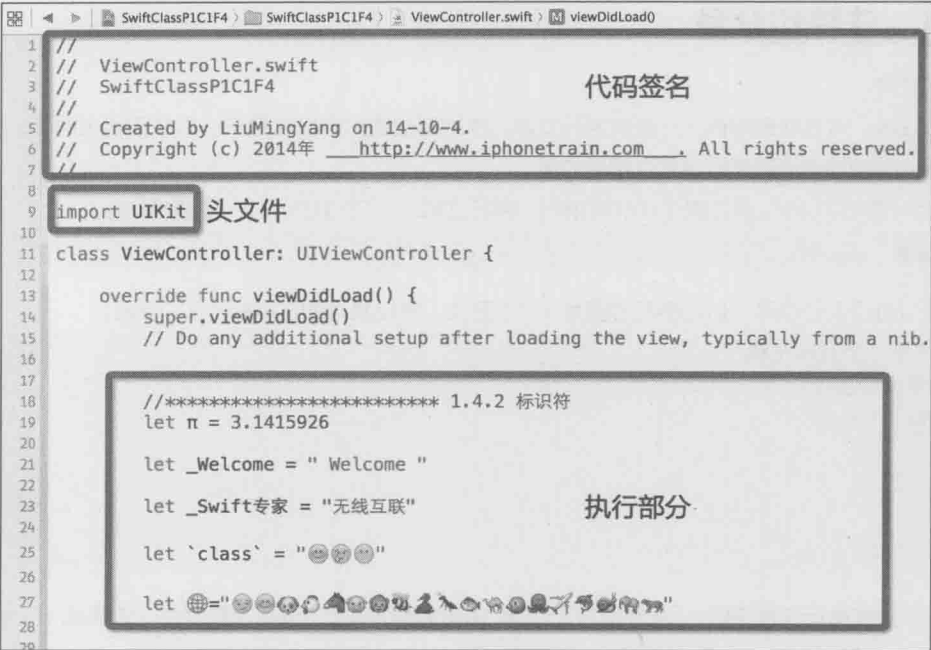


图 1-41 Swift 代码构成

## 1.5 使用帮助文档

### 1.5.1 简介

有一部分初学者在学习过程中遇到问题时，没有先从官方文档中查找解决方案，而是从微博，论坛，搜索引擎中搜索，通常情况下，这样也能够解决疑难问题。但也有可能会被错误的程序和理论经验所误导，使你误入歧途，也可能使你浪费很多时间用于从杂乱的资料中逐一筛选。

另一部分初学者担心英文文档无法看懂，或者不会有效使用苹果官方文档。因此学会如何利用苹果官方帮助文档是非常重要的。本节我们将介绍一下如何阅读帮助文档。

### 1.5.2 下载安装

首先我们要先下载帮助文档，选择 Xcode—>Preferences...菜单，然后选择 Downloads -> Documentation。在这里可以看到已经下载安装了文档库，还没有下载的文档库，可以酌情选择下载，如图 1-42 所示。

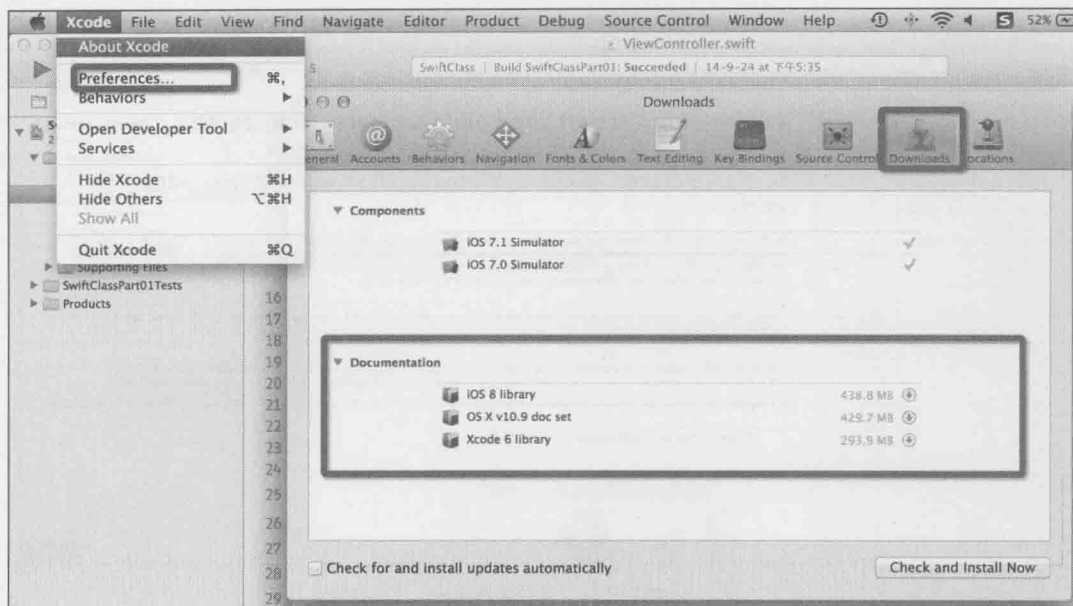


图 1-42 下载帮助文档

### 1.5.3 使用

下载成功之后, 点击 “Window→Documentation and API Reference” 打开帮助文档, 如图 1-43 所示。

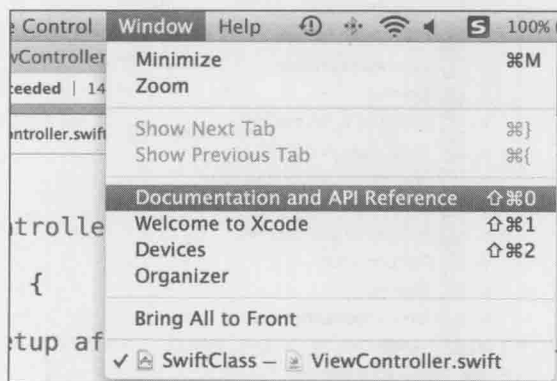


图 1-43 打开帮助文档

打开后, 整个文档界面由左面的侧栏和右面的内容区域构成。在 Xcode 里面, 一般来说有一组不同版本的 iOS 文档库、一组不同版本的 OS X 文档库, 以及一个 Xcode 文档库, 如图 1-44 所示。

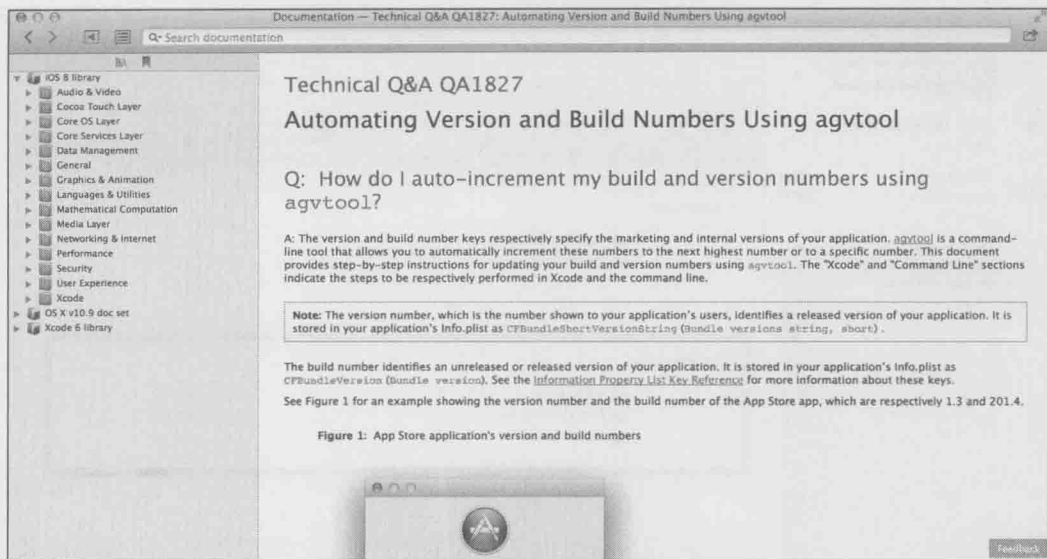


图 1-44 帮助文档界面

在文档内容区域左侧的导航区域，揭示了文档库的结构，可以选择不同的文档库，如图 1-45 所示。

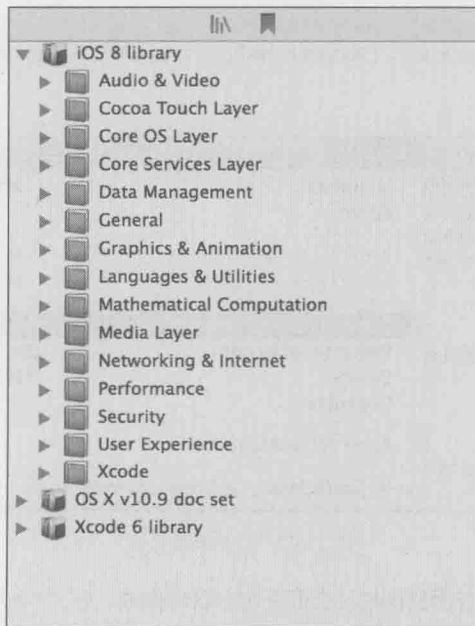


图 1-45 帮助文档导航区域

iOS 8 library 文档下可以看到很多话题，来帮你精准定位所需要的内容。

- Audio & Video——音视频。

● Cocoa Touch Layer——OS 架构中最重要层之一，Cocoa Touch 层。它包括开发 iOS 应用的关键框架，当开发 iOS 应用时，开发者总是从这些框架开始，然后向下寻找到需要的较低层框架。CocoaTouch 层包括 AddressBookUI（电话本 UI 框架）、EventKitUI、GameKit、iAd、MapKit、MessageUI、NotificationCenter、PhotosUI、Twitter、UIKit 框架，如图 1-46 所示。



图 1-46 Cocoa Touch 层

● Core OS Layer——核心 OS 层，它包含操作系统的内核环境、驱动和基本接口。内核基于 Mac 操作系统，负责操作系统的各个方面。它管理虚拟内存系统、线程、文件系统、网络和内部通信。核心 OS 层的驱动也提供了硬件和系统框架之间的接口。然而，出于安全的考虑，只有有限的系统框架类能访问内核和驱动。iPhone OS 提供了许多访问操作系统低层功能的接口集，iPhone 应用通过 LibSystem 库来访问这些功能，这些接口集如下：线程（POSIX 线程）、网络（BSDsockets）、文件系统访问、标准 I/O、Bonjour 和 DNS 服务、现场信息（LocaleInformation）、内存分配、数学计算。许多 Core OS 技术的头文件位于目录<iPhoneSDK>/usr/include/，iPhoneSDK 是 SDK 的安装目录。

● Core Services Layer——核心服务层，核心服务层为所有应用提供基础系统服务，即使不直接使用这些服务，也应该了解内置在该系统中的技术。

- Data Management——数据管理。
- General——一般性的问题。
- Graphics & Animation——图形和动画。
- Languages & Utilities——语言和工具，Objective-C 的一些知识，App Store 的管理工具等。
- Mathematical Computation——数学计算。

● Media Layer——媒体层，包括图像、音频和视频技术，可以应用这些技术在手机上创建最好的多媒体体验。更重要的是，用这些技术开发的应用将有更好的视听效果。利用 iPhone OS 高层框架可以快速地创建生动的图像和动画。媒体层包括图像技术（Graphics Technologies，包括 Quartz、CoreAnimation 和 OpenGL ES），音频技术（Audio Technologies，包括 CoreAudio 和 OpenAL）和视频技术（Video Technologies）。

- Networking & Internet——网络问题。
- Performance——性能。
- Security——安全。
- User Experience——用户体验。

- Xcode——Xcode 开发环境。

帮助文档内容区域的上面搜索框。实现文档的过滤，如图 1-47 所示。

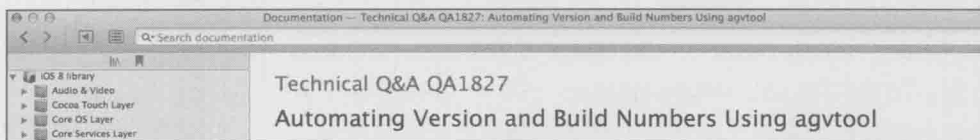


图 1-47 帮助文档搜索框

如果想查看 UILabel 类的相关内容，只需要在左面的导航中，依次展开 Cocoa Touch Layer→UIKit→Resource，点击 UILabel class Resource，右侧内容区域即可出现相关介绍。另外也可以利用搜索框在上面的文档过滤器，输入 UILabel，也可以搜索 UILabel Class 相关介绍，如图 1-48 所示。

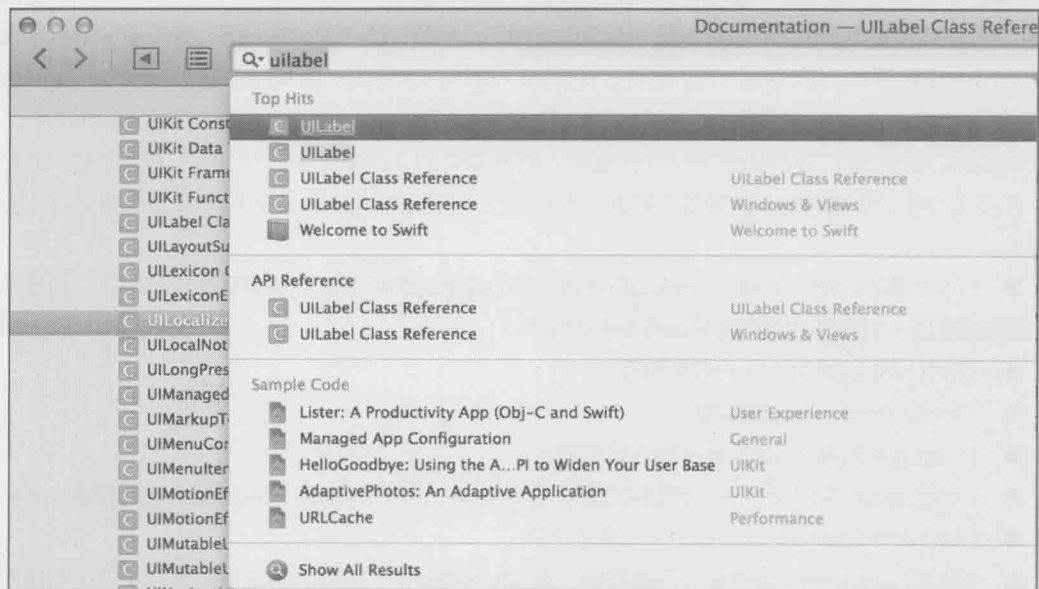


图 1-48 帮助文档搜索过滤器

帮助文档的阅读界面如图 1-49 所示。

需要注意的是，标题下面的几种说明。

- Inherits from 继承关系，继承自。
- Conforms to 遵循什么协议。
- Framework 属于什么框架。
- Declaration 相关例子代码。
- Availability 从什么 iOS 版本开始支持。

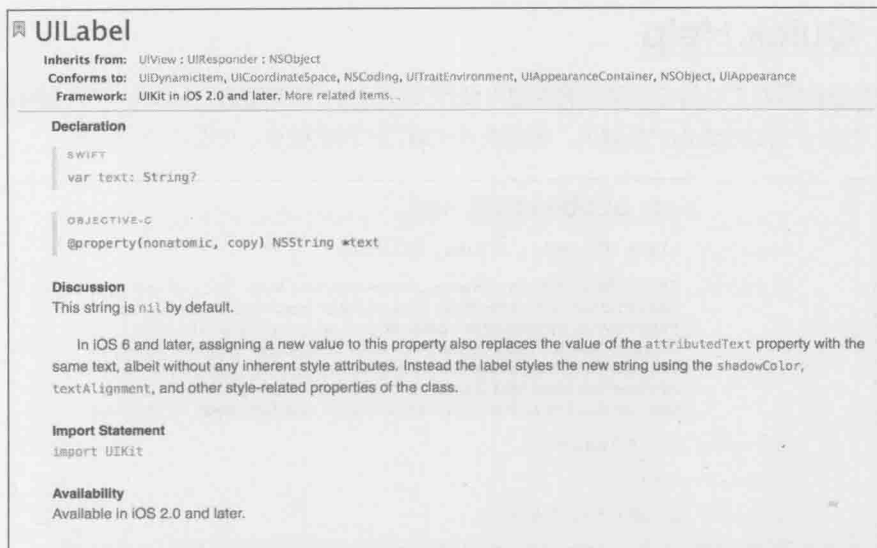


图 1-49 阅读界面

这里介绍 UILabel 所有相关的属性, 比如 Text、font、textColor、textAlignment 等, 有兴趣的读者可以慢慢浏览, 里面介绍得非常详细。

帮助文档中也是有很多代码示例可供借鉴, 依次展开 Cocoa Touch Layer->UIKit-> Sample Code, 我们会看到有很多工程代码, 点击“Open Project”即可运行工程, 如图 1-50 所示。

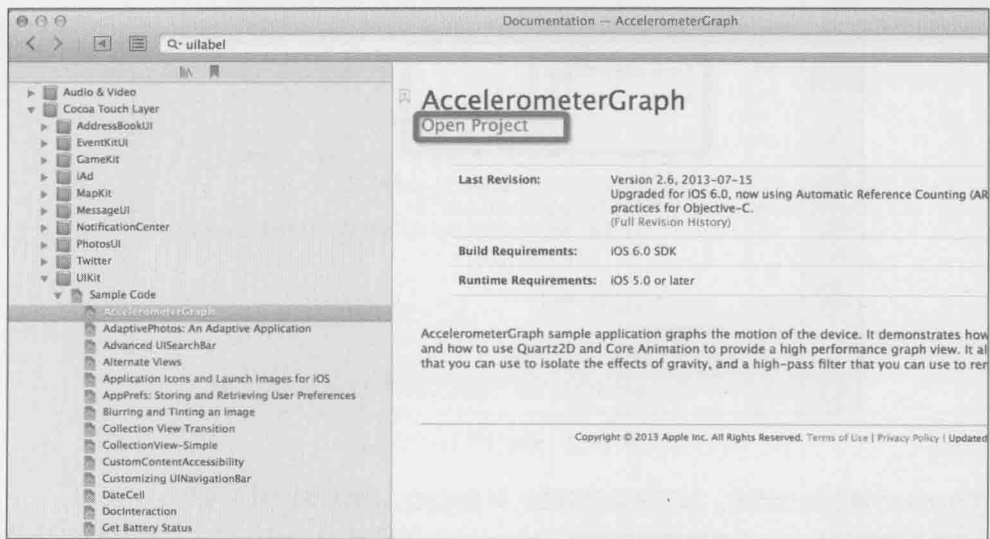


图 1-50 运行示例代码



## 1.5.4 Quick Help

最快捷的查询帮助文档的方法是不需要键入任何关键词的。只需要在 Xcode 代码编辑器里，按住 Option 键，然后点击你想查询的关键词，就会获得关键词的帮助信息，如图 1-51 所示。

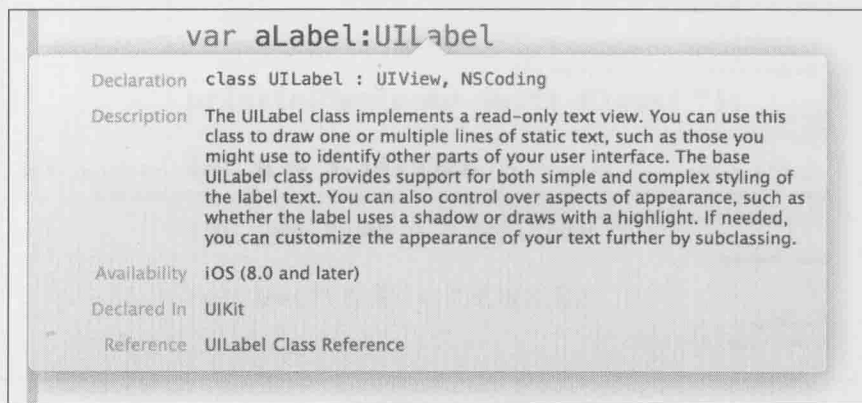


图 1-51 快捷查询 Cocoa 库的内容

帮助信息包括：一些简单的描述、从哪一个系统版本开始支持、头文件、参考文档。头文件和参考文档是可以直接点击的。

即使你点击的关键字不是 Cocoa 库的内容，而是自己代码里面的类或者方法，也可以获得相关的定义信息，如图 1-52 所示。

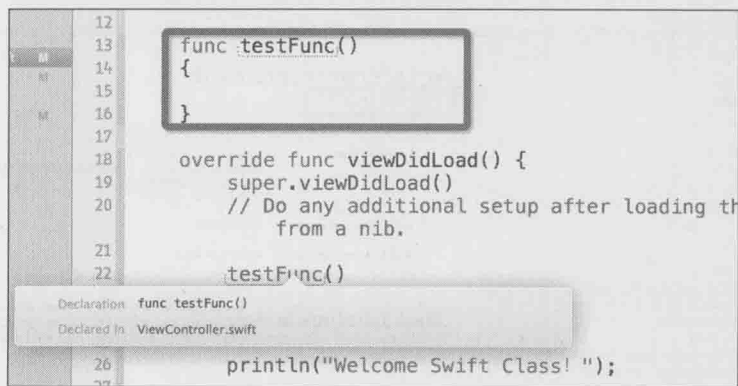


图 1-52 快速查询自己的方法

Command 键加鼠标点击，即可跳转到对应的类名或者方法名所定义的头文件。

快速查询帮助的另外一个方法是直接打开 Quick Help 栏，如图 1-53 所示，首先找到“右侧栏开关”，然后找到“Quick Help”开关即可打开。

---

本书仅提供部分阅读，如需完整版，请联系QQ: 461573687

提供各种书籍pdf下载，如有需要，请联系 QQ: 461573687

PDF制作说明：

本人可以提供各种PDF电子书资料，计算机类，文学，艺术，设计，医学，理学，经济，金融，等等。质量都很清晰，而且每本100%都带书签和目录，方便读者阅读观看，只要您提供给我书的相关信息，一般我都能找到，如果您有需求，请联系我 QQ: 461573687, 或者 QQ: 2404062482。

本人已经帮助了上万人找到了他们需要的PDF，其实网上有很多PDF,大家如果在网上不到的话，可以联系我QQ。因PDF电子书都有版权，请不要随意传播，最近pdf也越来越难做了，希望大家尊重下个人劳动，谢谢！

**备用QQ:2404062482**