

一、简单阐述几种常用的界面布局

1.LinearLayout 线性布局：通常指定布局内的子控件水平或者竖直排列

属性名称	功能描述
android: orientation	设置布局内控件的排列顺序
android: layout_weight	在布局内设置控件权重，属性值可直接写int值

2.RelativeLayout 相对布局：通过相对定位的方式指定子控件的位置。

属性名称	功能描述
android: layout_centerInParent	设置当前控件位于父布局的中央位置
android: layout_centerVertical	设置当前控件位于父布局的垂直居中位置
android: layout_centerHorizontal	设置当前控件位于父控件的水平居中位置
android: layout_above	设置当前控件位于某控件上方
android: layout_below	设置当前控件位于某控件下方
android: layout_toLeftOf	设置当前控件位于某控件左侧
android: layout_toRightOf	设置当前控件位于某控件右侧
android: layout_alignParentTop	设置当前控件是否与父控件顶端对齐
android: layout_alignParentLeft	设置当前控件是否与父控件左对齐
android: layout_alignParentRight	设置当前控件是否与父控件右对齐
android: layout_alignParentBottom	设置当前控件是否与父控件底端对齐
android: layout_alignTop	设置当前控件上边界与某控件的上边界对齐
android: layout_alignBottom	设置当前控件下边界与某控件的下边界对齐
android: layout_alignLeft	设置当前控件左边界与某控件的左边界对齐
android: layout_alignRight	设置当前控件右边界与某控件的右边界对齐

3.FrameLayout 帧布局：用于在屏幕上创建一块空白区域，添加到该区域的每个子控件占一帧，这些帧会一个一个叠加在一起，后加入的控件会叠加在上一个控件上层。

属性名称	功能描述
android: foreground	设置帧布局容器的前景图像（始终在所有子控件之上）
android: foregroundGravity	设置前景图像显示的位置

4.TableLayout 表格布局: 采用行、列的形式来管理控件，它不需要明确声明包括多少行、多少列，而是通过在TableLayout布局中添加TableRow布局或控件来控制表格的行数，可以在TableRow布局中添加控件来控制表格列数。

属性名称	功能描述
android: stretchColumns	设置该列被拉伸
android: shrinkColumns	设置该列被收缩
android: collapseColumns	设置该列被隐藏

5. GridLayout 网格布局: 所有子控件默认在GridLayout中横向依次排列，当只等每行的列数时，到达指定列数
会自动换行显示。

属性名称	功能描述
android: layout_column	在网格的第几列
android: layout_row	在网格的第几行
android: layout_columnSpan	跨列
android: layout_rowSpan	跨行
android: layout_gravity	在一个网格中的重心位置
android: columnCount	每行列总数

6.ConstraintLayout 约束布局: 多用鼠标操作

android:constraintLeft_toLeftOf	控件的左边与另外一个控件的左边对齐
android: constraintLeft_toRightOf	控件的左边与另外一个控件的右边对齐
android: constraintRight_toLeftOf	控件的右边与另外一个控件的左边对齐
android: constraintRight_toRightOf	控件的右边与另外一个控件的右边对齐
android: constraintTop_toTopof	控件的上边与另外一个控件的上边对齐
android: constraintTop_toBottomOf	控件的上边与另外一个控件的底边对齐
android: constraintBaseline_toBaselineOf	控件间的文本内容基准线对齐
android: constraintStart_toEndOf	控件的起始边与另外一个控件的尾部对齐
android: constraintStart_toStartOf	控件的起始边与另外一个控件的起始边对齐
android: constraintEnd_toStartOf	控件的尾部与另外一个控件的起始边对齐
android: constraintEnd_toEndOf	控件的尾部与另外一个控件的尾部对齐
android: layout_constraintHorizontal_bias	横向的倾向
android: layout_constraintVertical_bias	纵向的倾向