

课程目录



1

布局与视图组

视图组和视图的层次性

- 视图组包含“子”视图，是视图的层次化表达，常见的视图组包括：
 - ConstraintLayout：使用约束来限制UI元素的位置。约束包括与其他元素之间或是与布局边缘之间的约束
 - ScrollView：包含一个元素并能滚动
 - RecyclerView：包含多个元素，并通过动态添加和删除元素来滚动



布局与视图组

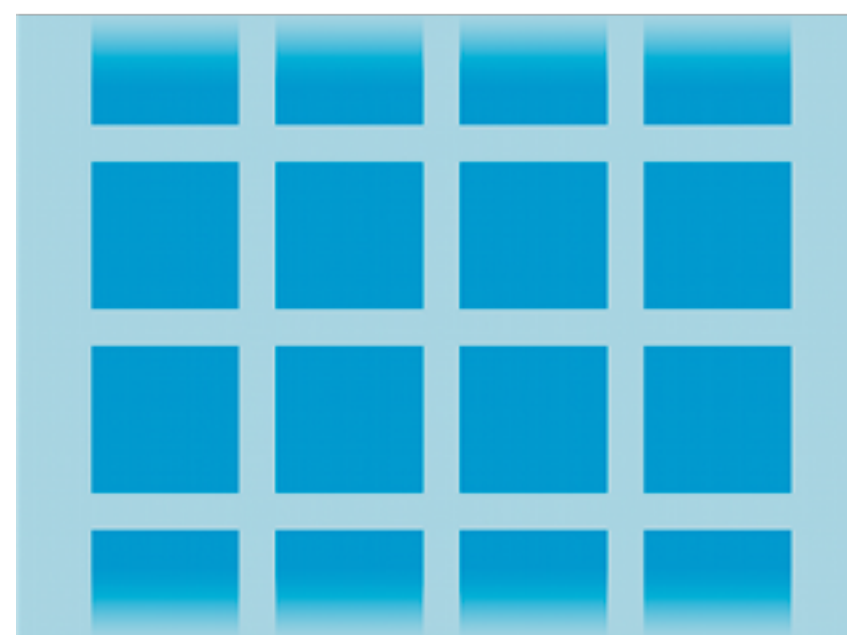
- **布局**是视图组的一种特定类型（布局类是视图组的子类），因此也包含子视图。常见的布局包括：LinearLayout, ConstraintLayout, GridLayout, TableLayout....



LinearLayout



ConstraintLayout



GridLayout



TableLayout

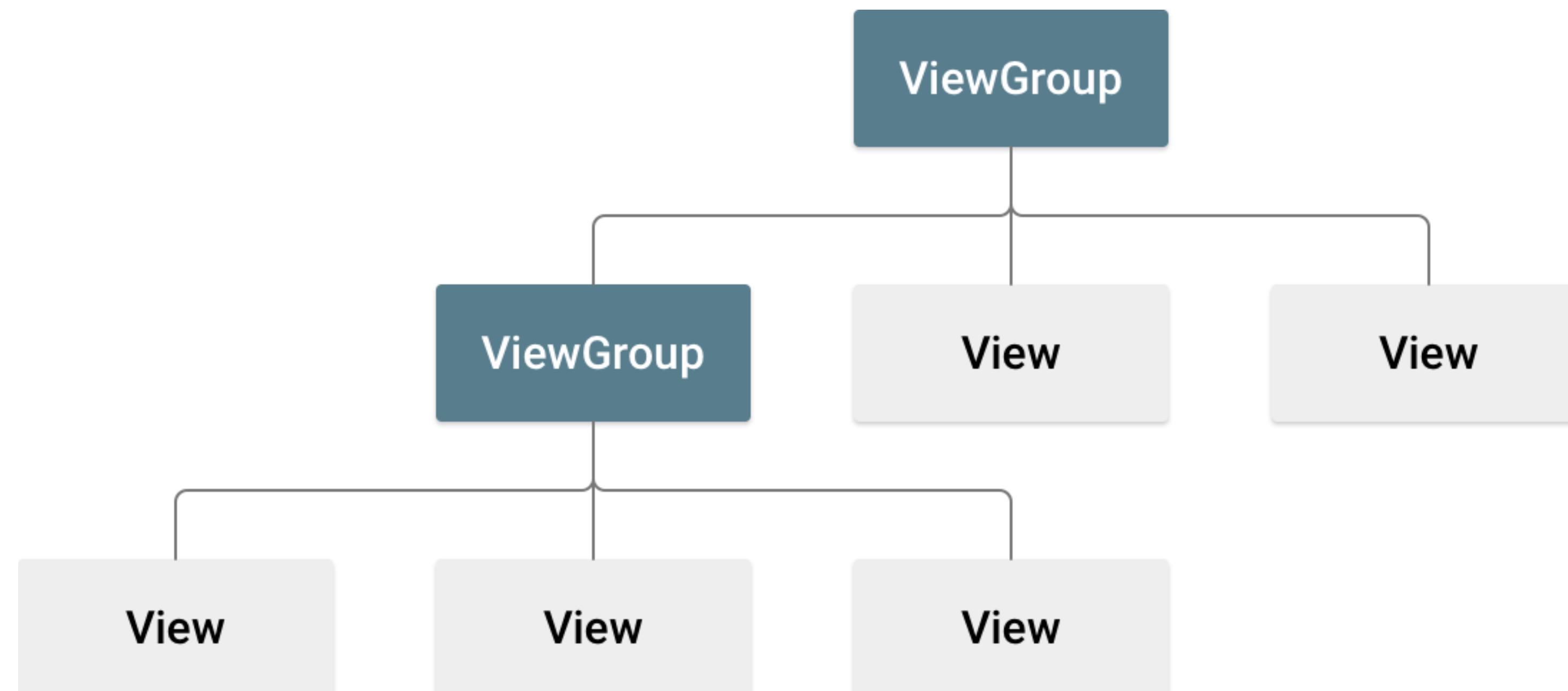


视图的层次性 vs 布局的层次性

- 视图类的层次结构是标准的面向对象中的类继承
 - 例如，Button is-a TextView is-a View is-an Object
 - 上述是is-a的关系
- 布局的层次结构则是描述视图如何排列
 - 例如，LinearLayout可以包含排列在一行的多个Button
 - 上述是has-a的关系



视图组和视图的层次关系



常用布局



LinearLayout



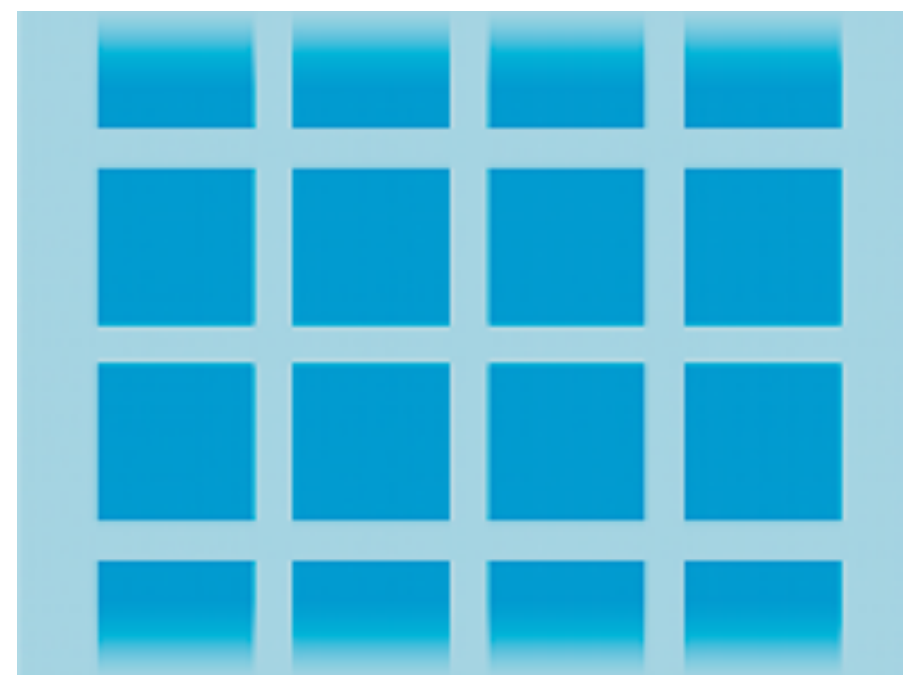
FrameLayout



TableLayout



RelativeLayout



GridLayout



ConstraintLayout



常用布局



LinearLayout

将子元素组织成单个水平或垂直行的布局。
如果窗口的长度超过屏幕的长度，它会创建一个滚动条。

1. **orientation** 布局中组件的排列方式
2. **gravity** 控制组件所包含的子元素的对齐方式，可多个组合
3. **layout_gravity** 控制该组件在父容器里的对其方式
4. **Background** 为该组件设置一个背景图片，或者是直接用颜色覆盖
5. **divider** 分割线
6. **showDividers** 设置分割线所在的位置，none（无），beginning（开始），end（结束），middle（每两个组件间）
7. **dividerPadding** 设置分割线的 padding
8. **layout_weight**(权重) 该属性是用来等比例的划分区域



常用布局



RelativeLayout

能够指定子对象相对于彼此或相对于父对象的位置。

根据父容器定位

1. `layout_alignParentLeft`
2. `layout_alignParentRight`
3. `layout_alignParentTop`
4. `layout_alignParentBottom`
5. `layout_centerHorizontal`
6. `layout_centerVertical`
7. `layout_centerInParent`

根据兄弟组件定位

1. `layout_toLeftOf`
2. `layout_toRightOf`
3. `layout_above`
4. `layout_below`
5. `layout_alignTop`
6. `layout_alignBottom`
7. `layout_alignLeft`
8. `layout_alignRight`

通用属性

1. `margin` 设置组件与父容器的边距
 - `layout_margin`
 - `layout_marginLeft`
 - `layout_marginRight`
 - `layout_marginTop`
 - `layout_marginBottom`

2. `padding` 设置组件内部元素的边距



常用布局



FrameLayout

1. **android:foreground** 设置前景
2. **android:foregroundGravity** 设置前景位置

所有放在布局里的控件，都按照层次堆叠在屏幕的左上角，后加进来的控件覆盖前面的控件。虽然默认会将控件放置在左上角，但是我们也可以通过layout_gravity属性，指定到其他的位置



常用布局



TableLayout

1. **android:collapseColumns** 设置需要被隐藏的列的序号，从0开始
2. **android:stretchColumns** 设置允许被拉伸的列的列序号，从0开始
3. **android:shrinkColumns** 设置允许被收缩的列的列序号，从0开始

子控件设置属性

1. **android:layout_column** 显示在第几列
2. **android:layout_span** 横向跨几列

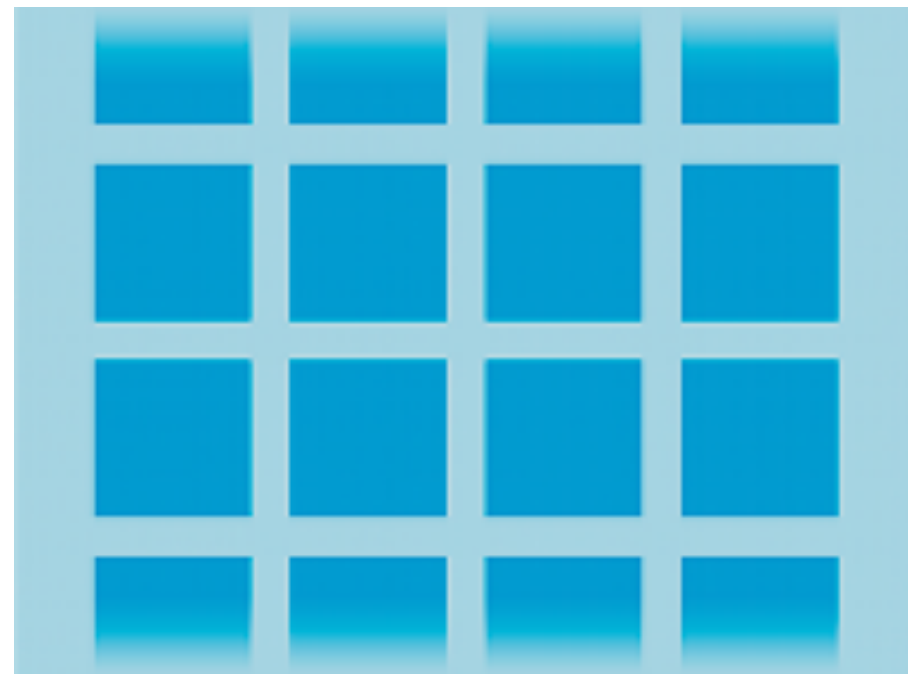
适用于多行多列的布局格式

每个TableLayout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。

常用布局



浙江大学
ZHEJIANG UNIVERSITY



GridLayout

显示列和行的滚动网格。

1. **android:orientation** 设置水平显示还是垂直显示
2. **android:columeCount** 设置行的显示个数
3. **android:rowCount** 设置列的显示个数

子控件设置属性

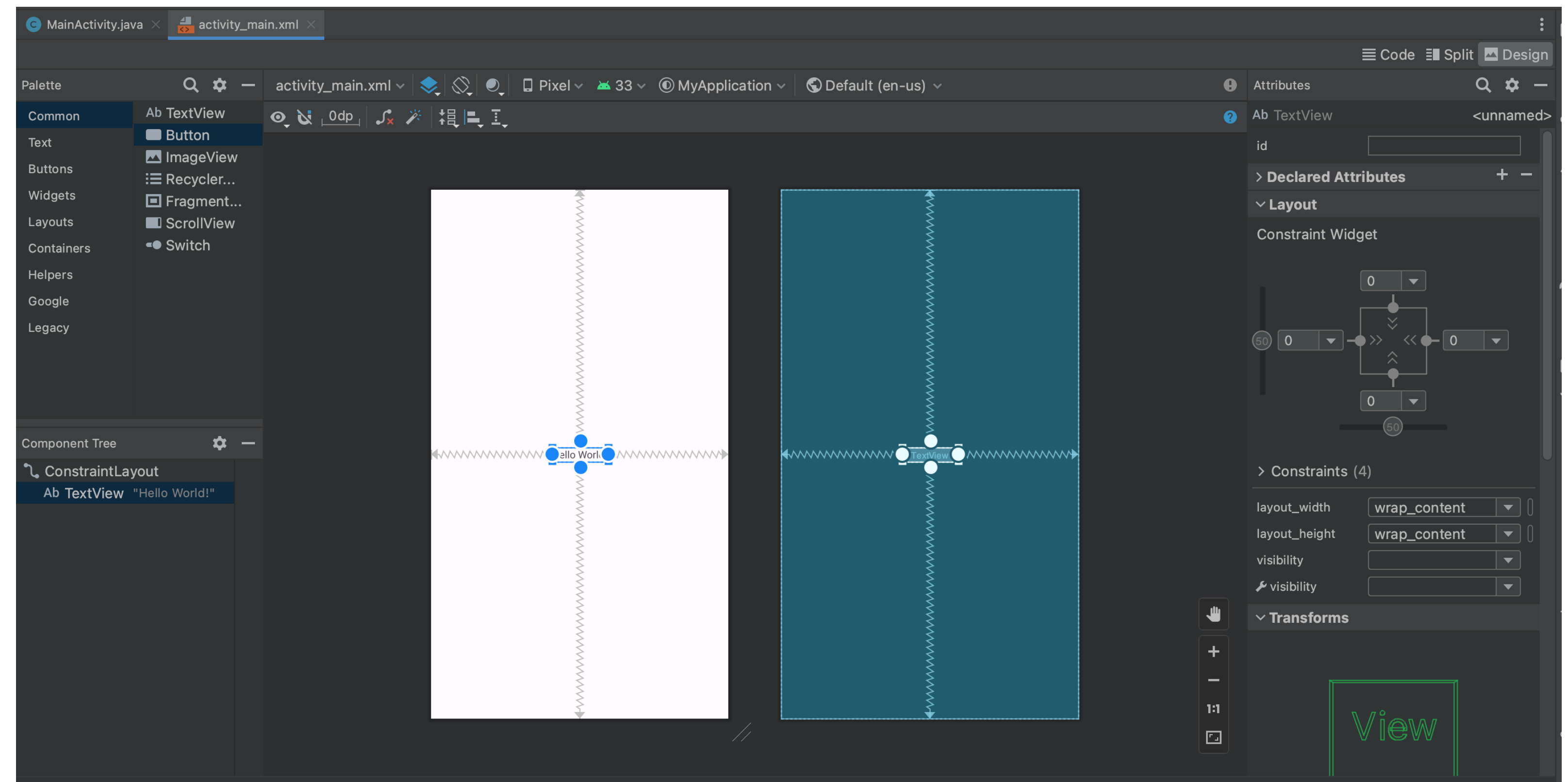
1. **android:layout_column** 显示在第几列
2. **android:layout_columnSpan** 横向跨几列
3. **android:layout_columnWeight** 横向剩余空间分配方式
4. **android:layout_gravity** 在网格中的显示位置
5. **android:layout_row** 显示在第几行
6. **android:layout_rowSpan** 纵向跨几行
7. **android:layout_rowWeight** 纵向剩余空间分配方式



常用布局



ConstraintLayout



Thanks!