День 6 Layouts — разметка

Сегодня мы не будем писать конкретное приложение, но разберём все разметки, вы должны понимать, что от того, какую разметку вы выберите, будет зависеть отображение компонентов на вашем экране.

Любая *Activity* — окно приложения, состоит из компонентов — *View* (например, кнопка, текстовое поле, список и так далее) и *ViewGroups* — разметка, которая позволяет расположить, так или иначе, наши  компоненты — *View*.

На сегодняшний день *Android Studio* содержит следующие *ViewGroups*, которые вы сможете найти в библиотеке компонентов и разделе **Layout**:

1. **FrameLayout**;
2. **LinearLayout** (Horizontal и Vertical);
3. **TableLayout**;
4. **TableRow**;
5. **GridLayout**;
6. **RelativeLayout**.

Создаём новый проект под названием **LayoutApp** и приступаем.

FrameLayout

**FrameLayout** является самой простой из разметок. Эта разметка используется для отображения одного элемента. По умолчанию элемент выравнивается по верхнему левому углу. Зайдите в**activity\_main.xml** и во вкладке **Text** наберите следующий код:

<FrameLayout

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ImageView

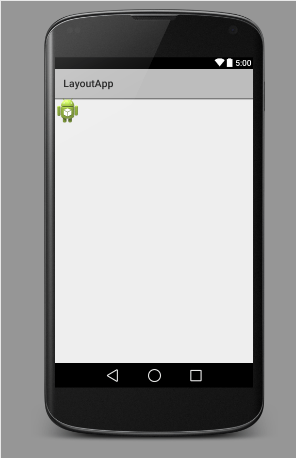
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_launcher" />

</FrameLayout>

Во вкладке **Design** можно посмотреть результат:



В разметке **FrameLayout** можно добавлять несколько элементов, но они будут наложены друг на друга. Эту разметку обычно используют для нескольких элементов, которые скрываются друг за другом и появляются программно. Скрытие элемента создаётся с помощью атрибута **android:visibility**, где **visibility** может принимать одно из трёх значений:

1. **VISIBLE** — показать элемент;
2. **INVISIBLE** — элемент скрыт, но занимает место в разметке;
3. **GONE** — элемент скрыт и не занимает место в разметке.

<FrameLayout

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:src="@drawable/ic\_launcher" />

<Button

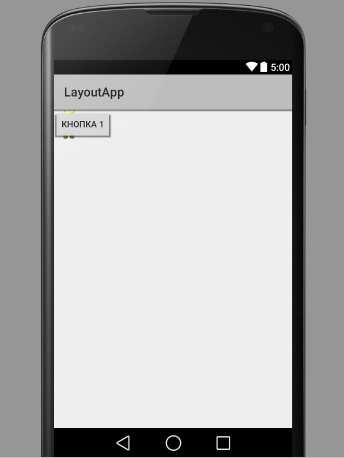
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Кнопка 1"

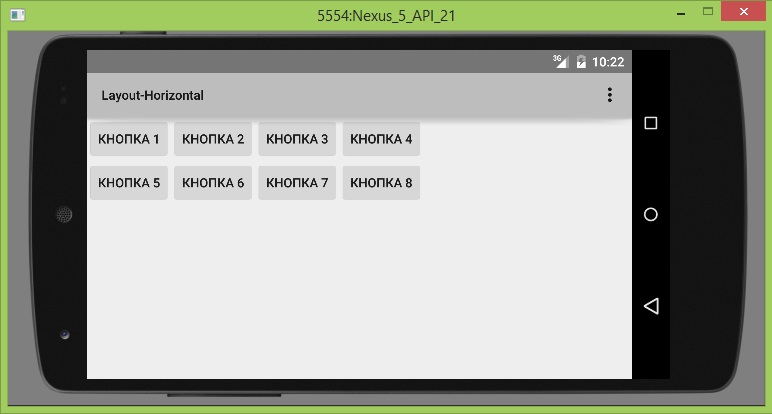
/>

</FrameLayout>



LinearLayout

Из названия можно понять, что эта разметка нужда для отображения одного или нескольких элементов в линию. Разметку **LinearLayout** можно расположить горизонтально и вертикально (на [прошлом уроке](http://mobile-dev.ru/urok-5-smena-orientacii-ekrana/) мы эти уже занимались), за это отвечает атрибут**android:orientation**, который принимает два значения: **horizontal** и **vertical**. Примеры можете посмотреть в [Урок 5. Смена ориентации экрана](http://mobile-dev.ru/urok-5-smena-orientacii-ekrana/).

TableLayout и TableRow

Разметка **TableLayout** нужна для расположения элементов в виде таблицы. Создадим пустую разметку:

<TableLayout

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

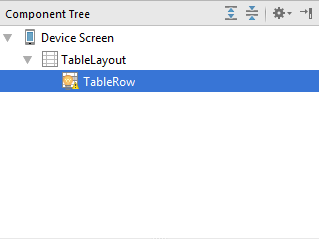
android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

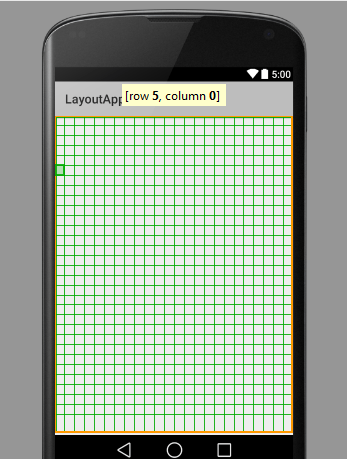
</TableLayout >

Для добавления ряда используют теги **<TableRow></TableRow>**, между этими тегами создают элементы. Количество колонок будет равняться максимальному количество элементов в ряду, например, если в вашей программе два ряда: в первом ряду четыре элемента, а во втором две, то количество колонок будет равняться четырём. Ширина колонки будет равна самому широкому элементу в ней. В разметке **TableLayout**с помощью атрибута **android:layout\_column**задаётся колонка, в которой необходимо разместить элемент. Атрибут **android:layout\_span** объединяет указанное количество столбцов. Нумерация столбцов начинается с нуля.

Можно создавать ряды с помощью тега **TableRow**, а можно во вкладке **Design** перенести его с библиотеки компонентов в дерево компонентов и продолжать работать с ним.



При перетаскивании **TableRow**, рабочая область будет покрыта сеткой, в которой вы можете выбрать количество строк и нужный столбец:



Перейдите во вкладку **Text**и напишите следующий код:

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent">

<TableRow>

<ImageView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_column="1"

android:src="@drawable/ic\_launcher" />

</TableRow>

<TableRow>

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="1 колонка "

android:textSize="20sp" />

<TextView

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text=" 2 колонка"

android:textSize="20sp"/>

</TableRow>

<TableRow>

<Button

android:layout\_width="wrap\_content"

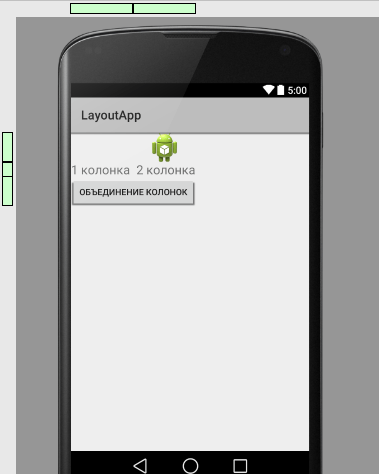
android:layout\_height="wrap\_content"

android:layout\_span="2"

android:text="Объединение колонок" />

</TableRow>

</TableLayout>



GridLayout

Эта разметка очень похожа на предыдущую, но она гораздо функциональнее. В ней также есть колонки, ряды, ячейки, но они гибче настраиваются. Если вы создаёте разметку во вкладке **Design**, то при перетаскивании элемента появится сетка для упрощения создания разметки.

Вставьте следующий код:

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

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

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_gravity="center"

android:columnCount="8"

android:rowCount="10"

android:orientation="horizontal">

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="\*"

android:id="@+id/buttonMul"

android:layout\_row="5"

android:layout\_column="7" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="1"

android:id="@+id/button1"

android:layout\_row="6"

android:layout\_column="4" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="2"

android:id="@+id/button2"

android:layout\_row="6"

android:layout\_column="5" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="3"

android:id="@+id/button3"

android:layout\_row="6"

android:layout\_column="6" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="+"

android:id="@+id/buttonPlus"

android:layout\_row="6"

android:layout\_column="7" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="4"

android:id="@+id/button4"

android:layout\_row="7"

android:layout\_column="4" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="5"

android:id="@+id/button5"

android:layout\_row="7"

android:layout\_column="5" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="6"

android:id="@+id/button6"

android:layout\_row="7"

android:layout\_column="6" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="-"

android:id="@+id/buttonMin"

android:layout\_row="7"

android:layout\_column="7" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="7"

android:id="@+id/button7"

android:layout\_row="8"

android:layout\_column="4" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="8"

android:id="@+id/button8"

android:layout\_row="8"

android:layout\_column="5" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="9"

android:id="@+id/button9"

android:layout\_row="8"

android:layout\_column="6" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="/"

android:id="@+id/buttonDiv"

android:layout\_row="8"

android:layout\_column="7" />

<Button

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="log"

android:id="@+id/buttonLog"

android:layout\_row="9"

android:layout\_column="7" />

</GridLayout>



RelativeLayout

Это разметка, в которой расположения элементов производится относительно родителя или друг друга. Элементы начинают располагаться в указанном порядке. Необходимо, элемент был описан до того, как другой элемент будет на него ссылаться.

Пользуясь этой разметкой, не забывайте про следующие атрибуты:

**Расположение элемента относительно родительского элемента (Значением атрибута является «true»):**

1. **android:layout\_alignParentTop** — верхняя граница элемента располагается на верхней границе RelativeLayout;
2. **android:layout\_alignParentLeft** — левая граница элемента располагается на левой границе RelativeLayout;
3. **android:layout\_alignParentRight** — правая граница элемента располагается на правой границе RelativeLayout;
4. **android:layout\_alignParentBottom** — нижняя граница элемента располагается на нижней границе RelativeLayout;
5. **android:layout\_centerInParent** — элемент располагается в центре родительского по горизонтали и вертикали;
6. **android:layout\_centerHorizontal** — элемент располагается в центре родительского по горизонтали;
7. **android:layout\_centerVertical** — элемент располагается в центре родительского по вертикали.

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

1. **android:layout\_above** — элемент располагается сверху от другого элемента;
2. **android:layout\_toLeftOf** — элемент располагается слева от другого элемента;
3. **android:layout\_toRightOf** — элемент располагается справа от другого элемента;
4. **android:layout\_below** — элемент располагается снизу от другого элемента;
5. **android:layout\_alignBaseline** — базовая линия элемента выравнивается по базовой линии другого элемента;
6. **android:layout\_alignTop** — верхняя граница элемента выравнивается по верхней границе другого элемента;
7. **android:layout\_alignLeft** — левая граница элемента выравнивается по левой границе другого элемента;
8. **android:layout\_alignRight** — правая граница элемента выравнивается по правой границе другого элемента;
9. **android:layout\_alignBottom** — нижняя граница элемента выравнивается по нижней границе другого элемента.

<RelativeLayout

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" >

<TextView

android:id="@+id/TextView1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_alignParentLeft="true"

android:layout\_toLeftOf="@+id/Button1"

android:text="Просто поле с текстом"

android:textSize="20sp">

</TextView>

<Button

android:id="@+id/Button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignParentRight="true"

android:text="Просто кнопка" >

</Button>

</RelativeLayout>



[Share On Facebook](javascript:)

[Share On Twitter](javascript:)

[Share On Google Plus](javascript:)