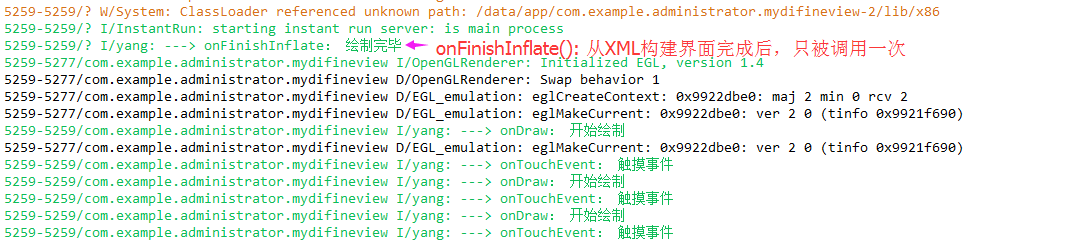
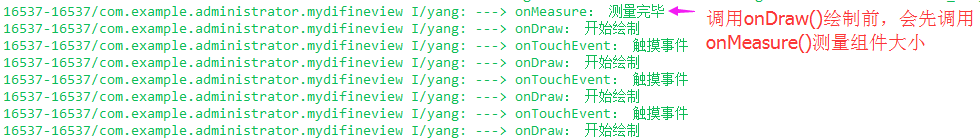
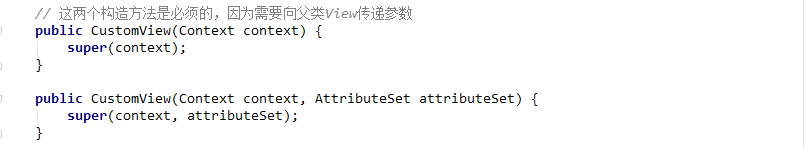
几个方法的调用顺序：







自定义View大部分时候只需重写两个函数：onMeasure()、onDraw()。onMeasure()负责对当前View的尺寸进行测量，onDraw负责将View绘制出来。当然了，至少还得写2个构造函数：



onMeasure():

自定义的View首先要测量宽高尺寸。为什么要测量宽高尺寸？虽然在xml文件中已经指定好了宽高尺寸了，但是在xml布局文件中，属性layout\_width和layout\_height可以不写具体的尺寸，而是wrap\_content或match\_parent。这两个设置值并没有指定真正的大小，但绘制到屏幕上的View必须是要有具体的宽高值。正是因为这个原因，开发者必须自己去处理和设置尺寸。当然了，View类给了默认的处理，但是如果View类的默认处理不满足要求，就得重写onMeasure()函数。举个例子，如果希望View是个正方形，若在xml中指定宽高为wrap\_content，且使用View类提供的measure处理方式，就无法满足需求(如图1)。

onMeasure()函数原型：

protected void onMeasure(int widthMeasureSpec, int heightMeasureSpec)

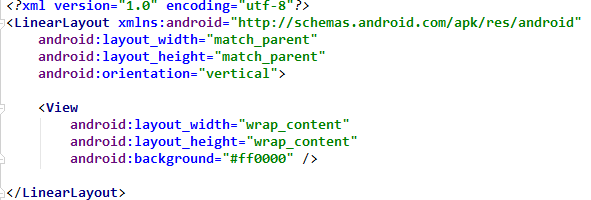
=>

图1

两个参数widthMeasureSpec和heightMeasureSpec包含了宽和高的信息、以及测量模式，也就是说，一个int整数，里面放了测量模式和尺寸大小。那么一个数怎么放两个信息呢？开发者在设置宽高时有3个选择：wrap\_content、match\_parent以及指定固定尺寸，而测量模式也有3种：UNSPECIFIED，EXACTLY，AT\_MOST。如果使用二进制，只需要使用2个bit就可以指定测量模式，因为2个bit取值范围是[0, 3]里面可以存放4种值。int型数据占32bit，将int数据的前2个bit用于区分不同的测量模式，后面30个bit存放的是尺寸数据。

怎么从int数据中提取测量模式和尺寸？不用开发者每次都要写一次移位<<和取且&操作，Android内置类MeasureSpec可以完成上述任务：

int widthMode = MeasureSpec.getMode(widthMeasureSpec);

int widthSize = MeasureSpec.getSize(widthMeasureSpec);

既然能通过widthMeasureSpec拿到宽度尺寸大小，还需要测量模式干嘛？注意：这里的尺寸大小并不是View最终的尺寸大小，而是父View提供的参考大小。测量模式含义如下：

测量模式 表示意思

UNSPECIFIED 父容器没有对当前View有任何限制，当前View可以任意取值

EXACTLY 当前尺寸 (widthSize) 就是当前View应该取的尺寸

AT\_MOST 当前尺寸 (widthSize) 是当前View能取的最大值

而上面的测量模式跟wrap\_content、match\_parent以及写成固定尺寸有什么对应关系呢？

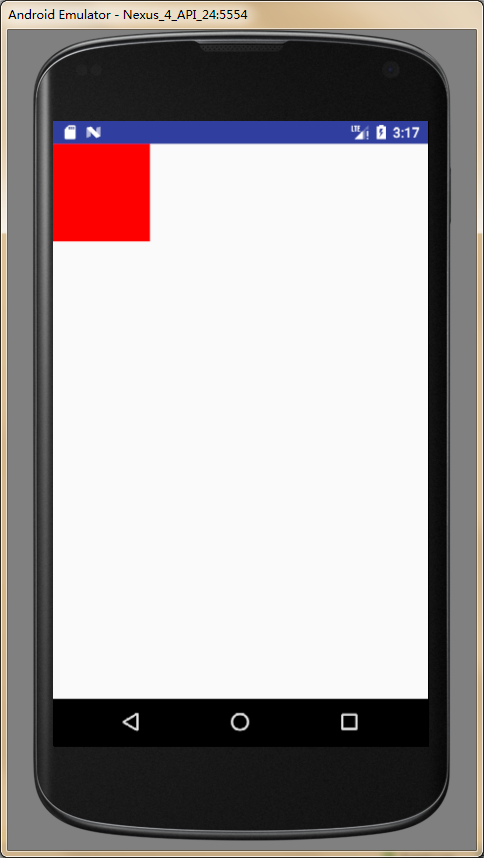
match\_parent—>EXACTLY：match\_parent就是要利用父View给我们提供的所有剩余空间，而父View剩余空间是确定的。

wrap\_content—>AT\_MOST：就是我们想要将大小设置为包裹我们的view内容，那么尺寸大小就是父View给我们作为参考的尺寸，只要不超过这个尺寸就可以

固定尺寸（如100dp）—>EXACTLY：用户自己指定了尺寸大小

自定义onMeasure()函数：

实现效果为：将当前的View以正方形的形式显示，即要宽高相等，并且默认的宽高值为100像素(如图2)

==>

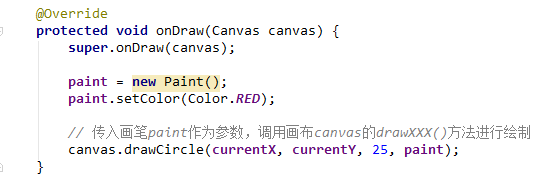
若去掉上述代码，则为：

==>

图2

onDraw():

onMeasure()负责测量View的大小，onDraw负责在此View上绘制

==>

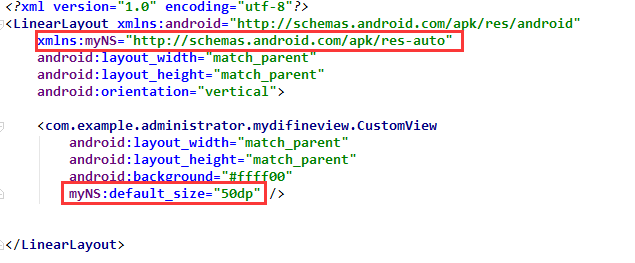
自定义View属性：

如果有些属性希望由用户指定，只有当用户不指定的时候才使用默认值，比如上面的默认尺寸。如果想要由用户自己在布局文件里面指定属性值就需要定义自定义属性。

首先需要在res/values/styles.xml文件（没有则新建）里面声明一个自定义的属性集：



在xml文件中使用自定义属性：



需要在根标签（LinearLayout）里面设定命名空间，命名空间名称可以随便取，比如myNS，命名空间后面取得值是固定的："http://schemas.android.com/apk/res-auto"

