Broadview
www.broadview.com.cn

FK 疯狂软件教育 www.fkit.org 标准教材

疯狂Swift讲义

李刚 编著

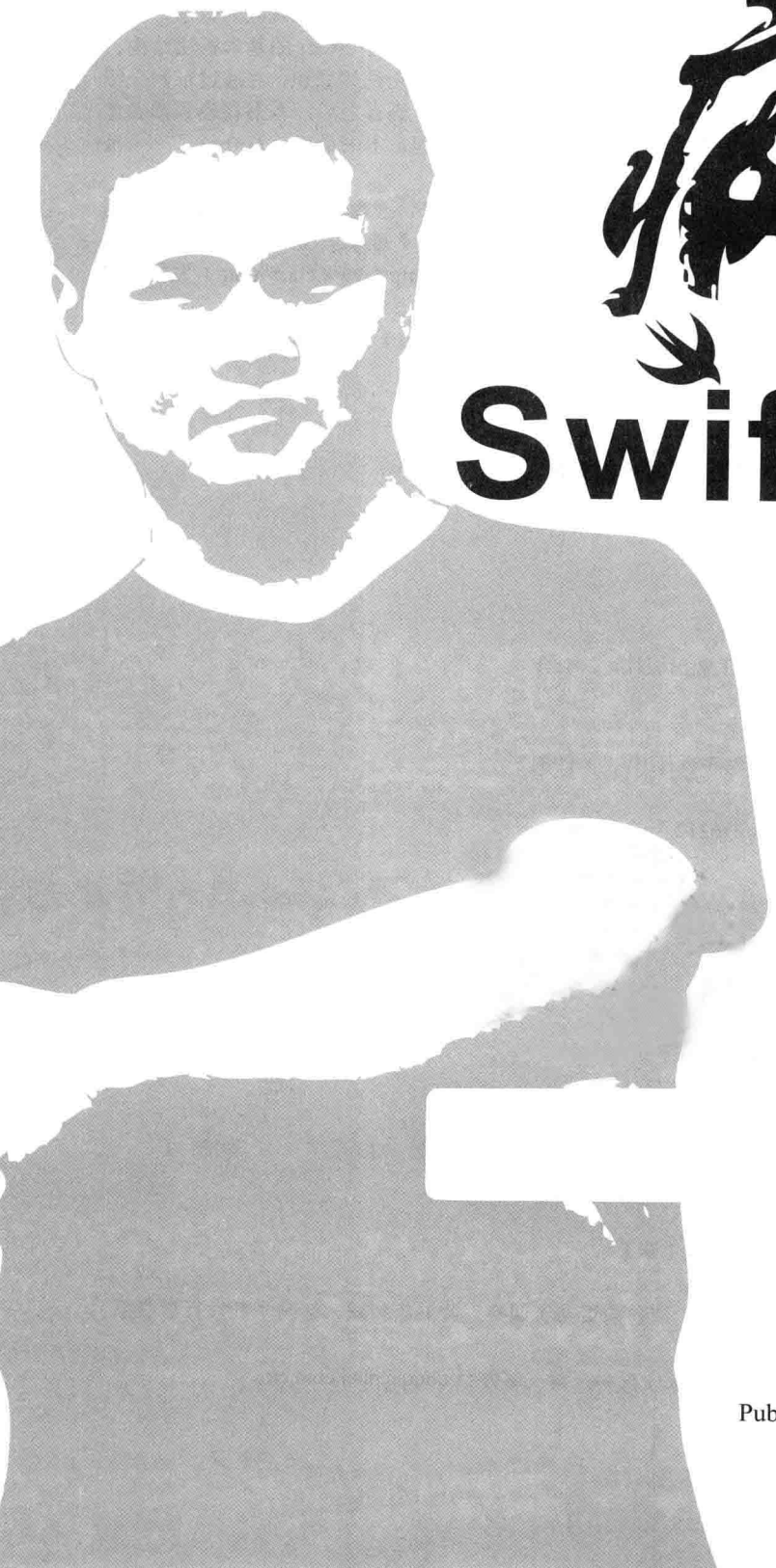
疯狂源自梦想

技术成就辉煌

疯狂源自梦想

技术成就辉煌

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn



疯狂Swift讲义

李 刚 编著

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

Swift 是 2014 年 6 月由 Apple 公司发布的编程语言，Swift 发布之初就引起广大开发者强烈的兴趣。目前，Swift 已经发布了正式版，其语法也趋于稳定。

本书以最新的 OS X 10.10 为平台，以 Xcode 6.1 为开发工具，全面介绍了 Swift 正式版的语法，以及使用 Swift 开发 iOS 应用的知识。全书从 Swift 基本语法开始介绍，详细介绍了 Swift 的基本语法结构、Swift 函数式编程特征、Swift 的面向对象特征、Foundation 框架的核心类库用法等知识，并通过示例介绍了如何在 iOS 应用中混合使用 Swift 与 Objective-C 进行开发。掌握 Swift 语言之后，本书将带领读者掌握 iOS 应用开发的基本理论，以及 iOS 应用的 MVC 设计和事件处理编程。本书最后一章介绍了一个俄罗斯方块游戏。

本书并不局限于介绍 Swift 的简单语法，而是从“项目驱动”的角度来讲授理论，全书为 Swift 所有语法提供了大量的示例程序，大部分地方甚至从正、反两方面举例，务求使读者能举一反三地真正掌握 Swift 编程。如果读者在阅读本书时遇到了技术问题，可以登录疯狂 Java 联盟（http://www.crazyit.org）发帖，笔者将会及时予以解答。

本书为所有打算深入掌握 Swift 编程的读者而编写，适合各种层次的 Swift 学习者和开发者阅读，也适合作为大学教育、培训机构的 Swift 教材。

图书在版编目（CIP）数据

疯狂 Swift 讲义 / 李刚编著. —北京：电子工业出版社，2015.1
ISBN 978-7-121-24981-5

Ⅰ. ①疯… Ⅱ. ①李… Ⅲ. ①程序语言—程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2014）第 276212 号

策划编辑：张月萍
责任编辑：葛　娜
印　　刷：北京中新伟业印刷有限公司
装　　订：三河市皇庄路通装订厂
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：787×1092　1/16　　印张：25.75　　字数：733 千字　　彩插：1
版　　次：2015 年 1 月第 1 版
印　　次：2015 年 1 月第 1 次印刷
印　　数：4000 册　　定价：69.00 元（含光盘 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 zlts@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

2014 年 6 月 2 日，Apple 公司召开了 2014 年度 WWDC 大会，并在大会上发布了全新的编程语言：Swift。Swift 甫一发布，立即引起广大开发者强烈的热情，这是因为 iOS 应用开发在全世界范围内都非常火爆。但在 Swift 发布前，iOS 和 OS X 的开发语言只能选择 Objective-C，作为一款具有 30 多年历史的古老语言，Objective-C 不可避免地具有一些先天性的不足，而且其方括号语法让人十分别扭，和其他 C 语言风格的编程语言有着极大的区别。Objective-C 语法确实吓退了不少原本希望转行从事 iOS 开发的程序员，而 Swift 的发布不仅重燃了希望转行 iOS 开发的广大程序员的希望，也让已有的 iOS 开发者满怀热情。

出于一种全球性战略布局的考虑，Apple 公司自然希望在世界范围内赢得更为广泛的开发者，这样就可让 App Store 变得愈加强大、产品线愈加丰满，从而更牢地黏住已有的 Apple 产品用户，同时也能吸引更多新的 Apple 产品用户。

因此，新发布的 Swift 语法十分简洁，Swift 从 JavaScript、Go 等现代语言中学习了不少内容，从而让 Swift 具有非常简洁的语法，十分接近程序员的习惯。Swift 不仅能很好地兼容 Objective-C 已有的框架和库，而且 Swift 与 Objective-C 完全可以进行混合编程。此外，Swift 语言还有已经十分成熟的 Cocoa、Cocoa Touch 框架的支持，更加可靠！

需要说明的是，Swift 虽然提供了简洁的语法，但 Swift 的功能并不简单，绝对是一门具有工业品质的强大语言。虽然 Swift 编程语言面世的时间不长，但网络上已有大量关于 Swift 入门的介绍文章，大部分人（包括一些所谓的“大牛”）凭着对 Swift 的第一印象，贸然给出一些似是而非的结论：Swift 很简单，花一天时间就可以掌握 Swift 语法。如果有人抱着这种想法来学习 Swift，那基本上可以放弃阅读本书了。

Swift 绝不是一门简单的编程语言，Swift 既支持函数式编程方式，也支持面向对象的编程方式。Swift 的函数式编程方式完全支持主流的函数和闭包，传参方式既支持值传递，也支持引用传递，还支持多返回值的函数，语法功能非常丰富；在面向对象的支持方面，Swift 完全支持面向对象的封装、继承、多态等基本功能，而且 Swift 提供了枚举、结构体和类 3 种面向对象的类型，这 3 种类型都可创建实例、调用方法，而且都允许定义属性、方法、构造器、下标和嵌套类型，语法功能非常丰富。如果真正全面掌握了 Swift 的所有功能，就会发现 Swift 绝不简单。

网络上大部分人要么只是看了一下 Swift 的 HelloWorld 示例，要么只是看了 Apple 公司发布的关于 Swift 的快速入门，或者在 Playground 中试验了几个入门示例……于是人云亦云地传播：Swift 很简单。

此处告诉读者 Swift 绝非一门简单的语言，并不是想打击广大开发者的学习热情，而是希望广大开发者在开始学习 Swift 之前做好心理准备：Swift 是一门语法简洁，但功能强大的工业语言，因此它的语法功能非常丰富。本书作为“疯狂体系”Swift 学习图书，当然不是仅提供浅尝辄止的入门语法，而是真正全面、细致地介绍了 Swift 语言的各种功能，100%完整覆盖 Apple 官方的“The Swift Programming Language”编程指南，而且大部分语法功能都从正、反两方面举例，务求让读者能举一反三地真正掌握 Swift 编程。

本书有什么特点

正如前面介绍的，本书并不是一本简单的 Swift 入门教材，而是一本全面而完整的 Swift 编程手册，不仅 100%完整覆盖 Apple 官方的“The Swift Programming Language”编程指南，而且示例更加丰富，行文更加口语化，更符合中国人的学习习惯。这一点，一直都是“疯狂体系”图书的风格。

本书内容大致可分为 4 部分。

- 第 1 部分：由第 1 章组成，这部分简要介绍了 Swift 语言的相关背景，主要介绍了如何搭建 Swift 的开发环境，包括 Playground 的功能和用法，重点介绍了如何在终端窗口编译、运行 Swift 程序。
- 第 2 部分：由第 2 章到第 6 章组成，主要介绍了 Swift 的基础语法和函数式编程支持，也介绍了 Swift 的 String、Array、Dictionary 等内置类型。
- 第 3 部分：由第 7 章到第 11 章组成，主要介绍了 Swift 的面向对象编程支持，并介绍了使用 Swift 调用 Foundation 框架的常用类。
- 第 4 部分：由第 12 章到第 14 章组成，主要介绍了如何使用 Swift 开发 iOS 应用，详细介绍了 iOS 项目的结构，并归纳了 iOS 应用的 MVC 设计，在代码中获取 UI 控件的两种方式（必须先获取 UI 控件，然后即可修改它来更新用户界面），iOS 应用事件处理的 3 种方式，通过这种方式可以让读者快速掌握 iOS 编程的思路。第 14 章还介绍了一个 iOS 游戏：俄罗斯方块。

本书不是一本看完之后可以“吹嘘、炫耀”的书——因为本书并没有堆砌一堆“深奥”的新名词、一堆“高深”的思想，本书保持了“疯狂体系”的一贯风格：操作步骤详细，编程思路清晰，语言平实。只要读者想学习 Swift 编程，just do it——Swift 虽不像网络流传的那么简单，但只要认真阅读此书，并准备动手“敲代码”，就完全可以真正掌握它。无须担心没有编程基础（本书从最基础的定义变量开始讲起），也无须担心没有 Apple 电脑（网络上使用普通电脑装黑苹果的文章铺天盖地）——只要你有决心和毅力，就可以真正掌握 Swift 编程，进而掌握 iOS 应用开发。

不管怎样，只要读者在阅读本书时遇到知识上的问题，都可以登录疯狂 Java 联盟（http://www.crazyit.org）与广大学习者交流，笔者也会通过该平台与大家一起交流、学习。

本书还具有如下 3 个特点。

1．版本成熟，语法稳定

本书并非是基于 Swift 测试版写成的，而是以 OS X 10.10 为平台，基于 Xcode 6.1 写成的，此时 Swift 语言已经发布了稳定的正式版，而且语法已经十分稳定，因此读者阅读此书时极少会发现书中知识已经过时或被抛弃的情形。

2．知识全面，覆盖面广

本书或者不像某些入门级的小册子看上去那么简单，但本书全面而细致地介绍了 Swift 编程的各种语法，不仅 100%完整覆盖 Apple 官方的“The Swift Programming Language”编程指南，而且示例更加丰富。

3．注释详细，轻松上手

为了降低读者阅读的难度，书中代码的注释非常详细，几乎每两行代码就有一行注释。不

仅如此，本书甚至还把一些简单理论作为注释穿插到代码中，力求让读者能轻松上手。

本书所有程序中的关键代码都以粗体字标出，也是为了帮助读者能迅速找到这些程序的关键点。

本书写给谁看

本书为所有打算深入掌握 Swift 编程的读者而编写，适合各种层次的 Swift 学习者和开发者阅读，也适合作为大学教育、培训机构的 Swift 教材。

2014-11-24

光盘说明

一、光盘内容

本光盘是《疯狂 Swift 讲义》一书的配书光盘，书中的代码按章、按节存放，即第 2 章第 1 节所使用的代码放在 codes 文件夹的 02\2.1 文件夹下，依此类推。

另：书中每份源代码也给出了与光盘源文件的对应关系，方便读者查找。

本光盘 codes 目录下有 14 个文件夹，其内容和含义说明如下：

（1）01~14 文件夹名对应于《疯狂 Swift 讲义》中的章名，即第 2 章所使用的代码放在 codes 文件夹的 02 文件夹下，依此类推。

（2）本书的大部分 Swift 程序都是*.swift 源程序，通常推荐读者只用如下命令编译这些 Swift 程序：

```
swiftc <源程序名>
```

有些示例需要多个*.swift 源程序一起运行，此时可用如下命令编译：

```
swiftc <源程序 1> <源程序 2>...main.swift
```

关于 swiftc 编译器的用法可参考本书第 1 章。

如果读者喜欢使用 Playground，也可直接把*.swift 文件中的代码复制到 Playground 中执行。

（3）本书的小部分项目是 Xcode 项目，因此项目文件夹下包含*.xcodeproj、*.pch、*.plist 等文件，它们是 Xcode 项目相关文件，请不要删除。

二、运行环境

本书中的程序在以下环境调试通过：

（1）本书使用的操作系统为 OS X 10.10 正式版。一般来说，在 OS X 10.10 或更高版本的操作系统上都可正常使用本书光盘中的代码。

（2）安装 Xcode 6.1 正式版，包括 Xcode 6.1 对应的命令行工具。

（3）如果需要使用真机（包括 iPhone 或 iPad）运行光盘中代码，则需要读者自己加入 iOS Developer Program——这个需要向 Apple 公司每年支付 99 美元。

本书绝大部分程序基本都可以在模拟器上调试成功。

三、注意事项

（1）本书包含的 iOS 项目都是 Xcode 工程，读者只要使用 Xcode 打开这些工程即可。

（2）在使用本光盘的程序时，请将程序拷贝到硬盘上，并去除文件的只读属性。

四、技术支持

如果在使用本光盘中遇到不懂的技术问题，你可以登录如下网站与作者联系：

http://www.crazyit.org

目 录 CONTENTS

CHAPTER

1

第 1 章
Swift 语言与开发环境

本章要点

- Swift 简介和对 Swift 的正确认识
- 下载并安装 Xcode 和 SDK
- 安装辅助工具和文档
- Swift 程序的入口
- 使用 Playground 测试 Swift 代码
- 使用 Xcode 工具开发 Swift 程序
- 使用 swiftc 命令编译 Swift 程序
- 掌握 swift 交互命令的用法
- 熟悉 Xcode 开发工具
- 掌握创建 iOS 项目的步骤
- 熟悉 Xcode 导航面板
- 熟悉 Xcode 检查器面板
- 熟悉 Xcode 库面板
- 使用 Xcode 的帮助系统

在正式开始学习 Swift 之前，必须拥有一台运行 OS X 系统的计算机（俗称电脑），通常建议购买一台 Apple 电脑即可，直接购买的 Apple 电脑都装有最新的 OS X 系统（建议选择运行 OS X 10.10 系统的电脑）。如果经济条件有限，也可使用普通电脑，甚至使用虚拟机来安装 OS X 系统，这样也可学习 Swift 开发。

Apple 公司提供了一个强大的 IDE 工具：Xcode，Xcode 不仅集成了各种 iOS 开发工具、iOS 模拟器，也允许初学者使用 Playground 来试验 Swift 的语法功能，而且 Xcode 还集成了 Swift 程序的编译器，因此在开始学习 Swift 之前，必须先安装 Xcode 工具。

本章并不打算介绍 Swift 的编程语法，本章只是讲解如何下载、安装 Xcode 工具，以及安装辅助工具和文档，并详细介绍使用 Xcode 内置文档的几种方式。

1.1 Swift 语言简介

Swift 是 Apple 公司新推出的一种编程语言，程序员可使用 Swift 开发 iOS、OS X 应用程序。Swift 一经推出，立即激发了广发开发者的极大热情。

1.1.1 Swift 语言

Swift 是 Apple 公司于 2014 年 6 月 2 日发布的全新的开发语言，Swift 吸收了众多现代编程语言的优点，Swift 尽力提供简洁的编程语言和强大的功能。

2014 年 6 月 2 日发布的 Swift 只是一个测试版，Apple 公司在接下来的几个月中不断地修改 Swift 的规范，尤其是 2014 年 7 月、2014 年 8 月，Swift 语言都被修改过大量内容。

本书将基于 OS X 10.10、Xcode 6.1 来介绍 Swift，Xcode 6.0 正式版才真正支持正式版的 Swift，Swift 语言的语法规范才开始稳定下来。从 Xcode 6.0 到 Xcode 6.1，Swift 也有了一些小的改进，这些都会在本书中得到体现。

作为一种 Apple 独立发布、支持的开发语言，从发布之初就得到广泛的关注，而且引起大量开发者的学习热情。从某种意义来看，Swift 代表了 Apple 公司新的商业布局，Apple 公司希望吸引更多的开发者加入 Apple 的开发阵营，Apple 传统的 Objective-C 语言那种略显古怪的语法确实会“吓退”不少开发者，而 Swift 的语法则显得更加自然，因此更容易被大家接受。

1.1.2 关于 Swift 的几个误解

虽然 Swift 编程语言面世的时间不长，但网络上已有大量关于 Swift 入门的介绍文章，大部分人（包括一些所谓的“大牛”）凭着对 Swift 的第一印象，贸然地给 Swift 下了一些结论，这些结论已经形成了一些关于 Swift 的误解，本书先对这些误解进行一些澄清。

（1）Swift 很简单

Swift 绝不是一门简单的编程语言，Swift 既支持面向过程的编程机制，也支持面向对象的编程方式，而且 Swift 的面向对象功能非常强大，如果需要全面掌握 Swift 的所有功能，就会发现 Swift 绝不简单。

网络上之所以流传着“Swift 很简单”的论调，无非是由于如下原因。

- Swift 的 HelloWorld 很简单，不管是 Apple 的发布会，还是 Swift 入门教程，第一个 Swift 示例程序总是 println("hello world")，这行代码即可代表一个程序。与 C、Objective-C 等程序需要从 main()函数开始不同的是，Swift 提供了隐式的程序入口，因此 Swift 的 HelloWorld 确实很简单，但这并不能代表 Swift 本身就很简单。

实际上，Swift 确实提供了非常简洁的语法，但在简洁的语言背后，Swift 提供了强大的功能：Objective-C 支持的（除了原生的指针运算）功能，Swift 几乎都可以支持；大量 Objective-C 不支持的功能，Swift 也可以支持。由此可见，Swift 是一门简洁、但并不简单的编程语言。

- Playground 提供了即时查看功能，网络上有关 Swift 的介绍文章中，绝大部分都是基于 Playground 介绍的，在 Playground 中可以实时看到程序中变量的值，以及程序的执行结果，这个工具提高了开发者的体验，于是给人形成一个错觉：Swift 很简单。

  实际上，Playground 只是一个快速练习 Swift、试验 Swift 语法功能、测试 API 调用的快速测试环境，这个测试环境既不支持人机交互，也不能用于开发真正的应用。可以这样说，完全基于 Playground 介绍 Swift 的学习资料都是不负责的。

  虽然 Playground 提供了非常友好的开发界面，但这与 Swift 本身是否简单并没有任何关系。

（2）Swift 是弱类型的语言

这也是一个广泛流传的误解，由于 Swift 采用了类似于 JavaScript 的定义变量的方式，例如如下代码：

```
var name = "fkit.org"
```

上面定义了一个 name 变量，但这行代码并没有显式定义变量 name 的类型，因此有些人误以为 Swift 是弱类型的语言。实际上，Swift 不仅是强类型的语言，而且提供了非常严格的类型检查。比如上面的 name 变量，Swift 将可以推断该变量类型是 String 类型，因此该变量只能接受字符串值，而且不能接受 nil（空值）。

（3）Swift 是脚本语言

这大概也是由于 Playground 形成的一个误解，大部分人都是通过 Playground 开始接触 Swift 的：开发者只要在 Playground 中编写 Swift 代码，在 Playground 的右边即可实时看到 Swift 的执行效果——这个效果和脚本语言的解释、执行效果非常相似，因此不少人误以为 Swift 是脚本语言。

但实际上 Swift 也是编译型语言，这一点与 Objective-C 完全相同，Xcode 提供了 swiftc 编译器来编译 Swift 程序，编译之后会得到一个可执行性的目标文件。

（4）Swift 很快就会取代 Objective-C

无论从 Apple 公司的官方宣传口径来看，还是从各种第三方框架、工具来看，Swift 都很难在短时间内取代 Objective-C。

而且 Swift 作为一门新生的编程语言，虽然提供了强大的功能，但由于 Swift 目前本身处于高速成长期，以 2014 年 7 月、2014 年 8 月的更新日志来看，Swift 已经进行了大量的更新，这种高速成长带来的不稳定势必影响企业开发者对 Swift 的真正使用。

由此可见，Swift 不可能在短期内取代 Objective-C。据保守估计，在 2~3 年之内，无论是 iOS 开发，还是 OS X 开发，Objective-C 将依然是主流。

1.2 搭建 Swift 开发环境

在搭建 Swift 开发环境之前，需要指出的是，Swift 开发环境必须搭建在 Apple 公司的 OS X 系统上。因此，无论读者采用何种方式，至少首先需要一台装有 OS X 操作系统的电脑。

为了得到装有 OS X 操作系统的电脑，有以下两种解决方式。

- 直接购买 Apple 公司的电脑，如 MacBook 笔记本电脑。这种电脑默认自带了 Mac OS X 操作系统。
- 如果读者没有 Apple 公司的电脑，则可以选择在普通电脑上安装所谓的“黑苹果”操作系统。关于安装系统的知识，此处就不再赘述了。

提示：

网络上有大量关于如何安装“黑苹果”系统的文章，实际上，安装 OS X 并不比安装 Windows 复杂，如果读者的经济条件有限，可以暂时选择“黑苹果”来体验 Swift 开发。对于想真正掌握 Swift 开发的用户来说，推荐选择使用苹果电脑。

需要指出的是，只有使用 10.9（代号小牛）以上版本的 OS X 系统才能安装 Xcode 6，这样才可支持 Swift。理想的开发平台是 OS X 10.10（代号优胜美地），该版本的 OS X 平台可安装 Xcode 6.1 正式版，也可直接支持运行 Swift 程序。

1.2.1 下载和安装 Xcode

Xcode 是 Apple 公司为开发 OS X 应用、iOS 应用提供的 IDE（集成开发环境，就像 Java 开发者习惯使用的 Eclipse、NetBeans 等）工具，这是一款非常优秀的 iOS 开发工具。除此之外，如果需要开发 iOS 应用，还需要下载和安装 iOS 的 SDK（Software Development Kit，软件开发工具集）。

提示：

早期的 Xcode 和 SDK 可能需要分开下载，但目前 Xcode 和 SDK 已经捆绑在一起了，因此只要下载一个文件即可。

下载并安装 Xcode 的步骤如下。

① 登录 https://developer.apple.com/xcode/downloads/站点，浏览器将会显示如图 1.1 所示的下载页面。

图 1.1　下载页面

在图 1.1 所示的页面中有两个链接可以下载 Xcode，第一个链接是通过 Mac App Store 安装 Xcode。这种方式是一种“傻瓜”式的安装（就像为 iPhone 手机安装 App 一样），但这种安装方式只能安装当前最新正式版的 Xcode，而且安装完成后没有离线安装文件。

因此，一般建议通过第二个链接来安装 Xcode。

② 单击图 1.1 所示页面中的“View downloads”链接，系统将会提示用户输入 Apple ID，如图 1.2 所示。

图 1.2 登录或注册 Apple ID

如果读者已有 Apple ID，则可通过该页面的登录按钮来登录页面；否则，用户需通过该页面的注册链接来注册 Apple ID，注册完成后再次登录该页面。

登录成功后即可进入真正的下载页面，如图 1.3 所示。

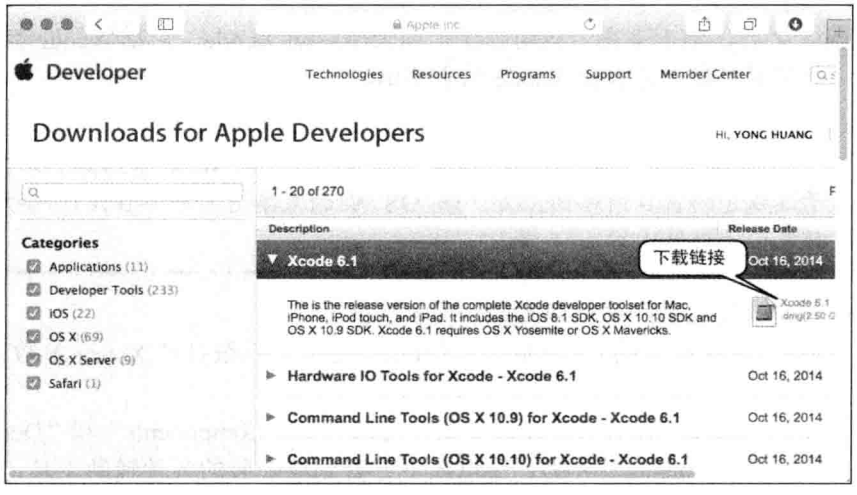


图 1.3 下载链接

③ 单击图 1.3 所示的下载链接，下载 Xcode 的最新版本：Xcode 6.1。该版本的 Xcode 已经包含了 Xcode 6.1 和 iOS 8.1 SDK，而且包含了 iPhone、iPad 模拟器。也就是说，只要下载和安装该文件，即可开发 iOS 应用。

提示：

安装 Xcode 6.1 需要 OS X 10.10 或 OS X 10.9 系统，因此读者在安装 OS X 操作系统时，应该安装 OS X 10.10 或 OS X 10.9 系统。

④ 下载完成后得到一个 xcode_6.1.dmg 文件，该文件就是 Xcode 的安装文件，双击该文件开始安装 Xcode 工具，此时将会看到如图 1.4 所示的窗口。

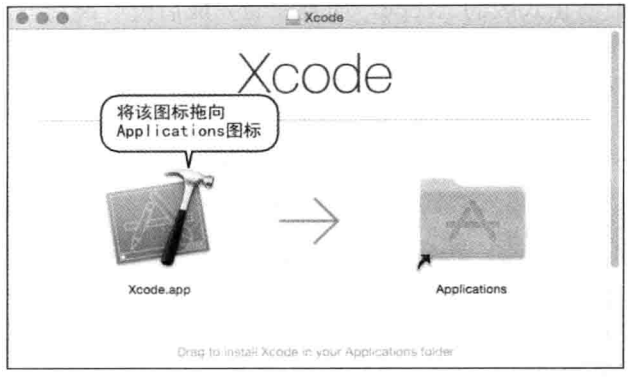


图 1.4 安装 Xcode 和 SDK

⑤ 将图 1.4 所示的 Xcode.app 图标拖向右边的 Applications 图标，这将把 Xcode 安装到 Applications 目录中，接下来就可以在“应用程序”列表中启动 Xcode。

提示：

安装 OS X 系统的应用程序与安装 Windows 程序不同，安装 OS X 应用程序通常只要把 xxx.app 程序包复制到 Applications 目录下即可。反过来，卸载 OS X 应用程序也比较简单，通常只需从 Applications 目录下删除 xxx.app 程序包即可。

1.2.2 安装辅助工具和文档

安装 Xcode 之后，接下来在“应用程序”列表中启动 Xcode，启动 Xcode 后并不会看到任何窗口，只会在屏幕最上方看到 Xcode 的主菜单。

提示：

OS X 管理菜单的方式与 Windows 存在较大差异：Windows 应用程序的菜单通常位于该程序的窗口内部的上方；而 OS X 的菜单通常并不在该应用程序的窗口内，而是位于屏幕上方。

为 Xcode 安装辅助工具和文档请按如下步骤进行。

① 选择屏幕上方菜单中的“Xcode”→“Preferences”，系统打开 Xcode 参数设置对话框，如图 1.5 所示。

② 单击“Downloads”标签页，在该页面中可以看到“Components”和“Documentation”两个分类。单击“Components”分类，可以在下方看到能安装的各种辅助工具，如果各工具右边显示 √ 图标，则表明该工具已经安装；如果显示下载按钮，则表明该工具还未安装，单击下载按钮就会安装该工具。

建议按图 1.5 所示的对话框安装 iOS 7.1 Simulator（iOS 7.1 模拟器），Xcode 6.1 默认自带 iOS 8.1 模拟器。

③ 单击“Documentation”分类，可以看到 Documentation 分类下的各种文档。

④ 在图 1.5 所示的对话框中列出了各种可安装的文档，同样可通过单击下载按钮来安装这些文档，直到所有文档右边的状态都显示 √ 图标（安装成功）。