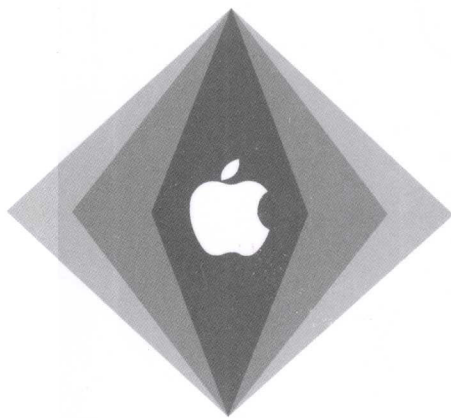




Swift 3

开发指南

博为峰 51Code 教研组◎组编



全面讲解了 Swift 语言新版本的新功能、新应用

用贴近实战的案例全面讲解 Swift 的语法,帮助读者透彻掌握新的 Swift 语言。

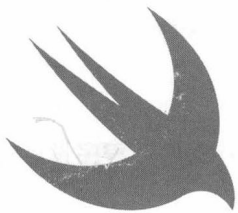
综合项目将书中语法和开发技术巧妙结合在一起,便于学以致用。



中国工信出版集团

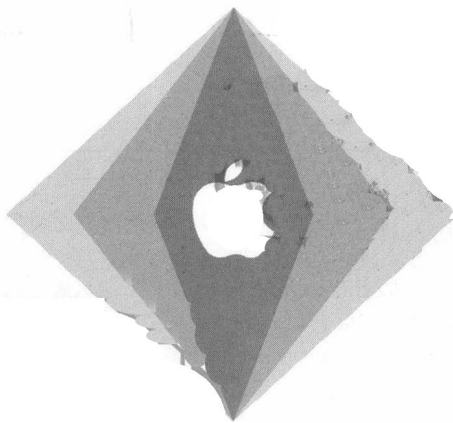


人民邮电出版社
POSTS & TELECOM PRESS



Swift 3 开发指南

—— 博为峰 51Code 教研组 © 组编 ——



人民邮电出版社

北 京

图书在版编目(CIP)数据

Swift 3开发指南 / 博为峰51Code教研组组编. --
北京: 人民邮电出版社, 2017.6
ISBN 978-7-115-45387-7

I. ①S… II. ①博… III. ①程序语言—程序设计—
指南 IV. ①TP312-62

中国版本图书馆CIP数据核字(2017)第083673号

内 容 提 要

本书是基于Xcode 8编写而成的,通过大量通俗易懂的案例全面讲解了Swift语言开发的相关内容。书中包含常量与变量、基本数据类型,以及Swift语言中的新数据类型—元组型和可选型的内容。同时,本书中还讲解了运算符和表达式、流程控制语句、字符和字符串、collection类型、函数和闭包。此外,Swift面向对象的枚举、结构体、类,以及内存管理、协议与抽象类型、错误处理、链式编程等内容本书也有所涉及。本书最后的两章讲解了实战项目开发的内容,供读者学以致用。

本书适合iOS开发者、其他移动开发平台开发者,或者有兴趣从事iOS开发的读者阅读,也适合作为大专院校计算机专业的师生用书和培训学校的教材。

◆ 组 编 博为峰 51Code 教研组

责任编辑 张 涛

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>

北京市艺辉印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 13.5

字数: 323千字

印数: 1—2 000册

2017年6月第1版

2017年6月北京第1次印刷

定价: 49.00元

读者服务热线: (010)81055410 印装质量热线: (010)81055316

反盗版热线: (010)81055315

广告经营许可证: 京东工商广登字 20170147号

前 言

Swift 语言从发布至今，版本在不断地更新。Swift 3 是苹果公司在 2016 年 6 月发布的 Swift 语言的最新版本，它是 Swift 语言的一个重大的里程碑。在本书编写时，Xcode 8 仍处于测试版阶段，因此，本书内容基于 Xcode 8.0 Beta 版编写。

本书内容

我们团队编写本书的目的是满足从事 iOS 开发的广大读者学习 Swift 3 语言的需要。同时，已有 Objective-C 开发经验的人员通过阅读本书能够快速转型到使用 Swift 语言开发 iOS 应用。

本书共分为五大部分。

第一部分为基础语法篇，共 7 章，主要介绍 Swift 语言的基础知识。

第 1 章主要对 Swift 语言进行简单的介绍，Xcode 的安装、卸载以及使用，如何阅读 Swift 开发文档，并介绍了如何使用 Xcode 的 Playground 编写和运行 Swift 程序代码等内容。

第 2 章主要介绍 Swift 语言的基本语法，包括标识符、关键字、表达式、语句、注释、常量和变量，以及 Swift 的基本数据类型，例如整型、浮点型、元组型、可选类型等内容。

第 3 章主要介绍 Swift 语言的运算符和表达式，包括算术运算符、赋值运算符、关系运算符、逻辑运算符、条件运算符等内容。

第 4 章主要介绍 Swift 语言的流程控制语句，包括循环语句（for in、while、repeat-while），分支语句（if、switch），控制转移语句（continue、break、fallthrough），语句嵌套等内容。

第 5 章主要介绍 Swift 语言的字符和字符串，包括获取字符串的长度、字符串的比较，字符串前缀后缀，字符串大小写转换，字符串的插入、添加、删除、提取、替换、遍历等常见操作，以及 String 与 NSString 的关系等内容。

第 6 章主要介绍 Swift 语言的 Collection 类型，包括数组、字典和集合类型等内容。

第 7 章主要介绍 Swift 语言的函数和闭包，包括函数的声明和调用、参数、返回值，函数类型、泛型和泛型函数，闭包的概念、表达式，尾随闭包和捕获值等内容。

第二部分为面向对象篇，共 4 章，主要介绍 Swift 语言面向对象的相关知识。

第 8 章主要介绍 Swift 语言的面向对象编程、枚举的定义和方法、值枚举和类型枚举等内容。

第 9 章主要介绍 Swift 的结构体，包括结构体的定义、属性、方法、构造器，以及结构体嵌套、可选链、扩展等内容。

第 10 章主要介绍 Swift 的类，包括类和结构体的区别、类的属性和方法、继承多态、重载、构造、类型检测，以及类对象的内存管理等内容。

第 11 章主要介绍 Swift 的协议与抽象类型，包括声明协议和遵守协议、协议的属性和方

法、抽象类型等内容。

第三部分为错误处理篇，共 1 章，主要介绍 Swift 语言在实际开发应用中如何进行错误处理。

第 12 章主要介绍在使用 Swift 语言进行实际开发中遇到错误，如何来捕捉和处理错误等内容。

第四部分为 Swift 与 Objective-C 对比篇，共两章，主要介绍如何在 Swift 项目中调用 Objective-C 代码，以及 Swift 语言的优势。

第 13 章主要介绍 Swift 与 Objective-C 的区别，以及如何在 Swift 项目中调用 Objective-C 代码等内容。

第 14 章主要介绍 Swift 可以支持链式编程等优势。

第五部分为项目实战篇，共两章，主要介绍如何使用 Swift 语言开发汽车商城项目。

第 15 章主要介绍 iOS 应用开发的一般流程，使用纯 Swift 代码来完成汽车商城项目等内容。

第 16 章主要介绍对 iOS 应用开发中的项目进行测试，我们以汽车商城项目为例，对该项目进行测试。

读者可以扫描如下二维码，观看本书配套视频。



扫码观看本书配套视频

致谢

首先，感谢您选择本书，希望本书能够切实帮助您解决在实际开发中遇到的一些困难。

此外，感谢我们团队每一位成员的努力，这让我们用短短几个月的时间完成了本书的编写工作。

由于时间仓促，书中难免存在不妥之处。如果读者在使用本书时，发现差错或遇到问题，敬请批评指正，并将指正内容发至本书编辑邮箱 zhangshuang@ptpress.com.cn。

博为峰 51Code 教研组

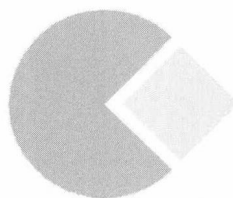
目 录

第一部分 基础语法篇	1	2.5 思考练习	23
第 1 章 Hello Swift	3	第 3 章 常见运算符和表达式	24
1.1 Swift 简介	3	3.1 算术运算符	24
1.1.1 Swift 语言简介	3	3.2 赋值运算符	24
1.1.2 Swift 语言开发平台	3	3.3 关系运算符	25
1.1.3 Swift 语言开发工具	3	3.4 逻辑运算符	25
1.1.4 Swift 语言开发文档	5	3.5 三元运算符	25
1.2 Swift 语言编程体验	7	3.6 Nil Coalescing 运算符	26
1.2.1 Hello, 51Code 程序	7	3.7 复合表达式	26
1.2.2 演练利器 Playground	9	3.8 本章小结	27
1.3 本章小结	11	3.9 思考练习	27
1.4 思考练习	11	第 4 章 流程控制语句	28
第 2 章 变量和基本数据类型	12	4.1 循环语句	28
2.1 Swift 语言基础语法	12	4.1.1 for-in 语句	28
2.1.1 标识符	12	4.1.2 while 语句	29
2.1.2 关键字	12	4.1.3 repeate-while 语句	30
2.1.3 表达式	13	4.2 分支语句	30
2.1.4 语句	13	4.2.1 if 语句	30
2.2 常量和变量	14	4.2.2 switch 语句	32
2.2.1 常量	14	4.2.3 在 switch 语句中使用范围 匹配	34
2.2.2 变量	15	4.2.4 在 switch 语句中比较元 组型	34
2.3 数据类型	15	4.3 控制转移语句	35
2.3.1 布尔型	15	4.3.1 continue 语句	35
2.3.2 整型	16	4.3.2 break 语句	37
2.3.3 浮点型	17	4.3.3 fallthrough 语句	38
2.3.4 字符串型	18	4.4 流程嵌套	38
2.3.5 数据类型的转换	18	4.5 本章小结	39
2.3.6 元组型	20	4.6 思考练习	39
2.3.7 可选型	21		
2.4 本章小结	23		

第 5 章 字符和字符串	40	第 7 章 函数和闭包	62
5.1 Swift 语言中的字符	40	7.1 函数的声明和调用	62
5.2 字符串常见操作	41	7.2 函数参数	63
5.2.1 字符串长度	41	7.2.1 无参函数	63
5.2.2 字符串比较	41	7.2.2 含参函数	63
5.2.3 字符串前缀和后缀判断	42	7.2.3 函数参数标签和参数名	63
5.2.4 字符串的字符大小写转换	42	7.2.4 参数默认值	64
5.2.5 字符串插入	43	7.2.5 可变参数	65
5.2.6 字符串添加	43	7.2.6 参数的传递引用	65
5.2.7 字符串删除	43	7.3 函数返回值	66
5.2.8 字符串提取	44	7.3.1 无返回值函数	66
5.2.9 字符串替换	45	7.3.2 有返回值函数	67
5.2.10 遍历字符串	45	7.4 函数类型	68
5.3 String 与 NSString 关系	46	7.4.1 作为参数类型使用	68
5.4 本章小结	47	7.4.2 作为返回值类型使用	69
5.5 思考练习	47	7.5 内嵌函数	69
第 6 章 Collection 类型	48	7.6 泛型和泛型函数	70
6.1 数组	48	7.6.1 泛型要解决的问题	70
6.1.1 数组的创建	48	7.6.2 泛型函数	70
6.1.2 数组的访问	49	7.7 闭包的概念	71
6.1.3 数组的编辑	51	7.8 闭包表达式	73
6.1.4 数组的复制	52	7.8.1 类型推断简化	73
6.1.5 Array 与 NSArray 的 关系	53	7.8.2 省略 return 关键字	74
6.2 字典	53	7.8.3 使用位置参数简化闭包 书写	75
6.2.1 字典的创建	53	7.8.4 使用闭包返回值	76
6.2.2 字典的访问	54	7.9 尾随闭包	76
6.2.3 字典的编辑	55	7.10 捕获上下文的常量和变量	77
6.2.4 字典的复制	57	7.11 逃逸闭包	78
6.2.5 Dictionary 与 NSDictionary 的关系	57	7.12 自动闭包	79
6.3 集合	58	7.13 本章小结	80
6.3.1 集合的创建	58	7.14 思考练习	80
6.3.2 集合的访问	58	第二部分 面向对象篇	81
6.3.3 集合的编辑	59	第 8 章 枚举	83
6.3.4 集合的关系	60	8.1 Swift 的面向对象	83
6.4 本章小结	61	8.2 枚举类型	84
6.5 思考练习	61	8.2.1 枚举定义	84
		8.2.2 枚举的方法	84

8.3 值枚举	86	10.2.4 类的属性监听	112
8.3.1 成员值	86	10.2.5 类的静态属性	113
8.3.2 原始值	87	10.3 类的方法	113
8.3.3 哈希值	88	10.3.1 类的实例方法	114
8.4 类型枚举	89	10.3.2 类的静态方法	114
8.5 本章小结	90	10.4 类的继承和多态	115
8.6 思考练习	90	10.4.1 类的继承	115
第9章 结构体	91	10.4.2 类的重写	116
9.1 结构体的定义	91	10.4.3 类的重载	119
9.2 结构体属性	91	10.5 类的构造和析构	120
9.2.1 实例属性	92	10.5.1 类的构造	120
9.2.2 懒加载实例属性	92	10.5.2 类的析构	123
9.2.3 计算属性	93	10.6 类的类型检测和转换	123
9.2.4 静态属性	94	10.6.1 类型检测	124
9.3 结构体的属性监听	95	10.6.2 类型转换	125
9.4 结构体的方法	96	10.6.3 Any 和 AnyObject 转换	126
9.4.1 实例方法	97	10.7 类对象的内存管理	127
9.4.2 静态方法	97	10.7.1 内存管理概述	127
9.5 下标	98	10.7.2 强引用循环	128
9.6 结构体的构造器	99	10.7.3 打破强引用循环	129
9.6.1 默认构造器	99	10.7.4 闭包中的强引用循环	132
9.6.2 构造器参数	100	10.8 本章小结	134
9.6.3 指定构造器和便利构造器	101	10.9 思考练习	134
9.7 结构体嵌套	102	第11章 协议与抽象类型	135
9.8 可选链	102	11.1 协议	135
9.9 扩展	104	11.1.1 声明和遵守协议	135
9.9.1 声明扩展	105	11.1.2 协议属性	136
9.9.2 扩展计算属性	105	11.1.3 协议方法	137
9.9.3 扩展方法	106	11.2 抽象类型	139
9.9.4 扩展构造器	107	11.3 扩展中声明协议	141
9.10 本章小结	107	11.4 本章小结	142
9.11 思考练习	107	11.5 思考练习	142
第10章 类	108	第三部分 错误处理篇	143
10.1 类和结构体的区别	108	第12章 错误处理	145
10.2 类的属性	110	12.1 错误抛出	145
10.2.1 实例属性	111	12.2 错误的捕捉和处理	147
10.2.2 懒加载实例属性	111	12.3 错误与可选值	148
10.2.3 计算属性	111		

12.4	拦截错误传导	148	15.2.2	创建项目	165
12.5	收尾操作	148	15.2.3	搭建架构	167
12.6	本章小结	149	15.3	业务逻辑实现	168
12.7	思考练习	149	15.3.1	启动页模块	169
第四部分	Swift 与 Objective-C 对比篇	151	15.3.2	首页模块	171
第 13 章	Swift 与 Objective-C	153	15.3.3	地图模块	181
13.1	Swift 与 Objective-C 对比	153	15.3.4	发现模块	184
13.2	Swift 工程中调用 Objective-C	153	15.4	本章小结	195
13.3	本章小结	155	15.5	思考练习	196
13.4	思考练习	155	第 16 章	iOS 应用开发的测试	197
第 14 章	链式编程	156	16.1	iOS 测试框架	197
14.1	链式编程	156	16.2	XCTest 测试框架	197
14.2	链式编程的应用	157	16.2.1	添加 XCTest 测试框架	197
14.3	本章小结	159	16.2.2	XCTest 测试方法	200
14.4	思考练习	159	16.3	使用 XCTest 进行测试	201
第五部分	项目实战篇	161	16.3.1	常用测试工具	201
第 15 章	Swift 项目实战——汽车商城	163	16.3.2	单元测试	202
15.1	项目需求分析	163	16.3.3	UI 测试	203
15.1.1	项目功能需求	163	16.4	本章小结	204
15.1.2	项目界面设计	164	16.5	思考练习	204
15.2	项目架构搭建	165	附录		205
15.2.1	架构设计	165			



第一部分

基础语法篇

- 第 1 章 Hello Swift
- 第 2 章 变量和基本数据类型
- 第 3 章 常见运算符和表达式
- 第 4 章 流程控制语句
- 第 5 章 字符和字符串
- 第 6 章 Collection 类型
- 第 7 章 函数和闭包

第1章 Hello Swift

1.1 Swift 简介

1.1.1 Swift 语言简介

Swift 是一种新的编程语言，用于编写 iOS、OS X 和 watch OS 应用程序。Swift 采用安全的编程模式并添加了很多新特性，这将使得编程更简单、灵活，也更有乐趣。基于成熟而且倍受喜爱的 Cocoa 和 Cocoa Touch 框架的 Swift 将重新定义软件开发。

Swift 的开发从很久之前就开始了。为了给 Swift 打好基础，苹果公司改进了编译器、调试器和框架结构。我们使用自动引用计数（Automatic Reference Counting，ARC）来简化内存管理，在 Foundation 和 Cocoa 的基础上构建现代化和标准化的框架栈。

Swift 对于初学者来说也很友好。它是第一个既满足工业标准，又像脚本语言一样充满表现力和趣味的语言。它支持代码预览，这个革命性的特性可以允许程序员在不编译和运行应用程序的前提下运行 Swift 代码，并实时查看结果。

Swift 结合了现代编程语言的精华和工程师的智慧。编译器对性能进行了优化，编程语言对开发进行了优化，两者互不干扰，可谓“鱼与熊掌兼得”。Swift 既可以用于开发“hello, world”这样的简单程序，也可以用于开发一套完整的操作系统。Swift 是编写 iOS、OS X 和 watch OS 应用程序的极佳语言，并将伴随着新的特性和功能持续演进。

1.1.2 Swift 语言开发平台

Swift 语言开发平台基于 Mac 操作系统。Mac 系统由苹果公司自行研发，是基于 UNIX 内核的首个在商用领域成功的图形用户界面操作系统。一般情况下，在普通 PC 上无法安装操作 Mac 系统。Mac 系统已经到了 OS 10，代号为 Mac OS X（X 为 10 的罗马数字写法）。Mac 系统非常可靠，它的许多特点和服务都体现了苹果公司的理念。现行的最新的系统版本是 OS X 10.11.5，本书中的代码都是基于 Mac 系统 OS X 10.11.5。

2016 年 6 月召开的苹果开发者大会正式将 Mac 操作系统 OS X 更名为 Mac OS。

1.1.3 Swift 语言开发工具

Xcode 是苹果公司向开发人员提供的集成开发环境，也是用于开发 Mac OS X 和 iOS 应用

程序最快捷的方式。Xcode 前身来自 Next 的 Project Builder。

Mac OS X 和 iOS 开发的主要工具是 Xcode。在 Xcode 3.1 发布后, Xcode 成为 iPhone 软件开发工具包的开发环境。Xcode 有正式版和 Beat 版(测试版)。我们可以从 App Store 直接下载并使用正式版, 其性能比较稳定, 测试版不能在 App Store 中下载。如果想下载使用 Xcode beat 版, 我们可以在苹果开发者官网下载。在 App Store 和苹果开发者官网下载 Xcode 时都需要注册苹果开发者账号。

Xcode6 及其以后的版本支持用 Swift 语言开发 Mac OS X 和 iOS。Xcode6 整合了苹果公司在苹果开发者大会上发布的新语言 Swift 1.0 版本, 目前, Swift 语言已经更新到了 3.1 版本。本书中的代码都是基于 Swift 3.1 编写的, 开发环境为 Xcode 8.2.1 版。下面我们来学习如何安装和卸载 Xcode。

1. Xcode 的安装

Xcode 必须安装在 Mac OS 系统上, Xcode 的版本与 Mac OS 系统的版本有着严格的对应关系。我们使用的 Xcode 8.2.1 必须要在 Mac OS 10.11.5 及其以上的系统中使用。

我们可以在苹果开发者官网上下载 Xcode 8.2.1 版。打开浏览器, 输入网址, 进入到如图 1-1 所示的最新 Xcode 下载入口。

单击首页下方的 Develop 栏下的“Downloads”, 下载最新的 Xcode。

如果你没有登录, 单击“Downloads”, 此时会弹出一个如图 1-2 所示的登录界面。如果已经注册过苹果开发者账号, 我们可以直接输入; 如果没有注册, 则需要注册后才可以继续登录。

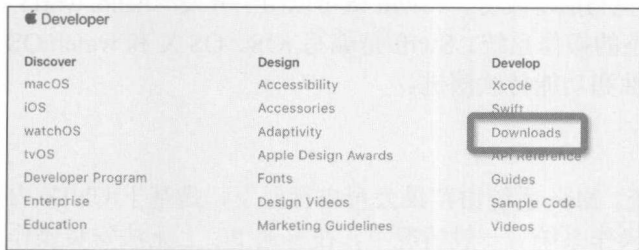


图 1-1 下载 Xcode 入口



图 1-2 登录界面

输入苹果开发者账号和密码, 单击“Sign In”, 我们会进入到如图 1-3 所示的下载最新版本 Xcode 的界面。

目前, 最新版本的 Xcode 是 Xcode 8.2.1, 单击“Download”, 我们就可以下载最新的 Xcode。下载完成后, 单击安装包, 根据提示就可以在 Mac 电脑上安装 Xcode 8.2.1。

我们也可以在 App Store 直接下载 Xcode 的最新版本 Xcode 8.2.1。



图 1-3 下载 Xcode 8.2.1 界面

2. Xcode 的卸载

Xcode 的卸载非常简单，在 Mac OS 的应用程序中直接删除即可。如图 1-4 所示，打开应用程序，右击“Xcode”弹出菜单，选择“移到废纸篓”，卸载 Xcode 应用。如果想彻底删除，继续清空废纸篓即可。



图 1-4 卸载 Xcode

1.1.4 Swift 语言开发文档

对于初学者来说，学会在 Xcode 中使用 API 帮助文档是非常重要的。下面通过一个例子来介绍 API 帮助文档的用法。

在创建一个新工程时，可以看到在 main.swift 文件中的代码，具体如下。

```
1 | import Foundation
2 | print("Hello, World!")
```

如果想对 print 函数有更多的了解，我们可以查看帮助文档。如果想要简单地查看帮助信

息，可以选中该方法，然后选择右侧的帮助检查器“？”；或者选中该函数并按住“option”键，在该函数下方会出现一个弹框，和右侧的帮助检查器中的内容完全一致，如图 1-5 所示。

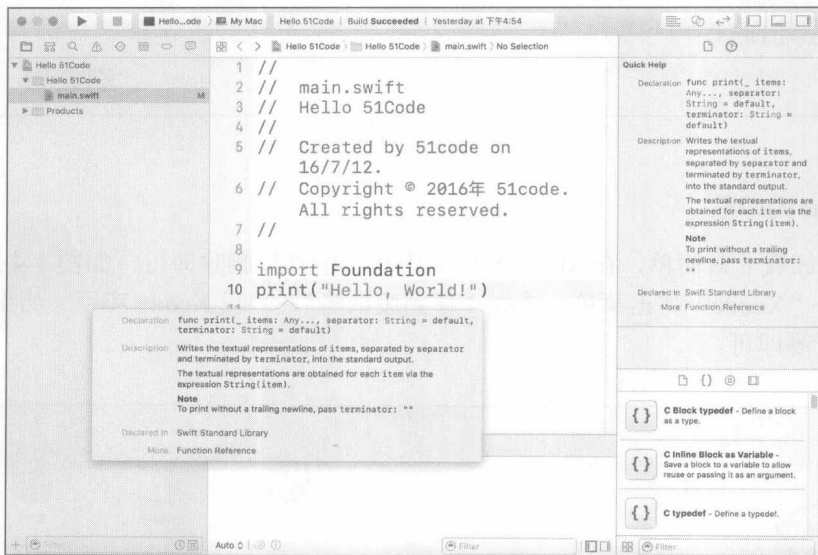


图 1-5 Xcode 快捷帮助检查器

在右侧的“？”，即 Xcode 快捷帮助检查器窗口中对该函数有详细的描述，包括使用的 iOS 版本、相关主题以及一些相关示例。

如果想要查询比较完整且全面的帮助文档，我们可以在选中 `print` 函数的同时，按住组合键“option”+“Shift”。这样就会打开一个 Xcode API 帮组搜索结果窗，如图 1-6 所示，右侧目录包括对 Swift、Objective-C、JavaScript 语言的简要介绍。在实际开发过程中，我们可以根据自己的需要，打开目录，查阅资料。

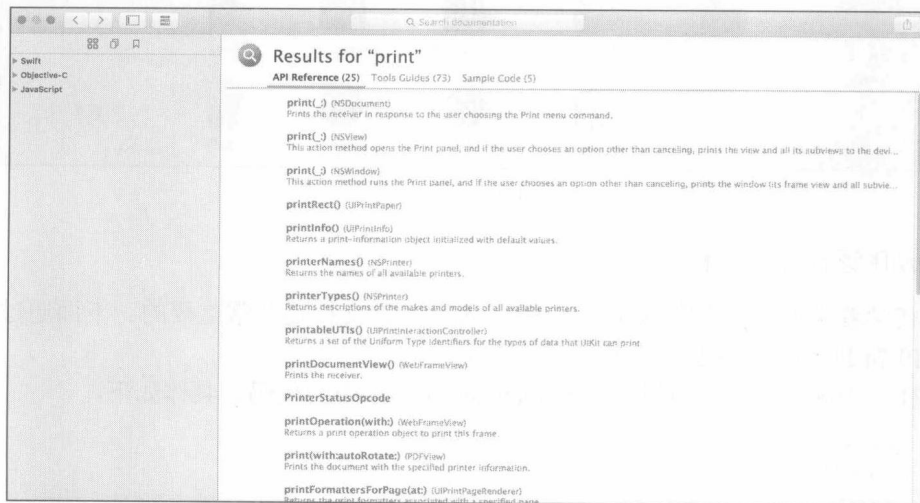


图 1-6 Xcode API 帮助搜索结果窗口

当我们需要查询工程中没有的某个知识点的时候，可以在工程中选择“Xcode->Window->Documentation and API Reference”，如图 1-7 所示，同样也能进入如图 1-6 所示的 Xcode API 帮助搜索结果窗口。

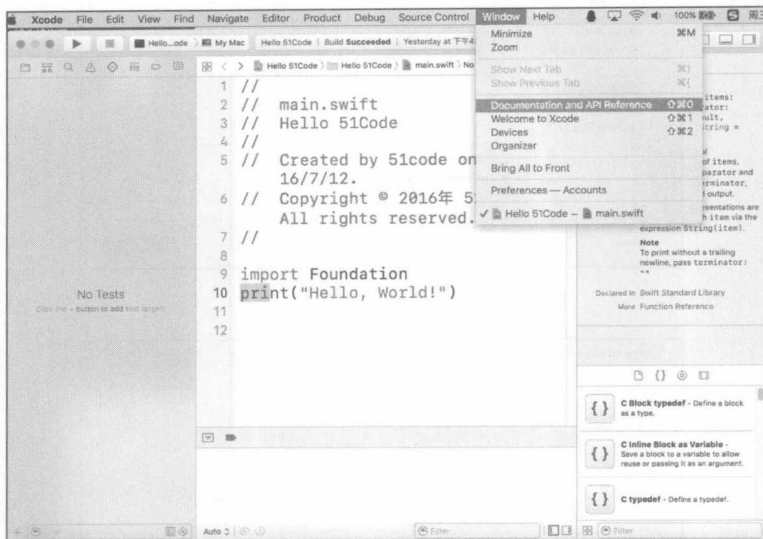


图 1-7 查找 API

1.2 Swift 语言编程体验

使用 Swift 编程，有两种方式编写代码：使用 Xcode 的 Playground，或创建工程运行 Swift 程序代码。本节以一个 Swift 程序为例，学习创建工程运行一个简单的 Swift 程序。

1.2.1 Hello, 51Code 程序

首先，新建一个工程。打开 Dock 栏的 Xcode 8.0，选择“Create a new Xcode project”或单击桌面顶部的菜单“File->New->Project”，在打开的 Choose a template for new project 界面中 OS X 下的 Application 中选择“Command Line Tool”工程模板。该模板主要用于 Swift 基础语法学习过程，如图 1-8 所示。

单击“Next”，如图 1-9 所示，在出现的界面中 Product Name 表示工程名，一般输入英文名称，为了清晰地表达项目的含义，这里我们将工程命名为“Hello 51Code”。Organization Name 表示组织者名，可以输入个人或公司的名字，在这里我们输入“51code”。Organization identifier 表示组织者域名，可以输入个人或公司的网址（网址需倒序书写），在这里我们输入“com.51code”。Language 表示选择的语言，可选择的语言有 Objective-C、C、C++、Swift，在这里我们选择“Swift”。

设置完相关的工程选项后，单击“Next”按钮，进入下一级界面，根据提示选择项目的保存位置。然后单击“Creat”按钮，将出现如图 1-10 所示的 Xcode 编辑界面。

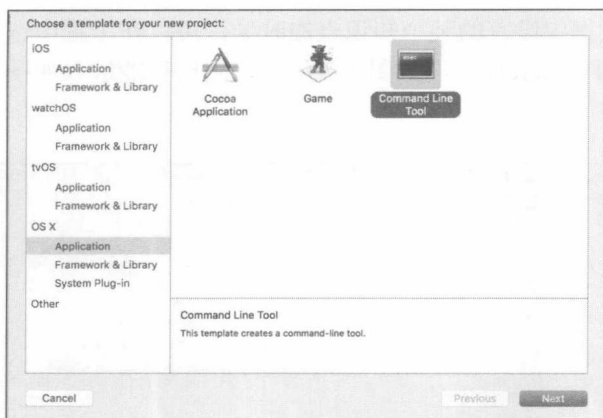


图 1-8 选择模板

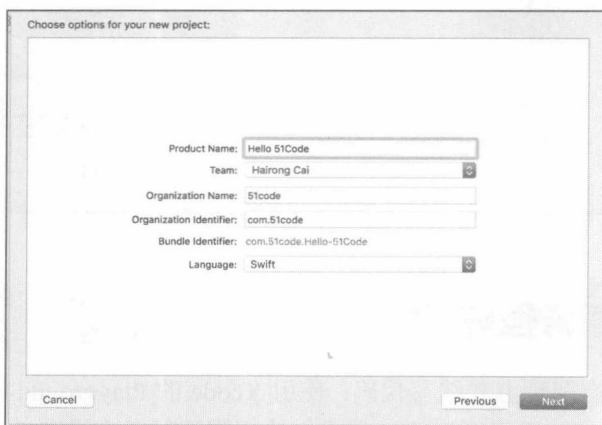


图 1-9 填写信息

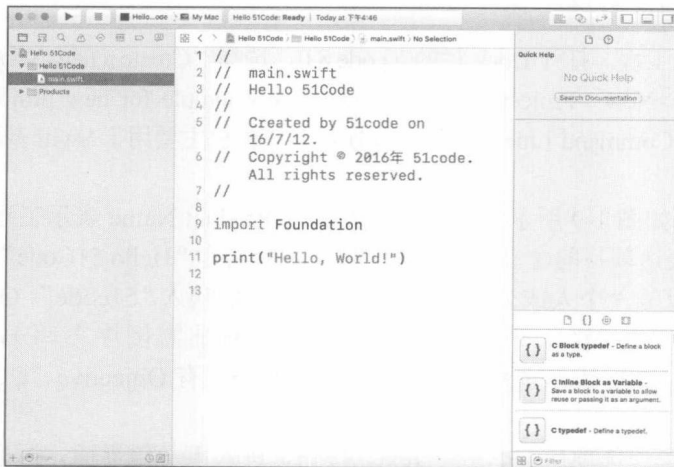


图 1-10 Xcode 编辑界面