iPhone 和 iPad 编程入门 零基础学习编写 App

 (美) Wendy L. Wise
 著

 张久修 陈乐辉 赵 勉 译

清华大学出版社

EISBN: 978-1617292651

Anyone Can Create an App: Beginning iPhone and iPad Programming

Wendy L. Wise

Original English language edition published by Manning Publications, 178 South Hill Drive, Westampton, NJ 08060 USA. Copyright © 2017 by Manning Publications. Simplified Chinese-language edition copyright © 2018 by Tsinghua University Press. All rights reserved.

本书中文简体字版由 Manning Publications CO.授权清华大学出版社出版。未经出版者书面许可,不得以任何方式复制或抄袭本书内容。

北京市版权局著作权合同登记号 图字: 01-2017-4217

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话: 010-62782989 13701121933

图书在版编目(CIP)数据

iPhone 和 iPad 编程入门 零基础学习编写 App/(美) W. L. 怀斯(Wendy L. Wise) 著; 张久 修,陈乐辉,赵勉 译. 一北京: 清华大学出版社,2018

(移动开发经典丛书)

书名原文: Anyone Can Create an App: Beginning iPhone and iPad Programming ISBN 978-7-302-48875-0

I. ①i··· II. ①W··· ②张··· ③陈··· ④赵 III. ①移动终端-应用程序-程序设计 IV. ①TN929.53

中国版本图书馆 CIP 数据核字(2017)第 287728 号

责任编辑: 王 军 韩宏志

装帧设计: 孔祥峰

责任校对: 牛艳敏

责任印制:沈露

出版发行: 清华大学出版社

网 址: http://www.tup.com.cn, http://www.wqbook.com

地 址:北京清华大学学研大厦 A 座 邮 编: 100084

社 总 机: 010-62770175 邮 购: 010-62786544

投稿与读者服务: 010-62776969, c-service@tup.tsinghua.edu.cn 质 量 反 馈: 010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者: 三河市金元印装有限公司

经 销:全国新华书店

开 本: 170mm×240mm 印 张: 20.75 字 数: 407 千字

版 次: 2018年1月第1版 印 次: 2018年1月第1次印刷

印 数: 1∼3500

定 价: 68.00 元

产品编号: 075685-01

译者序

iOS 系统诞生于 2007 年 1 月初,由苹果(Apple)公司发布。在 iOS 系统诞生的同时,颠覆手机行业的 iPhone 也横空出世,这预示着智能手机时代从此到来。多年来,搭载 iOS 操作系统的 iPhone 因其流畅的操作体验以及精湛的制作工艺,一直备受用户的好评与青睐。每年的苹果新品发布会以及苹果开发者大会也都备受关注。

2017年是苹果推出 iPhone 后的第十年,因此苹果公司于北京时间 9 月 13 日凌晨 1 点的新品发布会上发布了第一款 iPhone 全屏幕手机,并将其命名为 iPhone X,以纪念 iPhone 诞生十周年(在罗马数字中 X 代表 10)。这意味着 iPhone 全屏幕手机时代的到来。一周后,于北京时间 9 月 20 日凌晨 1 点,苹果正式发布了最新一代的操作系统 iOS 11,而作为 iOS 应用的集成开发工具(IDE)的最新版 Xcode 9 的更新也同时推出,这也意味着拥有增强现实(AR)功能和机器学习功能的应用,将如雨后春笋般陆续出现在 App Store 中。而作为 iOS 开发者或即将成为 iOS 开发者的你,面对如今的智能手机时代,以及正在到来的人工智能时代,怎能不怦然心动、跃跃欲试?

iOS 开发的主流编程语言是 Objective-C 和 Swift,本书所使用的编程语言是 Swift,且是最新的 4.0 版本。Swift 是苹果于 2014 年在苹果开发者大会上发布的新开发语言,它易学易用,是第一套具有与脚本语言同等表现力和趣味性的系统编程语言,可与 Objective-C 共同运行于 Mac OS 和 iOS 平台,用于搭建基于苹果平台的应用程序。Swift 在发布后短短几年间便获得了高速发展,且于一年后便开源,在不久的将来该语言也将在服务器端大行其道。因此 Swift 编程语言是一名合格的 iOS 开发者必须掌握的。

本书是一本优秀的 iOS 开发入门书籍,作者本着"人人都可以学习编程"的宗旨,用最朴素幽默的文笔、浅显易懂的语言和循序渐进的学习方式,引领初学者一步步掌握 iOS 编程概念和基础技能。不仅如此,作者还向读者时刻传达着一

种学习编程的理念:相信自己并保持耐心。确实,不止在学习编程的过程中应当如此,在实际项目开发中,保持自信和耐心尤为重要。因为在这个过程中你将遇到各种困难、Bug、甚至技术瓶颈,唯有拥有自信和耐心者,才能时刻保持冷静的头脑去分析并制定相应的策略,最终冲破一道道关卡从而获得宝贵的开发经验并不断成长。当你决定拿起本书时,所要做的便是坚持再坚持,以及一次又一次的实践,在这个过程中,你要不断鼓励自己,相信你会在本书中获得一次又一次的成就感。

本书的作者有着多年的 iOS 平台开发及项目管理经验,即使是最基础的编程概念都有着自己的真知灼见,且讲解独到,想必读者们定会受益匪浅!而本书的译者们同样有着多年且丰富的 iOS 项目开发经验,来自国内知名的互联网公司。希望这本中文版的 iOS 开发入门书籍能够真正为你打开一扇通往编程世界的大门。

这里要感谢清华大学出版社的编辑,尤其是李阳编辑,在翻译过程中通过远程沟通来耐心地指导我们翻译上的问题。没有你们的帮助和指导,本书中文版也不可能这么快与读者见面。

作为本书的译者,我们本着"诚惶诚恐"的态度翻译这本入门书籍,为避免 误导初学者,所以文中的一字一词都会反复斟酌。但鉴于译者水平有限,错误和 失误在所难免,如有任何意见和建议,请不吝指正。本书全部章节由张久修、陈 乐辉、赵勉翻译,参与本书翻译工作的还有张淼、聂家发、陈昕、刘超,在此一 并表示感谢。 终于完成了这本书,兴奋之情难以言表。我期望看到更多的人学习编程,但 有些人面对庞大的编程世界心生畏惧,难以寻得一个好的学习起点。我希望本书 能够对他们有所帮助。

在大学里我并没有攻读计算机科学专业,所以我在开启计算机生涯时的处境与你现在的处境或许很相近。在 2007 年的第一款 iPhone 面世时,我也拥有了属于自己的第一部 iPhone,从此我便深深地迷恋上了它。我希望能够创建自己的应用,所以我找到了一个讲授 iOS 编程的本地培训机构,于是我便去聆听了。从那时起,我就一直在 iPhone 上编程。

自你开始编程后,你会注意到的一件事是,你的朋友会经常提到他们对应用有一个很棒的主意。然后他们会询问你是否可以为他们创建这样的应用,并且他们将许诺与你一起分享他们认为肯定会获得的巨额利润。我收到了很多这样的请求,而我的一般回应是:他们应该学会自己创建应用。他们通常也会跟进一两句评论,如不知道从何处入手、不够聪明、他们的大脑不适合编程,或者编程太难。我不断否认这些东西——编写一个应用并不是让火箭升空,只要拥有足够的时间及工具,我认为每个人都可以做到。

我也参与了很多关于女性的倡议:具体而言,就是试图鼓励更多女性进入科学、技术、工程和数学(STEM)领域。为让她们对 STEM 感兴趣,同时结合上面我所提到的对我的朋友的回应,导致我想为从零开始的初级程序员写一本书。虽然本书不是专为女性而写的,但这绝对是我灵感的来源。

为什么我会选择 Manning 出版社?那是因为我以前买了 Manning 出版社出版的很多图书,其质量都很好。并且多年前,我自愿为 Manning 出版社审阅手稿;我已经审阅过很多手稿了,所以我对图书出版的整个过程已经非常熟悉(在很高的层次上)。当初我决定写本书时,就知道必须与 Manning 出版社合作。我提交了我

对本书的想法,只等了几个星期,就收到了 Manning 出版社给予的肯定答复! 这让我诚惶诚恐!

这是一段漫长而有趣的旅程——一段免费之旅。这是一种交织着爱、乐趣、 恐慌、漫漫长夜、压力和幸福的劳作。这是我所撰写的第一本书,衷心感谢你的 阅读,希望你喜欢。

请记住: 你足够聪明,并且你的大脑会理解这个主题。保持耐心——你可以做到!

作者简介

Wendy L. Wise 在移动和应用开发方面有着广泛的背景,并与几家《财富》500强公司进行过合作。在17年的技术生涯中,Wendy曾担任软件开发高级总监、国际移动应用高级产品经理、网络和移动技术的实践开发人员,以及其他许多技术角色。Wendy完全拥抱其书呆子/极客的一面,正如你在阅读本书时所发现的一样。在其业余时间,Wendy喜欢啤酒、咖啡、摄影、野营以及户外运动。

有很多人我都需要表示感谢,因为在他们的帮助下,我的这个疯狂梦想才得以实现。感谢我最大的支持者:我的妻子 Jocelyn Whitfield。她相信我、支持我、鼓励我,在我失落时让我重新振作。没有她我绝对完成不了这本书。感谢我父母的鼓励和帮助,还有他们的养育之恩!感谢我所有的家人和朋友。或许你们并不知道你们的爱和支持给予了我多大帮助,特别是 Clay、Kristine、Nancy 和 Ja。我爱你们!

我还要感谢 Manning 出版社的优秀人物,是他们促成了这本书的出版:发行人 Marjan Bace 以及编辑和制作团队的所有人,包括 Christina Taylor、Janet Vail、Tiffany Taylor、Corbin Collins、Melody Dolab、Dennis Dallinik,以及所有在幕后默默奉献的其他人员。

我由衷地感谢由 Ozren Harlovic 领导的非常了不起的技术审稿小组——Stephen Byrne、Mark Cooper、Igor Delovski、Olivier Ducatteeuw、Laurence Giglio、Pieter Gyselinck、Marius Horga、Jocelyn Jeriah、Kelvin Meeks、Drew Monrad、Jason Pike 和 Stuart Woodward——以及才华横溢的论坛贡献者们。他们的贡献包括: 捕捉技术错误、术语错误以及打字错误,并提出主题建议。通过了审查过程的每一个建议以及通过论坛主题得以落实的每一条反馈,共同形成并塑造了该手稿。特别感谢担任本书的技术编辑 Robin Dewson,以及技术校对员 Scott Steinman。

我假定你买这本书的原因是想要学习如何开发 iOS 应用,并且你从未有过任何编码经验(好吧,或许你认识我,并且想要支持我)。无论哪种方式,你都将学到很多东西,并乐在其中。本书适用于从未编写过任何代码的人,或者编写了一些代码,但这些代码绝非 iOS 应用。我还假定你有耐心阅读本书、尝试书中的例子,并在这些例子第一次没有有效地工作时能够重写它们。这就是编码的精髓所在:调试(debugging)。即使是最好的开发人员也会出错,所以当你的代码在第一次工作不正确的时候,请不要沮丧。保持耐心最重要。

为什么选择 iOS 应用?如果你告诉你的朋友,你正在学习 iOS 开发(或 Swift,或者开发 Apple 手机),他们可能会说:"哇,我听说这很难。为什么你不从更容易的事情开始呢?"你的答案可以是:"我有编写应用的想法,我有耐心并且有意愿去学习。我相信我能行。"人们可能会鼓励你去学习更简单的东西,比如超文本标记语言(HTML,用于网页等)。其实这也挺好的,但你是真的因为想学习如何为Apple 设备(使用 Swift)创建应用,所以你正在阅读本书。在这里我要告诉你,你完全可以从 iOS 开始,并且你能够学会编写应用;只要付出时间、耐心和决心,只要你敢想,终有一天你会成为这方面的专家。

在这本书中,我的目标是仅提供给你足够的用于完成下一步操作所需要知道的内容,而并不会以填鸭方式向你灌输你现在所不需要知道的一些事情。换句话说,我正在尝试使用即时学习技巧来讲授这门课程。这意味着虽然可能会有大量资料来让你了解一个主题,但你并不需要超前学习所有这些内容。我会将所有这些资料汇总,然后将其分解成你真正需要了解的较小部分,以便开始学习。

许多人对学习编程心存疑虑。编程世界是庞大的——有太多的编程语言和首字母缩略词,并且有些程序员似乎常年不见天日。但请放心,你可以从学习基础知识开始,并且我将引领你了解首字母缩略词和程序员行话。同时,可将编程视

为另一种兴趣。你可以按需要花费尽可能多的时间,投入的时间越多,收获也将越多。想象着在你学习编程的旅途中有我陪伴——我们将一起到达目的地。

每个开发人员在一开始学习基础知识时都是循序渐进的。即使是经验最丰富的程序员也不得不从某处开始学起!没有哪个"俱乐部"是只允许某些特定的人成为开发人员的。每个人都可以做到,也包括你。只要肯花时间学习基础知识、理解概念,并进行实践,那么你也将很快成为 iOS 开发人员。

本书读者对象

本书针对纯粹的编程初学者(从未编写过一行代码,并且不了解所要做的事情的基本概念)。我做出如下假设:

- 你没有开发经验。
- 你想学习如何创建 iOS 应用。
- 你有可以用来编码的 Mac,或者你愿意购买 Mac。
- 你有耐心。

我希望本书能成为你开始学习编程的最佳之地,因为我知道你可以做到!

本书不话合哪些读者

如果你已是开发人员,并且已经熟悉诸如 for 循环、while 循环和 if 语句等概念,而你只想了解有关 iOS 的更多信息的话,这本书可能并不适合你。你可以找到其他讲授 Swift 语法以及介绍移动开发与其他平台开发之间差异的书籍,而且你可能很快就会觉得这本书很无聊。但是,如果你是一名程序员,且只熟悉 HTML或 COBOL 等语言,那么绝对可以从这本书中学到一些东西。

路线图

本书分为三个部分:

- 第 I 部分(第 1~8 章)——该部分将开启你的编程生涯。你将学习编程的基础知识、了解 Xcode 和 Swift Playground,并将从编写一些简单的应用来开始学习。
- 第II 部分(第 9~16 章)——这些章节将向你介绍创建应用所需的其他一些技能和概念,包括 while 语句、switch 语句、数组和集合、故事板(storyboard)、ViewController(视图控制器)以及表。第II 部分的内容比第 I 部分更高级,所以请先确保你已理解第 I 部分内容。
- 第Ⅲ部分(第 17~27 章)——在这些章节中, 你将创建一个 LioN(喜欢或不喜欢)应用。LioN 应用允许用户将项目添加到列表中, 并评价他们是否喜

前 i

欢这些项目。当我在商店里购物时,永远不会记得我喜欢哪种牙膏;所以,我会打开 LioN 应用并搜索牙膏(toothpaste),该应用将显示我所喜欢的牙膏。这款应用将作为你继续创建自己的应用的一个完整示例。

源代码下载

你可以下载本书的所有项目,并随时进行参考。其可在 Manning 网站(www. manning.com/books/anyone-can-create-an-app)和 GitHub(https://github.com/wlwise/ Anyone Can CreateAnApp)上找到。

另外, 也可扫描本书封底的二维码下载源代码。

软件/硬件要求

以下是你开始学习编程的必备条件:

- Mac 电脑——如果你没有 Mac, 可根据第 1 章列出的基本要求去购置。
- Xcode——该集成开发环境(IDE)是创建应用所需的主要工具。你可能已经使用诸如 Microsoft Word 的程序来创建文档。而 Xcode 是你用来创建程序的应用。附录 A 有安装该免费工具的说明,你将在第 2 章中开始使用它。我还将在第 4 章中详细介绍 Xcode。
- 苹果开发者计划(Apple Developer Program)的会员资格——附录 A 说明了如何加入。有两个选择:免费会员(我推荐)和 99 美元的会员资格。

在线资源

本书最能帮助到你的办法是浏览其网站: www.manning.com/books/anyone-cancreate-an-app。你可以下载示例、在 Author Online 论坛上提出有关练习的问题,并与其他读者交流。我会尽可能回应你的论坛问题,当然你也可以告诉我你对这本书的看法。

Apple 是另一个用于探索和学习的好地方,包括苹果开发者中心(Apple Developer Member Center,http://mng.bz/3OjD)的资源。你将需要成为苹果开发者计划的成员;有关更多信息,请参阅附录 A。你并不需要为了使用这本书而去阅读那些文档——我仅是为你提供定位。Getting Started 的资源是开始学习的好地方,且 Guides 部分是另一种很好的资源。

Stack Overflow(www.stackoverflow.com)是获取有关具体问题的答案的另一个好地方。可以根据你的具体需求进行搜索,几乎可以保证该问题已被问过,且其他人已经回答。在搜索时应尽可能具体,否则将收到很多信息。

目 录

筹	専Ⅰ部分 第一个应用		2.1.3 步骤 3: 设置你的项目 选项15
第 1 章	入门3		2.1.4 步骤 4: 运行空白应用 … 17
1.1	蓝图: iPhone 和 iPad 开发3		2.1.5 步骤 5: 添加 Hello
	1.1.1 一些关键术语		World 文本18
	1.1.2 开发和编程的区别4		2.1.6 步骤 6: 运行应用 19
	1.1.3 客观地看待 Swift ··········· 5		2.1.7 步骤 7: 回顾本章内容 … 20
	1.1.4 你将创建的应用5	2.2	小结21
1.2	了解需要记住哪些内容6	第3章	讲解第一个应用23
	1.2.1 理解并牢记关键概念7	3.1	讲解 Xcode 模板 ······· 23
	1.2.2 语法7	3.2	了解单视图应用24
	1.2.3 伪代码的重要性8	3.3	标签的定义25
1.3	创建 iPhone 和 iPad 应用需要	3.4	模拟器的定义 27
	做哪些准备9	3.5	小结 29
	1.3.1 你需要一台 Mac 9	第4章	深入了解开发工具:
	1.3.2 Xcode: iPhone 和 iPad		Xcode31
	开发环境11	4.1	Xcode 面板说明 31
	1.3.3 一些有用的资源11		4.1.1 Standard Editor 32
1.4	小结12		4.1.2 Utilities 面板 ······· 34
第2章	构建你的第一个应用·······13		4.1.3 Main.storyboard······ 35
2.1	首次启动 Xcode ······ 14		4.1.4 Navigator 面板35
	2.1.1 步骤 1: 启动 Xcode ······ 14	4.2	Xcode 图标说明 37
	2.1.2 步骤 2: 创建一个新的	4.3	自由探索 38
	项目14	4.4	小结 39

第5章	捕获用户的操作:添加		7.1.1 步骤 1: 创建一个新的
	按钮······41		单视图应用66
5.1	添加标签和按钮 ······41		7.1.2 步骤 2: 向视图添加一个
	5.1.1 步骤 1: 使用 Single View		按钮和一个标签66
	App 模板开始创建新		7.1.3 步骤 3: 向视图添加一个
	项目42		文本框67
	5.1.2 步骤 2: 向故事板添加		7.1.4 步骤 4: 将按钮、标签和
	按钮和标签,然后运行		文本框连接到代码(关联),
	应用进行测试42		并测试应用67
	5.1.3 步骤 3: 将按钮和标签		7.1.5 步骤 5: 添加代码以更改
	连接到代码(关联),然后		标签,并测试应用 69
	运行应用进行测试43		7.1.6 步骤 6: 代码注释 71
	5.1.4 步骤 4:添加代码以便按	7.2	小结·······71
	钮被点击时更改 Label	第8章	玩转 playground ······73
	上的文本,并运行应用	8.1	Swift Playgrounds: 学习与
	进行测试47		他人互动73
5.2	改变标签的显示 49	8.2	Framework 74
5.3	小结52	8.3	变量类型 75
第6章	按钮应用详解 ······55		8.3.1 字符串77
6.1	按钮说明55		8.3.2 回到数学课堂
	6.1.1 创建 outlet ······55		8.3.3 双精度浮点型 79
	6.1.2 创建 action ······56	8.4	小结81
	6.1.3 Xcode 有超能力 ·······57		
	6.1.4 应用的前端: 用户	第Ⅱ部	分 理解关键的开发概念
	界面58	空 0 辛	校判应用协运和 05
6.2	文档59	第9章	控制应用的流程85
6.3	注释:尽量简短明了61	9.1	控制流程
	6.3.1 注释是你的朋友61	9.2	简单的 if 语句
	6.3.2 如何注释代码61	9.3	复合型 if 语句: 或() ·········· 88
6.4	小结62	9.4	复合型 if 语句: 与(&&) ······ 89 else if 语句······ 90
第7章	捕获用户的输入:添加	9.5	else 语句 90
	文本框 ······65	9.6	小结 ······ 92
7 1	添加文本框66	9.7	小垣94

第 10 章	while 控制语句······95	第 12 章	用故事板讲故事 119
10.1	使用 while 语句控制	12.1	故事板 119
	代码95	12.2	创建一个故事板应用 119
	10.1.1 while 语句的实践 ······96		12.2.1 步骤 1: 创建一个名为
	10.1.2 while 语句小结 ·······97		StoryboardExample 的
10.2	switch 语句98		新应用120
10.3	我伸出了几根手指?100		12.2.2 步骤 2: 添加第二个
	10.3.1 将所有组件添加到		场景120
	故事板中101		12.2.3 步骤 3: 在第二个场景
	10.3.2 关联故事板101		上添加导航栏 122
	10.3.3 创建变量以捕获所		12.2.4 步骤 4: 将 Cancel 按钮
	猜测的数字:		链接到第一场景 124
	numberGuessed·······102	12.3	segue 的转场动画类型…124
	10.3.4 当步进器被点击时更改	12.4	小结125
	变量 numberGuessed	第 13 章	深入讲解 View-
	的值103	77 10 7	Controller ················ 127
	10.3.5 连接 Guess!按钮104	13.1	继承·······127
10.4	小结106	13.1	关键字 override ··········· 130
第 11 章	集合107	13.2	ViewController 的生命
11.1	数组107	13.3	周期131
11.2	for 循环语句109	13.4	应用的生命周期133
11.3	字典110	15.4	13.4.1 步骤 1: 创建一个名为
11.4	创建查找州名的应用 113		Lifecycle 的新项目···· 133
	11.4.1 步骤 1: 创建一个名为		13.4.2 步骤 2: 添加第二个
	StateAbbreviationLookup		ViewController ······· 133
	的应用113		13.4.3 步骤 3: 创建一个可解
	11.4.2 步骤 2: 将 UI 组件		约的 segue134
	添加到故事板113		13.4.4 步骤 4: 重写五个
	11.4.3 步骤 3: 将 UI 组件		函数135
	连接到代码114		13.4.5 步骤 5: 测试应用…135
	11.4.4 步骤 4: 创建州的缩写	13.5	小结137
	和名称的字典115	第 14 章	→ ···· ··············· 139 创建选项栏··············· 139
	11.4.5 步骤 5: 创建代码以实现		
	当用户键入州名时查找	14.1	Tab Bar Controller 139
	州的缩写115		14.1.1 步骤 1: 创建一个新的
11.5	小结118		应用140

	14.1.2 步骤 2	: 删除现有的		16.1.3	可维护性165
	场景…	140		16.1.4	可扩展性165
	14.1.3 步骤 3	: 在故事板中	16.2	设计模	真式的类型 166
	添加一	一个 Tab Bar		16.2.1	Model-View-Controller
	Contro	oller141			设计模式166
	14.1.4 步骤 4	: 在不同的选		16.2.2	代理模式168
	项卡上	·添加标签·····142		16.2.3	备忘录模式168
	14.1.5 步骤 5	: 在应用中添	16.3	小结…	169
	加第三	三个选项卡145	kh m	Ja 1105	All 7th, was a second
14.2		146	第皿	部分	创建 LioN 应用
第 15 章	深入讨论表	视图······147	第 17 章	活学活	5用:LioN 应用···· 173
15.1	代理	147	17.1	Like it	or Not 173
	15.1.1 从头开	F始制作	17.2	开始…	174
	比萨饼	f147		17.2.1	创建应用175
	15.1.2 委托比	公萨饼制作148		17.2.2	添加 Navigation
15.2	协议	148			Controller 176
15.3	数据源	149		17.2.3	添加 iPhone 4s
15.4	创建表视图图	应用 ⋯⋯⋯ 149			模拟器177
	15.4.1 步骤 1	: 创建一个		17.2.4	将数据与表视图
	新的应	☑用 ······150			关联······179
	15.4.2 步骤 2	: 将表视图添加		17.2.5	实现表视图的函数…181
	到 Vie	wController ···· 150	17.3	小结…	183
	15.4.3 步骤3	: 建立一个原	第 18 章	为 Lic	N 应用添加
	型单元	□ たね152			185
	15.4.4 步骤 4	: 为 UITable-	18.1		N 应用中添加硬
	View i	没置协议152	10.1	-	女据185
	15.4.5 步骤 5	: 创建比萨的			创建一个包含模拟
	数据源	ā157		10.1.1	数据的数组186
	15.4.6 步骤 6	: 将数据与		18.1.2	通过硬编码数据将
	表关耶	ŧ159		10.11.2	lionData 连接到表
15.5	小结	161			视图186
第 16 章	模式: 学习	逢纫163	18.2	添加档	英型187
16.1		其定义163	18.3		、 至单元格的布局····· 192
		経洁164	10.5	18.3.1	改变故事板中的单元
		上解164			格以显示描述 192
	>> • • • -		I		14 2/3E/4/4HAE 172

	18.3.2	更新函数以显示	20.3	小结…	216
		描述193	第 21 章	AddE	ditView 场景 ······ 219
18.4	小结…	194	21.1	创建新	新的详情视图·······219
第 19 章	显示L	_ioN 的详细信息····195		21.1.1	添加一个新的 Table View-
19.1	捕获被	皮点击的索引行 195			Controller 220
19.2	在故事	耳板中添加		21.1.2	添加一个新的 AddEdit-
	详情页	ī ······196			ViewController 类…·221
	19.2.1	将一个 ViewController		21.1.3	关联 Cancel 和 Done
		添加到故事板196			按钮223
	19.2.2	创建一个新的		21.1.4	检查点223
		ViewController 类…·198	21.2	添加新	折的 LioN224
19.3	将数据	居传递给		21.2.1	禁止单元格被选中…226
	Detail	ViewController ····· 199		21.2.2	设置键盘的行为227
	19.3.1	准备 DetailView		21.2.3	在用户点击时关闭
		Controller 以接受			键盘229
		LioN200	21.3	小结…	231
	19.3.2	更新 MainViewController	第 22 章	代理え	无处不在··········233
		以传递数据201	22.1	关联初	见图233
19.4	小结…	205		22.1.1	实现协议
第 20 章	为详情	青视图添加内容⋯207		22.1.2	更新 Cancel 以及 Done
20.1	将标签	E添加到详情			按钮的事件 235
	界面 "	207		22.1.3	捕获用户输入236
	20.1.1	使用 description 将 Int	22.2	MainV	ViewController 的
		转换成字符串208		一致性	± ····· 237
	20.1.2	使用 String 将 Int 转换	22.3	将 Lio	N 对象添加到 lion
		成字符串208		数组中	≒239
20.2	向列表	長添加新的 LioN… 209	22.4	设置1	ike 和 dislike
	20.2.1	在视图中添加+		属性…	242
		按钮210	22.5	小结…	243
	20.2.2	创建函数以处理 action	第 23 章	编辑	LioN245
		并将二者链接211	23.1		 见有的 LioN ········ 245
	20.2.3	将硬编码值添加到 LioN	_5.1	23.1.1	设置 AddEditView-
		列表中213			Controller 以接受要编
	20.2.4	从列表中删除			辑的 LioN 对象245
		LioN215			7-7HJ LIOI (/13) 2-13

	23.1.2	为文本框填充 LioN 的		25.1.6	让 MainView 场景
		名称及其描述246			更美观278
	23.1.3	显示对 LioN 的喜欢		25.1.7	更新导航栏279
		或不喜欢247	25.2	添加图	图标280
	23.1.4	将 LioN 对象传递给	25.3	更新启	目动场景281
		Add / Edit 控制器····250	25.4	小结…	281
	23.1.5	点击 Done 时保存 LioN,	第 26 章	使用自	自动布局 ·······283
		并非创建新的	26.1		5局以适配所有
		LioN252	20.1		ラブラスとHII// 17 ラブ ····································
23.2	小结…	256		26.1.1	
第 24 章	保存I	_ioN257		20.1.1	场景284
24.1		·盒···················257		26.1.2	
24.2		文据·······259		20.1.2	格的颜色292
	24.2.1	更改 LioN 对象的	26.2	小结…	293
		类定义259			
	24.2.2	保存编码的数据262	第 27 章		LioN295
	24.2.3	加载解码的数据262	27.1		搜索功能·······295
	24.2.4	添加 loadLions()	27.2		月户的输入过滤
		函数264		LioN.	297
	24.2.5	数据加载总结265		27.2.1	创建过滤器函数298
	24.2.6	添加保存功能265		27.2.2	使用闭包来过滤
24.3		1载与保存功能267			数组299
24.4	小结 "	268		27.2.3	更改表视图的
第 25 章	让 Lio	N 应用更美观 ·····269			数据源 301
25.1	基本的	的修复269		27.2.4	润饰应用303
	25.1.1	创建两个 section ······270	27.3	搜索基	其他字段304
	25.1.2	为 Like 和 Dislike	27.4	小结…	305
		添加图片271	27.5	你将何	可去何从? 306
	25.1.3	更改表视图的	 附录 A	加何安	装 Xcode 以及注册
		背景色275	אאנוין		
	25.1.4	根据选择来切换		成为苹	果开发者
		图片276	附录 B	在你的	设备上运行应用 ··· 311
	25.1.5	在单元格上设置			
		图片277			

第Ⅰ部分

第一个应用

第 I 部分介绍编程的基本概念,旨在让你尽快涉足编程世界。第 2 章创建第一个应用,然后第 3 章详细介绍要做什么以及为什么这么做。第 4 章介绍用于创建应用的工具(Xcode)。第 5~7 章将通过在应用中添加按钮和文本框的操作来更深入地讲解编程方法。最后,在第 8 章中,将介绍一个令人兴奋的工具 Swift Playground,可以通过它快速轻松地学习和测试代码。

第一章入门

本章涵盖

- 概述 iPhone 和 iPad 应用的创建
- 学习策略——记住什么
- 创建 iOS 应用需要做哪些准备

社会在不断发展,mobile-first 也在其中起到推动作用。mobile-first 是什么?是一种想法,即让人们通过 iPhone 或 iPad 获取信息——无论是电子邮件、新闻、社交媒体、互联网、购物、发短信或电话。因此,如今一些公司正考虑如何在移动设备上提供这些信息,为此首先要制作网页,然后添加可用来访问网页的移动应用。

技术的进步和发展是多么激动人心!你将乐在其中。本书将教你如何开始创建 iPhone 和 iPad 应用,适合于以前没有任何开发经验的初学者,将带这些初学者初次踏入应用创建的精彩世界。如果你已具备应用开发经验,也可通过本书复习和巩固学过的知识,何乐而不为?

1.1 蓝图: iPhone 和 iPad 开发

为 iPhone 和 iPad 创建应用是一件令人兴奋的事情。但很多人一想到要为这些很棒的设备创建应用可能会望而生畏,不过请放心,我们将一起完成应用的创建,并且你将在学完本书之后完成多个应用。为 iPhone 和 iPad 创建应用是非常复杂的,因为它有自己的词汇,不过对于这些你不必担心,我都将一一传授于你。

创建应用称为开发软件(developing software),当你学完本书之后,将开发出多个应用,并将初步了解软件术语和基础知识。让我们从零基础开始,以确保我们在同一页(双关语)。

1.1.1 一些关键术语

首先,学习开发 iPhone 和 iPad,意味着将创建只能安装和运行在 iPhone 和 iPad 上的应用。这些应用无法在任何其他设备上运行,因此请确保你已了解了自己正在进行的事情。你将学习开发在 iOS 上运行的应用,iOS 是在 iPhone 和 iPad 上运行的操作系统(OS)。而操作系统是在机器上运行的底层软件,如 Windows 95(已被淘汰)、Windows 8.1 或 Mac OS X(现为 macOS)。这些操作系统负责与组成电脑的电子元件进行交互的所有艰巨任务,因此,按下计算机按键时,是在与操作系统进行交互,而不是在尝试将你的意图告诉计算机。本书前几章介绍 iPhone 和 iPad 开发的关键概念,但第一个例子将重点围绕 iPhone 开发。开发 iPad 的功能并不是那么有意义,随着我们进入后面的章节,将会展现这两种开发的差别。

1.1.2 开发和编程的区别

术语开发(development)与编程(programming)是同义词,因为学习开发或编程应用是为了让计算机能够做事情。有非常多的编程语言可供选择,因为编程语言 (programming language)是用于与机器通信而被正式构建的语言。当 Apple 创建 iPhone 时,也创建了一种名为 Objective-C 的语言(Swift 之前使用的编程语言),当由程序员使用并被编译器(compiler,稍后再介绍)编译或翻译后,可被 iPhone 理解并执行。图 1.1 展示了这种编程语言的实现过程。

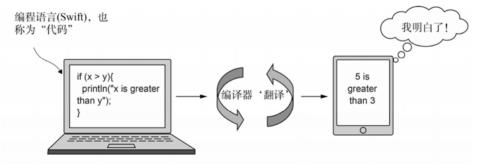


图 1.1 应用的创建过程(应用的基本常识)

如图中所示,使用编程语言为 iPhone 或 iPad 创建应用时包括以下步骤:

- (1) 在屏幕上键入单词,这些是命令(command)或代码(code)。
- (2) 编译器接受这些单词,如果程序中没有出现错误,便将其翻译成机器可

以理解的语言。

学习程序很像学习外语。首先需要了解这些词和它们的含义,然后必须把 它们组织在一起,以正确的顺序形成句子。

1.1.3 **客观地看待** Swift

如前所述,有非常多的编程语言,而它们都具有与机器通信的相同目的,然后让机器按照用这些语言编写的程序进行操作。对于 iPhone 和 iPad 的编程,现在有两种编程语言可选择:

- Objective-C
- Swift

本书并不打算告诉你这两种语言的历史,也不会试图让你相信哪一种更好。自从 iPhone 诞生以来,Objective-C 就作为 iPhone 和 iPad 编程的首选语言。Apple 公司在 2014 年发布了 Swift,与 Objective-C 完全不同,Swift 采用了不同的方法开发 iPhone 和 iPad。相对于许多其他语言的复杂性,Swift 简化了许多,从而更容易学习。而这些差异并不是本书的重点,本书甚至不会讨论为什么Apple 创建了一种新的语言。最基本的一点就是你将会学习 Swift,而在这本书里你会用 Swift 编写代码。

值得庆幸的是,你和所有人一样,都是刚开始学习 Swift。因为这是一种新的语言,所以没有任何一个人已经拥有五年的 Swift 经验。想想看:如果你愿意,就可以在五年内成为那样的专家!相对于 Objective-C 语言,在 Swift 中更容易学习语法的编写。如果想了解有关 Objective-C 的更多信息,可以通过苹果开发者(Apple Developer)网站(developer.apple.com)获取。

1.1.4 你将创建的应用

接下来是令人兴奋的部分:通过阅读本书,你将创建自己的应用。我们将从一些较小的应用开始,以帮助你了解基础知识,但最终将构建 LioN 应用,你现在也可以从 App Store 免费下载它。以下是本书开发的内容:

- Hello World——这是每个开发者踏入编程世界的成人礼。应用启动后将显示"Hello World"。此应用很简单,但它的成功启动确保了所有设置的正确性,并且让你知道如何创建应用。
- Hello Button——该应用允许你按下一个按钮来更改应用内标签的文本。
- Textbox——该应用允许你在文本框中输入内容,然后在屏幕上打印出所

输入的文本。

- How Many Fingers——该应用会让你玩游戏:一个人(指此应用)将举起的 手指藏在身后,你要猜出其举起了多少根手指。
- State Abbreviation Lookups——该应用允许你输入国家的名称,然后将显示国家的缩写。
- LioN——款非常有用的应用,能够帮助你记住"Like it or Not."。我创建此应用,是因为我容易忘记自己喜欢或不喜欢的品牌——比如牙膏。我去杂货店买一个品牌,是因为我记得,但事实证明,是因为我不喜欢它我才记得。但现在我可以打开我的 LioN 应用并搜索牙膏,它可以告诉我,我不喜欢的品牌以及喜欢的品牌都有哪些。非常方便! 此应用具有搜索功能,因此可以根据名称或描述搜索你的 LioN 项目。除了屏幕底部的广告和一些注意事项外,你构建的 LioN 应用将与 App Store 中的完全一样。

在我们开始学习 Swift 语言及其语法之前,将介绍一些学习策略,以使编程 更容易。

1.2 了解需要记住哪些内容

你将开始明白,编程世界也是浩瀚无边的。尽管在互联网上有大量的可用资源,却要么无处着手,要么无迹可寻。本书内容将尽最大可能做到通俗 易懂。

学习概念的关键是完成整个编程练习。可以先阅读概念,然后做相应的练习:全部的。要达到学习并理解的目的,没有比亲自动手更好的方法了。

做练习时,难免会犯错。即使是世界上最好的程序员也会犯错。关键是要有耐心,能从错误当中汲取经验和教训。请确保完全按照练习来做,否则它们可能不会按照你所期望的方式来工作。起初你可能不会明白为什么这么做,但随着我们讲解的继续,一切都会变得清晰起来。

你可以随时检查你的成果

你可以随时从 Manning 网站(www.manning.com/books/anyone-can-create-an-app)或 GitHub(https://github.com/wlwise/AnyoneCanCreateAnApp)上的本书页面中下载练习和源代码,以了解我的代码是如何写的,并将其与你所写的代码进行比较。

1.2.1 理解并牢记关键概念

现在,你需要在编程的海洋中记住哪些内容呢?最重要的是记住我教你的概念(concept)。此时,在一开始就记住基本概念,比起要记住确切的语法(编程中使用的单词和符号的组合)内容更重要。

为什么?你可以随时写下语法或轻松在线查找它,但如果你不了解基本概念,知其然而不知其所以然,那么你只是在做无用功。只学习语法将类似于我要求你用德语复制一个句子(假设你不懂德语)。你可以把它复制下来并记住它,但是除非你懂德语,否则它对你没有任何意义。当我们一起踏上这趟学习之旅时,我会说清楚哪些主题是概念(concept),而哪些是语法(syntax)。你对概念的理解越多,你的编程经验就越丰富。一些关键概念将在页面上单独的框中给出,而语法主题将形如:Syntax:example。

1.2.2 语法

如果查看语法(syntax)的定义,将看到一些诸如"在语言中通过组织单词来创建格式良好的句子"的内容。你将学习 Swift 语言的语法——如何编写出组成 iPhone 或 iPad 代码的"句子"。正如我所说,这与学习外语相似。在某些语言(包括英语)中,习惯上这样说或写:

- 首先是句子的主语
- 然后是动词
- 之后是对象

例如, I read a book yesterday 在英文中很好理解。如果用德语(例如)来表达或者写同一句话,将形成这样一个句子 Ich habe gestern ein Buch gelesen.(I have yesterday a book read.)。这就是不同语言间的语法差异。

每种语言都有语法规则,以使撰写者或演讲者能够轻松地形成句子,并使读者或听者能够理解这种语言。编程语言也有语法规则,以使计算机能理解你要试图做什么。当创建应用时,能够编写格式良好语法正确的代码是很重要的——但理解基本概念更重要。随着你记住语法的代码越来越多,记忆"句子"将变得越来越容易。本书中讲授的基本概念很重要,因为这将成为你理解编程的基石。

那么该如何学习编程的基本概念呢? 我很高兴你这么问。

1.2.3 伪代码的重要性

程序员必须理解他们所要实现的任何代码的基本概念或计划。一种理解和学习程序的基本概念的方式是,先将任何你想要程序实现的代码写成伪代码(pseudocode)。

伪代码——在本书中意指"假代码"——这种方法将概念和想法简化并分解成逻辑上的步骤,以便更容易理解。编写伪代码有助于自己对将要实现的程序代码有进一步的理解和规划,而不是为了告诉计算机你要做什么。想象一下第一次在房子的墙上作画。你显然希望自己能创作出一幅漂亮的墙画,但是要从哪里开始呢?如果从心中预想的最终结果开始(墙画已经完成)并试图弄清楚如何实现结果,那么你将为了实现最终结果而需要去考虑明确的步骤。花点时间想想你创作出这幅墙画的步骤,然后把它们写下来。

下面是我写下的步骤:

- (1) 去当地的五金店,浏览涂料的样品芯片。
- (2) 选择一个或多个涂料样品芯片并购买样品涂料。
- (3) 将样品涂料涂抹在墙上的几个区域。
- (4) 在白天多次检查已涂抹的区域,看看它们在不同光线下的效果。
- (5) 选择其中一种颜色,做出使用与否的决定。
- (6) 回到当地的五金店,购买更多所选的涂料。
- (7) 拿出滚筒、罩布、油漆刷和工作服。
- (8) 为所有东西的边缘贴上胶带。
- (9) 绘画。

你可能比我写下来的步骤少了几步或者多了几步,但这并不意味着我们的任何一个列表都是"更正确"或"更不正确"。当你看着自己的列表时,便清楚了绘画所需的步骤。这难道对你没有意义吗?它是否足够完整到你可以正确阅读并理解它?如果是,那么很好,你写出了绘画的伪代码。除非你最终没能创作出一幅漂亮的墙画,否则伪代码便没有正确与错误之分。可以根据实际需要多次完成步骤,以达到预期结果。

步骤里说:"拿出滚筒、罩布"等。而不是说"拿出滚筒、罩布,确保它们干净,将它们放入房间,如果是新的,就将包装物从滚筒上取下"等。画了这么多墙画后,我认为"拿出滚筒"已经包括了这部分。这对于我是有意义的。你需要确保你的步骤对你也是有意义的。本书中含有大量的伪代码示例的练习以及代码,通过这些你将能进行充分的实践和学习。记住:最关键的一点是列出的最终

步骤是对你有意义的。只要是以符合逻辑、有序的方式实现最终结果,做法就没 有正确或错误之分。

现在你掌握了一些如何有效学习的方式,接下来让我们谈一谈如何开始编程。

1.3 创建 iPhone 和 iPad 应用需要做哪些准备

那么,怎样开始呢?

- 首先, 你需要一台 Mac 电脑。
- Xcode,它是一个集成开发环境(IDE, Integrated Development Environment), 是创建应用的主要工具。你可能使用过像 Microsoft Word 这样的程序来 创建文档。而 Xcode 是用来创建其他程序的一种程序。附录 A 中有安装 此免费工具的说明。我们将在第 2 章开始使用它。
- 加入苹果开发者计划(Apple Developer Program)。附录 A 中说明了如何加入。有两种选择:免费计划(推荐)或 \$99 计划。可以查看附录 A 了解更多相关信息。

在此之前,先概览 Mac 和 Xcode,并介绍一些有用的资源,之后再带你到 附录 A 中查看如何安装 Xcode 并阅读有关苹果开发者计划的更多信息。

1.3.1 **你需要一台** Mac

Apple 要求 Mac、iPhone 和 iPad 的所有开发都要在 Mac 上进行。然而一些野心勃勃的开发人员已经找到了解决这个问题的方法——不用在 Mac 上也可以进行开发,但需要掌握很多技术诀窍。请相信我——如果你还没有一台 Mac 的话,那就去购置一台吧。并不一定需要一个超大型的、全新的、高性能的 Mac,但如果有一台更快的 Mac,你将更容易编写和测试代码。这台 Mac 至少应满足以下几点:

- 硬盘上有大量可用的磁盘空间。
- 至少 8 GB 的 RAM, 越多越好。
- OS X 10.12.x 或 Sierra 或更高版本(在撰写本文时, macOS Sierra 是最新版本)。重要的是使用最新版本的 Mac OS——它与最新版本的 Xcode 最兼容,并将支持最新的 iPhone 功能。

通常可在 Craigslist 或 eBay 这样的网站上买到相当不错的 Mac, 但在购买之前请先确保你知道自己想要得到的是一个什么样的 Mac。我建议从卖家那里获

取要购买的 Mac 的序列号(如果不是新的),并在 Apple 网站上进行检查以确保它是正牌产品,并查看是否仍在保修范围内(CheckCoverage.Apple.com)。你可以使用 iMac、MacBook(Pro 或 Air)或 Mac Mini 进行开发。线上 Apple Store 可以让你轻松选择和配置 Mac,如图 1.2 所示。



图 1.2 Apple Store 机器选项——在线工具可让你比较不同的 Mac 配置

Apple 提供了一个方便的比较工具,可以在 www.apple.com/mac/compare 上轻松地查看不同配置之间的差异。还可以通过 www.apple.com/shop/browse/home/specialdeals/mac 从 Apple 购买翻新的 Mac,这样可以节省一些钱。翻新的 Mac与新 Mac 一样具有一年保修期。如果你是在校学生,还可以通过从 Apple Education 商店购买来节省开支。打开苹果商店主页(http://store.apple.com),一直滚动到页面的底部,找到链接 For Education,即可进入 Apple Education 商店。

请确保运行在 Mac 上的是最新操作系统(macOS)。如果你的 Mac 比较陈旧 且在上面运行的不是最新操作系统,可通过 App Store 进行升级。根据 Mac 操作 系统的版本而定,升级可能需要付费,但许多更新是免费的。

本书的练习需要 Sierra OS X 10.12.4 或更高版本

为与本书中的练习保持一致,至少需要安装 Sierra(OS X 10.12.4 或更高版本)。这些练习都是使用 macOS Sierra 编写的。

如果你不知道现在安装了什么版本的 OS X, 很容易找出来。单击屏幕左上角的 Apple(♠), 然后选择 About This Mac(中文系统: "关于本机")。可在图 1.3 中看到, 我的 MacBook 正在运行 macOS Sierra, 版本是 10.12.6。



图 1.3 单击 Apple 图标(▲), 然后单击 About This Mac(关于本机),来确定操作系统的版本

1.3.2 Xcode: iPhone 和 iPad 开发环境

有了 Mac 后,接下来就需要安装 Xcode。此 IDE 是用于创建应用的主要工具。可将 IDE 看成是 English 和 iPhone 或 iPad 之间的翻译器。IDE 的行为有点像 Google Translate(http://translate.google.com):

- (1) 在你所知道的语言框中输入单词。
- (2) 选择一种你不懂的语言, 然后单击 Translate。
- (3) 给出译文。

如果英文单词拼写有误,Google Translate 便不知道该怎么做,从而显示错误。如果输入格式不正确的句子,翻译器可能会给出一个错误或不是你所期望的答案。

IDE 以同样的原则行事。当键入格式正确的"句子"时,IDE 将其翻译为 iPhone 或 iPad 可以理解的机器语言。当拼写错误的单词或格式不正确的"句子"时,将出现错误,或者设备的行为方式与你所期望的不一样。许多编程语言都有自己的 IDE。Apple 使用 Xcode 进行所有开发,包括 iPhone 和 iPad。

附录 A 中包含了 Xcode 的安装说明和有关注册苹果开发者(Apple Developer) 账户的详细信息。完成本章后并在开始第 2 章之前,你应该参阅附录 A 完成这些步骤。

接下来,我们来看一些有用的资源,以使 iPhone 和 iPad 的程序编写更加简便。

1.3.3 一些有用的资源

本书将为你提供创建多个应用所需的所有步骤,并讲解实现这些步骤的基本概念。在某些情况下,你可能想要了解有关特定主题的更多信息,或者可能想深入了解一个主题。下面这些很好的资源可帮助你实现这一点:

- 本书的网站——可以在这里下载示例,在 Author Online 论坛中提出有关练习的问题,并与其他读者聊天。网站地址是 www.manning.com/books/anyone-can-create-an-app。我会尽可能地回复你在论坛上的问题,你还可以提出对本书的看法。
- Apple——这是一个开始学习和探索的好地方,但起初你可能无从下手。可以在苹果开发者中心(Apple Developer Member Center, https://developer.apple.com)中查看一些"入门"资源。但首先需要成为Developer Program 的成员(更多相关信息请参阅附录 A)。你并不需要为了使用本书而去阅读这些文档。如果你想了解有关主题的更多信息,我

将会为你提供定位。首页顶部的 Develop 链接是开始进入苹果开发者中心的入口: 然后网页跳转后选择 Guides 链接。

• Stack Overflow——此网站(www.stackoverflow.com)是获取有关具体问题的答案的另一个好地方。可以根据你的具体需求进行搜索,几乎可以保证该问题已被问过且其他人已经回答。在搜索时应尽可能具体,否则将收到很多信息。

记住,学习编写 iPhone 和 iPad 应用可以说是一项艰巨的任务,不要幻想一次就会成功。本书会在你需要的时候给你提供信息,如果你想了解有关主题的更多信息,也会给你提供进一步资源的建议。想想古老的格言: How do you eat an elephant? One bite at a time。你无法一口吃成个胖子。你需要一步一个脚印地前进。这就是为什么我要你学习"just in time"的重要主题(所以我们很清楚,我无法容忍这种一蹴而就的事)。为能在第 2 章中开始开发,现在是时候转到附录 A来安装你需要的工具了。

要牢记的概念

- 希望你为创建 iPhone 和 iPad 应用而感到兴奋,并意识到你可以在本书的帮助下不断提高水平。
- 你学到了伪代码就是"假代码"。你将创建自己的伪代码,并作为一种路线图,来帮助你开发应用。
- 你学到了语法(syntax)是编码"句子"形成的方式。你必须使用正确的语法,以使计算机按照你的指令去工作。
- 你学到了 Xcode 是用于创建代码的 IDE。IDE 可将代码编译(翻译) 为 iPhone 和 iPad 可以理解的语言。
- 你学到了要成为 iPhone 或 iPad 开发人员需要什么类型的机器(Mac)。
- 你学到了需要记住的内容,将来可以参考什么以及可以在线查询什么。
- 你学到了去哪里寻求更多帮助。

1.4 小结

恭喜——你刚刚学完了本技术书中的第 1 章! 记住一定要脚踏实地地去学习。本书将在你需要时为你提供需要的内容。在本书结束时,你将熟练掌握编码术语。你会知道一些编码行话,你将为 iPhone 和 iPad 创建多个应用。让我们赶紧开始吧,你将在下一章创建第一个应用(但首先请参阅附录 A)。