**Всё о LinearLayout - 3**

**Тяготение – атрибуты android:layout\_gravity и android:gravity**

Тяготение (gravity) по существу является выравниванием. Например, если нужно выровнять текст справа, то нам нужно присвоить атрибуту gravity значение right, а если снизу, то bottom и т.д. Значения можно группировать, через «побитовое или» |, например, bottom|center.

Приведу табличку с возможными значениями.

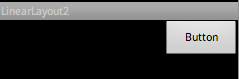
| **XML значение** | **JAVA константа** | **Описание** |
| --- | --- | --- |
| top | Gravity.TOP | Тяготение вверх |
| bottom | Gravity.BOTTOM | Тяготение вниз |
| left | Gravity.LEFT | Тяготение влево |
| right | Gravity.RIGHT | Тяготение вправо |
| center | Gravity.CENTER | Тяготение по центру, как по горизонтальной, так и по вертикально оси |
| center\_horizontal | Gravity.CENTER\_HORIZONTAL | Тяготение по центру горизонтальной оси |
| center\_vertical | Gravity.CENTER\_VERTICAL | Тяготение по центру вертикальной оси |
| clip\_horizontal | Gravity.CLIP\_HORIZONTAL | Place object in the horizontal center of its container, not changing its size. |
| clip\_vertical | Gravity.CLIP\_VERTICAL | Place object in the vertical center of its container, not changing its size. |
| fill | Gravity.FILL | Grow the horizontal and vertical size of the object if needed so it completely fills its container. |
| fill\_horizontal | Gravity.FILL\_HORIZONTAL | Grow the horizontal size of the object if needed so it completely fills its container. |
| fill\_vertical | Gravity.FILL\_VERTICAL | Grow the vertical size of the object if needed so it completely fills its container. |

**Сравнение android:gravity и android:layout\_gravity**

В Android определены два похожих атрибута тяготения – android:gravity и android:layout\_gravity.  
Разница между ними состоит в следующем:

* layout\_gravity используется контейнером компоновки и выравнивает сам элемент на экране.
* gravity используется представлением и занимается выравниванием внутри элемента

Таким образом можно задать у кнопки layout\_gravity равным right, а gravity равным center и получим кнопку, расположенную по правому краю и с текстом в центре.



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

<**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:gravity**=**"center"**

**android:layout\_gravity**=**"right"**

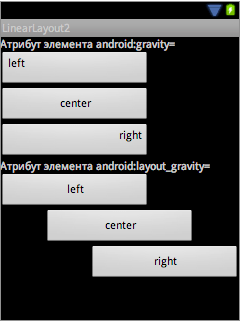
**android:layout\_height**=**"50dp"**

**android:layout\_width**=**"100dp"**

**android:text**=**"Button"**/>

</**LinearLayout**>

Ещё один пример на тему сравнения этих атрибутов:



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

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

**android:orientation**=**"vertical"**

**android:layout\_width**=**"fill\_parent"**

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

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Атрибут элемента android:gravity="**/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"left"**

**android:gravity**=**"left"**

/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"center"**

**android:gravity**=**"center"**

/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"right"**

**android:gravity**=**"right"**

/>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Атрибут элемента android:layout\_gravity="**/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"left"**

**android:layout\_gravity**=**"left"**

/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"center"**

**android:layout\_gravity**=**"center"**

/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

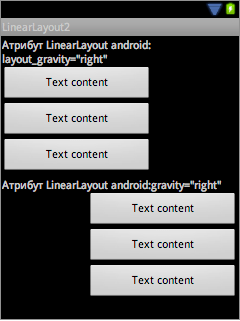
**android:text**=**"right"**

**android:layout\_gravity**=**"right"**

/>

</**LinearLayout**>

Думаю пример ясен без лишних объяснений.  
Двигаемся дальше.  
  
В предыдущих примерах мы использовали атрибуты у элементов представления, а сейчас попробуем применить к контейнеру компоновки LinearLayout.



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

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

**android:orientation**=**"vertical"**

**android:padding**=**"3dp"**

**android:layout\_width**=**"fill\_parent"**

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

<**LinearLayout** **android:orientation**=**"vertical"**

**android:layout\_width**=**"fill\_parent"**

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

**android:layout\_gravity**=**"right"**

**android:layout\_weight**=**"1.0"**>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Атрибут LinearLayout android:layout\_gravity=«right»"**/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Text content"**

/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Text content"**

/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Text content"**

/>

</**LinearLayout**>

<**LinearLayout** **android:orientation**=**"vertical"**

**android:layout\_width**=**"fill\_parent"**

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

**android:gravity**=**"right"**

**android:layout\_weight**=**"1.0"**>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Атрибут LinearLayout android:gravity=«right»"**/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Text content"**

/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Text content"**

/>

<**Button** **android:layout\_width**=**"200dp"**

**android:layout\_height**=**"wrap\_content"**

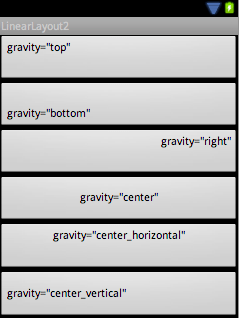
**android:text**=**"Text content"**

/>

</**LinearLayout**>

</**LinearLayout**>

Как видно из картинки, атрибут layout\_gravity в контейнере LinearLayout никак не повлиял на выравнивание элементов, а атрибут gravity выровнял все вложенные элементы по правому краю.  
  
 Приведём ещё 2 примера работы атрибута gravity. Мы будем выравнивать текст внутри кнопки во всевозможных вариациях.



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

<**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:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"top"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«top»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"bottom"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«bottom»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"right"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«right»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"center"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«center»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"center\_horizontal"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«center\_horizontal»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

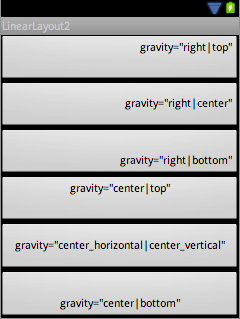
**android:gravity**=**"center\_vertical"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«center\_vertical»"**/>

</**LinearLayout**>

Группировка выравнивания



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

<**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:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"right|top"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«right|top»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"right|center"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«right|center»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"right|bottom"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=»right|bottom»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"center|top"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«center|top»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"center\_horizontal|center\_vertical"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«center\_horizontal|center\_vertical»"**/>

<**Button** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"center|bottom"**

**android:layout\_weight**=**"1"**

**android:text**=**"gravity=«center|bottom»"**/>

</**LinearLayout**>

**Атрибут android:baselineAligned**

Данный атрибут позволяет выравнивать элементы представления по базовой линии.  
Чтобы поскорее понять о чём речь и как это выглядит, прикладываю картинку вместе с кодом.



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

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

**android:orientation**=**"vertical"**

**android:layout\_width**=**"fill\_parent"**

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

>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"center\_horizontal"**

**android:text**=**"baselineAligned false"**/>

<**LinearLayout**

**android:orientation**=**"horizontal"**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:baselineAligned**=**"false"**

>

<**Button**

**android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Hello World"**

/>

<**Button**

**android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"100dp"**

**android:text**=**"Hello World"**

/>

</**LinearLayout**>

<**TextView** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:gravity**=**"center\_horizontal"**

**android:text**=**"baselineAligned true"**/>

<**LinearLayout**

**android:orientation**=**"horizontal"**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:baselineAligned**=**"true"**

>

<**Button**

**android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Hello World"**

/>

<**Button**

**android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"100dp"**

**android:text**=**"Hello World"**

/>

</**LinearLayout**>

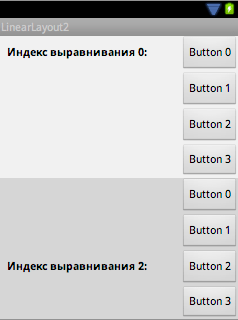
</**LinearLayout**>

По умолчанию этот атрибут установлен в значение true – нижний блок на картинке.  
Если же нужно выровнять элементы внутри контейнера по верхней линии, то задайте атрибуту baselineAligned значение false.  
На JAVA это делается так:

linearLayout.setBaselineAligned(**false**);

**Атрибут android:baselineAlignedChildIndex**

Этот атрибут позволяет установить установить индекс дочернего элемента, по чьей базовой линии будут выравниваться остальные элементы представления. Звучит страшно, но страхи можно быстро развеять примером.



<**LinearLayout**

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

**android:layout\_width**=**"fill\_parent"**

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

**android:layout\_gravity**=**"center\_horizontal"**

**android:background**=**"#fff"**

**android:orientation**=**"vertical"**

>

<**LinearLayout**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:orientation**=**"horizontal"**

**android:baselineAligned**=**"true"**

**android:background**=**"#eee"**

**android:paddingLeft**=**"10dp"**

**android:layout\_weight**=**"1.0"**

>

<**TextView**

**android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_weight**=**"1"**

**android:textColor**=**"#000"**

**android:textStyle**=**"bold"**

**android:gravity**=**"left"**

**android:text**=**"Индекс выравнивания 0:"**

/>

<**LinearLayout**

**android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:orientation**=**"vertical"**

**android:baselineAlignedChildIndex**=**"0"**

>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Button 0"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Button 1"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Button 2"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Button 3"**/>

</**LinearLayout**>

</**LinearLayout**>

<**LinearLayout**

**android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:orientation**=**"horizontal"**

**android:baselineAligned**=**"true"**

**android:background**=**"#ccc"**

**android:paddingLeft**=**"10dp"**

**android:layout\_weight**=**"1.0"**

>

<**TextView**

**android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:layout\_weight**=**"1"**

**android:textColor**=**"#000"**

**android:textStyle**=**"bold"**

**android:gravity**=**"left"**

**android:text**=**"Индекс выравнивания 2:"**

/>

<**LinearLayout**

**android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:orientation**=**"vertical"**

**android:baselineAlignedChildIndex**=**"2"**

>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Button 0"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Button 1"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Button 2"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:text**=**"Button 3"**/>

</**LinearLayout**>

</**LinearLayout**>

</**LinearLayout**>

Что мы сделали? Мы установили текст напротив первой и третьей кнопки.  
Текст в этом случае оказался дочерним элементом, а чтобы выравнивание сработало, мы присвоили контейнеру компоновки LinearLayout, в котором лежит наш текст, индекс кнопки по которой нам нужно выровнять текст. Индекс нужно начинать считать с нуля.  
На языке JAVA присваивание значение делается методом setBaselineAlignedChildIndex(int index).

linearLayout.setBaselineAlignedChildIndex(2); *// установили индекс 2*

**Атрибут android:layout\_margin**

Тем кто знаком с CSS, думаю всё ясно без лишних пояснений.  
Данный атрибут позволяет задать внешний отступ со всех сторон.

| **XML атрибут** | **Назначение** |
| --- | --- |
| layout\_margin | Внешний отступ со всех сторон |
| layout\_marginTop | Внешний отступ сверху |
| layout\_marginBottom | Внешний отступ снизу |
| layout\_marginLeft | Внешний отступ слева |
| layout\_marginRight | Внешний отступ справа |

На JAVA значение задаются во вложенном классе LayoutParams, либо методом setMargins(int left, int top, int right, int bottom), либо задав значение полям leftMargin, rightMargin, topMargin и bottomMargin.

LinearLayout.LayoutParams param = **new** LinearLayout.LayoutParams(

ViewGroup.LayoutParams.WRAP\_CONTENT,

ViewGroup.LayoutParams.WRAP\_CONTENT

);

param.setMargins(10 */\* left \*/*, 10 */\* top \*/*, 15 */\* right \*/*, 15 */\* bottom \*/*);

*// или так:*

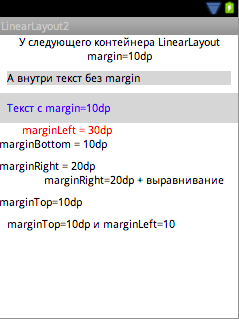
param.leftMargin = 10;

param.topMargin = 10;

param.rightMargin = 15;

param.bottomMargin = 15;

Разберём пример и посмотрим, как это работает:



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

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

**android:background**=**"#fff"**

**android:orientation**=**"vertical"**

**android:layout\_width**=**"fill\_parent"**

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

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:gravity**=**"center\_horizontal"**

**android:text**=**"У следующего контейнера LinearLayout margin=10dp"**/>

<**LinearLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:background**=**"#ccc"**

**android:layout\_margin**=**"10dp"**

**android:orientation**=**"vertical"**>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:text**=**"А внутри текст без margin"**/>

</**LinearLayout**>

<**LinearLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:background**=**"#ccc"**

**android:orientation**=**"vertical"**>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#00f"**

**android:layout\_margin**=**"10dp"**

**android:text**=**"Текст с margin=10dp"**/>

</**LinearLayout**>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#f00"**

**android:layout\_marginLeft**=**"30dp"**

**android:text**=**"marginLeft = 30dp"**/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:layout\_marginBottom**=**"10dp"**

**android:text**=**"marginBottom = 10dp"**/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:layout\_marginRight**=**"20dp"**

**android:text**=**"marginRight = 20dp"**/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:layout\_gravity**=**"right"**

**android:layout\_marginRight**=**"20dp"**

**android:text**=**"marginRight=20dp + выравнивание"**/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:layout\_marginTop**=**"10dp"**

**android:text**=**"marginTop=10dp"**/>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:layout\_marginLeft**=**"10dp"**

**android:layout\_marginTop**=**"10dp"**

**android:text**=**"marginTop=10dp и marginLeft=10"**/>

</**LinearLayout**>

Мы задали второму контейнеру LinearLayout внешний отступ равным 10dp и чтобы были видны границы контейнера, мы задали ему фоновый серый цвет. Как видно из скриншота, контейнер сделал отступ со всех сторон. Далее мы создали ещё один контейнер, но уже без отступов. На этот раз мы задали отступ не контейнеру, а тексту находящемуся внутри и он отступил от границ контейнера. После этого, мы один за другим попробовали свойства отступа отдельно слева, снизу, справа и сверху, а также скомбинировали отступы сверху и слева.

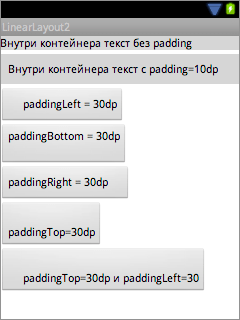
**Атрибут android:padding**

Атрибут padding очень похож на layout\_margin, за тем лишь исключением, что отступы у него не внешние, а внутренние.  
Например, если у нас есть кнопка и мы хотим больше пространства между текстом и границей кнопки, то используем атрибут padding.

| **XML атрибут** | **Назначение** |
| --- | --- |
| padding | Внутренний отступ со всех сторон |
| paddingTop | Внутренний отступ сверху |
| paddingBottom | Внутренний отступ снизу |
| paddingLeft | Внутренний отступ слева |
| paddingRight | Внутренний отступ справа |

В JAVA значения задаются методом setPadding(int left, int top, int right, int bottom)

linearLayout.setPadding(10, 10, 10, 15);



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

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

**android:background**=**"#fff"**

**android:orientation**=**"vertical"**

**android:layout\_width**=**"fill\_parent"**

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

<**LinearLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:background**=**"#ccc"**

**android:layout\_marginBottom**=**"5dp"**

**android:orientation**=**"vertical"**>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:text**=**"Внутри контейнера текст без padding"**/>

</**LinearLayout**>

<**LinearLayout** **android:layout\_width**=**"fill\_parent"**

**android:layout\_height**=**"wrap\_content"**

**android:background**=**"#ccc"**

**android:layout\_marginBottom**=**"5dp"**

**android:orientation**=**"vertical"**>

<**TextView** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:padding**=**"10dp"**

**android:text**=**"Внутри контейнера текст с padding=10dp"**/>

</**LinearLayout**>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:paddingLeft**=**"30dp"**

**android:text**=**"paddingLeft = 30dp"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:paddingBottom**=**"30dp"**

**android:text**=**"paddingBottom = 30dp"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:paddingRight**=**"30dp"**

**android:text**=**"paddingRight = 30dp"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:paddingTop**=**"30dp"**

**android:text**=**"paddingTop=30dp"**/>

<**Button** **android:layout\_width**=**"wrap\_content"**

**android:layout\_height**=**"wrap\_content"**

**android:textColor**=**"#000"**

**android:paddingLeft**=**"30dp"**

**android:paddingTop**=**"30dp"**

**android:text**=**"paddingTop=30dp и paddingLeft=30"**/>

</**LinearLayout**>

Ну вот мы почти и разобрали всю компоновку на простых и понятных примерах.  
  
Исходники этой статьи и [второй части](http://devpad.ru/post/item/3/android-vse-o-linearlayout-2.html) выкладываю на гитхаб - <https://github.com/devpad/Android-LinearLayout-2-3>