**как сделать так, чтобы высоты и ширины этих блоков были в процентах**

LinearLayout c параметром layout\_weight. например:

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

<LinearLayout

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

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:orientation="vertical">

<Button

android:id="@+id/button1"

android:layout\_height="0dp"

android:layout\_width="match\_parent"

android:text="B1"

android:layout\_weight="1">

</Button>

<Button

android:id="@+id/button2"

android:layout\_width="match\_parent"

android:layout\_height="0dp"

android:text="B2"

android:layout\_weight="3">

</Button>

<Button

android:layout\_height="0dp"

android:layout\_width="match\_parent"

android:id="@+id/button3"

android:text="B3"

android:layout\_weight="2">

</Button>

</LinearLayout>

Из нативных разметок Android задать процентное отношение (точнее долю в контейнере) между виджетами позволяет атрибут layout\_weight (вес виджета).

* Атрибут применим к контейнерам LinearLayout, TableLayout.  
  Атрибут позволяет указать долю только "основного" размера виджета в контейнере. Например для LinearLayout с атрибутом android:orientaion = "vertical" таким образом можно задать только высоту элементов (для горизонтальной ориентации соответственно только ширину). Виджету, для которого указывается "вес" значение "основого" размера при этом указывается *0dp* (для вертикальной ориентации android:layout\_height="0dp"), так как данное значение не будет напрямую использоваться при расчетах. Пример (два виджета делят высоту контейнера пополам (по 50% высоты каждому):
* <LinearLayout
* xmlns:android="http://schemas.android.com/apk/res/android"
* android:layout\_width="match\_parent"
* android:layout\_height="match\_parent"
* android:orientation="vertical">
* <ImageView
* android:layout\_width="match\_parent"
* android:layout\_height="0dp"
* android:id="@+id/imageView"
* android:layout\_weight="50" />
* <LinearLayout
* android:orientation="vertical"
* android:layout\_width="match\_parent"
* android:layout\_height="0dp"
* android:layout\_weight="50">
* </LinearLayout>
* </LinearLayout>

Если требуется указать долю в контенере не для всех виджетов этого контейнера, то к контейнеру применяется атрибут android:weightSum:

<LinearLayout

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:weghtSum="2">

<Button

android:id="@+id/button"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_weight="1" />

</LinearLayout>

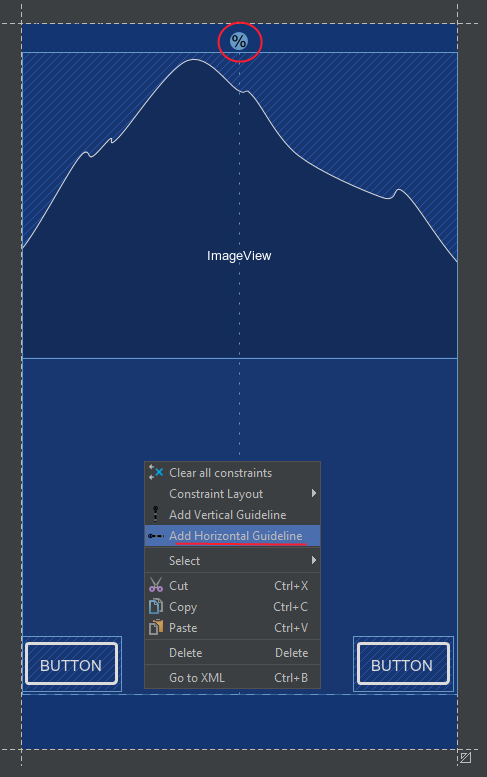
Данный атрибут указывает сколько долей всего в контейнере (в примере - две доли всего), а атрибут layout\_weight виджета сколько из этих долей приходится на виджет (в примере - одна доля), что равносильно 50% высоты контейнера. Оставшиеся 50% высоты будут разделены между другими виджетами согласно их атрибутов размеров (в примере пространство просто останется пустым).

Значение самого атрибута веса может быть любым, доля будет рассчитана исходя из их отношения между собой (например значения 50\50, 1\1, 7\7 эквивалентны и поделят имеющееся пространство поровну). Рекомендуется использовать простые числа, облегчающие расчеты.

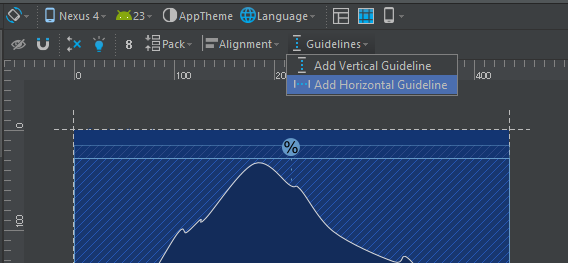
Данный инструмент имеет очень малые возможности и существует более "серьезные" контейнеры, но они доступны в виде библиотек поддержки.

* PercentRelativeLayout и PercentFrameLayout (библиотека поддержки *android.support:percent*) - контейнеры имеют все те же атрибуты, что и их "родители", кроме того появилась возможность указывать отступы, расстояния и размеры для вложенных в контейнер виджетов в процентах от размеров самого контейнера. Здесь в процентном отношении можно указать намного больше атрибутов.  Возможная разметка (Здесь высота и ширина виджета 50%, отступы слева и сверху по 25% размера контейнера.:
* <android.support.percent.PercentRelativeLayout
* xmlns:android="http://schemas.android.com/apk/res/android"
* xmlns:app="http://schemas.android.com/apk/res-auto"
* xmlns:tools="http://schemas.android.com/tools"
* android:layout\_width="match\_parent"
* android:layout\_height="match\_parent">
* <View
* android:id="@+id/first"
* app:layout\_heightPercent="50%"
* app:layout\_widthPercent="50%"
* app:layout\_marginLeftPercent="25%"
* app:layout\_marginTopPercent="25%" />
* </android.support.percent.PercentRelativeLayout>
* ConstrainLayout (библиотека поддержки *android.support.constraint:constraint-layout*), очень гибкий контейнер, который позволяет в том числе работать и с процентами через опорные линии (Guideline).

Начинаем работу с того, что задаем две опорные линии. Для этого кликаем правой кнопкой мыши по области редактирования и выбираем *Add Vertical Guidline* для вертикальной и *Add Horizontal Guidline* для горизонтальной:

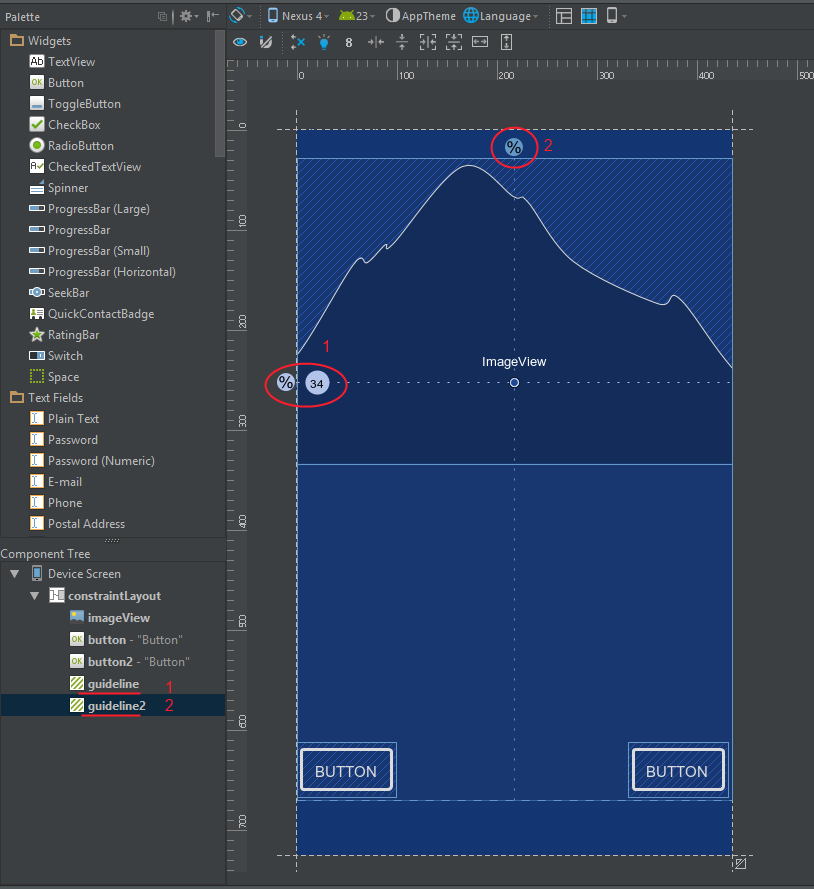
[](https://i.stack.imgur.com/I0pB2.png)

**UPD:** C *Android Studio 2.2 Preview3* опорные линии можно задать из панели инструментов вверху редактора:

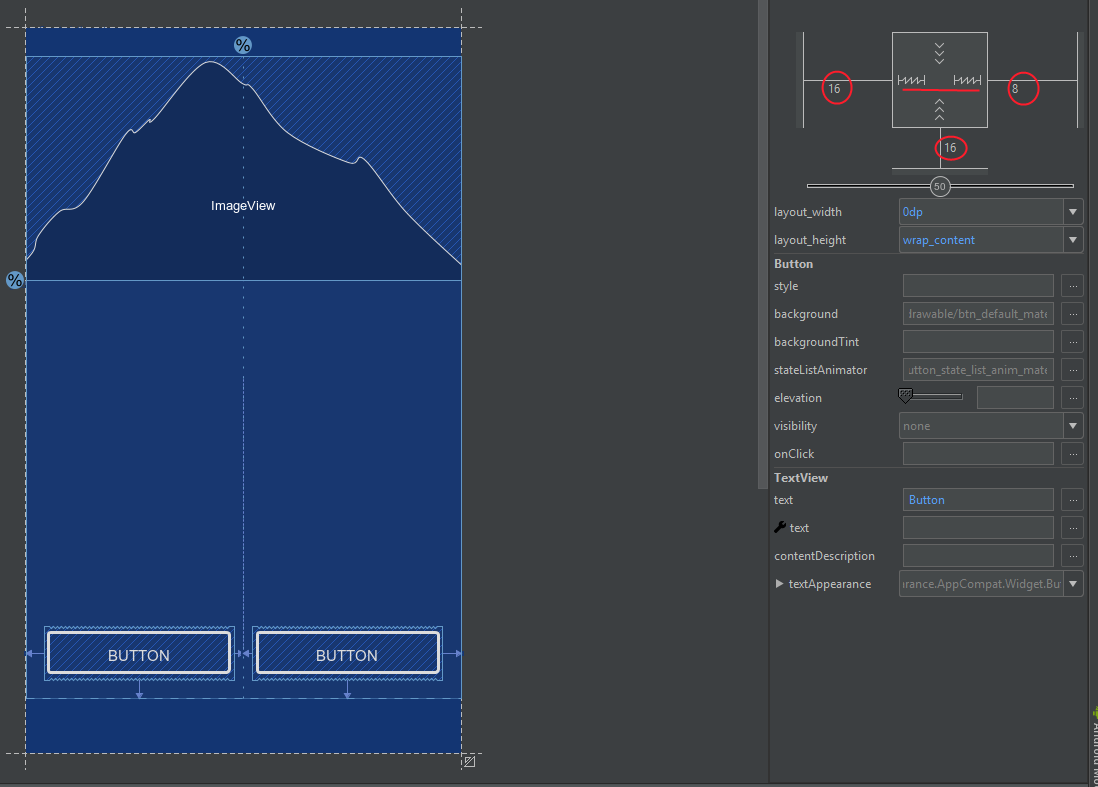
[](https://i.stack.imgur.com/F1dCI.png)

Отключаем автоматическое связывание виджетов (подковка вверх с синими концами, в левом верхнем углу, должна быть зачеркнута), так как эта функция не всегда хорошо понимает, что хочет человек.

По умолчанию опорная линия позиционируется по относительным координатам (расстоянию в *dp*), о чем сообщает изображение стрелочки в кружочке этой линии. Но нам нужно относительное позиционирование в процентах, поэтому кликаем мышкой на эту стрелочку в кружочке и она приобретает вид с процентами - двигаем за этот кружочек в нужные позиции: вертикальную устанавливаем в 50%, а горизонтальную устанавливаем в 34% (треть экрана), затем бросаем на разметку три виджета, которые нам нужны (ImageView и две Button):

[](https://i.stack.imgur.com/Djfq7.png)

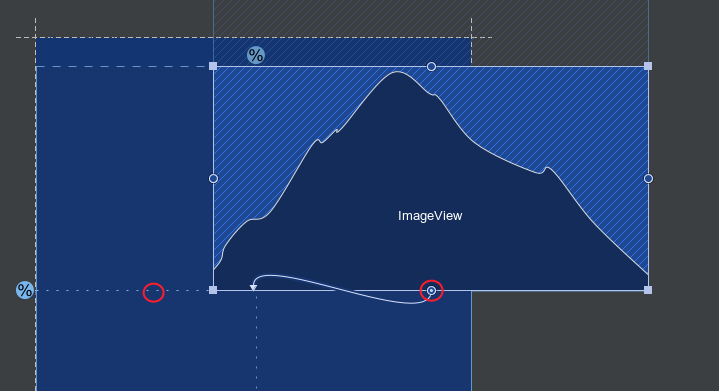
Теперь привязываем к этим линиям наши виджеты, кликаем на виджет и соединяем кружочки на их гранях с привязочными линиями: кнопки - левую кнопку с левого края с краем экрана и нижнюю точку с низом экрана, правый кружок с вертикальной опорной линией, правую кнопку так же, только в зеркально (**UPD** С версии 1.0 появился [режим компоновки chains](https://ru.stackoverflow.com/a/669212/177345) (цепочки), который позволяет обойтись без вертикальной опорной линии):

[](https://i.stack.imgur.com/CJbe9.png)

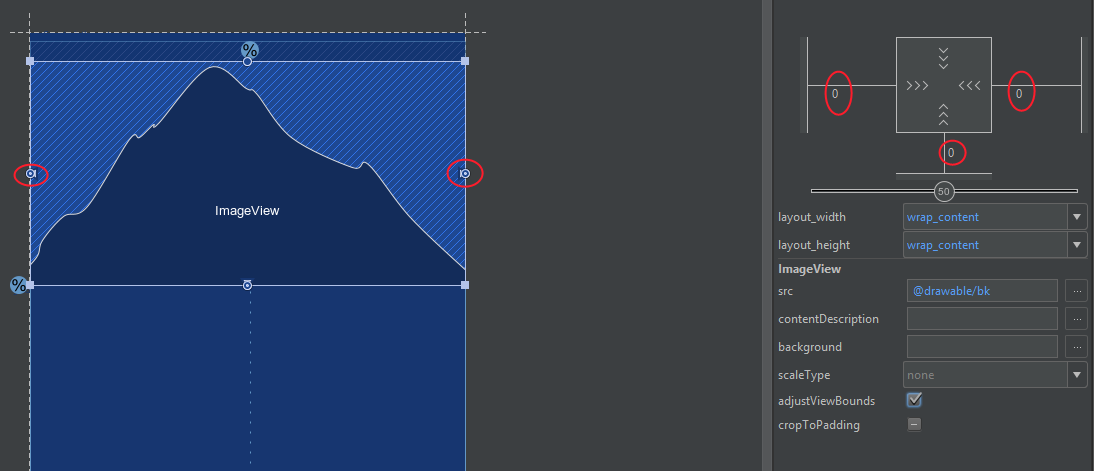
Размер кнопкам по горизонтали назначаем - максимальное расстояние (аналог match\_parent), в квадрате справа щелкаем по горизонтальным линиям, пока не появится в виде "пилы" - [введите сюда описание изображения](https://i.stack.imgur.com/QMgnK.png). (треугольнички - wrap\_content, отрезок - фиксированный размер). Вертикальный оставляем, как есть (в виде треугольничков - wrap\_content). Задаем маржины, числа сбоку от пиктограмм размера виджета (здесь 16dp и 8dp).

С кнопками все.

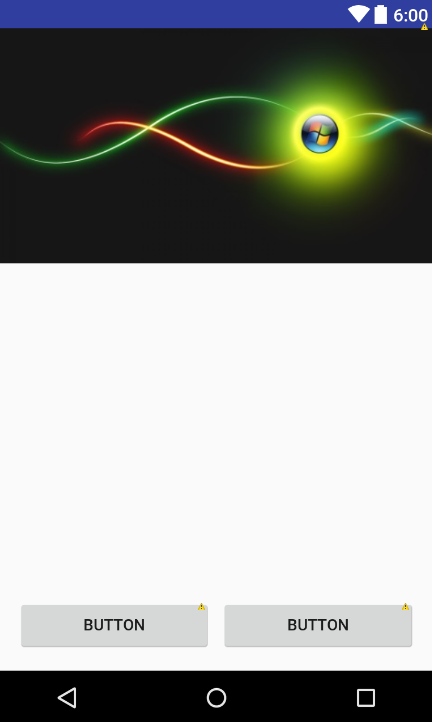
Теперь нижний край (кружок снизу) у ImageView соединяем с горизонтальной опорной линией. Прочие края соединяем с краями экрана:

[](https://i.stack.imgur.com/blsOu.png)

ставим маржины у ImageView в 0, размер виджета wrap\_content:

[](https://i.stack.imgur.com/O7pOK.png)

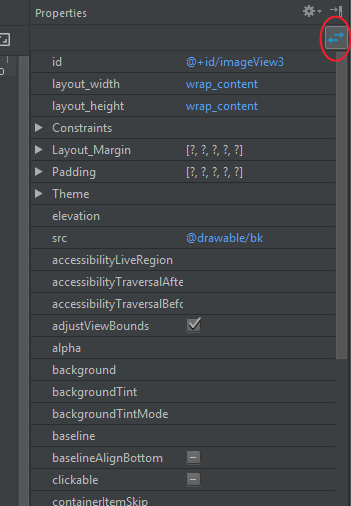
Вот и все. Описание получилось очень длинным, но в реальности эта верстка заняла меньше минуты. В итоге получаем следующее:

[](https://i.stack.imgur.com/C7J9R.png)

Данная разметка будет сохранять свой относительный вид на любых размерах экранов с любой плотностью.

Так же следует заметить, что ConstraintLayout предпочтительнее RelativeLayout, так как гораздо лучше оптимизирован и рассчитывается быстрее, кроме того имеет больше возможностей по позиционированию. Например, в данной разметке пришлось бы вообще использовать несколько вложенных "классических" контейнеров.

Если виджету требуется указать дополнительные свойства, то кликаем на пиктограмму разнонаправленных стрелочек в верхнем правом углу редактора - появятся все атрибуты, доступные для виджета:

[](https://i.stack.imgur.com/oLKf3.png)