# **GridLayout**

#### 知识解析

GridLayout: GridLayout 提供了一种新的布局方式,它可以将子视图放入到一个矩形网格中。

GridLayout 中的元素一般不采用 layout\_width/layout\_height 来界定大小

采用 layout\_gravity= "fill\_horizontal"或者"fill\_vertical",并配合 GridLayout 的 orientation 属性来定义它里面的视图元素的大小。也可以指定"fill",则会根据情况在水平、垂直方向填充。默认情况下,它里面的元素大小为 wrap\_content

GridLayout 中的 orientation 属性,决定了其中的视图元素的摆放方式:如果为 vertical,则先摆第一列,然后第二列…;如果为 horizontal,则先摆第一行,然后第二行…。

```
<GridLayout
xmlns:android="http://schemas.android.com/apk/res/and
roid"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:columnCount="4"
    android:rowCount="4" >
    <TextView android:text="1,1" />
    <TextView android:text="1,2" />

<TextView
    android:text="3,3 longer" />
... ...
```

上面的代码片段在 GridLayout 的 orientation 属性设置成水平和垂直的情况下的现实下过如下 2 图所示。

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4
3,1	3,2	3,3 longer	3,4
4,1	4,2	4,3	4,4

水平设置

1,1	2,1	3,1	4,1
1,2	2,2	3,2	4,2
1,3	2,3	3,3 longer	4,3
1,4	2,4	3,4	4,4

垂直设置

可以直接指定某个元素的行/列,那么,该元素将会直接"移动到"对应的位置,且其后的元素将从该位置往后放置:

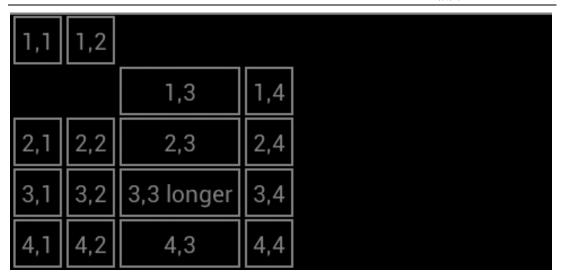
● layout\_row: 指定行

● layout\_column: 指定列

行和列一般只指定一个。

下面这段放在 GridLayout 中的代码,显示效果如下图所示。

```
<TextView
    android:layout_gravity="fill_horizontal"
    android:layout_row="1"
    android:text="1,3"/>
```

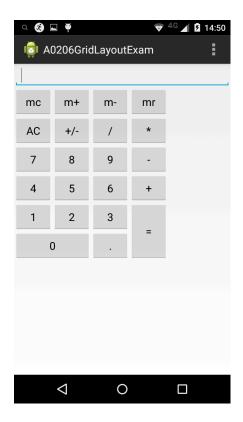


另外可以使用 rowSpan 和 columnSpan 来指定跨行、跨列列数。如:

layout\_columnSpan="2": 跨 2 列

layout\_rowSpan="2": 跨 2 行

### 功能演示



### 实战操作

```
<?xml version="1.0" encoding="utf-8"?>
<GridLayout
xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="wrap_content"
    android:columnCount="4"
    android:orientation="horizontal"
    android:rowCount="7" >
    <EditText
    android:id="@+id/editText1"
    android:layout_columnSpan="4"</pre>
```

```
android:layout gravity="fill" >
    </EditText>
    <Button
        android:id="@+id/button1"
        android:text="mc" />
   <Button
        android:id="@+id/button2"
        android:text="m+" />
   <Button
        android:id="@+id/button3"
        android:text="m-" />
   <Button
        android:id="@+id/button4"
        android:text="mr" />
   <Button android:text="AC" />
   <Button android:text="+/-" />
   <Button android:text="/" />
   <Button android:text="*" />
   <Button android:text="7" />
<Button android:text="8" />
   <Button android:text="9" />
   <Button android:text="-" />
   <Button android:text="4" />
   <Button android:text="5" />
   <Button android:text="6" />
   <Button android:text="+" />
   <Button android:text="1" />
   <Button android:text="2" />
   <Button android:text="3" />
   <Button
        android:layout gravity="fill vertical"
       android:layout rowSpan="2"
```

## 职业素质

网格布局是 Android4.0 以后新增的布局。因此需要在 Android4.0 以后的版本中才能使用网格布局,如果希望在更早的 Android 平台上使用该布局,侧需要导入相应的支持库。

网格布局的作用类似于 HTML 中的 Table 标签,他把整个容器划分为 rows X 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。