

1. TableLayout

```
<TableLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:stretchColumns="1" >
    <TableRow >
        <TextView android:text="@string/row1_col1" android:padding="3dp"/>
        <TextView android:text="@string/row1_col2" android:padding="3dp" />
        <TextView android:text="@string/row1_col3" android:padding="3dp" />
    </TableRow>
    <TableRow >
        <TextView android:text="@string/row2_col1" android:padding="3dp" />
        <TextView android:text="@string/row2_col2" android:padding="3dp" />
    </TableRow>
</TableLayout>
```

- 1.1 android:stretchColumns=“1”使用第 1 列（其实就是第 2 列）拉伸以填充满整个行，给第一个 TableRow 的每个 TextView 加个背景色看得更清楚

1.2

2. 上两种布局比较单调，如何使用嵌套的方式



- 2.1 最外层 LinearLayout 控制第二层。第二层上方层内部使用横向布局

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >
    <LinearLayout
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:layout_weight="1"
        android:orientation="horizontal" >
```

```

<TextView
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:background="#aa0000"
    android:gravity="center_horizontal"
    android:text="red" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:background="#00aa00"
    android:gravity="center_horizontal"
    android:text="green" />
<TextView
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:background="#0000aa"
    android:gravity="center_horizontal"
    android:text="blue" />

<TextView
    android:layout_width="wrap_content"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:background="#aaaa00"
    android:gravity="center_horizontal"
    android:text="yellow" />
</LinearLayout>
<LinearLayout
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:layout_weight="1"
    android:orientation="horizontal" >
    <TableLayout
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:stretchColumns="0,1,2">
        <TableRow>
            <TextView
                android:padding="3dp"
                android:text="第一行第一列" />
            <TextView

```

```

        android:padding="3dp"
        android:text="第一行第二列" />
    <TextView
        android:padding="3dp"
        android:text="第一行第三列" />
</TableRow>
<TableRow>
    <TextView
        android:padding="3dp"
        android:text="第二行第一列" />
    <TextView
        android:padding="3dp"
        android:text="第二行第二列" />
</TableRow>
</TableLayout>
</LinearLayout>
</LinearLayout>

```

3. 相对布局

- 3.1 相对布局的基本概念
- 3.2 相对布局的其他布局的区别
- 3.3 相对布局常用属性介绍

```
<?xml version="1.0" encoding="utf-8"?>
```

```
<!--
```

android:layout_above 将该控件的底部至于给定ID的控件之上

android:layout_below 将该控件的顶部至于给定ID的控件之下

android:layout_toLeftOf 将该控件的右边缘和给定ID的控件的左边缘对齐

android:layout_toRightOf 将该控件的左边缘和给定ID的控件的右边缘对齐

android:layout_alignBaseline 该控件的baseline和给定ID的控件的baseline对齐

android:layout_alignBottom 将该控件的底部边缘与给定ID控件的底部边缘

android:layout_alignLeft 将该控件的左边缘与给定ID控件的左边缘对齐

android:layout_alignRight 将该控件的右边缘与给定ID控件的右边缘对齐

android:layout_alignTop 将给定控件的顶部边缘与给定ID控件的顶部对齐

android:layout_alignParentBottom 如果该值为true，则将该控件的底部和父控件的底部对齐

android:layout_alignParentLeft 如果该值为true，则将该控件的左边与父控件的左边对齐

android:layout_alignParentRight 如果该值为true，则将该控件的右边与父控件的右边对

齐

android:layout_alignParentTop 如果该值为true，则将该空间的顶部与父控件的顶部对齐

android:layout_centerHorizontal 如果值为真，该控件将被至于水平方向的中央

android:layout_centerInParent 如果值为真，该控件将被至于父控件水平方向和垂直方向的

中央

android:layout_centerVertical 如果值为真，该控件将被至于垂直方向的中央

```
-->
```

```
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="fill_parent"
```

```
    android:layout_height="wrap_content"
```

```

        android:padding="10px" >

<TextView android:id="@+id/label"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:text="Type here:" />

```

```

<EditText android:id="@+id/entry"
    android:layout_width="fill_parent"
    android:layout_height="wrap_content"
    android:background="@android:drawable/editbox_background"//可以设置不同

```

的背景

```

    android:layout_below="@id/label" />

```

```

<Button android:id="@+id/ok"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_below="@id/entry"
    android:layout_alignParentRight="true"
    android:layout_marginLeft="10px"
    android:text="OK" />

```

```

<Button android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_toLeftOf="@id/ok"
    android:layout_alignTop="@id/ok"
    android:text="Cancel" />

```

```

</RelativeLayout>

```

4. 使用相对布局实现计算器

```

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent" >

```

```

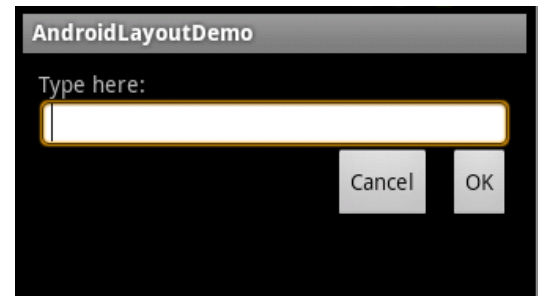
    <EditText
        android:id="@+id/txt_one"
        android:layout_width="100dp"
        android:layout_height="wrap_content"
        android:layout_alignParentLeft="true"
        android:layout_alignParentTop="true"
        android:layout_marginTop="18dp"
        android:ems="10" >
        <requestFocus />
    </EditText>

```

```

<TextView
    android:id="@+id/lblOperator"
    android:layout_width="wrap_content"

```



```
    android:layout_height="wrap_content"
    android:layout_alignBottom="@+id/txt_one"
    android:layout_alignTop="@+id/txt_one"
    android:layout_toRightOf="@+id/txt_one"
    android:gravity="center"    //文字居中
    android:text="加"
    android:textSize="20dp" />
```

<EditText

```
    android:id="@+id/txt_two"
    android:layout_width="100dp"
    android:layout_height="wrap_content"
    android:layout_alignBaseline="@+id/lblOperator"
    android:layout_alignBottom="@+id/lblOperator"
    android:layout_toRightOf="@+id/lblOperator"
    android:ems="10" />
```

<Button

```
    android:id="@+id/btn_Calculate"
    android:layout_width="70dp"
    android:layout_height="wrap_content"
    android:layout_alignTop="@+id/txt_two"
    android:layout_marginLeft="10dp"
    android:layout_toRightOf="@+id/txt_two"
    android:text="等于" />
```

<EditText

```
    android:id="@+id/txt_result"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:layout_alignParentLeft="true"
    android:layout_alignRight="@+id/btn_Calculate"
    android:layout_below="@+id/txt_two"
    android:layout_marginTop="50dp"
    />
```

</RelativeLayout>