



移动开发人才培养系列丛书

Swift

开发 | 标准教程

Beginning Swift Development

张明 吴琼 陈瑶 主编

- 以**全新 iOS 9** 为开发环境，采用开发工具 **Xcode 7** 进行讲解
- 近 **400** 个教学案例，**1** 个
综合项目，掌握从开发到 App 发布
- 提供**多媒体课件、源代码、习题、
上机实践**等丰富资源



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS



移动开发人才培养系列丛书

Swift

开发 | 标准教程

张明 吴琼 陈瑶 主编

人民邮电出版社

北 京

图书在版编目(CIP)数据

Swift开发标准教程 / 张明, 吴琼, 陈瑶主编. —
北京: 人民邮电出版社, 2016. 8
(移动开发人才培养系列丛书)
ISBN 978-7-115-42502-7

I. ①S… II. ①张… ②吴… ③陈… III. ①程序语
言—程序设计—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2016)第132403号

内 容 提 要

本书共分为15章, 主要内容包括编写Swift开发环境配置、Swift语言基础、语句和表达式、集合类型、程序控制结构、函数和闭包、类、继承、枚举和结构、构造器和析构器、扩展和协议、Swift语言的其他主题、使用Swift开发iOS应用、测试和发布App以及综合实例打砖块游戏等内容。书中对Swift语言中的一些开发技巧进行了展示。

本书内容丰富、结构新颖、难度适中、实用性强, 可作为普通高等院校Swift程序设计课程的教材, 也可供Swift开发初学人员参考阅读。

◆ 主 编 张 明 吴 琼 陈 瑶

责任编辑 刘 博

责任印制 沈 蓉 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>

北京隆昌伟业印刷有限公司印刷

◆ 开本: 787×1092 1/16

印张: 24

2016年8月第1版

字数: 630千字

2016年8月北京第1次印刷

定价: 59.80元

读者服务热线: (010)81055256 印装质量热线: (010)81055316

反盗版热线: (010)81055315

前 言

随着苹果官方大力推广全新的 iOS 开发语言 Swift, 该语言将会成为 iOS 应用程序设计教学的主流。然而, 目前市面上很多书籍对 Swift 语言都没有很系统地进行介绍, 并且大多是基于 Swift 1.0 等老版本的讲解。因此, 不仅需要使用 Swift 语言取代原来的 Objective-C 语言学习, 还需要选取最新的 Swift 语言版本。

在学习本书之前, 需要读者具备一些前提条件。第一, 读者应该熟悉 Mac 以及 iOS 操作系统, 了解苹果系统的特点和基本使用方法; 第二, 读者应该在计算机上安装合适的开发环境, 本书使用的是 Xcode 7.1。

Xcode 是苹果公司开发的基于 Swift 的图形化集成开发工具。其开发出来的应用程序执行效率高, 并且苹果公司提供了对 Xcode 的大力支持。苹果公司为用户提供了大量的标准类, 从而缩短了软件的开发周期。因此用 Xcode 开发编写 Swift 的应用程序可谓得天独厚。

掌握一门语言最好的方式就是实践。因此, 本书将着眼点放在理论知识讲解与实践操作相结合上, 使读者快速掌握 Swift 编程技术。本书是编者多年教学和应用开发经验的总结。书中既介绍了 Swift 编程语言涉及的所有知识内容, 又展示了开发过程中的开发经验和技巧, 希望对读者有所助益。

本书的多数章节, 将首先就相关的 Swift 基础知识进行介绍, 然后讲解如何使用 Swift 编程语言去开发 iOS 应用, 并布置了若干具有代表性的习题和上机练习题, 使读者可以通过自己动手, 在实践中掌握 Swift 程序设计的方法和技巧。

本书由张明、吴琼、陈瑶主编, 其中张明编写第 1~8 章, 吴琼编写第 9~12 章, 陈瑶编写第 13~15 章。

编 者

2016 年 3 月

目 录

第 1 章 编写第一个 Swift 程序 1

1.1 初识 Swift 1

1.1.1 Swift 的发展 1

1.1.2 Swift 的特点 1

1.1.3 Swift 语言的转换 2

1.2 构建开发环境 2

1.2.1 申请苹果账号 3

1.2.2 安装 Xcode 6

1.2.3 更新新组件和文档 9

1.3 编写第一个程序 11

1.3.1 创建项目 11

1.3.2 Xcode 界面介绍 13

1.3.3 编译和运行 14

1.3.4 编写代码 15

1.4 Swift 代码分析 15

1.4.1 代码构成 15

1.4.2 标识符 16

1.4.3 关键字 16

1.4.4 注释 17

1.5 调试 18

1.6 使用帮助文档 19

1.7 上机实践 20

第 2 章 Swift 语言基础 21

2.1 常量变量 21

2.1.1 常量 21

2.1.2 变量 21

2.1.3 为声明的变量和常量指定数据类型 22

2.2 简单数据类型 22

2.2.1 整数 22

2.2.2 整型 23

2.2.3 浮点类型 24

2.2.4 布尔类型 25

2.2.5 可选类型 25

2.3 字面值 26

2.3.1 整型字面值 26

2.3.2 浮点类型的字面值 26

2.3.3 布尔类型的字面值 28

2.4 高级数据类型——元组 28

2.5 类型别名 29

2.6 字符和字符串 29

2.6.1 字符类型与字面值 29

2.6.2 字符串类型与字面值 30

2.6.3 初始化空字符串 31

2.6.4 字符串连接 31

2.6.5 字符计数 32

2.6.6 判断字符串 32

2.6.7 大小写转换 34

2.6.8 插入和删除 35

2.7 编码格式 Unicode 37

2.7.1 什么是 Unicode 37

2.7.2 字符串的 Unicode 表示形式 38

2.8 综合案例 40

2.8.1 为圆周率 3.14159265359 指定数据类型 40

2.8.2 组成字符串，并插入特殊符号 40

2.9 上机实践 41

第 3 章 语句和表达式 42

3.1 语句 42

3.2 运算符与表达式 42

3.2.1 常用术语——元 42

3.2.2 赋值运算符和表达式 43

3.2.3 算术运算符和表达式 43

3.2.4 求余运算符和表达式 46

3.2.5 自增自减运算符和表达式 47

3.2.6 一元负号运算符 49

3.2.7 一元正号运算符 50

3.2.8 位运算符.....	50	4.4.6 确定集合的顺序.....	81
3.2.9 溢出运算符.....	55	4.5 集合的基本运算.....	81
3.2.10 比较运算符和表达式.....	57	4.5.1 $a \cap b$	81
3.2.11 三元条件运算符和表达式.....	58	4.5.2 $a \cup b$	82
3.2.12 逻辑运算符和表达式.....	58	4.5.3 $a - b$	83
3.2.13 范围运算符.....	61	4.5.4 $a - b \cup b - a$	83
3.2.14 复合赋值运算符和表达式.....	62	4.6 集合间关系.....	84
3.2.15 求字节运算符和表达式.....	63	4.6.1 相等判断.....	84
3.2.16 强制解析.....	63	4.6.2 子集的判断.....	85
3.2.17 空合运算符.....	63	4.6.3 父集合的判断.....	85
3.3 数值类型转换.....	64	4.6.4 其他判断.....	86
3.3.1 整数的转换.....	64	4.7 字典.....	86
3.3.2 整数和浮点数的转换.....	65	4.7.1 字典字面量.....	86
3.4 综合案例.....	66	4.7.2 字典的声明.....	87
3.4.1 水仙花数.....	66	4.7.3 字典的初始化.....	87
3.4.2 将 7489 逆序输出.....	66	4.8 字典的操作.....	88
3.5 上机实践.....	67	4.8.1 获取字典中的元素个数.....	88
第 4 章 集合类型.....	68	4.8.2 读取键的值.....	88
4.1 数组.....	68	4.8.3 添加元素.....	89
4.1.1 数组字面量.....	68	4.8.4 修改键关联的值.....	89
4.1.2 数组的声明.....	68	4.8.5 删除值.....	90
4.1.3 数组的初始化.....	69	4.9 综合案例.....	91
4.2 数组的操作.....	71	4.9.1 求 3 科成绩的平均值.....	91
4.2.1 获取数组中元素个数.....	71	4.9.2 获取奇数月.....	92
4.2.2 判断数组是否为空.....	71	4.10 上机实践.....	94
4.2.3 在末尾添加一个元素.....	72	第 5 章 程序控制结构.....	95
4.2.4 插入值.....	73	5.1 顺序结构.....	95
4.2.5 读取值.....	74	5.2 选择结构——if 语句.....	95
4.2.6 修改值.....	74	5.2.1 if 语句.....	96
4.2.7 删除值.....	75	5.2.2 if...else 语句.....	96
4.3 集合.....	76	5.2.3 if...else if 语句.....	97
4.3.1 集合的声明.....	76	5.2.4 if 语句的嵌套.....	98
4.3.2 集合的初始化.....	77	5.3 选择结构——switch 语句.....	99
4.4 集合的操作.....	78	5.3.1 switch 语句基本形式.....	99
4.4.1 获取集合中元素个数.....	78	5.3.2 switch 语句的使用规则.....	100
4.4.2 判断集合是否为空.....	78	5.4 循环结构——for 语句.....	103
4.4.3 判断集合中是否包含某一值.....	79	5.4.1 for...in 循环.....	103
4.4.4 插入值.....	79	5.4.2 for-condition-increment 条件循环.....	108
4.4.5 删除值.....	79	5.5 循环结构——while 语句.....	108

5.5.1 while 循环	109	6.6 函数类型	128
5.5.2 repeat while 循环	110	6.6.1 使用函数类型	129
5.6 跳转语句	110	6.6.2 使用函数类型作为参数类型	129
5.6.1 continue 语句	110	6.6.3 使用函数类型作为返回值类型	130
5.6.2 break 语句	111	6.7 标准函数	131
5.6.3 fallthrough	112	6.7.1 绝对值函数 abs()	131
5.7 标签语句	112	6.7.2 最大值函数 max()/最小值函数 min()	132
5.7.1 标签语句的定义	113	6.7.3 序列排序函数 sortInPlace()	132
5.7.2 标签语句的使用	113	6.7.4 序列倒序函数 reverse()	133
5.8 综合案例	114	6.8 函数的嵌套	133
5.8.1 打印九九乘法表	114	6.8.1 嵌套调用	134
5.8.2 使用 if else 比较 3 个数值大小	115	6.8.2 递归调用	135
5.8.3 计算 1~100 的奇数和	116	6.9 闭包	135
5.9 上机实践	116	6.9.1 闭包表达式	136
第 6 章 函数和闭包	117	6.9.2 Trailing 闭包	138
6.1 函数介绍	117	6.9.3 捕获值	139
6.1.1 函数的功能	117	6.10 综合案例	140
6.1.2 函数的形式	118	6.10.1 打印金字塔	140
6.2 使用无参函数	118	6.10.2 猴子吃桃	141
6.2.1 无参函数的声明定义	119	6.11 上机实践	142
6.2.2 无参函数的调用	119	第 7 章 类	143
6.2.3 空函数	119	7.1 类与对象	143
6.3 使用有参函数	120	7.1.1 类的组成	143
6.3.1 有参函数的声明定义	120	7.1.2 创建类	143
6.3.2 有参函数的调用	120	7.1.3 实例化对象	144
6.3.3 参数的注意事项	121	7.2 属性	144
6.4 函数参数的特殊情况	121	7.2.1 存储属性	144
6.4.1 函数参数名	122	7.2.2 计算属性	146
6.4.2 指定外部参数名	122	7.2.3 类型属性	148
6.4.3 忽略外部参数名	122	7.2.4 属性监视器	150
6.4.4 为参数设置默认值	123	7.3 方法	152
6.4.5 可变参数	123	7.3.1 实例方法	152
6.4.6 常量参数和变量参数	124	7.3.2 类型方法	153
6.4.7 输入-输出参数	124	7.3.3 存储属性、局部变量和全局变量的区别	155
6.5 函数的返回值	125	7.3.4 局部变量和存储属性同名的解决方法——self 属性	156
6.5.1 具有一个返回值的函数	125	7.4 下标脚本	157
6.5.2 具有多个返回值的函数	126		
6.5.3 可选元组返回类型	127		
6.5.4 无返回值	128		

7.4.1 定义下标脚本	157
7.4.2 调用下标脚本	157
7.4.3 使用下标脚本	158
7.5 类的嵌套	160
7.5.1 直接嵌套	160
7.5.2 多次嵌套	161
7.6 可选链接	162
7.6.1 使用可选链接调用代替强制解析	163
7.6.2 通过可选链接调用属性、下标脚本、方法	163
7.6.3 连接多个链接	165
7.7 综合案例	166
7.7.1 收支情况	166
7.7.2 根据周长计算面积	167
7.8 上机实践	168
第 8 章 继承	170
8.1 为什么要使用继承	170
8.1.1 重用代码、简化代码	170
8.1.2 扩展功能	170
8.2 继承的实现	170
8.2.1 继承的定义	170
8.2.2 属性的继承	172
8.2.3 下标脚本的继承	173
8.2.4 方法的继承	174
8.3 继承的特点	174
8.3.1 多层继承	174
8.3.2 不可删除	175
8.4 重写	176
8.4.1 重写属性	176
8.4.2 重写下标脚本	178
8.4.3 重写方法	179
8.4.4 访问父类成员	180
8.4.5 阻止重写	182
8.5 类型转换	184
8.5.1 类型检查	184
8.5.2 向下转型	185
8.5.3 AnyObject 和 Any 的类型转换	185
8.6 综合案例	187
8.7 上机实践	188

第 9 章 枚举和结构 189

9.1 枚举的构成	189
9.2 定义枚举	189
9.2.1 任意类型的枚举类型	189
9.2.2 指定数据类型的枚举类型	190
9.3 定义枚举的成员	190
9.3.1 定义任意类型的枚举成员	190
9.3.2 定义指定数据类型的枚举成员	191
9.3.3 定义枚举成员时的注意事项	192
9.4 实例化枚举的对象	192
9.5 枚举成员与 switch 匹配	192
9.6 访问枚举类型中成员的原始值	193
9.6.1 通过成员访问原始值	193
9.6.2 通过原始值访问成员	194
9.7 关联值	195
9.8 定义枚举的其他内容	195
9.8.1 定义属性	195
9.8.2 定义方法	197
9.8.3 定义下标脚本	198
9.9 递归枚举	199
9.10 结构的构成	200
9.11 结构的创建与实例化	200
9.11.1 结构的创建	200
9.11.2 结构体的实例化	200
9.12 定义结构中的内容	201
9.12.1 定义属性	201
9.12.2 定义方法	204
9.12.3 定义下标脚本	205
9.13 类、枚举、结构的区别	206
9.14 嵌套类型	206
9.15 综合案例	208
9.15.1 输出对应音符发音	208
9.15.2 根据棱长计算正方体的表面积和体积	208
9.16 上机实践	209

第 10 章 构造器和析构器 210

10.1 值类型的构造器	210
10.1.1 默认构造器	210

10.1.2	自定义构造器	211
10.1.3	构造器代理	214
10.2	类的构造器	216
10.2.1	默认构造器	217
10.2.2	自定义构造器	217
10.2.3	构造器代理	220
10.2.4	类的两段式构造过程	221
10.2.5	构造器的继承和重载	222
10.2.6	必要构造器	225
10.3	可失败构造器	226
10.3.1	定义可失败构造器	226
10.3.2	枚举类型的可失败构造器	227
10.3.3	类的可失败构造器	228
10.3.4	构造失败的传递	229
10.3.5	重写一个可失败构造器	230
10.3.6	可失败构造器 init!	231
10.4	构造器的特殊情况	232
10.4.1	可选属性类型	232
10.4.2	修改常量属性	232
10.5	设置默认值	233
10.5.1	在定义时直接赋值	233
10.5.2	在构造器中赋值	233
10.5.3	使用闭包设置属性的默认值	234
10.5.4	使用函数设置默认值	235
10.6	析构器	235
10.6.1	理解析构器	235
10.6.2	析构器的定义	236
10.6.3	使用析构器	236
10.6.4	使用析构器的注意事项	237
10.6.5	构造器和析构器的区别	239
10.7	综合案例	239
10.7.1	游戏属性	239
10.7.2	模拟下线通知	240
10.8	上机实践	240

第 11 章 扩展和协议 241

11.1 扩展	241
11.1.1 扩展的定义	241
11.1.2 扩展属性	241
11.1.3 扩展构造器	243

11.1.4	扩展方法	244
11.1.5	扩展下标脚本	246
11.1.6	扩展嵌套类型	247
11.2	协议	248
11.2.1	协议的定义	248
11.2.2	协议的实现	248
11.2.3	协议的成员声明——属性	249
11.2.4	协议的成员声明——方法	251
11.2.5	协议的成员声明——可变方法	252
11.2.6	协议的成员声明——构造器	253
11.3	可选协议	254
11.3.1	定义可选协议	254
11.3.2	声明可选成员	254
11.3.3	调用可选协议	255
11.4	使用协议	256
11.4.1	协议作为常量、变量等的数据类型	256
11.4.2	协议作为返回值或参数类型	257
11.4.3	协议作为集合的元素类型	258
11.5	在扩展中使用协议	259
11.5.1	在扩展中实现协议	259
11.5.2	定义协议成员	259
11.5.3	扩展协议声明	260
11.6	协议的继承	260
11.7	协议合成	262
11.8	检查协议的一致性	263
11.9	委托	264
11.10	综合案例	267
11.11	上机实践	268

第 12 章 Swift 语言的其他主题 269

12.1	自动引用计数	269
12.1.1	自动引用计数的工作机制	269
12.1.2	循环强引用的产生	270
12.1.3	循环强引用的解决方法	273
12.2	运算符重载	278
12.2.1	为什么使用运算符重载	278
12.2.2	算术运算符的重载	278
12.2.3	一元负号/正号运算符的重载	279
12.2.4	复合赋值运算符的重载	280

12.2.5	自增自减运算符的重载	281
12.2.6	比较运算符的重载	285
12.2.7	自定义运算符的重载	286
12.3	泛型	289
12.3.1	泛型函数	289
12.3.2	泛型类型	290
12.3.3	泛型类的层次结构	292
12.3.4	扩展一个泛型类型	294
12.3.5	具有多个类型参数的泛型	295
12.3.6	类型约束	295
12.3.7	关联类型	296
12.4	错误处理	300
12.4.1	抛出错误	300
12.4.2	捕获错误和处理错误	301
12.4.3	清理动作	302
12.5	综合案例	302
12.6	上机实践	303

第 13 章 使用 Swift 开发 iOS

应用

13.1	创建项目	304
13.2	运行程序	306
13.3	模拟器的操作	307
13.3.1	模拟器与真机的区别	307
13.3.2	退出应用程序	307
13.3.3	应用程序图标的位置	307
13.3.4	语言设置	309
13.3.5	旋转	312
13.3.6	删除应用程序	312
13.4	编辑界面	313
13.4.1	界面介绍	313
13.4.2	设计界面	314
13.4.3	视图对象库的介绍	316
13.4.4	编写代码	317
13.5	上机实践	320

第 14 章 测试和发布 App

14.1	测试 App 概述	321
14.1.1	测试驱动的软件开发流程	321
14.1.2	iOS 单元测试框架	322
14.2	使用 XCTest 测试框架测试驱动的软件开发案例	322
14.2.1	测试案例前期准备	322
14.2.2	添加 XCTest 到项目中	326
14.2.3	测试驱动的开发流程	328
14.3	发布前的准备工作	332
14.3.1	申请付费的开发者账号	332
14.3.2	申请 App ID	335
14.3.3	申请证书	336
14.3.4	添加图标	343
14.3.5	调整 Application Target 属性	344
14.3.6	为发布进行编译	345
14.3.7	应用打包	347
14.4	进行发布	349
14.4.1	创建应用及基本信息	349
14.4.2	应用定价信息	352
14.4.3	上传应用	353
14.5	常见审核不通过的原因	357
14.6	上机实践	358

第 15 章 综合案例：打砖块游戏

15.1	功能介绍	359
15.2	界面设计	360
15.2.1	准备工作	360
15.2.2	主界面设计	361
15.2.3	游戏界面设计	362
15.2.4	游戏介绍界面设计	363
15.3	功能实现	364
15.3.1	界面之间的切换	364
15.3.2	打砖块游戏功能	368
15.4	真机测试	374

第 1 章

编写第一个 Swift 程序

苹果的操作系统中各个丰富的功能都是由程序实现的。程序是为了实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合。现在苹果操作系统的程序由原来的 Objective-C 编写改为了由 Swift 编写，并且 Swift 将取代 Objective-C 成为官方推荐语言。本章将讲解什么是 Swift 语言，构建编写 Swift 需要的开发环境以及编写第一个 Swift 程序。

1.1 初识 Swift

Swift 是 2014 年 6 月苹果公司在 WWDC 开发者大会上推出的语言。它是全新的编程语言，用于编写 iOS、WatchOS、OS X 应用。本节将讲解 Swift 的发展、特点以及语言转换。

1.1.1 Swift 的发展

从 2014 年 6 月苹果公司推出 Swift，到现在已经有 1 年多的时间了。表 1-1 中列出了 Swift 语言在这一年中的发展。

表 1-1

Swift 的发展史

时 间	事 件
2010 年 7 月	开始着手开发 Swift 编程语言的工作
2014 年 6 月 3 日	在开发者大会上发布 Swift 语言
2014 年 6 月 4 日	《Swift 中文版》翻译小组在 github 上进行翻译
2014 年 6 月 12 日	《Swift 中文版》第一版问世
2014 年 9 月 9 日	Swift 语言发布 1.1 版本
2015 年 2 月	Swift 语言发布 1.2 版本
2015 年 6 月 8 日	Swift 语言发布 2.0 版本
2015 年 9 月 23 日	Swift 语言发布 2.1 版本
2015 年 12 月 8 日	Swift 语言发布 2.1.1 版本
2016 年 3 月 21 日	Swift 语言发布 2.2 版本

1.1.2 Swift 的特点

Swift 采用安全的编程模式，并添加了很多新特性，这将使编程更简单，更灵活。以下是 Swift

添加的新特性。

1. 安全

Swift 是一种类型安全的语言，它使用类型推断机制，限制对象指针使用、自动管理内存来使程序更安全，让开发人员更容易开发出安全稳定的软件。

2. 流行

Swift 具有 optional、泛型、元组等现代语言的特性。它比 Objective-C 语言更灵活，更接近于自然语言，使代码可读性更好。

3. 强大

使用 Swift 中强大的模式匹配特性可以写出更简单，更直观表意的代码。通过变量插值的方式可以更方便地格式化字符串，同时也可以方便地使用 Foundation and UIKit。

4. 交互性

使用 playgrounds 来试验新技术，分析问题，做所见即所得的界面原型。

5. 高效

Swift 的编译器使用高级的代码分析功能来调优开发者的代码，让开发者更专注于开发应用，而不必在性能优化上投入大量的精力。

6. 兼容

Swift 完全与 Objective-C 语言相兼容。这样，传统的苹果开发人员可以很轻松的从 Objective-C 过渡到 Swift 上。同时，Objective-C 的代码也可以放到 Swift 上使用。

虽然 Swift 是很不错，但是它也有很多方面的缺点，这里总结了三点。

(1) Swift 支持的复杂数据结构比较有限。

(2) 目前，它可以使用的第三方库也比较少。

(3) Swift 和 Objective-C 一样是基于 LLVM 编辑器，使它不可以在 Android、Windows Phone 上工作。

1.1.3 Swift 语言的转换

由于 Swift 是一种新的语言，所以现在还是不稳定的，这也是苹果公司一直升级 Swift 版本的原因。Swift 每升级一个版本会导致使用以前版本编写的 Swift 代码出现错误，这很可能让想要学习 Swift 语言的开发者对 Swift 语言失去信心。为了解决这一问题，苹果公司给出了对应的讲解办法。开发者只需要在 Xcode 的菜单栏中找到 Edit|Convert|To Latest Swift 命令，就可以将旧版本的 Swift 语言转换为新版本的 Swift 语言。



Xcode 是苹果公司推出的开发工具，可以用来进行 Swift 程序的编写。对于 Xcode 的安装，我们会在 1.2 节中介绍。

1.2 构建开发环境

软件开发环境 (Software Development Environment) 是为了支持系统软件和应用软件工程化开发和维护的一组软件，通常简称为 SDE。Swift 的开发环境就是苹果系统。它的开发工具分为两种：图形化开发工具和命令行开发工具。为了方便使用，本书的开发工具将使用图形化开发工

具 Xcode。本节将讲解苹果账号的注册、下载和安装 Xcode 以及更新组件和文档等相关方面的内容。

1.2.1 申请苹果账号

苹果账号是苹果公司专门为 iOS、Mac、WatchOS 开发成员提供的账号,也称开发者账号。有了此账号,开发成员才可以在 App Store 中进行 SDK 以及一些常用软件的下载以及安装。在苹果公司注册苹果账号,就可以成为开发成员。开发成员一共可以分为四种,如表 1-2 所示。

表 1-2

苹果账号的成员

用户种类	收 费
在线开发成员	免费
标准开发成员	\$99/年
企业开发成员	\$299/年
大学开发成员	免费

以下是申请免费苹果的具体操作步骤。

(1) 在 Dock (Dock 一般指的是苹果操作系统中的停靠栏)中,找到浏览器 Safari,如图 1-1 所示。

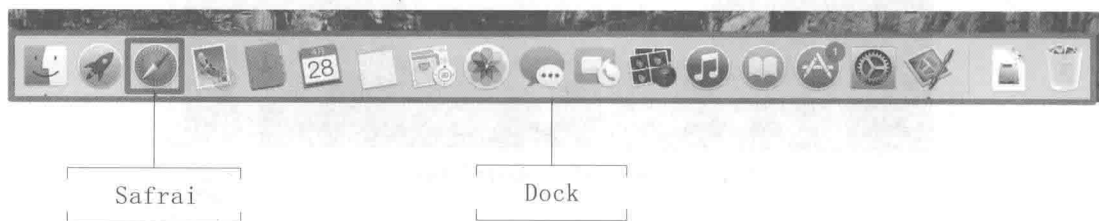


图 1-1 浏览器 Safari

(2) 单击“Safari”图标,打开 Safari 浏览器,如图 1-2 所示。



图 1-2 浏览器 Safari

(3) 在地址栏中输入网址 (<http://developer.apple.com>),按下回车,进入 Apple Developer 网页,如图 1-3 所示。



图 1-3 Apple Developer 网页

(4) 单击“Member Center”选项，进入登录 Apple ID 的网页，如图 1-4 所示。

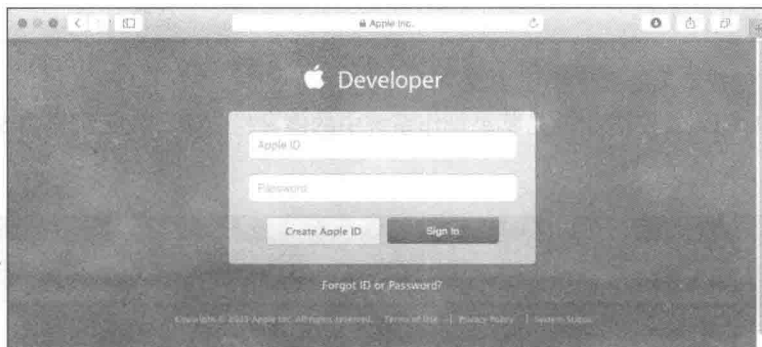


图 1-4 登录 Apple ID 的网页

(5) 单击“Create Apple ID”按钮，进入 Apple-My Apple ID 网页，如图 1-5 所示。



图 1-5 Apple-My Apple ID 网页

(6) 在网页中按照要求输入内容后, 单击网页最下方的“Create Apple ID”按钮, 进入确认邮件地址的网页, 如图 1-6 所示。

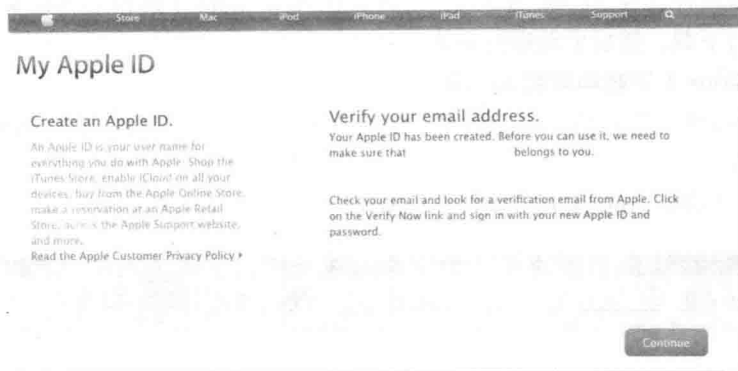


图 1-6 确认邮件地址的网页

(7) 单击“Continue”按钮, 进入到确定邮件地址的另一个网页。单击此网页中的“Send Verification Email”按钮, 苹果公司会向作为账号的邮箱发送一封确认邮件。

(8) 进入账号所使用的邮箱, 就会看到 Apple 发来的一封确定邮件地址的邮件。打开该邮件, 如图 1-7 所示。

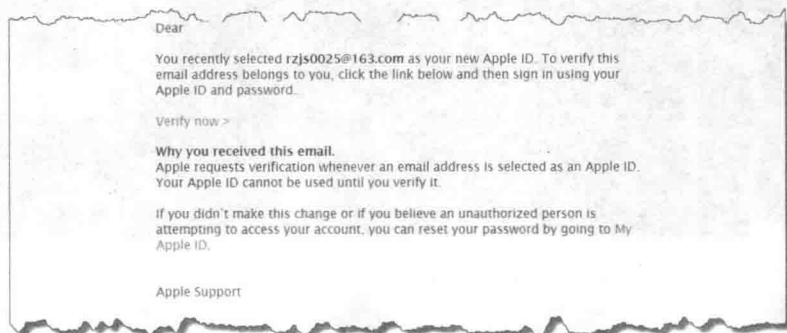


图 1-7 确认邮件

(9) 单击“Verify now”链接, 进入 Apple-My Apple ID-Email Verification 网页, 如图 1-8 所示。



图 1-8 Apple-My Apple ID-Email Verification 网页

(10) 输入需要验证的邮箱以及地址, 单击“Verify Address”按钮, 进入到下一个网页, 此网页会提示开发者注册的 Apple ID 现在已经可以使用了。

1.2.2 安装 Xcode

本小节将讲解两种安装 Xcode 的方式：一种是在 App Store 上进行下载和安装；另一种是在其他的网站中进行下载，然后手动进行安装。

1. 在 App Store 上下载和安装 Xcode

App Store 中提供了很多的软件，而 Xcode 也在其中。以下就是在 App Store 中下载和安装 Xcode 的具体步骤。

(1) 在 Dock 中找到 App Store，如图 1-9 所示。

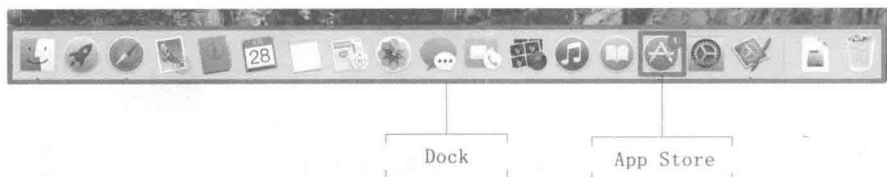


图 1-9 App Store 图标

(2) 单击 App Store 图标，打开 App Store 窗口，如图 1-10 所示。

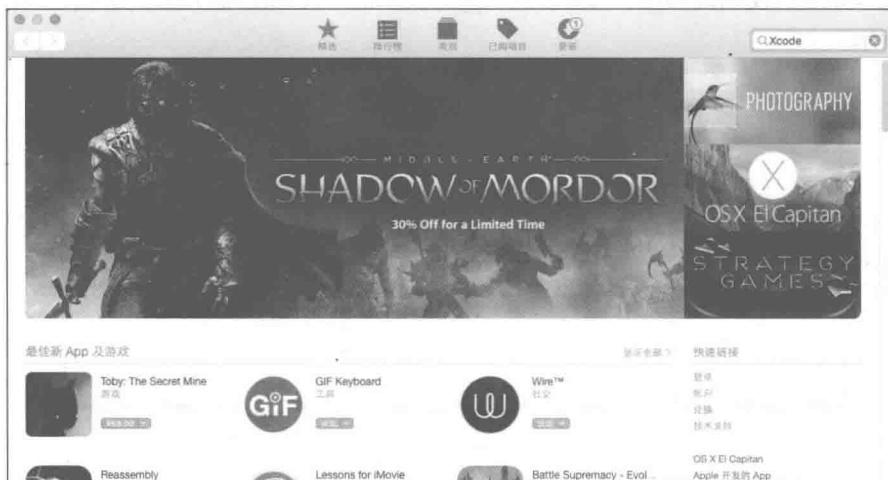


图 1-10 App Store 窗口

(3) 在搜索栏中输入要搜索的内容，即 Xcode，按下回车，进行搜索，如图 1-11 所示。

(4) 单击 Xcode 右下方的“获取”按钮，此时“获取”按钮变为了“安装 App”按钮，如图 1-12 所示。



图 1-11 搜索结果



图 1-12 安装 App

(5) 单击“安装 App”按钮，弹出“登录 App Store 来下载”对话框，如图 1-13 所示。

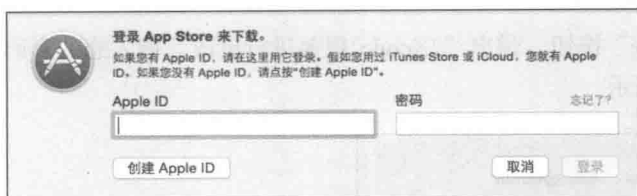


图 1-13 “登录 App Store 来下载”对话框

(6) 输入 Apple ID 以及密码后，单击“登录”按钮。此时，“安装 App”按钮变为了“安装”按钮，如图 1-14 所示。并且 Xcode 会在 Launchpad 中进行下载和安装，如图 1-15 所示。



图 1-14 “安装”按钮



图 1-15 开始下载

(7) 一般在 Launchpad 中下载的软件，都可以在应用程序中找到。选择“前往|应用程序”打开应用程序，如图 1-16 所示。



图 1-16 “应用程序”窗口