[https://youtu.be/Yo9XDWgSW-w](https://youtu.be/Yo9XDWgSW-w" \t "_blank)

Основы верстки для нативных андроид приложений

В этом уроке будут рассмотрены вопросы верстки при создании нативных приложений для android. Той верстки, которая описывается в xml файлах из директории res/layout/.

Существует пять стандартных типов верстки:

* AbsoluteLayout
* FrameLayout
* LinearLayout
* RelativeLayout
* TableLayout

AbsoluteLayout

( пользоваться не рекомендуется, deprecated )  
AbsoluteLayout — означает что каждый элемент верстки будет иметь абсолютную позицию относительно верхнего левого угла экрана задаваемую с помощью координат x и y. Т.е. верхний левый угол экрана при AbsoluteLayout имеет координаты x = 0, y = 0.  
Позиция указывается в атрибутах элемента android:layout\_x и android:layout\_y.  
Пример кода:

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

android:layout\_width="fill\_parent"

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

<**Button**

android:id="@+id/backbutton"

android:text="Back"

android:layout\_x="10px"

android:layout\_y="5px"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<**TextView**

android:layout\_x="10px"

android:layout\_y="110px"

android:text="First Name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<**EditText**

android:layout\_x="150px"

android:layout\_y="100px"

android:width="100px"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<**TextView**

android:layout\_x="10px"

android:layout\_y="160px"

android:text="Last Name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<**EditText**

android:layout\_x="150px"

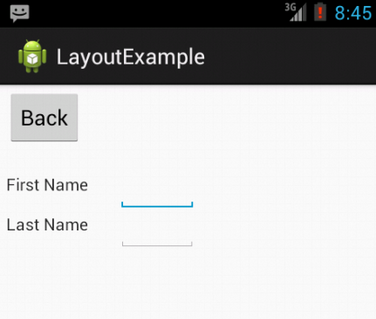
android:layout\_y="150px"

android:width="100px"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

</**AbsoluteLayout**>



FrameLayout

FrameLayout — тип верстки внутри которого может отображаться только один элемент в строке. Т.е. если внутри FrameLayout вы поместите несколько элементов, то следующий будет отображаться поверх предыдущего.  
Пример кода:

<**FrameLayout**

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

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

<**ImageView**

android:src="@drawable/icon"

android:scaleType="fitCenter"

android:layout\_height="fill\_parent"

android:layout\_width="fill\_parent"/>

<**TextView**

android:text="Learn-Android.com"

android:textSize="24sp"

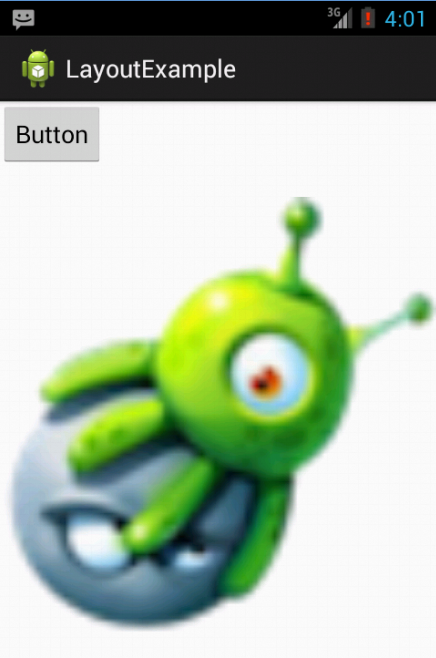
android:textColor="#000000"

android:layout\_height="fill\_parent"

android:layout\_width="fill\_parent"

android:gravity="center"/>

</**FrameLayout**>



LinearLayout

LinearLayout — тип верстки при котором область верстки делится на строки и в каждую строку помещается один элемент. Разбиение может быть вертикальное или горизонтальное, тип разбиения указывается в атрибуте LinearLayout android:orientation. Внутри верстки возможно комбинировать вертикальную и горизонтальную разбивки, а кроме того, возможна комбинация нескольких разных типов верстки например использование LinearLayout внутри FrameLayout.  
  
Пример вертикальной разбивки LinearLayout:

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

android:orientation="vertical"

android:layout\_width="fill\_parent"

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

<**Button**

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**Button**

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**Button**

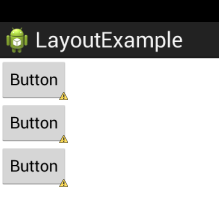
android:id="@+id/button3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

</**LinearLayout**>

  
  
Пример горизонтальной разбивки LinearLayout:

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

android:orientation="horizontal"

android:layout\_width="fill\_parent"

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

<**Button**

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**Button**

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**Button**

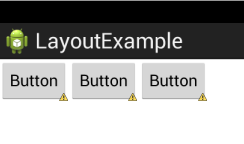
android:id="@+id/button3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

</**LinearLayout**>

  
  
Комбинация нескольких LinearLayout:

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

android:orientation="horizontal"

android:layout\_width="fill\_parent"

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

<**Button**

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**Button**

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**Button**

android:id="@+id/button3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**LinearLayout**

android:orientation="vertical"

android:layout\_width="fill\_parent"

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

<**Button**

android:id="@+id/button1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**Button**

android:id="@+id/button2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

<**Button**

android:id="@+id/button3"

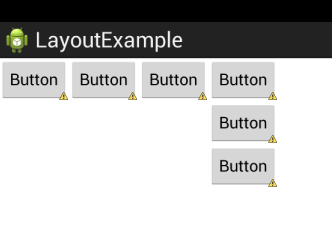
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="Button" />

</**LinearLayout**>

</**LinearLayout**>



RelativeLayout

RelativeLayout — тип верстки при котором позиционирование элементов происходит относительно друг друга и относительно главного контейнера.  
За то, каким образом будут позиционироваться элементы отвечают следующие атрибуты:  
Атрибуты позиционирования относительно контейнера

* android:layout\_alignParentBottom – Низ элемента находится внизу контейнера
* android:layout\_alignParentLeft – Левая часть элемента прилегает к левой части контейнера
* android:layout\_alignParentRight – Правая часть элемента прилегает к правой части контейнера
* android:layout\_alignParentTop – Элемент находится в верхней части контейнера
* android:layout\_centerHorizontal – Элемент позиционируется по центру относительно горизонтального размера контейнера
* android:layout\_centerInParent – Элемент позиционируется по центру относительно горизонтального и вертикального размеров размера контейнера
* android:layout\_centerVertical – Элемент позиционируется по центру относительно вертикального размера контейнера

Атрибуты позиционирования относительно других элементов.  
В качестве значений этих атрибутов ставятся id элемента относительно которого будет производится позиционирование.  
  
android:layout\_above – Распологает элемент над указанным  
android:layout\_below – Распологает элемент под указанным  
android:layout\_toLeftOf – Распологает элемент слева от указанного  
android:layout\_toRightOf – Распологает элемент справа от указанного  
  
Выравнивание относительно других элементов.  
  
android:layout\_alignBaseline – Выравнивает baseline элемента с baseline указаннго элемента  
android:layout\_alignBottom – Выравнивает низ элемента по низу указанного элемента  
android:layout\_alignLeft – Выравнивает левый край элемента с левым краем указанного элемента  
android:layout\_alignRight – Выравнивает правый край элемента с правым краем указанного элемента  
android:layout\_alignTop – Выравнивает верхнюю часть элемента в соответствие с верхней частью указанного элемента

<**RelativeLayout**

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

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

<**Button**

android:id="@+id/backbutton"

android:text="Back"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content" />

<**TextView**

android:id="@+id/firstName"

android:text="First Name"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/backbutton" />

<**EditText**

android:id="@+id/editFirstName"

android:width="100px"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_toRightOf="@id/firstName"

android:layout\_below="@id/backbutton"/>

<**EditText**

android:id="@+id/editLastName"

android:width="100px"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_below="@id/editFirstName"

android:layout\_alignLeft="@id/editFirstName"/>

<**TextView**

android:id="@+id/lastName"

android:text="Last Name"

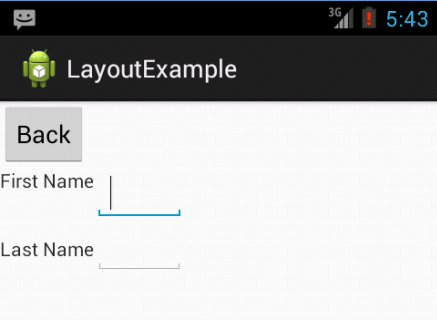
android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:layout\_alignBaseline = "@id/editLastName"

android:layout\_below="@id/editFirstName" />

</**RelativeLayout**>



TableLayout

TableLayout — табличная верстка.  
Организует элементы в строки и столбцы таблицы.  
Для организации строк служит таг

, а количество столбцов определяется максимальным количеством элементов внутри одного из .

В случае если элемент должен занимает несколько ячеек используется атрибут android:layout\_span.

По умолчанию организует строки таблицы, если мы хотим организовывать не строки а столбцы, нужно использовать атрибут android:layout\_column

<**TableLayout**

android:layout\_width="fill\_parent"

android:layout\_height="fill\_parent"

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

<**TableRow**>

<**TextView**

android:id="@+id/textView1"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="column1"

android:textAppearance="?android:attr/textAppearanceMedium" />

<**TextView**

android:id="@+id/textView2"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="column2"

android:textAppearance="?android:attr/textAppearanceMedium" />

<**TextView**

android:id="@+id/textView3"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text="column 3"

android:textAppearance="?android:attr/textAppearanceMedium"/>

</**TableRow**>

<**TableRow**>

<**TextView**

android:id="@+id/textViewSpan"

android:layout\_width="wrap\_content"

android:layout\_height="wrap\_content"

android:text=" span three column"

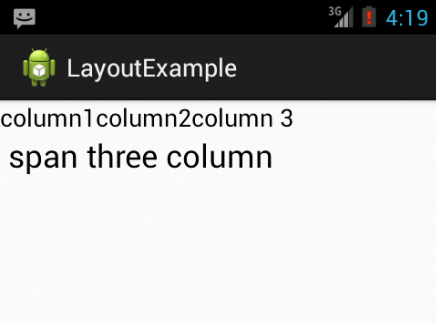
android:layout\_span = "3"

android:textSize="23sp"

android:textAppearance="?android:attr/textAppearanceMedium"/>

</**TableRow**>

</**TableLayout**>



Alternate Layouts

Alternate Layouts - альтернативная верстка. Позволяет использовать различную верстку для различных ориентаций экрана.  
XML для альтернативной верстки помещается в папки проекта:  
  
res/layout-land – альтернативная верстка для landscape UI  
res/layout-port –альтернативная верстка для portrait UI  
res/lauout-square – альтернативная верстка для square UI  
  
и перед тем как получить макет из res/lauout система проверяет наличие файлов в этих папках.  
  
И в завершении немного о стилях.

Стили

Во первых стили элемента могут быть описаны в атрибутах самого элемента.  
Например:

<**TextView**

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:textColor="#00FF00"

android:typeface="monospace"

android:text="@string/hello" />

Кроме того стили можно вынести в отдельный xml файл и сохранить его в папке res/values/  
Напимер:

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

<**resources**>

<**style** name="CodeFont" parent="@android:style/TextAppearance.Medium">

<**item** name="android:layout\_width">fill\_parent</**item**>

<**item** name="android:layout\_height">wrap\_content</**item**>

<**item** name="android:textColor">#00FF00</**item**>

<**item** name="android:typeface">monospace</**item**>

</**style**>

</**resources**>

Если мы вынесем стили в отдельный файл, то для описания стилей элемента будем использовать атрибут style.

<**TextView**

style="@style/CodeFont"

android:text="@string/hello" />

**Задание** разработать по макетам [OECH APP Final (Copy) – Figma](https://www.figma.com/file/2VqVdgyNEPDijiXewIZKfU/OECH-APP-Final-(Copy)?type=design&node-id=0-1&mode=design&t=jiFrcPnGesCSXEai-0) по 3 экрана из 1 и 2 сессии.