**RelativeLayout** — контейнер, в котором дочерние элементы располагаются относительно друг друга или же самого RelativeLayout.

Доступны следующие варианты расположения дочерних View:

* Относительно других дочерних View:
  + layout\_toStartOf — перед View
  + layout\_toEndOf — после View
  + layout\_above — над View
  + layout\_below — под View
  + layout\_alignStart — начинается там же, где начинается View
  + layout\_alignEnd — заканчивается там же, где заканчивается View
  + layout\_alignTop — верхняя граница начинается там же, где у View
  + layout\_alignBottom — нижняя граница начинается там же, где у View
* Относительно RelativeLayout:
  + layout\_alignParentStart — прижимается к началу RelativeLayout
  + layout\_alignParentEnd — прижимается к концу RelativeLayout
  + layout\_alignParentTop — прижимается к верху RelativeLayout
  + layout\_alignParentBottom — прижимается к низу RelativeLayout
  + layout\_centerVertical — располагается в центре по вертикали
  + layout\_centerHorizontal — располагается в центре по горизонтали
  + layout\_centerInParent — располагается в центре RelativeLayout

Позиционирование относительно RelativeLayout

В этом плане RelativeLayout сильно похож на FrameLayout, разве что атрибуты называются иначе.

Создадим такую разметку:

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

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.test.lessons.MainActivity">

<View

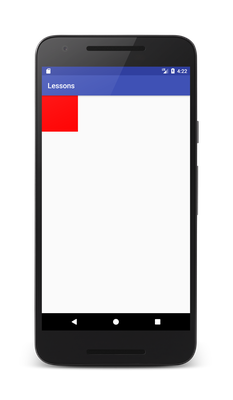
android:layout\_width="100dp"

android:layout\_height="100dp"

android:background="#ff0000" />

</RelativeLayout>

Как видите, View оказался в левом верхнем углу:

[](https://skillberg.com/media/uploads/2017/11/17/6f67610cf2394dccb5877bcc9b041cee.png)Расположение View в RelativeLayout по умолчанию

Теперь попробуем прижать его к правому нижнему углу:

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

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.test.lessons.MainActivity">

<View

android:layout\_width="100dp"

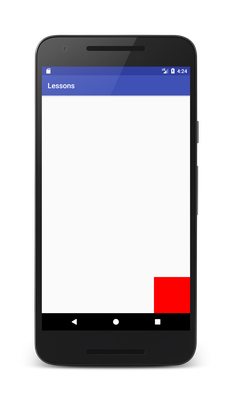
android:layout\_height="100dp"

android:layout\_alignParentBottom="true"

android:layout\_alignParentEnd="true"

android:background="#ff0000" />

</RelativeLayout>

[](https://skillberg.com/media/uploads/2017/11/17/fa52dddbc14b4cc9b7b64f8af67cd3d6.png)Расположение View в правом нижнем углу RelativeLayout

А теперь — справа и по центру по вертикали:

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

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.test.lessons.MainActivity">

<View

android:layout\_width="100dp"

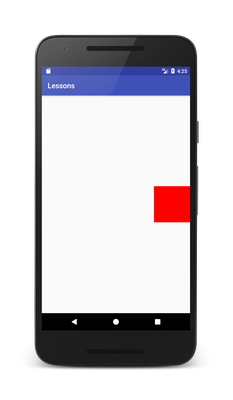
android:layout\_height="100dp"

android:layout\_alignParentEnd="true"

android:layout\_centerVertical="true"

android:background="#ff0000" />

</RelativeLayout>

[](https://skillberg.com/media/uploads/2017/11/17/a427ab49ba1f4d6b908968718372ee71.png)Расположение View справа в центре RelativeLayout

Думаю, тут все понятно. Попробуйте поэкспериментировать самостоятельно.

Я не буду рассказывать про каждый из параметров, а покажу на практике некоторые из них. Но прежде, чем мы начнем, вам нужно узнать еще кое-что об атрибутах в XML-разметке интерфейсов.

Что такое id у View?

Мы с вами уже не первый урок создаем простенькие интерфейсы. Однако, в реальном приложении мало просто сверстать интерфейс — нужно еще и как-то взаимодействовать со View. Вот, допустим, вы хотите что-то сделать с View в лэйауте. А как к нему обратиться?

Идентификатора-то у него никакого нет! К счастью, в Android у View можно указать атрибут android:id, по которому можно будет однозначно идентифицировать конкретный View.

Синтаксис таков:

<View

android:background="#ff0000"

android:id="@+id/red\_view"

android:layout\_alignParentEnd="true"

android:layout\_centerVertical="true"

android:layout\_height="100dp"

android:layout\_width="100dp" />

В начале идет строка @+id/, которая говорит о том, что мы хотим создать новый id. Затем мы пишем его имя.

Идентификатор View имеет следующие ограничения:

* Может содержать латинские буквы, цифры и знак подчеркивания
* Должен быть уникальным в пределах иерархии (не объявляйте два одинаковых id в одной Activity!)

На данный момент это все, что вам нужно знать об id. Теперь давайте продолжим разбираться с RelativeLayout.

Расположение относительно других View

Создадим еще один View, и расположим его слева от первого View:

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

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.test.lessons.MainActivity">

<View

android:id="@+id/red\_view"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_alignParentEnd="true"

android:layout\_centerVertical="true"

android:background="#ff0000" />

<View

android:layout\_width="100dp"

android:layout\_height="100dp"

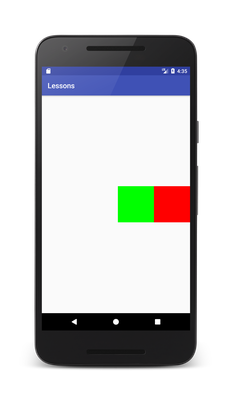
android:layout\_centerVertical="true"

android:layout\_toStartOf="@id/red\_view"

android:background="#00ff00" />

</RelativeLayout>

Как видите, мы используем ссылку на красный View, обращаясь к нему по id: @id/red\_view.

[](https://skillberg.com/media/uploads/2017/11/17/3c19386ecbe74ebb9c1ffc74aa54b29d.png)Зелёный View левее красного View

Теперь красный по центру, а зеленый слева над ним:

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

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.test.lessons.MainActivity">

<View

android:id="@+id/red\_view"

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_centerInParent="true"

android:background="#ff0000" />

<View

android:layout\_width="100dp"

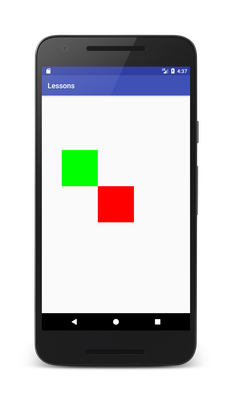
android:layout\_height="100dp"

android:layout\_above="@id/red\_view"

android:layout\_toStartOf="@id/red\_view"

android:background="#00ff00" />

</RelativeLayout>

[](https://skillberg.com/media/uploads/2017/11/17/049a5f7b82064c8995973e6524d9e5b3.png)Красный View в центре, зелёный — левее и выше

А теперь сделаем вот так:

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

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

xmlns:tools="http://schemas.android.com/tools"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

tools:context="com.test.lessons.MainActivity">

<View

android:id="@+id/red\_view"

android:layout\_width="200dp"

android:layout\_height="200dp"

android:layout\_centerInParent="true"

android:background="#ff0000" />

<View

android:layout\_width="100dp"

android:layout\_height="100dp"

android:layout\_above="@id/red\_view"

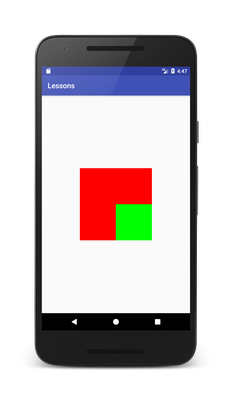
android:layout\_alignBottom="@id/red\_view"

android:layout\_alignEnd="@id/red\_view"

android:background="#00ff00" />

</RelativeLayout>

И зеленый квадратик окажется "внутри" красного:

[](https://skillberg.com/media/uploads/2017/11/17/77f985a089444d5ca1ebc78da1764343.png)Зелёный View снизу справа "внутри" красного

На этом все, а в следующем уроке я покажу, как взаимодействовать с элементами интерфейса из Java-кода!

ПРИМЕРЫ КОДА

[ИСХОДНЫЙ КОД УРОКА](https://github.com/Skillberg/LessonsAndroid/tree/lesson6)