**Android 5**种布局

一、FrameLayout

这个布局可以看成是墙脚堆东西，有一个四方的矩形的左上角墙脚，我们放了第一个东西，要再放一个，那就在放在原来放的位置的上面，这样依次的放，会盖住原来的东西。这个布局比较简单，也只能放一点比较简单的东西。

二、LinearLayout

线性布局，这个东西，从外框上可以理解为一个div，他首先是一个一个从上往下罗列在屏幕上。每一个LinearLayout里面又可分为垂直布局（android:orientation="vertical"）和水平布局（android:orientation="horizontal" ）。当垂直布局时，每一行就只有一个元素，多个元素依次垂直往下；水平布局时，只有一行，每一个元素依次向右排列。

linearLayout中有一个重要的属性 android:layout\_weight="1"，这个weight在垂直布局时，代表行距；水平的时候代表列宽；weight值越大就越大。

三、AbsoluteLayout

绝对布局犹如div指定了absolute属性，用X,Y坐标来指定元素的位置android:layout\_x="20px" android:layout\_y="12px" 这种布局方式也比较简单，但是在垂直随便切换时，往往会出问题，而且多个元素的时候，计算比较麻烦。

四、RelativeLayout

相对布局可以理解为某一个元素为参照物，来定位的布局方式。主要属性有：

相对于某一个元素  
  
android:layout\_below="@id/aaa" 该元素在 id为aaa的下面  
android:layout\_toLeftOf="@id/bbb" 改元素的左边是bbb

相对于父元素的地方  
  
android:layout\_alignParentLeft="true"  在父元素左对齐  
android:layout\_alignParentRight="true" 在父元素右对齐

还可以指定边距等，具体详见API

五.TableLayout

表格布局类似Html里面的Table。每一个TableLayout里面有表格行TableRow，TableRow里面可以具体定义每一个元素，设定他的对齐方式 android:gravity="" 。

各种Layout用到的一些重要的属性：

　　第一类:属性值为true或false

　　android:layout\_centerHrizontal 水平居中

　　android:layout\_centerVertical 垂直居中

　　android:layout\_centerInparent 相对于父元素完全居中

　　android:layout\_alignParentBottom 贴紧父元素的下边缘

　　android:layout\_alignParentLeft 贴紧父元素的左边缘

　　android:layout\_alignParentRight 贴紧父元素的右边缘

　　android:layout\_alignParentTop 贴紧父元素的上边缘

　　android:layout\_alignWithParentIfMissing 如果对应的兄弟元素找不到的话就以父元素做参照物

　　第二类：属性值必须为id的引用名“@id/id-name”

　　android:layout\_below 在某元素的下方

　　android:layout\_above 在某元素的的上方

　　android:layout\_toLeftOf 在某元素的左边

　　android:layout\_toRightOf 在某元素的右边

　　android:layout\_alignTop 本元素的上边缘和某元素的的上边缘对齐

　　android:layout\_alignLeft 本元素的左边缘和某元素的的左边缘对齐

　　android:layout\_alignBottom 本元素的下边缘和某元素的的下边缘对齐

　　android:layout\_alignRight 本元素的右边缘和某元素的的右边缘对齐

　　第三类：属性值为具体的像素值，如30dip，40px

　　android:layout\_marginBottom 离某元素底边缘的距离

　　android:layout\_marginLeft 离某元素左边缘的距离

　　android:layout\_marginRight 离某元素右边缘的距离

　　android:layout\_marginTop 离某元素上边缘的距离

　　EditText的android:hint

　　设置EditText为空时输入框内的提示信息。

　　android:gravity

　　android:gravity属性是对该view 内容的限定.比如一个button 上面的text. 你可以设置该text 在view的靠左，靠右等位置.以button为例，android:gravity=”right”则button上面的文字靠右

　　android:layout\_gravity

　　android:layout\_gravity是用来设置该view相对与起父view 的位置.比如一个button 在linearlayout里，你想把该button放在靠左、靠右等位置就可以通过该属性设置.以button为例，android:layout\_gravity=”right”则button靠右

　　android:layout\_alignParentRight

　　使当前控件的右端和父控件的右端对齐。这里属性值只能为true或false，默认false。

　　android:scaleType：

　　android:scaleType是控制图片如何resized/moved来匹对ImageView的size。ImageView.ScaleType / android:scaleType值的意义区别：

　　CENTER /center 按图片的原来size居中显示，当图片长/宽超过View的长/宽，则截取图片的居中部分显示

　　CENTER\_CROP / centerCrop 按比例扩大图片的size居中显示，使得图片长(宽)等于或大于View的长(宽)

　　CENTER\_INSIDE / centerInside 将图片的内容完整居中显示，通过按比例缩小或原来的size使得图片长/宽等于或小于View的长/宽

　　FIT\_CENTER / fitCenter 把图片按比例扩大/缩小到View的宽度，居中显示

　　FIT\_END / fitEnd 把图片按比例扩大/缩小到View的宽度，显示在View的下部分位置

　　FIT\_START / fitStart 把图片按比例扩大/缩小到View的宽度，显示在View的上部分位置

　　FIT\_XY / fitXY 把图片不按比例扩大/缩小到View的大小显示

　　MATRIX / matrix 用矩阵来绘制，动态缩小放大图片来显示.

布局中**dp**和**px**的区别

px: pixels(像素)，不同的设备不同的显示屏显示效果是相同的，这是绝对像素，是多少就永远是多少不会改变。

dp也是dip: device independent pixels(设备独立像素). 不同设备有不同的显示效果,这个和设备硬件有关，一般我们为了支持WVGA、HVGA和QVGA 推荐使用此单位。

    这里要特别注意dip与屏幕密度有关，而屏幕密度又与具体的硬件有关，硬件设置不正确，有可能导致dip不能正常显示。在屏幕密度为160的显示屏上，1dip=1px，有时候可能你的屏幕分辨率很大如480\*800，但是屏幕密度没有正确设置比如说还是160，那么这个时候凡是使用dip的都会显示异常，基本都是显示过小。