

Layouts



ANDROID

Layout - разметка

Любая Activity (деятельность, окно приложения) состоит из

Views и ViewGroups

View – виджет, который рисуется на экране, например, кнопка, текстовое поле, список. Все такие виджеты наследуются от класса *android.view.View*.

ViewGroup – разметка, позволяющая расположить один или несколько **View**. Разметки наследуются от класса *android.view.ViewGroup*, который в свою очередь наследуется от *android.view.View*.

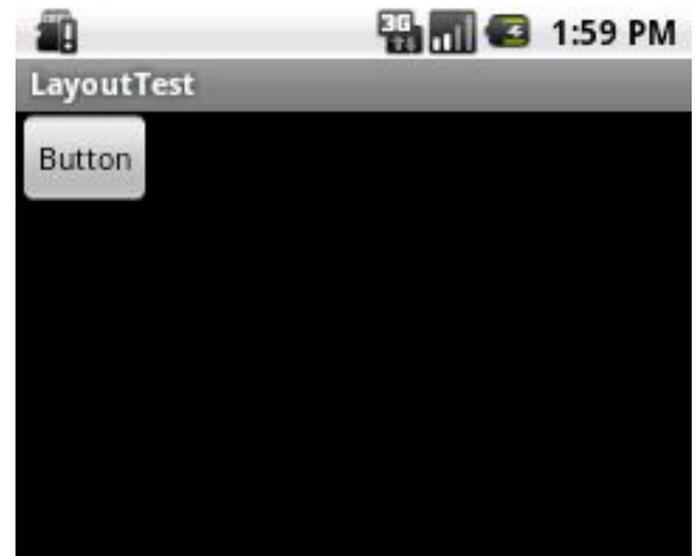
ViewGroups

- **FrameLayout**
- **LinearLayout**
- **TableLayout**
- **RelativeLayout**
- **ScrollView**

FrameLayout

Разметка для отображения одного элемента. Дочерние View или ViewGroup в FrameLayout выравниваются по верхнему левому углу. Разметка может содержать несколько элементов, но тогда они будут перекрывать друг друга

```
1.<FrameLayout
2.  android:layout_width="fill_parent"
3.  android:layout_height="fill_parent">
4.
5.  <Button
6.    android:layout_width="wrap_content"
7.    android:layout_height="wrap_content"
8.    android:text="@string/Button"
9.  />
10.</FrameLayout>
```



FrameLayout

```
1.<FrameLayout
2.  android:layout_width="fill_parent"
3.  android:layout_height="fill_parent">
4.
5.  <ImageView
6.    android:layout_width="wrap_content"
7.    android:layout_height="wrap_content"
8.    android:src="@drawable/image"
9.  />
10. <Button
11.   android:layout_width="wrap_content"
12.   android:layout_height="wrap_content"
13.   android:text="@string/Button"
14. />
15.</FrameLayout>
```



FrameLayout visibility

Атрибут *android:visibility* в xml или метод *setVisibility (int visibility)*, где visibility принимает одно из трех значений:

VISIBLE — элемент виден;

INVISIBLE — элемент невиден, но все еще занимает место в разметке;

GONE — элемент невидим и не занимает место в разметке

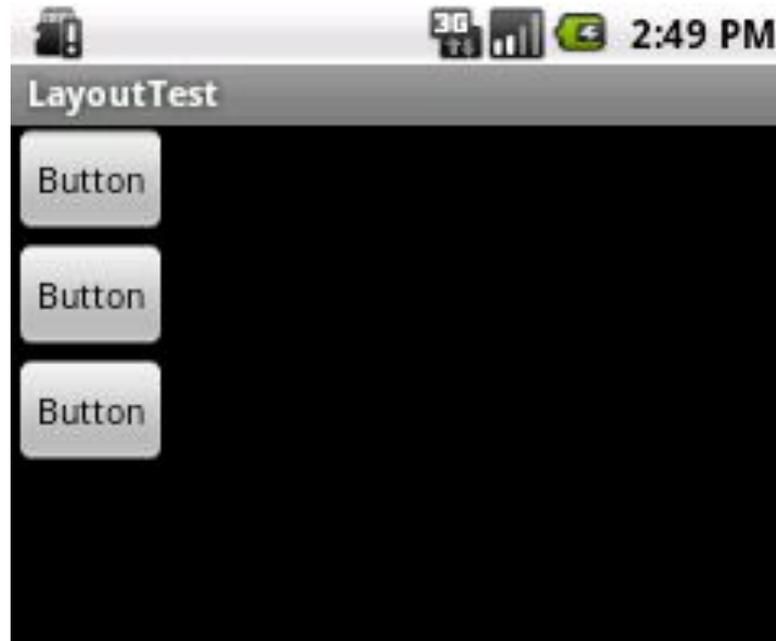
LinearLayout

Разметка для отображения одного или нескольких элементов в одну линию, горизонтально или вертикально. Для выбора ориентации используется атрибут **android:orientation** с двумя возможными значениями «**horizontal**» и «**vertical**»

LinearLayout *vertical*

```
1.<LinearLayout
2.  android:layout_width="fill_parent"
3.  android:layout_height="fill_parent"
4.  android:orientation="vertical"
5. >
6. <Button
7.   android:layout_width="wrap_content"
8.   android:layout_height="wrap_content"
9.   android:text="@string/Button"
10. />
11. <Button
12.  android:layout_width="wrap_content"
13.  android:layout_height="wrap_content"
14.  android:text="@string/Button"
15. />
16. <Button
17.  android:layout_width="wrap_content"
18.  android:layout_height="wrap_content"
19.  android:text="@string/Button"
20. />
21.</LinearLayout>
```

LinearLayout *vertical*



LinearLayout *horizontal*

```
1.<LinearLayout
2.  android:layout_width="fill_parent"
3.  android:layout_height="fill_parent"
4.  android:orientation="horizontal"
5. >
6. <Button
7.   android:layout_width="wrap_content"
8.   android:layout_height="wrap_content"
9.   android:text="@string/Button"
10. />
11. <Button
12.  android:layout_width="wrap_content"
13.  android:layout_height="wrap_content"
14.  android:text="@string/Button"
15. />
16. <Button
17.  android:layout_width="wrap_content"
18.  android:layout_height="wrap_content"
19.  android:text="@string/Button"
20. />
21.</LinearLayout>
```

LinearLayout *horizontal*



TableLayout

Разметка для расположения элементов в виде таблицы.

Ряды задаются в xml с помощью тега *TableRow*, а ячейки в каждом ряду создаются автоматически для каждого элемента. Количество колонок в таблице = **максимальному** количеству элементов в рядах.

Ширина колонки = самому широкому элементу в ней.

Ячейки таблицы можно оставлять пустыми, или объединить.

Атрибут ***android:layout_column*** — задает в какой столбец поместить данный элемент (нумерация столбцов начинается с 0).

Атрибут ***android:layout_span*** — позволяет объединить указанное количество столбцов (ряды объединить нельзя).

TableLayout

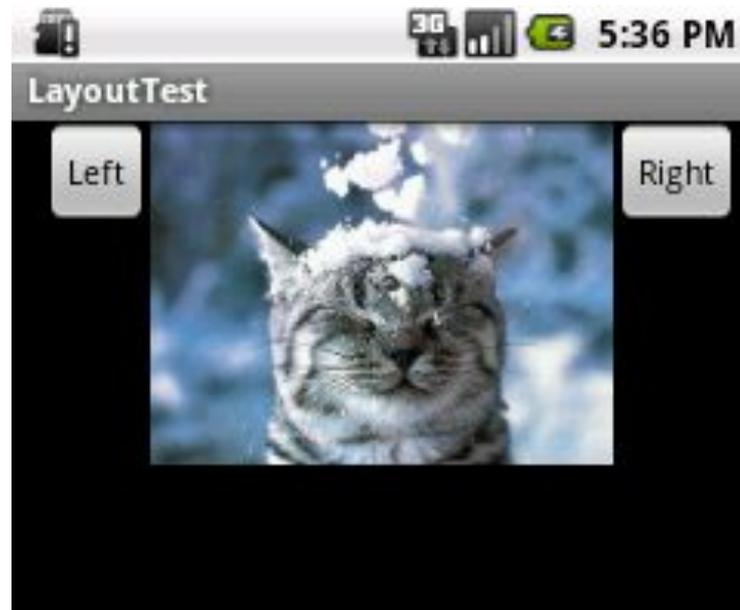
```
1.<TableLayout android:layout_width="fill_parent"  
  android:layout_height="fill_parent">  
2. <TableRow>  
3.   <ImageView android:layout_width="wrap_content"  
4.     android:layout_height="wrap_content" android:src="@drawable/image"  
5.     android:layout_column="1"/>  
6. </TableRow>  
7.  
8. <TableRow>  
9.   <Button android:layout_width="wrap_content"  
10.    android:layout_height="wrap_content" android:text="@string/Button"/>  
11. <Button android:layout_width="wrap_content"  
12.    android:layout_height="wrap_content" android:text="@string/Button"/>  
13. </TableRow>  
14.  
15. <TableRow>  
16.   <Button android:layout_width="wrap_content"  
17.    android:layout_height="wrap_content" android:text="@string/Button"  
18.    android:layout_span="2"/>  
19. </TableRow>  
20.</TableLayout>
```

TableLayout



RelativeLayout

Разметка для расположения элементов относительно родителя или друг друга. Элементы начинают располагаться в указанном порядке, поэтому необходимо чтобы элемент был описан до того, как другой элемент будет на него ссылаться.



RelativeLayout

```
1.<RelativeLayout
2.  android:layout_width="fill_parent"
3.  android:layout_height="fill_parent">
4.
5.  <ImageView android:id="@+id/image"
6.    android:layout_width="wrap_content"
7.    android:layout_height="wrap_content"
8.    android:src="@drawable/image"
9.    android:layout_centerHorizontal="true"
10. />
11. <Button android:layout_width="wrap_content"
12.  android:layout_height="wrap_content"
13.  android:layout_toLeftOf="@id/image"
14.  android:text="@string/Left"
15. />
16. <Button android:layout_width="wrap_content"
17.  android:layout_height="wrap_content"
18.  android:layout_toRightOf="@id/image"
19.  android:text="@string/Right"
20. />
21.</RelativeLayout>
```

RelativeLayout *attributes*

Расположение относительно родительского элемента.
Значением атрибута является «**true**»

android:layout_alignParentTop — верхняя граница элемента располагается на верхней границе RelativeLayout

android:layout_alignParentLeft — левая граница элемента располагается на левой границе RelativeLayout

android:layout_alignParentRight — правая граница элемента располагается на правой границе RelativeLayout

android:layout_alignParentBottom — нижняя граница элемента располагается на нижней границе RelativeLayout

android:layout_centerInParent — элемент располагается в центре родительского по горизонтали и вертикали

android:layout_centerHorizontal — элемент располагается в центре родительского по горизонтали

android:layout_centerVertical — элемент располагается в центре родительского по вертикали

RelativeLayout *attributes*

Расположение относительно других элементов.
Значением атрибута является *id* другого элемента.

android:layout_above — элемент располагается сверху от другого элемента

android:layout_toLeftOf — элемент располагается слева от другого элемента

android:layout_toRightOf — элемент располагается справа от другого элемента.

android:layout_below — элемент располагается снизу от другого элемента.

android:layout_alignBaseline — базовая линия элемента выравнивается по базовой линии другого элемента

android:layout_alignTop — верхняя граница элемента выравнивается по верхней границе другого элемента

android:layout_alignLeft — левая граница элемента выравнивается по левой границе другого элемента

android:layout_alignRight — правая граница элемента выравнивается по правой границе другого элемента

android:layout_alignBottom — нижняя граница элемента выравнивается по нижней границе другого элемента

ScrollView

ScrollView является наследником класса **FrameLayout**. Отличием является то, что он позволяет прокручивать элементы, если они занимают больше места чем физический размер экрана.

В **ScrollView** также может содержаться только один элемент (**View** или **ViewGroup**), чаще всего используется **LinearLayout**, в которого вложено несколько элементов.

ScrollView поддерживает только вертикальную прокрутку.

ScrollView

```
1.<ScrollView
2.  android:layout_width="fill_parent"
3.  android:layout_height="fill_parent">
4.  <LinearLayout
5.    android:layout_width="fill_parent"
6.    android:layout_height="wrap_content"
7.    android:orientation="vertical">
8.      <!-- Много элементов -->
9.    </LinearLayout>
10.</ScrollView>
```

