<shape>和<selector>在UI设计中经常用到。比如要自定义一个圆角Button，点击Button时效果需要变化，就要用到<shape>和<selector>。

**1.Shape**

作用：在XML中定义几何形状

位置：res/drawable/

使用的方法：

Java代码中：R.drawable.文件的名称

XML中：android:**background**="@drawable/文件的名称"

属性：

**<shape> 　形状**

android:shape=["rectangle" | "oval" | "line" | "ring"]

其中rectangle矩形，oval椭圆，line直线，ring环形

<shape>中子节点的5种属性:

<gradient> 渐变

android:startColor 　起始颜色

android:endColor 　结束颜色

android:angle 渐变角度，0从左到右，90从下到上，数值只能为45的整数倍，默认为0；

android:type 　渐变的样式 liner线性渐变 radial环形渐变

<solid > 填充

android:color 填充的颜色

<stroke >描边

android:width 描边的宽度

android:color 描边的颜色

android:dashWidth 表示'-'横线的宽度

android:dashGap 表示'-'横线之间的距离

<corners >圆角

android:radius 圆角的半径, 值越大角越圆

android:topRightRadius 右上圆角半径

android:bottomLeftRadius 右下圆角半径

android:topLeftRadius 左上圆角半径

android:bottomRightRadius 左下圆角半径

<padding >填充

android:bottom="1.0dip" 底部填充

android:left="1.0dip" 左边填充

android:right="1.0dip" 右边填充

android:top="0.0dip" 上面填充

2.Selector

分为**四大属性**：

android:**state\_selected** 是否选中

android:**state\_focused** 是否获得焦点

android:**state\_pressed** 是否点击

android:**state\_enabled** 是否响应事件,指所有事件

其他还有：  
 android:state\_hovered 光标是否悬停，通常与focused state相同，为4.0的新特性  
 android:state\_checkable 组件是否能被check。如：RadioButton是可以被check的。  
 android:state\_checked 被checked了，如：一个RadioButton可以被check了。  
 android:state\_activated 被激活  
 android:**state\_window\_focused** 应用程序是否在前台，当通知栏被拉下来或者一个对话框弹出的时候应用程序就不在前台了

使用的方法：

Java代码中：R.drawable.文件的名称

XML中：android:**background**="@drawable/文件的名称"

Selector实例：

<?xml version="1.0" encoding="utf-8" ?>

<selector xmlns:Android="http://schemas.android.com/apk/res/android">

<!-- 没有任何状态时为默认时的背景图片-->

<item android:drawable="@drawable/pic1" />

<!-- 不在前台时的背景图片 -->

<item

android:state\_window\_focused="false"

android:drawable="@drawable/pic\_blue"

/>

<!-- 非触摸模式下获得焦点并单击时的背景图片 -->

<item

Android:state\_focused="true"

android:state\_pressed="true"

android:drawable= "@drawable/pic\_red"

/>

<!-- 触摸模式下单击时的背景图片-->

<item

Android:state\_focused="false"

Android:state\_pressed="true"

Android:drawable="@drawable/pic\_pink"

/>

<!--选中时的图片背景-->

<item

Android:state\_selected="true"

android:drawable="@drawable/pic\_orange"

/>

<!--获得焦点时的图片背景-->

<item

Android:state\_focused="true"

Android:drawable="@drawable/pic\_green"

/>

</selector>

３.　layer-list

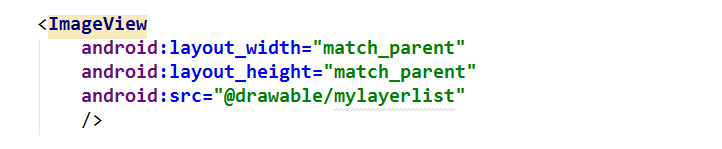
简介：

将多个图片或上面两种效果按照顺序层叠起来，先写的<item>放在下层

例子：



使用方法：



效果为：



以上三个标签可以揉合到一起使用，比如：

