**ImageView** — View, предназначенный для отображения изображений.

Существует несколько способов загрузки изображений в ImageView:

* Из ресурсов приложения в XML или из Java-кода
* Установкой Bitmap
* Установкой Drawable
* С использованием URI

Для начала давайте добавим ImageView в XML:

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

<LinearLayout 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"

android:orientation="vertical"

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

<ImageView

android:id="@+id/image\_iv"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

</LinearLayout>

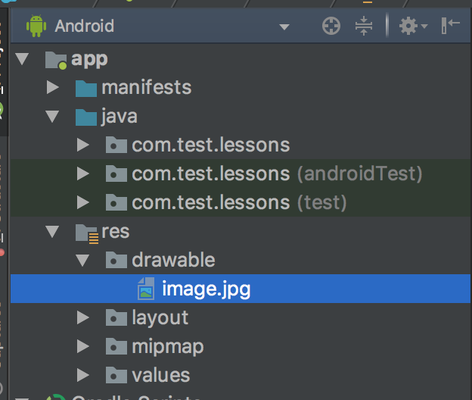
Если вы сейчас запустите приложение, то увидите, что ничего не отображается — изображение-то мы не задали!

Давайте по порядку рассмотрим различные способы.

Установка изображения из ресурсов

Самый простой способ установить изображение — ссылкой на ресурс из XML-верстки. Для этого у ImageViewесть атрибут android:src.

Однако, для начала нужно добавить изображение в ресурсы. Возьмите изображение (например, из Git-репозитория к этому уроку) и скопируйте его в ресурсы в директорию drawable:

[](https://skillberg.com/media/uploads/2017/11/17/019c0b345b5b4ed1a5d93cf4f48ab1a6.png)Изображение в директории Drawable

Теперь мы можем сослаться на это изображение в атрибуте android:src у нашего ImageView:

<ImageView

android:id="@+id/image\_iv"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:src="@drawable/image" />

Механизм примерно тот же, что и со строками, только вместо @string мы используем @drawable, а затем имя файла (без расширения).

Запустив приложение, вы увидите, что изображение отображается:

[](https://skillberg.com/media/uploads/2017/11/17/d37d1f021a6945dcbcac4d12177db397.png)Отображение картинки в ImageView

Теперь давайте установим это же изображение, но уже из Java-кода. Удалите атрибут android:src:

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

<LinearLayout 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"

android:orientation="vertical"

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

<ImageView

android:id="@+id/image\_iv"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent" />

</LinearLayout>

В коде MainActivity "найдите" ImageView:

package com.test.lessons;

import android.os.Bundle;

import android.support.v7.app.AppCompatActivity;

import android.widget.ImageView;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ImageView imageView = findViewById(R.id.image\_iv);

}

}

Чтобы установить изображение, нужно вызвать метод ImageView.setImageResource(int resId). В качестве параметра, соответственно, нужно передать id ресурса.

В нашем случае это будет выглядеть следующим образом:

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ImageView imageView = findViewById(R.id.image\_iv);

imageView.setImageResource(R.drawable.image);

}

Запустив приложение, вы увидите, что ничего не изменилось (ведь этот метод эквивалентен предыдущему):

[](https://skillberg.com/media/uploads/2017/11/17/d37d1f021a6945dcbcac4d12177db397.png)Изображение в ImageView, установленное из Java-кода

Установка Bitmap

Bitmap — это представление изображения в памяти.

Объект Bitmap хранит в себе информацию о каждом пикселе изображения.

**Внимание**: Bitmap может занимать **очень** много места в памяти! Файлы изображений обычно хранятся в форматах со сжатием (JPG/PNG), и поэтому занимают относительно мало места. Bitmap же хранит информацию о каждом пикселе, поэтому может "весить" в десятки/сотни раз больше, чем файл с изображением.

Итак, мы хотим установить изображение, используя Bitmap.

Сначала нужно его откуда-то взять. Самый простой способ — использовать метод BitmapFactory.decodeResource(), дабы создать Bitmap из ресурса.

Он принимает объект ресурсов и id ресурса.

Давайте загрузим наше изображение этим методом:

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ImageView imageView = findViewById(R.id.image\_iv);

Bitmap bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.image);

}

А теперь просто установим его в ImageView, используя метод ImageView.setImageBitmap():

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ImageView imageView = findViewById(R.id.image\_iv);

Bitmap bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.image);

imageView.setImageBitmap(bitmap);

}

Результат, как и в прошлый раз, не изменится.

Установка Drawable

Drawable — это объект, который может быть отрисован в каком-либо View. Drawable не обязательно является изображением — это может быть, например, форма (shape), созданная в ресурсах, или комбинация изображений.

Предположим, что мы хотим установить в ImageView ту же самую картинку, что и раньше, но используя Drawable. В этом нам поможет класс-"обертка" над Bitmap — BitmapDrawable.

Чтобы создать BitmapDrawable, используйте конструктор, принимающий объект ресурсов и Bitmap:

BitmapDrawable bitmapDrawable = new BitmapDrawable(getResources(), bitmap);

И после этого метод ImageView.setImageDrawable():

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ImageView imageView = findViewById(R.id.image\_iv);

Bitmap bitmap = BitmapFactory.decodeResource(getResources(), R.drawable.image);

BitmapDrawable bitmapDrawable = new BitmapDrawable(getResources(), bitmap);

imageView.setImageDrawable(bitmapDrawable);

}

Готово!

Масштабирование изображения

Существуют различные способы масштабирования изображения в ImageView. За способ масштабирования отвечает атрибут android:scaleType и метод ImageView.setScaleType(). Он принимает следующие значения:

* center — устанавливает изображение в центре ImageView, но не масштабирует его.
* centerCrop — равномерно (сохраняя пропорции) масштабирует изображение, высота и ширина изображения будут больше или равны высоте и ширине ImageView. После этого устанавливает изображение в центре ImageView.
* centerInside — равномерно (сохраняя пропорции) масштабирует изображение, высота и ширина изображения будут меньше или равны высоте и ширине ImageView. После этого устанавливает изображение в центре ImageView.
* fitCenter — масштабирует изображение, сохраняя пропорции, при этом хотя бы одна из сторон будет равна соответствующей стороне ImageView. После этого устанавливает его в центре ImageView.
* fitEnd — масштабирует изображение, сохраняя пропорции, при этом хотя бы одна из сторон будет равна соответствующей стороне ImageView. После этого устанавливает его в правом нижнем углу ImageView.
* fitStart — масштабирует изображение, сохраняя пропорции, при этом хотя бы одна из сторон будет равна соответствующей стороне ImageView. После этого устанавливает его в левом верхнем углу ImageView.
* fitXY — масштабирует каждую из сторон изображения независимо (то есть не сохраняя пропорции) таким образом, чтобы стороны совпадали со сторонами ImageView.
* matrix — масштабирует изображение, используя матрицу (задаваемую методом ImageView.setImageMatrix()).

Как видите, вариантов масса. Давайте попробуем каждый из них.

В первую очередь, удалите Java-код, написанный ранее, чтобы метод onCreate() выглядел вот так:

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

}

Теперь измените XML-лэйаут нашего Activity следующим образом:

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

<LinearLayout 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"

android:orientation="vertical"

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

<ImageView

android:id="@+id/image\_iv"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#990000"

android:src="@drawable/image" />

</LinearLayout>

Вы могли заметить, что я добавил атрибут android:background с красным цветом в ImageView. Я сделал это для того, чтобы было наглядно видно размеры ImageView и позицию изображения внутри него. Запустите приложение:

[](https://skillberg.com/media/uploads/2017/11/17/7a09a214d90a4794bfa807ae84224489.png)Красный фон ImageView

Супер! Теперь давайте экспериментировать.

center

Установим значение center в android:scaleType:

<ImageView

android:id="@+id/image\_iv"

android:layout\_width="match\_parent"

android:layout\_height="match\_parent"

android:background="#990000"

android:scaleType="center"

android:src="@drawable/image" />

[](https://skillberg.com/media/uploads/2017/11/17/00c6e999752348eb9b41a6728c76f337.png)android:scaleType = center

Как видите, изображение заняло весь ImageView, при этом видно лишь его малую (центральную) часть. Так произошло, потому что конкретно это изображение достаточно большого размера (оно существенно больше, чем ImageView). Если бы мы установили изображение меньшего чем ImageView размера, оно было бы в центре, при этом занимая только часть пространства.

centerCrop

[](https://skillberg.com/media/uploads/2017/11/17/5c096f79594648ceb3ab23ef8dd8880e.png)android:scaleType = centerCrop

Казалось бы, ничего не изменилось, но, если приглядеться, можно заметить, что на самом деле теперь изображение по высоте равно высоте ImageView. При этом соответственно изменилась ширина, и по ширине изображение по-прежнему не влезает в экран.

centerInside

[](https://skillberg.com/media/uploads/2017/11/17/5d294b0d04774da68910e5536ddedb48.png)android:scaleType = centerInside

Ну а это вариант, используемый по-умолчанию — теперь ширина изображения равна ширине ImageView, а высота уменьшилась пропорционально. Здесь изображение полностью влезло в ImageView.

fitCenter

[](https://skillberg.com/media/uploads/2017/11/17/c26f7ecabce448d88f621ba238041058.png)android:scaleType = fitCenter

Результат в этом случае ничем не отличается от centerInside

fitEnd

[](https://skillberg.com/media/uploads/2017/11/17/d16eb1d888eb4720bd75e6d15d593782.png)android:scaleType = fitEnd

То же самое, но изображение "прижалось" к низу экрана (на самом деле, к правому нижнему углу, но это не заметно из-за того, что ширина изображения и ImageView совпадают).

fitStart

[](https://skillberg.com/media/uploads/2017/11/17/67c7f9f7220b41698ca7773aeee7a7e9.png)android:scaleType = fitStart

То же самое, но теперь в левом верхнем углу экрана.

fitXY

[](https://skillberg.com/media/uploads/2017/11/17/b6f18af24be14e6486e6875662d53026.png)android:scaleType = fitXY

А теперь размеры изображения равны размерам ImageView. Соответственно, поскольку пропорции ImageView и изображения не совпадают, пропорции картинки изменились, и она выглядит вертикально растянутой.

AdjustViewBounds

Стоит обязательно обратить внимание на атрибут android:adjustViewBounds. Если выставить значение, равное true, то ImageView будет "подгонять" свои размеры под размеры изображения. Для наглядности давайте посмотрим на поведение по умолчанию. Возьмем вот такой layout:

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

<LinearLayout 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"

android:orientation="vertical"

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

<ImageView

android:id="@+id/image\_iv"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:background="#990000"

android:src="@drawable/image" />

</LinearLayout>

Он даст такой результат:

[](https://skillberg.com/media/uploads/2017/11/17/f2a1f0cb2d6a404d922034c8086aa5e1.png)android:adjustViewBounds = false

Как видите, несмотря на то, что высота ImageView равна wrap\_content (т.е. должна соответствовать высоте контента), он все равно занимает весь контейнер по высоте. А теперь установим android:adjustViewBounds в true:

<ImageView

android:id="@+id/image\_iv"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:adjustViewBounds="true"

android:background="#990000"

android:src="@drawable/image" />

Результат:

[](https://skillberg.com/media/uploads/2017/11/17/809391459c8843578a46c3e450be4baf.png)android:adjustViewBounds = true

Теперь ImageView занимает ровно столько места, сколько контент внутри него.

На этом, пожалуй, все, что на данный момент нужно знать об ImageView. Кажется, пора приступать к практическим урокам!

ПРИМЕРЫ КОДА

[ИСХОДНЫЙ КОД УРОКА](https://github.com/Skillberg/LessonsAndroid/tree/lesson10)

ДОМАШНЕЕ ЗАДАНИЕ

1. Поэкспериментируйте со scaleType, используя маленькое изображение.<br/>2. Установите ScaleType, используя Java-код.