

# 鲤刷刷屏幕适配简析

## 前言

- 为什么要进行Android屏幕适配?

由于Android系统的开放性，任何用户、开发者、OEM厂商、运营商都可以对Android进行定制，于是导致：

- Android系统碎片化：小米定制的MIUI、魅族定制的flyme、华为定制的EMUI等等当然都是基于Google原生系统定制的；
- Android机型屏幕尺寸碎片化：5寸、5.5寸、6寸等等
- Android屏幕分辨率碎片化：320x480、480x800、720x1280、1080x1920

据友盟指数显示，统计至2015年12月，支持Android的设备共有27796种；

当Android系统、屏幕尺寸、屏幕密度出现碎片化的时候，就容易出现同一元素在不同手机上显示不同的问题。

### 试想一下这么一个场景：

为4.3寸屏幕准备的UI设计图，运行在5.0寸的屏幕上，很可能在右侧和下侧存在大量的空白；而5.0寸的UI设计图运行到4.3寸的设备上，很可能显示不下。

为了保证用户获得一致的用户体验效果:使得某一元素在Android不同尺寸、不同分辨率的手机上具备相同的显示效果，于是，我们便需要对Android屏幕进行适配。

- 上次Android组开会中讨论到鲤刷刷屏幕适配的问题，会上听的一知半解，然后就花了点时间学习了一下鲤刷刷中所用到的屏幕适配方案：
  - 今日头条的适配方案；
  - dimens.xml中px和dp转化（由于我们的UI设计图都是给的px值，Android开发要将其转化为dp值）；

## 分析：

先讲上面第二种，所说的屏幕适配，我认为其实并不是真正的屏幕适配，仅仅是给了一个正确的dp值，但是在不同尺寸屏幕的手机上还是存在没有适配的问题，其原因在于：**不同尺寸的手机的宽和高的总的dp值是不一致的；**

## 接下来讲讲手机的宽高的dp值是如何计算的？

- 恶补基础知识

### 常见的单位

- px、pt、ppi、dpi、dp、sp

### 定义

- px：pixel，像素，电子屏幕上组成一幅图画或照片的最基本单元；

- **pt**: point, 点, 印刷行业常用单位, 等于1/72英寸;
- **ppi**: pixel per inch, 每英寸像素数, 该值越高, 则屏幕越细腻;
- **dpi**: dot per inch, 每英寸多少点, 该值越高, 则图片越细腻;
- **dp**: dip, Density-independent pixel, 是安卓开发用的长度单位, 1dp表示在屏幕像素点密度为160ppi时1px长度;
- **sp**: scale-independent pixel, 安卓开发用的字体大小单位, 可随系统字体改变大小;

## 换算公式

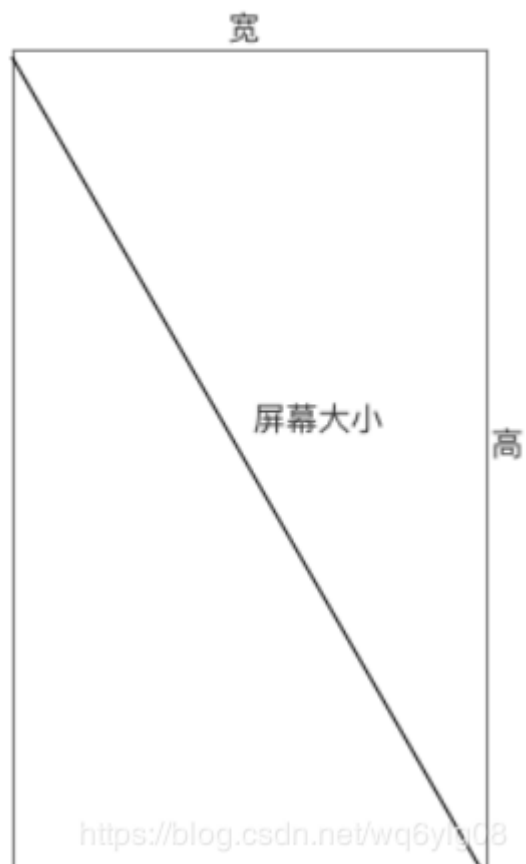
- $1\text{pt} = (\text{DPI} / 72) \text{ px}$
- $\text{dpi} = \text{ppi}$
- 当文字尺寸是“正常”时  $1\text{sp} = 1\text{dp}$ , 而当文字尺寸是“大”或“超大”时,  $1\text{sp} > 1\text{dp}$ 。  
一般情况下可认为  $\text{sp} = \text{dp}$ 。(Android中字体设置为  $\text{sp}$  时, 可根据系统设定变化);
- $\text{dp} * \text{dpi} / 160 = \text{px}$  比如  $1\text{dp} \times 320\text{ppi} / 160 = 2\text{px}$
- $\text{dp} = \text{px} / \text{density}$
- $\text{density} = \text{dpi} / 160$  (由上两式可得)

屏幕尺寸(英寸 **in**)、分辨率(**px**)、像素密度(**dpi**)三者关系

$$\text{密度 (单位/dpi)} = \frac{\sqrt{(\text{宽}^2 + \text{高}^2)} \text{ 单位 (/px)}}{\text{屏幕大小} \text{ 单位 (/inch)}}$$

### 讲解

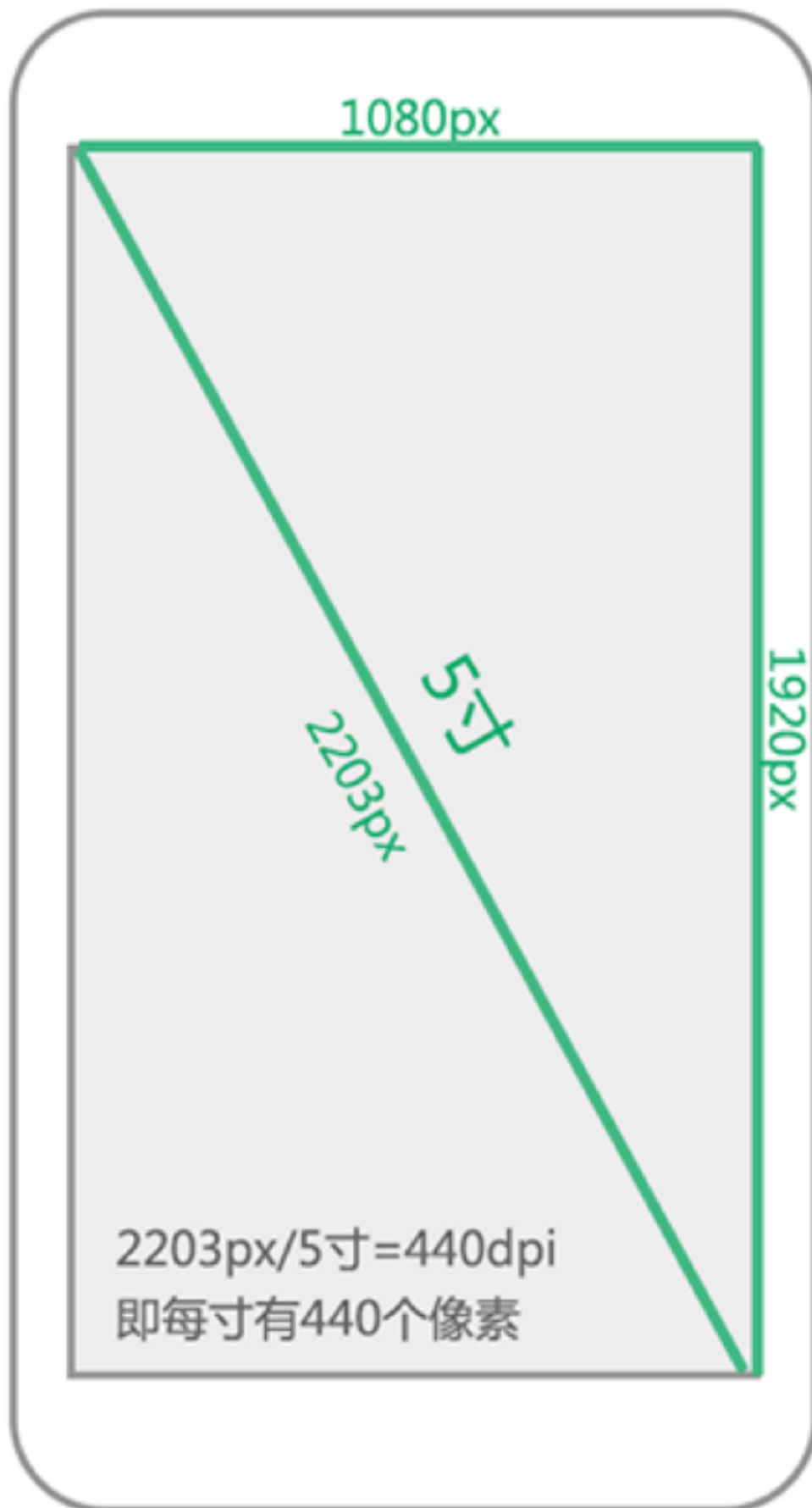
1. 密度即每英寸的像素点
2. 勾股定理求出手机的对角线物理尺寸
3. 再除以屏幕大小即可



【附】:  $1\text{in} = 2.54\text{cm}$

假设一部手机的分辨率是  $1080 \times 1920(\text{px})$ , 屏幕尺寸大小是  $5\text{in}$ , 问屏幕密度 (**dpi**) 是多少?

直接套上方公式



<https://blog.csdn.net/wq6ylg08>

然后 dpi 已知, px 已知, 则根据公式:  $dp * dpi / 160 = px$ , 则 dp 可得;

由于不同手机屏幕的尺寸（英寸 in）和屏幕分辨率不同（宽高 px），导致不同手机（Android）的 dpi 不同，因此不同手机的总 dp 不一致，（大概由于市场上现在的屏幕尺寸和分辨率，大差不差，所以 dpi 差不多，所以 dp 也差不多），但一个产品要适配各种手机或Android设备时，还是需要屏幕适配的，接下来讲讲今日头条的屏幕适配方案；

## 今日头条屏幕适配方案

- 核心

$$dp = px / density$$

### 适配方式

今日头条适配方案默认项目中只能以高或宽中的一个作为基准，进行适配，为什么不像 **AndroidAutoLayout** 一样，高以高为基准，宽以宽为基准，同时进行适配呢？

这就引出了一个现在比较棘手的问题，大部分市面上的 **\*\*Android\*\*** 设备的屏幕高宽比都不一致，特别是现在大量全面屏的问世，这个问题更加严重，不同厂商推出的全面屏手机的屏幕高宽比都可能不一致，这时我们只以高或宽其中的一个作为基准进行适配，就会有效的避免布局在高宽比不一致的屏幕上出现变形的问题

明白这个后，我再来说说 **density**，**density** 在每个设备上都是固定的， $DPI / 160 = density$ ，**屏幕的总 px 宽度 / density = 屏幕的总 dp 宽度**

- 设备 1，屏幕宽度为 1080px，480DPI，屏幕总 dp 宽度为  $1080 / (480 / 160) = 360dp$
- 设备 2，屏幕宽度为 1440，560DPI，屏幕总 dp 宽度为  $1440 / (560 / 160) = 411dp$

可以看到屏幕的总 dp 宽度在不同的设备上是会变化的，但是我们在布局中填写的 dp 值却是固定不变的

这会导致什么呢？假设我们布局中有一个 **View** 的宽度为 100dp，在设备 1 中该 **View** 的宽度占整个屏幕宽度的 27.8% ( $100 / 360 = 0.278$ )

但在设备 2 中该 **View** 的宽度就只能占整个屏幕宽度的 24.3% ( $100 / 411 = 0.243$ )，可以看到这个 **View** 在像素越高的屏幕上，dp 值虽然没变，但是与屏幕的实际比例却发生了较大的变化，所以肉眼的观看效果，会越来越小，这就导致了传统的填写 dp 的屏幕适配方式产生了较大的误差

这时我们要想完美适配，那就必须保证这个 **View** 在任何分辨率的屏幕上，与屏幕的比例都是相同的

这时我们该怎么做呢？改变每个 **View** 的 dp 值？不现实，在每个设备上都要通过代码动态计算 **View** 的 dp 值，工作量太大

如果每个 **View** 的 dp 值是固定不变的，那我们只要保证每个设备的屏幕总 dp 宽度不变，就能保证每个 **View** 在所有分辨率的屏幕上与屏幕的比例都保持不变，从而完成等比例适配，并且这个屏幕总 dp 宽度如果还能保证和设计图的宽度一致的话，那我们在布局时就可以直接按照设计图上的尺寸填写 dp 值

$$\text{屏幕的总 px 宽度} / density = \text{屏幕的总 dp 宽度}$$

在这个公式中我们要保证 **屏幕的总 dp 宽度** 和 **设计图总宽度** 一致，并且在所有分辨率的屏幕上都保持不变，我们需要怎么做呢？**屏幕的总 px 宽度** 每个设备都不一致，这个值是肯定会变化的，这时今日头条的公式就派上用场了

$$\text{当前设备屏幕总宽度 (单位为像素)} / \text{设计图总宽度 (单位为 dp)} = density$$

这个公式就是把上面公式中的 **屏幕的总 dp 宽度** 换成 **设计图总宽度**，原理都是一样的，只要 **density** 根据不同的设备进行实时计算并作出改变，就能保证 **设计图总宽度** 不变，也就完成了适配；