iOS编程基础：Swift、Xcode和Cocoa入门指南

作者：马特·诺伊贝格（Matt Neuburg）

O’Reilly精品图书系列

iOS编程基础：Swift、Xcode和Cocoa入门指南

iOS 9 Programming Fundamentals with Swift

（美）马特·诺伊贝格（Matt Neuburg） 著

张龙 译

ISBN：978-7-111-55635-0

本书纸版由机械工业出版社于2017年出版，电子版由华章分社（北京华章图文信息有限公司，北京奥维博世图书发行有限公司）全球范围内制作与发行。

客服热线：+ 86-10-68995265

客服信箱：service@bbbvip.com

官方网址：www.hzmedia.com.cn

新浪微博 @华章数媒

微信公众号 华章电子书（微信号：hzebook）

目录

O’Reilly Media，Inc.介绍

O’Reilly Media通过图书、杂志、在线服务、调查研究和会议等方式传播创新知识。自1978年开始，O’Reilly一直都是前沿发展的见证者和推动者。超级极客们正在开创着未来，而我们关注真正重要的技术趋势——通过放大那些“细微的信号”来刺激社会对新科技的应用。作为技术社区中活跃的参与者，O’Reilly的发展充满了对创新的倡导、创造和发扬光大。

O’Reilly为软件开发人员带来革命性的“动物书”；创建第一个商业网站（GNN）；组织了影响深远的开放源代码峰会，以至于开源软件运动以此命名；创立了Make杂志，从而成为DIY革命的主要先锋；公司一如既往地通过多种形式缔结信息与人的纽带。O’Reilly的会议和峰会集聚了众多超级极客和高瞻远瞩的商业领袖，共同描绘出开创新产业的革命性思想。作为技术人士获取信息的选择，O’Reilly现在还将先锋专家的知识传递给普通的计算机用户。无论是通过书籍出版，在线服务或者面授课程，每一项O’Reilly的产品都反映了公司不可动摇的理念——信息是激发创新的力量。

业界评论

“O’Reilly Radar博客有口皆碑。”

——Wired

“O’Reilly凭借一系列（真希望当初我也想到了）非凡想法建立了数百万美元的业务。”

——Business 2.0

“O’Reilly Conference是聚集关键思想领袖的绝对典范。”

——CRN

“一本O’Reilly的书就代表一个有用、有前途、需要学习的主题。”

——Irish Times

“Tim是位特立独行的商人，他不光放眼于最长远、最广阔的视野并且切实地按照Yogi Berra的建议去做了：‘如果你在路上遇到岔路口，走小路（岔路）。’回顾过去Tim似乎每一次都选择了小路，而且有几次都是一闪即逝的机会，尽管大路也不错。”

——Linux Journal

译者序

在2014年的WWDC大会上，苹果公司正式发布了Swift这门全新的编程语言。作为iOS与OS X平台上的老牌编程语言Objective-C的有益补充和替代者，Swift从发布伊始就激发了广大开发者的强烈兴趣。学习和尝试Swift编程语言的开发人员越来越多，这也促使Swift这门新语言在TIOBE编程语言排行榜上的排名一路攀升，成为一颗耀眼的编程语言新星，同时也是有史以来增长速度最快的语言。虽然Swift的初始版本存在着不少问题，但苹果公司仍在不遗余力地持续推动着这门语言的发展。作为iOS与OS X的开发者，我们欣喜地看到Swift语言不断增强的功能、不断增加的特性以及不断优化的性能。这些都是Swift能够迅速得到广大开发者青睐的重要因素。

值得一提的是，一年后苹果公司在WWDC 2015上正式宣布将Swift开源，并于同年年底发布了全新的网站https://swift.org。目前Swift开源代码托管在GitHub上，任何感兴趣的开发者都可以下载学习。Swift如此之快的发展速度一方面得益于苹果公司各项产品的推出，另一方面也是由于广大开发者的热烈追捧。作为一门年轻的编程语言，能在短短两年时间内就获得如此成功，这也是我们广大iOS开发者的一个福音。技术发展日新月异，只有跟上技术发展的步伐我们才能在未来立于不败之地。目前，国内外已经有不少公司将自己的iOS应用部分或全部由Objective-C迁移至Swift，很多新项目也已经开始使用Objective-C进行开发了。这都进一步证实了Swift未来巨大的发展潜力。

本书可谓是Swift编程语言的一部百科全书。在学习本书之前不需要读者具备任何Swift背景知识（当然，适当了解Objective-C将会有助于学习，但也并非必需），读者只需要打开本书，从第1章开始逐章阅读即可。全书采用了由浅入深、循序渐进的方式对Swift语言进行讲解，同时辅以大量可运行的代码示例帮助读者加深对理论知识的理解。毕竟，无论学习何种知识与技术，基础永远是最为重要的；坚实的基础将会帮助你更好地掌握技术，并且也会对后续的学习产生积极的作用。

全书共分13章，每一章都单独讲解一个主题，目的在于帮助读者集中精力掌握好Swift每一个重要且关键的知识点。从Swift架构概览开始，接着介绍了函数、变量、对象类型与流程控制，这些都是Swift重要的基础知识；然后又介绍了Xcode项目的管理、nib、文档以及项目的生命周期；全书最后对Cocoa类、Cocoa事件、内存管理与对象间通信等高级主题展开了详尽的介绍。此外，附录A对C、Objective-C与Swift之间的关系和调用方式进行了详尽的论述。学习完本书后，读者将会掌握Swift重要且关键的特性与知识点，完全可以着手通过Swift开发全新的iOS应用。

Swift编程语言涉及的知识点与特性非常多，没有任何一本书能够穷尽Swift的每一项特性，本书也不例外。本书可以作为读者学习Swift编程语言的入门指引，学习完本书后可以通过苹果公司的Swift编程语言官方文档等在线资源进一步加深对该门语言的理解和认识，并通过实际动手来掌握Swift的每一项特性。可以这么说，通过阅读本书，读者将会具备Swift开发的一般知识与技能，辅以一定的实践操作，相信经过一段时间的锤炼，你就可以真正精通这门优秀的编程语言。

技术图书的翻译是一项异常艰苦的劳动，这里我要将深深的感激之情送给我的家人，感谢你们在生活中对我无微不至的关怀，使我能够专心于翻译工作；此外，我要将这本书送给我亲爱的孩子张梓轩小朋友，每当爸爸感到疲惫时，看到你就会立刻获得无尽的动力，你永远是爸爸的开心果，如果你未来有志成为一名程序员，爸爸愿意祝你一臂之力；最后，非常感谢机械工业出版社华章公司的缪杰老师，感谢你对我持续的帮助，每一次与你沟通都非常顺畅，虽未曾谋面，但已然是老友。

虽然译者已经在本书的翻译工作上倾注了大量的心力，不过囿于技术与英文水平，书中难免出现一些瑕疵。如果在阅读过程中发现了问题，请不吝赐教并发邮件至zhanglong217@163.com，我会逐一检查每一项纰漏，以期重印时修订。

张龙

2016年于北京

作者介绍

Matt Neuburg从1968年就开始学习计算机编程了，那时他才14岁，是一家地下高中俱乐部的成员，成员们每周见一次面，在银行的PDP-10s上进行分时操作，方式则是使用原始的电传打字机。他还偶尔使用过普林斯顿大学的IBM-360/67，不过有一天弄丢了穿孔卡片，这令他感到非常沮丧，最后放弃了。他在斯沃斯莫尔学院主修希腊语，并于1981年获得康奈尔大学的博士学位，他在一台大型机上完成了博士论文（关于埃斯库罗斯）的编写工作。接下来，他在多家知名的高等院校教授古典语言、文学与文化课程，不过现在有很多高校都否认他所讲授的知识。他发表过大量学术文章，但这些文章对人毫无吸引力。与此同时，他收到了一台Apple IIc，在绝望中又开始从事计算机工作，后来在1990年迁移到了一台Macintosh上。他编写过一些教育与实用的自由软件，并且成为在线杂志TidBITS早期的一位定期撰稿人，他在1995年离开了学术界并开始编辑MacTech杂志。在1996年8月，他成为了一名自由职业者，这意味着从那时起他一直在寻找工作。他是《Frontier：The Definitive Guide》《REALbasic：The Definitive Guide》及《AppleScript：The Definitive Guide》的作者，同时也是《Programming iOS 7》（由O'Reilly Media出版）及《Take Control of Using Mountain Lion》（由TidBITS Publishing出版）的作者。

封面介绍

本书封面的动物是一只格陵兰海豹（Pagophilus groenlandicus），这是个拉丁语名字，表示“来自格陵兰的冰块爱好者”。这些动物生长在北大西洋与北冰洋，大部分时间都待在水中，只有在生产和脱毛的时候才浮出冰面。由于耳朵是密闭的，其流线型的身体与节省体力的游泳方式使得他们非常适合于水栖生活。虽然海狮之类的有耳物种是游泳好手，但他们却是半水栖的，因为它们在陆地上交配和休息。

格陵兰海豹拥有银灰色的皮毛，后背上有一个巨大的黑色标记，像是一个竖琴或是叉骨。成年格陵兰海豹会长至5到6英尺长，300到400磅重。由于栖息地寒冷，它们有着一层厚厚的鲸脂来隔绝外界。格陵兰海豹的饮食变化多样，包括几种鱼类与甲壳类动物。他们可以在水下潜伏16分钟左右来寻觅食物，并且可以潜入几百英尺的水下。

刚出生的格陵兰海豹并没有任何防护装备，它们通过白色的皮毛来保暖的，皮毛会吸收阳光的热量。12天后，小格陵兰海豹就会被父母丢弃，并且因为吃母乳，其体重会长到刚出生时的3倍。随后的几周一直到小格陵兰海豹可以在冰上游走为止，它们都非常容易遭受来自食肉动物的攻击，并且体重会减轻一半。幸存下来的格陵兰海豹会在4到8年后成熟（取决于性别），其平均寿命大约为35岁。

格陵兰海豹会在加拿大、挪威、俄罗斯与格陵兰的海岸遭到捕杀，其肉、油与皮毛会被拿来交易。虽然一些政府出台了条例并强制捕杀配额，但每年被捕杀的格陵兰海豹数量都会被少报。不过，自然资源保护论者的疾呼与努力使得市场上对于海豹皮毛与其他商品的需求量降低了。

本书封面图片来自于Wood的Animate Creation。

前言

2014年6月2日，苹果公司在WWDC大会最后宣布了一项令人震惊的公告：“我们开发了一门全新的编程语言。”开发者社区对此感到非常惊讶，因为他们已经习惯了Objective-C，因此开始怀疑苹果公司是否有能力将既有资产迁移过来。不过，这一次开发者社区错了。

Swift发布后，众多开发者立刻开始检视这门新语言：学习并批判它，决定是否该使用它。我的第一步就是将自己所有的iOS应用都转换为Swift；这足以说服我自己，虽然Swift有各种各样的缺点，但它值得每一个iOS编程新兵去掌握；自此以后，我的书都会假设读者使用的是Swift。

Swift语言从一开始的设计上就具备如下主要特性：

面向对象

Swift是一门现代化的、面向对象的语言。它是完全面向对象的：“一切皆对象。”

清晰

Swift易于阅读和编写，其语法糖很少，隐藏的捷径也不多。其语法清晰、一致且明确。

安全

Swift使用强类型，从而确保它知道（并且你也知道）在每一时刻每个对象引用都是什么类型的。

小巧

Swift是一门小巧的语言，提供了一些基本的类型与功能，除此之外别无其他。其他功能需要由你的代码，或你所使用的代码库（如Cocoa）来提供。

内存管理

Swift会自动管理内存。你很少需要考虑内存管理问题。

兼容于Cocoa

Cocoa API是由C和Objective-C编写的。Swift在设计时就明确保证可与大多数Cocoa API交互。

这些特性使得Swift成为学习iOS编程的一门优秀语言。

其他选择Objective-C依然存在，如果你喜欢还可以使用它。实际上，编写一个同时包含Swift代码与Objective-C代码的应用是很容易的；有时也需要这么做。不过，Objective-C缺少Swift的一些优势。Objective-C在C之上增加了面向对象特性。因此，它只是部分面向对象的；它同时拥有对象与标量数据类型，其对象需要对应于一种特殊的C数据类型（指针）。其语法掌握起来很困难；阅读与编写嵌套的方法调用会让人眼花，它还引入了一些黑科技，如隐式的nil测试。其类型检查可以而且经常关闭，这会导致程序员犯错，将消息发送给错误的对象类型并导致程序崩溃。Objective-C使用了手工的内存管理；新引入的ARC（自动引用计数）减轻了程序员的一些负担，并且极大地降低了程序员犯错的可能性，不过错误依旧有可能发生，内存管理最终还是要靠手工来完成。

最近向Objective-C增加或修订的特性（ARC、合成与自动合成、改进的字面值数组与字典的语法、块等）让Objective-C变得更加简单和便捷，不过这些修复也使语言变得更加庞大，甚至会引起困惑。由于Objective-C必须要包含C，因此其可扩展和修订的程度会受到限制。另一方面，Swift则是个全新的开始。如果你梦想完全修订Objective-C，从而创建一个更棒的Objective-C，那么Swift可能就是你所需要的。它将一个先进、合理的前端置于你与Cocoa Objective-C API之间。

因此，Swift就是本书通篇所使用的编程语言。不过，读者还需要对Objective-C（包括C）有所了解。Foundation与Cocoa API（这些内建的命令是你的代码一定会用到的，从而让iOS设备上的一切可以实现）依旧使用C与Objective-C编写。为了与它们进行交互，你需要知道这些语言需要什么。比如，为了在需要NSArray时可以传递一个Swift数组，你需要知道到底是什么对象可以作为Objective-C NSArray的元素。

因此，本书虽然不会讲解Objective-C，但我会对其进行足够充分的介绍，从而使你在文档和互联网上遇到这类问题时能够知道解决方案，我还会时不时地展示一些Objective-C代码。本书第三部分关于Cocoa的介绍会帮助大家以Objective-C的方式来思考——因为Cocoa API的结构与行为基本上是基于Objective-C的。本书最后的附录会详细介绍Swift与Objective-C之间的交互方式，同时还会介绍如何以Swift和Objective-C混合编程来编写应用。

本书范围

本书实际上是我的另一本书《Programming iOS 9》的配套参考书，该书以本书的结束作为起点。它们之间是互补的。我相信，这两本书的结构合理、内容通俗易懂。它们提供了开始编写iOS应用所需的完整基础知识；这样，在开始编写iOS应用时，你会对将要做的事情以及方向有着深刻的理解。如果编写iOS程序类似于用砖盖房子，那么本书将会介绍什么是砖以及如何使用它，而《Programming iOS 9》则会给你一些实际的砖并告诉你如何将其堆砌起来。

阅读完本书后，你将知道Swift、Xcode以及Cocoa框架的基础，接下来就可以直接开始阅读《Programming iOS 9》了。相反，《Programming iOS 9》假设你已经掌握了本书所介绍的内容；一开始它就会介绍视图与视图控制器，同时假设你已经掌握了语言本身和Xcode IDE。如果开始阅读《Programming iOS 9》并且想知道书中一些没有讲解过的东西，如Swift语言基础、UIApplicationMain函数、nib加载机制、Cocoa的委托与通知模式、保持循环等，那就不要尝试在该书中寻找答案了，我并没有在那本书中介绍这些内容，因为这里都介绍过了。

本书的3部分内容将会介绍iOS编程的基础知识：

·第一部分从头开始介绍Swift语言。我没有假设你知道任何其他的编程语言。我讲解Swift的方式与其他人不同，如苹果公司的方式；我会采取系统的方式，逐步推进，不断深入。同时，我会讲解最本质的内容。Swift并不是一门庞大的语言，不过却有一些微妙之处。你无须深入到全部内容当中，我也不会面面俱到地讲解。你可能永远都不会遇到一些问题，即便遇到了，那也说明你已经进入到了高级Swift的世界当中，而这已经超出了本书的讨论范围。比如，读者可能会惊奇地发现我在书中从来都没有提到过Swift playground和REPL。本书的关注点在于实际的iOS编程，因此我对Swift的介绍将会关注在这些常见、实际的方面上；以我的经验来看，这些内容才是iOS编程当中用得最多的。

·第二部分将会介绍Xcode，这是我们进行iOS编程的地方。我将介绍何为Xcode项目，如何将其转换为应用，如何通过Xcode来查阅文档，如何编写、导航与调试代码，以及如何在设备上运行应用并提交到App Store等过程。该部分还有重要的一章用来介绍nib与nib编辑器（Interface Builder），包括插座变量与动作，以及nib加载机制；不过，诸如nib中的自动布局限制等专门的主题则不在本书的讨论范围之中。

·第三部分将会介绍Cocoa Touch框架。在进行iOS编程时，你会使用到苹果公司提供的大量框架。这些框架共同构成了Cocoa；为iOS编程提供API的Cocoa叫作Cocoa Touch。你的代码最终将是关于如何与Cocoa进行通信的。Cocoa Touch框架提供了iOS应用所需的底层功能。不过要想使用框架，你需要按照框架的想法去做，将代码放到框架期望的位置处，实现框架要求你实现的功能。有趣的是，Cocoa使用的是Objective-C，你使用的是Swift：你需要知道Swift代码如何与Cocoa的特性与行为进行交互。Cocoa提供了重要的基础类，并添加了一些语言与架构上的设施，如类别、协议、委托、通知，以及关于内存管理的基本功能。该部分还会介绍键值编码与键值观测。

本书读者将会掌握任何优秀的iOS开发者所需的基础知识与技术；但本书并不会介绍如何编写一个有趣