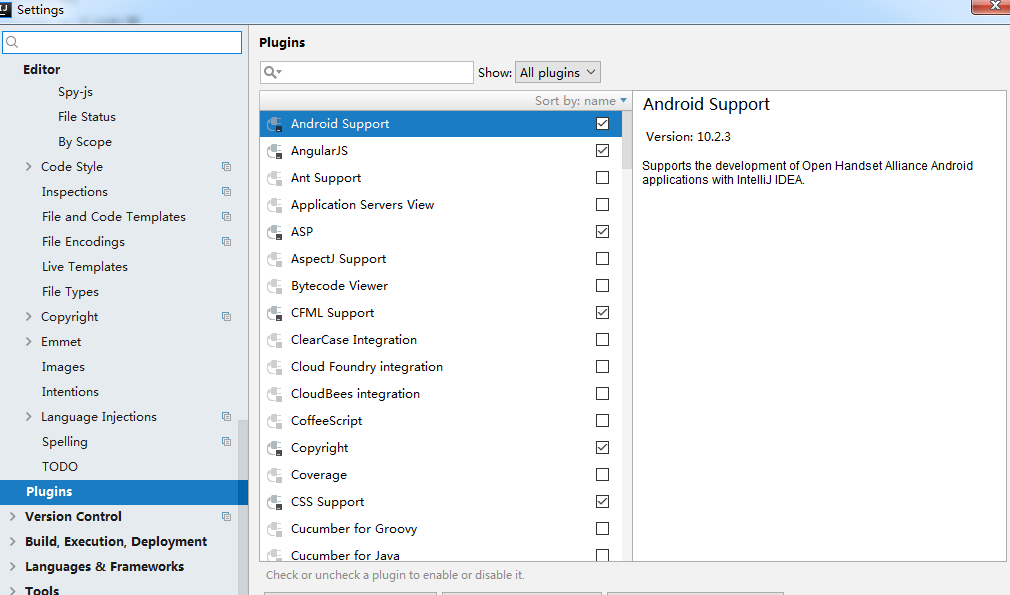
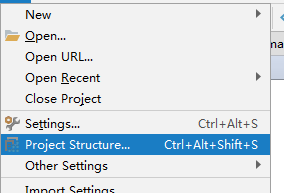
idea配置SDK

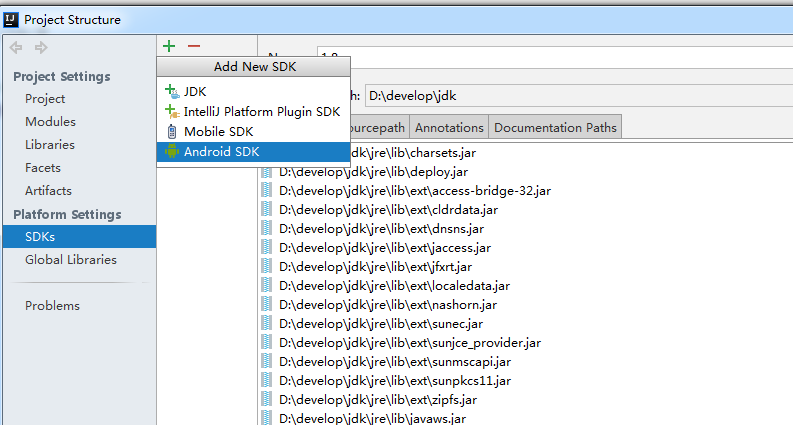
前提是JDK安装成功并设置了环境变量

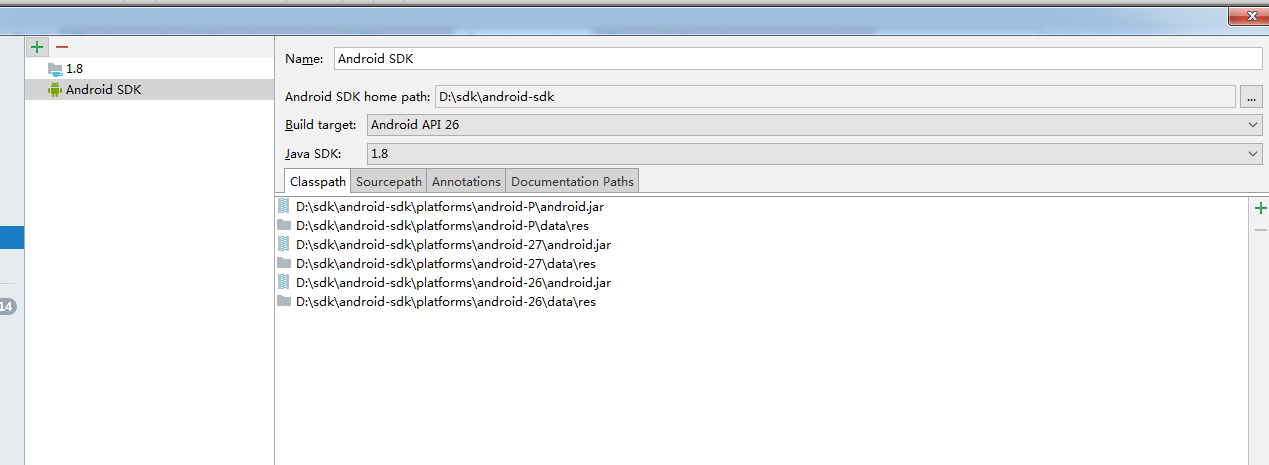
检查是否安装andriod插件



开始配置SDK







指定SDK的安装路径

注意安装SDK的时候要注意文件夹不要带windows否则找不到安装环境

idea下载gradel插件并配置环境变量

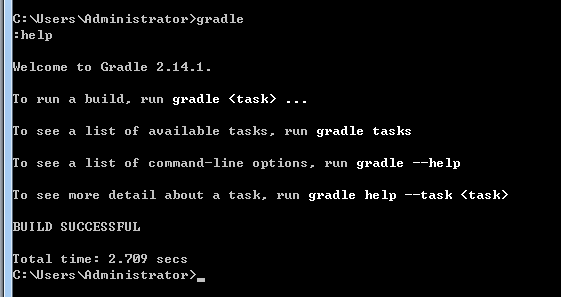
gradel文件地址

C:\Users\Administrator\.gradle

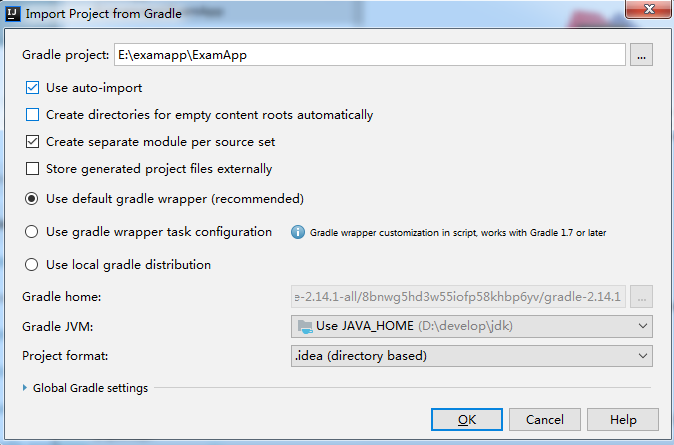
Path的value

C:\Users\Administrator\.gradle\wrapper\dists\gradle-2.14.1-all\8bnwg5hd3w55iofp58khbp6yv\gradle-2.14.1

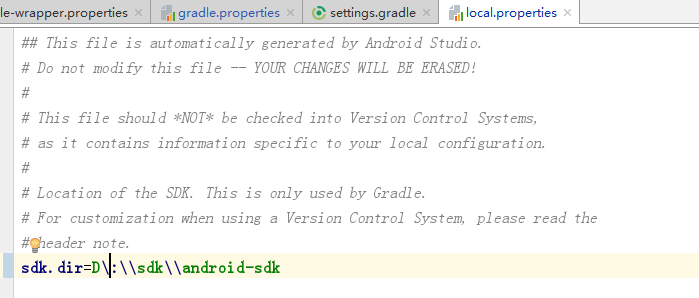
Dos 窗口检查gradle是否安装成功



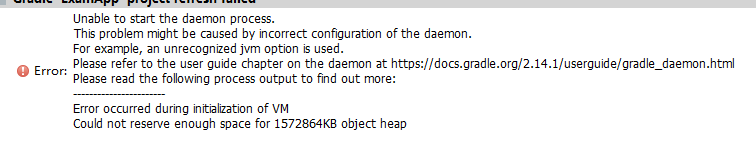
打开idea checkout项目



项目下载后设置本地sdk位置



项目下载后报错:内存溢出

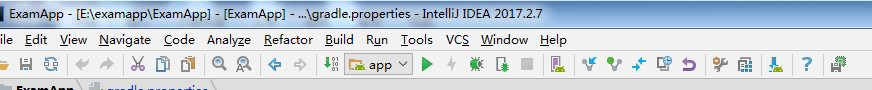


解决办法:去项目下的gradle.properties 改为如下行代码

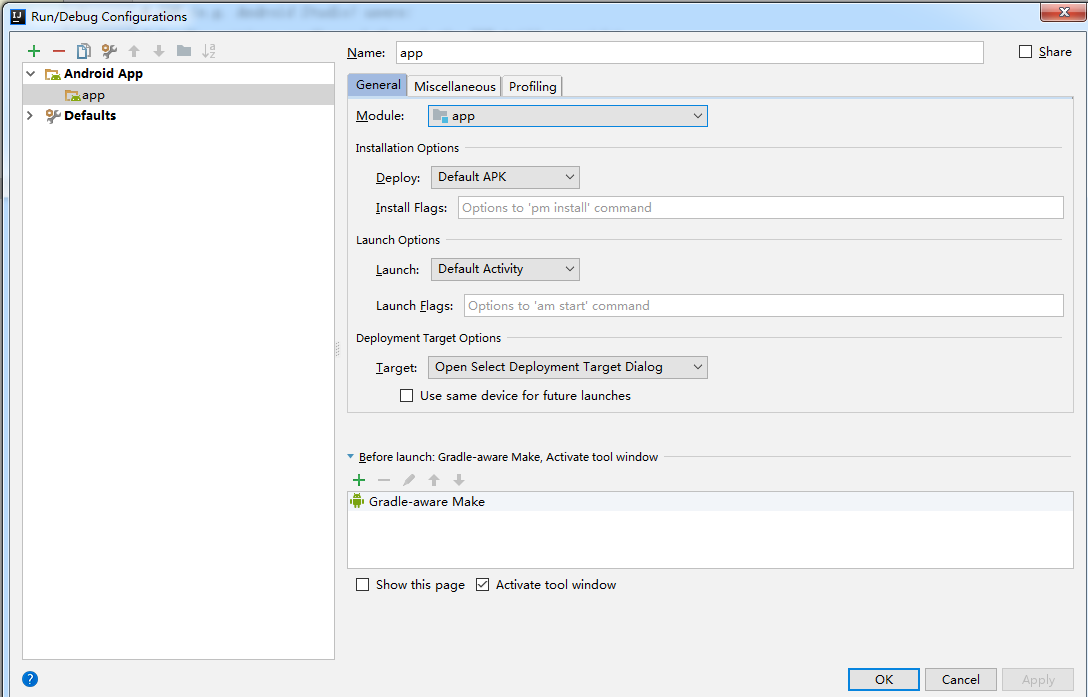




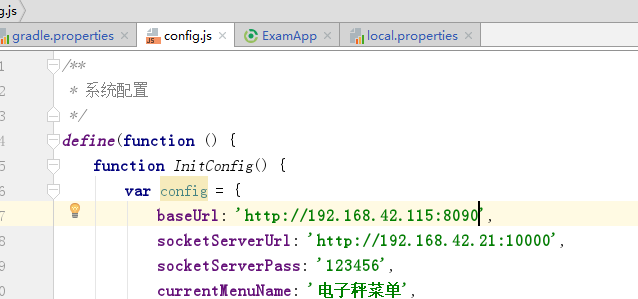
下载完项目后没有如下图所示的app选项



添加方法



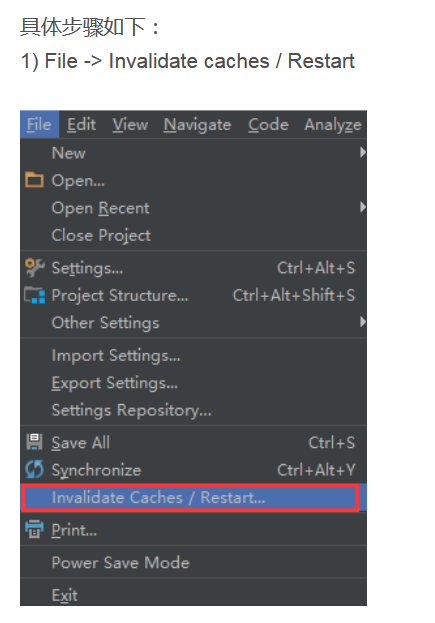
修改前台文件的ip

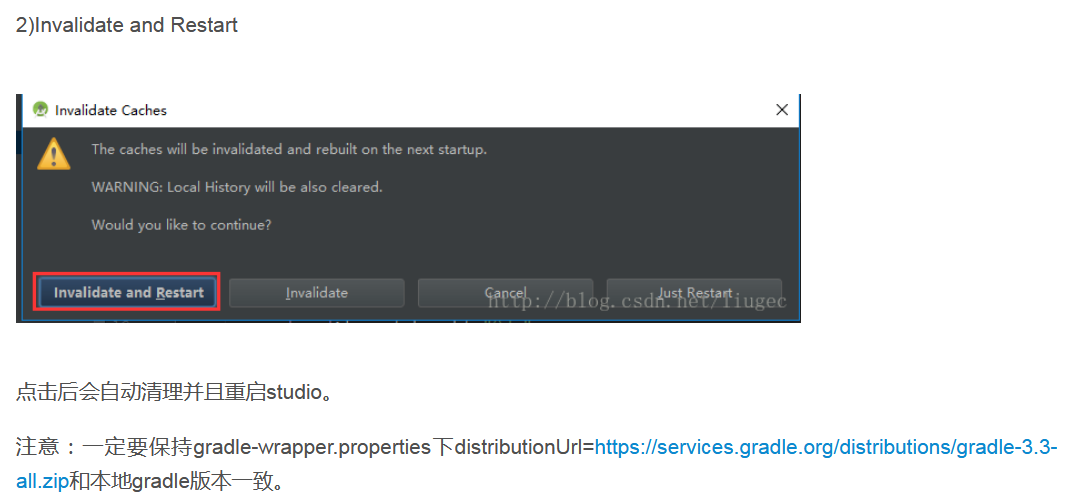


**错误内容：**   
Error:Could not run build action using Gradle installation ‘D:\AndroidStudio\AS2.x\gradle\gradle-3.3’.

报错内容:Your project path contains non-ASCII characters





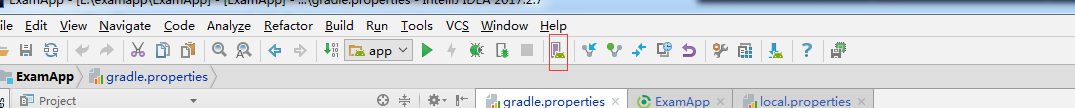


解决app有红叉问题





下载模拟器

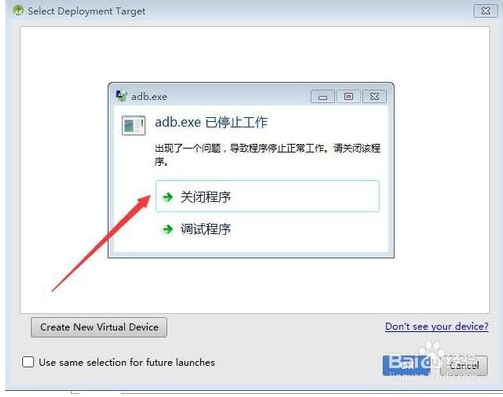


官方文档: http://www.runoob.com/w3cnote/android-tutorial-project-src-analysis.html

Andriod的目录结构

启动项目报

android studio 虚拟机adb.exe已停止工作的处理



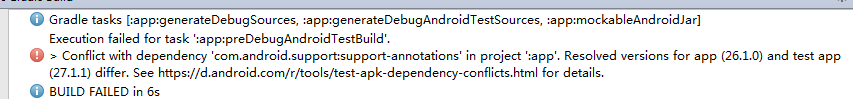
在运行里输入cmd，打开命令行工具，使用netstat -aon|findstr 5037查看adb.exe的5037端口是否被占用



我们可以看到5037被进程为6720的进程所占用，然后用命令tasklist /fi "PID eq 6720"的进程是什么程序，本人的是360电脑端的手机助手所占用。

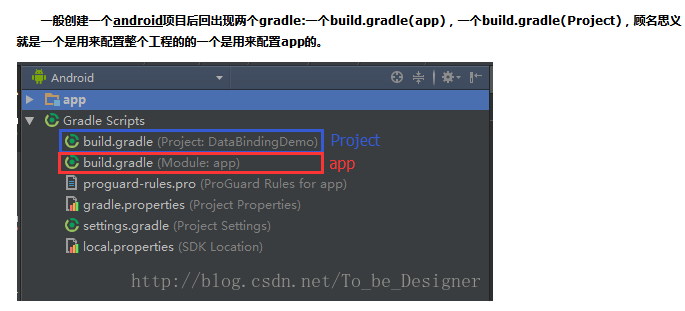


杀掉占用端口程序,重新启动虚拟机或真机

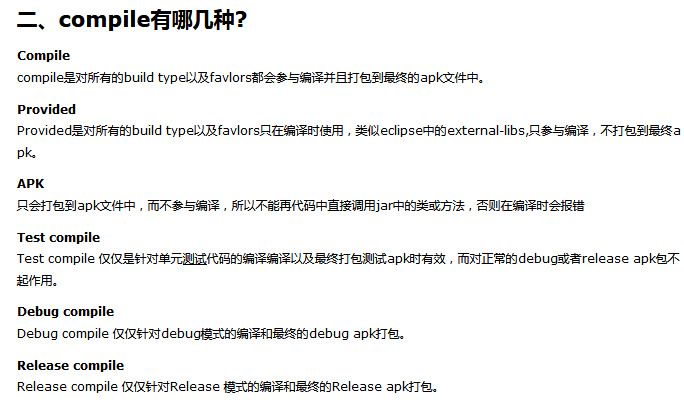


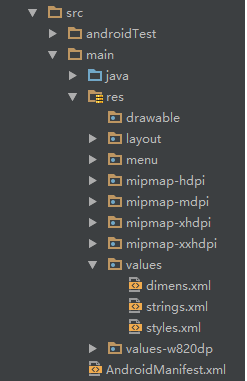
app：preDebugAndroidTestBuild'的执行失败。  
>与项目'：app'中的依赖'com.android.support:support-annotations'冲突。 应用程序（26.1.0）和测试应用程序（27.1.1）的已解决版本不同。

Clean并重新编译一下



## ****一、gradle中dependencies 的区别：****

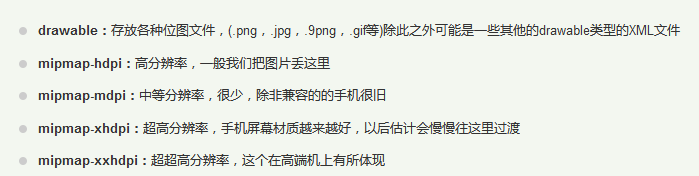
**在Project中的gradle的dependencies 指添加依赖是使用classpath的，classpath一般是添加buildscript本身需要运行的东西，那么buildscript是用来什么呢？buildScript是用来加载gradle脚本自身需要使用的资源，可以声明的资源包括依赖项、第三方插件、maven仓库地址等。   
　　在app中的gradle中dependencies 中添加的使应用程序所需要的依赖包，也就是项目运行所需要的东西。**



**res：**存放我们各种资源文件的地方，有图片，字符串，动画，音频等，还有各种形式的XML文件,另外还有提下这个**assets目录**，虽然这里没有，但是我们可以自己创建，两者的区别在于是否前者下所有的资源文件都会在R.java文件下生成对应的资源id，而后者并不会；前者我们可以直接通过资源id访问到对应的资源；而后者则需要我们通过AssetManager以二进制流的形式来读取！对了，这个R文件可以理解为字典，res下每个资源都都会在这里生成一个唯一的id

mipmap目录

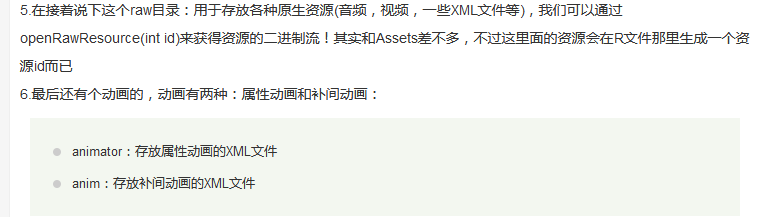
使用mipmap会在图片缩放在提供一定的性能优化，分辨率不同系统会根据屏幕分辨率来选择hdpi，mdpi，xmdpi，xxhdpi下的对应图片，这也不是绝对的,比如我们把所有的图片都丢在了drawable-hdpi下的话,即使手机 本该加载ldpi文件夹下的图片资源,但是ldpi下没有,那么加载的还会是hdpi下的图片! 另外,还有一种情况:比如是hdpi,mdpi目录下有,ldpi下没有,那么会加载mdpi中的资源! 原则是使用最接近的密度级别!另外如果你想禁止Android不跟随屏幕密度加载不同文件夹的资源,只需在AndroidManifest.xml文件中添加android:anyDensity="false"字段即可!



layout：该目录下存放的就是我们的布局文件，另外在一些特定的机型上，我们做屏幕适配，比如480\*320这样的手机，我们会另外创建一套布局，就行：layout-480x320这样的文件夹！

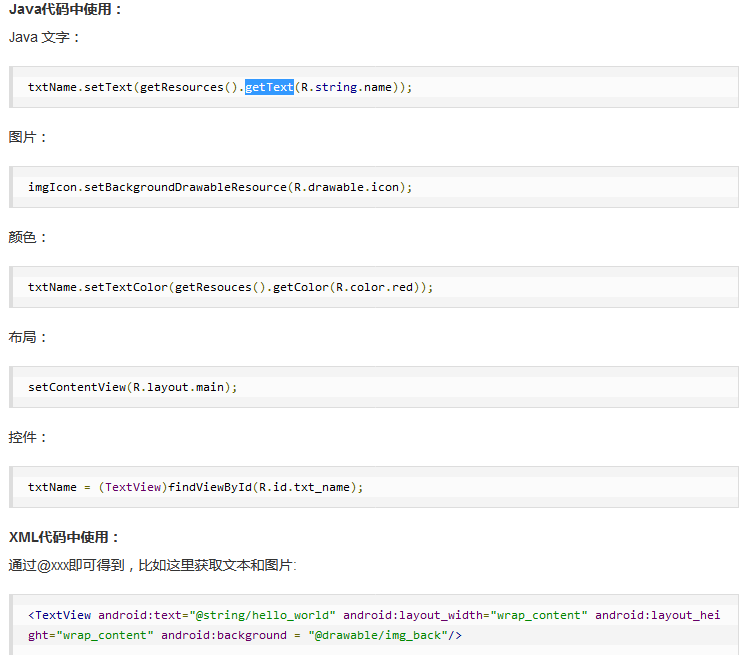
**menu：**在以前有物理菜单按钮，即menu键的手机上，用的较多，现在用的并不多，菜单项相关的资源xml可在这里编写，不知道谷歌会不会出新的东西来替代菜单了





资源的调用

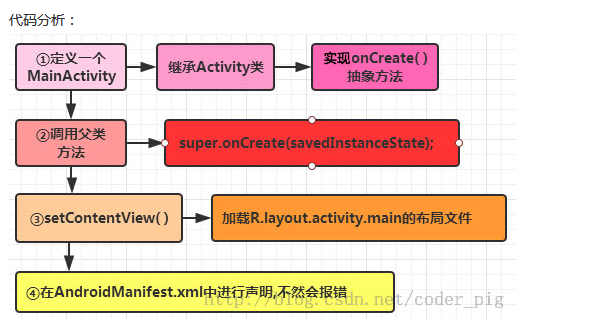
所有的资源文件都会在R.java文件下生成一个资源id，我们可以通过这个资源id来完成资源的访问，使用情况有两种：Java代码中使用和XML代码中使用

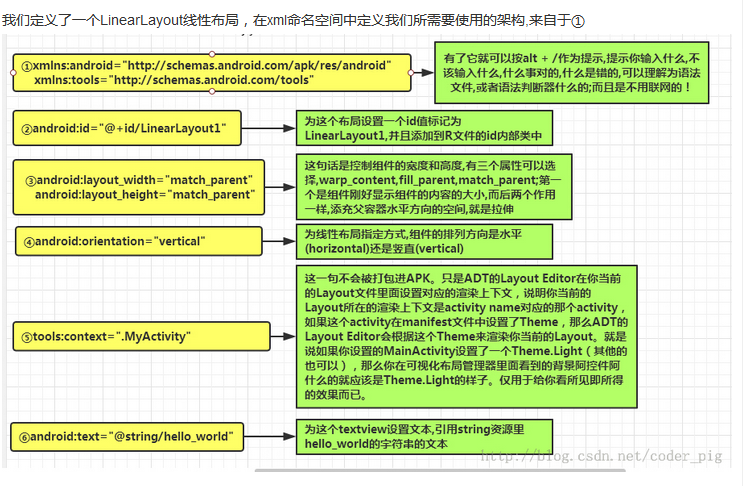


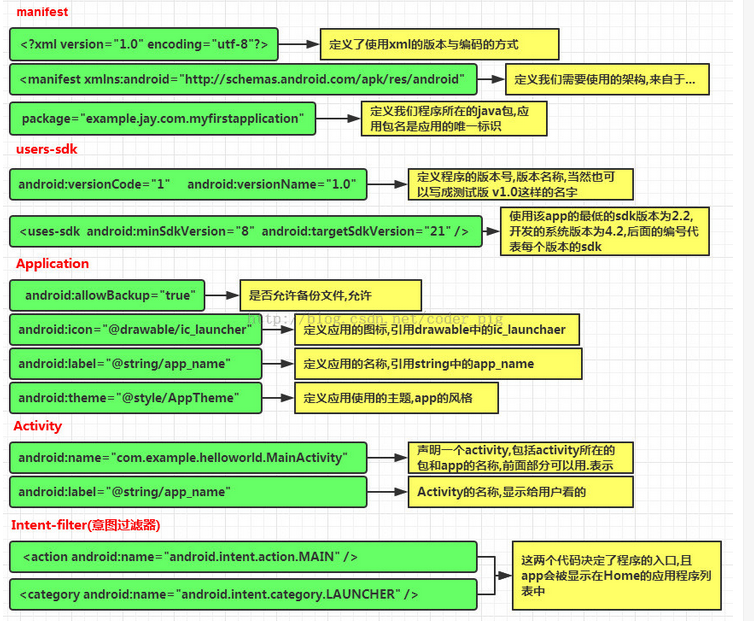
三个重要的文件









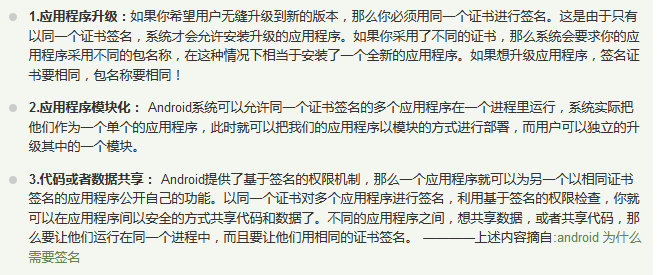




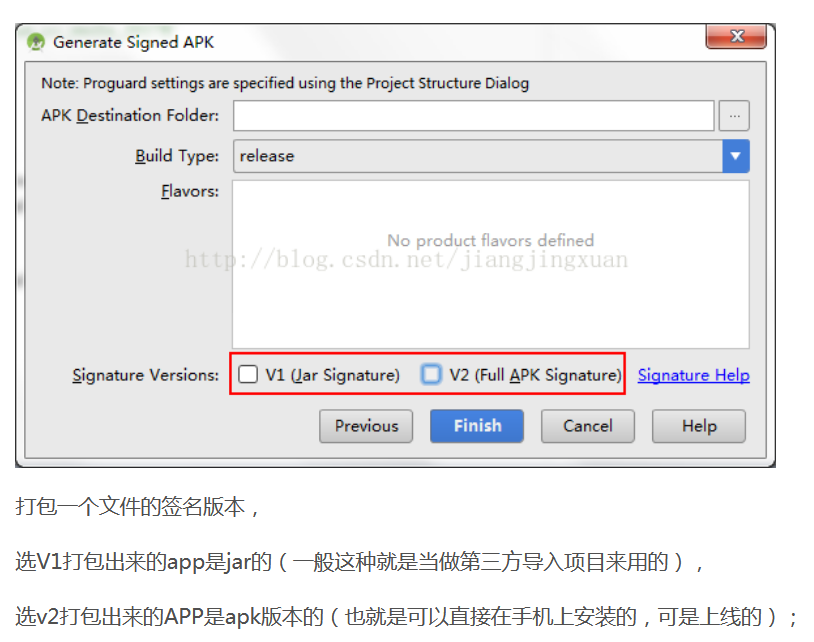
Android程序签名打包

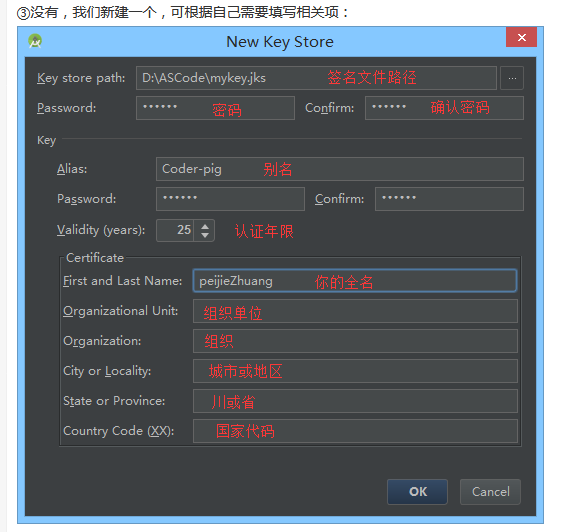
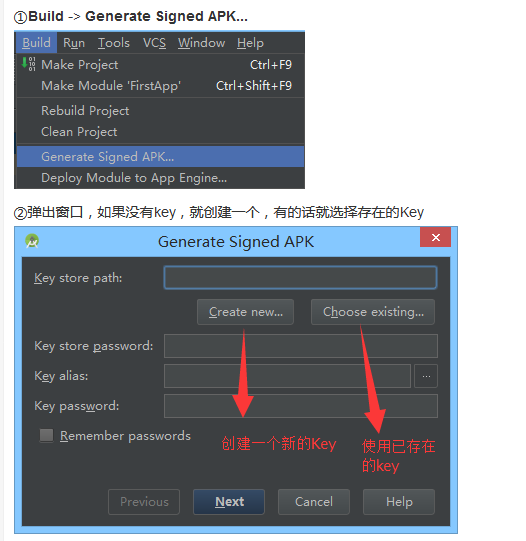
签名: Android APP都需要我们用一个证书对应用进行数字签名，不然的话是无法安装到Android手机上的，平时我们调试运行时到手机上时，是AS会自动用默认的密钥和证书来进行签名；但是我们实际发布编译时，则不会自动签名，这个时候我们就需要进行手动签名了

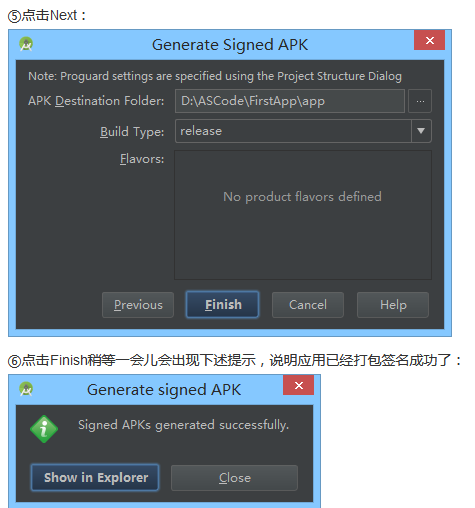
APK签名的好处

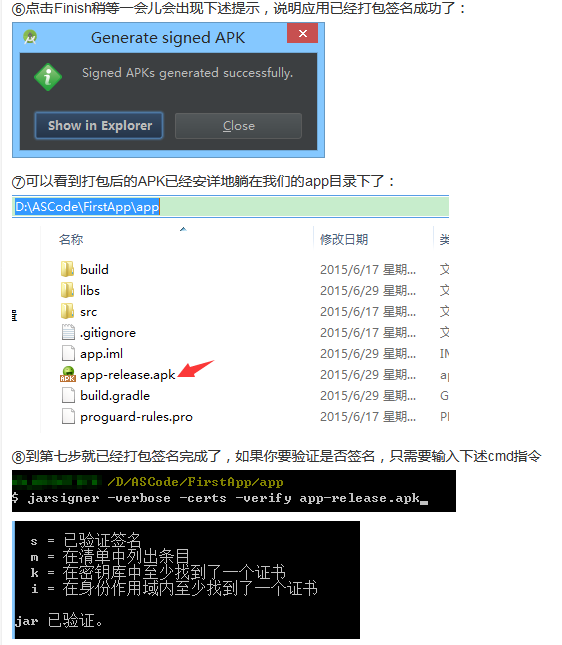


Idea打包app时新建一个密匙选项含义



打包步骤

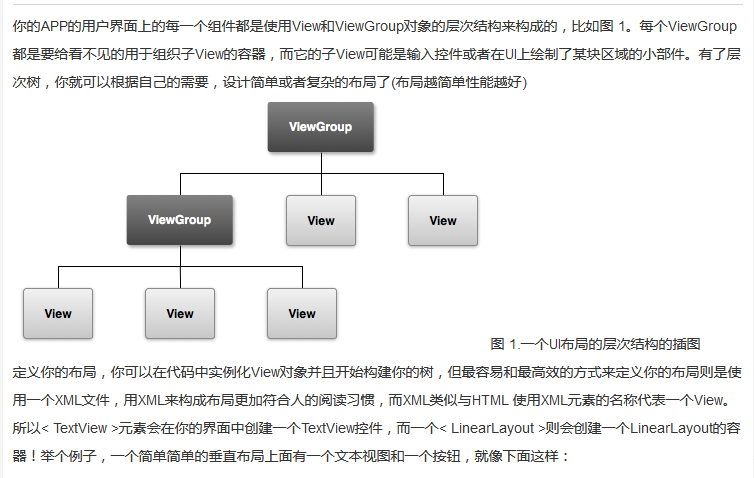




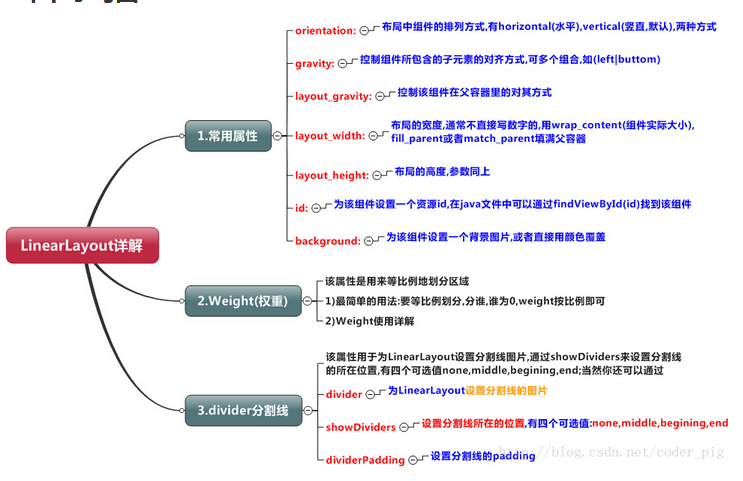
反编译: <http://www.runoob.com/w3cnote/android-tutorial-decompile-apk-get-code-resources.html>

国内API镜像[http://androiddoc.qiniudn.com/guide/topics/ui/overview.html 这个镜像是5.0的API](http://androiddoc.qiniudn.com/guide/topics/ui/overview.html%20这个镜像是5.0的API)

在Android APP中，所有的用户界面元素都是由View和ViewGroup的对象构成的。View是绘制在屏幕上的用户能与之交互的一个对象。而ViewGroup则是一个用于存放其他View（和ViewGroup）对象的布局容器！ Android为我们提供了一个View和ViewGroup子类的集合，集合中提供了一些常用的输入控件(比如按钮和文本域)和各种各样的布局模式（比如线性或相对布局

Android里的图形界面都是由View和ViewGroup以及他们的子类构成的： **View：**所有可视化控件的父类,提供组件描绘和时间处理方法 **ViewGroup：** View类的子类，可以拥有子控件,可以看作是容器 Android UI中的控件都是按照这种层次树的结构堆叠得，而创建UI布局的方式有两种， 自己在Java里写代码或者通过XML定义布局，后者显得更加方便和容易理解！ 也是我们最常用的手段！另外我们一般很少直接用View和ViewGroup来写布局，更多的 时候使用它们的子类控件或容器来构建布局！

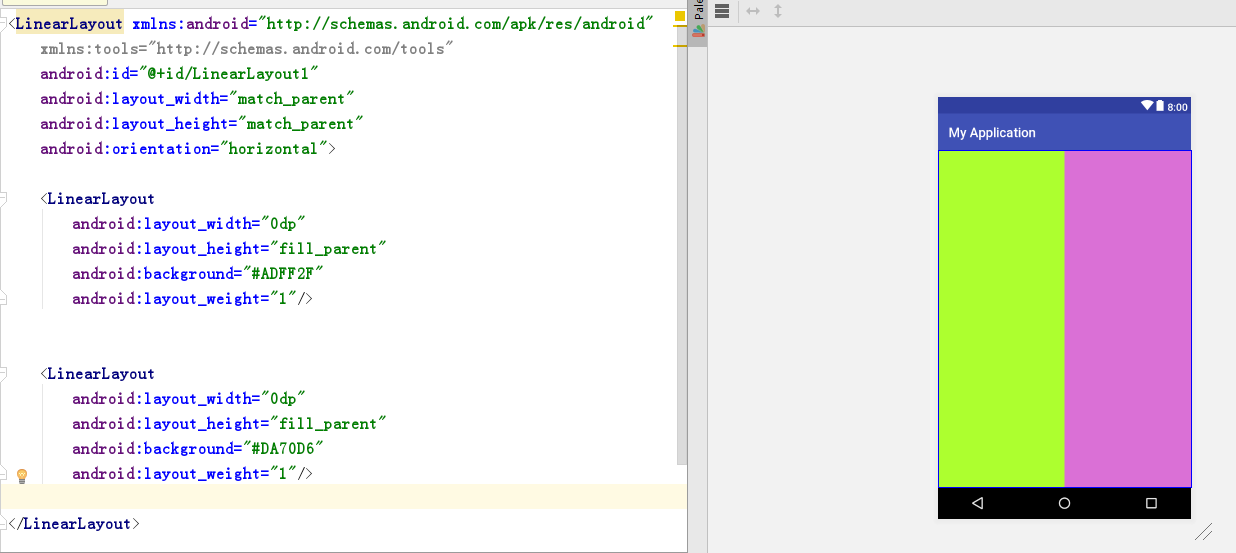
Android中有六大布局,分别是: LinearLayout(线性布局)，RelativeLayout(相对布局)，TableLayout(表格布局) FrameLayout(帧布局)，AbsoluteLayout(绝对布局)，GridLayout(网格布局)



布局标签及其属性值说明

要实现第一个的屏幕1:1的效果,只需要分别把两个LinearLayout的weight改成1就可以了

用法归纳: 按比例划分水平方向:将涉及到的View的android:width属性设置为0dp,然后设置为android weight属性设置比例即可;类推,竖直方向,只需设android:height为0dp,然后设weight属性即可！

**weight属性详解**

不使用上述那种设置为0dp的方式,直接用wrap\_content和match\_parent的话, 则要接着解析weight属性了,分为两种情况,wrap\_content与match\_parent！另外还要看 LinearLayout的orientation是水平还是竖直,这个决定哪个方向等比例划分

**wrap\_content比较简单,直接就按比例的了**





**match\_parent(fill\_parent):这个则需要计算了**







## **LinearLayout设置分割线**

## **对于这种线,我们通常的做法有两种 ①直接在布局中添加一个view,这个view的作用仅仅是显示出一条线,代码也很简单:**

## 

## **第二种则是使用LinearLayout的一个divider属性,直接为LinearLayout设置分割线 这里就需要你自己准备一张线的图片了 1)android:divider设置作为分割线的图片 2)android:showDividers设置分割线的位置,none(无),beginning(开始),end(结束),middle(每两个组件间) 3)dividerPadding设置分割线的Padding**

## 

## 

## **使用Layout\_gravity的一个很重要的问题!!! 问题内容: 在一个LinearLayout的水平方向中布置两个TextView,想让一个左,一个右,怎么搞? 或许你会脱口而出:"gravity设置一个left,一个right就可以啦!**

## 

## **RelativeLayout(相对布局)**

## **LinearLayout也是我们 用的比较多的一个布局,我们更多的时候更钟情于他的weight(权重)属性，等比例划分，对屏幕适配还是 帮助蛮大的;但是使用LinearLayout的时候也有一个问题，就是当界面比较复杂的时候，需要嵌套多层的 LinearLayout,这样就会降低UI Render的效率(渲染速度),而且如果是listview或者GridView上的 item,效率会更低,另外太多层LinearLayout嵌套会占用更多的系统资源,还有可能引发stackoverflow; 但是如果我们使用RelativeLayout的话,可能仅仅需要一层就可以完成了,以父容器或者兄弟组件参考+margin +padding就可以设置组件的显示位置,是比较方便的!当然,也不是绝对的,具体问题具体分析吧! 总结就是:尽量使用RelativeLayout + LinearLayout的weight属性搭配使用吧！**

## 

## 

## 

## 

## 

## **TableLayout(表格布局)**

## **其他布局详见**

## <http://www.runoob.com/w3cnote/android-tutorial-tablelayout.html>

## **GridLayout(网格布局)**

## 

## 

## **加名字 身份证号 名族 英烈子女**

## 

## **三级联动菜单**

## **签署协议:默认为空**

## **签署协议缺少下载提示**

## 

## **签署协议缺少附件展示**

**success**

**success**