

细致入门与最佳实践

案头宝典

完美搭配



陈强 ◎ 编著

中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

Swift

细致入门与最佳实践

陈强 编著



中国铁道出版社
CHINA RAILWAY PUBLISHING HOUSE

内 容 简 介

Swift 是苹果公司在 WWDC2014 大会上所发布的一门全新的编程语言, 用来编写 OS X 和 iOS 应用程序。随着苹果公司开发团队的不断努力, Swift 的版本也不断更新, 并且日趋稳定。本书基于稳定版本的 Swift 2.0 进行讲解, 详细介绍了开发 Swift 应用程序的知识点。本书共分 25 章, 循序渐进地讲解了 Swift 语言的基本语法知识, 并剖析了基于 Swift 开发 iOS 应用程序的方法。本书内容全面, 几乎涵盖了 Swift 开发的所有内容。全书内容言简意赅, 讲解方法通俗易懂、详细, 特别适合于初学者学习并消化。

本书适合 Swift 初学者、iOS 爱好者、iPhone 开发人员、iPad 开发人员、iOS 开发人员学习, 也可以作为相关培训学校和大、中专院校相关专业的教学用书。

图书在版编目 (CIP) 数据

Swift 细致入门与最佳实践 / 陈强编著. —北京:
中国铁道出版社, 2016.6
ISBN 978-7-113-21675-7

I. ①S… II. ①陈… 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 (2016) 第 066041 号

书 名: Swift 细致入门与最佳实践
作 者: 陈强 编著

责任编辑: 荆 波
责任印制: 赵星辰

读者热线电话: 010-63560056
封面设计: **MX** DESIGN
STUDIO

出版发行: 中国铁道出版社 (北京市西城区右安门西街 8 号 邮政编码: 100054)
印 刷: 北京明恒达印务有限公司
版 次: 2016 年 6 月第 1 版 2016 年 6 月第 1 次印刷
开 本: 787mm×1092mm 1/16 印张: 37.25 字数: 900 千
书 号: ISBN 978-7-113-21675-7
定 价: 79.00 元

版权所有 侵权必究

凡购买铁道版图书, 如有印制质量问题, 请与本社读者服务部联系调换。电话: (010) 51873174

打击盗版举报电话: (010) 51873659

Swift 是苹果公司在 WWDC2014 大会上所发布的一门全新的编程语言，用来编写 OS X 和 iOS 应用程序。苹果公司在设计 Swift 语言时，就有意将其与 Objective-C 共存，Objective-C 是苹果操作系统在导入 Swift 前使用的编程语言。随着苹果公司开发团队的不断努力，Swift 的版本也不断更新，并且日趋稳定。为了帮助读者迅速掌握 Swift 开发的核心技术知识，本书基于稳定版本的 Swift 2.0 进行讲解，详细介绍了 Swift 应用程序开发的知识点。

Swift 的优势

(1) 易学

作为一项苹果独立发布的支持型开发语言，Swift 语言的语法内容混合了 Objective-C、JS 和 Python，其语法简单、使用方便、易学，大大降低了开发者进入的门槛。同时 Swift 语言可以与 Objective-C 混合使用，对于用惯了高难度 Objective C 语言的开发者来说，Swift 语言更加易学。

(2) 功能强大

Swift 允许开发者通过更简洁的代码来实现更多的内容。在 WWDC2014 发布会上，工作人员演示了如何只通过一行简单的代码，完成一个完整图片列表加载的过程。另外，Swift 还可以让开发人员一边编写程序，一边预览应用程序，从而快速测试应用在某些特殊情况下的反应。

(3) 提升性能

Swift 语言可以提升程序性能，并同时降低开发难度，没有开发者不喜欢这样的编程语言。

(4) 简洁、精良、高效

Swift 是一种非常简洁的语言。与 Python 类似，不必编写大量代码即可实现强大的功能，并且有利于提高应用开发速度。Swift 可以更快捷有效地编译出高质量的应用程序。

(5) 执行速度快

Swift 的执行速度比 Objective-C 应用更快，这样会在游戏中看见更引人入胜的画面（需要苹果新的 Metal 界面的帮助），而其他应用也会有更好的响应性。与此同时，消费者不用购买新手机即可体验到这些效果。

(6) 全面融合

苹果对全新的 Swift 语言的代码进行了大量简化，在更快、更安全、更好的交互、更现代的同时，开发者们可以在同一款软件中同时使用 Objective-C、Swift、C 三种语言，实现了三类开发人员的完美融合。

(7) 测试工作更加便捷

方便快捷地测试所编写应用将帮助开发者更快地开发出复杂应用。以往对规模较大的应用来说，编译和测试过程极为冗繁，Swift 在这一方面带来较大的改进，应用开发者可以更快地发布经过彻底测试的应用。

本书特色

(1) 内容讲解循序渐进

本书从基础语法和搭建开发环境讲起,循序渐进地讲解了 Swift 语言开发的基本语法知识和核心应用技术。适合初学者学习。

(2) 结构合理

从用户的实际需要出发,科学安排知识结构,内容由浅入深,叙述清楚。全书详细地讲解了和 Swift 开发有关的知识点。

(3) 易学易懂

本书条理清晰、语言简洁,可帮助读者快速掌握每个知识点。使读者既可以按照本书编排的章节顺序进行学习,也可以根据自己的需求对某一章节进行针对性的学习。

(4) 实用性强

本书彻底摒弃枯燥的理论和简单的操作,注重实用性和可操作性,通过实例的实现过程,详细讲解了各个知识点的基本知识。

(5) 内容全面

本书可以号称市面中“内容最全面的一本 Swift 书”,无论是搭建开发环境,还是基本语法、面向对象、函数方法,在本书中您都能找到解决问题的答案。

本书参考内容

任何一门全新编程语言的推出,大家最初的学习资料往往是其官方资料。当苹果公司在 WWDC2014 大会发布 Swift 语言后,也一起公布了其参考使用文档《Swift Programming Language》,官方不但提供了此学习文档的 PDF 文件,而且提供了在线阅读版本。和广大初学者一样,笔者也将此使用文档作为学习 Swift 的第一手资料。

读者对象

iOS 开发初学者

Swift 初学者

大、中专院校的师生

毕业设计的学生

iOS 编程爱好者

相关培训机构的师生

从事 iOS 开发的程序员

本书在编写过程中得到了中国铁道出版社编辑的大力支持。正是各位编辑的求实、耐心和效率,才能使得本书得以出版。另外也十分感谢我的家人,在我写作的时候给予的巨大支持。另外,由于编者知识水平有限,书中如有纰漏和不尽如人意之处在所难免,诚请广大读者提出意见或建议,以便修订并使之更臻完善。

编者

2016 年 4 月

目 录

Contents

第 1 章 Swift 语言基础

1.1	Swift 概述	1
1.1.1	Swift 的创造者	1
1.1.2	Swift 的优势	2
1.2	搭建开发环境	3
1.2.1	Xcode 介绍	3
1.2.2	下载并安装 Xcode 7	4
1.3	使用 Xcode 开发环境	7
1.3.1	改变公司名称	7
1.3.2	通过搜索框缩小文件范围	8
1.3.3	格式化代码	8
1.3.4	代码缩进和自动完成	9
1.3.5	文件内查找和替代	10
1.3.6	快速定位到代码行	13
1.3.7	快速打开文件	13
1.3.8	使用书签	14
1.3.9	自定义导航条	15
1.3.10	使用 Xcode 帮助	16
1.3.11	调试代码	17
1.4	创建一个 iOS 9 项目	18
1.5	打开一个现有的 iOS 9 项目	23
1.6	第一段 Swift 程序	24
	实例 1-1 使用 Xcode 7 开发第一个 Swift 程序	24

第 2 章 Swift 的基础语法

2.1	分号	31
2.2	空白	32
2.3	标识符和关键字	32
2.3.1	标识符	32
2.3.2	关键字	33
2.4	常量和变量	34
2.4.1	声明常量	34

实例 2-1	输出常量的值	35
2.4.2	声明变量	35
实例 2-2	输出变量的值	37
2.4.3	输出常量和变量	39
实例 2-3	计算一个圆的面积	39
2.4.4	标注类型	40
实例 2-4	省略类型声明	40
2.4.5	常量和变量的命名规则	41
实例 2-5	修改变量值	41
2.5	注释	42
2.5.1	注释的规则	42
实例 2-6	演示注释的用法	43
2.5.2	使用注释的注意事项	43
2.6	数据类型	44
2.6.1	数据类型的分类	45
2.6.2	类型安全和类型推断	45
2.6.3	类型注解	46
2.6.4	类型标识符	46
2.6.5	元组类型	47
2.6.6	函数类型	47
2.6.7	数组类型	48
2.6.8	可选类型	48
2.6.9	隐式解析可选类型	49
2.6.10	协议合成类型	50
2.6.11	元类型	50
2.6.12	类型继承子句	50
2.6.13	类型推断	51
2.7	最基本的数值类型	51
2.7.1	整数	52
实例 2-7	输出整数	53
2.7.2	浮点数	54
实例 2-8	使用浮点数	54
2.8	字面量	55
2.8.1	数值型字面量	55
实例 2-9	使用数值型字面量	56

2.8.2 整型字面量..... 56

实例 2-10 输出不同进制的 17..... 57

2.8.3 浮点型字面量..... 57

2.8.4 文本型字面量..... 58

实例 2-11 演示特殊符号的使用..... 59

2.8.5 数值的可读性..... 59

2.9 数值型类型转换..... 60

2.9.1 整数转换..... 60

2.9.2 整数和浮点数转换..... 60

实例 2-12 演示整数和浮点数转换..... 61

2.9.3 进制的转换..... 61

2.10 类型别名..... 62

实例 2-13 演示类型别名的用法..... 63

2.11 布尔值..... 63

2.12 元组..... 65

2.12.1 定义元组类型..... 65

2.12.2 获取元组中的元素值..... 66

实例 2-14 演示元组的用法..... 66

2.13 可选类型..... 67

2.13.1 if 语句以及强制解析..... 68

实例 2-15 演示 if 语句的
强制解析..... 68

2.13.2 可选绑定..... 68

实例 2-16 使用可选绑定重写
实例 2-15..... 69

2.13.3 nil 空值..... 69

2.13.4 隐式解析可选类型..... 70

2.14 断言..... 71

2.14.1 使用断言进行调试..... 71

2.14.2 何时使用断言..... 72

2.15 综合演练..... 72

实例 2-17 综合演示 Swift 各个
基本语法的用法..... 72

第 3 章 字符、字符串和运算符

3.1 字符和字符串..... 74

3.1.1 字符和字符串基础..... 74

3.1.2 字符串字面量
(String Literals) 75

实例 3-1 演示各种特殊字符的
使用过程..... 75

3.1.3 初始化空字符串..... 76

实例 3-2 初始化空字符串
实例演示..... 76

3.1.4 字符串可变性..... 76

实例 3-3 使用换行符、双引号和
单引号..... 77

3.1.5 字符串是值类型..... 77

3.1.6 字符串遍历..... 78

实例 3-4 遍历字符串实例演示..... 78

3.1.7 计算字符数量..... 79

实例 3-5 设置字符值..... 79

3.1.8 连接字符串和字符..... 80

实例 3-6 连接字符串和字符
实例演示..... 80

3.1.9 字符串插值..... 81

3.1.10 比较字符串..... 81

实例 3-7 验证字符串是否相等..... 82

实例 3-8 实现字符串的大小
写转换..... 83

3.2 运算符概述..... 84

3.3 赋值运算符..... 85

3.3.1 基本赋值运算符..... 85

实例 3-9 使用基本的赋值运算符..... 86

3.3.2 复合赋值..... 86

实例 3-10 使用复合赋值运算符..... 87

3.4 算数运算符..... 87

3.4.1 单目运算符..... 88

实例 3-11 使用单目运算符..... 89

3.4.2 双目运算符..... 89

实例 3-12 使用双目运算符..... 90

3.4.3 求余运算..... 90

实例 3-13 使用求余运算符..... 91

3.4.4 浮点数求余计算..... 91

实例 3-14 使用浮点数求余运算符..... 92

3.5 比较运算符（关系运算符）..... 92

实例 3-15 使用比较运算符..... 93

3.6 三元条件运算..... 93

实例 3-16 使用三元条件运算符..... 94

3.7 区间运算符.....	94
3.7.1 闭区间运算符.....	95
实例 3-17 使用闭区间运算符.....	95
3.7.2 半闭区间运算符.....	95
实例 3-18 使用半闭区间运算符.....	95
3.8 逻辑运算.....	96
3.8.1 逻辑非.....	96
3.8.2 逻辑与.....	97
3.8.3 逻辑或.....	97
3.8.4 组合逻辑.....	97
3.8.5 使用括号设置运算优先级.....	98
实例 3-19 使用括号设置运算 优先级.....	98
3.9 位运算符.....	99
3.9.1 按位取反运算符.....	99
3.9.2 按位与运算符.....	100
3.9.3 按位或运算符.....	101
3.9.4 按位异或运算符.....	102
3.9.5 按位左移/右移运算符.....	103
实例 3-20 使用左移/右移运算符 ...	105
3.10 溢出运算符.....	106
实例 3-21 使用溢出运算符.....	106
3.11 运算符的优先级和结合性.....	109
实例 3-22 演示运算符的优先级 和结合性.....	111

第4章 集合类型

4.1 数组.....	113
4.1.1 定义数组.....	113
实例 4-1 定义一个数组.....	114
4.1.2 数组构造语句.....	114
4.1.3 访问和修改数组.....	115
实例 4-2 演示对数组的基本操作	117
4.1.4 数组的遍历.....	118
实例 4-3 实现对数组的遍历.....	119
4.1.5 创建并构造一个数组.....	119
实例 4-4 演示创建并且 构造一个数组.....	120
4.2 字典.....	122

4.2.1 字典字面量	122
实例 4-5 创建一个字典	123
4.2.2 读取和修改字典	124
实例 4-6 演示读取并修改字典	124
实例 4-7 对字典数据进行操作	125
实例 4-8 添加或修改字典数据	126
实例 4-9 在字典中移除键值对	127
4.2.3 字典遍历	128
实例 4-10 在字典中遍历数据	128
4.2.4 创建一个空字典	129
实例 4-11 实现字典复制操作	130
4.2.5 字典类型的散列值	130
4.3 集合的可变性	131
4.4 综合演练	131
实例 4-12 综合演练字典的操作	131

第5章 语句和流程控制

5.1 Swift 语句概述.....	133
5.1.1 循环语句	134
实例 5-1 简单演示使用 for 语句 遍历数组	134
实例 5-2 简单演示使用 while 语句	136
实例 5-3 演示 while 和 for 的对比	136
5.1.2 分支条件语句	138
实例 5-4 简单演示使用 if 分支语句	138
实例 5-5 简单演示使用 switch 分支语句	140
5.1.3 带标签的语句	141
5.1.4 控制传递语句	142
5.2 for 循环.....	143
5.2.1 for-in	143
实例 5-6 for in 语句应用: 打印 10 以内偶数	144
5.2.2 for 条件递增	145
实例 5-7 使用 for 条件递增语句	146
5.3 while 循环	147
5.3.1 While 语句	147

实例 5-8 while 循环语句应用:
100 以内 10 的倍数..... 149

5.3.2 do-while 语句..... 149

实例 5-9 do-while 循环语句应用:
蛇和梯子小游戏..... 150

5.4 条件语句..... 151

5.4.1 if 语句..... 151

实例 5-10 if 语句应用:
判断温度..... 151

5.4.2 switch 语句..... 152

实例 5-11 switch 语句应用: 匹配
名为 SomeCharacter 的
小写字符..... 153

5.5 控制转移语句..... 155

5.5.1 continue 语句..... 155

5.5.2 break 语句..... 155

实例 5-12 简单演示使用 break
语句..... 157

5.5.3 贯穿 (fallthrough)..... 157

5.5.4 带标签的语句 (labeled
statements)..... 158

第 6 章 函数

6.1 函数的定义..... 161

6.1.1 定义无参函数..... 162

6.1.2 定义有参函数..... 162

6.2 函数声明..... 164

6.2.1 函数声明的格式..... 164

实例 6-1 声明一个函数..... 165

6.2.2 声明中的参数名..... 165

实例 6-2 用 “...” 获取不固定
个数参数..... 166

6.2.3 声明中的特殊类型参数..... 167

6.3 函数调用..... 167

6.3.1 调用函数的格式..... 167

实例 6-3 调用定义的函数..... 168

6.3.2 函数调用的方式..... 169

实例 6-4 通过函数比较
两个数的大小..... 169

6.4 函数参数..... 170

6.4.1 多重输入参数..... 170

实例 6-5 演示多重输入
参数的用法..... 171

6.4.2 无参函数 (Functions Without
Parameters)..... 171

6.4.3 无返回值函数..... 171

6.5 返回值..... 172

实例 6-6 演示函数的返回值
的用法..... 173

6.6 函数参数的名称..... 174

6.6.1 外部参数名..... 174

6.6.2 简写外部参数名..... 175

6.6.3 默认参数值..... 175

6.6.4 默认值参数的外部参数名..... 176

6.6.5 可变参数..... 177

6.6.6 常量参数和变量参数..... 178

6.6.7 输入/输出参数..... 179

实例 6-7 编写函数求平均值..... 180

6.6.8 扩展参数..... 180

6.7 函数类型..... 181

6.7.1 使用函数类型..... 182

6.7.2 函数类型作为参数类型..... 182

实例 6-8 演示在函数中使用另一个
函数作为参数..... 183

6.7.3 函数类型作为返回类型..... 183

实例 6-9 演示将函数类型作为
返回类型..... 184

6.8 嵌套函数..... 185

实例 6-10 演示嵌套函数的用法..... 186

6.9 函数和闭包..... 186

实例 6-11 重写一个闭包来对所有
奇数返回 0..... 187

实例 6-12 在函数中定义函数..... 187

6.10 内置库函数..... 189

实例 6-13 查询 “.” 的位置..... 191

第 7 章 类

7.1 类和结构体基础..... 193

7.1.1 定义类和结构体..... 195

7.1.2	声明结构体字段.....	196
	实例 7-1 演示结构体的用法.....	196
7.2	类的成员.....	197
7.2.1	最简单的数据成员.....	198
7.2.2	最重要的函数成员.....	198
7.3	结构体成员.....	199
7.3.1	字段.....	199
	实例 7-2 创建结构体变量	
	赋初始值.....	199
7.3.2	函数.....	200
	实例 7-3 在结构体中直接	
	存储方法.....	200
7.3.3	属性.....	201
	实例 7-4 演示结构体属性的用法.....	201
7.4	类和结构体实例.....	201
	实例 7-5 定义并使用类的实例.....	202
7.5	类的继承.....	202
7.5.1	类的层次结构.....	202
7.5.2	继承概述.....	203
7.5.3	定义子类.....	203
	实例 7-6 创建 StudentLow 的子类	
	StudentHight.....	205
7.5.4	重写.....	206
	实例 7-7 演示面向对象、继承、重写	
	和构造.....	209
7.5.5	继承规则.....	210
7.6	属性访问.....	210
	实例 7-8 演示类中的属性、常量	
	和变量.....	210

第 8 章 构造函数和析构函数

8.1	构造函数概述.....	212
8.1.1	结构体中的构造函数.....	213
	实例 8-1 使用有参数构造函数.....	213
	实例 8-2 演示使用加了“_”标记的	
	参数构造函数.....	214
8.1.2	类中的构造函数.....	215
	实例 8-3 演示类中的构造函数.....	215
8.2	构造过程详解.....	217

8.2.1	为存储型属性赋初始值.....	218
	实例 8-4 演示为存储型属性赋	
	初始值.....	219
8.2.2	定制化构造过程.....	219
8.2.3	默认构造器.....	221
8.2.4	值类型的构造器代理.....	222
8.2.5	类的继承和构造过程.....	224
8.2.6	可失败构造器.....	231
8.3	析构函数.....	233
8.3.1	析构过程原理.....	233
	实例 8-5 在 Swift 中使用析构	
	函数.....	233
8.3.2	析构函数操作.....	234
8.4	综合演练.....	236
	实例 8-6 声明并调用 Swift 中各种	
	常用的函数.....	236

第 9 章 泛型

9.1	泛型所解决的问题.....	243
	实例 9-1 定义泛型.....	244
9.2	泛型函数.....	245
9.3	类型参数.....	246
	实例 9-2 使用函数和类支持泛型.....	246
9.4	命名类型参数和泛型类型.....	249
9.5	扩展一个泛型.....	252
9.6	类型约束.....	252
9.6.1	类型约束语法.....	253
9.6.2	类型约束行为.....	253
9.7	关联类型.....	255
9.7.1	关联类型行为.....	255
9.7.2	扩展一个存在的类型为一指定	
	关联类型.....	257
9.8	Where 语句.....	257

第 10 章 协议和扩展

10.1	协议的语法.....	259
	实例 10-1 定义并使用协议.....	260
10.2	对属性的规定.....	261
10.3	对方法的规定.....	263

10.4	对突变方法的规定.....	263
10.5	协议类型.....	264
10.6	委托（代理）模式.....	266
10.7	在扩展中添加协议成员.....	268
10.8	通过扩展补充协议声明.....	269
10.9	集合中的协议类型.....	269
10.10	协议的继承.....	269
10.11	协议合成.....	270
10.12	检验协议的一致性.....	271
10.13	对可选协议的规定.....	272
10.14	扩展详解.....	274
10.14.1	扩展语法.....	274
10.14.2	计算型属性.....	275
实例 10-2	演示计算型属性 的用法.....	275
10.14.3	构造器.....	276
实例 10-3	演示构造器的用法.....	277
10.14.4	扩展方法.....	278
实例 10-4	演示扩展方法的用法.....	278
10.14.5	下标.....	279
实例 10-5	演示下标的用法.....	279
10.14.6	嵌套类型.....	280
10.14.7	扩展字符串的用法.....	281
实例 10-6	演示扩展字符串 的用法.....	281

第 11 章 Swift 和 Objective-C 混编开发

11.1	在同一个工程中使用 Swift 和 Objective-C.....	282
11.1.1	Mix and Match 概述.....	282
11.1.2	在同一个应用的 target 中导入.....	283
11.1.3	在同一个 Framework 的 target 中导入.....	284
11.1.4	导入外部 Framework.....	285
11.1.5	在 Objective-C 中使用 Swift.....	285
11.1.6	实践练习.....	286

实例 11-1	在 Objective-C 中 调用 Swift.....	286
11.2	Swift 调用 C 语言函数.....	288
11.2.1	调用简单的 C 语言函数.....	288
实例 11-2	在 Swift 中调用简单的 C 语言函数.....	288
11.2.2	增加一个 C 语言键盘输入 函数.....	292
实例 11-3	演示增加 C 语言键盘 输入函数.....	292
11.3	Swift 调用 C 语言函数的综合演练.....	293
实例 11-4	综合演练调用 C 语言中的 各种函数.....	293

第 12 章 Xcode Interface Builder 界面开发

12.1	Interface Builder 基础.....	297
12.1.1	Interface Builder 的作用.....	297
12.1.2	Interface Builder 的新特色.....	298
12.2	Interface Builder 采用的方法.....	300
12.3	Interface Builder 的故事板.....	300
12.3.1	推出的背景.....	300
12.3.2	故事板的文档大纲.....	301
12.3.3	文档大纲的区域对象.....	303
12.4	创建一个界面.....	303
12.4.1	对象库.....	304
12.4.2	将对象加入视图中.....	305
12.4.3	使用 IB 布局工具.....	305
12.5	定制界面外观.....	308
12.5.1	使用属性检查器.....	309
12.5.2	设置辅助功能属性.....	309
12.5.3	测试界面.....	310

第 13 章 使用 Xcode 编写 MVC 程序

13.1	MVC 模式基础.....	312
13.1.1	诞生背景.....	312
13.1.2	分析结构.....	313
13.1.3	MVC 的特点.....	313

13.1.4 使用 MVC 实现程序设计的结构化	314
13.2 Xcode 中的 MVC	314
13.2.1 原理	314
13.2.2 模板就是给予 MVC 的	315
13.3 在 Xcode 中实现 MVC	316
13.3.1 视图	316
13.3.2 视图控制器	316
13.4 数据模型	318
13.5 综合演练	319
实例 13-1 使用 UISwitch 控件控制是否显示密码明文	319

第 14 章 基本组件

14.1 文本框 (UITextField)	324
14.1.1 文本框基础	324
14.1.2 实践练习	325
实例 14-1 为 TextField 添加震动效果	325
14.2 文本视图 (UITextView)	326
14.2.1 文本视图基础	326
14.2.2 实践练习	327
实例 14-2 显示 UITextView 中的文本	327
14.3 标签 (UILabel)	329
14.3.1 标签 (UILabel) 的属性	329
14.3.2 实践练习	329
实例 14-3 使用 UILabel 控件输出一个指定样式的文本	329
14.4 按钮 (UIButton)	331
14.4.1 按钮基础	331
14.4.2 实践练习	332
实例 14-4 自定义一个按钮	332
14.5 滑块控件 (UISlider)	334
实例 14-5 使用 UISlider 控件	334
14.6 步进控件 (UIStepper)	337
14.6.1 步进控件基础	337
14.6.2 实践练习	338

实例 14-6 使用步进控件自动增减数字	338
14.7 图像视图控件 (UIImageView)	339
14.7.1 UIImageView 的常用操作	340
14.7.2 实践练习	343
实例 14-7 使用 UIImageView 控件	344
14.8 开关控件 (UISwitch)	346
14.8.1 开关控件基础	346
14.8.2 实践练习	346
实例 14-8 使用 UISwitch 控件控制是否显示密码明文	346
14.9 分段控件 (UISegmentedControl)	349
14.9.1 分段控件基础	349
14.9.2 实践练习	350
实例 14-9 自定义 UISegmentedControl 控件的样式	350

第 15 章 提醒、操作表、工具栏和日期选择器

15.1 提醒视图 (UIAlertView)	351
15.1.1 UIAlertView 基础	351
15.1.2 实践练习	352
实例 15-1 演示如何使用 UIAlertView 控件	352
15.2 操作表 (UIActionSheet)	353
实例 15-2 使用 UIActionsheet 实现一个分享 App	353
15.3 工具栏 (UIToolbar)	356
15.3.1 工具栏基础	356
15.3.2 自定义工具栏	358
实例 15-3 使用 UIToolbar 控件制作自定义工具栏	358
15.4 选择器视图 (UIPickerView)	366
15.4.1 选择器视图基础	366
15.4.2 实践练习	367
实例 15-4 使用 UIPickerView 实现倒计时器	367

15.5 日期选择 (UIDatePicker)	376
15.5.1 UIDatePicker 基础	376
15.5.2 实践练习	377
实例 15-5 演示如何使用 UIDatePicker 控件	377

第 16 章 视图控制处理

16.1 Web 视图控件 (UIWebView)	382
16.1.1 Web 视图基础	382
16.1.2 实践练习	383
实例 16-1 加载指定的 HTML 网页并 自动播放网页音乐	383
16.2 可滚动视图控件 (UIScrollView)	386
16.2.1 UIScrollView 的基本用法	386
16.2.2 实践练习	388
实例 16-2 演示如何使用 UIScrollView 控件	388
16.3 翻页控件 (UIPageControl)	390
16.3.1 PageControll 控件基础	390
16.3.2 实践练习	391
实例 16-3 使用 UIPageControl 控件 设置四个界面	391
16.4 表视图 (UITableView)	394
16.4.1 表视图基础	394
16.4.2 添加表视图	395
16.4.3 UITableView 详解	397
16.4.4 实践练习	397
实例 16-4 在表视图中动态操作 单元格	397

第 17 章 活动指示器、进度条和检索条

17.1 活动指示器 (UIActivityIndicatorView)	403
17.1.1 活动指示器基础	403
17.1.2 实践练习	403
实例 17-1 演示如何使用 UIActivityIndicatorView 控件	404
17.2 进度条 (UIProgressView)	405

17.2.1 进度条基础	405
17.2.2 实践练习	406
实例 17-2 实现自定义进度条 效果	406
17.3 检索条 (UISearchBar)	412
17.3.1 检索条基础	412
17.3.2 实践练习	413
实例 17-3 演示如何使用 UISearchBar 控件	413

第 18 章 UIView 和视图控制器

18.1 UIView 基础	416
18.1.1 UIView 的结构	417
18.1.2 视图架构	418
18.1.3 视图层次和子视图管理	419
18.1.4 视图绘制周期	420
18.1.5 实践练习	420
实例 18-1 在 UIView 中创建一个 滚动图片浏览器	420
18.2 导航控制器 (UINavigationController) 简介	427
18.2.1 UINavigationController 基础	427
18.2.2 实践练习	428
实例 18-2 使用 UINavigationController 控件 创建会员登录系统	428
18.3 使用 UINavigationController	431
18.3.1 导航栏、导航项和栏 按钮项	433
18.3.2 UINavigationController 详解	433
18.3.3 在故事板中使用导航 控制器	434
18.3.4 实践练习	437
实例 18-3 创建主从关系的“主-子” 视图	437
18.4 选项卡栏控制器	439
18.4.1 选项卡栏和选项卡栏项	439
18.4.2 在选项卡栏控制器管理的场景 之间共享数据	441

18.4.3	UITabBarController	
	使用详解	441
18.4.4	实践练习	443
实例 18-4	开发一个界面选择器	443

第 19 章 图形、图像、图层和动画

19.1	图形处理	446
19.1.1	iOS 的绘图机制	446
19.1.2	实践练习	447
实例 19-1	使用 Quartz 2D 绘制移动的 曲线	447
19.2	图层	450
19.2.1	视图和图层	451
19.2.2	实践练习	451
实例 19-2	演示 CALayers 图层 的用法	451
19.3	实现动画	453
19.3.1	UIImageView 动画	453
19.3.2	视图动画 UIView	454
19.3.3	Core Animation 详解	454
19.3.4	实践练习	455
实例 19-3	图形图像的人脸 检测处理	455

第 20 章 多媒体应用

20.1	声音服务	461
20.1.1	声音服务基础	462
20.1.2	实践练习	462
实例 20-1	使用 AudioToolbox 播放 列表中的音乐	462
20.2	提醒和震动	474
20.2.1	播放提醒音	474
20.2.2	实践练习	474
实例 20-2	演示两种震动	474
20.3	Media Player 框架	476
20.3.1	Media Player 框架中的类	476
20.3.2	实践练习	476
实例 20-3	播放指定的视频	476
20.4	AV Foundation 框架	479

20.4.1	准备工作	480
20.4.2	实践练习	480
实例 20-4	使用 AVAudioPlayer 播放 和暂停指定的 MP3	480

20.5	图像选择器 (UIImagePickerController)	482
20.5.1	使用图像选择器	482
20.5.2	实践练习	482
实例 20-5	实现 UIImagePickerController 功能	482

第 21 章 定位处理

21.1	Core Location 框架	486
21.1.1	Core Location 基础	486
21.1.2	实践练习	487
实例 21-1	定位显示当前的 位置信息	487
21.2	获取位置	493
21.2.1	位置管理器委托	493
21.2.2	处理定位错误	494
21.2.3	位置精度和更新过滤器	494
21.2.4	获取航向	494
21.3	地图功能	495
21.3.1	Map Kit 基础	495
21.3.2	为地图添加标注	496
21.3.3	实践练习	496
实例 21-2	在地图中定位当前 的位置信息	496

第 22 章 和硬件之间的操作

22.1	CoreMotion 框架	500
22.1.1	CoreMotion 框架介绍	500
22.1.2	加速计基础	501
22.1.3	陀螺仪	502
22.1.4	实践练习	502
实例 22-1	使用 iPhone 中的 Motion 传感器	502
22.2	访问朝向和运动数据	505

22.2.1	通过 UIDevice 请求 朝向通知	505
22.2.2	使用 Core Motion 读取加速计 和陀螺仪数据	505
22.2.3	实践练习	506
实例 22-2	传感器综合练习: 海拔和 距离测试器	506

第 23 章 游戏开发

23.1	Sprite Kit 框架基础	518
23.1.1	Sprite Kit 的优点和缺点	518
23.1.2	Sprite Kit、Cocos2D、 Cocos2D-X 和 Unity 的选择	519
23.2	实践练习	519
实例 23-1	开发一个四子棋游戏	519

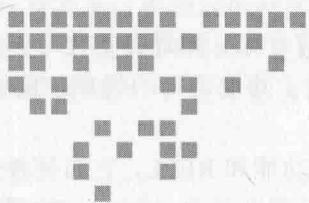
第 24 章 WatchKit 智能手表开发

24.1	Apple Watch 介绍	535
24.2	WatchKit 开发详解	537
24.2.1	WatchKit 架构	537
24.2.2	WatchKit 布局	540
24.2.3	Glances 和 Notifications	540
24.2.4	Watch App 的生命周期	540
24.3	开发 Apple Watch 应用程序	542
24.3.1	创建 Watch 应用	542
24.3.2	创建 Glance 界面	542
24.3.3	自定义通知界面	543
24.3.4	配置 Xcode 项目	543

24.4	实践练习	546
实例 24-1	开发一个综合性智能手表 管理系统	546
24.4.1	系统介绍	547
24.4.2	创建工程项目	547
24.4.3	iPhone 端的具体实现	549
24.4.4	Watch 端的具体实现	551

第 25 章 企业客服即时通信系统 (第三方框架+云存储)

25.1	即时通信系统介绍	559
25.2	系统模块结构	560
25.3	创建工程	561
25.4	使用 CocoaPods 配置第三方框架 ...	562
25.4.1	什么是 CocoaPods	562
25.4.2	CocoaPods 的核心组件	562
25.4.3	本项目的 CocoaPods	562
25.5	用户登录	563
25.5.1	登录主界面	563
25.5.2	新用户注册	568
25.6	系统聊天	571
25.7	UI 界面优化	573
25.7.1	文本框优化	573
25.7.2	HUD 优化	575
25.8	使用第三方框架	576
25.9	使用云存储保存系统数据	580
25.10	执行效果	581



Chapter 1 第 1 章

Swift 语言基础

Swift 是苹果公司在 WWDC2014 大会上所发布的一门全新的编程语言，用来编写 OS X 和 iOS 应用程序。苹果公司在设计 Swift 语言时，就有意将其与 Objective-C 共存，Objective-C 是苹果操作系统在导入 Swift 前使用的编程语言。在本章的内容中，将带领大家初步认识一下 Swift 这门神奇的开发语言，为读者步入本书后面知识的学习打下基础。

1.1 Swift 概述

Swift 是一种为开发 iOS 和 OS X 应用程序而推出的全新编程语言，是建立在 C 语言和 Objective-C 语言基础之上的，并且没有 C 语言的兼容性限制。Swift 采用安全模型的编程架构模式，使整个编程过程变得更容易、更灵活、更有趣。另外，Swift 完全支持市面中的主流框架：Cocoa 和 Cocoa Touch，这为开发人员重构软件和提高开发效率带来了很大的帮助。在本节的内容中，将带领大家一起探寻 Swift 的诞生历程。

1.1.1 Swift 的创造者

苹果 Swift 语言的创造者是苹果开发者工具部门总监 Chris Lattner 及其团队开发的，Chris Lattner 是 LLVM 项目的主要发起人和作者之一，Clang 编译器的作者。LLVM 是一种用于优化编译器的基础框架，能将高级语言转换为机器语言。LLVM 极大提高了高级语言的效率，Chris Lattner 也因此获得了首届 SIGPLAN 奖。

2005 年，Chris Lattner 加入 LLVM 开发团队，正式成为苹果公司的一名员工。在苹果公司的 9 年间，他由一名架构师一路升职为苹果开发者工具部门总监。目前 Chris Lattner 主要负责 Xcode 项目，这也为 Swift 的开发提供了灵感。

Chris Lattner 从 2010 年 7 月开始开发 Swift 语言，当时它在苹果内部属于机密项目，只有很少人知道这一语言的存在。Chris Lattner 在个人博客上称，Swift 的底层架构大多是他自己

开发完成的。2011 年，其他工程师开始参与项目开发，Swift 也逐渐获得苹果公司内部的重视，直到 2013 年成为苹果主推的开发工具。

Swift 的开发结合了众多工程师的心血，包括语言专家、编译器优化专家等，苹果其他团队也为改进产品提供了很大帮助。同时 Swift 也借鉴了其他语言的优点，例如 Objective-C、Rust、Ruby 等。

Swift 语言的核心吸引力在于 Xcode Playgrounds 功能和 REPL，它们使开发过程具有更好的交互性，也更容易上手。Playgrounds 在很大程度上受到了 Bret Victor 的理念和其他互动系统的启发。同样，具有实时预览功能的 Swift 使编程变得简单，学习起来也更加容易，目前已经引起了开发者的极大兴趣。这有助于苹果吸引更多的开发者，甚至将改变计算机科学的教学方式。图 1-1 是 Chris Lattner 在 WWDC14 大会上对 Swift 进行演示。

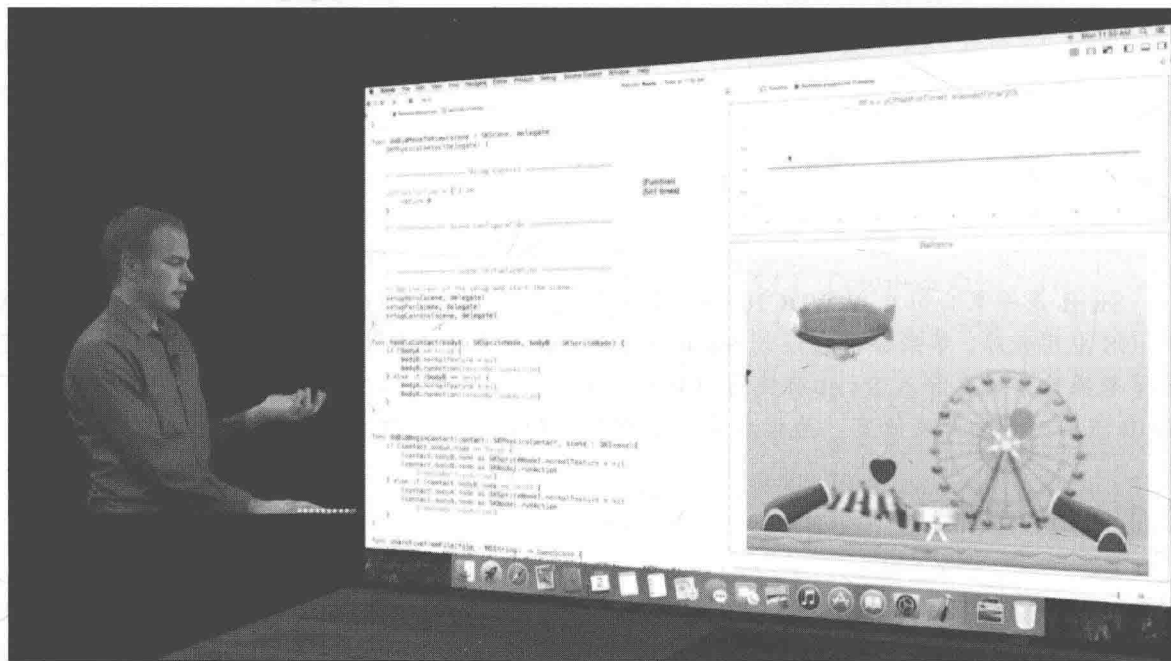


图 1-1 Chris Lattner 在 WWDC14 大会上对 Swift 进行演示

1.1.2 Swift 的优势

在 WWDC2014 大会中，苹果展示了如何能让开发人员更快地进行代码编写及显示结果的“Swift Playground”，在左侧输入代码的同时，可以在右侧实时显示结果。苹果公司表示 Swift 是基于 Cocoa 和 Cocoa Touch 而专门设计的。Swift 不仅可以用于基本的应用程序编写，比如各种社交网络 App，同时还可以使用更先进的“Metal”3D 游戏图形优化工作。由于 Swift 可以与 Objective-C 兼容使用，因此开发人员可以在开发过程中进行无缝切换。

具体来说，Swift 语言的突出优势如下所示。

(1) 易学

作为一项苹果独立发布的支持型开发语言，Swift 语言的语法内容混合了 Objective-C、JS 和 Python，其语法简单、使用方便、易学，极大降低了开发者入门的门槛。同时 Swift 语言可以与 Objective-C 混合使用，对于用惯了高难度 Objective C 语言的开发者来说，Swift 语言更加易学。