



Swift

让没有学过任何程序设计语言的人

轻松上手

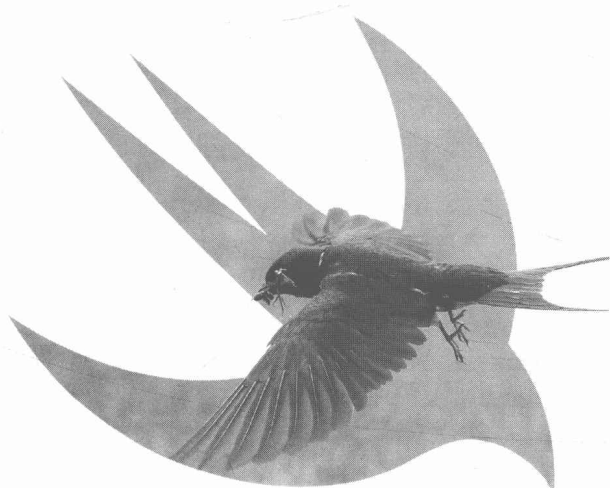
蔡明志 著

► 图文解释 + 丰富范例 + 课后练习

► 掌握最新技术精髓

► 教您用 Swift 语言开发苹果的 iOS 平台的应用

程序设计 实战入门



Swift

程序设计 实战入门



蔡明志 著



机械工业出版社
China Machine Press

图书在版编目 (CIP) 数据

Swift程序设计实战入门 / 蔡明志著. -- 北京 : 机械工业出版社, 2015.11

ISBN 978-7-111-51905-8

I. ①S… II. ①蔡… III. ①程序语言 - 程序设计 IV. ①TP312

中国版本图书馆CIP数据核字(2015)第252955号

本书版权登记号: 图字: 01-2015-1582

本书为经台湾碁峰资讯股份有限公司独家授权发行的中文简体版。本书中文简体字版在中国大陆之专有出版权属机械工业出版社所有。在没有得到本书原版出版者和本书出版者书面许可时, 任何单位和个人不得擅自摘抄、复制本书的一部分或全部以任何方式 (包括资料和出版物) 进行传播。

本书原版版权属碁峰资讯股份有限公司。版权所有, 侵权必究。

Swift 集合了 C 与 Objective-C 程序语言的优点, 但比这两种编程语言在编译与运行上更加快捷。本书依据此特性编写正文与设计范例程序, 必要时辅以图形解释, 全书内容简明易懂, 即使没有学过任何程序语言的人也能快速上手。

本书分 18 章, 从如何使用 Xcode 开始讲解, 包括变量与常量、循环语句、选择语句、函数、数组与字典等; 接下来是面向对象程序设计, 如类、继承、重载、重写以及泛型等内容; 最后讨论编写 iOS App 时用到的主题, 如初始化与析构、自动引用计数、可选链以及协议等。

本书内容全面, 不仅涵盖各种重要语言工具的基础知识, 更重要的是设计了大量的程序, 适合广大计算机爱好者及编程人员学习, 也可供大中专院校相关专业在校师生参考阅读。

Swift程序设计实战入门

出版发行: 机械工业出版社 (北京市西城区百万庄大街22号 邮政编码: 100037)

责任编辑: 夏非彼 迟振春

印 刷: 中国电影出版社印刷厂

开 本: 188mm×260mm 1/16

书 号: ISBN 978-7-111-51905-8

版 次: 2016年1月第1版第1次印刷

印 张: 19.25

定 价: 55.00元

凡购本书, 如有缺页、倒页、脱页, 由本社发行部调换

客服热线: (010) 88379426 88361066

购书热线: (010) 68326294 88379649 68995259

投稿热线: (010) 88379604

读者信箱: hzit@hzbook.com

版权所有·侵权必究

封底无防伪标均为盗版

本书法律顾问: 北京大成律师事务所 韩光/邹晓东

前言

笔者看过与研究过许多编程语言，大家可以列举的语言大多都有所涉及。最近因为编写 iOS 的 App，所以大部分时间都在使用 Objective-C。要学会 Objective-C 可能会有一些门槛，若用户已有一些 C 与 C++ 的语言基础，可能比较容易跨越。

其实 Objective-C 现在已经容易多了，以前的引用计数（reference count）不易掌控，可以说是程序员的梦魇，现在已经改为自动引用计数（automatic reference count），绝不会因为内存不足而死机，这是程序员的福音。

尽管如此，Apple 为了适应新的 Xcode 6 环境，2014 年 6 月也公布了开发 iOS 与 OS X App 新的程序语言，名为 Swift。它建立在 C 与 Objective-C 之上，并采用安全性高的程序设计模式以及加入最新的属性，使得程序设计更具灵活性和趣味性。在内存的管理上它也使用自动引用计数，和 Swift 的意思相同，它的编译与运行犹如燕子般轻盈、快速。

本书参考 Apple 官方公布的 Swift 程序语言，经过整理后以浅显易懂的语言阐述，配合丰富的范例程序、图表，以及每章结尾的习题，让用户可以快速编写 Swift 程序。本书共分 18 章，以编写简单的范例程序开始，接着是变量、常量与数据类型、运算符、循环语句、选择语句、集合类型、函数、闭包、类与结构、属性与方法、继承、初始化与析构、自动引用计数、可选链、类型转换与扩展、协议、泛型以及运算符等。

读完这 18 章的精彩正文，用户就可以进入编写 iOS 与 OS X App 的行列，为自己的人生注入新的契机。本书的封面取用燕子的图像，以 Apple 官方的燕子图片为底，搭配真实燕子飞翔的图片（由田念鲁先生提供），带用户翱翔天空，美梦成真。

本书范例和习题的程序代码下载地址为：<http://pan.baidu.com/s/1qiizs>。

如果下载有问题，请电子邮件联系 booksaga@126.com，邮件主题为“求 Swift 程序设计实战入门代码”。



2015 年 7 月

目 录

前 言

第1章 从简单的范例谈起	1
1.1 编写您的第一个程序	1
1.2 程序解析	6
1.3 Playground介绍	6
习题	10
第2章 变量、常量以及数据类型	12
2.1 简述变量与常量	12
2.2 简述数据类型	12
2.3 声明变量与常量	15
2.4 输出变量与常量	18
2.5 注释语句	19
2.6 分号	19
2.7 字符串类型	19
2.7.1 字符串的函数	21
2.7.2 字符串属于值类型	24
2.8 选项类型	24
习题	25
第3章 运算符	29
3.1 算术运算符	29
3.2 关系运算符	30
3.3 逻辑运算符	31
3.4 自增与自减运算符	32
3.5 赋值运算符	34
习题	35
第4章 循环语句	38
4.1 for循环语句	38

4.2 while 循环语句	42
4.3 do...while 循环语句.....	44
4.4 for-in 循环语句.....	45
4.5 嵌套循环	47
4.6 String(format:)格式	54
习题	54
第5章 选择语句	59
5.1 if 语句	59
5.2 if...else 语句.....	61
5.3 else...if语句.....	64
5.4 switch 语句.....	66
5.5 条件运算符.....	72
5.6 break、continue及fallthrough语句	72
习题	75
第6章 集合类型	79
6.1 数组的表示法	79
6.1.1 数组的运行与一些常用的API	81
6.1.2 二维数组	86
6.2 字典的表示法	87
6.3 集合类型的赋值与复制行为	90
6.3.1 数组的赋值与复制行为	90
6.3.2 字典的赋值与复制行为	93
习题	96
第7章 函数.....	101
7.1 定义与调用函数	101
7.1.1 函数的参数	104
7.1.2 函数的返回值.....	105
7.1.3 返回多个值	106
7.2 函数的参数名	108
7.2.1 外部参数名	108
7.2.2 默认参数值	109
7.2.3 可变参数	110
7.2.4 参数的类型	111
7.3 函数类型	113
7.3.1 函数类型作为变量的类型	115
7.3.2 函数类型作为参数的类型	116

7.3.3 函数类型作为返回值的类型	116
7.4 嵌套函数	117
7.5 局部与全局变量	118
习题	119
第8章 闭包	122
8.1 闭包表达式	122
8.1.1 推导类型格式	123
8.1.2 明确地从单一表达式的闭包返回	124
8.1.3 速记自变量名	125
8.1.4 运算符函数	126
8.2 尾随闭包	126
8.3 获取值	128
8.4 闭包是引用类型	129
习题	129
第9章 类、结构与枚举	133
9.1 类与结构的比较	133
9.1.1 值类型	135
9.1.2 引用类型	137
9.1.3 “==”与“!=”运算符	139
9.2 枚举的语法	141
9.2.1 在switch语句中使用枚举值	142
9.2.2 关联值	143
9.2.3 rawValue值	145
习题	146
第10章 属性与方法	149
10.1 存储型属性	149
10.2 计算属性	153
10.2.1 setter声明速记	155
10.2.2 只读计算属性	156
10.3 属性观察者	157
10.4 类型属性	159
10.5 实例方法	160
10.5.1 方法的局部与外部参数名称	161
10.5.2 self属性	163
10.5.3 修改值类型的实例方法	164
10.6 类型方法	165

习题	167
第11章 继承.....	171
11.1 父类.....	171
11.2 子类.....	173
11.3 重写.....	175
11.3.1 重写方法.....	175
11.3.2 重写访问的属性	177
11.3.3 重写属性观察者	180
习题	183
第12章 初始化与析构	188
12.1 初始化	188
12.2 类的继承与初始化	192
12.2.1 指定构造器与便捷构造器.....	192
12.2.2 语法与范例	194
12.3 析构.....	199
习题	200
第13章 自动引用计数	203
13.1 自动引用计数如何工作.....	203
13.2 类实例之间的强引用循环	204
13.3 解决类实例之间强引用循环的方法.....	207
13.3.1 弱引用	207
13.3.2 无主引用	210
13.3.3 无主引用与隐式解析可选属性	212
习题	213
第14章 可选链.....	216
14.1 可选链作为强制解析的方法.....	216
14.2 通过可选链调用属性、方法.....	217
14.2.1 通过可选链调用属性.....	218
14.2.2 通过可选链调用方法.....	220
14.3 多重链	221
习题	222
第15章 类型转换与扩展.....	224
15.1 检查类型	224
15.2 向下转型	226

15.3 对AnyObject和Any的类型转换.....	227
15.3.1 AnyObject.....	227
15.3.2 Any.....	228
15.4 扩展.....	229
15.4.1 属性的扩展.....	230
15.4.2 构造函数与方法的扩展.....	230
15.4.3 索引的扩展.....	232
习题.....	233
第16章 协议.....	237
16.1 属性的协议.....	237
16.2 方法的协议.....	239
16.3 作为类型的协议.....	241
16.4 协议以扩展方式加入.....	242
16.5 协议的继承.....	244
16.6 协议的组合.....	245
16.7 检查是否遵守协议.....	246
习题.....	248
第17章 泛型.....	252
17.1 泛型类型.....	252
17.1.1 两数对调.....	252
17.1.2 队列的运行.....	256
17.2 类型约束.....	264
17.2.1 查找某个值位于数组中的位置.....	264
17.2.2 气泡排序.....	267
17.3 关联类型.....	272
习题.....	277
第18章 位运算符与运算符函数.....	281
18.1 位运算符.....	281
18.1.1 用来判断与设置位的状态.....	284
18.1.2 实现乘除的功能.....	285
18.1.3 实现两数对调.....	286
18.2 运算符函数.....	288
18.2.1 prefix与 postfix运算符.....	288
18.2.2 组合赋值运算符.....	289
18.2.3 个性化运算符.....	293
习题.....	294

第 1 章

从简单的范例谈起

1.1 编写您的第一个程序

假设我们想编写一个简单的 Swift 程序，输出结果如下：

```
Learning Swift now!
```

安装完 Xcode 后，可以在应用程序中找到 Xcode 图标（icon），建议将 Xcode 图标拖动到 Dock 上，方便以后打开时使用。

打开后可以看见 Xcode 的欢迎画面，如图 1-1 所示。此处可以选择打开新的项目（Project），或打开之前编写的项目。



图 1-1 Xcode 欢迎画面

接着选择 Create a new Xcode project，以便打开一个新的项目。当然，也可以借助

系统 Xcode 的菜单，选取 File→New→Project…打开新的项目，如图 1-2 所示。

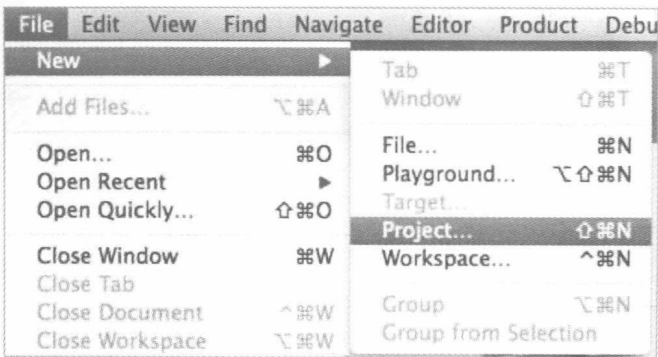


图 1-2 File 菜单下的选项

打开项目后，可选择要打开的项目类型，如图 1-3 所示。

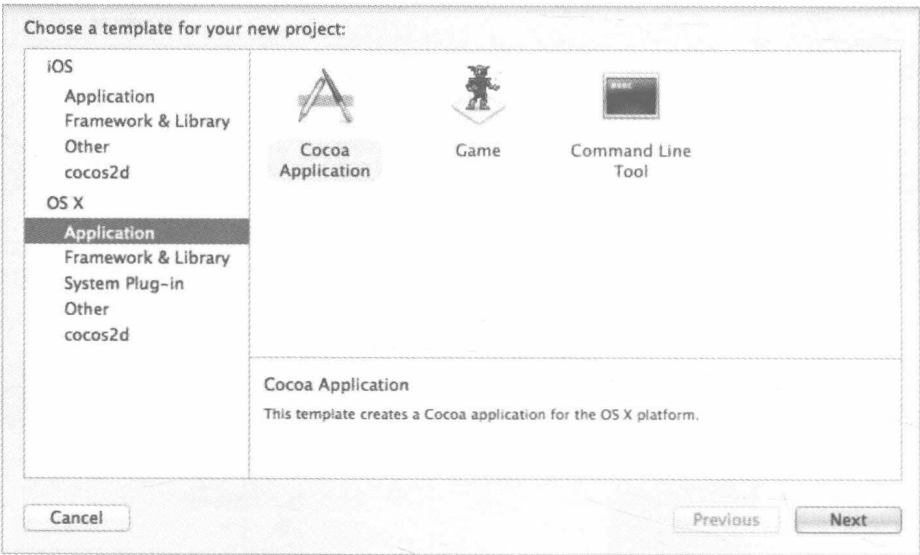


图 1-3 OS X 项目类型

OS X 的项目有许多类型可供选择，由于此处仅示范编写一个输出“Learning Swift now!”这一字符串的程序，因此，在左框中选择 OS X 下方的 Application，并在右框中选择 Command Line Tool（命令行工具），此项目仅介绍 Swift，所以使用 Command Line Tool。如果要编写有关 iPhone 程序的范例，此时在添加项目时需要使用 iOS 下方的 Application，如图 1-4 所示。

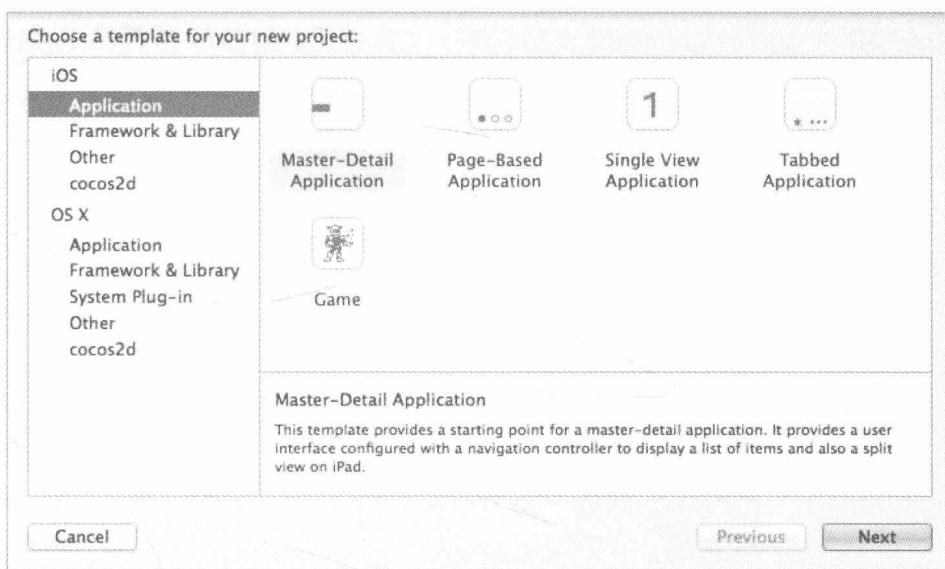


图 1-4 iOS 项目类型

当在如图 1-3 所示的对话框中选择 Command Line Tool，单击 Next 按钮后，画面如图 1-5 所示，请在 Project Name 中输入 myFirst，也在 Organization Name 和 Organization Identifier 中输入一个名称，这些都可以自行命名，以及 Language 的选项，请选取 Swift。

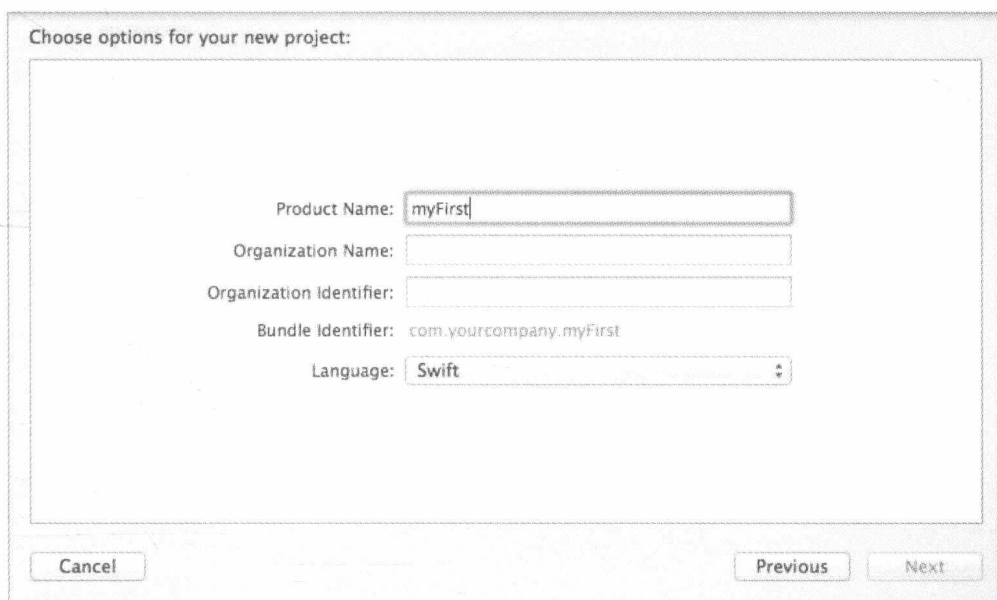


图 1-5 项目名称命名及编译语言选择

目前 Xcode 6 仍然提供对 Objective-C 语言的编译，但是由于目前要介绍的是 Swift，因此 Language 的选项请选择 Swift。

单击 Next 按钮后，请选择项目要存储的位置，位置可以任意选择，此范例存储在

已经事先在桌面中创建好的“Swift 范例程序”文件夹下，最后单击 Create 按钮，如图 1-6 所示。

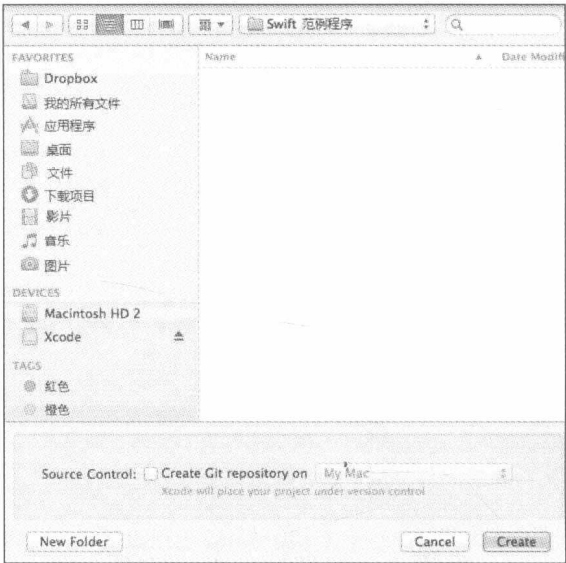


图 1-6 项目存储位置

完成后，将会出现如图 1-7 所示的编辑与运行画面，请单击 myFirst 下的 main.swift。



图 1-7 编辑与运行的画面

单击文件 main.swift，可以看见 Xcode 自动帮我们产生的程序模板，其中.swift 为 Swift 使用的扩展名。

接着做一点小小的修改，将下一行语句“println("Hello, world!")”改为：“println("Learning Swift now!")”，如图 1-8 所示。

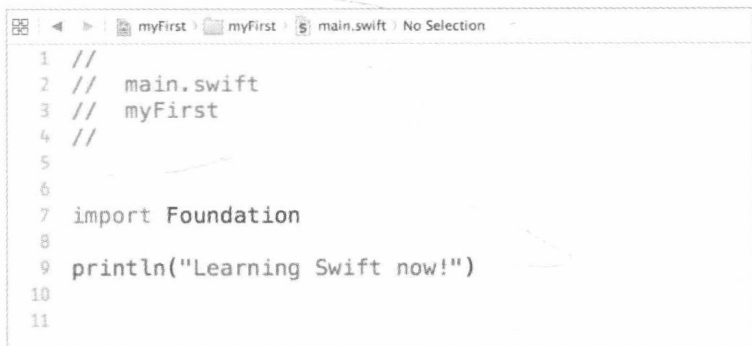


图 1-8 修改后的程序

println 函数是以双引号括起来的字符串为参数。字符串中出现什么就显示什么。详细说明请参阅第 2 章的内容。

值得一提的是，在 Swift 语言中每一行程序代码的结尾都不需要以分号“；”作为结束标记。

修改完成后，单击如图 1-7 所示的▶符号，Xcode 会开始编译与运行源代码。若无错误，则输出结果如图 1-9 所示。

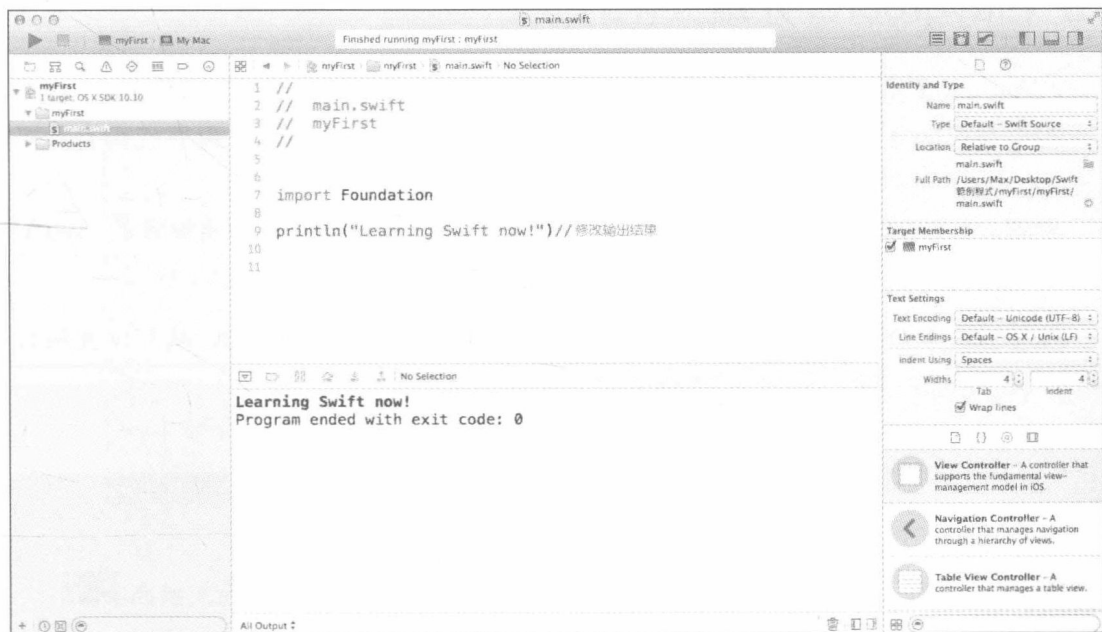


图 1-9 输出结果的画面

在图 1-9 的下方，我们所看到的输出结果是由 println 函数产生，如下所示：

Learning Swift now!

1.2 程序解析

程序的第一行是“import Foundation”，此语句的功能是加载所有对 Swift 有效的 Foundation API，包括 NSDate、NSURL、NSData，以及这些类的所有方法、属性以及对象。

如图 1-8 所示的程序代码都为系统自动产生，也就是说上述所列的语句可以不必加以理会，只要知道其功能即可，我们只修改了 println 函数这一行语句而已。

println 函数的参数是一个字符串，置于其中的文字将被输出，还可以使用 print 函数，它与 println 函数的差异是：print 函数输出完数据后不会换行，而 println 函数输出完数据后将跳到下一行。

Swift 与其他语言，如 C 或 Objective-C 不同的是，在每一行语句后面不需要加分号，表示此语句已结束。不过你在每个语句后加上分号也是可以的。一般而言，我们是不会加的。

本章只要知道如何创建一个项目，从而修改程序以符合需求，然后如何编译与运行就可以了，还有就是程序中所使用的英文字母是区分大小写的。

1.3 Playground 介绍

Playground 是 Xcode 6 中自带的 Swift 程序代码开发环境。以前 Xcode 5 对编写脚本程序代码的支持并不是很好，但对在 Playground 中编写 Swift 程序代码有了更好地支持。

使用 Playground 编写 Swift 程序代码，不需要编译或运行一个要编译的 Swift 程序，可以快速看到程序代码的运行过程以及运行结果。

步骤 1 首先打开 Xcode 6，直接单击 Get started with a playground，就可以直接打开一个 Playground 环境，如图 1-10 所示。



图 1-10 选取画面左边的第一项

步骤 2 接着设置 Playground 的名称和适用平台，在此处命名为 MyPlayground，并使用 iOS 的 Platform，然后单击 Next 按钮，如图 1-11 所示。

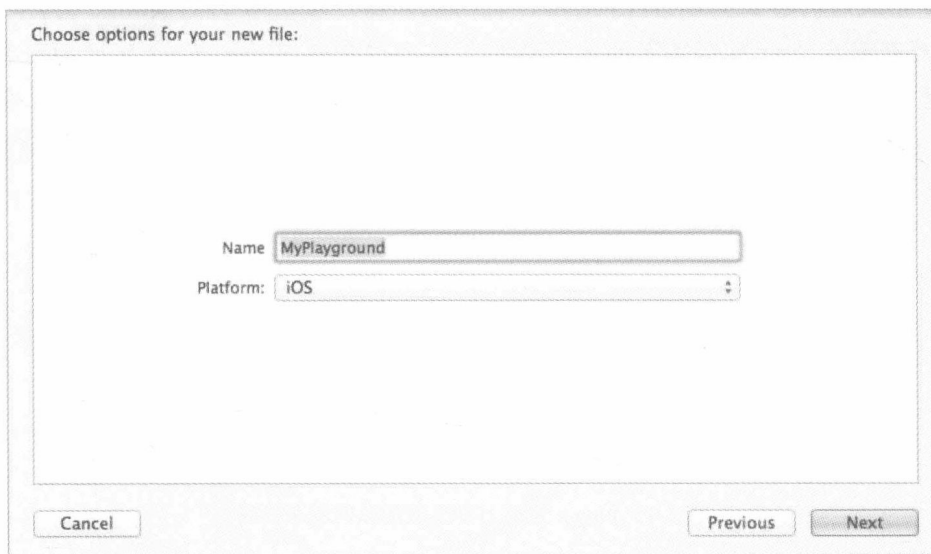


图 1-11 设置 Playground 的名称和适用平台

步骤 3 选择文件要存放的位置，并单击 Create 按钮，如图 1-12 所示。

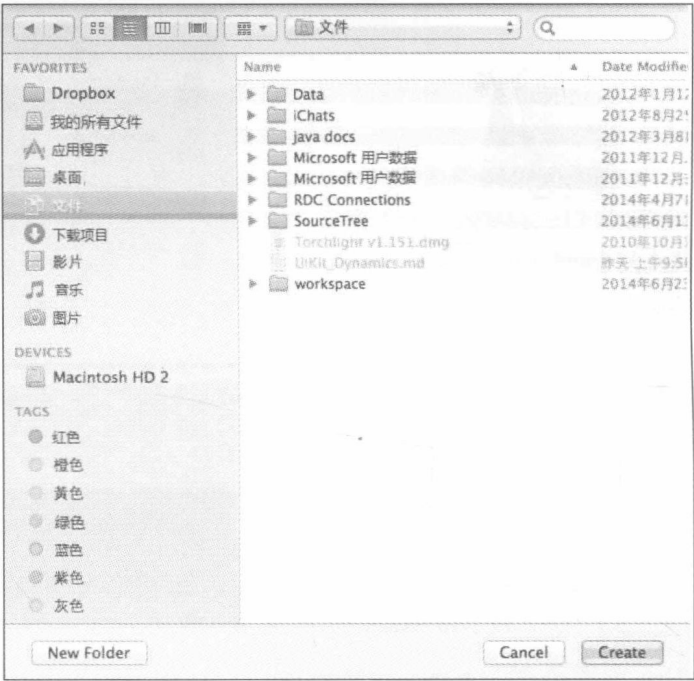


图 1-12 设置 Playground 的存放位置

步骤 4 完成后的界面如图 1-13 所示，我们就可以开始使用 Playground 的环境来编写 Swift 程序代码了。以自动生成的范例程序代码为例介绍 Playground 环境的内容。首先看到右边实时显示的部分（如图 1-14 所示），右侧有两个按钮，从左到右分别是 Quick Look 和 Value History，下面先介绍 Value History 按钮，单击 Value History 按钮之后会出现如图 1-15 所示的右侧区块。



图 1-13 Playground 自动生成的范例程序代码



图 1-14 Playground 的右边侧栏