

零壹快学程序设计系列丛书

书网合一

小白实战手册

零壹快学

# 零基础iOS 从入门到精通

立体化教学模式 零基础快速入门 零壹快学 编著

## 真人教学，手把手教你学iOS

- 丰富示例，贴近场景——丰富的代码示例，编程场景化
- 视频教学，动手操作——扫码即可学习配套视频，通俗易懂
- 线上问答，及时解惑——在线论坛，解答学习中遇到的疑问
- 海量题库，随时学习——大量题目练习，加快理解



# 零基础iOS 从入门到精通

零壹快学 编著

SPM 南方出版传媒 广东人民出版社

·广州·

图书在版编目 ( CIP ) 数据

零基础iOS从入门到精通 / 零壹快学编著. —广州: 广东人民出版社, 2019.8  
ISBN 978-7-218-13761-2


I. ①零… II. ①零… 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 ( 2019 ) 第159410号

Ling Jichu iOS Cong Rumen Dao Jingtong

零基础 iOS 从入门到精通

零壹快学 编著

 版权所有 翻印必究

出版人: 肖风华

统筹策划: 李婧玮

责任编辑: 严耀峰

封面设计: 画画鸭工作室

内文设计: 奔流文化

责任技编: 周杰 吴彦斌

出版发行: 广东人民出版社

地 址: 广州市海珠区新港西路204号2号楼 ( 邮政编码: 510300 )

电 话: ( 020 ) 85716809 ( 总编室 )

传 真: ( 020 ) 85716872

网 址: <http://www.gdpph.com>

印 刷: 广东信源彩色印务有限公司

开 本: 787毫米×1092毫米 1/16

印 张: 27 字 数: 530千

版 次: 2019年8月第1版 2019年8月第1次印刷

定 价: 69.80元

如发现印装质量问题, 影响阅读, 请与出版社 ( 020-32449105 ) 联系调换。

售书热线: 020-32449123

零壹快学

《零基础 iOS 从入门到精通》编委会

主 编：景悦诚

副 主 编：蔡宇轩

编委会成员：宋宏宇 吕鉴倬 吕 严 杨 森

郑显军

# 前言

历经七十多年的发展，无论是对于国内数以十万计的学习者而言，还是在有着多年培训经验的编者看来，学习编程语言，仍存在不小的难度，甚至有不少学习者因编程语言的复杂多变、难度太大而选择了中途放弃。实际上，只要掌握了其变化规律，即使再晦涩难懂的计算机专业词汇也无法阻挡学习者的脚步。对于初学者来说，若有一本能看得懂，甚至可以用于自学的编程入门书是十分难得的。为初学者提供这样一本书，正是我们编写本套丛书的初衷。

零壹快学以“零基础，一起学”为主旨，针对零基础编程学习者的需求和学习特点，由专业团队量身打造了本套计算机编程入门教程。本套丛书的作者都从事编程教育和培训工作多年，拥有丰富的一线教学经验，对于学习者常遇到的问题十分熟悉，在编写过程中针对这些问题花费了大量的时间和精力来加以阐释，对书中的每个示例反复推敲，加以取舍，按照学习者的接受程度雕琢示例涉及的技术点，力求成就一套真正适合初学者的编程书籍。

本套丛书涵盖了Java、PHP、Python、JavaScript、HTML、CSS、Linux、iOS、C语言、C++、C#等计算机语言，同时借助大数据和云计算等技术，为广大编程学习者提供计算机各学科的视频课程、在线题库、测评系统、互动社区等学习资源。

## ◆ 课程全面，聚焦实战

本套丛书涵盖多门计算机语言，内容全面、示例丰富、图文并茂，通过通俗易懂的语言讲解相关计算机语言的特性，以点带面，突出开发技能的培养，既方便学习者了解基础知识点，也能帮助他们快速掌握开发技能，为编程开发设计积累实战经验。

## ◆ 专业团队，紧贴前沿

本套丛书作者由一线互联网公司高级工程师、知名高校教师和研究所技术人员等组成，线上线下同步进行专业讲解及点评分析，为学习者扫除学习障碍。与此同时，团队

在内容研发方向上紧跟当前技术领域热点，及时更新，直击痛点和难点。

◆ **全网覆盖，应用面广**

本套丛书已全网覆盖Web、APP和微信小程序等客户端，为广大学习者提供包括计算机编程、人工智能、大数据、云计算、区块链、计算机等级考试等在内的多门视频课程，配有相关测评系统和技术交流社区，互动即时性强，可实现在线教育随时随地轻松学。

随着移动互联网的兴起，iOS成为了全球最流行的移动端操作系统之一。作为iOS开发语言，Swift语言因其易用、高效、开源的特性，逐渐超越Objective-C成为主流。此外，各种创新的iOS应用不断涌现，从电商到社交，从游戏到教育，可谓琳琅满目。

本书基于Swift 4编写而成，通过Swift的语言基础入门，再到iOS应用开发的讲解，最终带领读者进入iOS的开发领域。对于零基础的读者而言，本书可以作为iOS应用开发的快速入门教材。

## • 本书内容

---

◆ **基础知识**：第1~2章，主要介绍了iOS的概况和iOS应用开发工具Xcode的使用方式。

◆ **Swift语言**：第3~8章，主要介绍Swift语言的知识，包括变量常量的定义、控制语句、Swift提供的数据结构、函数和闭包、自定义类型，以及一些高级特性如协议和扩展。

◆ **iOS开发**：第9~20章，主要介绍iOS应用的开发，包括iOS应用的框架、控件、事件处理、图像和多媒体、硬件访问、文件系统、数据存储等。

◆ **项目实战**：第21章，通过一个小应用来深入理解之前章节所学的知识点。

## • 本书特点

---

◆ **由浅入深，循序渐进**。本书通过先介绍Swift语言基础，再以playground作为实操工具逐渐过渡到iOS应用开发的学习，讲解过程详尽，通俗易懂。

◆ **示例丰富，贴近场景**。本书提供了丰富的代码示例，每个知识点均有对应示例代码进行演示，便于读者清晰理解。这些示例大部分来自于工作场景，有利于读者理解其中的使用逻辑，快速掌握。

◆ **视频教学，动手操作**。本书每一章都配有教学视频，直观展示了代码的运行效果，并配有通俗易懂的解释。

◆ **知识拓展，难度提升。**本书的大部分章节结尾设有“知识拓展”，在讲解基础知识的同时带给读者一些深入思考的知识，为有能力的读者提供强化学习，加深对iOS开发的理解。

◆ **线上问答，及时解惑。**本书为确保广大读者的学习能够顺利进行，提供了在线答疑服务，希望通过这种方式及时解决读者在学习iOS开发的过程中所遇到的困难和疑惑。

#### • 本书配套资源（可扫下方二维码获取）

- ◆ **大量的代码示例。**通过运行这些代码，读者可以进一步巩固所学的知识。
- ◆ **零壹快学官方视频教程。**力求让读者学以致用，知行并进，加强实战能力。
- ◆ **在线答疑。**为读者解惑，帮助读者解决学习中的困难，快速掌握要点难点。

#### • 本书适用对象

- ◆ 编程的初学者、爱好者与自学者
- ◆ 高等院校和培训学校的师生
- ◆ 职场新人
- ◆ 准备进入互联网行业的再就业人群
- ◆ “菜鸟”程序员
- ◆ 初、中级程序开发人员



零壹快学微信公众号

《零基础iOS从入门到精通》从初学者角度出发，详细讲述了iOS应用开发所需的基础知识和开发实战中的必备技能。全书内容通俗易懂，示例丰富，步骤清晰，图文并茂，可以使读者轻松掌握iOS应用开发的精髓，活学活用，是iOS开发实战中必备的Swift参考手册。

编者

2019年7月

## 目 录

## CONTENTS

<b>第 1 章 走进 iOS</b> ..... 1	2.4.2 使用 iOS 设备运行应用 ..... 19
1.1 iOS 简介 ..... 1	2.5 Playground 介绍 ..... 21
1.1.1 智能手机操作系统 ..... 1	2.6 小结 ..... 22
1.1.2 iOS 发展历史 ..... 2	<b>第 3 章 Swift 基础知识</b> ..... 23
1.1.3 iOS 应用程序的设计理念 ..... 2	3.1 Swift 语言介绍 ..... 23
1.2 iOS 系统的结构 ..... 3	3.2 变量 ..... 23
1.2.1 Core OS ..... 3	3.2.1 声明和命名规则 ..... 23
1.2.2 Core Services ..... 4	3.3 常量 ..... 24
1.2.3 Media ..... 6	3.3.1 声明和命名规则 ..... 24
1.2.4 Cocoa Touch ..... 7	3.4 Swift 基础数据类型 ..... 25
1.3 iOS 和 macOS 对比 ..... 7	3.4.1 数字类型 ..... 26
1.4 学好 iOS 开发的建议 ..... 8	3.4.2 布尔值 (Bool) ..... 28
1.5 小结 ..... 8	3.4.3 字符串 (String) ..... 29
<b>第 2 章 创建第一个 iOS 应用</b> ..... 9	3.4.4 可空类型 ..... 30
2.1 iOS 开发环境 ..... 9	3.5 注释 ..... 33
2.1.1 安装 Xcode ..... 9	3.6 小结 ..... 34
2.1.2 认识 Xcode ..... 9	3.7 知识拓展 ..... 35
2.2 创建 iOS 应用——Hello World ..... 10	<b>第 4 章 控制语句</b> ..... 36
2.2.1 创建一个 Xcode 项目 ..... 10	4.1 条件语句 ..... 36
2.3 认识 Xcode 的面板 ..... 11	4.1.1 if···else 语句 ..... 36
2.3.1 Navigators 面板 ..... 11	4.1.2 比较运算符 ..... 37
2.3.2 Jump Bar 跳转栏 ..... 16	4.1.3 逻辑运算符 ..... 37
2.4 运行 iOS 应用 ..... 17	4.1.4 三目运算符 ..... 39
2.4.1 使用 iOS 模拟器运行应用 ..... 17	

4.1.5	else if	39
4.2	switch 语句	40
4.2.1	什么是 switch	40
4.2.2	开始使用 switch	41
4.2.3	switch 的区间	42
4.2.4	switch 值绑定	43
4.2.5	where 子句	44
4.2.6	元组和模式匹配	45
4.2.7	switch 与 if...case	46
4.3	循环语句	48
4.3.1	for...in 循环	48
4.3.2	类型推断	51
4.3.3	while 循环	52
4.3.4	repeat...while 循环	52
4.3.5	控制转移语句	53
4.4	小结	54
4.5	知识拓展	54
<b>第 5 章</b>	<b>数据容器</b>	<b>56</b>
5.1	数组	56
5.1.1	数组创建	56
5.1.2	数组的读取和修改	57
5.1.3	数组的遍历	61
5.1.4	数组的相等判断	61
5.1.5	不可变数组	62
5.2	字典	63
5.2.1	字典创建	63
5.2.2	字典的读取和修改	64
5.2.3	字典的遍历	66
5.2.4	字典到数组的转换	67
5.2.5	不可变字典	67
5.3	集合	67
5.3.1	集合创建	68
5.3.2	集合的修改	68
5.3.3	集合的遍历	69

5.3.4	集合的运用	69
5.4	小结	71
5.5	知识拓展	71
<b>第 6 章</b>	<b>函数和闭包</b>	<b>73</b>
6.1	函数	73
6.1.1	函数的基本概念	73
6.1.2	函数参数	74
6.1.3	默认参数	75
6.1.4	in-out 参数	76
6.1.5	函数返回值	77
6.1.6	嵌套函数	78
6.1.7	作用域	79
6.1.8	函数类型	79
6.2	闭包	80
6.2.1	闭包的概念	80
6.2.2	闭包的语法	80
6.2.3	闭包表达式	81
6.2.4	闭包捕获上下文	82
6.2.5	闭包式引用	83
6.3	小结	84
6.4	知识拓展	85
<b>第 7 章</b>	<b>自定义类型</b>	<b>86</b>
7.1	枚举	86
7.1.1	基本枚举	86
7.1.2	枚举的原始值	90
7.1.3	方法	93
7.1.4	关联值	95
7.2	结构体	96
7.2.1	结构体的声明	96
7.2.2	实例方法	99
7.2.3	关键字 mutating	100
7.3	类	101
7.3.1	类的声明	101
7.3.2	类的继承	102

7.4 属性 .....	104	8.4 内存管理 .....	139
7.4.1 属性的设置 .....	104	8.4.1 内存分配 .....	139
7.4.2 属性的懒加载 .....	105	8.4.2 ARC .....	140
7.4.3 属性的读取 .....	106	8.4.3 循环引用 .....	141
7.4.4 属性的写入 .....	107	8.5 小结 .....	144
7.4.5 观察属性的变化 .....	109	8.6 知识拓展 .....	144
7.4.6 类型的属性 .....	110	<b>第9章 iOS 应用开发概览</b> .....	<b>148</b>
7.4.7 访问控制 .....	112	9.1 iOS 应用基础框架 .....	148
7.5 初始化 .....	113	9.1.1 iOS 应用的生命周期 .....	148
7.5.1 初始化语法 .....	114	9.1.2 main() 函数 .....	149
7.5.2 结构体的初始化 .....	115	9.1.3 UIApplicationDelegate .....	150
7.5.3 类的初始化 .....	117	9.1.4 Nib 文件 .....	152
7.5.4 可失败的初始化 .....	119	9.1.5 事件处理流程 .....	152
7.6 值类型与引用类型 .....	120	9.2 iOS 应用运行环境 .....	154
7.6.1 值语义 .....	120	9.2.1 快速启动 .....	154
7.6.2 引用语义 .....	121	9.2.2 安全处理 .....	154
7.6.3 相等与同一 .....	122	9.2.3 系统内存管理 .....	154
7.7 小结 .....	123	9.3 iOS 应用程序包 .....	155
7.8 知识拓展 .....	123	9.3.1 Info.plist .....	155
<b>第8章 Swift 高级特性</b> .....	<b>127</b>	9.3.2 应用程序图标 .....	157
8.1 协议 .....	127	9.4 iOS 应用程序间的交互 .....	158
8.1.1 协议的定义 .....	127	9.4.1 使用 URL 打开其他应用程序 .....	158
8.1.2 协议的使用 .....	127	9.4.2 从外部 URL 打开应用程序 .....	161
8.1.3 符合协议 .....	129	9.5 小结 .....	163
8.1.4 协议继承 .....	131	9.6 知识拓展 .....	163
8.1.5 协议组合 .....	132	<b>第10章 窗口和视图</b> .....	<b>166</b>
8.2 扩展 .....	133	10.1 窗口和视图简介 .....	166
8.2.1 扩展已有类型 .....	133	10.1.1 窗口 .....	168
8.2.2 扩展自定义类型 .....	135	10.1.2 视图 .....	170
8.3 错误处理 .....	136	10.2 UIView 基础 .....	172
8.3.1 错误的分类 .....	136	10.2.1 UIView 的 Frame 布局 .....	172
8.3.2 抛出错误 .....	136	10.2.2 UIView 绘制 .....	175
8.3.3 捕获错误 .....	138	10.2.3 UIView 的层次结构 .....	176
8.3.4 错误处理的原则 .....	139	10.3 自动布局 Auto Layout .....	176

10.3.1 布局属性 .....	177	12.3.3 UITextFieldDelegate .....	220
10.3.2 约束 .....	178	<b>12.4 UISwitch .....</b>	<b>222</b>
<b>10.4 小结 .....</b>	<b>181</b>	12.4.1 在视图中添加 UISwitch .....	222
<b>10.5 知识拓展 .....</b>	<b>181</b>	12.4.2 UISwitch 的属性 .....	224
<b>第 11 章 视图控制器 .....</b>	<b>183</b>	<b>12.5 UISegmentedControl .....</b>	<b>225</b>
<b>11.1 视图控制器基础 .....</b>	<b>183</b>	12.5.1 在视图中添加 UISegmentedControl .....	225
11.1.1 视图控制器的功能 .....	183	12.5.2 UISegmentedControl 的属性和方法 .....	228
11.1.2 UIViewController 的子类 .....	185	<b>12.6 UISlider .....</b>	<b>229</b>
<b>11.2 UIViewController 的生命周期 .....</b>	<b>187</b>	12.6.1 在视图中添加 UISlider .....	230
11.2.1 初始化 .....	187	12.6.2 UISlider 的属性 .....	232
11.2.2 加载视图 .....	187	<b>12.7 UIImageView .....</b>	<b>234</b>
11.2.3 视图展示 .....	187	12.7.1 在视图中添加 UIImageView .....	234
11.2.4 视图消失 .....	189	12.7.2 UIImageView 的属性 .....	235
<b>11.3 UIViewController 的使用 .....</b>	<b>189</b>	<b>12.8 UIProgressView .....</b>	<b>237</b>
11.3.1 UIViewController 概述 .....	189	12.8.1 在视图中添加 UIProgressView .....	237
11.3.2 自定义 UIViewController .....	189	12.8.2 UIProgressView 的属性 .....	238
<b>11.4 UINavigationController 的使用 .....</b>	<b>194</b>	<b>12.9 UIActivityIndicatorView .....</b>	<b>239</b>
11.4.1 UINavigationController 概述 .....	194	12.9.1 在视图中添加 UIActivityIndicatorView .....	239
11.4.2 使用 UINavigationController .....	194	View .....	239
<b>11.5 UITabBarController 的使用 .....</b>	<b>200</b>	12.9.2 UIActivityIndicatorView 的属性 .....	241
11.5.1 UITabBarController 概述 .....	200	<b>12.10 UIDatePicker .....</b>	<b>241</b>
11.5.2 使用 UITabBarController .....	200	12.10.1 在视图中添加 UIDatePicker .....	241
<b>11.6 小结 .....</b>	<b>205</b>	12.10.2 UIDatePicker 的属性 .....	244
<b>11.7 知识拓展 .....</b>	<b>205</b>	<b>12.11 UIPickerView .....</b>	<b>246</b>
<b>第 12 章 简单控件 .....</b>	<b>209</b>	12.11.1 添加 UIPickerView .....	246
<b>12.1 UIKit 简介 .....</b>	<b>209</b>	12.11.2 UIPickerView 的属性 .....	247
12.1.1 控件的分类 .....	209	12.11.3 UIPickerViewDataSource .....	248
12.1.2 UIControl .....	210	12.11.4 UIPickerViewDelegate .....	248
<b>12.2 UIButton .....</b>	<b>212</b>	<b>12.12 UIAlertController .....</b>	<b>249</b>
12.2.1 在视图中添加 UIButton .....	212	12.12.1 展示 UIAlertController 警示框 .....	249
12.2.2 UIButton 的属性 .....	216	12.12.2 展示 UIAlertController 选择框 .....	254
<b>12.3 UITextField .....</b>	<b>217</b>	<b>12.13 UIStepper .....</b>	<b>257</b>
12.3.1 在视图中添加 UITextField .....	217	12.13.1 添加 UIStepper .....	257
12.3.2 UITextField 的属性 .....	219		

12.13.2 UIStepper 的属性 .....	259	14.3 手势 .....	304
12.14 UITextView .....	260	14.4 编辑事件 .....	305
12.14.1 添加 UITextView .....	261	14.4.1 粘贴板 .....	306
12.14.2 UITextView 的属性 .....	262	14.4.2 数据表示 .....	307
12.14.3 UITextViewDelegate .....	264	14.4.3 选项菜单 .....	308
12.15 小结 .....	265	14.4.4 剪切和复制 .....	312
12.16 知识拓展 .....	265	14.4.5 粘贴 .....	312
<b>第 13 章 复杂控件 .....</b>	<b>268</b>	14.5 小结 .....	313
13.1 UITableView .....	268	14.6 知识拓展 .....	313
13.1.1 添加 UITableView .....	268	<b>第 15 章 绘图 .....</b>	<b>314</b>
13.1.2 UITableViewDataSource .....	269	15.1 Quartz 2D .....	314
13.1.3 UITableViewDelegate .....	271	15.2 Quartz 2D 绘图 .....	315
13.1.4 UITableViewCell .....	273	15.2.1 坐标 .....	315
13.1.5 创建一个数据类 .....	276	15.2.2 颜色 .....	317
13.2 UICollectionView .....	278	15.2.3 图形 .....	318
13.2.1 添加 UICollectionView .....	278	15.3 贝塞尔曲线 .....	319
13.2.2 UICollectionViewDataSource .....	279	15.3.1 绘制曲线 .....	320
13.2.3 UICollectionViewDelegate .....	281	15.3.2 曲线内部检测 .....	322
13.2.4 UICollectionViewFlowLayout .....	283	15.4 小结 .....	324
13.2.5 UICollectionViewCell .....	284	15.5 知识拓展 .....	324
13.3 WKWebView .....	288	<b>第 16 章 动画 .....</b>	<b>326</b>
13.3.1 添加 WKWebView .....	288	16.1 Core Animation .....	326
13.3.2 WKNavigationDelegate .....	290	16.1.1 CALayer .....	326
13.3.3 WKUIDelegate .....	295	16.1.2 CAAAnimation .....	326
13.4 小结 .....	296	16.1.3 CATransaction .....	329
13.5 知识拓展 .....	296	16.1.4 动画实战 .....	330
<b>第 14 章 事件处理 .....</b>	<b>299</b>	16.2 小结 .....	331
14.1 触摸 .....	299	<b>第 17 章 多媒体开发 .....</b>	<b>332</b>
14.1.1 触摸机制 .....	299	17.1 初识 AVFoundation .....	332
14.1.2 触摸与事件 .....	299	17.2 音频 .....	333
14.1.3 事件传递 .....	300	17.2.1 音频支持 .....	333
14.1.4 事件响应过程 .....	301	17.2.2 播放音频 .....	334
14.1.5 事件处理方法 .....	302	17.2.3 录制音频 .....	338
14.2 动作 .....	303		

17.3 相册以及相机使用 .....	341	20.1.1 SQLite 中的数据类型 .....	367
17.3.1 UIImagePickerController.....	341	20.1.2 SQLite 常用函数及返回码 ...	368
17.3.2 UIImagePickerControllerDelegate	344	20.1.3 打开数据库 .....	373
17.4 小结 .....	345	20.1.4 创建表 .....	374
<b>第 18 章 传感器使用 .....</b>	<b>346</b>	20.1.5 操作语句 .....	375
18.1 Core Motion .....	346	20.1.6 数据查询 .....	377
18.1.1 加速度计 .....	346	<b>20.2 XML .....</b>	<b>379</b>
18.1.2 陀螺仪 .....	348	20.2.1 XML 简介 .....	379
18.1.3 磁力计 .....	349	20.2.2 XML 语法 .....	380
18.2 Core Location .....	350	20.2.3 XML 元素 .....	381
18.2.1 获取当前位置 .....	350	20.2.4 XML 属性 .....	382
18.2.2 获取城市信息 .....	352	<b>20.3 JSON .....</b>	<b>383</b>
18.3 小结 .....	353	20.3.1 JSON 简介 .....	383
<b>第 19 章 iOS 文件系统 .....</b>	<b>354</b>	20.3.2 JSON 语法 .....	383
19.1 iOS 文件操作 .....	354	20.3.3 JSON 实战 .....	384
19.1.1 iOS 常用目录 .....	355	<b>20.4 小结 .....</b>	<b>386</b>
19.1.2 获取常用目录的路径 .....	355	<b>20.5 知识拓展 .....</b>	<b>386</b>
19.1.3 iOS 应用更新机制 .....	358	<b>第 21 章 项目实战 .....</b>	<b>388</b>
19.1.4 iOS 文件备份和恢复 .....	358	21.1 发布流程 .....	388
19.2 iOS 数据读写操作 .....	358	21.2 倒计时提醒应用 .....	390
19.2.1 读写文件数据 .....	358	21.2.1 应用功能的确定 .....	390
19.2.2 读写属性列表数据 .....	359	21.2.2 创建项目 .....	390
19.2.3 归档器和解档器读写数据 ...	363	21.2.3 创建自定义的 Cell .....	394
19.2.4 保存状态 .....	365	21.2.4 搭建添加事件的界面 .....	398
19.3 小结 .....	366	21.2.5 事件的数据结构 .....	403
19.4 知识拓展 .....	366	21.2.6 事件的数据库读写 .....	405
<b>第 20 章 数据 .....</b>	<b>367</b>	21.2.7 开发添加事件的功能 .....	408
20.1 SQLite 数据库 .....	367	21.2.8 开发事件展示的界面 .....	416
		21.3 小结 .....	419

## » 第 1 章

# 走进iOS <<

## 1.1 iOS简介

iOS（最初称为iPhone Runs OS X）是由苹果公司开发并在其移动设备上使用的操作系统，属于类Unix系统，发布于2007年1月9日。iOS最初是搭载于iPhone推出的，之后扩展到了iPad、iPod等苹果产品上，目前是仅次于Android的第二大智能手机操作系统。本书将带领开发者走进iOS的世界。

### 1.1.1 智能手机操作系统

在了解iOS之前，先介绍下目前市场上较为流行的智能手机操作系统都有哪些。统计报告显示，谷歌的Android、苹果的iOS、微软的Windows Phone以及黑莓的Blackberry OS是目前占据市场份额前四的智能手机操作系统。下面我们将依次介绍Android、Windows Phone以及Blackberry OS三种智能手机操作系统。

首先介绍谷歌的Android。Android是一种以Linux为基础的开放源代码的操作系统，主要用于便携设备。Android操作系统最初由Andy Rubin开发，主要支持手机等移动设备。2011年第一季度，Android在全球的市场份额首次超过塞班系统，跃居全球第一。截止到2018年5月Android最新的版本为9.0。不过由于很多厂商都有自己深度定制的Android系统，因此，市面上的Android手机使用的版本普遍还是在4.X至8.X之内。

微软的Windows Phone是微软发布的一款手机操作系统，将微软旗下的Xbox Live游戏、Zune音乐与独特的视频体验整合至手机中。2010年10月21日，微软公司正式发布了智能手机操作系统Windows Phone。2011年2月，诺基亚与微软达成全球战略同盟并深度合作共同研发该系统。2015年1月21日，微软召开Windows 10发布会，会上提出Windows 10将是一个跨平台的系统，无论是手机、平板、笔记本、二合一设备还是个人电脑，Windows 10将全部适用，这也就意味着2010年发布的Windows Phone品牌将正式终结。而Windows Phone系统也在经历了Windows Phone 7、Windows Phone 7.1/7.5/7.8、Windows Phone 8和Windows Phone 8.1后正式谢幕。不过Windows Phone 8和Windows Phone 8.1将继续提供支持直到各自3年整的生命周期结束。2018年2月20日开始，微软关

闭Windows Phone 7.5和Windows Phone 8.0的通知推送服务，意味着Windows Phone的正式谢幕。

最后要介绍的是黑莓的Blackberry OS。“Blackberry OS”为黑莓手机系统的英文名称，它是由RIM研发推出的与黑莓手机配套使用的智能操作系统。在此系统的基础上，黑莓的手机更是独树一帜地在智能手机市场中拼搏，并在中国吸引了大量粉丝。2012年7月，黑莓占据了全球智能手机操作系统7%的市场份额，成为全球第四大智能操作系统。2013年1月30日起，RIM与Blackberry合并。最后，黑莓被微软的Windows Phone智能操作系统超越。由于种种原因，黑莓最终放弃了自己的操作系统，最新的黑莓手机已经使用了Android。

智能手机操作系统的市场竞争非常激烈，目前只剩下两大巨头：有良好生态的苹果iOS和有广大社区及厂商支持的谷歌Android。所以学好iOS开发，就等于掌握了移动端开发的“半壁江山”。

### 1.1.2 iOS发展历史

了解过其他的智能手机操作系统之后，现在回归到苹果的iOS上。简单罗列一下iOS的里程碑：

1. 2007年1月9日，苹果公布了iPhone OS，最初名为“iPhone Runs OS X”。
2. 2008年3月更名为“iPhone OS”，同年7月推出App Store。
3. 2010年改名为“iOS”，沿用至今。
4. iOS 5~iOS 9：苹果在图标设计和功能引入上做了很多的革新，例如扁平化的设计，以及诸如地图、全景相机、passbook和Siri等功能的加入。
5. iOS 10：苹果重塑了iOS、macOS、watchOS、tvOS四大操作系统。
6. iOS 11：苹果主推了ARKit、CoreML两个核心框架，用于增强虚拟现实与机器学习。

在十几年的发展中，苹果不断地创新，不断地优化产品，以此引领现有的智能手机市场。苹果的成功，很大程度取决于用户的体验和设计的理念。了解苹果iOS的设计理念，对开发一个完美的iOS应用有着巨大的帮助。

### 1.1.3 iOS应用程序的设计理念

在设计开发iOS应用程序时，为了能最大程度地提高应用程序的影响力和覆盖用户，开发者需要记住以下准则：

#### 1. 审美的完整

审美的完整，是指一款iOS应用程序需要将外观和行为与实际提供的功能进行完美结合。例如，一个帮助用户执行严肃任务的应用可以使用简单且不显眼的图形，或通过使用标准控件和可预测的行为来保持他们的专注；再如，沉浸式的应用（如游戏）可以通过极具吸引力的视觉表现来保证用户获得体验上的兴奋与刺激，同时也能鼓励用户去发现乐趣。

#### 2. 一致性

一致性，是指一款iOS应用程序需要通过使用系统提供的交互元素、明确意义的图标、标准的文本样式和统一的术语来实现熟悉的标准和范例，以人们期望的方式进行交互和功能展现。

### 3. 简单直接的操作

iOS设备屏幕上的内容可以直接操作，这样可以帮助用户便捷地使用与理解。用户在旋转设备或使用手势时，内容的变化能够快速反映在屏幕上。通过直接操作，用户可以快速且明显地看到他们预期的效果。

### 4. 实时的反馈

在应用程序中，确认行为并显示结果可以使用户了解程序的使用情况。内置的iOS应用程序可以响应每个用户的操作，并提供能够被感知的反馈，如轻触时会突出显示交互式元素、进度指示器会传达长时间运行的状态、动画和声音有助于阐明操作的结果。

### 5. 与真实世界相对应

当应用程序内的虚拟对象和行为能够对应到真实世界或虚拟世界的已有体验时，用户可以更快地学习如何使用。这种对应在iOS系统中运行得很好的原因是用户能与屏幕进行实际的接触。用户通过拖动和滑动视图的操作来展示更多的内容，以及使用切换开关、移动滑块和滚动控件来改变应用的行为，就像是真实地翻阅书籍和杂志一样。

### 6. 用户控制

在整个iOS系统中，是由用户而非应用程序来控制的。应用程序可以提示用户如何操作或警告可能出现的严重后果，但应用绝不能直接替用户做决策。一个好的应用应该能够在何时允许用户操作和何时能避免不可预测结果之间找到平衡。应用要给用户可以提供可以预测的交互行为及熟悉的控件，使用户能够按照预期进行使用。

以上六个准则是苹果官方指南中提出的，也是苹果想给用户带来的体验。如果想做出一款成功的iOS应用，这些设计理念需要在开发iOS应用时融会贯通。

## 1.2 iOS系统的结构

iOS应用设计的六大准则可以指引开发者做出完美的iOS应用，了解iOS开发者系统的结构则可以帮助开发者知道在完成某个目标时，应该去使用什么系统接口或者系统服务。

iOS系统的结构包含了四个主要的层次：Core OS、Core Services、Media和Cocoa Touch。从应用层次上看，iOS是帮助应用程序与底层硬件之间打交道的“中间人”。用户在打开应用时不能直接访问底层硬件，而是需要和系统接口进行交互，由系统接口与硬件驱动进行实际操作，这样的好处是防止发生应用程序直接改变底层硬件的行为。开发者在编写iOS应用时，应尽可能避免使用底层框架而使用高层框架，这样可以使代码简洁，也避免了不必要的异常情况。

### 1.2.1 Core OS

Core OS，又称核心OS，其实质是苹果基于FreeBSD和Mach所改写的Darwin，它是一个符合

POSIX标准的Unix核心。这一层提供了整个iOS的基础功能，包括硬件驱动、内存管理、程序管理、线程管理、文件系统、网络传输以及标准输入输出等。此外，Core OS层的驱动也提供了硬件和系统框架之间的接口，通过下面的框架（Framework），程序可以访问大部分和底层相关的硬件或者服务。

### 1. Accelerate

Accelerate框架是一个进行大规模的数学计算、图像计算以及针对高性能进行优化的框架，它借助了自身矢量运算的能力，在CPU上提供高性能和高能效的计算。开发者使用这个框架比开发者自己实现这些接口有优势，因为Accelerate框架已经根据所有iOS设备硬件配置进行过优化，开发者只需写一次代码就能确保它在所有设备上有效。

### 2. Core Bluetooth

Core Bluetooth为iOS应用提供了与蓝牙4.0低功耗的设备进行通信的能力，为此，它提供了所需要的类。须注意不要继承并且覆写这些类的方法，这样做会导致不必要的崩溃。这个框架的接口允许开发者扫描附近的蓝牙设备并与找到的设备进行连接、断开连接或进行一些信息的传递。

### 3. ExternalAccessory

ExternalAccessory为iOS应用提供了与外部设备通信的能力，这些设备可以通过蓝牙或者Apple Lightning连接器连接到iOS设备上。ExternalAccessory提供了获取每个可用配件的信息和初始通信会话的方式，通信成功后，开发者可以自由地操作配件，直接使用任何它支持的命令。

### 4. LocalAuthentication

LocalAuthentication可以让用户通过生物认证（如Touch ID、Face ID）或者密码进行身份验证。用户可以使用这些认证方式轻松地访问设备或者应用。为了最大程度地提高安全性，开发者无法直接获取这些验证数据，只能在某个验证之后得到验证的结果。

### 5. Network

Network提供了基于传输协议和安全协议的网络连接，用于发送和接收数据。当需要直接访问TLS、TCP、UDP等协议的时候可以使用Network，其中提供的NSURLSession可以方便地加载基于HTTP协议的URL资源。

### 6. Security

Security可以保护应用内的数据，并控制外部对应用的访问。Security提供了一些主流加密的实现方案，大部分场景不需要开发者自己去实现这些协议。此外，它还可以方便地在多台设备上共享认证信息，方便用户访问应用或内容。

## 1.2.2 Core Services

Core Services层为iOS应用程序提供基础的系统服务。其中最重要的是Core Foundation和Foundation框架，它们定义了所有应用都需要的基础类型。这一层还包含了诸如位置服务、iCloud、社交媒体