SAMS Teach Yourself

- 全球销量逾百万册的系列图书
- 连续十余年打造的经典品牌
- 直观、循序渐进的学习教程
- 掌握关键知识的最佳起点
- “Read Less, Do More”（精读多练）的教学理念
- 以示例引导读者完成最常见的任务

每章内容针对初学者精心设计，1小时轻松阅读学习，
24小时彻底掌握关键知识

每章案例与练习题助你轻松完成常见任务，
通过实践提高应用技能，巩固所学知识

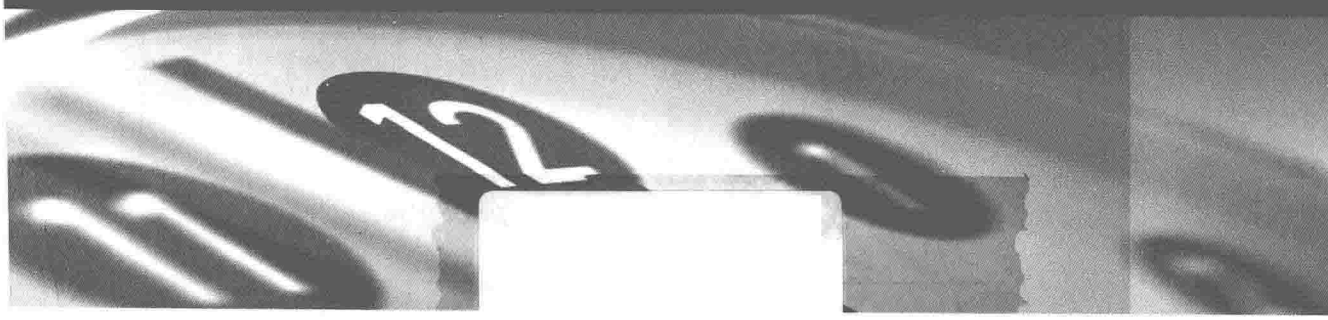
Swift 入门经典（第2版）

[美] BJ Miller 著
陈宗斌 译

中国工信出版集团
人民邮电出版社
POSTS & TELECOM PRESS

Swift 入门经典（第2版）

[美] BJ Miller 著
陈宗斌 译



人民邮电出版社
北京

图书在版编目（CIP）数据

Swift入门经典 / (美) BJ·米勒 (BJ Miller) 著 ; 陈宗斌译. — 北京 : 人民邮电出版社, 2017.2
ISBN 978-7-115-44439-4

Ⅰ. ①S… Ⅱ. ①B… ②陈… Ⅲ. ①程序语言－程序设计 Ⅳ. ①TP312

中国版本图书馆CIP数据核字(2017)第003963号

版权声明

◆ 著　[美] BJ Miller
译　陈宗斌
责任编辑　傅道坤
责任印制　焦志炜
◆ 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
固安县铭成印刷有限公司印刷
◆ 开本：787×1092　1/16
印张：22
字数：544 千字　2017 年 2 月第 1 版
印数：1－2 000 册　2017 年 2 月河北第 1 次印刷
著作权合同登记号　图字：01-2015-8791 号

定价：59.00 元

读者服务热线：(010)81055410　印装质量热线：(010)81055316
反盗版热线：(010)81055315

内容提要

本书基于 Apple 发布的 Swift 编程语言进行编写，循序渐进地介绍了使用 Swift 编写安全、强大的代码所需要的基本概念、架构和语法等知识。

本书分为 24 章，内容包括 Swift 开发环境简介，Swift 的基本数据类型，运算符，处理集合类型，使用条件语句控制程序流程，可选值，利用循环迭代代码，使用函数执行动作，了解高阶函数和闭包，结构体和类的类继承，枚举，自定义类，属性，添加高级类型功能，内存分配和引用的概念，处理可选链接，泛型简介，面向协议编程，错误处理，与 Objective-C 的交互性，以及 Swift 中的函数式编程。

本书内容深入浅出，通过简洁的语言和详细的步骤，帮助读者迅速掌握 Swift 开发所需要的知识。本书适合没有任何编程经验的新手阅读，也适合有志于从事 iOS 开发的人员阅读。

作者简介

BJ Miller 是 DXY Solutions 公司的一位 iOS 开发人员，DXY Solutions 是俄亥俄州克利夫兰地区的一家移动、Web 和设计咨询公司。BJ 拥有俄亥俄州伯里亚市鲍德温—华莱士学院（现在更名为鲍德温—华莱士大学）的计算机科学学士学位，这个城镇也是他长大的地方。他最近的职业涉及大规模企业网络管理、SQL 数据库管理，以及作为美国国防部的一名承包商负责 Microsoft SharePoint Server 和 Microsoft Project Server 的管理与集成，并且他所做的所有这些工作都具有 Microsoft 证书。在这之前，他曾经以 CCNA 的身份从事 LAN 工程，设计和实现网络基础结构。

BJ 在从事编程工作没多长时间后，就于 2009 年开始 iOS 开发，他对平台和 Objective-C 语言产生了深厚的兴趣。现在，他的爱好增加了 Swift，而他的兴趣依然很广泛。2013 年，他把自己的第一个应用发布到 iOS App Store，它的名称是 MyPrayerMap，是用于管理祈祷请求的简单工具。

当他没有利用 Objective-C 或 Swift 为工作或者本书编写程序时，他喜欢陪伴妻子和两个孩子，阅读、收听音乐或播客，以及玩 *The Legend of Zelda*（任何系统上的任何游戏他都感兴趣）。他还与 Daniel Steinberg 合作组建了 Cleveland CocoaHeads Meetup（http://www.meetup.com/Cleveland-CocoaHeads/），并且组建了该组织的一个分部，名称为 Paired Programming Fun，它是一个临时性的集会，致力于研究在配对编程风格中 Swift 中的测试驱动的开发（Test-Driven Development，TDD）。BJ 经常在 CocoaHeads 大会中介绍与 iOS 相关的主题，还在其他的大会（比如 MacTech、CocoaConf（俄亥俄州哥伦布市）和 CodeMash v2.0.1.5）上发表演讲。他还时不时地在 http://bjmiller.me 上写博客，并且在 Twitter（推特）上注册有账号@bjmillerltd。

献辞

谨以此书专门献给我的家人和朋友，你们在编写本书的整个过程中给了我极大的支持。感谢你们的爱和鼓励。

致谢

我想要感谢我的妻子和两个孩子，他们在我编写本书时一再容忍我在家庭生活中的疏忽。虽然本书主要的目的是进行更新，而不是编写全新的内容，但它仍然是一个令人疲惫不堪的过程。我很兴奋能够花更多的时间陪伴你们。

我还要感谢我的朋友、同事以及 Mac/iOS 社区其余的人，感谢他们的爱和鼓励。如果我没有经人层层介绍给 Daniel Steinberg，我就可能不会在 iOS 开发上走得更远，我也就可能不会编写本书。如果你有机会遇见那样一个人，那么你的生命将因此而变得富足。

此外，我不能不感谢那些帮助本书问世的好人：Trina MacDonald（组稿编辑）、Chris Zahn（高级开发编辑）、Olivia Basegio（编辑助理）、Valerie Shipbaugh（技术编辑）、Paula Lowell（文字编辑）和 Betsy Gratner（项目编辑）。

前　言

在 Apple 公司于 2014 年 6 月召开的年度全球开发者大会（World Wide Developer Conference，WWDC）上，Apple 公司公布了一种名为 Swift 的新编程语言，该公司从 2010 年起就在开发这种语言。这是一个重大的公告。多年来，在开发大多数 Mac 和 iOS 应用时，人们主要选择的语言是 Objective-C。可以明显感觉到人们对 Swift 编程语言的殷切期盼。Twitter 上关于 Swift 的言论不绝于耳，人们纷纷购买标题中具有 Swift 的域名，并且在公告后 24 小时内，Apple 公司 Swift iBook 的下载量超过 30 万次。人们为这种改变做好了准备。

但是，一种新语言不仅会带来语法上的区别，还会带来习惯的差异和新的约定。Swift 不仅是一种面向对象语言，它还引入了从其他语言收集到的特性，比如 C#、Haskell、Ruby 等。Swift 被标榜为“没有 C 的 Objective-C”，它在过去一年经历了如此大的改进，以至于有时很难看出它们的任何相似之处。Swift 构建于 Objective-C 中大家熟知的概念之上，但是它还包括了更现代、更安全的语法和多种范式（paradigm），比如，面向对象、函数式、强制性和块结构化，以及在 WWDC 2015 上把自身重新定义为一种面向协议的编程语言。

官方现在公布的 Swift 版本是 2.0，但它仍然在演进，甚至在编写本书时，还有更多的改变融入了 Beta 版。话虽如此，本书目前还是针对 Swift 2.0 和 Xcode 7。如果你在这些示例中发现了与书中描述的内容或者与界面不一致的地方，请检查 Apple 的发布文档和本书的电子版本，因为它们可能比你手上纸质的图书更新起来要快得多。此外，在 GitHub 资源库中可以找到本书中的所有代码示例（https://github.com/STYSwiftIn24H/ExamplesV2），并且它们将时刻保持最新。

Swift 已经被证明是一种非常优秀的语言，在其发布时，它与 iOS 7 及更高版本兼容。也可以为在 OS X Yosemite 及以后版本上运行的应用编写 Swift 程序。来自 Apple 的更新相当快，因此如果你所需要的某个功能不可用或者如果某个方面没有像预期的那样工作，可以考虑在 http://bugreport.apple.com 上把 bug 或特性请求进行归档。

本书读者对象

本书是为初中级程序员设计的，甚至那些还不熟悉 Swift 的高级程序员也可以从本书受

益。在阅读本书时，不必具有软件开发的背景（尽管这样可能有所帮助）。如果一点也不熟悉软件开发，那么先阅读一些更基础的书籍更好（尽管你也许能够很好地遵照本书内的示例进行操作）。

在本书中，我假定你具有学习 Swift 以及为 Mac 和/或 iOS 平台开发应用的激情。我还假定你愿意花费时间认真阅读本书，并且学习其中的概念。

本书主要内容

本书系统介绍了 Swift 编程语言，讨论了 Swift 的方方面面、最佳实践及其用途等。它不仅仅是一本语言参考书。到你读完本书时，应该牢固掌握了 Swift 中的许多概念，包括使它们发挥功效的语法。

你不应该期望仅仅通过阅读这一本书就能立即编写出可以获奖的 iOS 或 Mac 应用，因为本书并不打算作为一本用于学习关于应用创建的一切知识的一站式图书，这样一本图书的篇幅将会有几千页。相反，有更多的组件用于编写应用，特别是 Cocoa 和 Cocoa Touch 框架，它们都值得专门编写一本书（并且有许多这样的图书）。你应该通过认真的计划和开发来编写应用，并且根据你的应用将包括多少种不同的技术，你可能需要更多的资源。

在你自己尝试使用 Swift 编写应用之前，也不需要从头至尾阅读本书。在阅读本书的过程中，自始至终都可以利用你自己的应用灵活地验证所学的知识，或者当你在自己的应用中停滞不前并且需要一些指导时可以把本书用作参考资料。

还要记住，本书目前针对 Swift 2.0 和 Xcode 7，因此对于在本书最终编辑和付印后可能发生的改变请给予理解。随着 Swift 语言和 Xcode 环境发生改变，我将对代码示例进行更新。我在 GitHub 上提供了相关更新内容（https://github.com/STYSwiftIn24H/ExamplesV2）。

目 录

第 1 章

Swift 开发环境简介

在本章中你将学到：

- Swift 是什么以及它来自于哪里；
- 怎样从 Mac App Store 安装 Xcode 7；
- 怎样导航 Xcode 集成开发环境（Integrated Development Environment，IDE）；
- 怎样使用 playground（游乐场）；
- 怎样使用 Swift 的 REPL（Read-Eval-Print-Loop，读取—求值—输出—循环）；
- 怎样编写你的第一个 Swift 应用。

自从 2007 年推出 iPhone 以来，Apple 似乎不但点燃了基于消费者的电子产品行业的热情，而且使几乎所有的人能够有机会为他们的平台（即 Mac 或 iOS）编写应用。这对文化产生了显著的影响，现在你进入咖啡店或者任何企业，都能够看到大量的 MacBook Air、MacBook Pro、iPhone、iPad 以及现在的 Apple Watch。如果你正在阅读本书，有可能你想知道怎样才能编写一个应用，并且使之能够出现在你在那些咖啡店和企业里看到的几乎每一个人的屏幕上。

本书是关于 Swift 编程语言的，它是 Apple 在 2014 年全球开发者大会（World Wide Developer Conference，WWDC）上宣布的一种新的编程语言。在 Swift 推出之前，Mac 和 iOS 应用主要是用称为 Objective-C 的语言编写的，它是 C 编程语言一个严格的超集，这意味着可以用两种语言编写应用，并且有时不得不这样做。本书探讨了 Swift 编程语言，并且介绍了它的基础知识、结构和语法，这为你编写优秀的 Mac 和 iOS 应用打下了基础。

1.1 什么是 Swift

Swift 是由 Apple 自定义的一种编程语言，并且被视为“没有 C 的 Objective-C”。的确，这在一定程度上是正确的。Swift 不但借鉴了其他的语言（比如 Haskell、Ruby、Python、C# 及其他几种语言），而且在去年因为它自己的风格和方法而变得成熟起来。已证明 Swift 可以与现有的 Cocoa 和 Cocoa Touch 协同工作，它们包含现代 Mac 和 iOS 应用中使用的所有熟悉的类，用于支持它们的互操作性。

Swift 基于 3 根支柱：安全、强大和现代。Swift 提供了许多安全措施，比如，类型检查、用于保持不变性的常量、要求值在使用前初始化、内置的溢出处理以及自动内存管理。至于强大的功能，Swift 是使用高度优化的 LLVM 编译器生成的，包括许多低级的类似于 C 语言的函数，比如基本类型和流程控制，当然，在利用 Apple 的硬件生成 Swift 应用时谨记着最优的性能。Swift 还是现代的，这是由于它采纳了其他语言的许多特性，从而使该语言更简洁，却也更有表现力，比如闭包、泛型、元组、函数式编程模式等，后面的章节将会介绍它们。

1.2 起步

此时，最重要的假设是你已经具有一台 Mac 计算机，如果没有它，将不能安装 Xcode，它是 Apple 的 Mac 和 iOS 集成开发环境（Integrated Development Environment，IDE）。

By the Way

注意：下载 Xcode

Xcode 7 是从 Mac App Store 免费下载的，必须具有 Mac OS X 10.10.4 或更高版本。尽管可以在 Xcode 6.x 中编写 Swift 代码，但是本书还将包括 Swift 版本 2.0，它需要 Xcode 7。

在 Mac 上启动 App Store 应用，搜索 Xcode，然后单击安装软件。一旦安装完成，Xcode 就会列在/Applications 目录中。

1.2.1 四处看看

在打开 Xcode 时，可能会询问你是否想安装额外的工具；继续前进并安装它们。应该只有在第一次启动 Xcode 时才会发生这种情况。一旦打开了 Xcode，就会看到一个标准的菜单窗口，其中有一些选项，用于创建 playground、创建新项目或者打开现有的项目，右边是最近的项目和打开的 playground（如果打开了任何 playground）的列表。窗口现在应该如图 1.1 所示。

尽管在本书中主要在 playground 中工作，但是熟悉一下 IDE 是有好处的，因此就让我们快速完成这个任务。单击 Create a New Xcode Project，创建一个新的 Xcode 项目。下一个界面询问你想要创建的项目类型，对于这个试验，仅须使用 Single View Application，如图 1.2 所示，然后单击 Next 按钮。

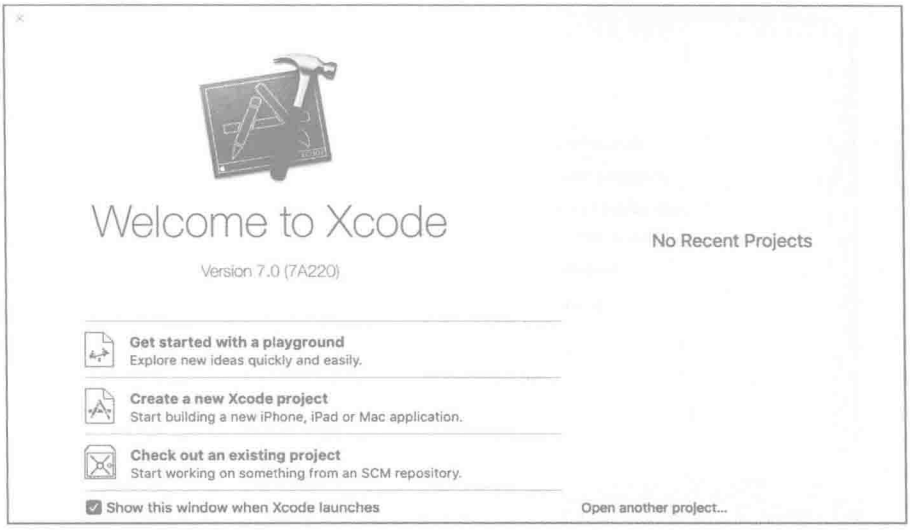


图 1.1

Welcome to Xcode 界面，在这里可以选择创建或编辑项目和 playground

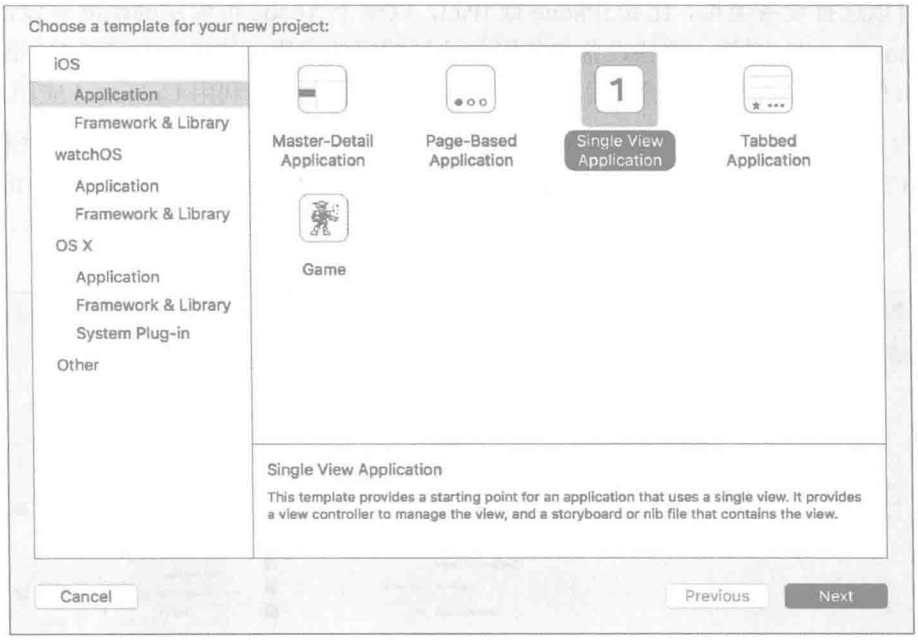


图 1.2

项目模板选择界面

接下来，将要求命名项目。选择 Organization Name、Identifier、Language（Swift 或 Objective-C）以及要运行项目的设备。这里还可以指示是否想要使用 Core Data，或者包括 Unit Tests 和 UI Tests，如图 1.3 所示。所有这些信息对于你将来创建的项目都是有用的，但是出于测试目的，我们还不需要关心它们，可以使这些复选框保持原样。Organization Identifier 通常是个人或公司 URL 的反向 DNS 名称，用于在组织级确保唯一性。Bundle Identifier 把 Project Name 附加到 Organization Identifier 末尾，用于确保每个应用包的唯一性。一旦把应用提交给 Mac App Store 或 iOS App Store，包标识符就需要是唯一的。