**Android 开发中的屏幕适配详解**

1. 为什么要屏幕适配

Android设备非常多，设备屏幕的碎片化，而且各种设备的尺寸和分辨率都不太一样，这就导致了开发时候需要考虑到对不同设备的屏幕适配问题。

1. 核心概念和相关单位详解
2. 屏幕的尺寸、分辨率、像素密度

屏幕尺寸：屏幕的对角线长度，单位是英寸，1英寸约等于2.54厘米

屏幕分辨率：指横纵向上的像素点，单位是px，1px=1个像素点，手机中一般以纵像素点\*横像素点，如1920\*1080

屏幕的像素密度：指每英寸上的像素点数，单位是dpi，即“dots per inch”的缩写，由此可知屏幕像素密度与屏幕尺寸和分辨率有关。下面以5英寸屏幕，分辨率为1920\*1080来计算其dpi为 440DPI，这个440dpi怎么算出来的呢？其实很简单

就是 首先计算出屏幕对角线上的像素点数：1920^2+1080^2=4852800，然后对4852800进行开平方得出2202.907170082，最后再除以5英寸，即 2202.907170082/5=440dpi。

1. px和dp

px就是 像素，1px=1个像素点

dp也叫dip，即device independent pixels，即与设备无关像素。在屏幕密度为160的显示屏上，1dip=1px。

1. SP

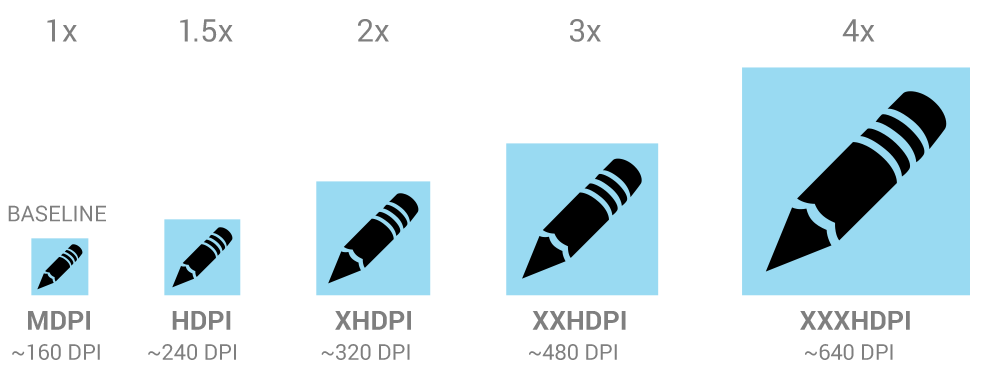
即Scale-Independent Pixels，可以根据文字大小首选项进行缩放。一般用于设置文字大小的单位。

1. 4 种dpi-mdpi、hdpi，xdpi，xxdpi。

下图就是对四种dpi的定义范围。



在开发中，需要对这四种dpi适配各自对应的图标大小。由于所有图标都需要适配不同的像素密度，这里引入 dp 这一单位，它以中等尺寸，MDPI的屏幕为基准提供像素密度无关的表示。



在设计图标时，对于五种主流的像素密度（MDPI、HDPI、XHDPI、XXHDPI 和 XXXHDPI）应按照 **2:3:4:6:8 的比例进行缩放**。例如，一个启动图标的尺寸为48x48 dp，这表示在 MDPI 的屏幕上其实际尺寸应为 48x48 px，在 HDPI 的屏幕上其实际大小是 MDPI 的 1.5 倍 (72x72 px)，在 XDPI 的屏幕上其实际大小是 MDPI 的 2 倍 (96x96 px)，依此类推。

1. 屏幕适配解决方法-适配各种屏幕尺寸

要适配各种屏幕尺寸，主要通过开发编码编程来适配大部分的不同分辨率尺寸。主要需要做以下几个方面的工作：

1. 多使用wrap\_content,match\_parent，weight（weight这个只能在LinearLayout中使用哦）

对了 ，这里顺便讲下 weight是怎么计算的，以宽度为例：其实weight计算出的宽度 = 自身原来宽度+剩余空间所占百分比的宽度，这就是为什么用到weight属性的时候经常把layout\_widht=”0dp” 了。

（2）多使用相对布局，尽量不用绝对布局

（3）推荐使用限定符

尺寸限定符（适配Android 3.2之前版本）：large限定符，比如，如果需要对大小为large的屏幕提供支持，需要在res目录下新建一个文件夹layout-large/并提供layout。

最小宽度限定符（适配Android3.2之后版本）：sw最小宽度限定符号，和上面的使用方法一样，如layout-sw600dp，然后提供对应的layout

屏幕方向限定符：和其他限定符一样，如：res/values-land/layouts.xml

（4）多使用自动拉伸位图，即点9图片

1. 屏幕适配解决方法-适配各种屏幕密度

采用这种方法，主要通过以下两种方法来达到适配目的：

1. 使用非密度制约像素，使用px为单位，这种与google提倡的dp为单位相背离。所以不太提倡。
2. 提供备用位图：（Android studio中：mipmap-mdpi、mipmap-hdpi、mipmap-xdpi、mipmap-xxdpi、mipmap-xxdpi）；（Eclipse ADT中：drawable-mdpi、drawable -hdpi、drawable -xdpi、drawable –xxdpi、drawable-xxxdpi）。最好对不同dpi切对应的分辨率的图，这样可以避免缩放带来的内存开销变大的问题。
3. 屏幕适配解决方法-实施自适应用户界面流程

该辅助方法主要是通过代码编程中根据布局以及屏幕配置的变化来做出适应各自的界面流程。

1. 确定当前布局，是手机布局还是平板布局等
2. 根据当前布局来做出响应操作
3. 重复使用其他活动中的片段
4. 处理屏幕配置的变化，比如横纵向的变化。