

移动开发丛书

赠送iOS UI开发视频教程和源代码
共36堂课，播放时长超过13小时

↓源代码下载↓

iOS

开发实战

从零基础到上架App Store

▪ 第3版 ▪

张益琿 编著

使用iOS 11+Xcode 9+Swift 4开发

清华大学出版社



移动开发丛书

iOS

开发实战

从零基础到上架App Store

▪ 第3版 ▪

张益琿 编著



清华大学出版社
北京

内 容 简 介

本书结合编者多年 iOS 开发经验,使用 iOS 11+Xcode 9+Swift 4 开发环境,系统地介绍了 iOS 应用程序从开发到上架的全过程。从开发环境搭建、界面开发、传感器技术、布局与动画技术、网络和数据技术到打包与上传发布流程等,包含了 iOS 软件开发核心技术的方方面面。值得注意的是,本书针对每章的技术要点配备了来自工作实践的项目案例,读者可以边学边练,在编写代码中学习编程。

本书深入浅出,注重实战,案例丰富,非常适合快速上手 iOS 开发的新人,也很适合有一定编程基础但缺少开发 iOS 应用经验的开发人员,还可以作为大中专院校及培训机构的教学用书。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

iOS 开发实战:从零基础到上架 App Store / 张益珉编著. —3 版. —北京:清华大学出版社, 2018
(移动开发丛书)

ISBN 978-7-302-51195-3

I. ①i… II. ①张… 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 (2018) 第 210616 号

责任编辑:王金柱

封面设计:王 翔

责任校对:闫秀华

责任印制:李红英

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质 量 反 馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

印 装 者:三河市国英印务有限公司

经 销:全国新华书店

开 本:190mm×260mm

印 张:22

字 数:563 千字

版 次:2016 年 8 月第 1 版

2018 年 10 月第 3 版

印 次:2018 年 10 月第 1 次印刷

定 价:79.00 元

产品编号:077351-01

前言

编写本书的目的

截止到本书出版，图书市场上已经有很多关于 iOS 开发教学的书籍，它们各有优势，比如有些书籍在某个技术面讲解的非常细致深入，有些书籍专注于 iOS 应用的性能优化，有些书籍则注重于开发语言的语法讲解等。然而如果你有一定的编程语言基础，并且没有任何完整的开发经验，想要学习 iOS 开发，本书依然是非常好的选择，本书为初学者定制，对基础知识点的讲解细致入微，并且配有非常详细的示例代码。本书的前两版，也得到了许多读者的认可，本版相比前两版扩充了一些新的技术点，并且采用最新的 iOS 11+Xcode 9+Swift 4 作为开发环境，以使读者了解 iOS 技术进展，掌握最新的开发技术。

开发一款完整的 iOS 软件是一个复杂的过程，开发者除了需要有编程语言的基础外，还需要对程序设计有宏观的把控。本书在编写时，定位的目标就是要帮助并无太多基础的读者快速上手 iOS 应用开发。从开发环境准备到程序界面开发，从网络与数据逻辑开发到动画与传感器技术，本书都有专门的章节进行介绍。如果说学习理论是一个枯燥的过程，那么在编程中的动手实践就是对理论学习成果的最好检验。本书中基本每一个模块都配备了实战项目，读者可以通过实战项目的练习，体会独立开发 App 应用程序的成就感。

本书主要内容

本书在结构上分 11 章，下面介绍各章的主要内容。

第 1 章是为学习应用开发做准备，介绍开发环境的搭建与开发工具的使用，这一章虽然为准备章节，但对初学者来说至关重要。

第 2 章介绍 iOS 开发中的一些基础 UI 控件，移动端应用一个很重要的特点就是要有绚丽的界面，应用程序的界面决定了用户使用这款应用程序的体验与心情，这一章向读者独立地介绍每个基础控件的用法，并通过一些综合实战来给读者提供使用这些控件的机会。

第 3 章在第 2 章的基础上，向读者介绍 iOS 开发中经常使用的更多高级控件的用法，同样也为读者提供了实战机会。

第 4 章主要讲解了 iOS 应用开发中的网络编程技术，由于网络编程的演示需要有网络数据支持，很多有关网络教学的文档书籍都只是讲授理论，却没有办法使读者切身地进行测试与练习。在编写本章时，特别注意了这个问题，除了讲授网络编程在 iOS 应用开发中的相关知识外，还教读者如何使用网上免费的 API 服务真正做出一款网络应用。

第 5 章主要讲解 iOS 应用程序开发中的音频与视频技术，这类技术在开发音频软件和视频软件中意义重大。

第 6 章将作为动画专题，向读者介绍 iOS 应用开发中的动画技术，章节设计由简到难，并且

都配有代码演示。

第 7 章将作为传感器专题，向读者介绍 iOS 开发中可以调用的设备传感器的相关知识。

第 8 章是界面布局专题，编写中参阅了很多 iOS 应用开发教材，其中都没有过多提到界面布局的相关知识，笔者认为这是一个十分大的弊端，界面布局技术是衡量一个开发者是否合格的重要指标，笔者相信读者学习 iOS 开发技术绝对不只是想简简单单地做出一个 Demo 自己玩，做出“产品”才是读者的真正目标，而一款成熟的产品一定是具有兼容性的，一定是优雅的。因此，本书特别将 iOS 界面布局技术作为一个单独的章节来向读者介绍。

第 9 章是数据持久化专题，本章将介绍有关 iOS 应用开发中的文件操作、数据库操作的相关知识。

通过前 9 章的学习，你已经具备独立开发一款 iOS 应用的基础能力，但是仅仅做出产品还不够，如何让自己的产品在市场发布，使用户可以下载使用也是开发者不得不去了解、学习的内容，第 10 章将完整地向读者介绍提交自己的应用到 App Store 的整个过程。

第 11 章是扩展章节，此章也是读者开发能力提升的一个章节，本章将介绍一些独立于前面章节，但在实际开发中也举足轻重的编程技术。

视频教学和源代码下载

读者可以从如下网址（注意区分数字与字母大小写）下载源代码和视频教学：

<https://pan.baidu.com/s/19Z5WSYAqKXLpif-t9kkNMg>

https://pan.baidu.com/s/1TaYi8cyW6zh_htMRjVOayA

也可以扫描下面的二维码获取网址。如果下载有问题，请联系电子邮箱 booksaga@126.com，邮件主题为“iOS 开发实战：从零基础到上架 App Store”。



如果你想获取更多关于 iOS 开发的视频教学，可以访问笔者的网络学院：<http://edu.csdn.net/lecturer/1283>。也可以访问笔者的技术博客：<https://my.oschina.net/u/2340880>。博客中有近 300 篇关于 iOS 开发的技术文章，这可以成为你的知识存储库。

对于本书前两版中出现的差错，衷心地向读者表示歉意。本版中对读者提出的问题都一一进行了修正，在终稿前，笔者花费了很多时间反复校稿，希望本版可以完美地出现在读者面前，然而世上完美之事少之又少，虽已尽力，仍不敢保证此书中再无差错。此书一经出版已成定局，如果你在阅读本书时发现任何差错或有任何疑问，都可以直接联系笔者，QQ: 316045346。

最后，本书得以顺利完成和出版，要感谢清华大学出版社的王金柱编辑，感谢他在笔者写作过程中的指导与鼓励，可以将编程经验顺利地分享给读者，王金柱编辑是不可替代的桥梁。

张益琿（网名：琿少）

2018.7.22

目 录

第 1 章 开发准备	1
1.1 iOS 11 新特性简述	1
1.1.1 新增拖放交互编程接口	2
1.1.2 其他新增功能	4
1.2 熟悉 iOS 开发环境	4
1.2.1 安装 Xcode 开发工具	4
1.2.2 了解 Xcode 开发工具主界面	6
1.2.3 Xcode 开发工具的使用技巧及常用快捷键	7
1.3 创建第一个 iOS 项目	9
1.4 使用 Git 进行项目版本管理	12
1.4.1 Git 与 Github 简介	12
1.4.2 注册 GitHub 会员	12
1.4.3 使用 Xcode 创建 Git 仓库	13
1.4.4 用 Xcode 建立本地 Git 仓库与 GitHub 代码托管平台的关联	15
第 2 章 基础 UI 组件	18
2.1 iOS 系统 UI 框架的介绍	18
2.1.1 MVC 设计模式	19
2.1.2 代理设计模式	19
2.2 视图控制器——UIViewController	20
2.2.1 UIViewController 的生命周期	20
2.2.2 UIViewController 的视图层级结构	24
2.3 文本控件——UILabel	24
2.3.1 使用 UILabel 在屏幕上创建一个标签控件	24
2.3.2 自定义标签控件的相关属性	25
2.3.3 多行显示的 UILabel 控件与换行模式	26
2.4 按钮控件——UIButton	27
2.4.1 创建一个按钮改变屏幕颜色	27
2.4.2 更加多彩的 UIButton 控件	29
2.5 文本输入框控件——UITextField	31
2.5.1 在屏幕上创建一个输入框	31

2.5.2	UITextField 的常用属性介绍	33
2.5.3	UITextField 的代理方法	33
2.5.4	实现一个监听输入信息的用户名输入框	34
2.6	开关控件——UISwitch	35
2.6.1	创建一个开关控件	35
2.6.2	为 UISwitch 控件添加触发方法	36
2.7	分页控制器——UIPageControl	37
2.8	分段控制器——UISegmentedControl	37
2.8.1	UISegmentedControl 基本属性的应用	38
2.8.2	对 UISegmentedControl 中的按钮进行增、删、改操作	38
2.8.3	UISegmentedControl 中按钮宽度的自适应	39
2.9	滑块控件——UISlider	40
2.9.1	UISlider 的创建与常规设置	40
2.9.2	对 UISlider 添加图片修饰	41
2.10	活动指示器控件——UIActivityIndicatorView	41
2.11	进度条控件——UIProgressView	42
2.12	步进控制器——UIStepper	43
2.12.1	步进控制器的基本属性使用	43
2.12.2	自定义 UIStepper 按钮图片	44
2.13	选择器控件——UIPickerView	44
2.13.1	创建一个 UIPickerView 控件	44
2.13.2	UIPickerView 选中数据时的回调代理	46
2.14	通过 CALayer 对视图进行修饰	46
2.14.1	创建圆角的控件	47
2.14.2	创建带边框的控件	47
2.14.3	为控件添加阴影效果	48
2.15	警告控制器——UIAlertController	48
2.15.1	UIAlertController 的警告框	48
2.15.2	UIAlertController 之活动列表	50
2.16	基础 UI 控件扩展篇	51
2.16.1	搜索栏控件——UISearchBar	51
2.16.2	日期时间选择器——UIDatePicker	52
2.16.3	警告视图——UIAlertView	54
2.16.4	活动列表——UIActionSheet	54
2.17	实战：登录注册界面的搭建	55
第3章	高级 UI 控件	59
3.1	导航控制器——UINavigationController	60
3.1.1	导航控制器的工作原理	60

3.1.2	使用导航控制器进行多界面搭建.....	61
3.1.3	导航栏 UINavigationController.....	63
3.1.4	导航按钮 UIBarButtonItem	65
3.1.5	导航控制器的工具栏	67
3.1.6	iOS 8 系统之后导航控制器的一些有趣功能	67
3.2	标签控制器——UITabBarController	68
3.2.1	标签控制器的工作原理	68
3.2.2	标签控制器的基础用法解析	68
3.2.3	关于 UITabBarItem 的使用	70
3.3	滚动视图——UIScrollView	71
3.3.1	使用 UIScrollView 展示视图内容	71
3.3.2	UIScrollView 的代理方法	72
3.4	网络视图——UIWebView	74
3.4.1	App 网络传输安全策略.....	74
3.4.2	通过网络请求加载 UIWebView	75
3.4.3	通过 HTML 字符串加载 UIWebView.....	75
3.4.4	通过 Data 数据加载 UIWebView.....	76
3.4.5	UIWebView 中常用方法解析	76
3.4.6	UIWebView 的代理方法	77
3.5	表格视图——UITableView	78
3.5.1	UITableView 的创建与复用机制.....	78
3.5.2	创建一个表格视图 UITableView.....	79
3.5.3	关于表格数据的载体 UITableViewCell.....	81
3.5.4	设置 UITableView 的行高和头尾视图.....	82
3.5.5	UITableView 的用户交互行为	83
3.5.6	为 UITableView 添加索引栏	85
3.6	复杂布局视图——UICollectionView	85
3.6.1	UICollectionView 控件的优势与布局方式	86
3.6.2	使用 UICollectionView 进行九宫格式的布局	86
3.6.3	创建更加灵活的流式布局	88
3.6.4	自定义 UICollectionViewFlowLayout 进行参差瀑布流布局.....	88
3.6.5	使用 UICollectionView 进行圆环布局	91
3.7	实战：开发一款手机网页浏览器.....	93
3.7.1	网页浏览器工程的搭建	93
3.7.2	核心网页视图的设计	94
3.7.3	历史记录界面的设计	101
3.7.4	收藏界面的设计	102
3.7.5	启动页面、图标及应用名称的相关优化.....	103
3.8	扩展内容.....	106

3.8.1	应用内评价组件	106
3.8.2	进行系统拨号与短信界面的调用	106
3.8.3	拖拽交互体验	107
第 4 章	网络编程技术	111
4.1	使用 NSURLConnection 请求网络数据	111
4.1.1	申请一个免费的 API 服务	112
4.1.2	使用 NSURLConnection 进行 API 服务数据的获取	114
4.1.3	使用 NSURLConnection 进行异步网络请求	115
4.1.4	使用 NSURLConnection 类通过代理回调的方式异步请求	116
4.2	设计封装一个更加易用的网络请求类	117
4.2.1	设计自定义的网络请求连接类	117
4.2.2	设计自定义的网络请求管理类	118
4.3	JSON 类型数据的解析与数据模型的设计	120
4.3.1	JSON 数据简介	120
4.3.2	在 iOS 中解析 JSON 数据	121
4.3.3	数据模型 Model 类的设计	122
4.4	使用 CocoaPods 进行第三方库的管理	124
4.4.1	在 MAC 上安装 CocoaPods 工具	125
4.4.2	用 CocoaPods 搭建一个使用第三方网络请求框架 Alamofire 的工程	126
4.5	使用 Alamofire 进行网络请求	128
4.5.1	详解 HTTP/HTTPS 协议	128
4.5.2	使用 Alamofire 进行网络请求	129
4.6	实战：开发“笑一笑”应用程序	130
4.6.1	工程项目框架的搭建	130
4.6.2	“笑一笑”界面数据载体 cell 的设计	132
4.6.3	“笑一笑”界面的搭建	134
4.6.4	实现下拉刷新与加载更多功能	137
4.6.5	“趣图吧”界面数据载体 cell 的设计	138
4.6.6	“趣图吧”界面的设计	141
第 5 章	音频、视频开发技术	145
5.1	iOS 音频开发基础——AVAudioPlayer 类的使用	145
5.1.1	使用 AVAudioPlayer 进行 MP3 音频文件的播放	145
5.1.2	进行音频播放相关属性的控制	147
5.1.3	后台播放音频及用户交互的优化	152
5.2	iOS 视频开发基础	154
5.2.1	使用 MPMoviePlayerController 向应用中嵌入视频模块	154
5.2.2	MPMoviePlayerController 常用属性与方法解析	155

5.3	视频播放器视图控制器——MPMoviePlayerViewController.....	158
5.4	AVPlayViewController 视频播放框架与画中画开发技术.....	159
5.4.1	使用 AVPlayerViewController 进行视频播放.....	160
5.4.2	iPad 的画中画播放技术.....	162
5.5	实战：“歌手王菲”音频播放器的开发.....	164
5.5.1	工程搭建与 LRC 歌词文件简介.....	164
5.5.2	LRC 歌词解析引擎的设计.....	165
5.5.3	核心播放器引擎的设计.....	168
5.5.4	歌曲列表与歌词显示视图界面的设计.....	173
5.5.5	播放器主页面的实现.....	176
5.5.6	后台播放音频用户交互的处理.....	181
第 6 章	动画开发技术.....	183
6.1	使用 UIImageView 播放图片组帧动画.....	184
6.2	UIView 层动画的应用.....	185
6.2.1	执行 UIView 层过渡动画的 3 个类方法.....	185
6.2.2	创建 UIView 层的阻尼动画.....	186
6.2.3	动画参数配置与组合动画.....	186
6.2.4	UIView 层过渡动画支持的属性.....	188
6.3	使用 commit 方式进行 UIView 层动画的创建.....	189
6.3.1	使用 commit 方式进行 UIView 层过渡动画的创建.....	189
6.3.2	两种 UIView 层动画创建方式的优劣.....	190
6.4	UIView 的转场动画.....	190
6.4.1	重绘 UIView 视图时使用的转场动画.....	191
6.4.2	切换 UIView 视图时使用的转场动画.....	191
6.5	核心动画编程技术——CoreAnimation.....	192
6.5.1	锚点对视图控件几何位置的影响.....	193
6.5.2	色彩梯度层——CAGradientLayer.....	194
6.5.3	视图拷贝层——CAReplicatorLayer.....	194
6.5.4	图形渲染层——CAShapeLayer.....	195
6.5.5	文本绘制层——CATextLayer.....	196
6.5.6	CAAnimation 动画体系介绍.....	197
6.5.7	使用 CABasicAnimation 创建基础动画.....	198
6.5.8	使用 CAKeyframeAnimation 类创建关键帧动画.....	200
6.5.9	CALayer 层的转场动画——CATransition.....	201
6.5.10	CALayer 层的组合动画——CAAnimationGroup.....	202
6.5.11	CATransform3D 变换的应用.....	203
6.6	炫酷的粒子效果.....	205
6.6.1	粒子发射器——CAEmitterLayer.....	205

6.6.2	粒子单元——CAEmitterCell	206
6.6.3	创建粒子火焰动画	207
6.7	播放 GIF 动态图	209
6.7.1	使用 UIWebView 进行 GIF 动态图播放	209
6.7.2	使用 UIImageView 帧动画进行 GIF 动态图播放	209
6.8	实战：小游戏 Flappy Bird 的设计与开发	210
6.8.1	小鸟对象的设计	211
6.8.2	游戏开始界面的设计	213
6.8.3	游戏结束界面的设计	214
6.8.4	Flappy Bird 游戏主框架的搭建	215
第 7 章	传感器开发技术	221
7.1	为应用程序添加手机密码及指纹识别的安全验证	221
7.1.1	使用手机密码为应用程序添加安全验证	222
7.1.2	使用用户指纹为应用程序添加安全验证	223
7.2	使用加速度传感器、螺旋仪传感器与磁力传感器获取设备空间状态	224
7.3	距离传感器的应用	227
7.4	iOS 蓝牙开发技术	228
7.4.1	中心设备管理类 CBCentralManager	229
7.4.2	外围设备管理类 CBPeripheralManager	232
7.5	GPS 应用与地图编程技术	236
7.5.1	进行设备地理位置定位	236
7.5.2	原生地图开发技术	238
7.5.3	在地图中添加大头针及标注	240
7.5.4	在地图视图中添加覆盖物	242
7.5.5	在地图中进行线路导航与附近兴趣点检索	244
7.6	实战：简易蓝牙对战五子棋	248
7.6.1	游戏核心通信类的设计	248
7.6.2	棋盘瓦片的设计	254
7.6.3	核心游戏视图与游戏核心逻辑的设计	256
7.6.4	核心游戏视图控制器的设计	262
第 8 章	界面布局技术	266
8.1	iOS 中传统的 UIViewAutoresizing 布局模式	266
8.1.1	通过代码设置视图控件的 UIViewAutoresizing 模式	267
8.1.2	在 xib 文件中可视化地配置控件的 autoresizing 属性	269
8.2	autolayout 自动布局框架	270
8.2.1	初识 autolayout	270
8.2.2	autolayout 的属性意义与一个简单的自动布局示例	272

8.2.3	使用代码进行 autolayout 布局	275
8.2.4	使用格式化的字符串进行 autolayout 布局对象的创建	277
8.2.5	与约束相关的几个方法	279
8.2.6	使用 autolayout 设计一个高度自适应的聊天输入框及动画优化	279
8.2.7	使用第三方库 SnapKit 进行 autolayout 约束布局	281
第 9 章	数据持久化技术	286
9.1	使用 plist 文件进行轻量级数据持久化管理	286
9.1.1	在工程中读取 Plist 文件数据	286
9.1.2	在程序沙盒 Documents 目录中创建和使用 plist 文件	288
9.1.3	使用 UserDefaults 类进行数据持久化	289
9.2	使用归档技术进行数据模型持久化	290
9.2.1	进行单一系统数据类型的归档与解归档操作	290
9.2.2	对多个对象进行数据归档	291
9.2.3	进行自定义数据模型的归档	292
9.3	小型数据库 SQLite 在 iOS 开发中的应用	293
9.4	核心数据管理框架 CoreData 的使用	296
9.4.1	使用 CoreData 设计数据模型	296
9.4.2	CoreData 编程框架中 3 个重要的类	299
9.4.3	CoreData 编程框架的数据操作	301
9.4.4	使用 CoreData 进行数据与页面的绑定	304
9.5	网络缓存策略	307
9.5.1	为网络请求设置缓存策略	308
9.5.2	应用缓存管理类 NSURLCache 简介	309
第 10 章	提交应用程序到 App Store	310
10.1	使用 Xcode 开发工具进行程序调试	310
10.1.1	使用自定义断点进行代码调试	310
10.1.2	添加全局异常断点	312
10.1.3	使用 LLDB 调试器进行程序调试	312
10.2	Apple 开发者账号的申请	313
10.2.1	几种类型的开发者账号	313
10.2.2	申请开发者账号的过程	314
10.3	进行应用程序打包	317
10.3.1	在 iTunes Connect 中进行应用的创建与配置	317
10.3.2	使用 Xcode 打包与提交 iTunes	323
第 11 章	更多功能与进阶技巧	327
11.1	iOS 通知中心 NotificaitonCenter 的应用	327

11.1.1	通知类 Notification 简介	327
11.1.2	通知中心 NotificationCenter 应用	328
11.2	多线程开发技术	329
11.2.1	使用 Thread 进行线程管理	329
11.2.2	使用 Operation 类与 OperationQueue 类进行多任务管理	331
11.2.3	iOS 中 GCD 编程技术简介	333
11.3	3D Touch 技术的应用	334
11.3.1	3D Touch 的 3 大模块	334
11.3.2	Home Screen Quick Action 使用与相关 API 详解	335
11.4	iOS 中语音识别技术的应用	338
11.4.1	SpeechFramework 框架中的重要类	338
11.4.2	申请用户语音识别权限与进行语音识别请求	338

第 1 章

开发准备

工欲善其事，必先利其器。在学习 iOS 移动开发之前，首先应该将开发环境配置完成并对所需要使用的开发工具进行了解与熟悉。本章首先向读者介绍 iOS 11 系统相比之前系统的一些新特性，使读者对目前主流的 iOS 系统有一个宏观上的了解，然后将一步步演示开发环境的搭建并介绍开发工具 Xcode 的常用功能。

通过本章的学习，读者能够掌握：

- 了解 iOS 11 的新特性和新功能。
- 申请免费的 Apple ID 账号。
- 使用 Xcode 开发工具创建 iOS 工程。
- 使用 Xcode 开发工具编写与调试程序。
- 熟悉 Xcode 工程结构。
- 编写第一个程序 Hello World。
- 使用 Git 工具进行版本管理。
- 使用 GitHub 代码托管平台。

1.1 iOS 11 新特性简述

随着 2017 年 iPhone X 的发布，iOS 系统也迎来了它的又一次重大升级。新版本 iOS 11 在用户交互维度新增了拖放操作，用户在 iOS 系统中可以将元素在不同界面间或不同 APP 间进行传递。在文件管理方面，iOS 11 提供了本地文档与 iCloud 云文档浏览器。除了一些细节上的调整和优化，iOS 11 开发框架中还新增了目前非常受欢迎的机器学习模型和 AR 开发框架。另外，Swift 语言也升级到了 4.0 版本，Swift 语言自从发布以来，就一直褒贬各半，首先其先进的现代编程语言特性的确有很强的安全性和编程效率，另一方面其频繁的更新和接口变动也给开发者的项目维护带来很

大的麻烦。从 Swift 语言的更新内容和趋势来分析，3.0 是一个分界点，Swift 3.0 对语言风格和 API 做了重构性质的升级，4.0 版本则只是完善与补充，并没有比较突出的修改，因此无论是学习还是开发项目，目前的 Swift 语言都是非常适合的。了解 iOS 11 的这些新特性与 Swift 4.0 的相关更新点，相信可以更好地帮助读者学习 iOS 应用程序开发。

1.1.1 新增拖放交互编程接口

在 Mac OS 软件开发时，拖拽交互是一种十分常用的交互方式，在 iOS 11 之前的系统中要实现拖拽交互往往比较困难。iOS 11 系统新引入了拖拽相关的 API，可以帮助开发者快速构建拖拽交互，在 iOS 11 系统中，使用这种 API 进行 APP 的开发为设计提供了一种全新维度的用户交互方式。

拖拽操作在 iPad 上是支持跨应用程序的，用户可以从一个应用中拖取项目，通过 Home 键回到主界面并打开另一个应用程序，然后将被拖拽的项目传递给这个应用程序中。在 iPhone 上，拖拽操作只支持当前应用程序内，用户可以将某个元素从一个界面拖拽到另一个界面，这种维度的操作可以使设计人员在设计产品时，有更大的灵活性。

对于拖拽操作，至少要有两个组件：一个组件作为拖拽源用来提供数据；另一个组件作为拖拽目的用来接收数据。当然，同一个组件既可以是拖拽源也可以是拖拽目的。

任意的 UIView 组件都可以作为拖拽源，让其成为拖拽源其实也十分简单，只需要 3 步：

步骤01 创建一个 UIDragInteraction 行为对象。

步骤02 设置 UIDragInteraction 对象的代理并实现相应方法。

步骤03 将 UIDragInteraction 对象添加到指定的 View 上。

最简单的可拖拽组件的创建示例代码如下：

```
lazy var dragView = { ()->UIView in
    let view = UIView(frame: CGRect(x: 100, y: 100, width: 100, height: 100))
    view.backgroundColor = UIColor.red
    view.addInteraction(self.dragInteraction)
    return view;
}()

lazy var dragInteraction = { ()->UIDragInteraction in
    let dragInteraction = UIDragInteraction(delegate: self)
    dragInteraction.isEnabled = true
    return dragInteraction
}()

func dragInteraction(_ interaction: UIDragInteraction, itemsForBeginning session:
UIDragSession) -> [UIDragItem] {
    let provider = NSItemProvider(object: "Hello World" as NSItemProviderWriting)
    let item = UIDragItem(itemProvider: provider)
    return [item]
}

override func viewDidLoad() {
    super.viewDidLoad()
    self.view.addSubview(self.dragView)
```

拖拽源是数据的提供者，放置目的地就是数据的接收者。同样，对于任何自定义的 UIView 视图，我们也可以让其成为放置目的地，需要以下三步完成：

步骤01 创建一个 UIDropInteraction 行为对象。

步骤02 设置 UIDropInteraction 对象的代理并实现协议方法。

步骤03 将其添加到自定义的视图中。

例如，我们将自定义的 UILabel 组件用来显示拖拽的文案，代码如下：

```
import UIKit

class ViewController: UIViewController, UIDragInteractionDelegate, UIDropInteractionDelegate {

    lazy var dragView = { ()->UIView in
        let view = UIView(frame: CGRect(x: 100, y: 100, width: 100, height: 100))
        view.backgroundColor = UIColor.red
        view.addInteraction(self.dragInteraction)
        return view;
    }()

    lazy var dragInteraction = { ()->UIDragInteraction in
        let dragInteraction = UIDragInteraction(delegate: self)
        dragInteraction.isEnabled = true
        return dragInteraction
    }()

    lazy var dropLabel = { ()->UILabel in
        let label = UILabel(frame: CGRect(x: 10, y: 300, width: 300, height: 30))
        label.backgroundColor = UIColor.green
        label.isUserInteractionEnabled = true
        label.addInteraction(self.dropInteraction)
        return label
    }()

    lazy var dropInteraction = { ()->UIDropInteraction in
        let dropInteraction = UIDropInteraction(delegate: self)
        return dropInteraction
    }()

    func dropInteraction(_ interaction: UIDropInteraction, canHandle session: UIDropSession)
-> Bool {
        return true
    }

    func dropInteraction(_ interaction: UIDropInteraction, sessionDidUpdate session:
UIDropSession) -> UIDropProposal {
        return UIDropProposal(operation: .copy)
    }

    func dropInteraction(_ interaction: UIDropInteraction, performDrop session: UIDropSession)
{
        let _ = session.loadObjects(ofClass: String.self) { (itemArray) in
            self.dropLabel.text = itemArray.first
        }
    }

    func dragInteraction(_ interaction: UIDragInteraction, itemsForBeginning session:
UIDragSession) -> [UIDragItem] {
        let provider = NSItemProvider(object: "Hello World" as NSItemProviderWriting)
```

```
        let item = UIDragItem(itemProvider: provider)
        return [item]
    }

    override func viewDidLoad() {
        super.viewDidLoad()
        self.view.addSubview(self.dragView)
        self.view.addSubview(self.dropLabel)
    }
}
```

上面的代码将我们自定义的拖拽源提供的 Hello World 拖放进了 UILabel 组件中。

1.1.2 其他新增功能

在 iOS 11 系统中新增了访问本地和 iCloud 文档的功能, 开发者可以使用 `UIDocumentBrowserViewController` 和 `UIDocumentBrowserTransitionController` 这两个视图控制器来管理文档浏览。

新增了 MusicKit 开发框架, 开发者可以在应用中访问完整的 Apple Music 音乐目录, 并且提供了更多与 Apple 音乐程序的交互接口。

新增了 ARKit 开发框架, 开发者结合摄像头可以更加容易地构建 AR 体验项目。

新增了用于检测人脸识别、条形码等视觉效果的开发框架。

新增了 CoreML 开发框架, 开发者更便于将机器学习模型集成到应用中。

1.2 熟悉 iOS 开发环境

Xcode 是进行 iOS 应用开发必备的开发软件。Xcode 开发工具功能十分强大且简单易用, 不需要过多的配置, 下载并安装后, 各种环境和模拟器即关联完毕, 对于初学者来说门槛很低。

1.2.1 安装 Xcode 开发工具

由于 iOS 系统的封闭性, 开发 iOS 软件的工具环境并不多, Xcode 是 Apple 公司自己开发的一套针对 OS、iOS、watchOS 和 tvOS 的开发环境, 使用方便且功能十分强大。用户可以在 App Store 上免费获取 Xcode 开发工具。

首先需要申请个人的 AppleID。AppleID 是 Apple 会员的凭证, 也是个人的信息管理凭证。申请个人的 AppleID 是免费的, 登录 www.apple.com/cn/Apple (中国) 官方网站, 在屏幕右上角的购物袋按钮中选择“登录”选项, 如图 1-1 所示。