



微信扫描二维码观看视频

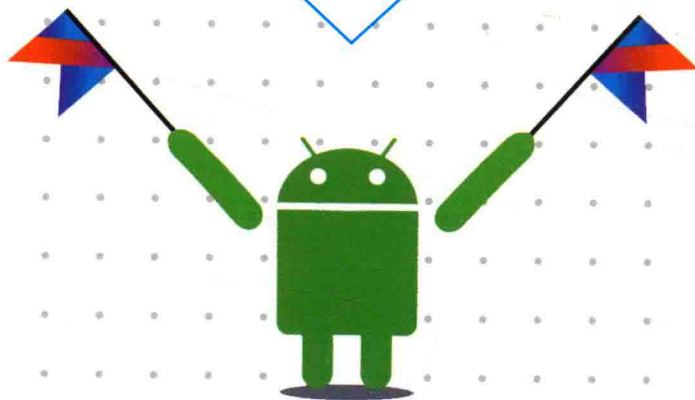
Kotlin程序开发 入门精要

李宁◎编著

独创闯关式学习

书 + 视频 + 测试

三位一体



一本全面介绍 Kotlin 开发知识的书
全面涵盖 Kotlin 开发的各種核心技术
案例驱动，让读者尽快进入实战角色



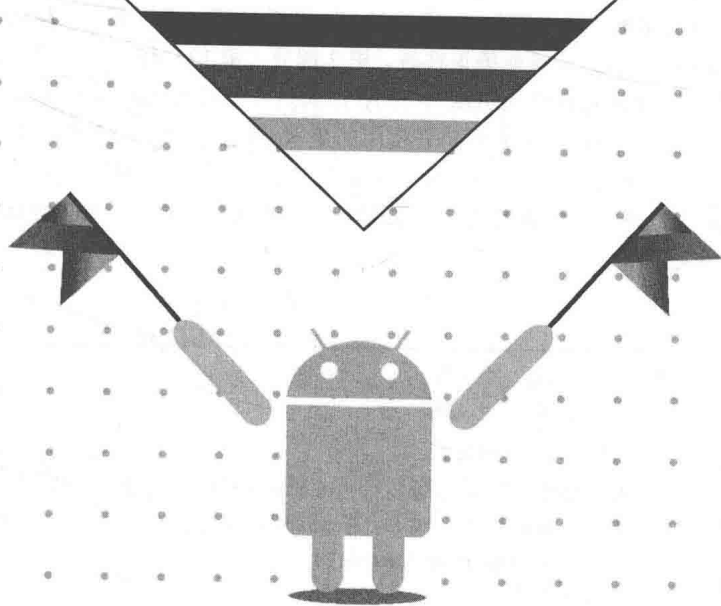
中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

Kotlin 程序开发 入门精要

李宁◎编著



人民邮电出版社
北京

图书在版编目(CIP)数据

Kotlin程序开发入门精要 / 李宁编著. -- 北京 :
人民邮电出版社, 2017.10
ISBN 978-7-115-46752-2

I. ①K… II. ①李… III. ①JAVA语言—程序设计
IV. ①TP312.8

中国版本图书馆CIP数据核字(2017)第213783号

内 容 提 要

本书分3部分讲解 Kotlin, 第1部分(第1~11章)是 Kotlin 语言的基础部分, 主要介绍了 Kotlin 的基础知识、语法以及大量的“语法糖”, 如搭建 Kotlin 开发环境、数据类型、控制流、类、对象、接口、扩展、委托、Lambda 表达式、操作符重载等。第2部分(第12~15章)主要介绍了如何用 Kotlin 开发 Android App。由于 Kotlin 可以调用 JDK 中的 API, 所以在使用 Kotlin 开发 Android App 的过程中, 很多都是调用 JDK 的 API 实现的, 但开发语言使用的是 Kotlin。因此, 这一部分详细介绍了用 Kotlin 开发 Android App 需要掌握的核心知识, 如 Activity、组件、布局、流文件、SQLite 数据库、网络等技术。第3部分(第16章)是一个完整的案例: 欧瑞天气。通过这个项目, 可以让读者了解利用 Kotlin 开发 Android App 的全过程。

◆ 编 著 李 宁

责任编辑 张 涛

责任印制 焦志炜

◆ 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>

北京鑫正大印刷有限公司印刷

◆ 开本: 800×1000 1/16

印张: 20

字数: 427千字

印数: 1~2 500册

2017年10月第1版

2017年10月北京第1次印刷

定价: 69.00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: 京东工商广登字 20170147号

前言

Android 到现在已经发展了 10 年了，从最初不怎么完善发展到现在，形成了一个由成千上万的 App 支撑，可以运行在包括手机、平板电脑、电视、汽车、手表、无人机在内的几乎所有智能设备中的完整生态系统，这一切足以让 Android 的拥有者 Google 感到无比自豪。

不过有一件事一直让 Google “很不爽”，那就是开发 Android App 的核心语言 Java 并不属于 Google，而属于 Oracle。而且 Oracle 一直在和 Google 打关于 Java 的“官司”，尽管 Google 赢得了“官司”，但也让 Google 清楚地看到，如果编程语言不掌握在自己的手里，那么总会受制于人，所以 Google 在数年时间里进行了多次尝试，如推出了 Go 语言。不过 Go 语言和 C++ 类似，并不适合开发 Android App。所以 2017 年 5 月之前，开发 Android App 的主要编程语言仍然是 Java。

然而在 2017 年 5 月的 I/O 大会上，Google 突然宣布，Kotlin 成为开发 Android 的一级编程语言，这就意味着，从这一刻开始，开发 Android App 可以使用两种语言：Java 和 Kotlin。

那么 Kotlin 是什么呢？就在 2017 年 5 月的 I/O 大会之前，我也不太清楚这个世界上还有一种叫 Kotlin 的编程语言，相信大多数读者和我一样，在此之前都不太了解什么是 Kotlin。其实与 Kotlin 类似的语言有很多，如 Scala，它们都是基于 JVM 的语言，也就是说，Kotlin 和 Scala 一样，都只提供了编译器，没有提供运行环境。运行 Kotlin 和 Scala 程序，需要将源代码编译成 Byte Code，然后在 JVM 上运行。

在刚开始接触 Kotlin 时，我感到很奇怪，现在基于 JVM 的语言非常多，例如，编写了 Spark 的 Scala 语言，还有 Groovy、JRuby、Clojure 等，那么 Google 为什么单独偏爱 Kotlin 呢？随着我对 Kotlin 的深入接触，逐渐对 Google 选择 Kotlin 的决定表示认可！

Kotlin 作为第二种开发 Android App 的核心编程语言，肯定是与 Java 有很大差异的，而且这种差异要弥补 Java 的不足或增强 Java 的功能。所以要求都是正向的差异。

Java 尽管历史悠久，应用众多，但也有很多不足的地方，例如，我们经常会遇到 NullPointerException 异常，这是个很讨厌的异常，产生异常的原因是因为访问了空对象的成员。Kotlin 巧妙地解决了这个问题，让访问空对象的成员不再抛出异常，而是直接返回 NULL。Kotlin 除了解决 Java 的遗留问题外，还加了很多“语法糖”，例如，Kotlin 支持扩展和操作符重载，这是两个非常大的“语法糖”。通过扩展，可以在没有源代码的情况下，为类添加方法和属性，通过操作符重载，可以让特定的类型支持原来没有的操作，如为字符串类型添加乘法和除法。总之，Kotlin 单凭语言本身就比较 Java 酷了不少。不过这些理由并不足以让 Google 对 Kotlin 如此偏爱，因为其他同类语言，如 Scala，也同样添加了很多“语法糖”。

那么 Kotlin 还有什么其他特性呢？其实 Kotlin 不仅可以在 JVM 上运行，还可以直接将

Kotlin 源代码转换为 JavaScript。这样一来，从理论上讲，Kotlin 可以在任何支持 JavaScript 的环境中运行，如 Web 应用、React Native (Android 和 iOS)、微信公众号、微信小程序、Node.js、Chrome 插件等。另外，还有一些地方是 JavaScript 做不到的，如开发本地应用。但 Kotlin 可以做到，Kotlin 不仅仅可以生成 JavaScript 代码，还可以直接编译成本地代码，如 Windows 的 exe 文件、iOS App 等，这样一来，Kotlin 几乎可以开发所有类型的应用了。所以从理论上讲，Kotlin 才是真正的跨平台语言，Kotlin 可以直接或间接地开发各种类型的应用。

说了这么多，相信大家一定已经清楚了 Kotlin 的强大，以及 Google 为什么选择了 Kotlin 的原因，当然，除了技术原因外，选择 Kotlin 还有一个原因，就是 Google 的 Android Studio 是基于 IntelliJ IDEA 的社区版本开发的，而 JetBrains 公司开发了 IntelliJ IDEA 和 Kotlin。所以，可以说 Android Studio 和 Kotlin 是师出同门，这样它们之间更容易兼容。

既然 Kotlin 这么强大，而且选择 Kotlin 的理由非常多，如开发效率高、语法优美、能跨平台开发、得到 Google 的强力支持。那么我们还等什么呢！赶紧通过本书将 Kotlin 的知识装进我们的大脑才是正确的做法！

本书适合 Kotlin 爱好者学习、程序员阅读，也可以作为大专院校和培训学校的教材。

· 作 者

如何免费获取视频课程

随本书赠送给读者有大量相关领域的免费视频课程。这些视频课程是离线观看的，使用专门的播放器，可在 Mac OS X、Windows、Android 和 iOS 平台上播放。读者在购买本书后，需要认证读者才可以观看。认证读者的步骤如下。

(1) 扫描本书封面右上角的二维码，关注欧瑞学院公众号。

(2) 进入公众号后，点击“我要”→“扫一扫”菜单项，扫描随书赠送的优惠卡背面右侧的二维码（需要刮掉涂层）。扫描优惠卡后，会赠送价值 300 元的优惠券，可以购买欧瑞学院付费课程。

(3) 在公众号中点击“我要”→“认证读者”菜单项，如图 1 所示。

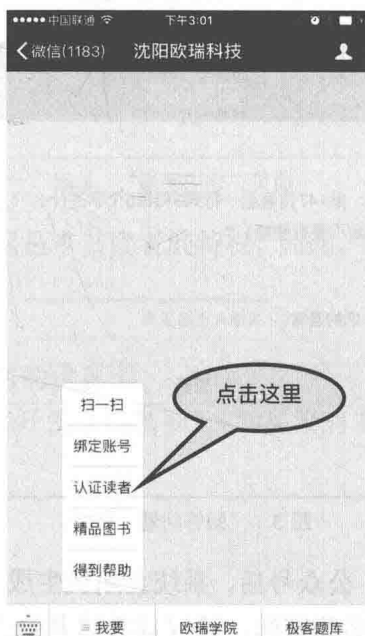


图 1 点击“认证读者”菜单项

(4) 进入“认证读者”页面，如图 2 所示，点击要认证的图书，会进入“回答问题”页面，如图 3 所示。

(5) 在“回答问题”页面会出现一个问题，只要回答正确，即可成功认证读者。当成功认证读者后，公众号会自动发放赠送的视频课程和专用播放器的下载地址和密码，以及具体的使用方法。要注意的是，一个优惠卡只能认证一本书，每本书只能认证一次。



图2 “认证读者”页面



图3 “回答问题”页面

(6) 第一次关注“欧瑞学院”公众号后，系统会随机生成一个用户名和密码，这个用户名和密码也是观看视频课程的用户名和密码。建议读者将其修改成自己容易记忆的用户名和密码。修改的方法是在公众号中点击“我要”→“绑定账号”菜单项，进入“账号中心”页面。要修改账号，点击“修改账号”列表项，在“账号”文本框中输入新的账号（注意，账号只能修改一次，请确认修改无误后再提交），点击“提交”按钮即可修改账号（账号不能是已经存在的，否则无法成功修改账号）。如果要修改密码，点击“修改密码”列表项，在“新密码”文本框中输入新密码即可，然后在“再次输入”文本框中输入同样的密码，点击“提交”按钮即可修改密码。“账号中心”页面如图4所示。

中国联通 下午6:19

< 返回 账号中心

修改账号 ^

每个微信只能绑定一个账号且只能修改一次!

账号

geekori

提交

修改密码 ^

新密码

再次输入

提交

绑定原有账号 v

图4 “账号中心”页面

观看随书赠送的免费视频课程是本书读者的特权，因此，需要满足下面3个条件才能观看赠送的视频课程。

- (1) 关注欧瑞学院公众号
- (2) 手中有一张欧瑞学院赠送的优惠卡
- (3) 手中有一本《Kotlin 程序开发入门精要》(如果没有书无法回答认证读者的问题)

目 录

第 1 章 Kotlin 开发环境搭建.....1	第 2 章 Kotlin 基础知识..... 35
1.1 Kotlin 概述.....1	2.1 Kotlin 的基本语法..... 35
1.1.1 Kotlin 是什么.....1	2.1.1 定义变量..... 35
1.1.2 为什么 Kotlin 突然成为热门.....2	2.1.2 定义函数..... 36
1.1.3 Kotlin 相对于 Java 有哪些优势.....2	2.1.3 注释..... 37
1.1.4 Kotlin 能做什么.....3	2.2 基础数据类型..... 37
1.2 在线运行 Kotlin 代码.....3	2.2.1 数值类型..... 38
1.3 安装和配置 Kotlin.....6	2.2.2 字符类型..... 39
1.3.1 安装和配置 JDK.....6	2.2.3 布尔类型..... 41
1.3.2 Windows 下安装和 配置 Kotlin.....9	2.2.4 数组..... 41
1.3.3 Mac OS X 下安装和 配置 Kotlin.....10	2.2.5 字符串..... 42
1.4 测试 Kotlin 编译和运行环境.....10	2.2.6 字符串模板..... 43
1.4.1 命令行方式使用 Kotlin.....11	2.3 包 (Package)..... 43
1.4.2 Kotlin 的 REPL 环境.....13	2.4 控制流..... 46
1.4.3 在命令行中运行脚本文件.....14	2.4.1 条件语句..... 46
1.4.4 在浏览器中运行 Kotlin 程序.....14	2.4.2 when 语句..... 46
1.4.5 使用 Node.js 运行 Kotlin 程序.....16	2.4.3 for 循环..... 48
1.4.6 在微信小程序中运行 Kotlin 程序.....17	2.4.4 while 循环..... 49
1.4.7 用 Kotlin 开启全栈开发模式.....21	2.5 小结..... 50
1.5 Kotlin 中的语法糖.....22	第 3 章 类和接口..... 51
1.6 Kotlin 的集成开发环境 (IDE).....24	3.1 类的声明..... 51
1.6.1 IntelliJ IDEA 的 Kotlin 基础开发 环境.....25	3.2 构造器..... 51
1.6.2 使用 Android Studio 和 Kotlin 开发 Android App.....29	3.2.1 主构造器..... 51
1.7 小结.....34	3.2.2 第二构造器..... 53
	3.2.3 Kotlin 中的 Singleton 模式..... 54
	3.2.4 Kotlin 函数中的默认参数..... 55
	3.2.5 创建类的实例..... 58
	3.3 类成员..... 58

3.3.1 属性的基本用法	58	5.1.2 对象复制	84
3.3.2 属性的 getter 和 setter 形式	60	5.1.3 数据类成员的解构	85
3.3.3 保存属性值的字段	60	5.2 封闭类 (Sealed Classes)	86
3.3.4 函数	61	5.3 小结	86
3.3.5 嵌套类	63	第 6 章 泛型	87
3.4 修饰符 (Modifiers)	64	6.1 泛型基础	87
3.5 类的继承	65	6.2 类型变异	88
3.5.1 Kotlin 类如何继承	65	6.3 类型投射	90
3.5.2 重写方法	65	6.4 星号投射	92
3.5.3 重写属性	66	6.5 泛型函数	92
3.6 接口	67	6.6 泛型约束	93
3.7 抽象类	68	6.7 小结	93
3.8 小结	68	第 7 章 对象和委托	94
第 4 章 枚举类和扩展	69	7.1 对象	94
4.1 枚举类	69	7.1.1 对象表达式	94
4.1.1 枚举类的基本用法	69	7.1.2 声明匿名对象	97
4.1.2 为枚举值指定对应的数值	70	7.1.3 访问封闭作用域内的变量	97
4.1.3 枚举类的其他功能	71	7.1.4 陪伴对象	99
4.2 扩展	72	7.2 委托	99
4.2.1 扩展原生 API	72	7.2.1 类的委托	99
4.2.2 扩展自定义类	73	7.2.2 委托属性	100
4.2.3 成员函数冲突的解决方案	75	7.2.3 委托类的初始化函数	102
4.2.4 扩展属性	76	7.2.4 委托的前提条件	104
4.2.5 扩展伴随对象 (Companion Object)	76	7.3 标准委托	104
4.2.6 扩展的范围	77	7.3.1 惰性装载	105
4.2.7 在类中使用扩展	77	7.3.2 可观察属性	105
4.2.8 调用特定类的成员函数	78	7.3.3 阻止属性的赋值操作	106
4.2.9 扩展成员的继承	79	7.3.4 Map 委托	107
4.3 小结	80	7.3.5 MutableMap 委托	108
第 5 章 数据类和封闭类	81	7.4 小结	108
5.1 数据类 (Data Class)	81	第 8 章 高阶函数与 Lambda 表达式	109
5.1.1 使用数据类	81	8.1 高阶函数	109

8.2 Lambda 表达式与匿名函数.....	110	10.3.3 常用工具函数	135
8.2.1 函数类型	111	10.4 类型检查与类型转换	136
8.2.2 Lambda 表达式的语法	111	10.4.1 is 与 !is 操作符	136
8.2.3 匿名函数	112	10.4.2 智能类型转换	138
8.2.4 闭包 (Closure)	113	10.4.3 强行类型转换	139
8.3 小结	113	10.5 this 表达式	140
第 9 章 函数	114	10.6 相等判断	140
9.1 函数基本用法	114	10.7 操作符重载	141
9.2 使用中缀标记法调用函数	114	10.7.1 一元操作符重载	141
9.3 单表达式函数	115	10.7.2 二元操作符	142
9.4 函数参数和返回值	116	10.7.3 方括号操作符重载	144
9.4.1 可变参数	116	10.7.4 赋值操作符重载	146
9.4.2 返回值类型	117	10.7.5 逻辑操作符重载	147
9.5 函数的范围	117	10.8 小结	149
9.5.1 局部函数	117	第 11 章 其他 Kotlin 技术 (2)	150
9.5.2 成员函数	118	11.1 null 值安全性	150
9.6 泛型函数	118	11.1.1 可为 null 的类型与不可为 null 的类型	150
9.7 内联函数	119	11.1.2 在条件语句中进行 null 检查	151
9.7.1 让 Lambda 表达式内联进 函数	119	11.1.3 安全调用操作符	152
9.7.2 内联部分 Lambda 表达式	121	11.1.4 Elvis 操作符	152
9.7.3 非局部返回 (Non-local return)	124	11.1.5 !!操作符	153
9.7.4 实体化的类型参数 (Reified type parameter)	125	11.1.6 安全的类型转换	153
9.7.5 内联属性	126	11.1.7 可为 null 的类型构成的集合	153
9.8 小结	127	11.2 异常类	153
第 10 章 其他 Kotlin 技术 (1)	128	11.3 注解 (Annotations)	155
10.1 数据解构	128	11.3.1 使用注解	156
10.2 集合	130	11.3.2 注解类的构造器	157
10.3 值范围	132	11.3.3 Lambda 表达式	158
10.3.1 值范围的应用	132	11.4 反射 (Reflection)	158
10.3.2 值范围的工作原理	134	11.4.1 类引用 (Class Reference)	159
		11.4.2 枚举类成员	159
		11.4.3 动态调用成员函数	161
		11.4.4 动态调用成员属性	161

11.5 小结	163	13.2.2 按钮组件 (Button)	202
第 12 章 Android 的窗口——Activity	164	13.2.3 文本编辑组件 (EditText)	206
12.1 什么是 Activity	164	13.2.4 图像组件 (ImageView)	210
12.2 Activity 的基本用法	164	13.2.5 进度条组件 (ProgressBar)	213
12.2.1 创建一个不包含任何 Activity 的 Android 工程	165	13.2.6 对话框组件 (AlertDialog)	216
12.2.2 手动创建 Activity	167	13.3 布局详解	217
12.2.3 创建和加载布局	168	13.3.1 线性布局 (LinearLayout)	217
12.2.4 在 AndroidManifest 文件中注册 Activity	172	13.3.2 相对布局 (RelativeLayout)	222
12.2.5 编译和运行 Android 工程	172	13.3.3 帧布局 (FrameLayout)	226
12.2.6 为 Activity 添加新组件	173	13.3.4 百分比布局 (PercentFrameLayout)	227
12.2.7 为 Activity 添加逻辑代码	174	13.4 小结	230
12.2.8 为 Activity 添加 Toast	175	第 14 章 永久保存数据的方式—— 持久化技术	231
12.2.9 关闭 Activity	178	14.1 SharedPreferences 存储	231
12.3 使用 Intent 连接多个 Activity	178	14.1.1 将数据存储在 SharedPreferences 中	231
12.3.1 使用显式 Intent	179	14.1.2 从 SharedPreferences 读取 数据	234
12.3.2 使用隐式 Intent	182	14.1.3 利用 SharedPreferences 存取用户名 和密码	235
12.3.3 为隐式 Intent 设置更多的过滤 条件	185	14.2 文件流操作	239
12.3.4 使用隐式 Intent 访问 系统 App	187	14.2.1 openFileOutput 和 openFileInput 方法	239
12.4 向 Activity 中传递数据	189	14.2.2 读写 SD 卡上的文件	242
12.5 从 Activity 返回数据	191	14.3 App 权限的授予和拒绝	248
12.6 Activity 的生命周期	192	14.4 读写 JSON 格式的数据	250
12.7 记录当前活动的 Activity	196	14.5 SQLite 数据库	257
12.8 小结	197	14.5.1 SQLite 数据库管理 工具	257
第 13 章 Android App 的装饰工具—— UI 组件与布局	198	14.5.2 创建 SQLite 数据库和 数据表	261
13.1 如何编写程序界面	198	14.5.3 向 SQLite 数据表中插入 数据	264
13.2 常用的 UI 组件	198		
13.2.1 文本显示组件 (TextView)	198		

14.5.4 删除 SQLite 数据表中的数据	265	第 16 章 项目实战——欧瑞天气 App	283
14.5.5 更新 SQLite 数据表中的数据	266	16.1 项目概述	283
14.5.6 查询 SQLite 表中的数据	267	16.2 添加依赖	284
14.5.7 将数据库与 App 一起发布	268	16.3 实现主窗口	285
14.6 小结	270	16.4 显示地区列表	286
第 15 章 连接外部世界——网络技术	271	16.4.1 描述城市信息的数据类	286
15.1 WebView 组件	271	16.4.2 处理 JSON 格式的城市列表信息	286
15.2 使用 HTTP 与服务端进行交互	273	16.4.3 天气信息描述类	288
15.2.1 使用 HttpURLConnection	273	16.4.4 获取城市信息的对象	290
15.2.2 使用 OkHttp	276	16.4.5 在 ListView 组件中显示地区列表	293
15.3 小结	282	16.5 显示天气信息	298
		16.6 小结	303

第1章 Kotlin 开发环境搭建

尽管 Kotlin 不是刚刚面世的编程语言，但以前一直未受到足够的重视，直到 Google 公司在 2017 年的 I/O 大会上宣布 Kotlin 成为开发 Android App 的一级语言后，Kotlin 才迅速“走红”。那么 Kotlin 到底有什么优于 Java 的地方呢？以至于让互联网巨头 Google 公司如此垂青。如果各位读者想知道原因，那就继续阅读本章后面的内容吧！在这一章里会对 Kotlin 进行全面的介绍，包括 Kotlin 迅速“蹿红”的原因、如何安装 Kotlin，以及 Kotlin 到底能做什么。

1.1 Kotlin 概述

本节主要介绍什么是 Kotlin，以及 Kotlin 较 Java 到底有什么特别之处，足以让 Google 公司将其选为开发 Android App 的一级语言。

1.1.1 Kotlin 是什么

可能很多读者看到 Kotlin 这个单词会感到很陌生，这很正常。就和 2007 年以前一样，在苹果公司推出 iPhone 以及相应的开发工具之前，Objective-C 就鲜为人知，至少在国内是这样的。上面提到的 Objective-C 是一种编程语言，而本书的主题是 Kotlin，那么首先要回答的是，Kotlin 到底是什么呢？没错，Kotlin 和 Objective-C 一样，是一种编程语言。

Kotlin 是由 JetBrains 创建的基于 JVM 的编程语言，那么 JetBrains 又是什么呢？我相信很多 Java 程序员使用过 IntelliJ IDEA，这个非常棒的 Java IDE（集成开发环境）就是 JetBrains 的杰作。如果你没听过或没用过 IntelliJ IDE，那么也没有太大关系。相信阅读本书的读者或多或少都了解一些 Android 的知识，Android 官方推荐的 IDE 是 Android Studio，简称 AS。这个 IDE 就是基于 IntelliJ IDEA 社区版开发的。JetBrains 旗下不只有 IntelliJ IDEA 一款产品，Kotlin 也是 JetBrains 旗下的一款产品，一种编程语言。这种编程语言运行在 JVM 上，也就是 Kotlin 编译器会将 Kotlin 源代码编译成 Java Byte Code（Java 字节码），可以直接运行在 JVM 上。从这一点看出，在技术层面，Java 和 Kotlin 是同一个级别的，都以 Java Byte Code 形式运行在 JVM 上。当然，Kotlin 编译器还可以将 Kotlin 源代码编译生成 JavaScript 代码，以便在没有 JVM 的情况下运行。在未来，Kotlin 编译器还可以将 Kotlin 源代码编译生成本地代码，完全脱离任何

虚拟机运行，也就是说，Kotlin 相对 Java 的优势之一是多目标编译语言，而 Java 只能编译生成 Java Byte Code (.class 文件)。

1.1.2 为什么 Kotlin 突然成为热门

尽管 Kotlin 已经推出有很长一段时间了，但并不怎么出名，可能是因为 JetBrains 对它的推广力度不够，加之现在的编程语言实在太多了，所以 Kotlin 就像一块石头扔进了大海，不见了踪影。

那么为什么现在 Kotlin 突然成为热门了呢？原因也很简单，Kotlin 可以开发 Android App，而且被 Google 公司选为开发 Android App 的一级语言，即在 Android Studio 3.0 及以上版本中会支持利用 Kotlin 语言开发 Android App。这也就意味着，到目前为止，开发 Android 本地 App 可以使用 Java 和 Kotlin 两种编程语言。这就很像开发 iOS App 的场景了，可以使用 Objective-C 和 Swift 两种语言开发 iOS App。因此，很多人把 Kotlin 比作 Android 世界的 Swift。而且 Kotlin 和 Swift 的确都是很棒的编程语言，都加入了很多“语法糖”，可以大幅提高程序开发的效率。

1.1.3 Kotlin 相对于 Java 有哪些优势

可能很多读者会问，既然有了 Java，为什么 Google 公司还要选择 Kotlin 来开发 Android App 呢？Kotlin 相比 Java 有哪些优势呢？

在这一节我们来简单了解一下 Kotlin 的特点，通过这些介绍，我们可以体会到 Kotlin 的优势所在。

- 更容易学习：Kotlin 是一门包含很多函数式编程思想的面向对象编程语言，而且相比 Scala 语言更容易学习。
- 轻量级：相比其他编程语言，Kotlin 函数库更小。由于 Android 存在 65K 方法数限制，使得这一点显得更为重要。虽然使用 ProGuard 或者打包成多个 dex 能够解决这个问题，但是所有这些解决方案都会提高复杂性，并增加调试的时间。Kotlin 函数库方法数小于 7000 个，相当于 support-v4 (Android 1.6) 的大小。
- 高度可互操作性：Kotlin 可以和其他 Java 类库友好且简单地进行互操作。Kotlin 团队在开发这门新语言时正是秉承了这个中心思想。他们希望可以使用 Kotlin 继续开发现有的使用 Java 语言编写的工程，而不是重写所有代码。因此，Kotlin 需要能够和 Java 很好地进行互操作。
- 非常好地集成 Android Studio 及 Gradle：Kotlin 有一个专门用于 Android Studio 的插件，以及另一个专门用于 Gradle 的插件。而且即将推出的 Android Studio 3.0 已经集成了 Kotlin，因此在 Android 工程中开始使用 Kotlin 并不困难。

当然，Kotlin 还有很多语法层面的特性，如数据模型类、空类型安全、扩展函数等，这些技术将会在后面的章节介绍中展开。

1.1.4 Kotlin 能做什么

从前面的内容可以知道, Kotlin 可以用来开发 Android App, 那么除了 Android App, Kotlin 还可以做什么呢? 在本节我们就来一窥究竟。

1. 服务端开发

既然 Kotlin 是基于 JVM 的编程语言, 那么自然而然就可以使用所有基于 JVM 的服务端框架。下面是几个 Kotlin 官方文档推荐的框架。

- **Spring:** 一种开源框架, 是为了解决企业应用程序开发复杂性问题而创建的。从 Spring 5 开始, Spring 就已经支持 Kotlin 的新特性了, 并可以使用 Spring 在线生成器 (<https://start.spring.io>) 生成基于 Kotlin 的工程。
- **Vert.x:** 用于建立基于 JVM 的响应式 Web 应用的框架。
- **Ktor:** 由 JetBrains 发布的一款基于 Kotlin 的本地 Web 框架
- **kotlinx.html:** 是一种 DSL (领域专用语言), 用于在 Web 应用中生成 HTML。Kotlin 服务端框架和 kotlinx.html 的关系就像 JSP 和 FreeMarker 的关系一样, FreeMarker 是基于 Java 的模板引擎。使用 FreeMarker, 可以不依赖于 HTML 或其他技术, 可以根据需要生成 HTML 或其他东西, 也就是一种与具体渲染技术无关的技术。

2. 以 JavaScript 方式运行

Kotlin 提供了生成 JavaScript 源代码的能力, 也就是将 Kotlin 代码直接转换为 JavaScript 代码。目前, 支持 ECMAScript 5.1 标准, 未来会支持 ECMAScript 6。

注意, 如果将 Kotlin 代码转换为 JavaScript 代码, 在 Kotlin 代码中只能包含 Kotlin 标准库, 不能包含任何 JDK API 以及任何第三方的 Java Library API, 任何不属于 Kotlin 本身 (Kotlin 语句和标准库) 的部分在转换的过程中将被忽略。

3. 开发 Android App

这一部分在上文提到过, Kotlin 和 Java 一样, 都可以开发 Android App, 而且 Kotlin 和 Java 源代码文件可以在同一个工程中, 可以联合进行调试。

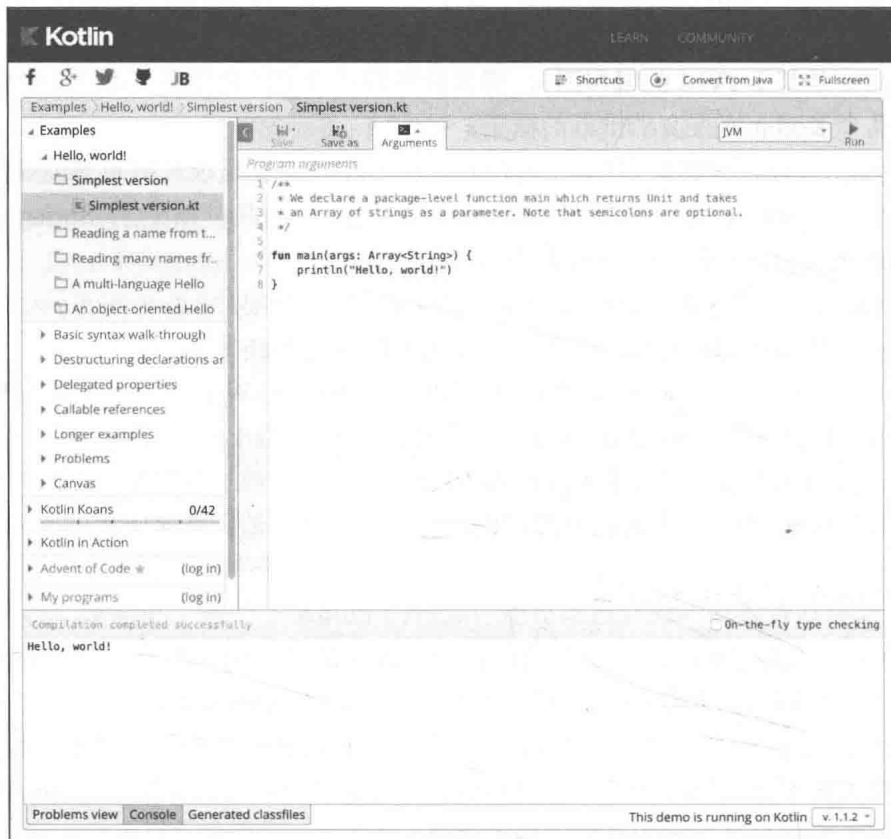
尽管 Kotlin 能做很多事 (Java 能做的, Kotlin 都能做), 但本书的主要关注点是开发 Android App, 因此, 在本书后续部分将着重介绍如何利用 Kotlin 开发 Android App。

12 在线运行 Kotlin 代码

可能很多读者使用过 Java 或其他编程语言, 学习这些技术的第一步, 就是配置, 一大堆的配置, 这让很多初学者摸不着头脑, 尤其是在没人指导的情况下更是如此。不过这一切在

Kotlin 这里就终结了，因为运行和测试 Kotlin 代码，根本就不需要进行配置，Kotlin 官方提供了一个在线运行和测试 Kotlin 代码的环境，可以运行 Kotlin 语句和标准库。

为了进入在线测试页面，可在浏览器地址栏中输入 <https://try.kotlinlang.org>，会进入如图 1-1 所示的页面。



▲图 1-1 Kotlin 的在线 IDE

在该页面左侧列出了一些例子，单击某个例子，会在右侧代码编辑区显示相应的代码，单击右上角的“Run”按钮，会执行 Kotlin 程序，并且在下面会显示执行结果。通过“Run”按钮前方的下拉列表框可以选择以 JVM 方式或 JavaScript 方式运行。在页面右下角，可以选择当前代码对应的 Kotlin 版本，本书写作时的 Kotlin 的最高版本是 1.1.2。

可能很多读者已经按耐不住自己激动的心情了，打算编写一段 Kotlin 代码试一试。不过，对于不熟悉 Kotlin 语法的人来说，可能连“hello world”这样简单的小程序都不知道如何编写（只能运行在线 IDE 提供的 Demo）。

为了让不熟悉 Kotlin 语法的读者尽快上手，这款在线 IDE 提供了 Convert from Java 功能，也就是可以将 Java 代码转换为 Kotlin 代码。现在单击 IDE 右上角的“Convert from Java”按钮，会显示一个转换窗口，左侧需要输入 Java 代码，单击该窗口右下角的“Convert to Kotlin”