

# Android应用程序设计

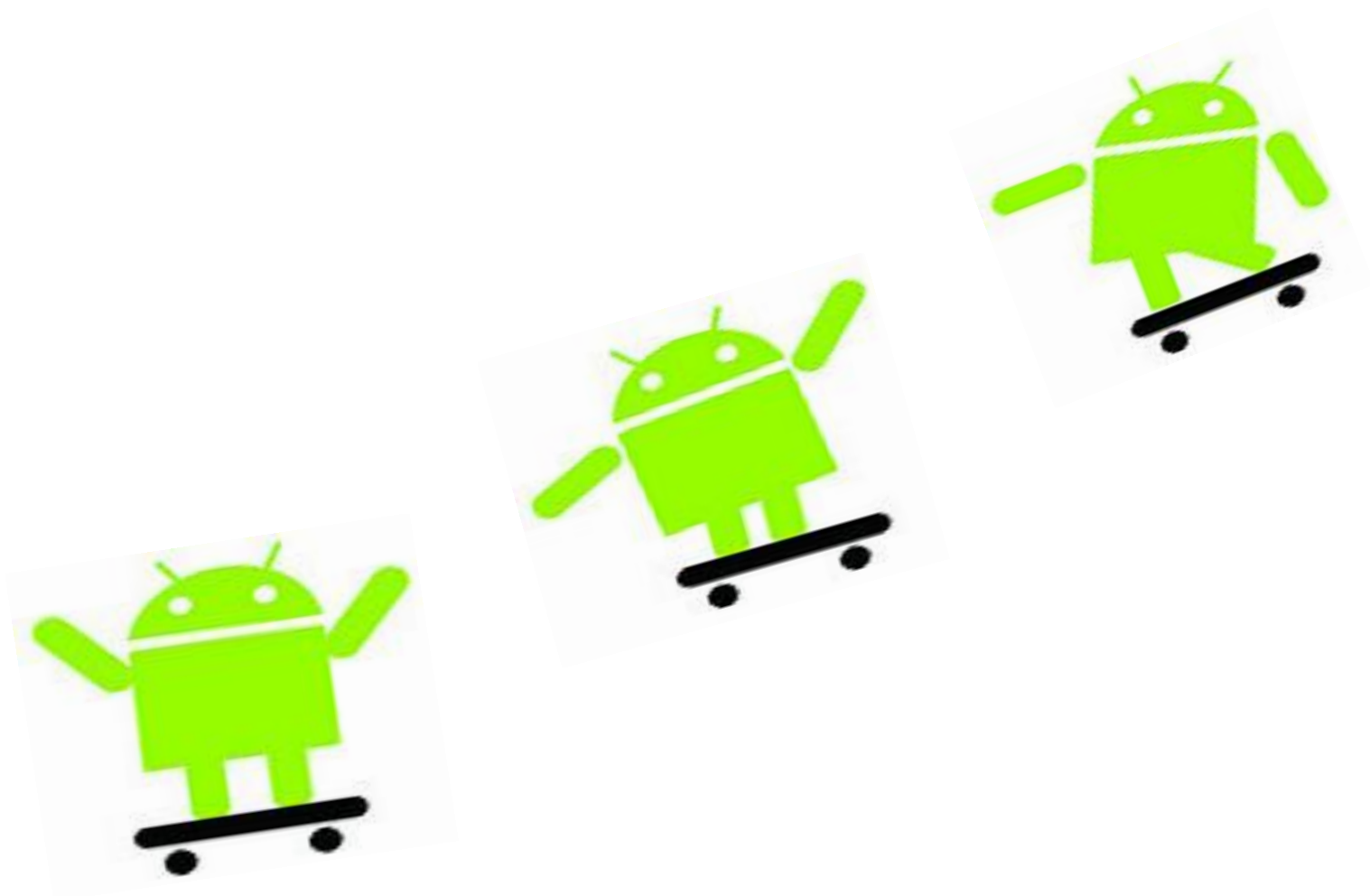
## 第一章 *Android*系统及其开发过程

石刚

2025年2月

## 1.1 Android系统概述

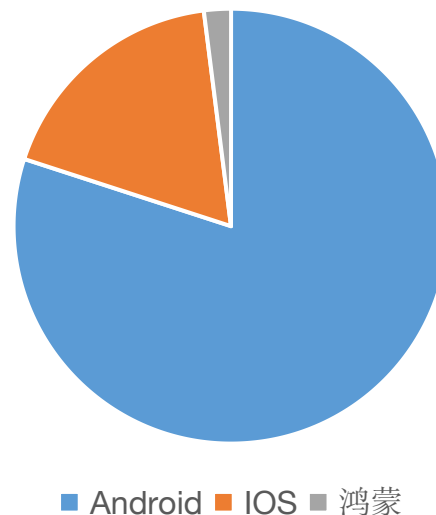
# 从如日中天到百花齐放——Android系统



# 智能移动终端操作系统格局



全球占比



- 鸿蒙OS 5.0开始不再兼容安卓，未来iOS、鸿蒙、安卓各自独立系统，形成手机操作系统三分天下的局面。
- HarmonyOS的国内智能手机操作系统市占率已由2023年第一季度的8%上涨至2024年第一季度的17%，iOS同期的份额却从20%下降至16%，华为HarmonyOS正式在中国市场首次超越苹果iOS，成长为中国第二大操作系统。全球智能手机操作系统主要由Android和iOS主导，2024年第一季度的市占率分别为77%、19%，均有所下降，排名第三的HarmonyOS的市占率上升至4%。

# 全球安卓电视风潮



# 全球安卓电视风潮



安卓电视机的电路主板

# 无处不在的Android嵌入式产品

小米手环2

触摸操作，精准计步





# 智能移动设备下一个战场



[https://zhuanlan.zhihu.com/p/602878363?utm\\_id=0](https://zhuanlan.zhihu.com/p/602878363?utm_id=0)



## 1.1.2 android系统的系统架构

- Android系统分为四个层，从高层到低层分别是应用程序层、应用程序框架层、系统运行库层和linux核心层。



# 1. 应用程序

Android系统自带了一套核心应用程序，应用程序包括电话拨号程序、短信程序、日历、音乐播放器、浏览器、联系人管理程序等，如图所示。



## 2. 应用程序框架

Android 系统的应用程序框架主要包含了 9 个部分：

活动页面管理 (Activity Manager)：用于管理程序的生命周期。

窗口管理 (Window Manager)：用于管理应用程序窗口。

内容供应 (Content Providers)：提供数据共享，使一个应用程序可以访问另一个应用程序的数据。

视图系统 (View System)：用于构建应用程序的可视化组件。

包管理 (Package Manager)：用于管理项目程序。

电话管理 (Telephone Manager)：移动设备的基本功能统一由电话管理器管理。

资源管理 (Resource Manager)：为应用程序提供所需的文字、声音、图片、视频或布局文件等资源。

位置管理 (Location manager)：用于提供位置服务。

通知管理 (Notification Manager)：在手机顶部状态栏发布消息提示。

### 3. 系统运行库

#### (1) 程序库

Android 包含一些 C/C++ 程序库，这些库能被 Android 系统中不同的组件使用。它们通过 Android 应用程序框架为开发者提供服务。

#### (2) Android 运行时库

Android 包括了一个核心库，该核心库提供了 JAVA 编程语言核心库的大多数功能。Android 系统的 Dalvik 虚拟机也包含在 Android 运行时库中。

### 4. Linux 内核

Android 的核心系统服务依赖于 Linux 内核，其安全性、内存管理、进程管理、网络协议栈和驱动模型等基本依赖于 Linux。

## 1.1.3 Android开发分类

对于开发者而言，Android 开发分为两大类：

### 1. 系统移植开发

移植开发是为了将 Android 系统能在手持式移动设备上运行，在具体的硬件系统上构建 Android 软件系统。这种类型的开发在 Android 底层进行，需要移植开发 Linux 中相关的设备驱动程序及 Android 本地框架中的硬件抽象层，也就是需要将设备驱动与 Android 系统联系起来。Android 系统对硬件抽象层都有标准的接口定义，移植时，只需实现这些接口即可。

### 2. Android 应用程序开发

应用程序开发可以基于硬件设备(用于测试的实体手机)，也可以基于 Android 模拟器。应用开发处于 Android 系统的顶层，使用 Android 系统提供的 Java 框架(API)进行开发设计工作，是大多数开发者从事的开发工作。本书所介绍的 Android 应用程序设计，都是在这个层次上进行的。



## 1.2 安装Android SDK开发环境

## 1.2.1 安装Android SDK前必要的准备

- 1、Android系统开发的操作平台与软件环境要求

一种是:

Eclipse + ADT(Android Development Tools插件)。因此, 需要安装Java SDK1.5以上和Eclipse 3.3以上版本的环境。(现在不用了)

- 2、另一种是Android Studio系统

到Android官方网站:

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

(Electric Eel 22年1月版, 最新的hedgehog 23年1月版)

下载3.5.2版的系统软件可以从这下载

<http://tools.android-studio.org/>

<https://developer.android.google.cn/studio/releases?hl=zh-cn>

Google开发者中国

[Developers.google.cn](https://developers.google.cn)

## 1.2.2 安装Android SDK详解

- 1、运行Android系统的安装文件

android-studio-ide-171.4443003-windows.exe

win64位版本

android-studio-ide-171.4443003-windows32.zip

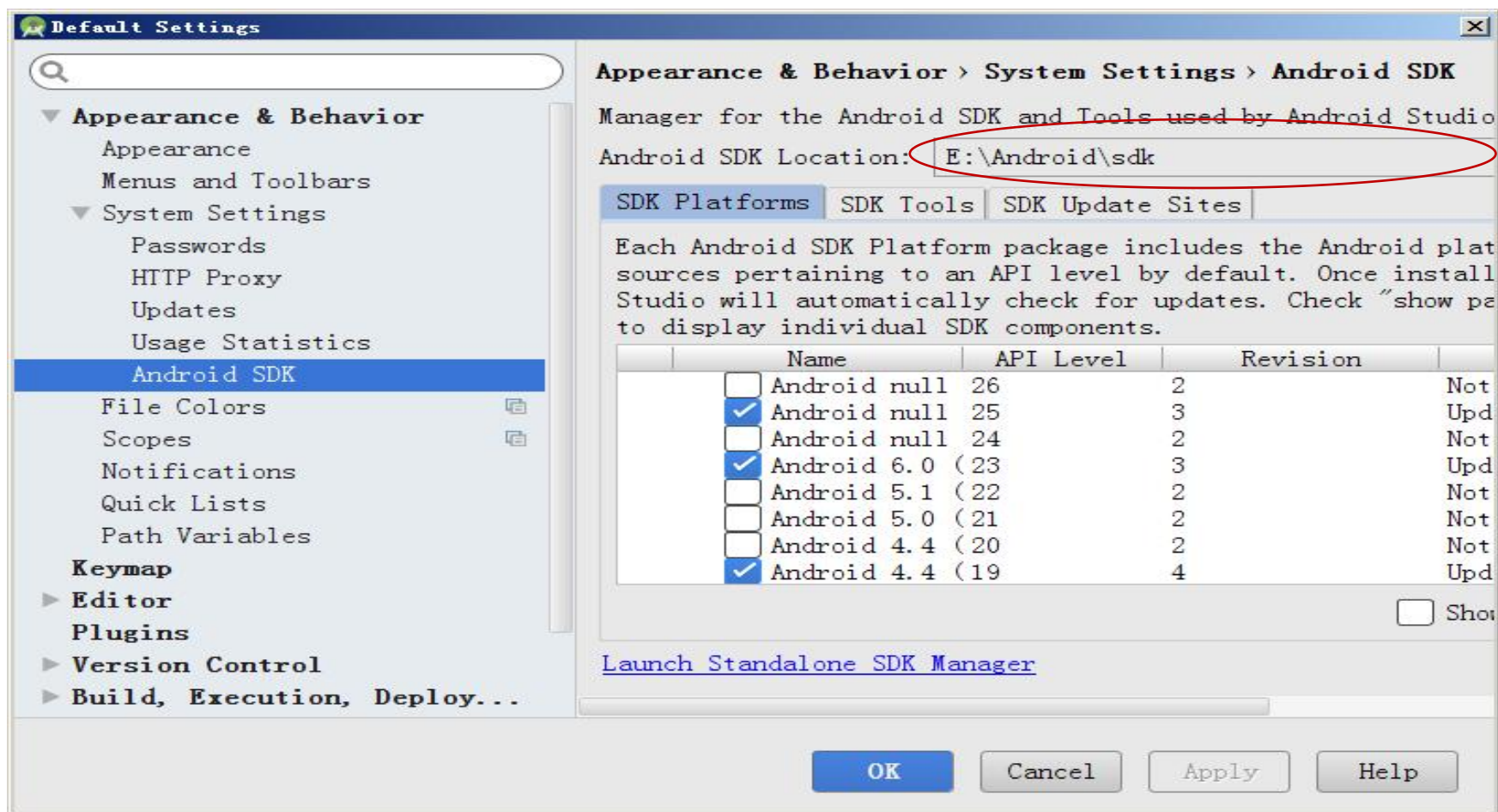
Win32位版本

按安装向导完成  
安装



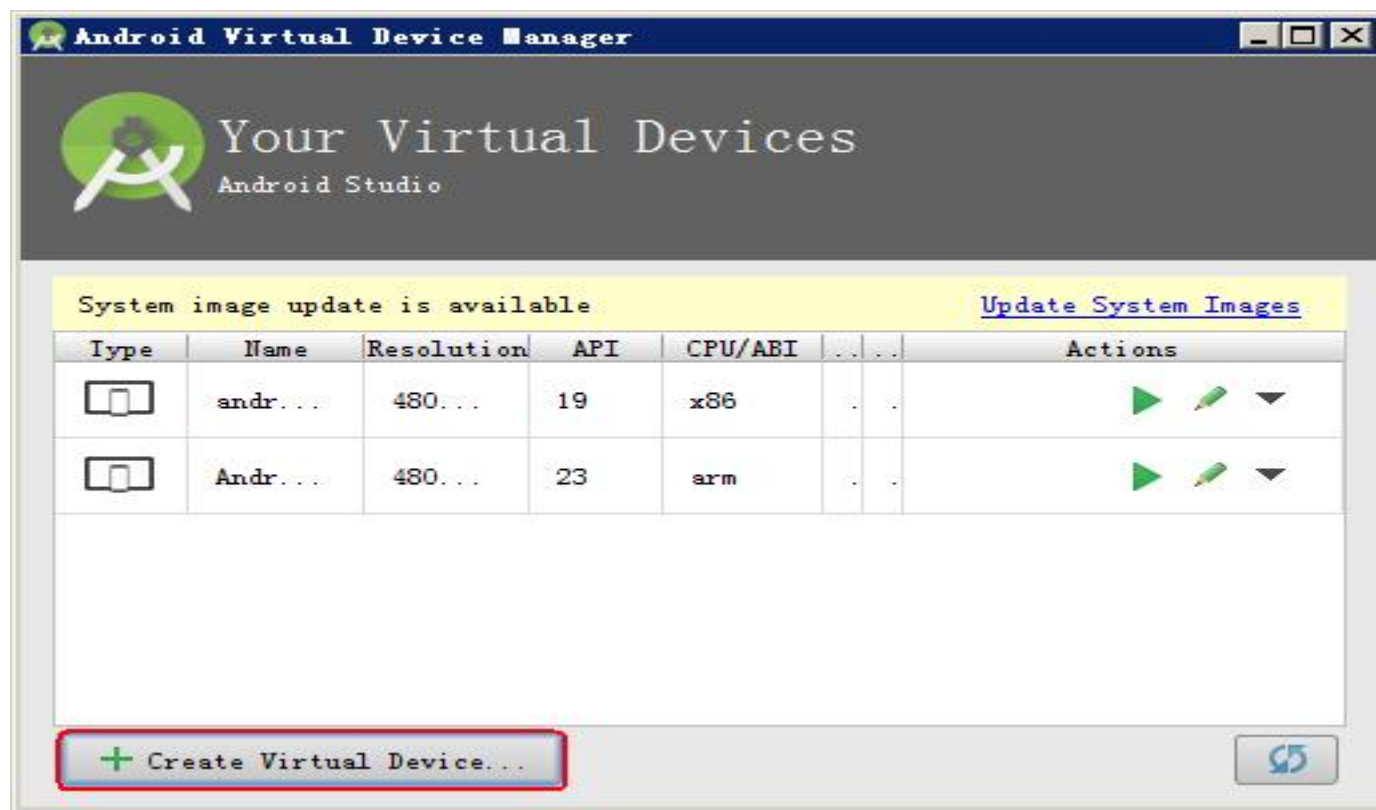
## 2. 设置Android SDK存放位置

安装完成后，第一次运行android studio系统，需要设置Android SDK，找到SDK的存放位置，



### 3. 创建Andriod虚拟设备AVD

选择Android Studio菜单“Tools”->“Android”->“AVD Manager”，在弹出的“Android Virtual Device Manager”对话框中可以看见已创建的AVD。单击下方“Create Virtual Device...”按钮创建一个新的AVD。







(2) 运行AVD模拟器。在如图1.6所示的“Android Virtual Device Manager”对话框中，选择已经建立的AVD，单击“Actions”栏中的 按钮，可以启动AVD模拟器。

## 1.2.3设置环境变量

需要增添设置 “Path”变量的值:

C:\Users\Administrator\AppData\Local\Android\sdk\tools;

及C:\Users\Administrator\AppData\Local\Android\sdk\ platform-tools;

如图1.8所示。



## 1.3 Android API和在线帮助

# 1.3 Android API和在线帮助文档

- 1、Android API

Android提供了标准类库Android API。

Android.app: 封装了高层的程序模型、提供基本的运行环境。

Android.content: 封装了各种对设备上的数据进行访问和发布的类

Android.database: 通过内容提供者浏览和操作数据库

Android.graphics: 底层的图形库，包含画布、颜色过滤、点、矩形，可以将它们直接绘制到屏幕上。

Android.location: 封装了定位和相关服务的类

Android.media: 封装了一些类管理多种音频、视频的媒体接口。

Android.net: 封装了帮助网络访问的类，超过通常的 java.net.\*接口。

Android.os: 封装了系统服务、消息传输、IPC 机制。

Android.opengl: 封装了 opengl 的工具、3D加速。

Android.provider: 封装了类访问 Android 的内容提供者

Android.telephony: 封装了与拨打电话相关的 API 交互。

Android.view: 封装了基础的用户界面接口框架。

Android.util: 涉及工具性的方法，例如时间日期的操作。

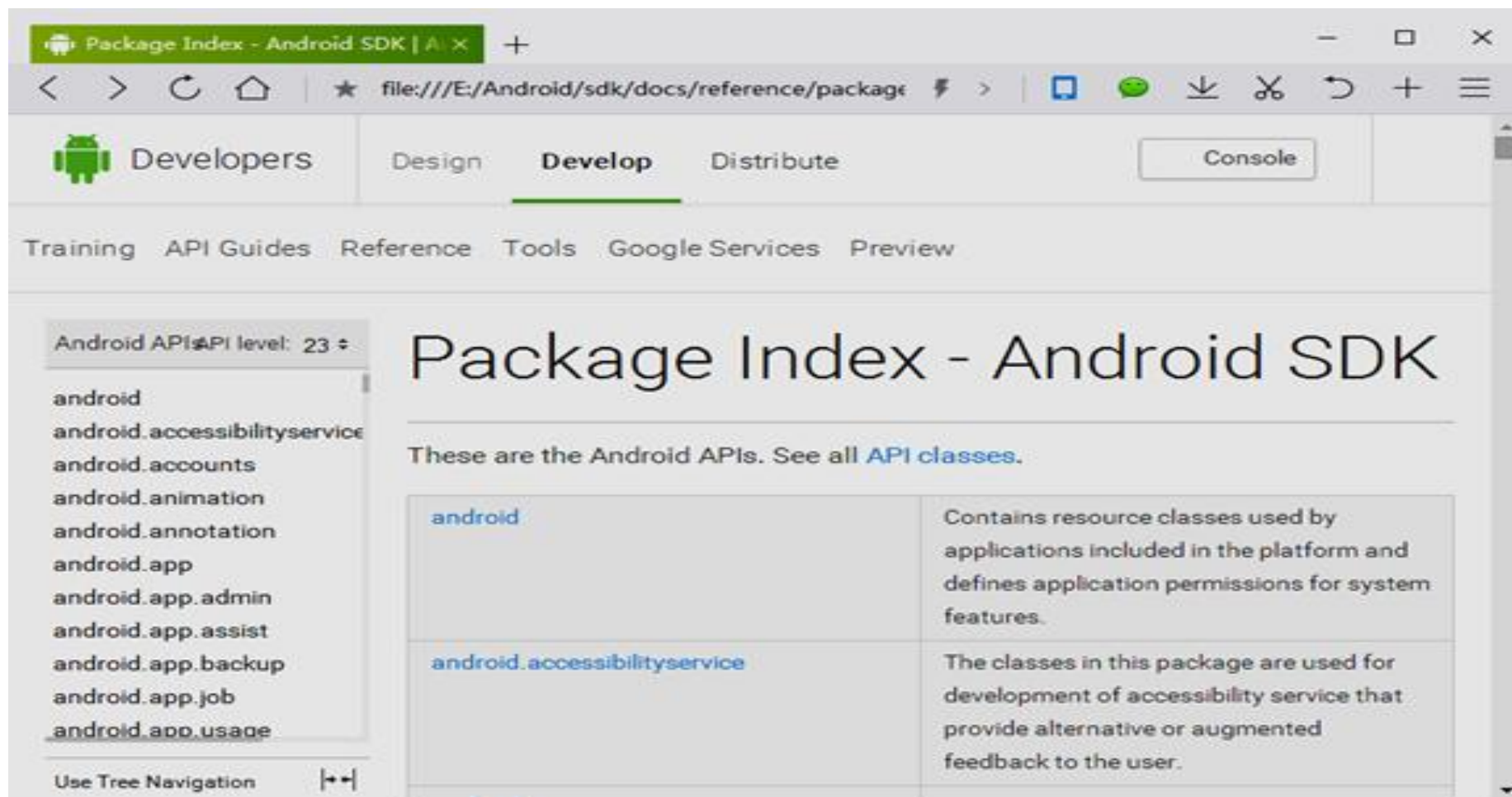
Android.webkit: 默认浏览器操作接口。

Android.widget: 封装了各种 UI 元素（大部分是可见的）在应用程序的屏幕中使用。



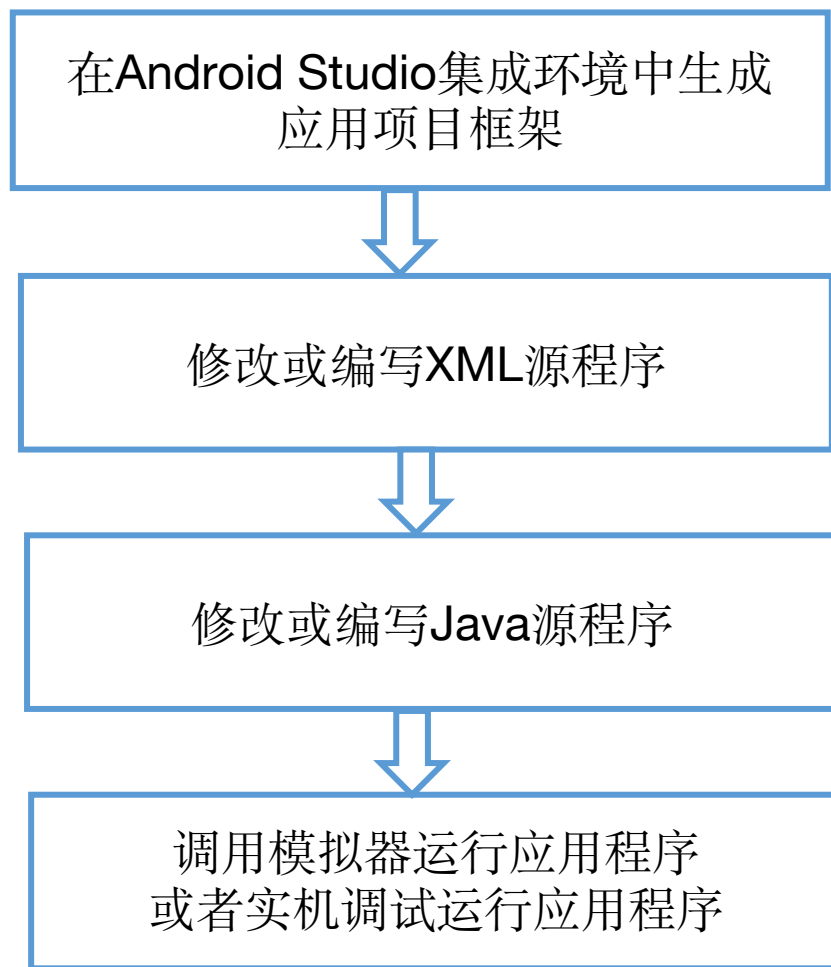
## 2. Android API帮助文档

运行Android Studio的安装目录下的docs\index.html文件



## 1.4 Android应用项目的开发过程

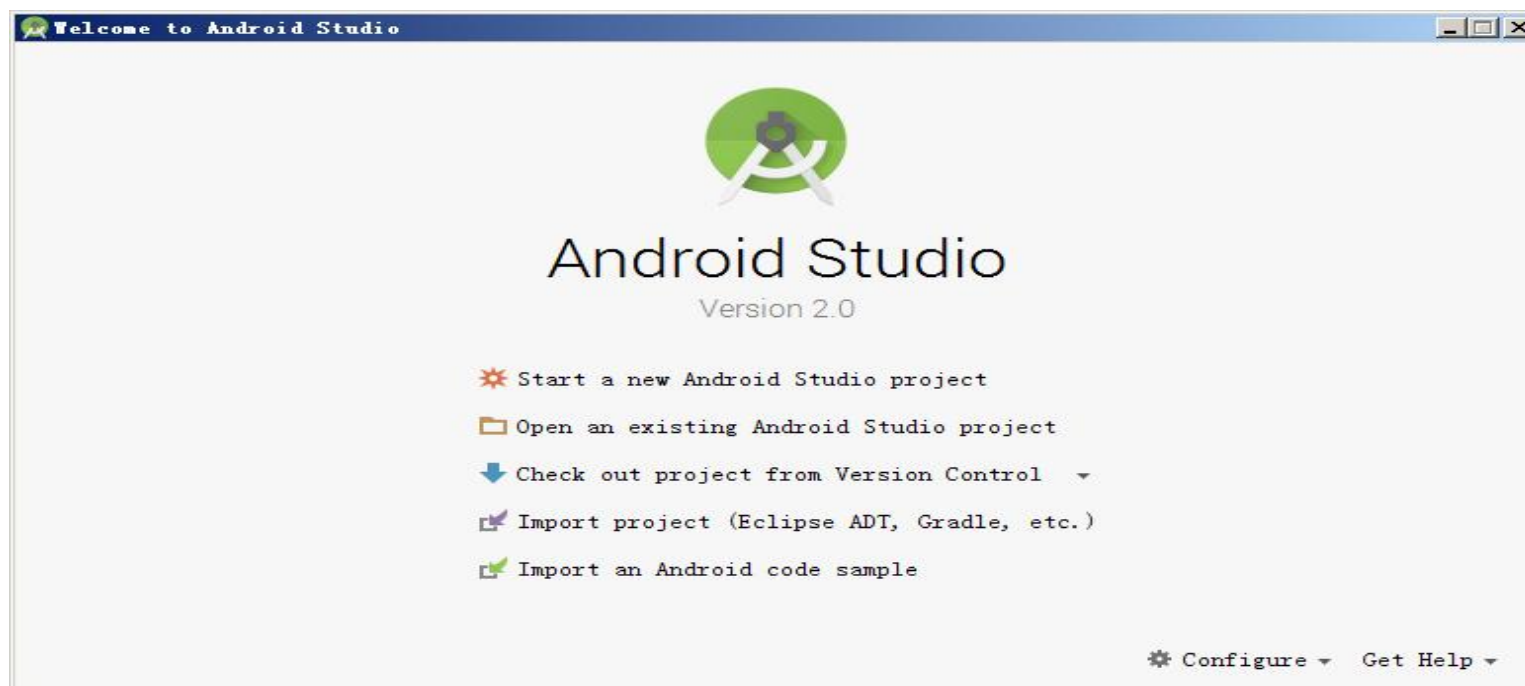
## 1.4.1 开发Android应用程序的一般过程



## 1.4.2 生成Android应用程序框架

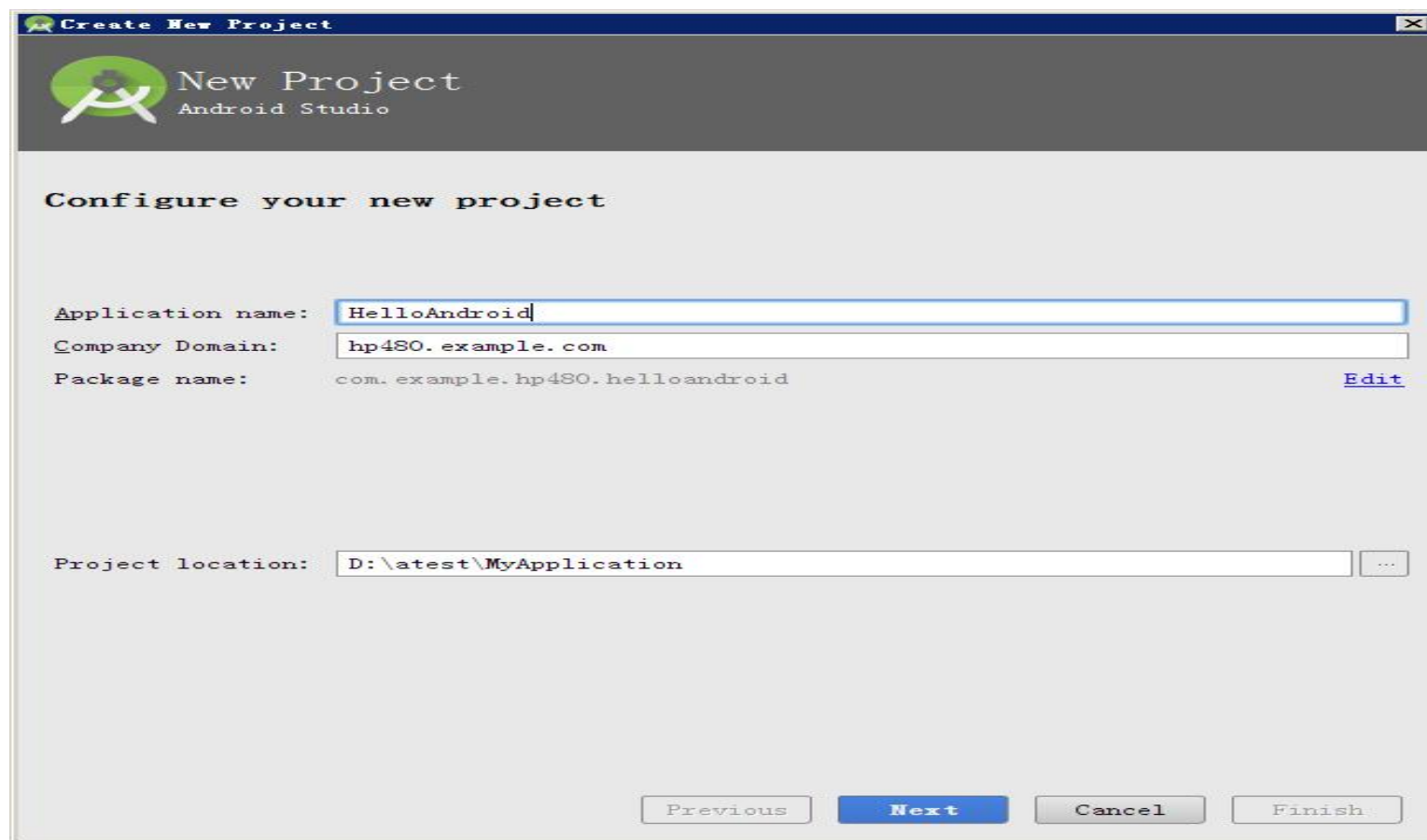
- 1、创建一个新的Android项目

启动Eclipse, 选择 File (文件) -> New (新建) -> Project (项目) 菜单, 在如图1.12所示的“新建项目”对话框选择“Android Application Project”, 点击“Next”按钮。



## • 2、填写应用程序的参数

在“Android新应用程序”信息对话框中输入应用程序名称、项目名称、包名等参数。并选择Android SDK的版本



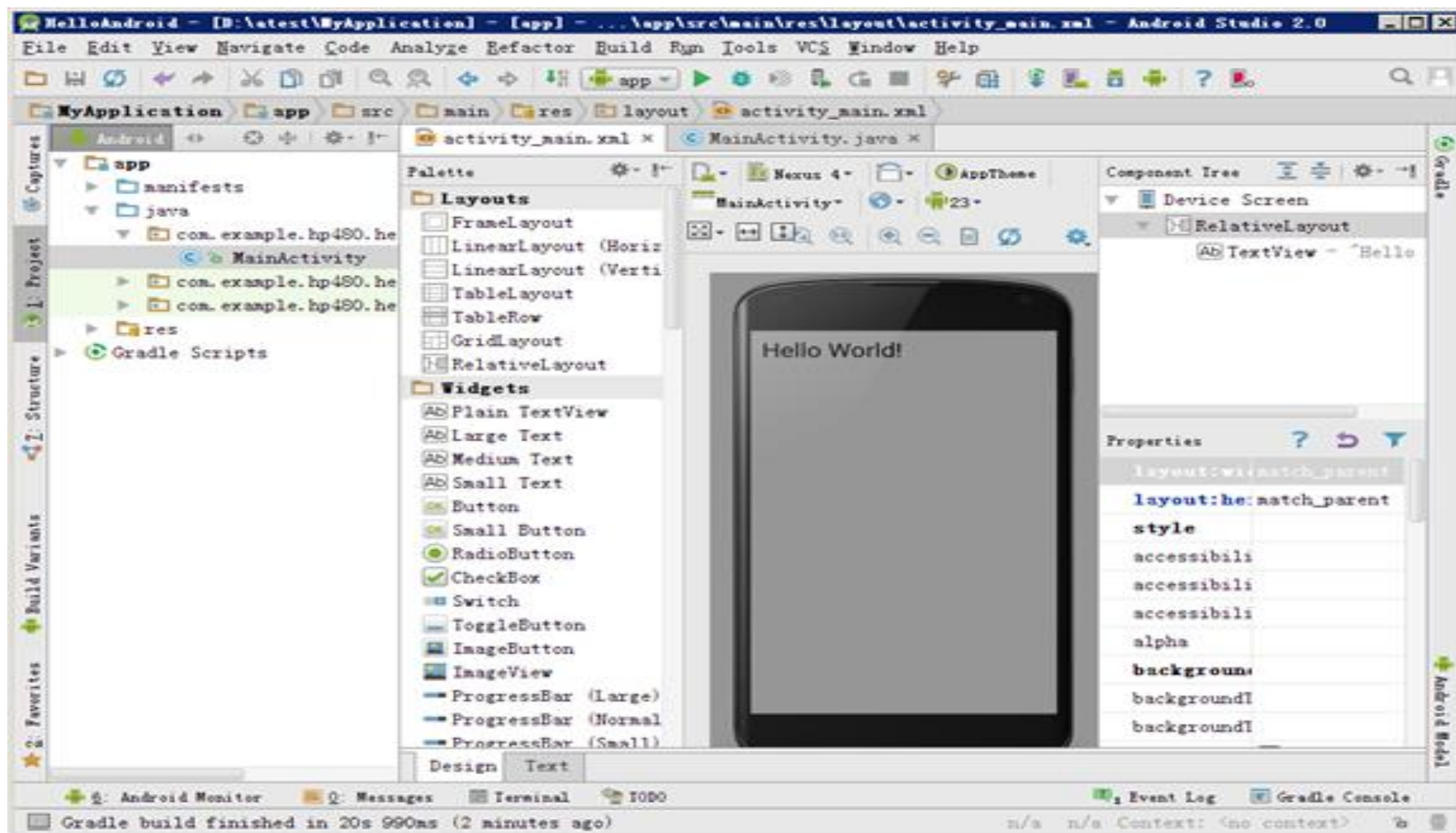
The screenshot shows the "Create New Project" dialog in Android Studio. The dialog has a title bar "Create New Project" and a header area with the Android Studio logo and the text "New Project" and "Android Studio". Below the header, the text "Configure your new project" is displayed. The dialog contains several input fields and buttons:

- Application name:** A text field containing "HelloAndroid".
- Company Domain:** A text field containing "hp480.example.com".
- Package name:** A text field containing "com.example.hp480.helloandroid". To the right of this field is a blue "Edit" link.
- Project location:** A text field containing "D:\atest\MyApplication". To the right of this field is a button with three dots "...".

At the bottom of the dialog, there are four buttons: "Previous", "Next" (which is highlighted in blue), "Cancel", and "Finish".



系统自动生成一个Android应用项目框架。



## 1.4.3 编写代码MainActivity.java

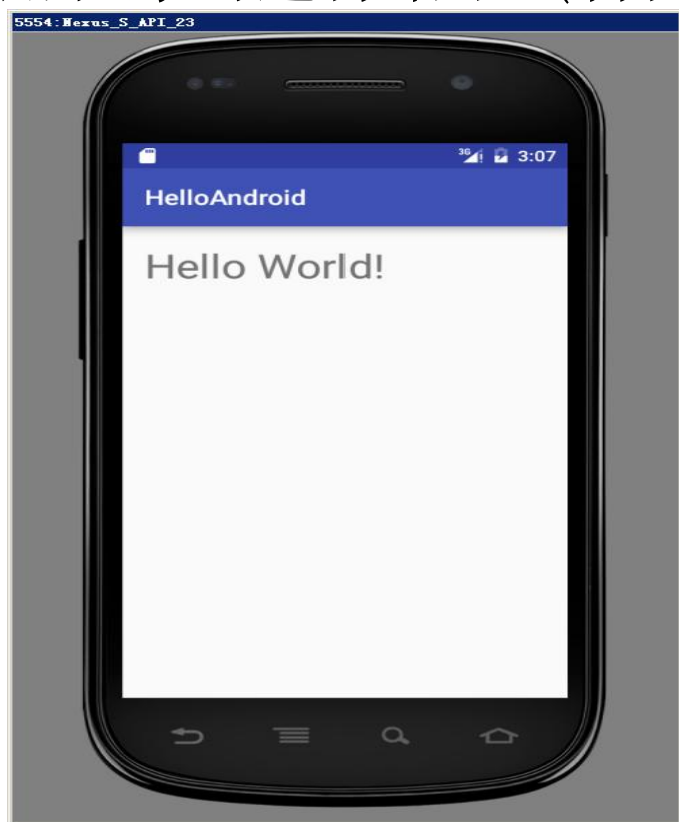
系统自动生成的如下代码:

```
1 package com.example.helloandroid;
2 import android.app.Activity;
3 import android.os.Bundle;
4 public class MainActivity extends Activity
5 { /** Called when the activity is first created. */
6     @Override
7     public void onCreate(Bundle savedInstanceState)
8     {
9         super.onCreate(savedInstanceState);
10        setContentView(R.layout.main);
11    }
12 }
```

显示main.xml定义的用户界面

## 1.4.5 在模拟器中运行应用程序

- 单击工具栏“运行Android Application”按钮，运行AVD模拟器，可以看到应用程序的运行结果（首次运行程序时可能耗时较长）。

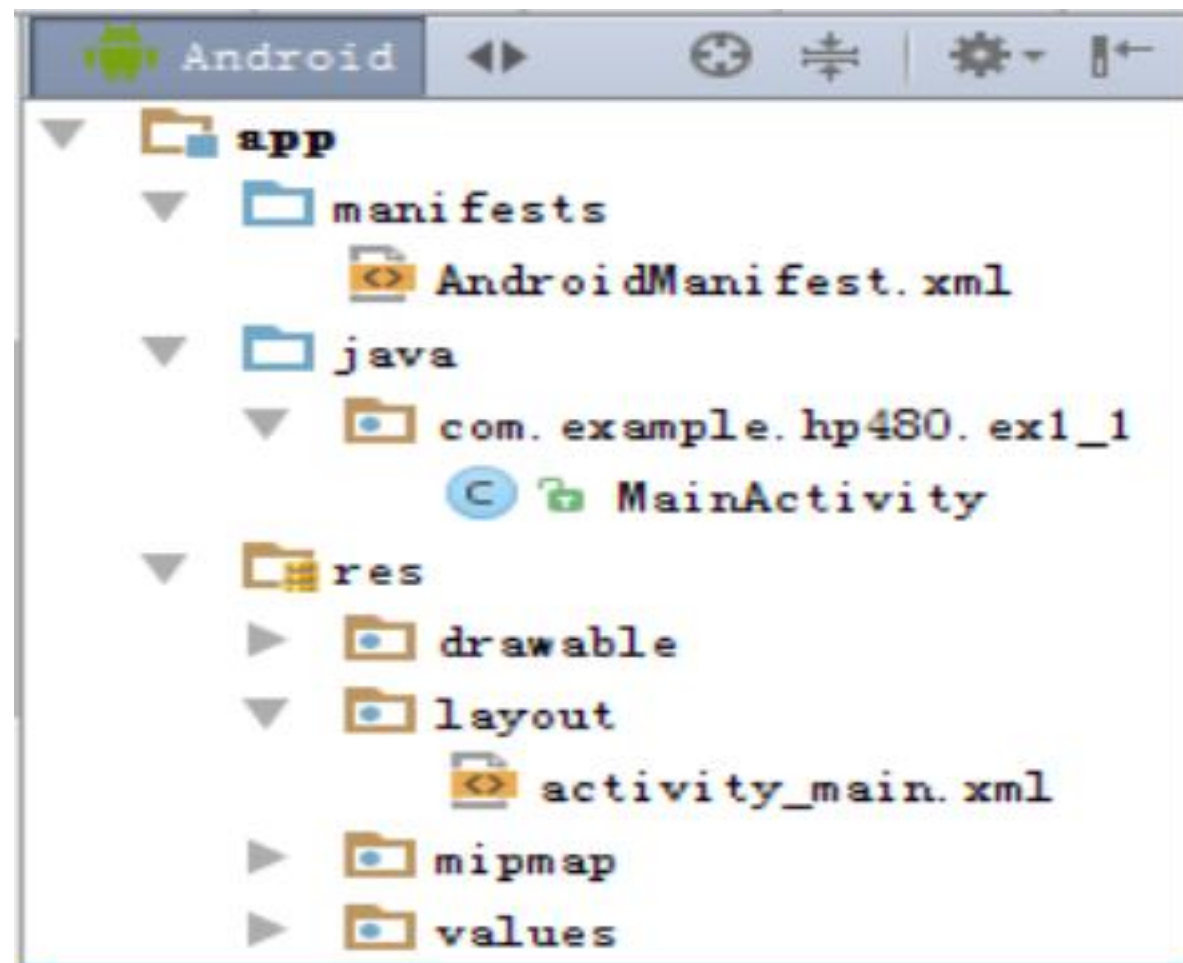


我建议大家还是在实机上进行调试，速度快，而且很方便

## 1.5 Android 目录结构

# 1.5 Android项目结构

- 打开HelloAndroid项目，在“项目资源管理器”中可以看到应用项目的目录和文件结构。



下面对 app 模块下的文件目录结构的基本内容进行介绍。

manifests:

AndroidManifest.xml: 项目的配置信息文件

java: 主要为源代码和测试代码

res: 主要是资源目录, 存储所有的项目资源

values: 存储 app 引用的一些值

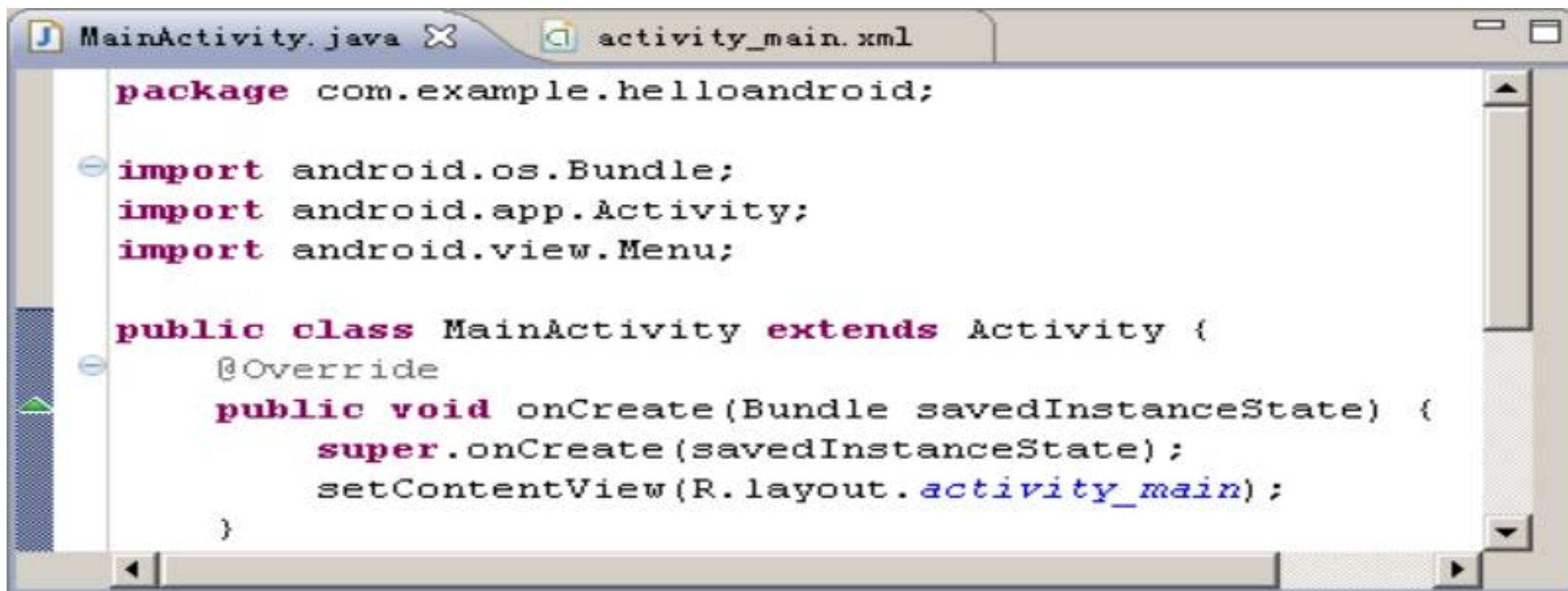
- colors.xml: 存储了一些 color 的样式
- dimensions.xml: 存储了一些公用的 dp 值
- strings.xml: 存储了引用的 string 值
- styles.xml: 存储了 app 需要用到的一些样式

Gradle Scripts: build.gradle 为项目的 gradle 配置文件



## 1、Java源代码目录

- Java目录存放Android应用程序的Java源代码文件。

A screenshot of an IDE window showing the MainActivity.java file. The window has two tabs: 'MainActivity.java' and 'activity\_main.xml'. The code is as follows:

```
package com.example.helloandroid;

import android.os.Bundle;
import android.app.Activity;
import android.view.Menu;

public class MainActivity extends Activity {
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }
}
```

## 2、资源目录res及资源类型


- **res**目录用于存放项目所需要的声音、图片、用户界面等资源文件。

目录结构	资源类型
res/values	存放字符串，颜色，尺寸，数组，主题，类型等资源
res/layout	xml布局文件
res/drawable	图片（bmp,png,gif,jpg等）
res/anim	xml格式的动画资源（帧动画 和 补间动画）
res/menu	菜单资源
res/raw	可以放任意类型文件，一般存放比较大的音频、视频、图片或文档，会在R类中生成资源id，封装在apk中。
assets	可以存放任意类型，不会被编译，与raw相比，不会在R类中生成资源id。

## 2、资源目录res及资源类型

- (1) drawable细分为 drawable-hdpi、drawable-ldpi、drawable-mdpi、drawable-xhdpi子目录，分别存放分辨率大小不同的图标资源,以便相同的应用程序在分辨率大小不同的显示窗体上都可以顺利显示。

# 几种分辨率大小不同的图标

子目录	图标分辨率大小	图例
<code>drawable-xhdpi</code>	$96 \times 96$	
<code>drawable-hdpi</code>	$72 \times 72$	
<code>drawable-mdpi</code>	$48 \times 48$	
<code>drawable-ldpi</code>	$36 \times 36$	

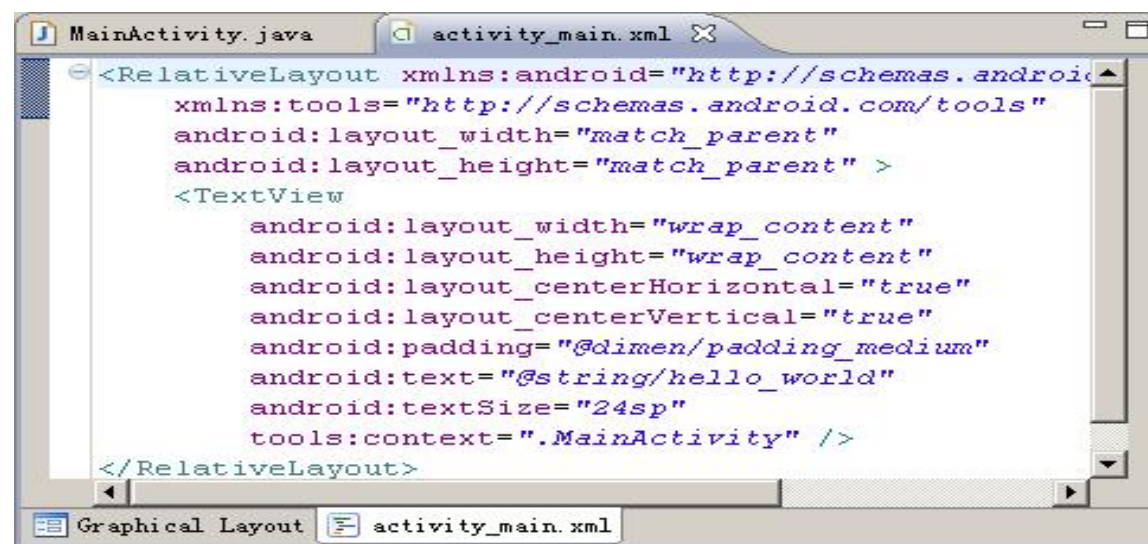
## 2、资源目录res及资源类型

- (2) 在layout子目录存放用户界面布局文件。

其目录中有一个系统自动生成 **activity\_main.xml** 文件，它可以按可视化的图形设计界面显示，也可以按代码设计界面显示，如图1.23 (a)、(b) 所示。



(a) 图形设计界面



(a) 代码设计界面

## 2、资源目录res及资源类型

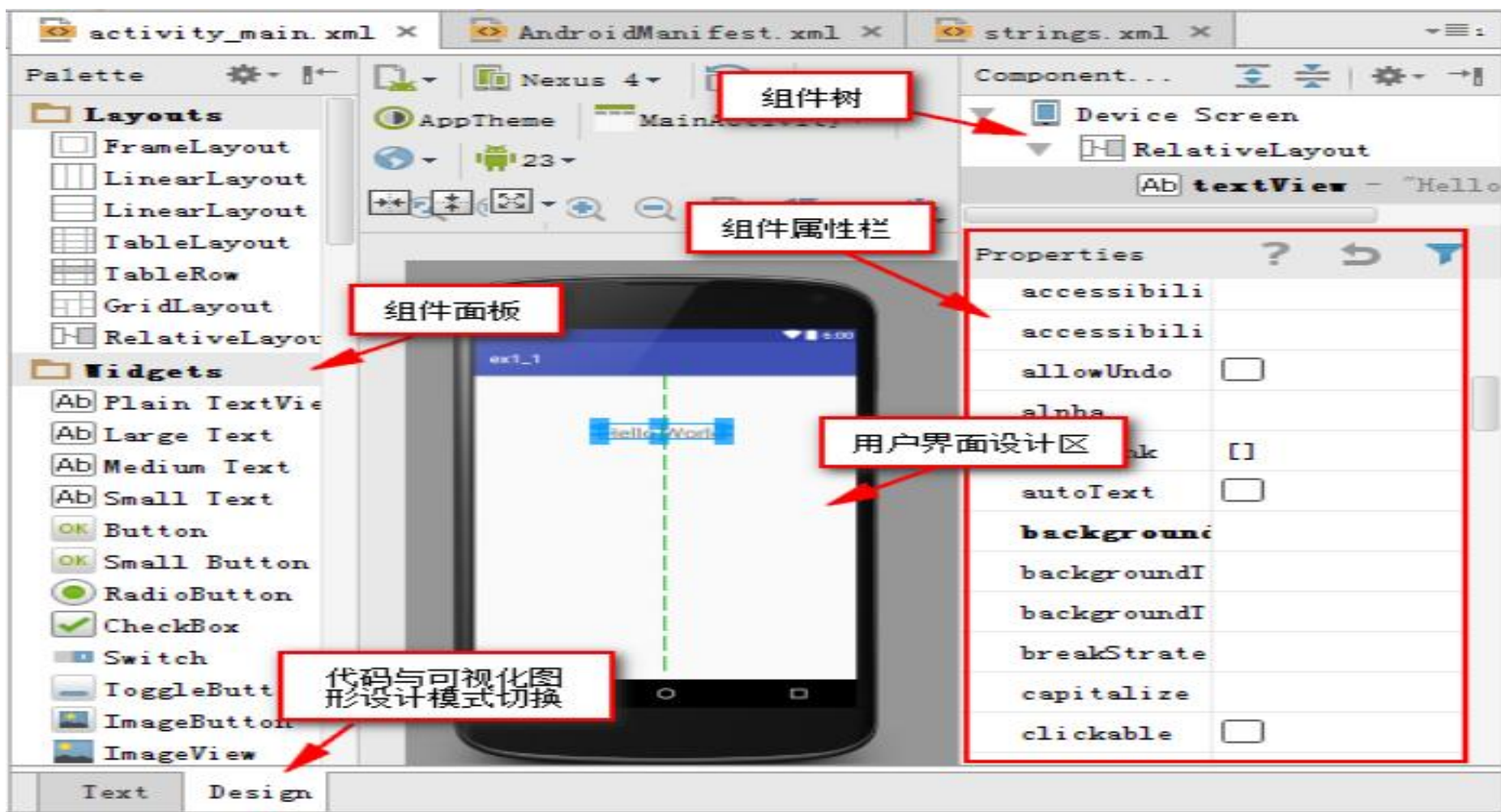
- (3) values子目录存放参数描述文件资源。

这些参数描述文件都是XML文件，如字符串（string.xml）、颜色（color.xml）、数组（arrays.xml）等。



## 2、layout子目录

- 在layout子目录存放用户界面布局文件。



### 3. r/debug目录

当在程序中引用资源需要使用R类，其引用形式如下：

**R.资源文件类型.资源名称**

例如，

(1) 在Activity中显示布局视图：

```
setContentView(R.layout.main);
```

(2) 程序要获得用户界面布局文件中的按钮实例Button1：

```
mButtn = (Button)findViewById(R.id.Button1);
```

(3) 程序要获得用户界面布局文件中的文本组件实例TextView1：

```
mEditText = (EditText)findViewById(R.id.EditText1);
```

## 4、AndroidManifest.xml项目配置文件

- AndroidManifest.xml是每个应用程序都需要的系统配置文件，它位于应用程序根目录下。

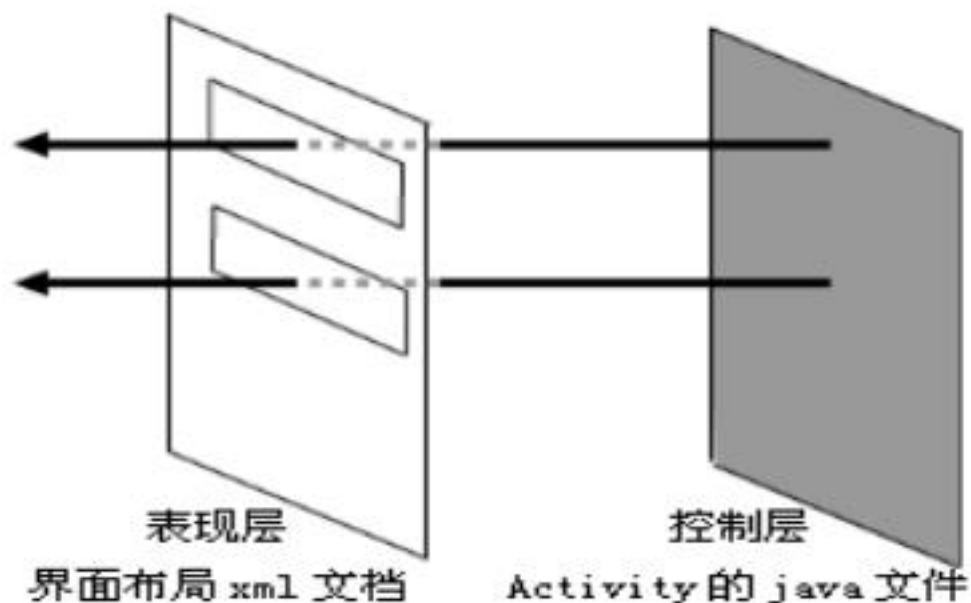
# AndroidManifest.xml文件代码说明

代码元素	说 明
<code>manifest</code>	xml文件的根结点，包含了 <code>package</code> 中所有的内容
<code>xmlns:android</code>	命名空间的声明。使得Android中各种标准属性能在文件中使用。
<code>package</code>	声明应用程序包
<code>uses-sdk</code>	声明应用程序所使用的Android SDK版本
<code>application</code>	<code>application</code> 级别组件的根结点。声明一些全局或默认的属性，如标签、图标、必要的权限等。
<code>android:icon</code>	应用程序图标
<code>android:label</code>	应用程序名称
<code>activity</code>	<code>Activity</code> 是一个应用程序与用户交互的图形界面。每一个 <code>Activity</code> 必须有一个 <code>&lt; activity &gt;</code> 标记对应。
<code>android:name</code>	应用程序默认启动的活动程序 <code>Activity</code> 界面
<code>intent-filter</code>	声明一组组件支持的Intent值。在Android中，组件之间可以相互调用，协调工作，Intent提供组件之间通讯所需要的相关信息
<code>action</code>	声明目标组件执行的Intent动作。
<code>category</code>	指定目标组件支持的Intent 类别

## 1.5.2 Android应用程序架构分析

- 1、逻辑控制层与表现层

在Android应用程序中，逻辑控制层与表现层是分开的设计的。逻辑控制层由Java应用程序实现，表现层由XML文档描述，如图所示。



## 2、Android程序的组成结构

```

1 package com. example.HelloAndroid;  ← 包声明语句

2 import android.app.Activity;  ] ← 导入包
3 import android.os.Bundle;

4 public class MainAndroid extends Activity  ← 类标志
5 {                                           ← 类名
6     public void onCreate(Bundle savedInstanceState) ← 重写 onCreate()方法
7     {
8         super.onCreate(savedInstanceState); ← 调用父类 Activity 的 onCreate()方法
9         setContentView(R.layout. activity_main);
10         }                                  ← 在屏幕上显示内容的方法
11 }

```



其中:

- (1) 第1行是包声明语句, 这个名字是在建立应用程序的时候指定。在这里设定为:  
**package com.HelloAndroid。**

这一行的作用是指出这个文档所在的名称空间。“package”(包)是其关键字。使用名称空间的原因是程序一旦扩展到某个大小, 程序中的变量名称、方法名称、类名等难免重复, 这时就可以通过定义名称空间, 将定义的名称区隔, 以避免相互冲突的情形发生。

- (2) 第2、3行是导入包的声明语句。这二条语句的作用是告诉系统编译器, 编译程序时要导入“android.app.Activity”和“android.os.Bundle”二个包。“import”(导入)是其关键字。在Java语言中, 使用任何API都要事先导入相对应的包。

- (3) 第4 ~ 11行是类的定义，这是应用程序的主体部分。  
Android应用程序是由类组成的，类的一般结构为：

```
public class MainAndroid extends Activity //类声明
{
    ... ; // 类体
}
```

- (4) 第6 ~ 10行是在MainAndroid类的类体中定义一个方法。

## 1.6 Android应用程序设计示例

## 1.6 Android应用程序设计示例

【例1-1】在模拟器中显示“我对学习Android很感兴趣!”。

- (1) 新建一个Android项目，其项目名称为Ex01\_01，包名为com.ex01\_01。
- (2) 在系统自动生成的应用程序中，打开修改资源目录res\values中的字符串文件string.xml，找到XML文档元素

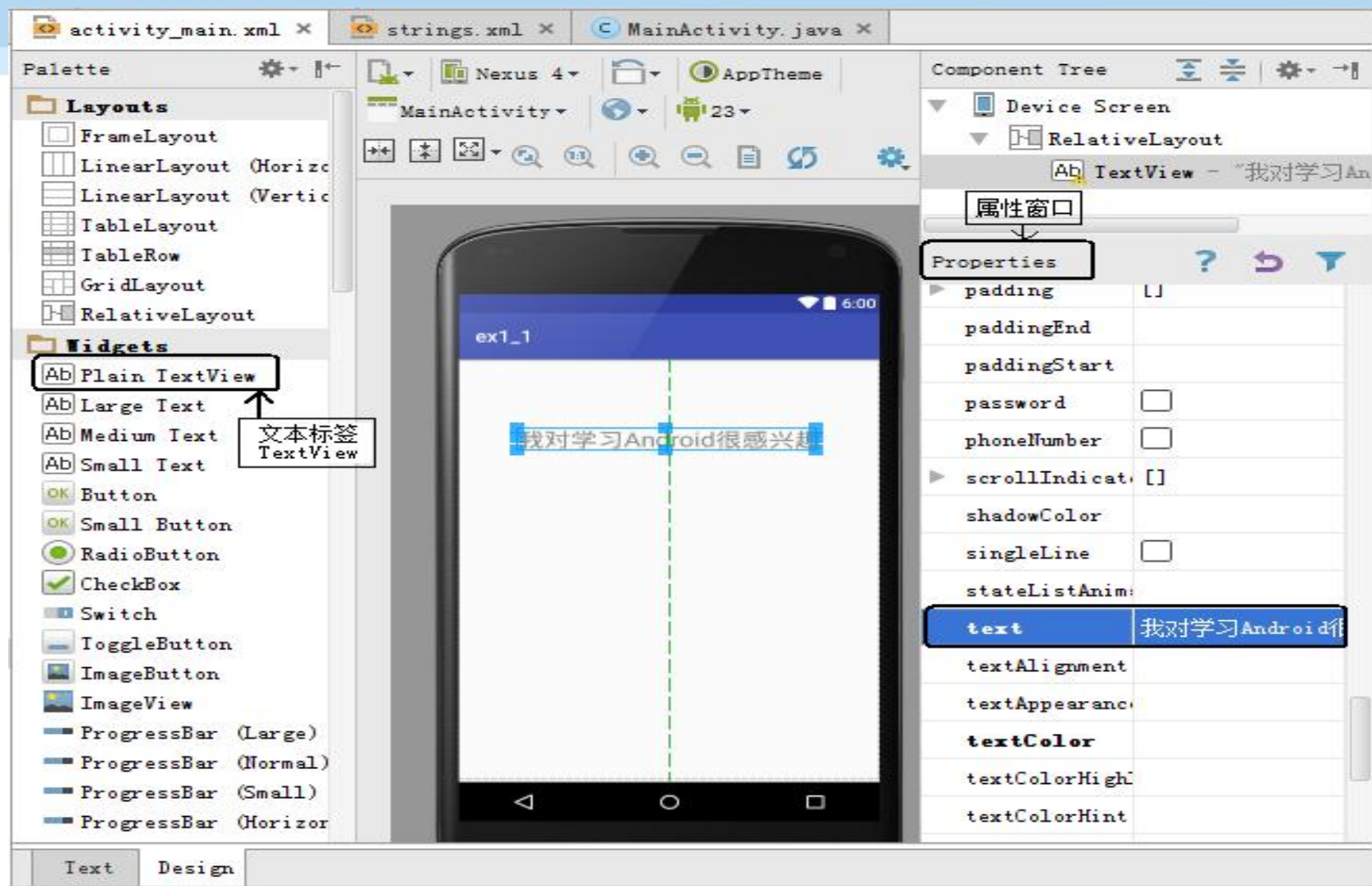
```
<string name="hello">
```

```
Hello World, Ex01_01Activity!</string>
```

将其修改为： <string name="hello">

```
我对学习Android很感兴趣!</string>
```

- (3) 保存程序。设置菜单“运行”的“运行配置”项，运行项目。



## 【例1-2】 设计一个显示资源目录中图片文件的程序。

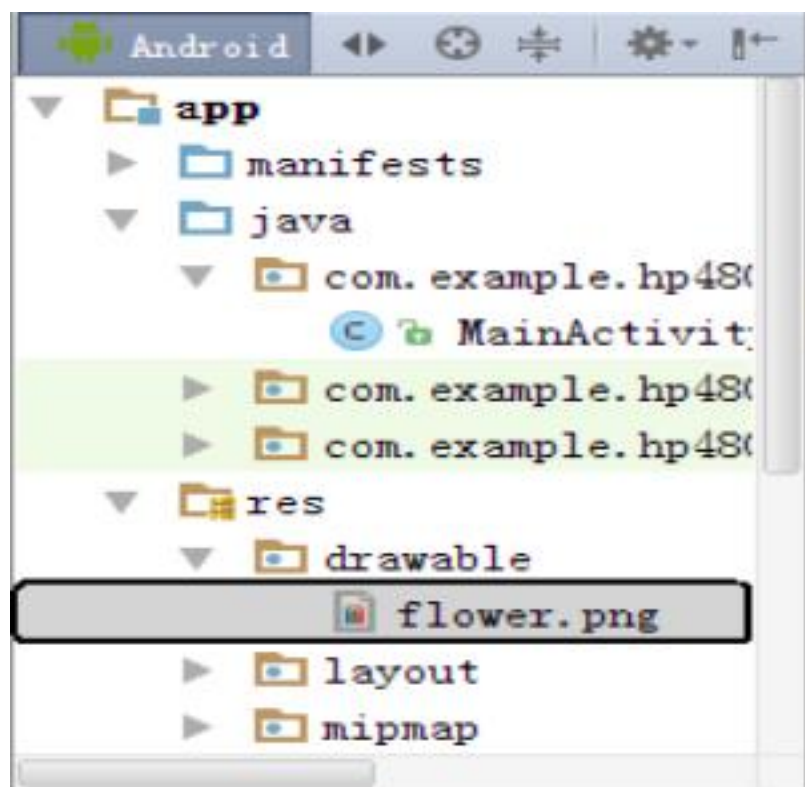
- (1) 在eclipse中新建一个Android项目，其项目名称Application Name为Ex01\_02，包名Package Name为com.ex01\_02。
- (2) 把事先准备的图片文件flower.png复制到资源目录res\drawable中，如图1.26 (a) 所示。
- (3) 打开源代码目录src中的MainActivity.java文件，编写代码如下：



增加导入**ImageView**类的语句

```
package com.ex01_02;
import android.app.Activity;
import android.os.Bundle;
import android.widget.ImageView;
public class MainActivity extends Activity {
    /** Called when the activity is first created. */
    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.main);
        //创建ImageView对象并实例化
        ImageView img = new ImageView(this);
        //ImageView对象设置引用图片资源
        img.setImageResource(R.drawable.flower);
        setContentView(img);
    }
}
```

- (4) 保存程序。设置菜单“运行”的“运行配置”项，运行项目。



# 上机实践

- 1、编写Android应用程序，在模拟器中显示“我喜欢Android!”。
- 2、编写Android应用程序，在模拟器中显示一个图形文件。

Q&A

谢谢大家！