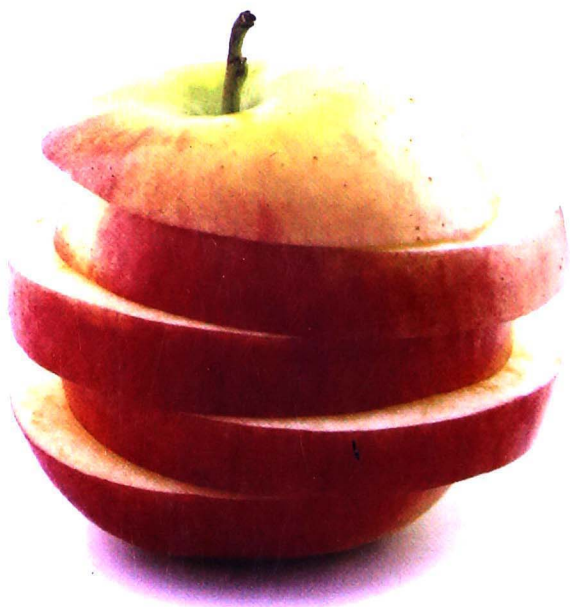




Objective-C 函数速查实例手册

张权 编著

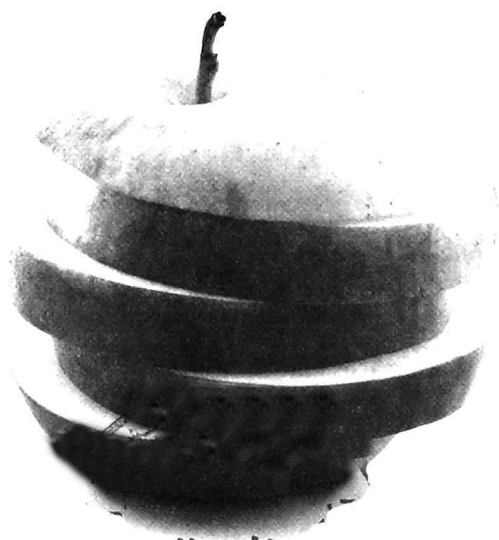
内容全面 全书涵盖756个方法

全程视频 论坛提供27小时讲解视频下载

完整测试 全部程序通过iOS 7模拟器和iPhone 5真机测试

实例众多 全书包括656个例子，快速入门

 人民邮电出版社
POSTS & TELECOM PRESS



Objective-C

函数速查实例手册

张权 编著

人民邮电出版社
北京

图书在版编目 (C I P) 数据

Objective-C函数速查实例手册 / 张权编著. -- 北京: 人民邮电出版社, 2014.2
ISBN 978-7-115-33641-5

I. ①O… II. ①张… III. ①C语言—程序设计—手册
IV. ①TP312-62

中国版本图书馆CIP数据核字(2013)第266582号

内 容 提 要

Objective-C 是一种通用、高级、面向对象的编程语言, 是 iOS 平台以及相关开发的主流编程语言。在 TIOBE 最新的编程语言排行榜中, Objective-C 仅次于 C 和 Java, 位居第三。

本书全面、系统地介绍了 Objective-C 中常用的函数和方法, 并提供了大量实例, 帮助读者掌握其用法。全书共分 3 篇。第 1 篇介绍 Objective-C 的发展历程、特点、技术构成、Objective-C 开发环境的构建方法、第一个 Objective-C 程序数学运算中常用的函数等; 第 2 篇介绍数字对象、字符串对象、数组对象、字典对象、集合、文件、时间和日历、进程、线程、锁、数据对象和归档中常用的方法; 第 3 篇介绍窗口和颜色、自定义视图、文本框和文本视图、图像和图像视图、表视图以及常用控件中常用的方法。此外, 作者专门为本书录制了大量的配套教学视频, 以帮助读者更好地学习本书内容。这些视频和书中的实例源代码一起收录于本书网络论坛上, 供读者下载。

本书涉及面广, 从基本的 Objective-C 语言函数, 到 Cocoa 中两个框架的方法, 几乎涵盖 Objective-C 开发所用到的所有方法。本书适合所有想全面学习 Objective-C 开发技术的人员阅读, 也适合各种使用 Objective-C 进行开发的程序员参考查阅。对于 Objective-C 程序员来说, 本书是一本不可多得的案头必备的参考用书。

◆ 编 著 张 权

责任编辑 陈冀康

责任印制 程彦红 杨林杰

◆ 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>

北京艺辉印刷有限公司印刷

◆ 开本: 880×1230 1/32

印张: 16.625

字数: 559 千字

印数: 1-3 000 册

2014 年 2 月第 1 版

2014 年 2 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: 京崇工商广字第 0021 号

前言

从1980年布莱德·确斯(Brad Cox)在其公司Stepstone发明Objective-C至今,已有33年了。在这期间,Objective-C语言以其高效的特点在苹果操作系统中经久不衰。Objective-C语言是通用、高级、面向对象的编程语言,所以被广泛应用于苹果的OS X和iOS操作系统及其相关API、Cocoa和Cocoa Touch的程序编写中。

作者结合自己多年的Objective-C开发经验和心得体会,花费了一年多的时间编写本书,希望各位读者能在本书的引领下跨入Objective-C开发大门,并成为一名开发高手。本书内容全面、系统,并以大量实例贯穿于全书的讲解之中,同时本书网络论坛提供大量多媒体教学视频,相信读者学习完本书,可以具备更高一级的项目开发的能力。

本书特色

1. 配有大量多媒体语音教学视频,学习效果好。

作者专门录制了大量的配套多媒体语音教学视频,以便让读者更加轻松、直观地学习本书内容,提高学习效率。这些视频与本书源代码可访问<http://rzchina.net/forum-323-1.html>。

2. 内容全面、系统。

本书介绍了Objective-C的特点、技术构成、开发Objective-C时的开发环境、Cocoa中的Foundation框架和Application框架中常用到的方法等内容。

3. 贯穿大量的实例,迅速提升开发水平。

本书在讲解知识点时贯穿了大量短小精悍的典型实例,以便让读者更好地理解各种方法,体验实际编程,迅速提高开发水平。

本书内容及结构

第1篇 Objective-C准备篇(第1~2章)

本篇主要包括: Objective-C的发展历程、特点、技术构成、Objective-C开发环境的构建、第一个Objective-C程序的编写、数学运算中常用到的函数等。通过本篇的学习,读者可以熟悉Objective-C的开发环境掌握数学运算中的函数使用方法。

第2篇 Cocoa 的 Foundation 框架篇（第3~11章）

本篇主要内容包括：数字对象、字符串对象、数组对象、字典对象、集合、文件、时间和日历、进程、线程、锁、数据对象和归档中常用的方法。

第3篇 Cocoa 的 Application 框架篇（第12~17章）

本篇主要内容包括：窗口和颜色、自定义视图、文本框和文本视图、图像和图像视图、表视图以及常用控件中常用方法。

本书读者对象

- ☐ 想全面学习 Objective-C 开发技术的人员
- ☐ Objective-C 专业开发人员
- ☐ Objective-C 开发爱好者
- ☐ 大中专院校的学生
- ☐ 社会培训班学员
- ☐ 需要一本案头手册的程序员

约定

在运行结果中，“J”之前的是运行时间和运行状态，之后的是运行的输出结果；读者重点看“J”之后的内容。

由于部分函数/方法功能类似，仅参数和返回值的类型不同，对于这类方法，本书将其归纳到一个小节中，只对第一个方法进行讲解。读者只需要做简单对照，就可以掌握所有类似的函数。

编者

目 录

第 1 篇 Objective-C 准备篇

第 1 章 Objective-C 学习环境准备	3
1.1 Objective-C 基础	3
1.1.1 Objective-C 的发展历程	3
1.1.2 Objective-C 语言的特点	3
1.1.3 技术架构	4
1.2 开发环境	5
1.2.1 注册开发者账号	5
1.2.2 下载安装 Xcode	8
1.2.3 更新帮助文档	12
1.3 编写第一个 Objective-C 程序	13
1.3.1 创建项目	13
1.3.2 编写程序	15
1.3.3 运行程序	15
1.3.4 使用帮助	16
1.4 小结	17
第 2 章 数学运算 (math) 函数	18
2.1 算术运算函数	18
2.1.1 rand()函数: 产生随机数	18
2.1.2 abs()函数/labs()函数: 整数的绝对值	19
2.1.3 fabs()/fabsf()/fabsl()函数: 浮点数的绝对值	19
2.1.4 floor()/floorf()/floorl()函数: 向下取整	20
2.1.5 ceil()/ceilf()/ceill()函数: 向上取整	21
2.1.6 round()/roundf()/roundl()函数: 四舍五入	22

2.1.7	sqrt()/sqrtf()/sqrtl()函数：求平方根	22
2.1.8	fmax()/fmaxf()/fmaxl()函数：求最大值	23
2.1.9	fmin()/fminf()/fminl()函数：求最小值	24
2.1.10	hypot()/hypotf()/hypotl()函数：求直角三角形斜边的长度	25
2.1.11	fmod()/fmodf()/fmodl()函数：求两数整除后的余数	26
2.1.12	modf()/modff()/modfl()函数：浮点数分解为整数和小数	27
2.1.13	frexp()/frexpf()/frexpl()函数：浮点数分解尾数和以二为底的指数	28
2.2	三角运算函数	29
2.2.1	sin()/sinf()/sinl()函数：求正弦值	29
2.2.2	sinh()/sinhf()/sinhl()函数：求双曲正弦值	30
2.2.3	cos()/cosf()/cosl()函数：求余弦值	30
2.2.4	cosh()/coshf()/coshl()函数：求双曲余弦值	31
2.2.5	tan()/tanf()/tanl()函数：求正切值	31
2.2.6	tanh()/tanhf()/atanhl()函数：求双曲正切值	32
2.3	反三角运算	33
2.3.1	asin()/asinf()/asinl()函数：求反正弦值	33
2.3.2	asinh()/asinhf()/asinh()函数：求反双曲正弦值	34
2.3.3	acos()/acosf()/acosl()函数：求反余弦值	34
2.3.4	acosh()/acoshf()/acoshl()函数：求反双曲余弦值	35
2.3.5	atan()/atanf()/atanl()函数：求反正切值	36
2.3.6	atan2()/atan2f()/atan2l()函数：求坐标值的反正切值	37
2.3.7	atanh()/atanhf()/atanhl()函数：求反双曲正切值	37
2.4	指数和对数运算	38
2.4.1	pow()/powf()/powl()函数：求 n 的 m 次方的值	38
2.4.2	exp()/expf()/expl()函数：求 e 的 x 次方的值	39
2.4.3	exp2()/exp2f()/exp2l()函数：2 的 x 的次方的值	39
2.4.4	log()/logf()/logl()函数：求以 e 为底的对数值	40
2.4.5	log10()/log10f()/log10l()函数：求以 10 为底的对数值	41
2.5	常量函数	41
2.6	小结	42

第2篇 Cocoa 的 Foundation 框架篇

第3章 数字对象 (NSNumber)	45
3.1 整型对象	45
3.1.1 numberWithShort:方法: 创建短整型数字对象	45
3.1.2 numberWithUnsignedShort:方法: 创建无符号 短整型数字对象	45
3.1.3 numberWithInteger:方法: 创建整型数字对象	45
3.1.4 numberWithUnsignedInteger:方法: 创建无符号 整型数字对象	46
3.1.5 numberWithInt:方法: 创建整型数字对象	46
3.1.6 numberWithUnsignedInt:方法: 创建无符号 整型数字对象	46
3.1.7 numberWithLong:方法: 创建并初始化长整型数字对象	46
3.1.8 numberWithUnsignedLong:方法: 创建并初始化 无符号长整型数字对象	46
3.1.9 numberWithLongLong:方法: 创建并初始化 长长整型数字对象	47
3.1.10 numberWithUnsignedLongLong:方法: 创建并初始化无符号长长整型对象	47
3.1.11 initWithShort:方法: 初始化短整型数字对象	47
3.1.12 initWithUnsignedShort:方法: 初始化 无符号短整型数字对象	47
3.1.13 initWithInteger:方法: 初始化整型数字对象	47
3.1.14 initWithUnsignedInteger:方法: 初始化 无符号整型数字对象	47
3.1.15 initWithInt:方法: 初始化整型数字对象	48
3.1.16 initWithUnsignedInt:方法: 初始化无符号整型数字对象	48
3.1.17 initWithLong:方法: 初始化长整型数字对象	48
3.1.18 initWithUnsignedLong:方法: 初始化无符号 长整型数字对象	48

3.1.19	initWithLongLong:方法: 初始化长长整型数字对象	48
3.1.20	initWithUnsignedLongLong:方法: 初始化无符号 长长整型数字对象	48
3.1.21	shortValue:方法: 取短整型数字对象的值	49
3.1.22	unsignedShortValue:方法: 取无符号短整型 数字对象的值	49
3.1.23	integerValue:方法: 取整型数字对象的值	49
3.1.24	unsignedIntegerValue:方法: 取无符号整型 数字对象的值	49
3.1.25	intValue:方法: 取整型数字对象的值	49
3.1.26	unsignedIntValue:方法: 取无符号整型数字对象的值	49
3.1.27	longValue:方法: 取长整型数字对象的值	49
3.1.28	unsignedLongValue:方法: 取无符号 长整型数字对象的值	50
3.1.29	longlongValue:方法: 取长长整型数字对象的值	50
3.1.30	unsignedLongLongValue:方法: 取无符号长长 整型数字对象的值	50
3.2	字符型对象	51
3.2.1	numberWithChar:方法: 创建并初始化字符型数字对象	51
3.2.2	numberWithUnsignedChar:方法: 创建并初始化 无符号字符型数字对象	51
3.2.3	initWithChar:方法: 初始化字符型对象	51
3.2.4	initWithUnsignedChar:方法: 初始化无符号字符型对象	52
3.2.5	charValue:方法: 取字符型数字对象的值	52
3.2.6	unsignedCharValue:方法: 取无符号 字符型数字对象的值	52
3.3	单精度型对象	53
3.3.1	numberWithFloat:方法: 创建并初始化单精度型数字对象	53
3.3.2	initWithFloat:方法: 初始化单精度型数字对象	53
3.3.3	floatValue:方法: 取单精度型数字对象的值	53
3.4	双精度型对象	54
3.4.1	numberWithDouble:方法: 创建并初始化 一个双精度型数字对象	54

3.4.2	initWithDouble:方法: 初始化双精度型数字对象	54
3.4.3	doubleValue:方法: 取双精度型数字对象的值	54
3.5	布尔型对象	55
3.5.1	numberWithBool:方法: 创建并初始化布尔型数字对象	55
3.5.2	initWithBool:方法: 初始化布尔类型对象	56
3.5.3	boolValue:方法: 取布尔型数字对象的值	56
3.6	数字对象通用方法	57
3.6.1	isEqualToNumber:方法: 比较两对象值是否相等	57
3.6.2	compare:方法: 比较值的大小	57
3.7	小结	58
第 4 章	字符串对象 (NSString)	59
4.1	创建及初始化	59
4.1.1	string:方法: 创建字符串	59
4.1.2	stringWithString:方法: 用字符串来创建字符串	59
4.1.3	stringWithCString:方法: 创建 C 字符串	60
4.1.4	stringWithFormat:方法: 创建 NSLog()格式的字符串	60
4.1.5	stringWithContentsOfFile:方法: 将创建的字符串 设置为指定文件的内容	61
4.1.6	stringWithContentsOfURL:方法: 将创建字符串 设置为 url 的内容	62
4.1.7	stringWithUTF8String:方法: 将创建的字符串 转换为 UTF8 字符串	62
4.1.8	init:方法: 初始化字符串	63
4.1.9	initWithString:方法: 用字符串来初始化字符串	63
4.1.10	initWithCString:方法: 初始化字符串	64
4.1.11	initWithFormat:方法: 用 NSLog()格式初始化字符串	64
4.1.12	initWithContentsOfFile:方法: 将初始化的 字符串设置为指定文件的内容	65
4.1.13	initWithContentsOfURL:方法: 将初始化的 字符串设置为 url 的内容	66
4.1.14	initWithUTF8String:方法: 将初始化的 字符串转换为 UTF8 字符串	66

4.2	判断和比较	67
4.2.1	isEqualToString:方法: 比较字符串是否相等	67
4.2.2	hasPrefix:方法: 判断字符串是否以某个字符开始	68
4.2.3	hasSuffix:方法: 判断字符串是否以某个字符结束	68
4.2.4	compare:方法: 比较字符串的大小	69
4.2.5	caseInsensitiveCompare:方法: 不考虑大小写的比较大小	70
4.3	大小写转换	71
4.3.1	uppercaseString:方法: 小写字母转为大写字母	71
4.3.2	lowercaseString:方法: 大写字母转为小写字母	72
4.3.3	capitalizedString:方法: 将每个单词的首字母大写	72
4.4	截取	73
4.4.1	substringToIndex:方法: 从字符串的 开头一直截取到指定的位置	73
4.4.2	substringFromIndex:方法: 从指定位置开始 截取字符串直到结束	73
4.4.3	substringWithRange:方法: 根据指定范围返回子字符串	74
4.4.4	characterAtIndex:方法: 返回索引号 所在字符串中的字符	75
4.5	转换类型	75
4.5.1	doubleValue:/floatValue:方法: 返回转换为浮点类型的值	75
4.5.2	intValue:方法: 返回转换为整型的值	76
4.5.3	boolValue:方法: 返回转换为布尔类型的值	76
4.6	对文件的处理	77
4.6.1	stringByAppendingPathExtension:方法: 为文件添加扩展名	77
4.6.2	pathExtension:方法: 获取文件扩展名	78
4.6.3	stringByDeletingPathExtension:方法: 删除扩展名	78
4.6.4	writeToFile:方法: 将字符串写入到文件	79
4.6.5	writeToURL:方法: 将字符串写入到 url 中	79
4.6.6	stringByExpandingTildeInPath:方法: 将“~”替换成系统的主目录	80
4.6.7	stringByAbbreviatingWithTildeInPath:方法: 将系统主目录替换为“~”	81

4.6.8	lastPathComponent:方法: 获取路径中的文件名	81
4.6.9	stringByDeletingLastPathComponent:方法: 获取路径中文件所在的位置	82
4.6.10	stringByAppendingPathComponent:方法: 组合位置和文件名	82
4.6.11	isAbsolutePath:方法: 判断绝对路径	83
4.7	其他	84
4.7.1	length:方法: 求字符串的长度	84
4.7.2	stringByAppendingString:方法: 字符串后面增加 一个新字符串	84
4.7.3	rangeOfString:方法: 查找字符串中是否包含 其他字符串	85
4.7.4	stringByTrimmingCharactersInSet:方法: 去除空格或回车	86
4.8	可变字符串	86
4.8.1	stringWithCapacity:方法: 按照固定长度生成空字符串	86
4.8.2	initWithCapacity:方法: 初始化一个固定长度的字符串	87
4.8.3	appendString:方法: 在字符串的末尾附加另一个字符串	87
4.8.4	appendFormat:方法: 附加一个格式化字符串	88
4.8.5	SetString:方法: 将字符串设置为规定的内容	88
4.8.6	insertString:方法: 在指定位置插入字符串	89
4.8.7	deleteCharactersInRange:方法: 删除指定范围的字符串	89
4.8.8	replaceCharactersInRange:方法: 使用字符串 代替指定范围的字符串	90
4.8.9	replaceOccurrencesOfString:方法: 替换	91
4.9	小结	91
第 5 章	数组对象 (NSArray)	92
5.1	创建及初始化	92
5.1.1	array:方法: 创建数组	92
5.1.2	arrayWithArray:方法: 通过一个数组创建另一个数组	92
5.1.3	arrayWithContentsOfFile:方法: 创建数组将内容 设置为指定文件内容	93

5.1.4	arrayWithContentsOfURL:方法: 创建数组 将内容设置为 url 指定内容	94
5.1.5	arrayWithObject:方法: 创建具有一个元素的数组	95
5.1.6	arrayWithObjects:方法: 创建具有多个元素的数组	95
5.1.7	init:方法: 初始化数组	96
5.1.8	initWithArray:方法: 用数组初始化数组	96
5.1.9	initWithContentsOfFile:方法: 初始化数组 将内容设置为指定文件内容	97
5.1.10	initWithContentsOfURL:方法: 初始化数组 将内容设置为 url 指定内容	98
5.1.11	initWithObjects:方法: 初始化具有多个元素的数组	98
5.2	数组元素的操作	99
5.2.1	containsObject:方法: 判断数组中是否包含某个元素	99
5.2.2	count:方法: 计算元素个数	100
5.2.3	firstObjectCommonWithArray:方法: 获取首元素	100
5.2.4	lastObject:方法: 获取最后一个元素	101
5.2.5	objectAtIndex:方法: 获取在某个位置的数组元素	101
5.2.6	objectAtIndexes:方法: 获取数组元素	102
5.2.7	arrayByAddingObject:方法: 在数组末尾添加元素	102
5.2.8	arrayByAddingObjectsFromArray:方法: 在数组的末尾添加另一个数组	103
5.2.9	subarrayWithRange:方法: 数组的一部分创建数组	104
5.2.10	isEqualToArray:方法: 比较数组是否相等	104
5.2.11	indexOfObject:方法: 返回元素所在的位置	105
5.2.12	indexOfObjectIdenticalTo:方法: 返回元素所在的位置	106
5.2.13	componentsJoinedByString:方法: 数组转换为字符串	106
5.2.14	componentsSeparatedByString:方法: 字符串转换为数组	107
5.2.15	sortedArrayHint:方法: 数组转换为数据对象	108
5.2.16	writeToFile:方法: 将数组中的内容写入到文件	108
5.2.17	writeToUrl:方法: 将数组中的内容写入到 url	109
5.2.18	objectEnumerator:方法: 数组元素从前向后访问	110

5.2.19	reverseObjectEnumerator:方法: 数组元素从后向前访问	111
5.2.20	pathsMatchingExtensions:方法: 查看某文件夹下的东西	112
5.2.21	sortedArrayUsingFunction:方法: 实现数组元素的简单排序	112
5.3	可变数组	113
5.3.1	arrayWithCapacity:方法: 创建一个具有 固定长度的可变数组	113
5.3.2	initWithCapacity:方法: 初始化一个具有 固定长度的可变数组	114
5.3.3	addObject:方法: 添加数组元素	114
5.3.4	addObjectsFromArray:方法: 用数组创建可变数组	115
5.3.5	removeObject:方法: 删除指定的元素	116
5.3.6	removeAllObjects:方法: 删除可变数组中的所有元素	116
5.3.7	removeLastObject:方法: 删除可变数组中的 最后一个元素	117
5.3.8	removeObjectAtIndex:方法: 删除指定位置的元素	118
5.3.9	removeObjectsAtIndex:方法: 删除可变数组中的元素	119
5.3.10	removeObjectsInRange:方法: 删除某个范围内的 可变数组中的元素	120
5.3.11	removeObjectsInArray:方法: 删除与 另一个数组相同的元素	120
5.3.12	replaceObjectAtIndex:方法: 替换可变数组中 某一位置的元素	121
5.3.13	replaceObjectsAtIndexes:方法: 替换可变数组中的 多个元素	122
5.3.14	replaceObjectsInRange:方法: 替换某一范围的 数组元素	123
5.3.15	insertObject:方法: 在某一位置插入数组元素	124
5.3.16	insertObjects:方法: 在某一位置或范围插入 另一数组元素	125
5.3.17	exchangeObjectAtIndex:方法: 交换两个元素	126

5.3.18	setArray:方法: 设置可变数组中内容	126
5.4	小结	127
第 6 章	字典对象 (NSDictionary)	128
6.1	创建及初始化	128
6.1.1	dictionary:方法: 创建字典	128
6.1.2	dictionaryWithContentsOfFile:方法: 将创建的字典内容设置为指定文件内容	128
6.1.3	dictionaryWithContentsOfURL:方法: 将创建的字典内容设置为指定 url 内容	129
6.1.4	dictionaryWithDictionary:方法: 用字典创建字典	130
6.1.5	dictionaryWithObject:方法: 创建一个键-值的字典	131
6.1.6	dictionaryWithObjects:方法: 创建具有多个键-值的字典	131
6.1.7	dictionaryWithObjectsAndKeys:方法: 创建具有多个键-值的字典	132
6.1.8	init:方法: 初始化字典	133
6.1.9	initWithContentsOfFile:方法: 将初始化的字典内容设置为指定文件内容	133
6.1.10	initWithContentsOfURL:方法: 将初始化的字典内容设置为指定 url 内容	134
6.1.11	initWithDictionary:方法: 用字典初始化字典	135
6.1.12	initWithObjects:方法: 初始化具有多个键-值的字典	135
6.1.13	initWithObjectsAndKeys:方法: 初始化具有多个键-值的字典	136
6.2	访问键-值	137
6.2.1	objectForKey:方法: 返回键的值	137
6.2.2	allKeys:方法: 返回所有的键	137
6.2.3	allValue:方法: 返回所有的值	138
6.2.4	allKeysForObject:方法: 返回值所对应的键	139
6.2.5	keyEnumerator:方法: 将字典中所有的键放到 NSEnumerator 对象中	139

6.2.6	objectEnumerator:方法: 将字典中所有的 值放到一个NSEnumerator 对象中	140
6.3	文件的处理	141
6.3.1	fileCreationDate:方法: 文件创建日期	141
6.3.2	fileModificationDate:方法: 文件修改的日期	141
6.3.3	fileSize:方法: 文件的大小	142
6.3.4	fileExtensionHidden:方法: 扩展名是否隐藏	142
6.3.5	fileType:方法: 文件的类型	143
6.3.6	fileGroupOwnerAccountID:方法: 文件所属组标识	144
6.3.7	fileGroupOwnerAccountName:方法: 文件所属组名	144
6.3.8	fileHFSCreatorCode:方法: 文件分层系统创建者编码	145
6.3.9	fileHFSTypeCode:方法: 文件分层系统类型编码	145
6.3.10	fileIsAppendOnly:方法: 文件是否只读	146
6.3.11	fileIsImmutable:方法: 文件是否可变	147
6.3.12	fileOwnerAccountID:方法: 文件所属人标识	147
6.3.13	fileOwnerAccountName:方法: 文件所属人	148
6.3.14	filePosixPermissions:方法: 权限	148
6.3.15	fileSystemFileNumber:方法: 文件系统的文件编号	149
6.3.16	fileSystemNumber:方法: 文件系统编号	149
6.3.17	writeToFile:方法: 字典内容写入文件中	150
6.3.18	writeToURL:方法: 字典内容写入 url 中	151
6.4	其他	152
6.4.1	count:方法: 字典键-值个数	152
6.4.2	isEqualToDictionary:方法: 判断字典是否相等	153
6.5	可变字典	153
6.5.1	dictionaryWithCapacity:方法: 创建固定长度的 可变字典	154
6.5.2	initWithCapacity:方法: 初始化固定长度的可变字典	154
6.5.3	setObject:方法: 设置键-值	154
6.5.4	setDictionary:方法: 用字典设置可变字典中的内容	155
6.5.5	removeAllObjects:方法: 删除所有的内容	156
6.5.6	removeObjectForKey:方法: 删除键所对应的值	156
6.5.7	removeObjectsForKeys:方法: 删除多个键所有的值	157

6.5.8	addEntriesFromDictionary:方法: 将字典中的 键-值添加到可变字典中	158
6.6	小结	159
第 7 章	集合 (NSSet)	160
7.1	创建以初始化	160
7.1.1	set:方法: 创建集合	160
7.1.2	setWithArray:方法: 用数组创建集合	160
7.1.3	setWithObject:方法: 创建具有一个元素的集合	161
7.1.4	setWithObjects:方法: 创建具有多个元素的集合	161
7.1.5	setWithSet:方法: 集合创建集合	162
7.1.6	init:方法: 初始化集合	163
7.1.7	initWithArray:方法: 用数组初始化集合	163
7.1.8	initWithObjects:方法: 初始化具有多个元素的集合	164
7.1.9	initWithSet:方法: 集合初始化集合	164
7.2	访问元素	165
7.2.1	objectEnumerator:方法: 将所有集合中的 元素放到 NSEnumerator 对象中	165
7.2.2	allObjects:方法: 返回集合中所有的元素	166
7.2.3	anyObject:方法: 返回任意一个元素	167
7.2.4	count:方法: 返回元素个数	167
7.3	判断比较	168
7.3.1	containsObject:方法: 判断集合中是否包含某个元素	168
7.3.2	member:方法: 判断集合中是否包含某个元素并返回	168
7.3.3	isSubsetOfSet:方法: 判断一个集合是否是 一个集合的子集	169
7.3.4	intersectsSet:方法: 判断交集	170
7.3.5	isEqualToSet:方法: 判断集合是否相等	170
7.4	可变集合	171
7.4.1	setWithCapacity:方法: 创建具有固定长度的可变集合	171
7.4.2	initWithCapacity:方法: 初始化具有 固定长度的可变集合	172
7.4.3	setSet:方法: 通过集合设置可变集合的内容	172