

高级搜索

公告：免费真机测试应用

# Android的Layout完全介绍

目录

1 1.FrameLayout

2 2.LinearLayout

3 3.RelativeLayout

4 4.AbsoluteLayout

5 5.TableLayout

浏览次数: 3856 次

评论次数: 0 次

编辑次数: 1 次 [历史版本](#)

更新时间: 2010-07-09

文档创建者

klzj



初学乍练

☆

UID 51294

积分 343

学分 0

[学分如何兑换礼品?](#)

来自 天津-天津市

最新编辑者

tigerding



登峰造极

😊😊😊

推荐文档

1.FrameLayout

[编辑本段](#) [回](#)

[FrameLayout](#)是最简单的一个布局对象。它被定制为你屏幕上的一个空白备用区域，之后你可以在其中填充一个单一对象 — 比如，一张你要发布的图片。所有的子元素将会固定在屏幕的左上角；你不能为FrameLayout中的一个子元素指定一个位置。后一个子元素将会直接在前一个子元素之上进行覆盖填充，把它们部份或全部挡住（除非后一个子元素是透明的）。

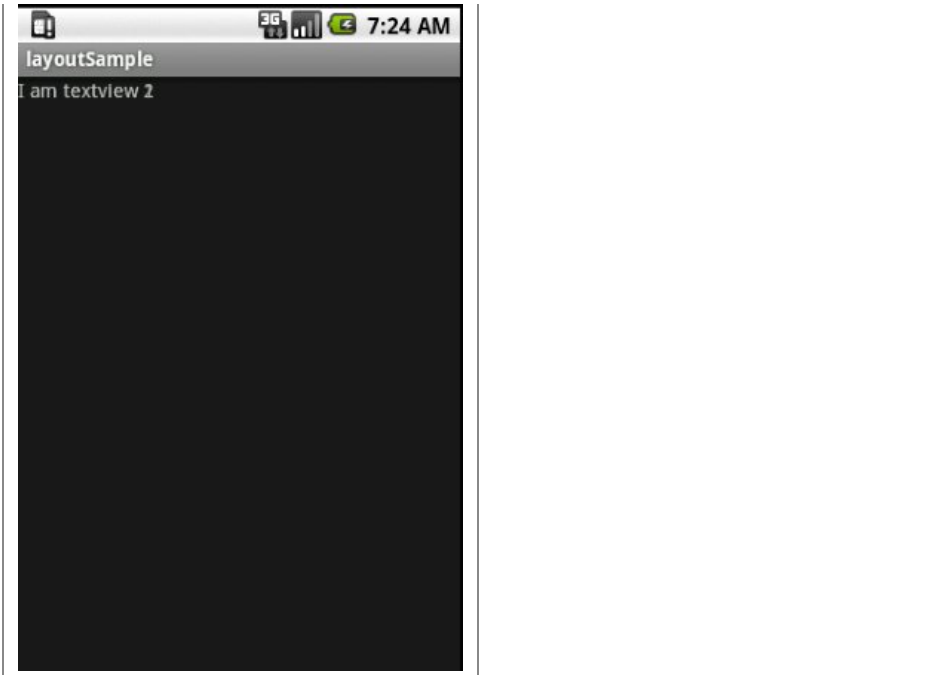
FrameLayout is the simplest type of layout object. It's basically a blank space on your screen th at you can later fill with a single object — for example, a picture that you'll swap in and out. All child elements of the FrameLayout are pinned to the top left corner of the screen; you cannot s pecify a different location for a child view. Subsequent child views will simply be drawn over pre vious ones, partially or totally obscuring them (unless the child object is transparent).

FrameLayout默认填充widget等在左上角，然后后面的控件遮住前面的，比如说有两个TextView，T ext内容分别是I am textview 1 和 I am textview 2 看到的效果会如下：

分享

回到顶部

```
?xml version="1.0" encoding="utf-8"?>
HTML <FrameLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="I am textview 1"
    />
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="I am textview 2"
    />
</FrameLayout>
```



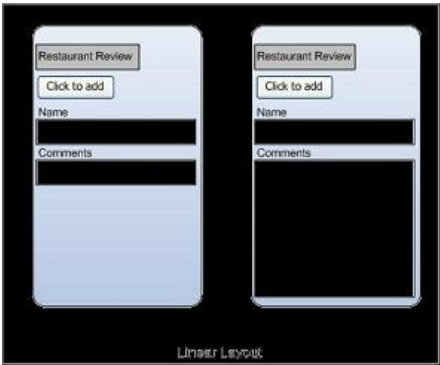
2.LinearLayout

[编辑本段](#) [回](#)

[LinearLayout](#)以你为它设置的垂直或水平的属性值，来排列所有的子元素。所有的子元素都被堆放在其它元素之后，因此一个垂直列表的每一行只会会有一个元素，而不管他们有多宽，而一个水平列表将会只有一个行高（高度为最高子元素的高度加上边框高度）。LinearLayout保持子元素之间的间隔以及互相对齐（相对一个元素的右对齐、中间对齐或者左对齐）。

LinearLayout 还支持为单独的子元素指定weight。好处就是允许子元素可以填充屏幕上的剩余空间。这也避免了在一个大屏幕中，一串小对象挤成一堆的情况，而是允许他们放大填充空白。子元素指定一个weight值，剩余的空间就会按这些子元素指定的weight比例分配给这些子元素。默认的weight值为0。例如，如果有三个文本框，其中两个指定了weight值为1，那么，这两个文本框将等比例地放大，并填满剩余的空间，而第三个文本框不会放大。

下面的两个窗体采用LinearLayout，包含一组的元素：一个按钮，几个标签，几个文本框。两个窗体都为布局做了一番修饰。文本框的width被设置为FILL\_PARENT；其它元素的width被设置为WRAP\_CONTENT。默认的对齐方式为左对齐。左边的窗体没有设置weight（默认为 0）；右边的窗体的comments文本框weight被设置为1。如果Name文本框也被设置为1，那么Name和Comments这两个文本框将会有同样的高度。

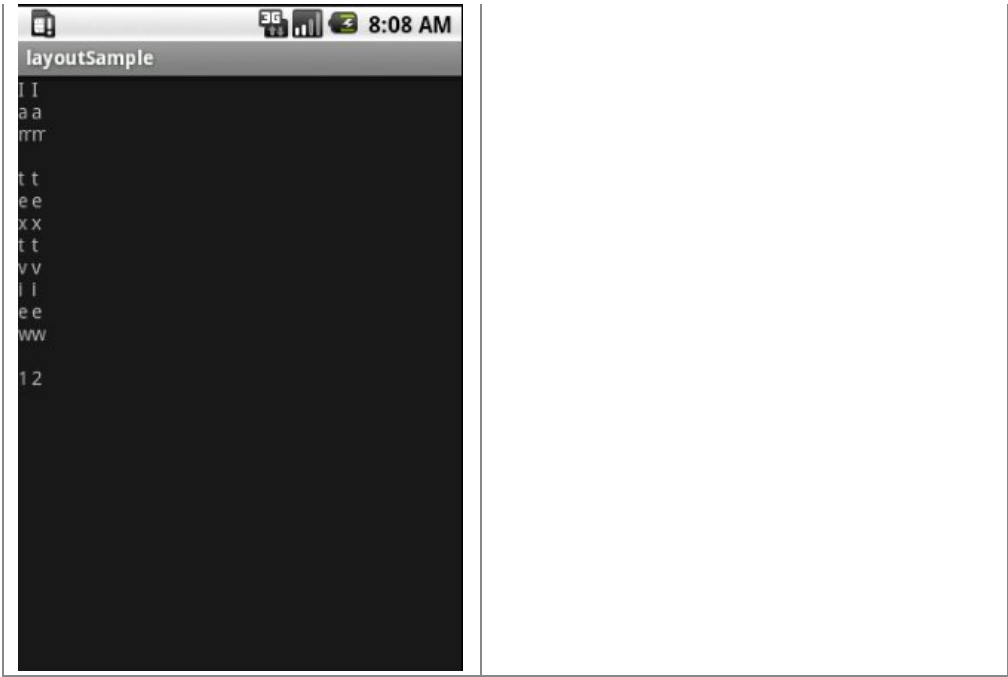


在一个水平排列的LinearLayout中，各项按他们的文本基线进行排列（第一列第一行的元素，即最上或最左，被设定为参考基线）。因此，人们在一个窗体中检索元素时，就不需要七上八下地读元素的文本了。我们可以在layout的XML中设置 android:baselineAligned="false"，来关闭这个设置。

[LinearLayout](#) aligns all children in a single direction — vertically or horizontally, depending on how you define theorientation attribute. All children are stacked one after the other, so a vert

ical list will only have one child per row, no matter how wide they are, and a horizontal list will only be one row high (the height of the tallest child, plus padding). A [LinearLayout](#) respects *margins* between children and the *gravity*(right, center, or left alignment) of each child. LinearLayout是Android [SDK](#)创建project时的默认Layout,支持两种方式，默认是垂直vertical的线性layout,另一个是水平horizontal方向的线性排列。  
效果如下

	<pre>&lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"     android:orientation="vertical"     android:layout_width="fill_parent"     android:layout_height="fill_parent"     &gt;     &lt;TextView         android:layout_width="fill_parent"         android:layout_height="wrap_content"         android:text="I am textview 1"     /&gt;     &lt;TextView         android:layout_width="fill_parent"         android:layout_height="wrap_content"         android:text="I am textview 2"     /&gt;  &lt;/LinearLayout&gt;</pre>
	<pre>&lt;?xml version="1.0" encoding="utf-8"?&gt; &lt;LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"     android:orientation="horizontal"     android:layout_width="fill_parent"     android:layout_height="fill_parent"     &gt;     &lt;TextView         android:layout_width="10px"         android:layout_height="fill_parent"         android:text="I am textview 1"     /&gt;     &lt;TextView         android:layout_width="10px"         android:layout_height="wrap_content"         android:text="I am textview 2"     /&gt;  &lt;/LinearLayout&gt;</pre>

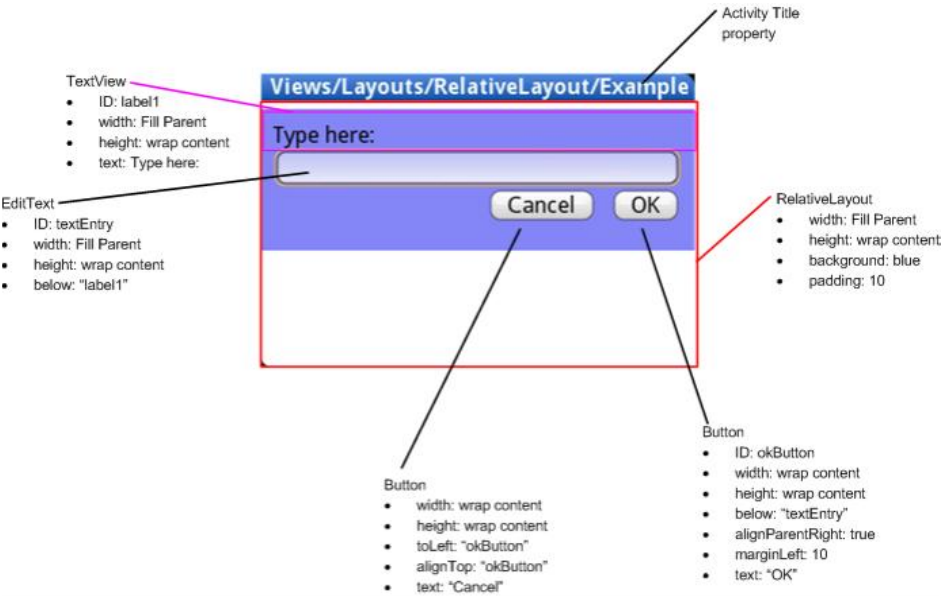


3.RelativeLayout

[编辑本段](#) [回](#)

[RelativeLayout](#)允许子元素指定他们相对于其它元素或父元素的位置（通过ID指定）。因此，你可以以右对齐，或上下，或置于屏幕中央的形式来排列两个元素。元素按顺序排列，因此如果第一个元素在屏幕的中央，那么相对于这个元素的其它元素将以屏幕中央的相对位置来排列。如果使用XML来指定这个layout，在你定义它之前，被关联的元素必须定义。

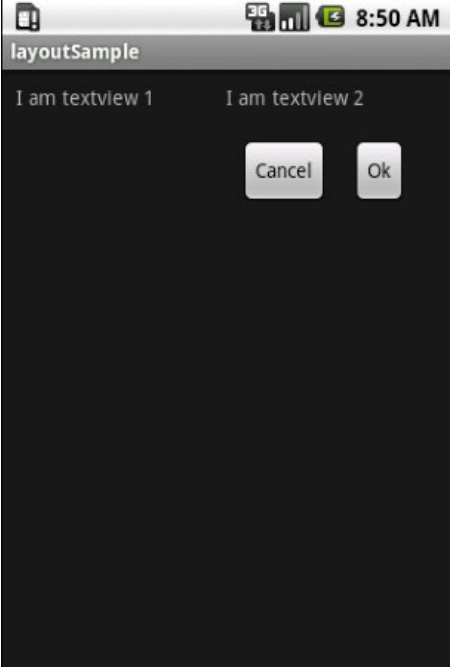
这是一个RelativeLayout例子，其中有可视的和不可视的元素。基础的屏幕layout对象是一个RelativeLayout对象。



这个视图显示了屏幕元素的类名称，下面是每个元素的属性列表。这些属性一部份是由元素直接提供，另一部份是由容器的LayoutParams成员（RelativeLayout的子类）提供。RelativeLayout参数有 width, height, below, alignTop, toLeft, padding和marginLeft。注意，这些参数中的一部份，其值是相对于其它子元素而言的，所以才RelativeLayout。这些参数包括toLeft, alignTop和below，用来指定相对于其它元素的左，上和下的位置。

[RelativeLayout](#) lets child views specify their position relative to the parent view or to each other (specified by ID). So you can align two elements by right border, or make one below another, centered left, and so on. Elements are rendered in the order given, so if t

he first element is centered in the screen, other elements aligning themselves to that element will be aligned relative to screen center. Also, because of this ordering, if using XML to specify th is layout, the element that you will reference (in order to position other view objects) must be lis ted in the XML file before you refer to it from the other views via its reference ID.



```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.a
ndroid.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:padding = "10px"
    >
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="I am textview 1"
/>
<TextView
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="I am textview 2"
    android:layout_marginLeft = "150px"
/>
<Button
    android:id="@+id/Button02" android:layout_wid
h="wrap_content" android:text="Ok"
    android:layout_height="wrap_content"
    android:layout_marginLeft = "240px"
    android:layout_marginTop = "40px"
></Button>
<Button android:id="@+id/Button01" android:lay
out_width="wrap_content" android:text="Cancel"
    android:layout_height="wrap_content"
    android:layout_marginLeft = "160px"
    android:layout_marginTop = "40px"
></Button>

</RelativeLayout
```

4.AbsoluteLayout

[编辑本段](#) [回](#)

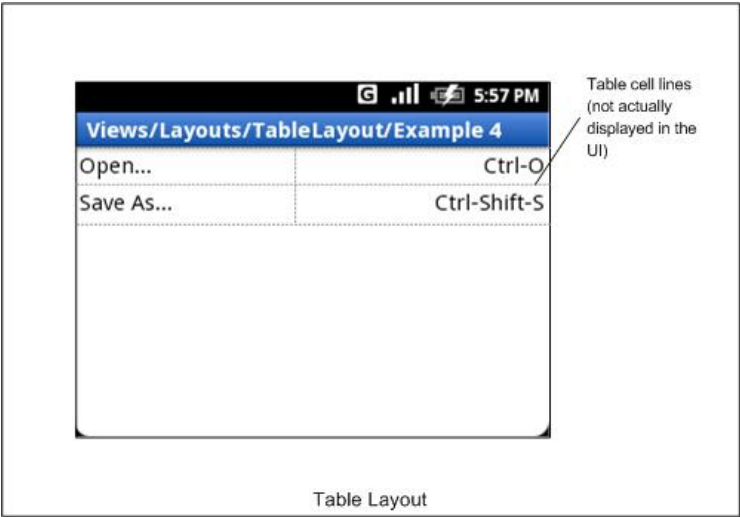
[AbsoluteLayout](#)可以让子元素指定准确的x/y坐标值，并显示在屏幕上。(0, 0)为左上角，当向下或向右移动时，坐标值将变大。AbsoluteLayout没有页边框，允许元素之间互相重叠（尽管不推荐）。我们通常不推荐使用AbsoluteLayout，除非你有正当理由要使用它，因为它使界面代码太过刚性，以至于在不同的设备上可能不能很好地工作。

5.TableLayout

[编辑本段](#) [回](#)

[TableLayout](#)将子元素的位置分配到行或列中。**android**的一个TableLayout由许多的TableRow组成，每个TableRow都会定义一个row（事实上，你可以定义其它的子对象，这在下面会解释到）。TableLayout容器不会显示row、cloumns或cell的边框线。每个row拥有0个或多个的cell；每个cell拥有一个 View对象。表格由列和行组成许多的单元格。表格允许单元格为空。单元格不能跨列，这与HTML中的不一样。下图显示了一个TableLayout，图中的虚线代表不可视的单元格边框。





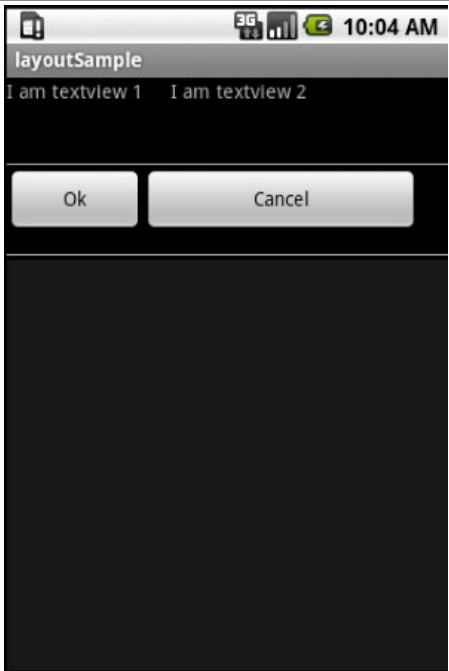
列可以被隐藏，也可以被设置为伸展的从而填充可利用的屏幕空间，也可以被设置为强制列收缩直到表格匹配屏幕大小。对于更详细信息，可以查看这个类的参考文档。

**TableLayout** positions its children into rows and columns. TableLayout containers do not display border lines for their rows, columns, or cells. The table will have as many columns as the row with the most cells. A table can [leave](#) cells empty, but cells cannot span columns, as they can in HTML.

**TableRow** objects are the child views of a TableLayout (each TableRow defines a single row in the table). Each row has zero or more cells, each of which is defined by any kind of other View. So, the cells of a row may be composed of a variety of View objects, like ImageView or TextView objects. A cell may also be a ViewGroup object (for example, you can nest another TableLayout as a cell).

TableLayout和HTML的<table>基本上类似，还提供TableRow class，直接等价与<tr> 比如说要做成一个类似下面的HTML页面

I am textview1	I am textview2
[OK button]	[Cancel button]



```
<?xml version="1.0" encoding="utf-8"?>
<TableLayout xmlns:android="http://schemas
.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    >
    <TableRow android:background="@drawabl
e/a1">
        <TextView
            android:text="I am textview 1" />
        <TextView
            android:paddingLeft = "20px"
            android:width = "200px"

            android:text="I am textview 2"
        />
    </TableRow>
    <TableRow android:background="@drawa
ble/a1">
        <Button
            android:id="@+id/Button02" android:text="
Ok"
        ></Button>
        <Button android:id="@+id/Button01" androi
d:text="Cancel"
        ></Button>
    </TableRow>
</TableLayout>
```

转载收藏到：

 Gmail

 百度搜索

 新浪微博

 人人网

 开心网

 豆瓣/九点

 Myspace

 139社区

 搜狐微博

 雅虎收藏

本文档对我有用

9

[查看论坛相关文章>>](#)

[法律声明](#) | [联系我们](#) | [帮助中心](#)

京ICP备05002571号 | 中国移动通信版权所有