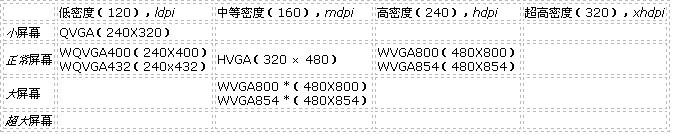
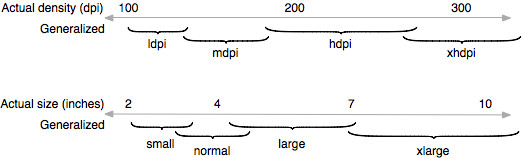
**Android屏幕密度（Density）和分辨率的解释**

移动设备有大有小，那么如何适应不同屏幕呢，这给我们编程人员造成了很多困惑。我也是突然想到这些问题，然后去网上搜搜相关东西，整理如下。  
　　首先，对下面这些长度单位必须了解。  
　　Android中的长度单位详解（dp、sp、px、in、pt、mm）  
　　看到有很多网友不太理解dp、sp和px的区别：现在这里介绍一下dp和sp。dp也就是dip。这个和sp基本类似。如果设置表示长度、高度等属性时可以使用dp 或sp。但如果设置字体，需要使用sp。dp是与密度无关，sp除了与密度无关外，还与scale无关。如果屏幕密度为160，这时dp和sp和px是一 样的。1dp=1sp=1px，但如果使用px作单位，如果屏幕大小不变（假设还是3.2寸），而屏幕密度变成了320。那么原来TextView的宽度 设成160px，在密度为320的3.2寸屏幕里看要比在密度为160的3.2寸屏幕上看短了一半。但如果设置成160dp或160sp的话。系统会自动 将width属性值设置成320px的。也就是160 \* 320 / 160。其中320 / 160可称为密度比例因子。也就是说，如果使用dp和sp，系统会根据屏幕密度的变化自动进行转换。  
　　下面看一下其他单位的含义  
　　px：(pixels)（像素）：屏幕上的点，与密度相关。密度大了，单位面积上的px会比较多。表示屏幕实际的象素。例如，320\*480的屏幕在横向有320个象素，在纵向有480个象素。  
　　dip或dp（与密度无关的像素）。这个和设备硬件有关，为了支持WVGA、HVGA和QVGA 5进制空间推荐使用这个。一种基于屏幕密度的抽象单位。设置一些view的宽高可以用这个，一般情况下，在不同分辨率，都不会有缩放的感觉。如果用px的话，320px占满HVGA的宽度，到WVGA上就只能占一半不到的屏幕了，那一定不是你想要的。  
　　sp（与刻度无关的像素）放大像素– 主要处理字体的大小。  
  
　　in：表示英寸，是屏幕的物理尺寸。每英寸等于2.54厘米。例如，形容手机屏幕大小，经常说，3.2（英）寸、3.5（英）寸、4（英）寸就是指这个单位。这些尺寸是屏幕的对角线长度。如果手机的屏幕是3.2英寸，表示手机的屏幕（可视区域）对角线长度是3.2\*2.54 = 8.128厘米。读者可以去量一量自己的手机屏幕，看和实际的尺寸是否一致。  
　　mm：表示毫米，是屏幕的物理尺寸。  
　　pt：表示一个点，是屏幕的一个物理长度，大小为1英寸的1/72。  
　　换算公式：  
　　px = (density/160)dp  
　　density一般为3个常用固定值240/160/120  
  
　　分别对应WVGA/HVGA/QVGA  
　　不知道知己做的分辨率对应的density是多少可以点击AVD Manager里新建一个模拟器或者选择已有的模拟器点details就可以看到了(1.6以上包括1.6才可以查)  
　　补充:dp跟dip是一样的,但px跟dp(dip)绝对不一样,唯一的情况只有在HVGA分辨率下才是一样,因为HVGA分辨率对应density为160,与除数一样,所以做的时候才会没感觉到差别  
　　其次，我们在创建一个模拟器的时候，有一个选项是选择模拟器的大小，其实就是选择模拟器的屏幕大小，就是我们常说的分辨率。比如：320\*48等。  
　　Density的意思是“密度”。密度，就是说单位面积内的容量大小。  
　　HVGA屏density=160  
　　QVGA屏density=120  
　　WVGA屏density=240  
　　WQVGA屏density=120  
　　density值表示每英寸有多少个显示点，与分辨率是两个概念。  
　　不同density下屏幕分辨率信息：  
　　480\*800的WVGA(density=240)  
　　density=120时 屏幕实际分辨率为240\*400 （两个点对应一个分辨率）  
  
　　横屏是屏幕宽度400px 或者800dip,工作区域高度211px或者480dip  
　　竖屏时屏幕宽度240px或者480dip,工作区域高度381px或者775dip   
　　density=160时 屏幕实际分辨率为320px\*533px （3个点对应两个分辨率）  
　　状态栏和标题栏高个25px或者25dip   
　　横屏是屏幕宽度533px 或者800dip,工作区域高度295px或者480dip  
　　竖屏时屏幕宽度320px或者480dip,工作区域高度508px或者775dip  
  
　　density=240时 屏幕实际分辨率为480px\*800px （一个点对于一个分辨率）  
　　状态栏和标题栏高个38px或者25dip   
　　横屏是屏幕宽度800px 或者800dip,工作区域高度442px或者480dip  
　　竖屏时屏幕宽度480px或者480dip,工作区域高度762px或者775dip  
  
　　apk的资源包中，当屏幕density=240时使用hdpi标签的资源  
　　当屏幕density=160时，使用mdpi标签的资源  
　　当屏幕density=120时，使用ldpi标签的资源。  
　　不加任何标签的资源是各种分辨率情况下共用的。  
　　布局时尽量使用单位dip，少使用px  
  
　　屏幕 宽度 高度 尺寸 大小 密度   
  
　　Type (Pixels) (Pixels) Range (inches) Size Group  
  
　　QVGA 240 320 2.6 - 3.0 Small Low  
　　WQVGA 240 400 3.2 - 3.5 Normal Low  
　　FWQVGA 240 432 3.5 - 3.8 Normal Low  
　　HVGA 320 480 3.0 - 3.5 Normal Medium  
　　WVGA 480 800 3.3 - 4.0 Normal High  
　　FWVGA 480 854 3.5 - 4.0 Normal High  
　　WVGA 480 800 4.8 - 5.5 Large Medium  
　　FWVGA 480 854 5.0 - 5.8 Large Medium  
  
　　屏幕大小和密度对照表 ：  
  
　　  
  
  
　　下面是在manifest中设置app在不同分辨率时，是否支持多密度的方法。  
　　<manifest xmlns:android="http://schemas.android.com/apk/res/android">  
　　...  
　　<supports-screens  
　　android:smallScreens="true"  
　　android:normalScreens="true"  
　　android:largeScreens="true"  
　　android:xlargeScreens="true"  
　　android:anyDensity="true" />  
　　</manifest>  
  
　　  
　　3、位图资源的缩放  
　　当对位图资源进行缩放处理时，充分考虑设备的高度和宽度，在程序运行时获取设备的高度和宽度代码如下：  
　　1 2 3 4  
　　WindowManager mWMgrmWMgr = (WindowManager)context.getSystemService(Context.WINDOW\_SERVICE);int width = mWMgr.getDefaultDisplay().getWidth();  
　　int height – mWMgr.getDefaultDisplay().getHeight();  
　　下面列举一些强烈推荐的建议：  
　　在XML布局，使用wrap\_content和fill\_parent来填充整个父窗口；  
　　使用FrameLayout的，而不是AbsoluteLayout，减少界面布局对屏幕大小的依赖；  
　　NEVER use hard-coding for pixel value, use dip (density independent pixel)；  
　　根据density和resolution 为不同的设备准备合适的图片资源。

http://bbs.tianya.cn/post-949-547-1.shtml