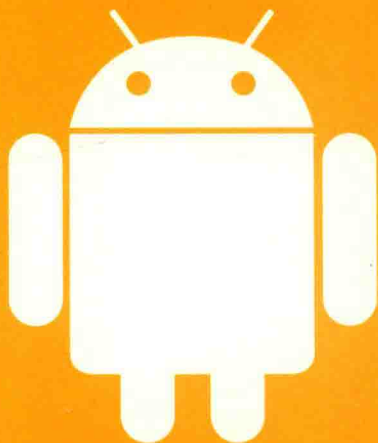




扫码看图书配套视频



爱上 Android

于连林 编著

本书前面部分知识点配有相应视频，帮助读者零基础学通 Android 开发

作者微信答疑，解决读者学习中的困难

循序渐进地讲述开发过程中必备的知识，宗旨是不重复开发一个“轮子”，在给大家讲明原理的同时，介绍一些好的开发框架，这些框架能够让读者很快上手开发完成一个应用。



中国工信出版集团



人民邮电出版社
POSTS & TELECOM PRESS

爱上 Android

于连林◎编著



人民邮电出版社
北京

图书在版编目 (CIP) 数据

爱上Android / 于连林编著. — 北京 : 人民邮电出版社, 2017.8
ISBN 978-7-115-46175-9

I. ①爱… II. ①于… 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 (2017) 第166769号

内 容 提 要

本书深入浅出, 详细讲解了 Android 开发的知识, 主要内容包括: Android 的体系架构、Android Studio、项目的创建、Android 工程目录、调试程序、界面的搭建; 常用的控件和属性、提示信息 Toast 和 Snackbar、点击事件 3 种写法等; Activity 之间的跳转、Activity 生命周期、Activity 启动模式等; 数据存储、网络编程; 图片的处理; 复杂控件的使用; Fragment、广播接收者、Service、动画等; 以及 Android 新特性、Kotlin 语言、性能优化、屏幕适配、自定义控件、JNI/NDK 开发等高级知识。并通过开发一个真实的项目让读者学以致用。

本书内容通俗易懂, 比较适合初学者阅读, 也可以作为专业人员的工具书, 以及大专院校相关专业师生的学习用书和培训学校的教材。

-
- ◆ 编 著 于连林
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>
北京市艺辉印刷有限公司印刷
 - ◆ 开本: 800×1000 1/16
印张: 26.5
字数: 696 千字
印数: 1—3 000 册
-

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

欢迎您阅读这本书

Android 已经在短短几年从无到有成为这个世界上最受欢迎的智能手机操作系统。无论您是公众开发应用程序，还是为自己开发 App，都会发现 Android 是一个令人兴奋和具有挑战性的平台。您会爱上 Android 开发！

最重要的是，我衷心希望您能找到它对您开发有用的地方。

IT 知识本身也许是枯燥的，本书尽量把它讲得幽默点。

本书结构

本书并不像其他的书那样，把所有的知识点堆到一起。这本书循序渐进地讲述开发过程中必备的知识，虽然可能忽略一些很少用的知识，但大家在实际开发过程中就会发现学的已经够用了。本书宗旨是不重复开发一个“轮子”，在给大家讲明白原理的同时会介绍一些高手开发的框架，这些框架能够让大家很快上手开发完成一个应用。

读者对象

本书内容通俗易懂，比较适合初学者阅读，也可以作为专业人员的工具书。学习本书之前您不需要任何的 Android 基础，但是需要有一定的 Java 基础（包括数组、运算符、面向对象思想、线程、IO 流等），因为大部分 Android 开发都是使用 Java 语言的，而本书很少会介绍 Java 方面的知识。

阅读本书时，您可以根据自身的情况来决定如何阅读，如果没有 Android 开发基础，建议不要跳过前面的章节。

因为考虑到新手初学时很难入门，所以本书刚开始的部分内容会附赠开发教学视频，帮助读者尽快入门学习。

微信公众号

为了方便大家看到此书勘误，本书提供一个网址：book.520wcf.com。

IT 行业技术日新月异，本书将尽快更新最新的知识点。为了方便大家及时收到本书更新提醒，建议大家关注微信公众账号。微信添加公众号搜索 likeDev，或者扫码加入：



为了方便交流，可以加 QQ 交流群：488929846，添加时请输入备注：爱上 Android。

目 录

第 1 章 初识 Android	1
1.1 Android 是什么	1
1.2 Android 体系架构	1
1.3 Android 发展史	4
第 2 章 选择您的开发工具	5
2.1 准备软件, 认识 Android Studio	5
2.1.1 什么是 Android Studio	5
2.1.2 为什么使用 Android Studio	6
2.2 安装 Android Studio	6
2.2.1 安装配置要求	6
2.2.2 下载地址	6
2.2.3 安装 JDK	7
2.2.4 安装 Android Studio	7
2.2.5 设置 JDK 和 Android SDK 目录	14
2.3 项目的创建	16
2.3.1 创建项目的步骤	16
2.3.2 解决错误 (没有错误最好)	19
2.4 Android Studio 界面预览	21
2.5 常用设置	22
2.5.1 设置主题	22
2.5.2 设置字体和格式	22
2.5.3 设置文件编码	24
2.5.4 设置快捷键	24
2.5.5 其他设置	26
2.6 常用快捷键	26
2.7 Android 工程目录	27
2.7.1 工程目录介绍	27
2.7.2 Gradle 使用详解	28

2.7.3 app/build.gradle	28
2.8 SDK 目录介绍	30
2.9 调试程序	31
2.9.1 创建模拟器	31
2.9.2 连接真实手机	33
2.9.3 Genymotion 模拟器	34
2.10 程序启动分析	34
2.11 日志和注释	36
2.11.1 注释	37
2.11.2 日志	37
2.11.3 设置 Android Studio 日志显示颜色	38
2.11.4 实际开发中控制日志	39
2.11.5 Logger 的使用	41
总结	42
第 3 章 界面的搭建	43
3.1 眼见皆 View	43
3.2 布局的搭建方式	43
3.3 常用的控件和属性	44
3.3.1 TextView	44
3.3.2 Button	45
3.3.3 EditText	46
3.3.4 ImageView	47
3.3.5 ProgressBar	49
3.4 布局的介绍	50
3.4.1 LinearLayout (线性布局)	50
3.4.2 RelativeLayout (相对布局)	52
3.4.3 FrameLayout (帧布局)	55
3.4.4 GridLayout (网格布局)	56

3.4.5	CoordinatorLayout	58	4.2.3	返回数据给之前的 Activity	93
3.4.6	ConstraintLayout (约束 布局)	61	4.3	Android 6.0 权限的管理	95
3.5	提示信息 Toast 和 Snackbar	61	4.3.1	Android 6.0 新的权限机制	96
3.5.1	Toast 使用详解	61	4.3.2	申请权限	96
3.5.2	修改 Toast 位置	63	4.3.3	第三方库 RxPermissions	100
3.5.3	自定义 Toast 布局	63	4.4	Activity 生命周期	101
3.5.4	避免内存泄露	65	4.4.1	生命周期的方法	101
3.5.5	Snackbar	65	4.4.2	Activity 销毁时保存数据	102
3.6	点击事件三种写法	68	4.4.3	锁定横竖屏	103
3.6.1	通过匿名内部类或内部类 实现	68	4.4.4	开发时注意事项	104
3.6.2	让类实现接口	69	4.5	Activity 任务栈	104
3.6.3	在布局文件中注册事件	70	4.6	Activity 启动模式	105
3.7	使用 Lambda 表达式代替匿名 内部类	70	4.6.1	standard	105
3.7.1	什么是 lambda 呢	70	4.6.2	singleTop	106
3.7.2	使用 Lambda 表达式	71	4.6.3	singleTask	106
3.8	AlertDialog 提示对话框	73	4.6.4	singleInstance	107
3.8.1	一般对话框	75	4.6.5	统一管理 Activity	108
3.8.2	Material Design 风格的 对话框	76	4.7	Toolbar 和 Navigation Drawer	109
3.8.3	列表对话框	78	4.7.1	AppBar 的简介	109
3.8.4	单选按钮对话框	79	4.7.2	创建菜单	111
3.8.5	多选按钮对话框	79	4.7.3	Toolbar	112
3.8.6	自定义 AlertDialog	80	4.7.4	Toolbar 遇上 Navigation Drawer	114
3.9	ProgressDialog	81	4.8	主题样式设置	118
总结		83	总结		120
第 4 章	Activity 介绍	84	第 5 章	数据存储	121
4.1	Activity 之间的跳转	84	5.1	SharedPreferences	121
4.1.1	显示意图	84	5.2	MD5 加密	123
4.1.2	隐式意图	86	5.3	文件存储数据	124
4.1.3	隐式意图的常见操作	88	5.3.1	保存到手机内存 (Internal Storage)	124
4.1.4	IntentFilter 匹配规则	89	5.3.2	SD 卡存储 (External Storage)	127
4.2	Activity 之间传递数据	92	5.4	SQLite 存储	132
4.2.1	通过 Intent 传递数据	92	5.4.1	创建数据库	132
4.2.2	静态工厂设计模式传递 数据	93	5.4.2	升级数据库	135
			5.4.3	数据库增删改查 (CURD)	137
			5.4.4	SQLite 数据库的事务操作	144

5.5 常见的数据库框架	146	7.2 大图的加载	199
总结	146	7.3 图片加水印	202
第 6 章 网络编程	148	7.4 图片特效, Matrix	205
6.1 HTTP 协议	148	7.4.1 缩放	205
6.1.1 URL 简介	148	7.4.2 倒影、镜面	207
6.1.2 HTTP 简介	149	7.4.3 旋转	208
6.1.3 GET 和 POST 对比	151	7.4.4 位移	208
6.2 HttpURLConnection	151	7.5 图片颜色处理——打造自己的美图秀秀	209
6.2.1 为什么废弃 HttpClient	151	7.5.1 颜色过滤器 ColorMatrixColorFilter	209
6.2.2 使用 HttpURLConnection 联网	152	7.5.2 实现图片美化功能	210
6.3 多线程编程	154	7.6 案例——随手涂鸦	214
6.3.1 线程的同步和异步	154	7.7 加载网络图片	217
6.3.2 AsyncTask	158	7.7.1 网络图片的缓存策略	217
6.3.3 RxJava	161	7.7.2 图片加载库 Picasso 的使用	218
6.4 网络请求实例	164	总结	219
6.5 JSON 解析	168	第 8 章 复杂控件的使用	220
6.5.1 使用 Android 原生方式解析 JSON	169	8.1 ListView	220
6.5.2 Gson 的使用	170	8.1.1 初识 ListView	221
6.5.3 插件 GsonFormat 快速实现 JavaBean	174	8.1.2 定制 ListView 条目的界面	222
6.5.4 完成请求实例	176	8.1.3 优化 ListView	225
6.6 网络请求框架——Retrofit	179	8.1.4 ListView 的点击事件	226
6.6.1 使用 Retrofit	179	8.1.5 ListView 常用的属性	228
6.6.2 常用的注解	181	8.2 GridView	229
6.6.3 完成请求案例	181	8.3 RecyclerView	231
6.6.4 RxJava 和 Retrofit 结合	183	8.3.1 初识 RecyclerView	231
6.7 WebView	184	8.3.2 使用 RecyclerView	232
6.7.1 WebView 配置	186	8.3.3 不同的布局排列方式	234
6.7.2 WebViewClient 方法	187	8.3.4 RecyclerView 添加点击 事件	236
6.7.3 设置 WebChromeClient	189	8.3.5 RecyclerView 添加删除 数据	237
6.7.4 WebView 常用的方法	190	8.3.6 下拉刷新 SwipeRefreshLayout	237
6.7.5 WebView 模板代码	191	8.4 CardView	239
总结	196	8.5 ViewPager	241
第 7 章 图片的处理	197	8.6 BottomNavigationView (底部	
7.1 Bitmap 和 Drawable	197		

导航)	243	12.1 补间动画 (Tween Animation)	301
8.7 TabLayout	246	12.2.1 AlphaAnimation (透明度动画)	302
8.7.1 TabLayout 使用	246	12.2.2 ScaleAnimation (缩放动画)	304
8.7.2 TabLayout 自定义条目样式	249	12.2.3 TranslateAnimation (平移动画)	305
总结	250	12.2.4 RotateAnimation (旋转动画)	306
第9章 探索 Fragment	251	12.2 逐帧动画 (Frame Animation)	308
9.1 使用 Fragment	252	12.3 属性动画	310
9.1.1 Fragment 的生命周期	252	总结	313
9.1.2 创建 Fragment	252	第13章 新特性	314
9.1.3 向 Activity 添加 Fragment	254	13.1 Android 7.0 分屏开发	314
9.1.4 管理片段	256	13.1.1 如何分屏呢	315
9.1.5 Fragment 的向下兼容	257	13.1.2 多窗口生命周期	315
9.2 FragmentTabHost 实现底部标签	257	13.1.3 针对多窗口进行配置	316
9.3 ViewPager 和 Fragment 结合	259	13.1.4 多窗口模式中运行应用 注意事项	316
总结	273	13.1.5 在多窗口模式中启动新 Activity	317
第10章 广播接收者	274	13.1.6 支持拖放	318
10.1 广播简介	274	13.2 Android 7.0 快速设定	318
10.2 实现一个 BroadcastReceiver	275	13.3 约束布局 ConstraintLayout	321
10.3 发送自定义广播	279	13.3.1 ConstraintLayout 简介	321
10.4 桌面快捷方式	283	13.3.2 添加约束布局	322
总结	285	13.3.3 使用约束布局	322
第11章 Service 介绍	286	13.3.4 添加约束	323
11.1 服务的基本用法	286	13.3.5 使用自动连接和约束推断	325
11.1.1 创建服务	286	13.3.6 快速对齐 Align	326
11.1.2 启动和停止服务	287	13.4 使用 Kotlin 语言开发 Android	326
11.1.3 绑定服务	289	总结	330
11.1.4 服务的生命周期	292	第14章 性能优化	331
11.2 IntentService	292	14.1 性能检测	331
11.3 Service 和 BroadCastReceiver 结合 使用的案例 (兼容 Android 7.0)	294	14.1.1 检测内存泄露	331
11.3.1 Android 7.0 错误原因	297	14.1.2 LeakCanary	334
11.3.2 使用 FileProvider	298	14.1.3 追踪内存分配	335
总结	300		
第12章 动画	301		

14.1.4 查询方法执行的时间	335	15.4.2 百分比布局	360
14.2 过度绘制 (OverDraw)	336	15.4.3 约束布局	362
14.2.1 过度绘制概念	336	总结	362
14.2.2 追踪过度绘制	336	第 16 章 自定义控件	363
14.2.3 去掉不合理背景	337	16.1 自定义控件简介	363
14.2.4 不合理的 XML 布局对 绘制的影响	338	16.2 View 的生命周期	364
14.3 避免 ANR	338	16.2.1 构造函数	365
14.3.1 ANR 分析	339	16.2.2 onAttachedToWindow	369
14.3.2 ANR 解决方式	341	16.2.3 onMeasure	370
总结	341	16.2.4 onLayout	372
第 15 章 屏幕适配	342	16.2.5 onDraw	372
15.1 Android 屏幕适配出现的原因	342	16.2.6 View 更新	373
15.2 相关重要概念	344	16.2.7 动画	374
15.2.1 屏幕尺寸	344	总结	374
15.2.2 屏幕分辨率	344	第 17 章 JNI/NDK 开发	375
15.2.3 屏幕像素密度	344	17.1 NDK 配置 (最新的 CMake 方式)	375
15.2.4 屏幕尺寸、分辨率、像素 密度三者关系	344	17.1.1 下载	376
15.2.5 dip	345	17.1.2 创建项目	376
15.2.6 sp	345	17.1.3 运行项目	378
15.3 尺寸适配解决方案	346	17.1.4 手动添加 native 方法	379
15.3.1 “布局”适配	346	总结	380
15.3.2 尺寸 (size) 限定符	347	第 18 章 开发一个真实的项目	381
15.3.3 最小宽度 (Smallest-width) 限定符	348	18.1 项目需求分析	381
15.3.4 使用布局别名	349	18.2 创建项目	382
15.3.5 屏幕方向 (Orientation) 限定符	350	18.3 界面实现	383
15.3.6 “布局组件”匹配	352	18.3.1 启动界面	383
15.3.7 Layout_weight 详解	352	18.3.2 引导页面	385
15.3.8 “图片资源”匹配	355	18.3.3 主界面	388
15.3.9 .9 的制作	355	18.3.4 列表界面	390
15.3.10 “用户界面流程”匹配	357	18.4 请求网络	395
15.4 屏幕密度适配	359	18.5 新闻列表和详情	400
15.4.1 “布局控件”适配	359	18.6 完成整个项目	409
		总结	412

第1章 初识 Android

毫无疑问，你肯定急于开始学习 Android 应用程序开发。毕竟，编程对程序员来说吸引力也是很大的。然而，在开始实际编程前，需要先了解 Android 的系统。

1.1 Android 是什么

Android，中文名为安卓，直接翻译是机器人的意思，它是 Google 公司推出的一款开源免费的智能操作系统，不仅限于手机，现在很多终端都在使用 Android 操作系统，如手表、电视、汽车、平板电脑、微波炉等，如图 1-1 和图 1-2 所示。一般而言，还是以手机开发为主，本书也是介绍如何进行手机软件开发。



▲图 1-1 Android 系统的微波炉



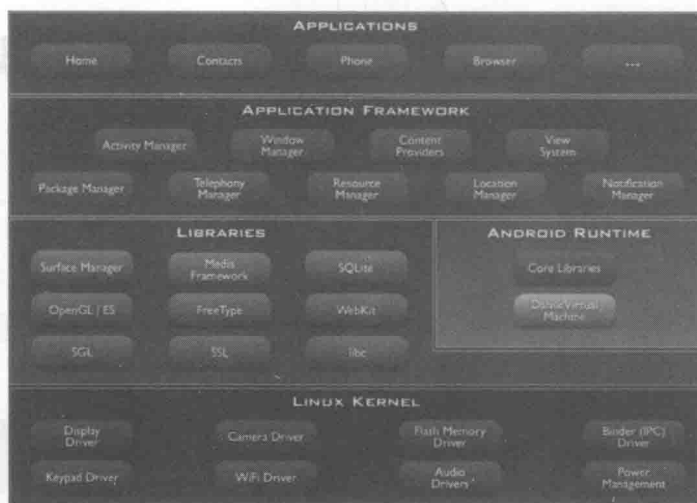
▲图 1-2 Android 智能电视

由于 Android 系统是免费的，所以推广得很快。目前 Android 手机市场占有率已经超过 80%。

1.2 Android 体系架构

为了更好地理解 Android 系统是如何工作的，接下来解剖一下 Android 系统，看看其内部是

如何构建的, 如图 1-3 所示。



▲图 1-3 Android 系统架构

Android 大致可以分为 4 层架构、5 块区域。图 1-3 从下到上依次为:

- (1) Linux 内核层;
- (2) 系统运行库层;
- (3) 应用框架层;
- (4) 应用层。

1. Linux 内核层

如图 1-4 所示, Android 系统是基于 Linux 2.6 内核的, 这一层为 Android 的各种硬件提供了驱动程序, 如显示驱动、照相机驱动、声音驱动、电池管理等。当手机开机的时候, 这一层肯定先运行。



▲图 1-4 Linux 内核层

2. 系统运行库层

如图 1-5 所示, 这一层通过一些 C 或者 C++库为 Android 系统提供了主要的特性支持。如 Webkit 提供了浏览器支持 (保证手机能够正常使用浏览器), SQLite 库提供了数据库的支持 (可以用来存储一些数据), OpenGL ES 库提供了 3D 绘图的支持等。

这一层还有 Android 运行时库, 它主要提供了一些核心库, 能够允许开发者使用 Java 语言编写 Android 应用。另外, Android 运行时库中还包含了 Dalvik 虚拟机 (Android 5.0 系统以后替换

成了 ART 虚拟机), 因为 Java 语言的特性决定它只能通过虚拟机去运行, 这个有点像我们在电脑上玩小霸王的游戏需要装模拟器一样。



▲图 1-5 系统运行库层

无论是 Dalvik 虚拟机还是 ART 虚拟机, 都是专门为移动设备定制的, 它针对手机内存、CPU 性能有限等情况, ART 虚拟机安装程序时间稍微长一点, 但是运行程序快一点。Google 工程师最终决定用 ART 虚拟机 (这点毋庸置疑, 没人愿意天天装程序)。

不管是 Dalvik 虚拟机还是 ART 虚拟机, 都要比 Java 语言官方本身的 JVM 虚拟机要好用。当然最主要的原因是版权问题, 虽然 Java 语言本身是开源免费的, 但是 JVM 虚拟机不是开源免费的, 所以不能直接用 JVM 虚拟机。

3. 应用框架层

如图 1-6 所示, 这一层主要提供了构建应用程序时可能用到的各种 API, Android 自带的一些核心应用就是使用这些 API 完成的, 开发者也可以通过这些 API 构建自己的应用程序。这层的使用频率要比上面介绍的两层使用频率高。



▲图 1-6 应用框架层

4. 应用层

如图 1-7 所示, 所有安装到手机上的应用都属于这一层, 例如系统自带的联系人、短信等程序, 或者是自己下载的一些应用、游戏, 肯定也会包括自己写的程序。



▲图 1-7 应用层

应用层和应用框架层大部分都是采用 Java 代码编写的, Linux 内核层和系统运行库层大部分采用 C 或者 C++ 编写。

计算机语言核心就是 0 和 1, 理论上用一个电闸都能编程, 闭合就是 1, 打开就是 0, 一开一闭程序就运行了, 估计干这行的肯定瞧不上做 C 开发的。编程语言不分好坏, 只是功能划分不一样。千万不要和资深程序员争论哪门编程语言好, 他们眼中只有 0 或者 1, 基本上他能说的让你

高山仰止。

1.3 Android 发展史

Android 从 2008 年发布 1.0 至今已经发布了 20 多个版本了，目前最新版本是 Android 8，每一个系统版本都对应一个开发的 API 版本号，如 Android 5.1 对应 API 版本号 21。每一个 Android 版本还都有一个代号，包括甜甜圈、姜饼、三明治、果冻豆、棒棒糖等。

如图 1-8 所示，表中最右面一栏就是当前版本的市场份额，可以发现 Android 4.4 是当前最流行的版本，Android 5.0 以上的会越来越多，而 Android 4.1 以下的手机基本上没有了，Android 7.0 还不到百分之一。所以本书重点讲解 Android 5.0~7.1 的知识，兼容到 Android 4.1。

Version	Codename	API	Distribution
2.2	Froyo	8	0.1%
2.3.3 - 2.3.7	Gingerbread	10	1.7%
4.0.3 - 4.0.4	Ice Cream Sandwich	15	1.6%
4.1.x	Jelly Bean	16	6.0%
4.2.x		17	8.3%
4.3		18	2.4%
4.4	KitKat	19	29.2%
5.0	Lollipop	21	14.1%
5.1		22	21.4%
6.0	Marshmallow	23	15.2%

数据来源 Android 官网

▲图 1-8 Android 各个版本市场占有率

好了，Android 基本信息就介绍到这，接下进入真正的 Android 开发之旅。

第2章 选择您的开发工具

工欲善其事，必先利其器。选择一个好的 IDE 可以大幅提高开发效率，节省下的时间可以去多学习新知识，多陪陪家人。接下来就手把手领着大家把开发环境搭建起来。

2.1 准备软件，认识 Android Studio

之前开发 Android 一般用 Eclipse+ADT 插件，但是 Google 已经停止维护 ADT 了，本书郑重地建议大家，不要用 Eclipse 开发了，改用 Android Studio 开发。

当然 Android 程序大部分是使用 Java 语言开发的，所以安装 Java 环境是必须的，需要大家下载 JDK，目前最新版本是 JDK 8。需要注意的是，大家必须准备 JDK 7 以上的版本（如果使用的是 Android Studio 2.1 以上的版本，必须准备 JDK 8 以上），否则达不到 Android Studio 的安装要求，建议准备 JDK 8（可以使用 lambda 语法）。如果是 64 位操作系统，建议安装 64 位的 JDK；如果是 32 位操作系统，只能安装 32 位的 JDK。

在 Windows 操作系统中右键单击我的电脑→属性就可以查看当前系统的位数。

JDK 下载方式非常简单，直接用百度搜索 JDK 的下载网站，下载下来直接打开，一直点击下一步按钮就可以安装成功。

上面都不是重点，大家如果有 Java 基础，这些应该都会了；如果没有 Java 基础，最好先了解一下 Java，接下来就进入重点，了解一下 Android Studio。

2.1.1 什么是 Android Studio

Android Studio 是一个基于 IntelliJIDEA 的新的 Android 开发环境。与 Eclipse ADT 插件相似，Android Studio 提供了集成的 Android 开发工具，用于开发和调试。

Android Studio 于 2013 年 5 月 16 日在谷歌 I/O 大会正式对外发布，Google 希望 Android Studio 能让应用开发更简单、高效。

Windows、Mac OS X、Linux 三大平台全部支持。

2015 年年底，Google 开发者大会期间推出了 Android Studio 2.0，让 Android Studio 编译和运行速度提高了 50 倍。

Android Studio 2.2 完美兼容了 NDK 开发，可以完全摆脱 ADT。

2.1.2 为什么使用 Android Studio

原因一：Android Studio 是谷歌开发的，专门为 Android 开发量身定做的编辑器。

原因二：Android Studio 最核心的功能就是智能代码编辑器，它能帮助我们非常高效地完成代码补全、重构和代码分析。做过开发的读者都知道，没有代码提示，估计 90% 以上的人就不会写代码了。

原因三：Android Studio 的速度更快、更加智能，集成了版本控制系统、代码分析工具、UI 编辑器、Gradle 构建工具、Android Monitor、模拟器、各种模板和示例等，还有各种强大的插件支持。

原因四：谷歌宣布将在 2015 年年底前停止对 Eclipse Android 开发工具的一切支持，包括 ADT 插件、Ant 构建系统、DDMS、Traceview 与其他性能和监控工具。

有了上面这四大原因，你还有什么理由不用 Android Studio？

2.2 安装 Android Studio

2.2.1 安装配置要求

现在的电脑配置越来越高，图 2-1 所列的配置要求基本上很容易就能达到了。如果你的电脑配置不满足这些要求，就需换台电脑了。

平台	Mac OS X	Windows	Linux
操作系统版本	Mac OS X® 10.8.5 或更高版本	Windows® 8/7/Vista/2003 (32 位或 64 位) 或更高版本	GNOME、KDE、Ubuntu 等 推荐 Ubuntu
内存大小	最低：2GB RAM，推荐：4GB RAM		
硬盘空间	400 MB 硬盘空间		
预留空间	Android SDK、模拟器系统映像及缓存至少需要 1GB 空间		
JDK 版本	Java 开发工具包 (JDK) 7 或更高版本		
屏幕分辨率	最低屏幕分辨率：1280 x 800		

▲图 2-1 需要满足的基本系统要求

2.2.2 下载地址

官方下载地址：

官网：<http://developer.android.com/sdk/index.html>。

中文官网：<https://developer.android.google.cn/studio/index.html>。

官网打开样式如图 2-2 所示，这里提供了三大平台的安装包，选择对应的平台进行下载。



▲图 2-2 Android Studio

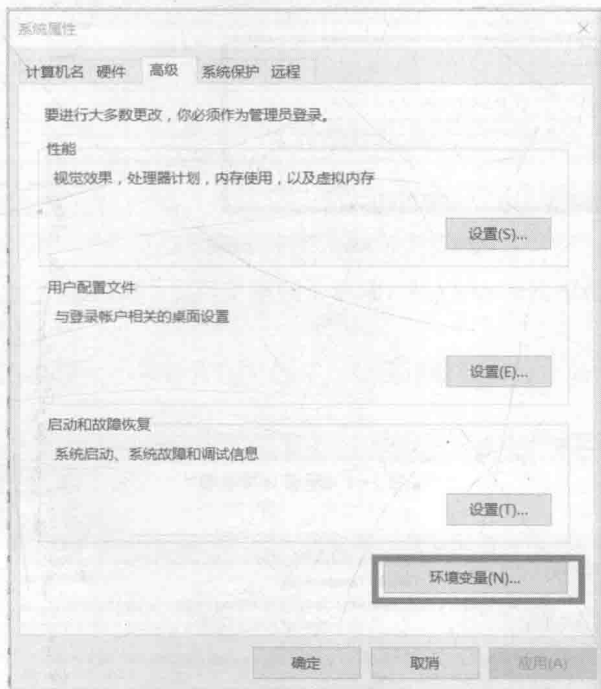
采用官方链接，可以直接下载全部安装包，里面主要包含 Android Studio 安装包和 Android 开发 SDK（开发工具包）。当然也可以分开下载。

2.2.3 安装 JDK

首先需要下载 JDK，安装完成 JDK，配置 JDK 环境变量。下面以 Windows 平台为例进行演示。

右键单击我的电脑→属性→环境变量，配置 JAVA_HOME 环境变量。

参考如图 2-3 和图 2-4 所示的内容。



▲图 2-3 配置环境变量

2.2.4 安装 Android Studio

安装过程中的简单操作在这就不进行截图讲解了，因为只需要点击 Next 按钮。

本文使用“包含 SDK”的安装文件进行讲解，其中包含了“不包含 SDK（软件开发工具包）的安装文件”的安装步骤。如果使用“不包含 SDK 的安装文件”进行安装，安装步骤只会比这些步骤少。

如果使用压缩包安装，直接解压缩就可以用了。本节内容可直接跳过。

步骤 1

如图 2-5 所示，第一个选项：Android Studio 程序，必选。第二个选项：Android SDK（安卓开发工具包），如果电脑中已经存在 Android SDK，可以不勾选。第三个选项和第四个选项都和虚