

*IOS
Interface
Development*

7天玩转 iOS界面开发

深入浅出的讲解 丰富详细的配图
清晰经典的范例 灵活多变的应用

张益珩 编著



中国工信出版集团



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

7天玩转 iOS界面开发

张益珲 编著

电子工业出版社

Publishing House of Electronics Industry

北京•BEIJING

内 容 简 介

本书是一套系统的 iOS UI 开发教程,基本涵盖 iOS 开发中所用到的全部系统 UI 控件,在详细讲解基础运用的基础上,更深层次地分析了高级用法,为有编程基础但不是 iOS 开发者的程序员转 iOS 开发提供了便捷的学习路径。同样,对于没有开发基础,但有 Objective-C 基础的读者而言,也是一套很好的使用教程。

本书适合的读者有:1.转 iOS 开发的其他软件开发者。2.有语法基础的 iOS 开发学习者。3.想更深入理解 iOS UI 开发的开发者。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有,侵权必究。

图书在版编目(CIP)数据

7 天玩转 iOS 界面开发 / 张益琤编著. —北京: 电子工业出版社, 2016.7
ISBN 978-7-121-29045-9

I. ①7… 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 (2016) 第 131859 号

责任编辑: 安 娜

印 刷: 三河市鑫金马印装有限公司

装 订: 三河市鑫金马印装有限公司

出版发行: 电子工业出版社

北京市海淀区万寿路 173 信箱

邮编: 100036

开 本: 787×980 1/16 印张: 16

字数: 320 千字

版 次: 2016 年 7 月第 1 版

印 次: 2016 年 7 月第 1 次印刷

定 价: 55.00 元

凡所购买电子工业出版社图书有缺损问题, 请向购买书店调换。若书店售缺, 请与本社发行部联系, 联系及邮购电话: (010) 88254888, 88258888

质量投诉请发邮件至 zlts@phei.com.cn, 盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式: 010-51260888-819 faq@phei.com.cn。

目 录

第 1 天 iOS 基础 UI 控件 (上)	1
1.1 承载文字的精灵——UILabel	1
1.1.1 创建第一个 iOS 项目 Hello World	1
1.1.2 使用代码重建 Hello World 项目	7
1.1.3 iOS 系统中的 UI 坐标系	8
1.1.4 UILabel 的相关属性设置	9
1.1.5 多行 UILabel 控件与换行模式	11
1.2 最简单的交互控件——UIButton	13
1.2.1 在屏幕上创建一个 UIButton 控件	13
1.2.2 UIButton 控件的交互状态	14
1.2.3 UIButton 类中可以自定义的一些常用属性	15
1.2.4 为 UIButton 对象添加触发方法	18
1.3 专业显示图片的控件——UIImageView	19
1.3.1 使用 UIImageView 在屏幕上创建一个图片元素	20
1.3.2 连续变换播放的图片形成动画——UIImageView 的帧动画	21
1.4 搜索框控件——UISearchBar	23
1.4.1 添加一个搜索框控件	23
1.4.2 UISearchBar 中常用属性的用法	24
1.4.3 UISearchBar 的扩展栏	27
1.4.4 UISearchBar 的代理方法	28
1.5 简洁小巧的切换控件——UISegmentedControl	30
1.5.1 创建和初始化 UISegmentedControl 对象	30

- 1.5.2 UISegmentedControl 中的常用属性与方法 31
 - 1.5.3 UISegmentedControl 中切换模块按钮的增、删、改以及布局操作 32
 - 1.5.4 为 UISegmentedControl 控件对象添加触发方法 35
- 1.6 文本输入交互的基础——UITextField 36
 - 1.6.1 搭建一个用户登录界面 36
 - 1.6.2 为 UITextField 添加挂件 39
 - 1.6.3 监听用户的输入操作 40
 - 1.6.4 收键盘操作 42
- 1.7 连续变化的美——UISlider 43
 - 1.7.1 通过滑块控件控制视图的颜色渐变 43
 - 1.7.2 UISlider 中常用的可自定义属性 44

第 2 天 基础 UI 控件（下） 47

- 2.1 简洁小巧的开关控件——UISwitch 47
 - 2.1.1 创建一个 UISwitch 控件 47
 - 2.1.2 UISwitch 控件中常用属性解析 48
 - 2.1.3 为 UISwitch 控件添加触发方法 49
- 2.2 让等待不再寂寞——UIActivityIndicatorView 49
 - 2.2.1 在视图上创建一个 UIActivityIndicatorView 控件 50
 - 2.2.2 设计一个全局共享的活动指示器控件 52
- 2.3 扁平化风格的进度条——UIProgressView 55
- 2.4 分页控制器——UIPageControl 57
- 2.5 步进控制器——UIStepper 58
 - 2.5.1 创建一个步进控制器控件 58
 - 2.5.2 为 UIStepper 控件添加图片修饰 60
- 2.6 警告框视图——UIAlertView 62
 - 2.6.1 在视图上弹出一个警告框 62
 - 2.6.2 UIAlertView 的代理方法 64
- 2.7 活动列表——UIActionSheet 65
 - 2.7.1 在视图上弹出一个 UIActionSheet 控件 65
 - 2.7.2 UIActionSheet 控件的代理方法 66
- 2.8 统一的警告控制器——UIAlertController 67

2.8.1 关于 UIAlertAction 类	67
2.8.2 使用 UIAlertController 创建警告框和活动列表	68
2.8.3 为警告框风格的 UIAlertController 添加文本输入框	69
第 3 天 高级 UI 控件	71
3.1 自由移动的视图——UIScrollView	71
3.1.1 创建一个 UIScrollView 控件来展示大图片	71
3.1.2 UIScrollView 中常用方法与属性解析	72
3.1.3 UIScrollViewDelegate 协议中的方法	74
3.1.4 关于 UIScrollView 的缩放操作	75
3.2 基础的表格视图——UITableView	77
3.2.1 使用表格视图展示列表数据	77
3.2.2 UITableView 中数据载体 cell 的复用机制	79
3.2.3 数据载体 UITableViewCell	80
3.2.4 UITableView 的编辑模式	82
3.2.5 UITableView 的头视图与尾视图	84
3.2.6 为 UITableView 添加索引栏	85
3.3 高级文本视图——UITextView	86
3.3.1 创建一个 UITextView 控件	86
3.3.2 UITextView 中的常用属性与方法	87
3.3.3 UITextViewDelegate 协议中的常用方法	88
3.4 高级集合布局视图——UICollectionView	90
3.4.1 UICollectionView 控件的优势与布局方式	91
3.4.2 使用 UICollectionView 进行九宫格式的布局	91
3.4.3 创建更加灵活的流式布局	93
3.4.4 参差瀑布流布局	94
3.4.5 使用 UICollectionView 进行圆环布局	98
3.5 选择器控件——UIPickerView	101
3.5.1 创建一个 UIPickerView 控件	102
3.5.2 UIPickerView 选中数据时的回调代理	103
3.6 时间选择器——UIDatePicker	104
3.7 网页视图与 iOS 原生视图的完美结合——UIWebView	106

3.7.1	APP 网络传输安全策略	106
3.7.2	通过网络请求加载 UIWebView	107
3.7.3	通过 HTML 字符串加载 UIWebView	108
3.7.4	通过 NSData 数据加载 UIWebView	109
3.7.5	UIWebView 中常用方法解析	110
3.7.6	UIWebView 的代理方法	111
第 4 天	视图与逻辑的管理者	113
4.1	基础的视图控制器——UIViewController	113
4.1.1	UIViewController 的生命周期	113
4.1.2	UIViewController 的视图层级结构	118
4.1.3	UIViewController 的切换跳转	119
4.2	表格视图控制器——UITableViewController	121
4.2.1	自定义一个 UITableViewController	121
4.2.2	UITableViewController 自带的下拉刷新控件	122
4.3	导航控制器——UINavigationController	124
4.3.1	导航控制器的工作原理	124
4.3.2	使用导航控制器进行多界面搭建	125
4.3.3	关于导航栏 UINavigationController	129
4.3.4	导航按钮 UIBarButtonItem	131
4.3.5	导航控制器的工具栏	133
4.3.6	iOS 8 之后导航控制器的一些有趣功能	134
4.4	标签控制器——UITabBarController	135
4.4.1	标签控制器的工作原理	135
4.4.2	标签控制器的基础用法解析	135
4.4.3	UITabBarItem 的使用	138
第 5 天	iOS 动画开发基础	139
5.1	UIView 层动画的应用	139
5.1.1	执行 UIView 层过渡动画的三个类方法	139
5.1.2	创建 UIView 层的阻尼动画	141
5.1.3	动画参数配置与组合动画	142

5.1.4	UIView 层过渡动画支持的属性	144
5.2	UIView 层动画的另一种创建方式	145
5.2.1	使用 commit 方式进行 UIView 层过渡动画的创建	145
5.2.2	两种 UIView 层动画创建方式的优劣	147
5.3	UIView 的转场动画	147
5.3.1	重绘 UIView 视图时使用的转场动画	147
5.3.2	切换 UIView 视图时使用的转场动画	148
5.4	核心动画编程技术——CoreAnimation	149
5.4.1	锚点对视图控件几何位置的影响	150
5.4.2	色彩梯度层——CAGradientLayer	151
5.4.3	视图备份层——CAReplicatorLayer	152
5.4.4	图形渲染层——CAShapeLayer	153
5.4.5	文本绘制层——CATextLayer	154
5.4.6	CAAnimation 动画体系介绍	155
5.4.7	使用 CABasicAnimation 创建基础动画	157
5.4.8	使用 CAKeyframeAnimation 类创建关键帧动画	160
5.4.9	CALayer 层的转场动画——CATransition	161
5.4.10	CALayer 层的组合动画——CAAnimationGroup	162
5.4.11	CATransform3D 变换的应用	163
5.5	炫酷的粒子效果	165
5.5.1	粒子发射器——CAEmitterLayer	166
5.5.2	粒子单元——CAEmitterCell	168
5.5.3	创建粒子火焰动画	169
5.6	播放 GIF 动态图	171
5.6.1	使用 UIWebView 进行 GIF 动态图播放	171
5.6.2	使用 UIImageView 帧动画进行 GIF 动态图播放	172

第 6 天 iOS 界面布局

6.1	传统的 UIViewAutoresizing 布局模式	175
6.1.1	通过代码来设置视图控件的 UIViewAutoresizing 模式	176
6.1.2	可视化地配置控件的 autosizing 属性	179
6.2	autolayout 自动布局框架	180

6.2.1	初识 autolayout.....	180
6.2.2	autolayout 的属性意义与一个简单的自动布局示例.....	182
6.2.3	使用 Objective-C 风格的方法进行代码 autolayout 布局.....	186
6.2.4	使用格式化的字符进行 autolayout 布局对象的创建.....	188
6.2.5	管理约束相关的几个方法.....	191
6.2.6	使用 autolayout 设计一个高度自适应的聊天输入框及动画优化.....	191
6.2.7	使用第三方库 Masonry 进行 autolayout 约束布局.....	193
第 7 天 iOS 界面开发扩展知识		199
7.1	调用系统拨号与短信息应用.....	199
7.1.1	跳转到系统拨号应用界面.....	199
7.1.2	跳转到系统短消息界面.....	200
7.2	iOS 视频播放技术.....	202
7.2.1	向应用中嵌入视频模块.....	203
7.2.2	MPMoviePlayerController 常用属性与方法解析.....	204
7.3	视频播放器视图控制器——MPMoviePlayerViewController.....	208
7.4	AVPlayerViewController 视频播放框架与画中画开发技术.....	209
7.4.1	使用 AVPlayerViewController 进行视频播放.....	210
7.4.2	iPad 上的画中画播放技术.....	213
7.5	翻书效果的阅读器视图——UIPageViewController.....	215
7.5.1	创建 UIPageViewController 视图控制器.....	216
7.5.2	双翻页效果的 UIPageViewController.....	220
7.6	更加灵活地创建标签文字——NSAttributedString 的应用.....	221
7.6.1	NSAttributedString 类简介.....	222
7.6.2	可变的属性字符串 NSMutableAttributedString.....	222
7.7	毛玻璃效果的应用——界面模糊处理.....	224
7.8	iOS 地图视图编程技术.....	226
7.8.1	进行设备地理位置定位.....	226
7.8.2	原生地图开发技术.....	229
7.8.3	在地图中添加大头针及标注.....	232
7.8.4	在地图视图中添加覆盖物.....	235
7.8.5	在地图中进行线路导航与附近兴趣点检索.....	238

iOS 基础 UI 控件（上）

本章将介绍讲解 iOS 开发中必然用到的一些基础控件，这些控件有些仅仅用于展示文字或者图片，如 UILabel、UIImageView；有些控件则可以进行简单的用户交互，例如 UIButton、UISegmentedControl 等。本章中的这些控件十分简单，却是一个应用程序中必不可少的部分，通过对这些简单控件的封装和扩展，开发者们才能够创造出各种酷炫的界面效果。

1.1 承载文字的精灵——UILabel

如果要问在一款 iOS 应用的开发中，什么控件出现的频率最高，大多数开发者都会说那一定是标签控件（UILabel）。UILabel 用于展示一行或多行的文本，除了展示，其还可以设置文字的相关属性。学习 iOS 界面开发，UILabel 也是最易理解和扩展的控件，通过它，读者将踏进 iOS 界面开发的神奇世界，循序渐进，在双手间的代码中创造出炫酷迷人的 UI 效果。

1.1.1 创建第一个 iOS 项目 Hello World

大多程序员都有 Hello World 情结，这并不奇怪。Hello World 是大多数语言学习教程的第一个示例程序，虽然大多数情况它只是命令行中的一串字符，但却给无数初学者带来了程序世界的真挚问候。关于 UILabel 的学习，我们也从 Hello World 开始。

打开 Xcode 开发工具，在如图 1-1 所示的欢迎窗口中，单击“Create a new Xcode project”选项，创建一个新的工程。



图 1-1 Xcode 初始欢迎界面

建议：如果你对 Xcode 的使用和功能还不太了解，可以查看相关资料先行了解，也可以跟着本书中的步骤来操作，在实践中熟悉 Xcode 的使用。

在模板选择界面中单击 iOS 中的“Application”选项，接着单击“Single View Application”选项，之后单击“Next”按钮进行下一步，如图 1-2 所示。

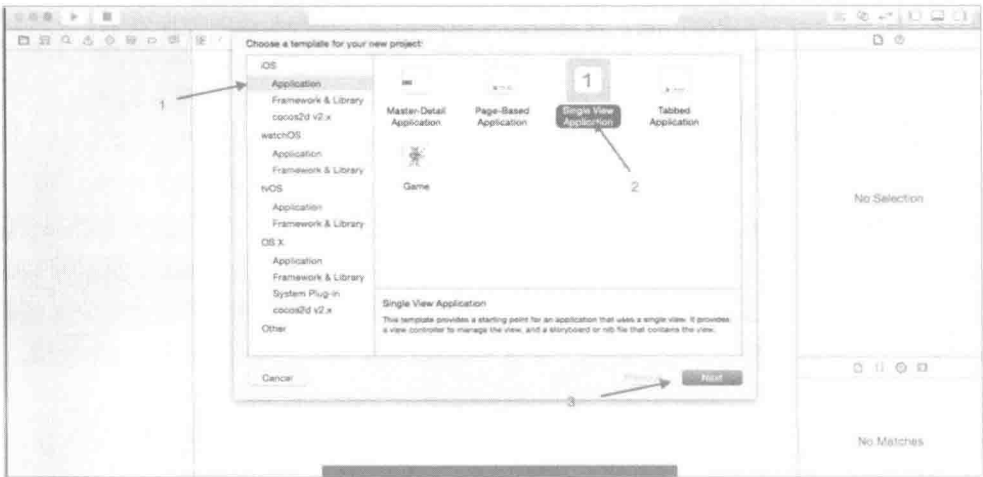


图 1-2 使用 Xcode 创建工程模板

在工程设置界面，需要填写工程的一些基本信息，如图 1-3 所示。Product Name 栏需要填写工程的名字，Organization Name 栏用于填写组织或者公司的名字，Organization Identifier 用来填写组织或公司的标识符，Bundle Identifier 是当前工程的标识符，默认创建为公司组织 ID 加上工程名。工程的标识符可以在工程创建后的配置 plist 文件中进行修改，如果应用程序上线，那么这个 Bundle Identifier 将作为应用程序的唯一标识。Language 栏用于选择开发使用的语言，

观察创建好的 iOS 模板项目工程，可以在 Xcode 界面的左侧导航栏中看到如图 1-5 所示的目录结构，结构中主要分为了两个目录：Hello World 文件夹和 Products 文件夹。Hello World 文件夹和工程名对应，其中存放工程代码及一些素材和配置文件，Products 文件夹用于存放编译后的可执行文件。

在 Hello World 文件夹中，系统模板为开发者创建好了这样几个文件：AppDelegate 类是程序的入口；ViewController 是程序的根视图控制器；以 storyboard 为后缀的文件是一种可视化界面编程的工具文件；Assets.xcassets 是图片素材管理器，用于存放管理项目中需要使用的图片素材；最后 Info.plist 文件存储项目的一些属性配置字段。



图 1-5 模板工程的结构

了解了 iOS 项目的目录结构后，下面来牛刀小试一把。单击 Main.storyboard 文件，Xcode 界面将切换到可视化的编辑界面，如图 1-6 所示。系统默认是选中“Use Size Classes”选项的，初学者可以先将这种自适应的配置关闭，以便于观察效果。取消勾选“Use Size Classes”选项，之后在弹出的对话框中将适配类型设置为 iPhone，最后单击“Disable Size Classes”按钮确认。

可以看到 Xcode 界面中间区域出现了一个类似于 iPhone 屏幕的视图容器，可以通过拖曳的方式将一些系统控件布局在这个 iPhone 屏幕上。这种可视化的编程工具的确很赞，不仅使编程门槛降低很多，便于初学者学习，并且在很大程度提高了界面相关代码的编写效率。在 Xcode 右下角的控件区找到 Label 控件，如图 1-7 所示。



图 1-6 取消勾选“Use Size Classes”选项

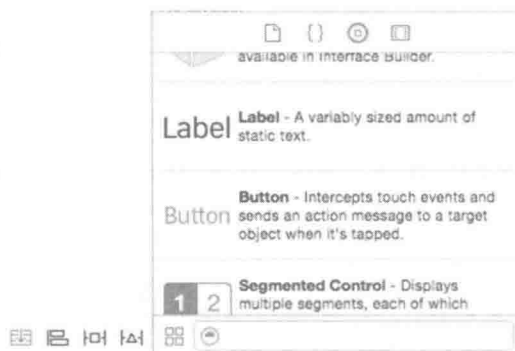


图 1-7 storyboard 中的 Label 控件

选中“Label”项，将其拖曳到视图的中央，如图 1-8 所示。

通过拖曳 Label 控件的边界，可以调整 Label 的大小，双击 Label 控件，可以进入编辑状态。例如，可以在其中输入文字“Hello World”。选择一个 iPhone 模拟器，单击“运行”按钮，过程如图 1-9 所示。

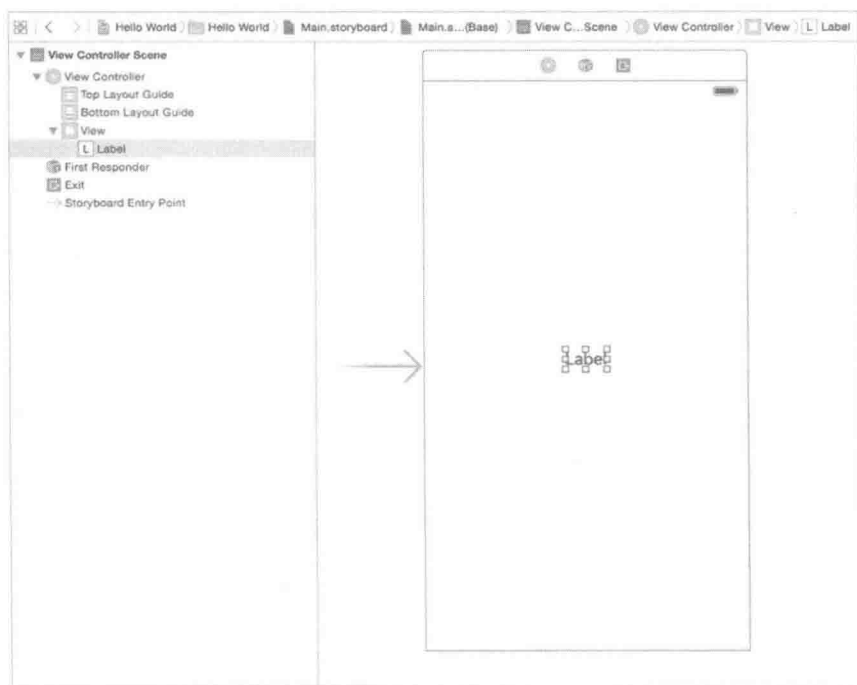


图 1-8 将 Label 控件拖曳进视图



图 1-9 在模拟器中运行工程

小提示：第一次启动 iPhone 模拟器可能会需要一些时间，无须着急，耐心等待即可。

待 iPhone 模拟器运行后，读者的第一个 iOS 程序就完成了，即使用 UILabel 控件在手机屏幕上展示了 Hello World 文字标签，如图 1-10 所示。

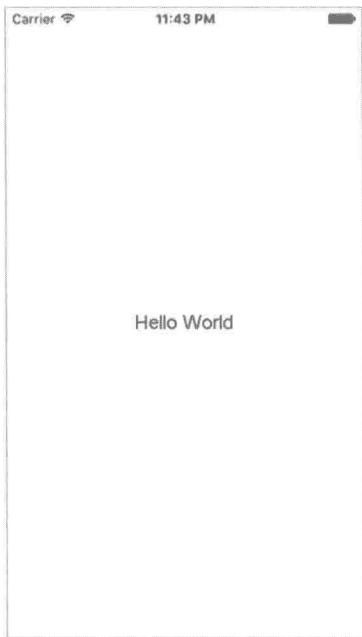


图 1-10 Hello World 工程的运行效果

1.1.2 使用代码重建 Hello World 项目

通过前面的小试，你或许会发现 storyboard 是如此的强大与便捷，有时候开发者甚至可以不写一句代码就创造出炫酷的界面效果，但这对于初学者来说并不一定全是好事。千里之行，始于足下，要想成为一名优秀的 iOS 开发者，对代码的掌控能力远远要比学会使用 storyboard 来拖曳界面重要得多。现在，就通过代码再来重建一下 Hello World 项目吧。

建议：初学开发时以代码为主，熟练之后再使用 storyboard 进行机械重复的 UI 开发工作。

打开前面已经创建好的 Hello World 项目，在 Main.storyboard 文件中将拖入的 Label 控件删掉。单击 ViewController.m 文件，这时 Xcode 的主界面将变为编码区，ViewController 类其实就是在 Main.storyboard 中看到的根视图控制器，从 storyboard 文件中向其上面添加控件和在 ViewController.m 文件中添加控件的效果相同。

在 ViewController 的 viewDidLoad 方法中编写如下代码：

```
- (void)viewDidLoad {
    [super viewDidLoad];
    //创建 UILabel 控件
    UILabel * label = [[UILabel alloc] initWithFrame:CGRectMake(100, 100, 120,
40)];
    //设置 Label 控件文字
    label.text = @"Hello World";
    //将控件添加到视图上
    [self.view addSubview:label];
}
```

编写上面的代码，运行模拟器后，可以看到屏幕上显示了写有“Hello World”的 Label 标签控件，后面我们会对这段代码做详细的讲解。

1.1.3 iOS 系统中的 UI 坐标系

在 1.1.2 节的代码中，下面这句代码完成了 UILabel 标签控件的创建与初始化：

```
UILabel * label = [[UILabel alloc] initWithFrame:CGRectMake(100, 100, 120, 40)];
```

alloc 方法在系统中为 label 指针分配了地址空间，initWithFrame:方法是 UILabel 类的一个初始化方法，其中需要传入一个 CGRect 类型的参数，这个参数决定了 UILabel 控件的位置和尺寸。

CGRectMake()方法用于创建一个 CGRect 对象，这个方法中需要传入四个浮点类型的参数，分别代表着控件的 X 坐标、Y 坐标、宽度和高度。例如，1.1.2 节的代码中传入的 100、100、120 和 40 可以理解为在 X 轴坐标为 100 个单位、Y 轴坐标为 100 个单位的位置，创建一个宽度为 120 个单位、高度为 40 个单位的 UILabel 控件。

确定一个位置时，除了坐标外，还需要一个确定的坐标系为参照，不然坐标是没有任何意义的。在数学中，坐标系的规定为横向为 X 轴，向右为正；纵向为 Y 轴，向上为正，如图 1-11 所示。在 iOS 的 UI 系统中，坐标系与其相似，有略微不同。在 iOS 的 UI 系统中，横向为 X 轴，向右为正；纵向为 Y 轴，不同的是，Y 轴是向下为正，如图 1-12 所示。可以这样理解，我们的手机屏幕就是一个坐标平面，屏幕的左上角的坐标系为原点，向右，X 坐标渐增；向下，Y 坐标渐增。