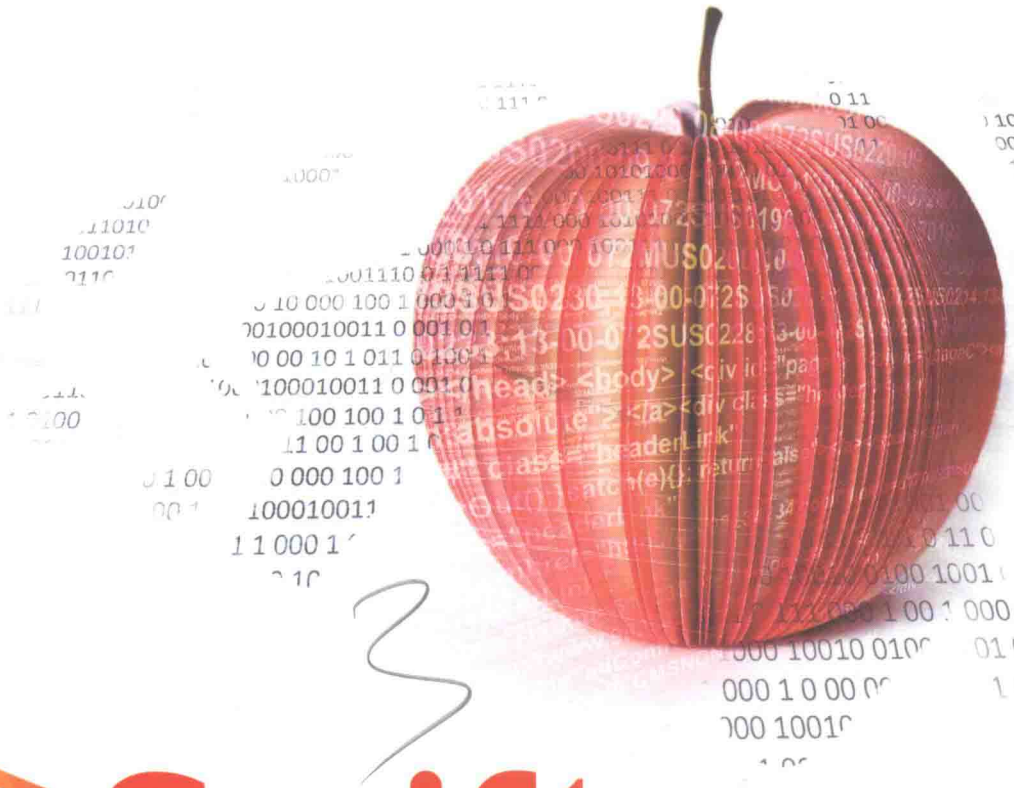
Pearson



Swift 编程之旅

Learning Swift 2 Programming

[美] Jacob Schatz 著 / 王芳 译

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Swift 编程之旅

[美] Jacob Schatz 著
王 芳 译

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

版权贸易合同登记号 图字：01-2015-7568

图书在版编目（CIP）数据

Swift 编程之旅/（美）雅各布·沙茨（Jacob Schatz）著；王芳译. —北京：电子工业出版社，2016.11
书名原文：Learning Swift 2 Programming

ISBN 978-7-121-29972-8
Ⅰ. ①S… Ⅱ. ①雅… ②王… Ⅲ. ①程序语言－程序设计 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2016）第 230627 号

策划编辑：张 迪（zhangdi@phei.com.cn）
责任编辑：张 迪
印 刷：北京京科印刷有限公司
装 订：北京京科印刷有限公司
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：787×1 092 1/16 印张：12 字数：308 千字
版 次：2016 年 11 月第 1 版
印 次：2016 年 11 月第 1 次印刷
印 数：3 500 册 定价：39.00 元

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）88254888，88258888。

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。

本书咨询联系方式：（010）88254469，Zhangdi@phei.com.cn。

译 者 序

非常荣幸能翻译这本书，成功开拓出了我职业生涯的另一个领域，让我学了十几年的英语和专业有了结合，英语没白学，卷子没白做，考试没白考。希望以后能继续发光发热。原来翻译这么辛苦，更不用说写书了，一定要细心阅读，才能不枉费作者的用心。由于 Swift 更新速度很快，加上翻译时间，总是很难赶上最新内容，翻译过程中还出现了一个小插曲，由于 Swift 从 1.0 升级到了 Swift 2.0，原来翻译的进度暂停，坐等美女编辑给我寄来第二版，但是语法的核心内容变化不大，影响较小，大家放心阅读。相信大家触类旁通，举一反三的能力。

现在只希望 Swift 3.0 出来慢点，哈哈哈哈哈哈（为了充字数多写几个）。

对于本书的结构，作者已经在简介中进行了概括，所以大家尽快开始阅读吧。相信你们一定能在轻松的过程中，站在作者为我们构想的思维逻辑角度去掌握可爱的 Swift。能阅读这本书的人，在书中作者也提到了，即使没有学过编程的都能看懂，当然如果曾经学过一门或者多门相关语言更好。作者用很浅显并且非常生动例子为我们诠释了 Swift 语言的特性，如果有 Objective - C 基础，那读起来就更加神速了。

和查阅 Swift 语言的苹果文档不同，在书中有大量的实例，结合 Swift 语法，能让你在无形中学到新知识。好像是空气和阳光，在不知不觉中汲取能量，得来全不费功夫。如果你懒得阅读 Swift 语法书籍，想直接上手写代码，直接在 playground 看到成效，让你不再陷入语法的泥沼，这本书是最好的选择。正如 Linux 之父说的，Talk is cheap，show me the code。作者直接给你 show code，并且已经在 YouTube 上教会了上万人如何使用 Swift。在 YouTube 上搜索 Skip Wilson，便能看到大量的 Swift 视频，配合本书更能达到事半功倍的效果。你可以关注 Skip Wilson（顺便提一下，推荐免费翻墙工具 Lantern）。有一个简短的小女孩的视频，估计是他女儿，上面提到了作者 work for Apple。让我们更想要去阅读了，果粉就这样，喜欢一切和苹果有关的。

由于计算机专业用语大家会有不同的翻译，难免有疏漏和差别，恳请读者们批评指正。译者联系方式：wangfanglyf@126.com。

另外，感谢老公，在翻译期间听我唠叨，还会监督我按时完成，偶尔还会对我的翻译挑挑毛病，进行校对，顺便成功学习了 Swift（奉旨感谢）。感谢父母，大恩不言谢，一切都在心中（必须感谢）。

译 者

2016 年 9 月

关于作者

Jacob Schatz 是一名有着超过八年开发经验并写了大量代码的高级软件开发工程师。他的代码被几百万人使用，并且他的建议经常就是大家所查找的。Jacob 还有个名字叫 Skip Wilson，他在 YouTube 上有一个很有名的涵盖 Swift 和 Python 的专题频道。Jacob 总是会选择了解最新的编程趋势。他有做出改变，以及持续解决问题的激情。最近，他深入学习了 Swift，但是他也写了大量的 JavaScript、Python、Objective - C 和其他语言的代码。他总是学习更多的语言并且十分享受创造新东西的过程。他是一个热心的教师，他很乐于教学并发现新的方式去解释复杂的概念。

题词　Tiffany 和 Noa

感谢

没有这么多人的帮助，我是无法完成这本书的。感谢以下人们：

Logan Wright，和我一起写了大量的 YouTube 上的练习并帮助我完成这本书；

Cody Romano，很热心地帮助我完成这本书并进行校对，用他无穷的知识帮助我调试出没有 bug 的程序；

Mike Keen，孜孜不倦地帮我校对所有章节，并确保我的例子都是正确的。同时也提供了无穷的灵感；

妈妈和爸爸，虽然他们不知道他们读的是什么，但是他们坐在那里把书读完并且给出了很明智的建议；

我的妻子，在我的电脑前忍受了无数个小时，并且在这本书完成的过程中变成了一个高级程序员。

期待收到你的反馈

作为本书的读者，你是我们最重要的批评家和评论家。我们很重视你的意见并且想要知道我们哪里做得很对，哪里可以做得更好，你想要看到我们出版哪些领域的书籍，以及任何你想要传达给我们的充满智慧的话语。

我们欢迎你的评论。你可以发邮件或者直接写信让我们知道你做了什么，或者不喜欢这本书的哪些地方，以及任何我们可以让这本书变得更好的东西。

请注意，我们不能帮助你解决和这本书技术相关的问题，因为我们收到了大量的这种邮件，可能没有办法针对每封都答复。

在你写信时，确保包括书名和作者，以及你自己的姓名、电话和邮箱地址。

邮箱：errata@ informit. com

地址：Addison – Wesley/Prentice Hall Publishing

ATTN：Reader Feedback 330 Hudson Street 7th Floor

New York，New York，10013

读者服务

在 informit. com/地址中使用 Learning Swift 2 Programming（ISBN 978 – 0 – 13 – 443159 – 8）去注册，方便你去下载、更新，以及更正的内容。

前　言

欢迎阅读 Swift2 编程的第二个版本。这本书将带领你进入使用全新的、激动人心的 Swift 语言的 iOS 编程世界。本书用一种快速但全面的方式，涵盖了 Swift 编程语言的开始到结束。

包括以下内容：

（1）谁应该阅读这本书。

（2）为什么你要阅读这本书。

（3）从这本书中你能收获什么。

（4）什么是 Swift，为什么它很棒。

（5）这本书是如何组织的。

（6）到哪里寻找示例代码。

准备好了吗？

谁应该阅读这本书

这本书适合那些已经熟练掌握一种或多种编程语言的人。你可以使用 Swift 作为你的第一门语言来学习这本书，但是如果你能将它和其他的语言关联起来，你会发现这将变得很容易。如果你有使用 Objective－C 进行 iOS 编程的经验，那么你应该能很快掌握 Swift。这本书会经常将 Swift 的概念和其他那些流行的编程语言关联起来，包括 JavaScript、Python、C 和 Objective－C。

为什么你要阅读这本书

这本书将教会你 Swift 编程的所有方面，这样你可以尽快开始编写高质量应用。然而，它并不是一个全面详细的参考文档，而是一个完整的、容易消化的、学习 Swift 语言的入门文档。Swift 是很多不同语言的混合，在这里你势必会学到很多新的概念，这本书将会使你成为更加优秀的开发者。Swift 语言本身很健壮，同时它允许你混入 Objective－C 语言。

如果你正在阅读这本书，你可能已经听人们谈论过 Swift 的惊人特性。你已经听说过它先进的设计，它运行得有多快，它会让你的开发变得有多容易。这本书将向你展示 Swift 语言的所有这些特点，同时也包括我在使用它的过程中一些令人激动的发现。你将成为革命中的一部分，同样地，你将成为世界上第一批 Swift 开发者中的一员。虽然 Swift 出现只有短短的几个月，但是可以预料到它会持续存在很长一段时间。现在正是加入它的最佳时刻。这本书将会让你完全沉浸在其中，并且提供了所有你入门和深入学习所需要的所有东西。

从这本书中你能收获什么

阅读这本书将会使你成为一个正式的 Swift 程序员，并且使你编写出真实的、高质量的

应用。你将利用 Swift 那些最先进的特性来编写应用，所以你的代码将会很精简、干净。在阅读完这本书以后，你可以使用 Swift 创建任何你想创建的应用。在你读本书时你将会学到：

（1）如何将现有的 Objective - C 代码结合到新的 Swift 应用中。

（2）如何使用类似泛型的高级特性，去编写更少的代码。

（3）如何用更快的方式创建可选项来确保你的代码不会由于不存在的值导致运行时崩溃。

（4）如何通过一些小的功能块编写闭包，这样可以写的和 4 个字符一样长。

（5）如何通过使用 SpritKit 创建一个二维横向卷轴（side - scrolling）游戏。

（6）如何通过使用 SceneKit 创建一个三维游戏。

（7）如何读取比特和字节，这样你就可以做一些类似阅读 PDF 文档的事情了。

什么是 Swift

Swift 是苹果公司的一门新编程语言，用来替代像 C 和 Objective - C 这样的语言，当然也能和它们一起使用。Swift 的产生主要是为了在 iOS 上使用一种耳目一新的、更简单的语言编写应用。Swift 语言和其他很多语言有关联。它是可定制的，这样你能用很多不同的方式编写 Swift。例如，Swift 允许你自定义方括号能做什么；而不是总是使用它们去访问数组和字典，从技术角度上说，你可以让它们做任何你想做的事。Swift 允许你自己定义操作符，并且能重写已经存在的。如果你想创建一个新的能增加两次而不是一次的 3 个加号（如 +++），那么你可以这样做。另外，你可以给你的自定义类创建自定义操作符，这意味着你将会写很少的代码，因此你的编程工作会简单一些。例如，如果你要写一个关于汽车的程序，你可以定义如果你让两个汽车实例相加会发生什么。正常情况下，你只能添加两个数字，但是在 Swift 中，你可以重写“ + ”操作符去做你想做的任何事情。

Swift 结构良好，并且完全兼容 Objective - C。所有在 Objective - C 中可使用的库也可以在 Swift 中使用。Swift 允许你创建桥梁去联系其他语言。

这本书是如何组织的

这本书被分为 12 章，涵盖了语言本身，并且带领你创建了一些小应用：

（1）第 1 ～ 4 章涵盖了基本的语言语法，包括变量、常量、数组、字典、函数、类、枚举和结构体。这些是 Swift 语言的基本构造部分。

（2）第 5 章使用 SpriteKit 实现一个小游戏。

（3）第 6 ～ 9 章包括了更多的高级语言特性，包括闭包、下标访问、更高级的操作符、协议和扩展、泛型，以及在比特和字节级别上的编程。

（4）第 10 ～ 12 章向你展示了如何通过你从前面章节中学到的知识创建一个现实世界中的应用。

享受这段旅程

我的目标是让这本书读起来很有趣，并且写这本书我也获得了很多乐趣。我想向你展示学习一门新的语言是一件多么令人激动的事情。

当一门新的语言出现时，通常并不是有很多关于它的知识产生。这本书的目的就是给你

展示那些很难找到的知识，并且它是那些很难读懂的知识的一个很容易理解的版本。

随着 Swfit 的不断发展在网上搜索答案会变得很困难，但是我们仍然要一起搞清楚 Swift。当然，Swift 语言中还存在 bugs，并且肯定还会持续出现 bugs。

我在写这本书的时候，Swift 还在测试中（并且还在持续更新），当完成这本书的时候 Swift 将变成 1.0 版本。随着时间的推移，Swift 将会持续改变，促使更多的人使用它，提出 bugs。这本书已经针对最新版本的 Swift（截至现在书写的时间）测试过了，但是这并不意味着 Swift 不会改变。我希望你能享受去学习使用 Swift。

反侵权盗版声明

目　录

第 1 章

通过实践练习：变量、常量、循环

Swift 是苹果公司创造的一种新的编程语言，其目的是使苹果产品的开发变得更加容易。如果你有 C 和 Objective - C 的经验，你应该会发现 Swift 非常容易。所有在 C 和 Objective - C 可用的类和类型都可以被移植，在其准确转换为 Swift 后都是可用的。

但是，如果你有 Ruby 或 Python 的背景，你会发现 Swift 的语法是你的拿手好戏。Swift 借鉴和迭代了 Python 和 Ruby 的许多想法。

如果你来自 JavaScript 的世界，你会很高兴地知道，Swift 不让你声明类型，但是旧版本严格的 java 需要。你也会很高兴地知道，Swift 拥有自己的 indexOf 版本和其他许多熟悉的 JavaScript 函数。如果它们不是所说的功能的精确副本，但它们至少是熟悉的。

如果你来自 java 的世界，你会很高兴地知道，虽然 Swift 不强迫你声明类型，但是你还是可以声明的，并且 Swift 肯定会非常严格地判断这些类型。

这些都是基本的语法比较，Swift 真正神奇的是具有变色龙一样的能力，用任何方式让程序员舒适。如果你想写一套简洁的一行式就可以一下子做你想做的任何事，那么 Swift 可以为你实现。如果你想写 Haskell 函数式编程，Swift 可以做到这一点。如果你想写漂亮的面向对象编程的经典设计模式，Swift 也可以这样做。

在未来（或现在，这取决于你何时读到这个），Swift 将开放源码，这样你就可以正式（理论上）在 Linux 或者 Windows 上写 Swift。有人甚至可能会使用 Swift 创建一个像在 Rails 中的 Ruby 一样的网络框架。

本章包括了 Swift 基础的创建块。以变量和常量开始，它们可以让你通过名字引用一个存储位置。通过这些知识，你将可以在内存中存储一些东西。Swift 有一个特性，叫作可选项，它允许你去判断一个值的存在（通过判断这个变量或常量是不是 nil）。Swift 可以进行强类型推断，这允许你（所声明的变量或常量）有一个严格的类型定义而不需要去声明一个类型。本章回顾了 Swift 如何处理循环和 if/else 语句。

1.1 建立 Swift 块

Swift 允许你通过将一个名字和某些类型的值联系在一起去使用变量和常量。例如，

如果你想在一个名为 greeting 变量中存储字符串“Hi”，你可以使用一个变量或常量。你通过 var 关键字创建变量，这就建立了一个在程序执行时可以改变的关联值。如果你不想让这个值改变，你可以使用常量。例如，你可能会记录一个用户在被拒绝访问这个网站前能够尝试登录的数目。在这种情况下，你可能会想要建立一个常量，如下面例子中所展示的：

```
var hiThere = "Hi there"
hiThere = "Hi there again"

let permanentGreeting = "Hello fine sir"
permanentGreeting = "Good morning sir"
```

注意，你没有像在其他语言中一样使用分号。你不需要使用分号来表示一个声明的结束，除非你想要将很多声明结合在一起。Swift 和 JavaScript 有很大的不同，在 JavaScrip 中，省略分号通常会被认为是不安全的。下面的例子向你展示了在 Swift 中当你需要使用分号将多行语句组合成一行的场景：

```
let numberOfRetries = 5; var currentRetries = 0
```

在 Swift 中，还有一个很独特的地方，你可以使用几乎所有的 Unicode 字符去命名你的变量或常量。开发者们可以使用希伯来语，简体中文，甚至一些特殊的 Unicode 字符，如全色的树袋熊表情（苹果的独特发明）来给资源命名。

```
var yes = 0, no = 0
```

1.1.1 计算型属性（Getter 和 Setter）

在 Swift 中你也可以声明变量作为计算属性。当你想在运行时计算变量的值时，你会使用这个特性。这里有一个 getter 的例子，评分值由剩下多少时间确定。在这个例子中，我们创建了一个只读的计算属性。

```
var timeLeft = 30
var score:Int {
get{
    return timeLeft * 25
}
}
print(score)
```

在这个例子中，我们可以在任何地方引用（或读取）成绩，因为它是在全局范围内的。真正有趣的是，如果我们试着设置分数，它会给我们一个错误，因为我们创建的是一个只读属性。如果我们要设置这个属性，我们需要创建一个 setter。你不能创建一个 setter 而没有 getter。撇开不说这样做是没有意义的事实，重要的是它也不会工作了。让我们创建一个 setter 的同时也创建一个 getter，直接给 setter 设置为可计算属性是没有意义的，因为这个属性的值是在运行期计算的。因此，当你想设置其他值作为 setter 的值时，便使用 setter。同时，在某些组织结构中 setter 也能很好地工作，目前我们还没有讨论过，但它值得简要地说一下。这里是一个完整的例子，其中包括我们还没有涉及的许多元素。