

TURING 图灵程序设计丛书

Peachpit
Press

(第2版) Swift基础教程

零基础上手Swift · 大量代码+实例

【美】Boisy G. Pitre 著 袁国忠 译



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

TURING

图灵程序设计丛书

(第2版)

Swift基础教程

【美】Boisy G. Pitre 著 袁国忠 译

人民邮电出版社
北 京

图书在版编目 (C I P) 数据

Swift基础教程 / (美) 皮特 (Pitre, B. G.) 著 ; 袁国忠译. -- 2版. -- 北京 : 人民邮电出版社, 2016. 6
(图灵程序设计丛书)
ISBN 978-7-115-42230-9

I. ①S… II. ①皮… ②袁… 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083125号

内 容 提 要

本书针对初学者, 将类、函数、闭包等 Swift 概念的介绍贯穿全书始终, 结合 Swift 实例, 一步步指导读者开发 App。书的第二部分创建了 2 个简单但完备的应用程序, 并研究了一个完整 2D 游戏的源代码。第 2 版针对 Swift 2 进行了全面更新, 并增加了实例, 旨在基于概念和交互式场景让读者快速扎实掌握苹果开发技能。

本书适合任何想学习 Swift 的人参考。

-
- ◆ 著 [美] Boisy G. Pitre
 - 译 袁国忠
 - 责任编辑 朱 巍
 - 责任印制 彭志环
 - ◆ 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>
 - 三河市海波印务有限公司印刷
 - ◆ 开本: 800×1000 1/16
 - 印张: 16
 - 字数: 378千字 2016年6月第2版
 - 印数: 5 401—8 400册 2016年6月河北第1次印刷
 - 著作权合同登记号 图字: 01-2016-0260号
-

定价: 49.00元

读者服务热线: (010)51095186转600 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广字第 8052 号

版权声明

Authorized translation from the English language edition, entitled *Swift For Beginners* by Boisy G. Pitre, published by Pearson Education, Inc., publishing as Peachpit. Copyright © 2016 by Boisy G. Pitre.

All rights reserved. No part of this book may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying, recording or by any information storage retrieval system, without permission from Pearson Education, Inc.

Simplified Chinese-language edition copyright © 2016 by Posts & Telecom Press. All rights reserved.

本书中文简体字版由 Pearson Education Inc. 授权人民邮电出版社独家出版。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。

版权所有，侵权必究。

谨以此书献给这些女孩：Toni、Hope、Heidi、Lillian、Sophie 和 Belle。

致 谢

Peachpit 执行编辑 Cliff Colby 提议我编写《Swift 基础教程》第 2 版，鉴于以下几个原因，我欣然接受了提议。首先，Apple 对 Swift 语言做了很大的改进，有必要对第 1 版进行修订，以涵盖 Swift 2 和 Xcode 7 所做的改进；其次，第 1 版的出版团队非常杰出，我渴望与他们再次携手合作。

这个项目启动后不久，Cliff 离开 Peachpit 另谋高就，但离开前向我引见了新任执行编辑 Connie Jeung-Mills。Connie Jeung-Mills 召集了第 1 版编辑团队的成员——编辑 Robyn Thomas 和技术编辑 Steve Phillips，并邀请新人 Scout Festa 加入团队，以加强编辑力量。在本书的出版过程中，这些团队成员都扮演着不可或缺的重要角色，这里要感谢他们的协助。

在技术层面，我依然从众多朋友的作品中吸取了灵感。这些朋友都是 iOS 和 Mac OS 开发人员社区的作者，他们是 Chris Adamson、Bill Cheeseman、James Dempsey、Bill Dudney、Daniel Steinberg 和 Richard Warren。感谢 MacTech 杂志社的 Ed Marczak 和 Neil Ticktin 以及 CocoaConf 协调人 Dave Klein，感谢他们给我提供写作和演讲的机会。感谢 Dave et Ray's Camp Jam/Supper Club 的朋友，本书多个编码示例的灵感都来自他们。还要感谢 Troy Deville 将其游戏 Downhill Challenge 的代码贡献出来。

感谢苹果公司的员工开发并改善 Swift，推出了 Swift 2。仅发布一年后，Swift 就已相当完善；作为一门计算机语言，它如此年轻，却如此受欢迎，确实非比寻常。

最后，感谢我的家人，尤其是妻子 Toni 在我写作本书期间的耐心和鼓励。

前言

欢迎阅读本书！Swift 是苹果公司新推出的用于开发 iOS 和 Mac OS X 应用程序的语言，注定将成为移动和桌面领域首屈一指的计算机语言。作为一门新的计算机语言，Swift 犹如闪闪发亮的新车般充满诱惑：谁都想凑近瞧一瞧，踢踢它的轮胎，开着它去兜风。这可能就是你阅读本书的原因：你听说过 Swift，并决定一探究竟。

Swift 是一门易学易用的语言，这无疑是优点，相比于其前身 Objective-C 来说尤其如此，Objective-C 虽然功能强大，但学习起来更难。长期以来，苹果公司一直将 Objective-C 作为其平台的软件开发语言，但随着 Swift 的面世，情况正在发生变化。

针对的读者

本书是为初学者编写的。鉴于 Swift 面世时间极短，从某种意义上说我们都是初学者。然而，对很多人来说，Swift 是其学习的第一门或第二门语言，他们大多未接触过 Objective-C 以及 C 和 C++ 等相关语言。

读者最好对计算机语言有一定认识和经验，但本书也适合有足够学习欲望的新手。经验较丰富的开发人员可能发现前几章属于复习材料，类似休闲读物，因为其中介绍的概念在众多计算机语言中都有，但对于初学者来说，这些概念必须介绍。

如何使用本书

与其他同类图书一样，本书也最适合从头到尾地按顺序阅读，因为后续章节要求你已经掌握之前介绍的知识。然而，几乎每章的示例代码都自成一体。

本书篇幅适中，既涵盖了丰富的内容，又不会让读者不堪重负。书中包含大量的屏幕截图，让初学者能够全面了解 Swift 和 Xcode 工具集。

你将如何学习

学习 Swift 的最佳方式是使用它，而本书包含大量的代码和示例，始终将使用 Swift 作为重点。

每章都包含基于其中的概念编写的代码。Swift 提供了两种交互式环境，可供你用来测试概念以及加深对 Swift 本身的认识——REPL 和游乐场（playground）。在本书的第二部分，你将创

建两个简单而完备的应用程序：一个运行于 Mac OS X 系统下的贷款计算器和一个 iOS 记忆游戏；在最后一章，你将研究一个完整 2D 游戏的源代码，这个游戏使用了 Apple 的多种游戏开发技术。

对 Swift 概念的介绍贯穿本书始终，这包括类、函数、闭包等。建议你不要着急，慢慢地阅读每一章，并在必要时反复阅读，然后再进入下一章。

www.peachpit.com/swiftbeginners2 提供了本书源代码，你可下载每章的源代码。直接下载代码可节省大量的输入时间，但我深信应手工输入，这样做可获得仅阅读本书并依赖于下载的代码无法获得的洞见和认识。请花点时间输入所有的代码示例。

为清晰起见，代码和类名语言结构使用了等宽字体。

```
1> let candyJar = ["Peppermints", "Goody Bears", "Happy Ranchers"]
candyJar: [String] = 3 values {
  [0] = "Peppermints"
  [1] = "Goody Bears"
  [2] = "Happy Ranchers"
}
2>
```

对于 REPL 显示的错误消息，使用了粗体。

```
8> x = y
repl.swift:8:5: error: cannot assign a value of type 'Double'
→ to a value of type 'Int'
x = y
  ^

8>
```

“注意”提供了有关当前介绍的主题的额外信息。

注意 字典键并不一定是按字母顺序排列的，Swift 总是采用可最大限度地提高检索和访问效率的顺序来排列它们。

你将学到哪些知识

本书的终极目标是，介绍如何使用 Swift 代码来表达思想。等阅读到本书最后一页时，你将对 Swift 的功能有深入认识，并具备开始编写应用程序所需的技能。本书第二部分提供了 iOS 和 Mac OS X 应用程序示例。

本书并非包罗万象的 Swift 编程语言综合指南，要全面了解 Swift，苹果公司的官方文档是最佳的资源。本书的重点是 Swift 语言本身，但为帮助理解示例，简要地介绍了相关的 Cocoa 和 CocoaTouch 框架。

欢迎来到 Swift 世界

Swift 是苹果公司新推出的一款有趣而易学的计算机语言。掌握本书介绍的知识后，你将能够开始编写 iOS 和 Mac OS X 应用程序。要开始学习 Swift，Xcode 集成开发环境（IDE，Integrated Development Environment）是必须有的主要工具。Xcode 提供了 Swift 编译器以及 iOS 和 Mac OS X 软件开发包（SDK，Software Development Kit），这些 SDK 包含为你开发的应用程序提供支持的基础设施。

技术

在这次 Swift 学习之旅中，你将领略下面的风景。

Swift 2

Swift 2 是你将在本书中学习的语言，这是一款从头打造的现代语言，功能强大且易于学习。苹果公司已将其作为日益增长的 iOS、watchOS、tvOS 和 Mac OS X 应用程序开发语言。



Xcode 7

Xcode 7 是苹果公司首要的应用程序开发环境，提供了编辑器、调试器、项目管理和编译器，其中编译器用于将 Swift 代码转换为能够运行的代码。Xcode 可从 Apple Mac App Store 下载。



LLVM

在 Xcode 中，LLVM 在幕后工作，这种编译器技术让 Swift 语言变得优雅，并将 Swift 代码转换为 Apple 设备处理器能够运行的机器码。



REPL

REPL（Read-Eval-Print-Loop）是一个命令行工具，可用于快速尝试 Swift 代码。在 Mac OS X 中，可在应用程序 Terminal 中运行它。

```

Terminal — lldb — 81 x29
23> for loopCounter in 0..  
24.     print("value at index \(loopCounter) is \(numbersArray[loopCounter])")  
25. }  
value at index 0 is 11  
value at index 1 is 22  
value at index 2 is 33  
value at index 3 is 44  
value at index 4 is 55  
value at index 5 is 66  
value at index 6 is 77  
value at index 7 is 88  
value at index 8 is 99  
26> for loopCounter = 0; loopCounter < 9; loopCounter = loopCounter + 2 {  
27.     print("value at index \(loopCounter) is \(numbersArray[loopCounter])")  
28. }  
value at index 0 is 11  
value at index 2 is 33  
value at index 4 is 55  
value at index 6 is 77  
value at index 8 is 99  
29> for loopCounter = 8; loopCounter >= 0; loopCounter = loopCounter - 2 {  
30.     print("value at index \(loopCounter) is \(numbersArray[loopCounter])")  
31. }  
value at index 8 is 99  
value at index 6 is 77  
value at index 4 is 55  
value at index 2 is 33  
value at index 0 is 11  
32>

```

REPL

游乐场

Xcode 游乐场提供了交互性和实时的结果，在学习 Swift 时非常适合用来尝试代码。

```

Ready | Today at 9:04 PM
Chapter 4
1 //: Playground - noun: a place where people can play
2
3 import Cocoa
4
5 var str = "Chapter 4 Playground"                                "Chapter 4 Playground"
6
7 func fahrenheitToCelsius(fahrenheitValue : Double) -> Double {
8     var result : Double
9
10    result = (((fahrenheitValue - 32) * 5) / 9)                    31.22222222222222
11
12    return result;                                                31.22222222222222
13 }
14
15 var outdoorTemperatureInFahrenheit = 88.2                      88.2
16 var outdoorTemperatureInCelsius = fahrenheitToCelsius
17    (outdoorTemperatureInFahrenheit)                             31.22222222222222
18
19 func celsiusToFahrenheit(celsiusValue : Double) -> Double {
20     var result : Double
21
22     result = (((celsiusValue * 9) / 5) + 32)                      88.2
23
24     return result                                                88.2
25 }
26
27 outdoorTemperatureInFahrenheit = celsiusToFahrenheit           88.2
28    (outdoorTemperatureInCelsius)
29
30 func buildASentenceUsingSubject(subject : String, verb : String, noun :
31    String) -> String {
32     return subject + " " + verb + " " + noun + "!"               (2 times)
33 }
34
35 buildASentenceUsingSubject("Swift", verb: "is", noun: "cool")    "Swift is cool!"
36 buildASentenceUsingSubject("I", verb: "love", noun: "languages") "I love languages!"
37

```

游乐场

目 录

第一部分 基础知识

第 1 章 Swift 简介	2
1.1 革命性的改良	2
1.2 准备工作	3
1.2.1 专业工具	3
1.2.2 与 Swift 交互	3
1.3 准备出发	4
1.4 开始探索 Swift	6
1.4.1 帮助和退出	6
1.4.2 Hello World	7
1.5 声明的威力	7
1.6 常量	9
1.7 类型	10
1.7.1 检查上限和下限	11
1.7.2 类型转换	11
1.7.3 显式地声明类型	13
1.8 字符串	13
1.8.1 字符串拼接	14
1.8.2 Character 类型	14
1.9 数学运算符	15
1.9.1 表达式	15
1.9.2 混用不同的数值类型	16
1.9.3 数值表示	16
1.10 布尔类型	17
1.11 轻松显示	18
1.12 使用类型别名	19
1.13 使用元组将数据编组	20
1.14 可选类型	21
1.15 小结	22

第 2 章 使用集合	23
2.1 糖果罐	23
2.1.1 数组中所有元素的类型都必须相同	26
2.1.2 增长数组	26
2.1.3 替换和删除值	27
2.1.4 将值插入到指定位置	28
2.1.5 合并数组	29
2.2 字典	30
2.2.1 查找条目	31
2.2.2 添加条目	32
2.2.3 更新条目	33
2.2.4 删除条目	33
2.3 数组的数组	34
2.4 创建空数组和空字典	36
2.4.1 空数组	36
2.4.2 空字典	37
2.5 迭代集合	37
2.5.1 迭代数组	38
2.5.2 迭代字典	39
2.6 小结	40
第 3 章 流程控制	41
3.1 for 循环	41
3.1.1 计数	41
3.1.2 包含还是不包含结束数字	42
3.1.3 老式 for 循环	43
3.1.4 简写	44
3.2 游乐场	45
3.3 决策	48
3.3.1 if 语句	49

3.3.2 检查多个条件	52	5.3.4 便利初始化方法	104
3.3.3 switch 语句	53	5.4 枚举	106
3.3.4 while 循环	57	5.5 结构	109
3.3.5 检查代码	59	5.6 值类型和引用类型	110
3.3.6 提早结束循环	62	5.7 小结	112
3.4 小结	62	第 6 章 使用协议和扩展进行规范化	113
第 4 章 编写函数和闭包	63	6.1 遵循协议	113
4.1 函数	63	6.1.1 类还是协议	113
4.1.1 使用 Swift 编写函数	64	6.1.2 协议并非只能定义方法	115
4.1.2 执行函数	65	6.1.3 遵循多个协议	117
4.1.3 参数并非只能是数字	66	6.1.4 协议也可继承	118
4.1.4 可变参数	67	6.1.5 委托	119
4.1.5 函数是一级对象	70	6.2 扩展	122
4.1.6 从函数返回函数	71	6.2.1 扩展基本类型	123
4.1.7 嵌套函数	74	6.2.2 在扩展中使用闭包	127
4.1.8 默认参数	76	6.3 小结	129
4.1.9 函数名包含哪些内容	77	第二部分 使用 Swift 开发软件	
4.1.10 清晰程度	78	第 7 章 使用 Xcode	132
4.1.11 用不用参数名	79	7.1 Xcode 简史	132
4.1.12 变量参数	79	7.2 创建第一个 Swift 项目	133
4.1.13 inout 参数	81	7.3 Xcode 界面	135
4.2 闭包	82	7.3.1 与 Xcode 窗口交互	136
4.3 小结	84	7.3.2 运行应用程序	138
4.4 类	84	7.4 开发应用程序	139
第 5 章 使用类和结构组织代码	85	7.4.1 腾出空间	139
5.1 对象无处不在	85	7.4.2 创建界面	141
5.2 Swift 对象是使用类定义的	86	7.4.3 美化	143
5.2.1 定义类	86	7.4.4 编写代码	145
5.2.2 创建对象	88	7.4.5 建立连接	149
5.2.3 开门和关门	88	7.5 小结	151
5.2.4 锁门和开锁	89	第 8 章 改进应用程序	152
5.2.5 查看属性	92	8.1 细节很重要	152
5.2.6 门应是各式各样的	92	8.1.1 显示金额	152
5.2.7 修改颜色	94	8.1.2 再谈可选类型	154
5.3 继承	95	8.1.3 可选类型拆封	155
5.3.1 创建基类	96	8.1.4 美化	155
5.3.2 创建子类	98		
5.3.3 实例化子类	100		

8.1.5 另一种格式设置方法	156	10.1.2 引用计数	202
8.2 计算复利	160	10.1.3 引用循环	203
8.2.1 连接起来	162	10.1.4 演示引用循环	203
8.2.2 测试	165	10.1.5 编写测试代码	204
8.3 调试	165	10.1.6 断开引用循环	206
8.3.1 bug 在哪里	165	10.1.7 闭包中的引用循环	207
8.3.2 断点	166	10.1.8 感恩	209
8.3.3 复杂的复利计算	169	10.2 逻辑运算符	209
8.4 测试的价值	170	10.2.1 逻辑非	209
8.4.1 单元测试	170	10.2.2 逻辑与	210
8.4.2 编写测试	171	10.2.3 逻辑或	210
8.4.3 如果测试未通过	173	10.3 泛型	211
8.4.4 始终运行的测试	174	10.4 运算符重载	212
8.5 小结	175	10.5 相等和相同	214
第 9 章 Swift 移动开发	176	10.6 错误处理	216
9.1 移动设备和台式机	176	10.6.1 引发错误	216
9.2 挑战记忆力	176	10.6.2 捕获错误	217
9.2.1 考虑玩法	177	10.7 Swift 脚本编程	219
9.2.2 设计 UI	177	10.7.1 创建脚本	219
9.3 创建项目	178	10.7.2 设置权限	221
9.4 创建用户界面	180	10.7.3 执行脚本	221
9.4.1 创建按钮	181	10.7.4 工作原理	222
9.4.2 在模拟器中运行	183	10.8 获取帮助	223
9.4.3 设置约束	184	10.9 小结	225
9.5 MVC	187	第 11 章 高山滑雪	226
9.6 编写游戏代码	187	11.1 游戏开发技术	226
9.6.1 类	190	11.1.1 GameKit	226
9.6.2 枚举	191	11.1.2 SpriteKit	227
9.6.3 视图对象	191	11.2 始于构思	227
9.6.4 模型对象	191	11.2.1 高山滑雪	227
9.6.5 可重写的方法	192	11.2.2 社交功能	227
9.6.6 游戏的方法	193	11.3 出发	227
9.6.7 处理输赢	196	11.3.1 怎么玩	228
9.7 回到故事板	198	11.3.2 玩一玩	228
9.8 开玩	200	11.4 研究这个项目的组织结构	231
第 10 章 成为专家	201	11.4.1 类	231
10.1 Swift 内存管理	201	11.4.2 素材	232
10.1.1 值和引用	201	11.4.3 场景	232
		11.5 探索源代码	232

11.5.1	场景 Home	232	11.6.1	研究苹果公司提供的框架	241
11.5.2	场景 Game	235	11.6.2	加入苹果开发者计划	242
11.5.3	游戏视图控制器	239	11.6.3	成为社区的一分子	242
11.5.4	全面了解	241	11.6.4	活到老学到老	242
11.6	独闯江湖	241	11.6.5	一路平安	242

Part 1

第一部分

基础知识

无论学习什么新东西，都得从基础开始，Swift 语言也不例外。Swift 是一款卓越的新语言，这部分介绍了使用它编写一流应用程序所需的全部知识！

本 部 分 内 容

- 第 1 章 Swift 简介
- 第 2 章 使用集合
- 第 3 章 流程控制
- 第 4 章 编写函数和闭包
- 第 5 章 使用类和结构组织代码
- 第 6 章 使用协议和扩展进行规范化



欢迎来到Swift这个美丽的新世界。Swift是在2014年苹果全球开发者大会上推出的，仅仅一年后，它就成了一门功能强大的新编程语言。由于苹果的倡导和开发人员的积极响应，Swift被广泛采纳，成了开发iOS、watchOS和Mac应用程序的主流语言。Swift不但功能强大，还简单易学，你在不知不觉间就能编写出简单应用。

Swift提供了一些编写代码的新方式，比功能强大而著名的前身Objective-C容易理解得多。Swift向开发人员提供了全新而有趣的方式表达，其功能学习起来也很有趣。

Swift作为计算机语言虽已推出一年之久，但依然是一门新语言，苹果公司很可能对其进行修改和增补。从未有一种计算机语言像Swift这样，在即将修改和修订前能获得如此高的曝光度和采纳度，这都要归功于Swift的创新带来的刺激。

1.1 革命性的改良

语言是分享、交流和传达信息的工具，人类通过它向朋友、家人和同事表达自己的意图。与计算机系统交流也需要通过计算机语言。

与人类的语言一样，计算机语言也非新鲜事物，事实上，它们以这样或那样的形式存在了很多年。计算机语言的目的始终是让人类能够与计算机交流，命令它执行特定的操作。

不断发展变化的是计算机语言本身。早期的计算机开拓者意识到，以0和1的方式向计算机发指令既繁琐又容易出错。一路上人们始终在不断努力，旨在在语言语法的丰富性和处理与解读它所需的计算能力之间寻求平衡，最终诸如C和C++语言在争夺现代计算机应用程序通用语言之战中取得了胜利。

在C和C++被广泛接受，得以用于主要的计算平台的同时，苹果携Objective-C给这场盛宴带来了清新之风。Objective-C是一款建立在C语言基础之上的面向对象语言。苹果生态系统由Macintosh计算机和iOS设备构成，在为该生态系统开发应用程序中，Objective-C多年来始终发挥着中流砥柱的作用。

Objective-C虽然功能强大而优雅，但也存在着其前身——C语言遗留下来的包袱。对于熟悉C语言的人来说，这根本就不是什么问题，但近年来大量新开发人员进入Mac和iOS平台，他们渴望更容易理解和使用的新语言。

为满足这种需求，并降低进入门槛，苹果公司推出了Swift。使用它编写应用程序容易得多，向应用程序发出指令也更加简便。

1.2 准备工作

你可能会问，要学习Swift需要满足哪些条件呢？实际上，开始阅读本书就迈出了学习Swift的第一步。学习新的计算机语言可能令人望而却步，这正是笔者为Swift初学者编写本书的原因所在。如果你是Swift新手，本书正是为你编写的；如果你从未使用过C、C++和Objective-C，本书也适合你阅读。即便你是经验丰富的开发人员，熟悉前面提及的各种语言，本书也可帮助你快速掌握Swift。

虽然并非绝对必要，但熟悉或大致了解其他编程语言对阅读本书很有帮助。本书不介绍如何编程，也不提供有关软件开发的基本知识，而假定你对计算机语言的基本概念有一定认识，因此你必须对计算机语言有所了解。

虽然如此，本书将向你提供尽可能多的帮助：详尽地解释新引入的术语，并对概念做尽可能清晰的阐述。

1.2.1 专业工具

至此，你做好了学习Swift的心理准备。这很好！但首先得将学习用品准备妥当。回想一下上小学时的情形吧，开学前父母都会收到所需学习用品清单：笔记本、剪刀、美术纸、胶水、2号铅笔等。当然，阅读本书不需要这些东西，但要学习Swift，必须有合适的专业工具。

首先，强烈建议你以交互方式运行本书列出的代码。为此，需要一台运行OS X 10.10（Yosemite）或OS X 10.11（El Capitan）的Macintosh计算机；还需要Xcode 7，它提供了Swift编译器和配套环境。最重要的是，你需要加入苹果开发者计划，这样才能充分利用El Capitan和Xcode 7。如果你还未加入苹果开发者计划，可访问<https://developer.apple.com/programs>，其中提供了有关如何加入该计划的完整信息。

将Xcode 7下载并安装到Mac计算机后，便可以开始学习Swift了。

1.2.2 与 Swift 交互

首先，我们将通过一个有趣的交互式环境——REPL，来探索Swift。REPL是Read-Eval-Print-Loop（读取-执行-输出-循环）的首字母缩写，这指出了这个工具的特征：它读取指令、执行指令、输出结果，再重新开始。

事实上，这种交互性是Swift有别于C和Objective-C等众多编译型语言的特点之一。如果你使用过Ruby或Python等提供了REPL环境的脚本语言，就知道这并非什么新东西，但对编译型语言来说，这种理念还是很新颖的。只要问问C、C++或Objective-C开发人员就知道，他们很多时候都希望能够直接运行代码，而不用创建包含调试语句的源代码文件，再编译、运行并查看结果。Swift REPL的优点在于，它让上述重复而漫长的工作流程一去不复返了。

这种交互性带来的另一大好处是，它让学习新语言这种原本艰难的任务变得容易多了。你不用再学习一系列复杂的编译工具，也无需了解集成开发环境的细枝末节，只需将全部精力都放在新语言本身上。事实上，本书前半部分将探索、测试、细究Swift的方方面面，你将很快发现，以这种交互方式学习能够更快地理解Swift语言本身。

不需要运行阶段环境就能实时运行代码，一开始这可能让人感觉怪怪的，但很快你就会喜欢它提供的即时结果。事实上，REPL会让有些人想起以前的岁月：在家用计算机革命的早期，BASIC等解释型语言就提供了这种交互性。真是从终点又回到了起点。

1.3 准备出发

已下载了Xcode 7？这很好，但请暂时将它抛在脑后吧。事实上，我鼓励你去探索Xcode 7及其新特性，但接下来的几章将把注意力完全放在Terminal^①中的REPL上。

如果你以前没有运行过Terminal应用程序，也不用担心。在Mac计算机中，它位于文件夹Applications/Utilities下。要运行它，最简单的方式是单击图标Spotlight，再输入Terminal，如图1-1所示。

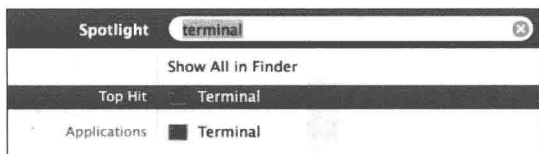


图1-1 使用Spotlight来查找应用程序Terminal

另一种方法是，单击Dock中的Finder图标，再选择菜单Go>Utilities^②，如图1-2所示。



图1-2 Finder菜单栏中的Go菜单

① 在简体中文版Mac操作系统中被称为“终端”，故本书有时也会用终端来代指它。——编者注

② 在简体中文版Mac操作系统中被称为“实用工具”。——编者注