



电子信息工程学院

移动终端开发技术

常用的布局类型

讲 师：陈媛媛

1

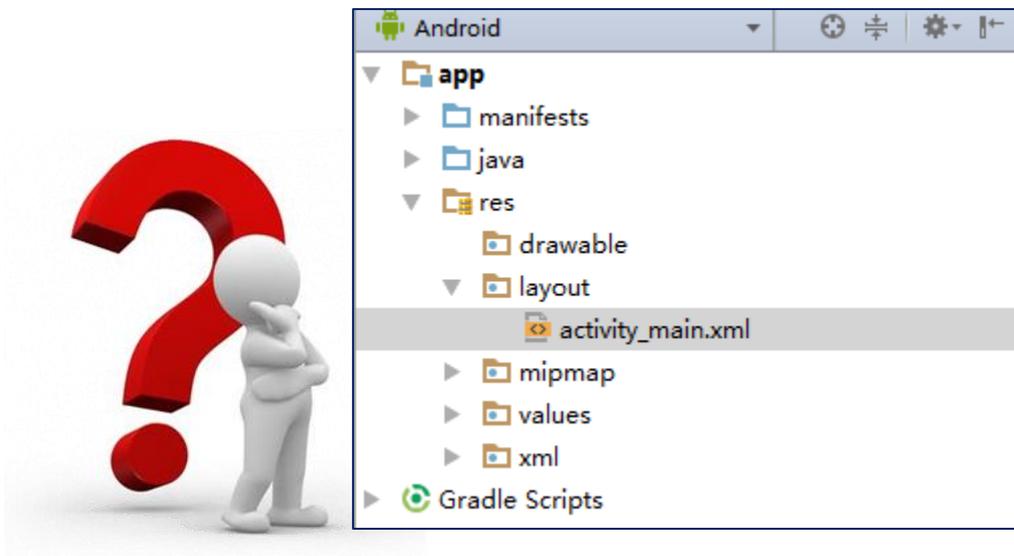
常见的布局类型



布局文件的创建

在Android应用程序中，**界面**是通过**布局文件**设定的。

布局文件采用**XML**格式，每个应用程序默认包含一个主界面布局文件，该文件位于项目的“**res/layout**”目录中。



 返回目录

2.1 布局类型

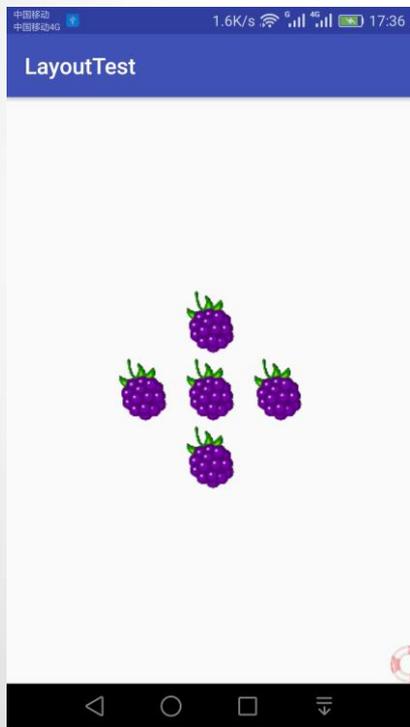
3.2.1 ● 相对布局

3.2.2 ● 线性布局

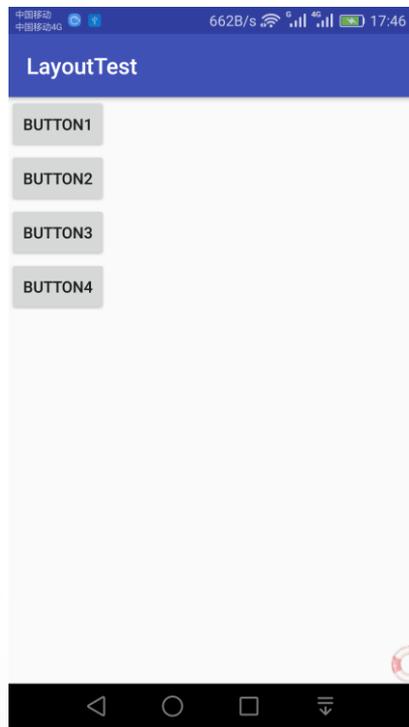
3.2.3 ● 表格布局

3.2.4 ● 网格布局

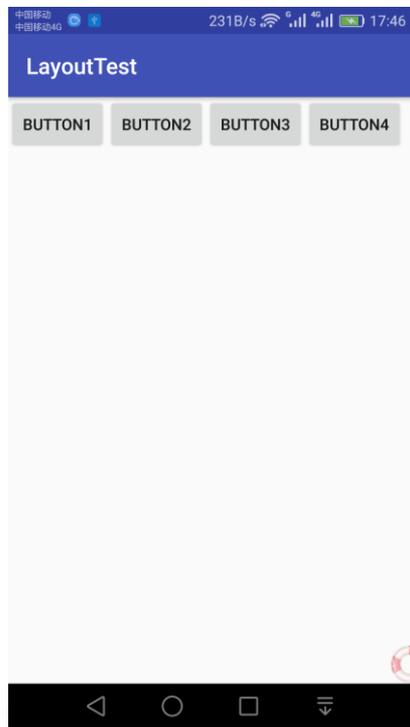
3.2.5 ● 帧布局



使用相对布局实现梅花布局效果



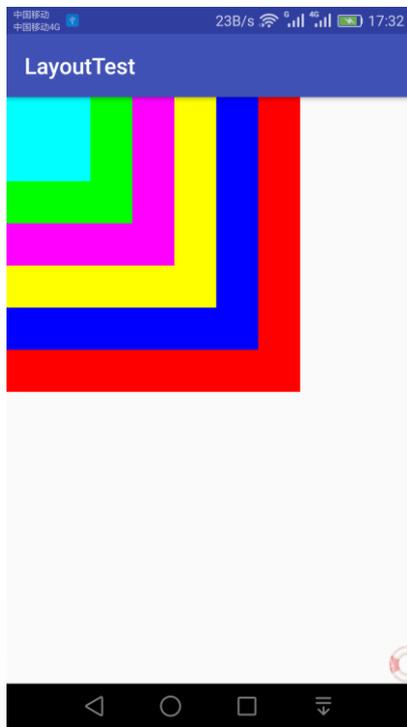
使用线性布局实现按钮的排列



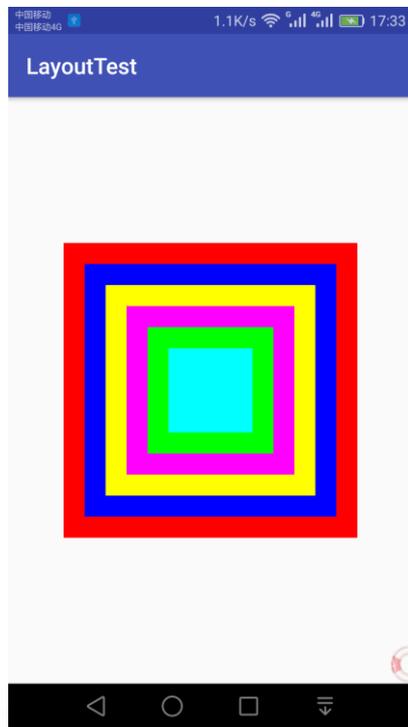
用户登录界面



丰富的表格布局



使用帧布局实现霓虹灯效果



使用网格布局实现计算器界面

相对布局 (RelativeLayout)

- 在Android Studio中开发Android程序时，**默认**采用的就是**相对布局**。
- 相对布局通常有两种形式，一种是**相对于容器**而言的，一种是**相对于控件**而言的。

具体来看两个例子吧

相对布局 (RelativeLayout)



相对布局

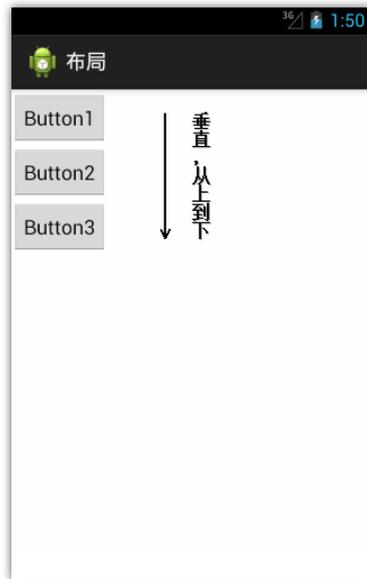
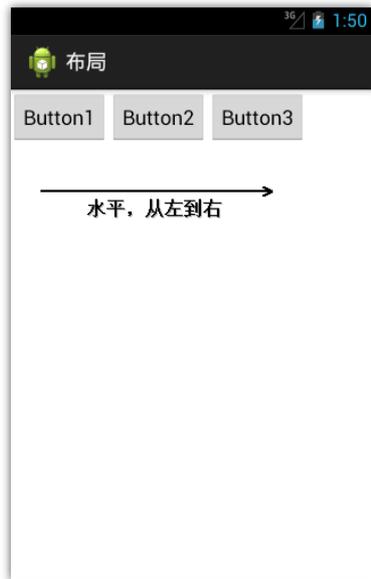
线性布局 (LinearLayout)

线性布局是Android中较为常用的布局方式，它使用`<LinearLayout>`标签表示。

线性布局有两种方式指定控件位置，一种是**水平方向**，一种是**竖直方向**。

`android:orientation="vertical"` 垂直
`android:orientation="horizontal"` 水平

`android:gravity` 设置布局管理器内组件的对齐方式。



表格布局 (TableLayout)

- ✓ 表格布局就是让控件**以表格**的形式来排列组件的，只要将组件或信息放在单元格中，控件就可以**整齐**的**排列**。
- ✓ 在TableLayout中，**行数**由**TableRow**对象控制的，即布局中有多少TableRow对象，就有多少行。

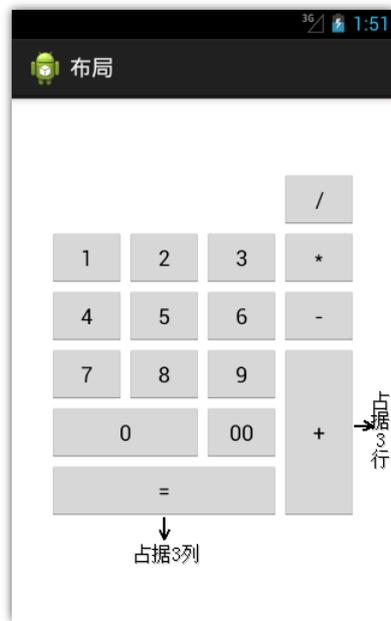
表格布局 (TableLayout)

- ✓ 在表格布局管理容器中，可以为单元格设置如下3种行为方式：
 - **Shrinkable**: 该列所有单元格的宽度可以被收缩，以保证表格适应父容器的宽度。
 - **Stretchable**: 该列所有单元格的宽度被拉伸，以保证组件能完全填满表格剩余空间。
 - **Collapsed**: 该列的所有单元格会被隐藏。



网格布局 (GridLayout)

- 网格布局是**Android4.0**新增的布局，它实现了控件的**交错显示**，能够避免因布局嵌套对设备性能的影响，更利于自由布局的开发。
- 网格布局用一组**无限细的直线**将绘图区域分成**行、列**和**单元**，并指定控件的显示区域和控件在该区域的显示方式。



网格布局 (GridLayout)

- GridLayout的作用把整个容器划分为rows和columns个网格，每个网格可以放置一个组件。除此之外，也可以设置一个组件横跨多少列、一个组件纵跨多少行。
- GridLayout提供了setRowCount(int)和setColumnCount(int)方法来控制该网格的行数量和列数量。属性rowCount和columnCount控制网格的行数和列数。

网格布局 (GridLayout)

表1 GridLayout的XML属性及相关方法说明

XML属性	相关方法	说明
android:alignmentMode	setAlignmentMode(int)	设置该布局管理器采用的对齐方式
android:columnCount	setColumnCount(int)	设置该网格的列数量
android:columnOrderPreserved	setColumnOrderPreserved(boolean)	设置该网格容器是否保留列序号
android:rowCount	setRowCount(int)	设置该网格的行数量
android:rowOrderPreserved	setRowOrderPreserved(boolean)	设置该网格容器是否保留行序号
android:useDefaultMargins	setDefaultMargins(boolean)	设置该布局管理器是否使用默认的页边距

网格布局 (GridLayout)

表2 GridLayout.LayoutParams的XML属性及相关方法说明

XML属性	相关方法	说明
android:layout_column		设置该子组件在GridLayout的第几列
android:layout_columnSpan		设置该子组件在GridLayout横向上跨几列
android:layout_gravity	setGravity(int)	设置该子组件采用何种方式占据该网格的空间
android:layout_row		设置该子组件在GridLayout的第几行
android:layout_rowSpan		设置该子组件在GridLayout纵向上跨几行

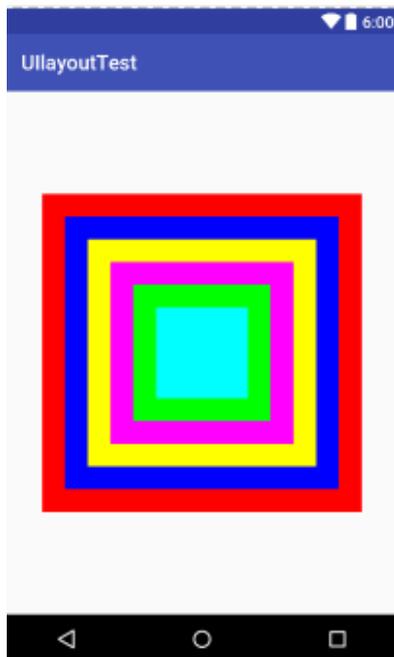
网格布局 (GridLayout)

- 为实现计算器界面，可以考虑将该界面分解成一个6*4的网格，其中第一个文本框横跨4列，第二个按钮横跨4列，后面每个按钮各占一格。
- 为实现该界面，考虑如下步骤来操作：
 - ① 在布局管理器中定义一个GridLayout，并在该GridLayout中依次定义文本框、按钮，该文本框、按钮各横跨4列。
 - ② 在java代码中循环16次，依次添加16个按钮。



帧布局 (FrameLayout)

- 帧布局为每个加入其中的控件创建一个**空白区域**（称为一帧，每个控件占据一帧）。
- 采用帧布局设计界面时，只能在屏幕**左上角显示**一个控件，如果添加多个控件，这些控件会按照顺序在屏幕的左上角**重叠显示**，且会**透明**显示之前控件的**文本**。
加上`layout_gravity="center"`可以居中显示。





1

相对布局

2

线性布局

3

表格布局、网格布局
