

CoderGhui

学习 | 分享 | 进步

🏠 首页

📁 归档

💬 留言

👤 关于

对于Android中ImageView的ScaleType你的理解可能是错的

📅 Feb 25, 2017 | 📁 技术 | 🏆 Hits

💬 0 Comments

当ImageView的大小与它内容的大小不一致时，就出现了一个问题，里面的内容应该怎么展示？放大、缩小、靠上、靠下、居中、居右...针对这个问题ImageView提供了ScaleType属性来控制。

ImageView一共有8种缩放规则，有些规则特别容易搞混，在选择时经常会懵圈。网上能搜索到解释很多要么模棱两可要么根本就是错误的，我这里直接通过一个Demo展示一下各种ScaleType到底有什么不同，最后再从源码的层面上去分析为什么是这样。

8种缩放规则

ImageView一共有：

1. MATRIX
2. FIT_XY
3. FIT_START
4. FIT_CENTER
5. FIT_END
6. CENTER
7. CENTER_CROP
8. CENTER_INSID

文章目录

1. 8种缩放规则
2. 详细解释
 - 2.1. FIT_XY
 - 2.2. FIT_START
 - 2.3. FIT_CENTER
 - 2.4. FIT_END
 - 2.5. CENTER
 - 2.6. CENTER_CROP
 - 2.7. CENTER_INSID
 - 2.8. MATRIX
3. 源码

8种缩放规则。下面一个一个介绍，不过MATRIX这种缩放方式放到最后再介绍（理论上它能实现任何你想要的缩放效果）。

为了更清楚的对比各种ScaleType的区别，我这里使用了三种比例大小不同的图片，原图如下：



3张图片1张极长，1张极宽，1张宽高差不太多但极小。每张图片在4个边上都有红色的边框以方便对比图片的缩放是否超出了边界。

详细解释

FIT_XY

这种方式最好理解.

在X、Y方向上同时拉伸以填满ImageView控件的大小。效果和把图片直接设为ImageView的background一样。

真实效果如下图所示：



图片都完整显示了(红色边框都显示出来了，只不过在x,y方向上拉伸的幅度不一样，所以会显得X、Y方向上的边框粗细不一样)

FIT_START

这种方式用的应该不是很多，但网上关于这种方式的解释基本都是错误的：~~FIT_START是置于顶部~~首先这种解释没有说明：图片内容到底会不会缩放。然后，说内容是置于顶部并不准确。完整且正确的解释如下：

1. 图片内容会缩放，长宽同时按比例缩放，直到其中一个方向上先撑满 **ImageView** 控件。
2. 若在 **X** 方向上先撑满，则图片内容会居上显示；若在 **Y** 方向上先撑满则会在居左显示。

真实效果如下图所示：



FIT_CENTER

这种方式是 **ImageView** 的默认缩放方式，网上关于这种方式的解释也是不准确的，正确的解释如下：

1. 图片内容会缩放，长宽同时按比例缩放，直到其中一个方向上先撑满 **ImageView** 控件。
2. 缩放完毕后，图片内容会被移动到 **ImageView** 的正中心（图片的中心与 **ImageView** 的中心对齐）

真实效果如下图所示：



FIT_END

和FIT_START一样，网上关于这种方式的解释也是错误的。正确的解释如下：

1. 图片内容会缩放，长宽同时按比例缩放，直到其中一个方向上先撑满 **ImageView** 控件。
2. 若在 **X** 方向上先撑满，则图片内容会居下显示；若在 **Y** 方向上先撑满则会在居右显示。

真实效果如下图所示：



CENTER

这种方式很简单，网上的解释基本都没什么问题。完整的解释如下：

1. 图片内容不会缩放
2. 图片内容居中显示（图片的中心与 **ImageView** 的中心对齐），若图片过大会出现图片周边区域超出 **ImageView** 的范围

真实效果如下图所示：



CENTER_CROP

这种方式，从名字上并不能完全清楚它所有的规则，完整的解释如下：

1. 图片内容会缩放，长宽同时按比例缩放，直到最后一个方向上撑满 **ImageView** 控件为止。
2. 图片内容居中显示（图片的中心与 **ImageView** 的中心对齐），图片的周边可能会被裁剪

真实效果如下图所示：



可以看到，对于第1张图片，其在X方向最后撑满，后两张图片在Y方向上最后撑满。周边的区域被裁剪了。

CENTER_INSID

这种缩放方式很容易与默认的缩放方式(**FIT_CENTER**)搞混，这是因为除非你的图片足够小，否则你看不出它们的区别。完整的解释如下：

1. 图片内容不一定会缩放，只有在图片长或宽大于 **ImageView** 控件的长或宽时才会缩放（而且是缩小）
2. 缩放完成后，图片会居中显示

真实效果如下图所示：



可以看到第2张小图并未缩放。

MATRIX

上面7种缩放方式是系统内置的，若上面这些缩放方式并不能满足你的需求，**MATRIX**这种方式就派上用场了。简单说来这种缩放类型，支持你自定义图片的缩放的规则。

假设你有这样一种缩放需求：图片在**X**轴方向上撑满**ImageView**控件。在**Y**轴方向上置于顶部。效果上如下：



而这张图的原图是这样的：



至于Y轴方向上的底部可能被图片填充，也可能不被，这就看图片够不够高了。

那么如何实现这种效果呢？

1. 第1步把ImageView的ScaleType设置为MATRIX

2. 自定义一个ImageView，重写其中的setFrame方法

关键代码如下：

```
@Override
protected boolean setFrame(int l, int t, int r, int b) {
    boolean changed = super.setFrame(l, t, r, b);
    if (getScaleType() == ScaleType.MATRIX) transformMatrix();
    return changed;
}

private void transformMatrix() {
    Matrix matrix = getImageMatrix();
    float w = getWidth();
    float cw = getDrawable().getIntrinsicWidth();
    float widthScaleFactor = w / cw;
    matrix.setScale(widthScaleFactor, widthScaleFactor, 0, 0);
    setImageMatrix(matrix);
}
```

1. 首先，我们计算出宽度撑满需要的缩放比例：

```
1 | float widthScaleFactor = w / cw;
```

2. 然后对matrix进行设置，改变其Scale值。

```
1 | matrix.setScale(widthScaleFactor, widthScaleFactor, 0, 0);
```

这里只介绍了宽度撑满并居上这一种情况，另外还有3种情况：宽度撑满并居下、高度撑满居左、高度撑满居右，如何实现及ScaleType源码的解读限于篇幅的原因我放在[下一篇](#)中介绍。

源码

ImageScaleTutorial

个人微信公众号已开通：CoderGhui, 欢迎关注！



版权声明

文章版权归本人所有，如需转载需在明显位置处保留作者信息及原文链接！

Android

分享到

Android Scaletype源码分析及自定义Matrix缩放规则

iShanbay(如果你也是一个在用Alfred的扇贝网用户) ▶

0条评论 ghui.me

1 登录 ▾

推荐 分享

评分最高 ▾



开始讨论...

通过以下方式登录

或注册一个 DISQUS 帐号 ?

姓名

来做第一个留言的人吧！

在 GHUI.ME 上还有

如何通过Gradle实现一套代码开发不同特性的APK

4条评论 • 2年前

石心 — 为什么我运行只有一种！而且上面的ImageVlewer下面有一个红色的j是什么意思，貌似没有办法运行！

iOS实现自定义绘图的几种方式

1条评论 • 2年前

Victor — 总结的很条统很好啊！ 赞

Fruit - 让你像解析Json一样解析Html

2条评论 • 10个月前

ghui Zhang — 可以这么理解，不限于Android

用好你的Control键(针对类Unix系统用户)

2条评论 • 2年前

ghui Zhang — 是的