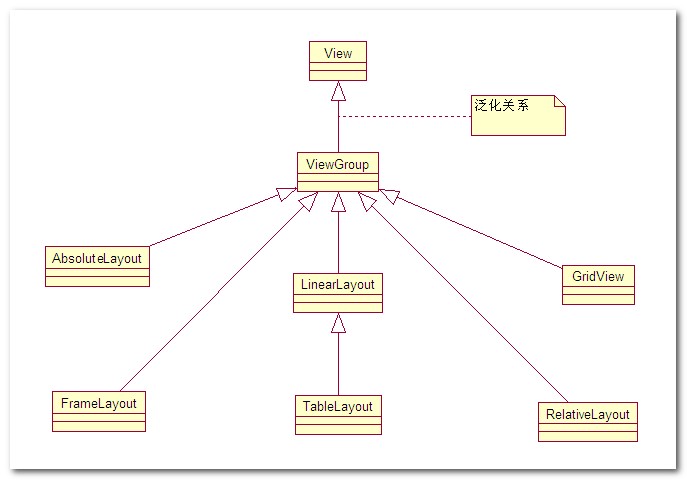
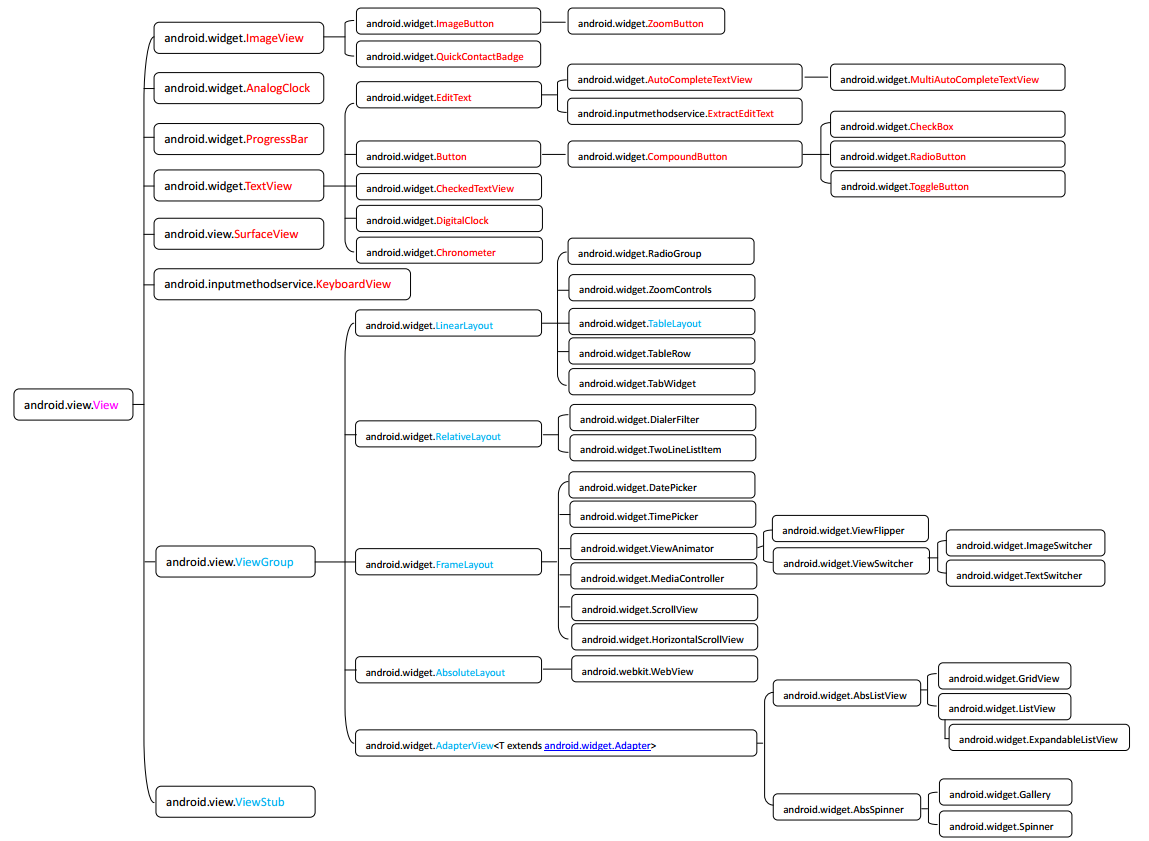
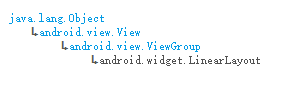
Android中任何可视化的控件都是从android.veiw.View继承而来的，系统提供了两种方法来设置视图：第一种也是我们最常用的的使用XML文件来配置View的相关属性，然后在程序启动时系统根据配置文件来创建相应的View视图。第二种是我们在代码中直接使用相应的类来创建视图。

各种布局与View的继承关系：





一、LinearLayout（线性布局）



1、线性布局的特点：每个LinearLayout里面又可分为垂直布局（由属性android:orientation=”vertical”设置）和水平布局（android:orientation=”horizontal”）。当垂直布局时，每一行只有一个元素，多个元素依次垂直往下；当水平布局时，只有一行，每一个元素依次先右排列，当组件排列到窗口的边缘后，后面的组件将不会被显示出来。

2、线性布局的常用属性

**每一种布局管理器的属性都是被定义在LayoutParams内部类中。**

（1）android:orientation属性：用于设置布局器内部组件排列方式，其可选值为：horizontal和vertical，默认值为vertical。

（2）android:gravity属性：用书设置布局管理器内组件的对齐方式，其可选值包括：top、bottom、left、right、center\_vertical、fill\_vertical、center\_horizontal、fill\_horizontal、center、fill、clip\_certical和clip\_horizontal。这些属性可以同时指定，各属性之间用竖线“|”隔开。

（3）android:layout\_width属性：设置组件的基本宽度，可选值有fill\_oparent、match\_parent、wrap\_content。其中fill\_match表示该组件的宽度与父容器的宽度相同；match\_parent与fill\_parent作用完全相同。wrap\_content表示该组件的宽度恰好能包裹它的内容。

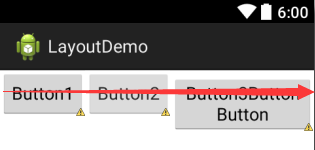
（4）android:layout\_heigth属性：用于设置组件的基本高度，可选值有fill\_oparent、match\_parent、wrap\_content。其中fill\_match表示该组件的高度与父容器的高度相同；match\_parent与fill\_parent作用完全相同。wrap\_content表示该组件的高度恰好能包裹它的内容。

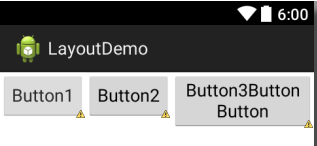
（5）android:id属性：用于为当前组件指定一个id属性，在Java代买中可以应用该属性单独引用这个组件。为组件指定id属性后，在R.java文件中会自动派生一个对应的属性，在Java代码中可以通过方法findViewById()来获取它。

（6）android:background属性：用于设置组件的背景，可以是图片，可以是颜色。若背景设置为图片，则为android:background=”@drawable/background”；若设置为颜色，则为android:background=”#FFFFFFFF”，背景用颜色填充时，格式有：#rgb，#rgba等。

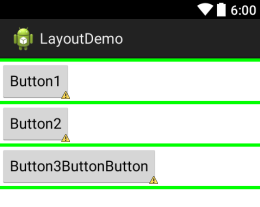
（7）android:baselineAligned属性（基线对齐）：如果该属性为**false**, 就会**阻止**该布局管理器与其子元素的**基线对齐**。

例如：当组件的内容太多时：



红箭头方向表示基线，因为基线对齐是按照第一行文字对齐的。当设置属性值为false时：。

（8） android:divider属性（设分隔条）：设置**垂直布局**时两个按钮之间的分隔条。



要将设置的分割条显示出来，必须使用属性showDividers来设置分隔条显示的位置。其实该属性也可以用于水平布局，但是看起来不太合适。

以上8种属性都是常见针对布局管理器本身而进行设置的属性，下面介绍的属性是针对在该布局中的子组件的布局进行设置。

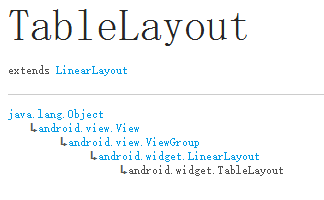
LinearLayout中的组件, **都受到LinearLayout.LayoutParams控制**, 因此LinearLayout包含的子元素可以执行下面的属性。

（1）android:layout\_gravity：指定该元素在LinearLayout(父容器)的对齐方式, 也就是**该组件本身的对齐方式**, 注意要与android:gravity区分。

（2）android:layout\_weight：指定该元素在LinearLayout(父容器)中所占的权重, 例如都是1的情况下，那个方向(LinearLayout的orientation方向)长度都是一样的。

注：在各属性的标识中就可以看出，只要是android:layout…的都是针对布局管理器本身进行设置的属性，而android:...的属性则是针对布局管理器的子元素进行设置的属性。

二、表格布局（TableLayout）



**1、表格布局的特点：**适用于多行多列的布局格式，每个TableLayout是由多个TableRow组成，一个TableRow就表示TableLayout中的每一行，这一行可以由多个子元素组成。实际上TableLayout和TableRow都是LineLayout线性布局的子类。但是TableRow的参数android:orientation属性值固定为horizontal，且android:layout\_width = MATCH\_PARENT，android:layout\_height = WRAP\_CONTENT。所以TableRow实际是一个横向的线性布局，且所以子元素宽度和高度一致。

注意：在TableLayout中，单元格可以为空，但是不能跨列，意思是单元格与单元格中间不能有空的单元格。

表格布局继承于LinearLayout，所以LinearLayout的所有属性它都支持，除此外TableLayout还支持3个属性：

（1）android:collapseColumns：设置需要被隐藏的列的序列号（序号从0开始），多个序号之间用逗号“，”分隔。

（2）android:shrinkColumns：设置允许被收缩的列的列序号（序号从0开始），多个序号之间用逗号“，”分隔。

（3）android:stretchColumns：设置允许被拉伸的列的列序号（序号从0开始），多个序号之间用逗号“，”分隔。

例如：android:stretchColumns = "0,1,2,3"// 表示产生4个可拉伸的列

三、RelativeLayout（相对布局）



1、相对布局的特点：相对布局（Relative Layout）的作用就像它的名字所描述的那样：它相对于另一控件或者它本身的父控件来布置控件。它意味着子控件，像ImageView，TextView和Button这些控件，能够摆放在其他控件的上、下、左或右方。子控件同样也能相对于父控件（相对布局的容器）去摆放，例如对齐在父布局的上、下、左或右边缘上。

2、相对布局管理器中的一些常用属性

**（1）第一类属性：属性值为true或者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

**（2）第二类属性：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

**（3）第三类属性：属性值为具体像素值，如30dip，40px等**

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

四、帧布局管理器（FrameLayout）



一、FrameLayout的特点：FrameLayout对象好比一块在屏幕上提前预定好的空白区域，可以将一些元素填充在里面，如图片。所有元素都被放置在FrameLayout区域的最左上区域，而且无法为这些元素指定一个确切的位置，若有多个元素，那么后面的元素会重叠显示在前一个元素上。

二、常用属性

（1）android:foreground：设置该帧布局容器的前景图像。

（2）android:foregroundGravity:定义绘制前景图像的gravity属性，即前景图像的显示位置。

五、绝对布局管理器（AbsoluteLayout）



该布局已经在API3之后被淘汰，用其他的布局管理器实现它的功能。

六、网格布局管理器（GridLayout）（该布局管理器是在android4.0之后才开始使用的）



GridLayout布局使用虚细线将布局划分为行、列和单元格，也支持一个控件在行、列上都有交错排列。而GridLayout使用的其实是跟LinearLayout类似的API，只不过是修改了一下相关的标签而已，所以对于开发者来说，掌握GridLayout还是很容易的事情。GridLayout的布局策略简单分为以下三个部分：

（1）首先它与LinearLayout布局一样，也分为水平和垂直两种方式，默认是水平布局，一个控件挨着一个控件从左到右依次排列，但是通过指定android:columnCount设置列数的属性后，控件会自动换行进行排列。另一方面，对于GridLayout布局中的子控件，默认按照wrap\_content的方式设置其显示，这只需要在GridLayout布局中显式声明即可。

（2）其次，若要指定某控件显示在固定的行或列，只需设置该子控件的android:layout\_row和android:layout\_column属性即可，但是需要注意：android:layout\_row=”0”表示从第一行开始，android:layout\_column=”0”表示从第一列开始，这与编程语言中一维数组的赋值情况类似。

（3）最后，如果需要设置某控件跨越多行或多列，只需将该子控件的android:layout\_rowSpan或者layout\_columnSpan属性设置为数值，再设置其**layout\_gravity**属性为**fill**即可，前一个设置表明该控件跨越的行数或列数，后一个设置表明该控件填满所跨越的整行或整列。

一、GridLayout布局管理器的常用属性

（1）**android:alignmentMode**：设置对齐模式，作用是设置网格布局管理器的对齐模式。

（2）**android:columnCount：**设置列数，作用是设置该网格布局的列数。

（3）**android:columnOrderPreserved**：设置是否保留列序列号，**作用**是设置网格容器是否保留列序列号。

（4）**android:rowCount：**设置该网格的行数。

（5）**android:rowOrderPreserved：**设置该网格容器是否保留行序列号。

（6）**android:useDefaultMargins：**设置该布局是否使用默认的页边距。

GridLayout的LayoutParams属性

（1）**android:layout\_column**：设置子组件在GridLayout的哪一列。

（2）**android:layout\_columnSpan**：设置该子组件在GridLayout中横向跨几列。

（3）**android:layout\_gravity：**设置该组件采用何种方式占据该网格的空间。

（4）**android:layout\_row：**设置该子组件在GridLayout的第几行。

（5）**android:layout\_rowSpan：**设置该子组件在GridLayout纵向横跨几行。