**Android系统图片和字体**

我们在做Android移动APP设计的时候，字号的选择也是很让人头疼，今天整理了一份有关Android系统字体规范，主要从以下几个方面：

1. 移动设计中与字号有关的基本概念

* px：Pixels即像素，基本原色素及其灰度的基本编码。
* DPI：dots per inch，是印刷上的记量单位，意思是每个英寸上，所能印刷的网点数。
* PPI：pixels per inch 数字影像的解析度，意思是每英寸所拥有的像素数，即像素密度。
* PPI = √（长度像素数² + 宽度像素数²） / 屏幕对角线英寸数
* pt：Point，磅因，国际通行的印刷单位，是一个自然界标准的长度单位。

inch = 2.54cm = 25.4 mm = 72pt， 1pt≈0.35mm  
Photoshop里的字体单位pt和point不一样。在Photoshop里，同样是10pt的字，只要变换字体，字的高度就会变化。

* dp：Density-independent pixels，以160PPI屏幕为标准，则1dp=1px
* dp和px的换算公式 ：dp\*ppi/160 = px
* sp：Scale-independent pixels.安卓的字体单位，以160PPI屏幕为标准，当字体大小为 100%时， 1sp=1px
* sp 与 px 的换算公式：sp\*ppi/160 = px

2. Android 系统默认字体 Rotobo,限定使用以下字号



3. Android 字体单位 sp 与 px 的换算

* PPI = √（长度像素数2 + 宽度像素数2） / 屏幕对角线英寸数
* sp与px的换算公式：sp\*ppi/160 = px

以三星Note2为例，PPI = √（12802 + 7202） / 5.5=267  


经测量，三星Note2的字体高度确实如此，见下图：  


4. 规范字号的意义

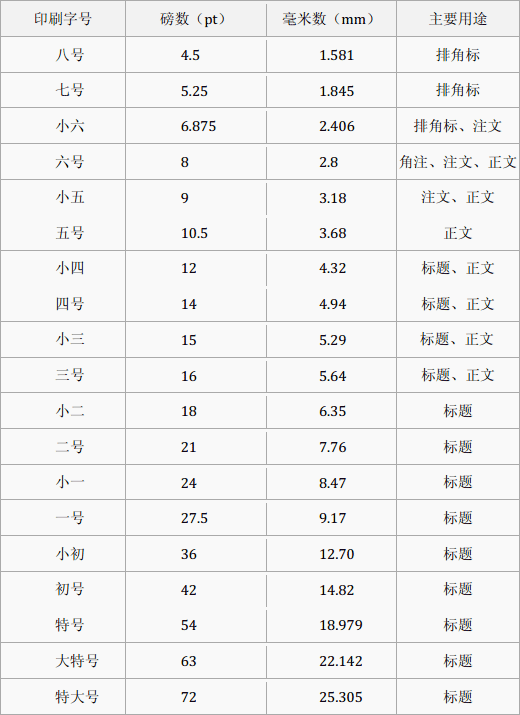
保证相同字号的字在不同PPI屏幕上显示的物理高度一致，下面来验证一下。

* 设定 PPI为 267 时，高度18sp（即 30px）的字的物理高度为 X，则：  
  一英寸里的像素数 267 / 一英寸里的毫米数 25.4mm = 30 / X  
  X ≈ 2.86mm
* 设定 PPI为 160 时，高度18sp（即 18px）的字的物理高度为 Y，则：  
  一英寸里的像素数 160 / 一英寸里的毫米数 25.4mm = 18 / Y  
  Y ≈ 2.86mm

综上，X = Y，这就证明了同为18sp的字在 PPI分别为 267 和 160 时，显示的物理高度都是 2.86mm

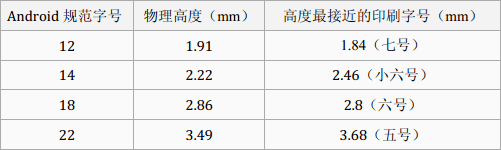
5. 印刷业的规范字号的用法

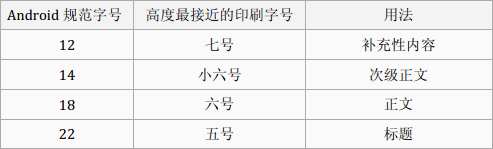
看书的时候眼睛距离书本的最佳距离为 33cm，书本的阅读距离比较接近于手机的阅读距离，所以我们来参考下书本上印刷文字的规范。

印刷业已经有成熟的行业字号标准，如下表所示：  


其中正文常用的字号是六号（7.5p）、小五号（9p）、五号（10.5p），如下表所示：  


6. Android 规范字号的近似用法

通过计算出 Android规范字号的物理高度，找到每个字号最接近的印刷字号：  


再根据印刷字号的用途，近似得出Android 规范字号的用法  


7. 如何向前端输出？

第一步：将sp 换算成px。但是px随PPI 变化而变化，这一点可以从sp 与px的换算公式“sp\*ppi/160 = px ”看出来。我们不能算出所有的情况，所以只计算首选需要适配的ppi 对应的像素高度。

第二步：把算好的像素高度和换算公式“ sp \* ppi/160 = px ”同时输出给前端，这样以后再换ppi，前端可以自动计算。另外，前端代码里定义字体高度用的也是px，所以设计师向前端输出以px计算的字号尺寸是非常合适的。例如：  


8. 如何在photoshop里选择字号？

上文中，我们已经算出了字体像素高度，把这些像素高度画到ps 里，  
再把你要使用的字体调到这样的像素高度，得出photoshop 里对应的字号，用于设计。也就是说Photoshop 里的字号都需要根据实际情况，手动调出来，没有捷径。例如ppi 为240 时，对应的像素高度和字体字号如下图所示：  


如果ppi 变化，上面的图就不能用了，又要重新算，比较劳神费力，但我目前还没有更好的办法，欢迎大家拍砖，我会积极采纳意见，优化方案。

9. 常见机型与其对应的参数

PPI = √（长度像素数² + 宽度像素数²） / 屏幕对角线英寸数

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 机型 | 长(px) | 宽(px) | 大小(英寸) | PPI | Android大小 |
|  | 800 | 480 | 4.0 | 233 | hdpi |
| 三星S3 | 1280 | 720 | 4.8 | 305 | xhdpi |
| 三星S4 | 1920 | 1080 | 5 | 461 | xxhdpi |
| 三星S5 | 1920 | 1080 | 5.1 | 452 | xxhdpi |
| 三星Note2 | 1280 | 720 | 5.5 | 267 | hdpi |
| 三星Note3 | 1920 | 1080 | 5.7 | 404 | xxhdpi |
| HTC one | 1920 | 1080 | 4.3 | 536 | xxhdpi |

sp\*ppi/160 = px

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 名称 | Android规范字号（SP） | mdpi  （PPI为160时PX） | hdpi  （PPI为240时PX） | xhdpi  （PPI为320时PX） | xxhdpi  （PPI为480时PX） | xxxhdpi  （PPI为640时PX） |
| Micro | 12 | 12 | 18 | 24 | 36 | 48 |
| Small | 14 | 14 | 21 | 28 | 42 | 56 |
| Medium | 18 | 18 | 27 | 36 | 54 | 72 |
| Large | 22 | 22 | 33 | 44 | 66 | 88 |

10. 图片与dp/px关系

Android中支持许多资源，包括图片（Bitmap），对应于bitmap的文件夹是drawable，除了drawable，还有drawable-ldpi、drawable-mdpi、drawable-hdpi、drawable-xhdpi、drawable-xxhdpi等，同一张图片放到上面不同的文件夹中是有区别的，比如一张100 \* 100像素大小的图片，分别放在上述各个文件夹中，然后将其设置为ImageView（假设宽高都是wrap\_content）的图片，那么这个ImageView的大小是多少呢，或者说图片的大小是多少呢？本文主要和大家阐述这个问题。当然，还有一个问题，如果在上述各个文件夹中都放置一张相同的图片（内容相同，尺寸不同），那么系统会选择加载哪一张图片呢？这个问题，系统有内部的选择机制，简单来说：系统会选择最接近手机屏幕密度的图片，由于这个问题和本文主题关系不是很大，所以暂时不讨论此类问题。

先介绍一些概念:

屏幕密度：单位英寸面积上的像素点数

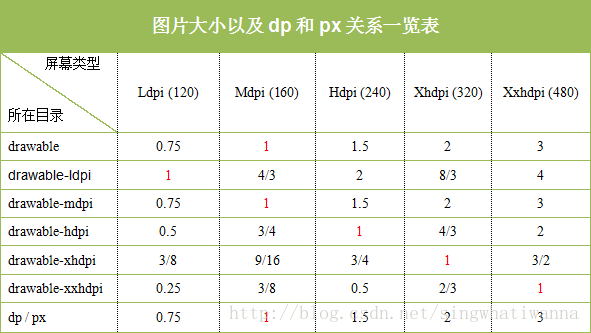
ldpi：   屏幕密度为120的手机设备

mdpi： 屏幕密度为160的手机设备（此为baseline，其他均以此为基准，在此设备上，1dp = 1px）

hdpi：  屏幕密度为240的手机设备

xhdpi： 屏幕密度为320的手机设备

xxhdpi：屏幕密度为480的手机设备



注意点：

1. 图片放在drawable中，等同于放在drawable-mdpi中，原因为：drawable目录不具有屏幕密度特性，所以采用基准值，即mdpi

2. 图片放在某个特定drawable中，比如drawable-hdpi，如果设备的屏幕密度高于当前drawable目录所代表的密度，则图片会被放大，否则会被缩小

　　放大或缩小比例 = 设备屏幕密度 / drawable目录所代表的屏幕密度

3. 为了更全面的适配所有设备，我们应该提供一套针对主流屏幕密度的图片（目前为hdpi或xhdpi），其他密度通过系统自动缩放得到图片