Layouts в Android

[admin](http://androiddocs.ru/author/admin/) | 11.04.2014

[0 Comment](http://androiddocs.ru/layouts-v-android/#comments)

Layout определяет внешний вид экрана (а также виджета), его структуру и отдельные компоненты. Когда мы создаем визуальный интерфейс в приложении, то мы как раз работаем с Layouts.

Существует **два основных способа создания Layout**:  
1. Описание в xml-файле  
2. Программный способ создания Layout (так сказать, «на лету»)

Рассмотрим пример создания Layout с помощью xml. Допустим, нам необходимо создать экран (активити), которое будет отображать одну кнопку и один текстовый элемент TextView. В формате xml это выглядело бы так:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | <?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" >  <TextView android:id="@+id/text"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Здесь какой-то текст" />  <Button android:id="@+id/my\_button"  android:layout\_width="wrap\_content"  android:layout\_height="wrap\_content"  android:text="Это кнопка" />  </LinearLayout> |

Как видим, в структуре имеется строка, сообщающая, что это xml версии 1.0. Далее идет группа из двух элементов **TextView** и **Button**, помещенных внутрь тега **LinearLayout**. LinearLayout — это один из видов Layouts (о видах немного позже).

Каждый из упомянутых компонентов имеет атрибуты, задающие определенные свойства (все они начинаются со слова «android:»). Например, атрибут

|  |  |
| --- | --- |
| 1 | android:id="@+id/my\_button" |

задает уникальный идентификатор «my\_button». Благодаря этому идентификатору в дальнейшем в коде мы можем обращаться к нужному элементу по его id. Символ «@» дает понять xml-парсеру, что этот атрибут нужно идентифицировать как ID-ресурс. Значок «+» означает, что это новый ресурс и его необходимо добавить при компиляции в файл ресурсов R.java.

Для обращения к элементу в коде по ID можно воспользоваться такой строчкой:

|  |  |
| --- | --- |
| 1 | Button myButton = (Button) findViewById(R.id.my\_button); |

Также мы используем несколько других атрибутов. В частности, **layout\_width** определяет ширину элемента, а **layout\_height** — его высоту. Значение «wrap\_content» означает, что размеры должны соответствовать внутреннему содержимому, то есть в нашем случае — тексту внутри кнопки или элемента TextView.

Готовый файл разметки помещается в директорию **res/layout/**.

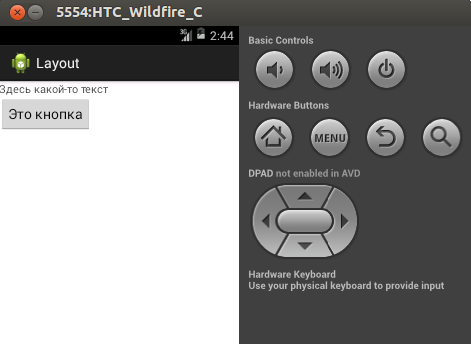
В процессе компиляции этот файл будет скомпилирован в ресурс **View**. Чтобы отобразить данный layout мы должны в нашем активити в методе onCreate() вызвать метод **setContentView()**, передав ему в качестве параметра имя (идентификатор) ресурса layout. Например, если мы сохраним xml-файл как**main.xml**, то вызвать его мы можем следующим образом:

|  |  |
| --- | --- |
| 1 | setContentView(R.layout.main) |

Полное содержимое файла **MainActivity.java**:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | package ru.android.layout;    import android.os.Bundle;  import android.app.Activity;    public class MainActivity extends Activity {  @Override  protected void onCreate(Bundle savedInstanceState) {  super.onCreate(savedInstanceState);  setContentView(R.layout.main);  }  } |

Запустив наше приложение мы получим такой вид экрана:



Существует несколько видов Layout:

[Linear Layout](http://androiddocs.ru/linearlayout/)  
Relative Layout  
Web View  
[List View](http://androiddocs.ru/listview-sozdaem-prostejshij-spisok-s-obrabotkoj-vybora-punkta/)  
Grid View  
[Table Layout](http://androiddocs.ru/tablelayout-v-verstke-ekrana-primer/)

Мы поговорим о каждом более подробно по отдельности.