ImageView

#### **Attributes of ImageView:**

Now let’s  we discuss some important attributes that helps us to configure a ImageView in your xml file.

**1. id:** id is an attribute used to uniquely identify a image view in android. Below is the example code in which we set the id of a image view.

<ImageView

**android:id="@+id/simpleImageView"**

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

/>

**2. src:** src is an attribute used to set a source file or you can say image in your imageview to make your layout attractive.

Below is the example code in which we set the source of a imageview lion which is saved in drawable folder.

<ImageView

android:id="@+id/simpleImageView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

**android:src="@drawable/lion"** /><!--set the source of an image view-->

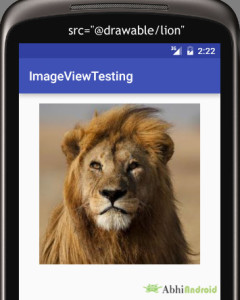
**In Java:**

We can also set the source image at run time programmatically in java class. For that we use setImageResource() [method](http://abhiandroid.com/java/method) as shown in below example code.

/\*Add in Oncreate() funtion after setContentView()\*/

ImageView simpleImageView=(ImageView) findViewById(R.id.simpleImageView);

simpleImageView.setImageResource(R.drawable.lion);//set the source in java class



**3. background:** background attribute is used to set the background of a ImageView. We can set a color or a drawable in the background of a ImageView.

Below is the example code in which we set the black color in the background and an image in the src attribute of image view.

<ImageView

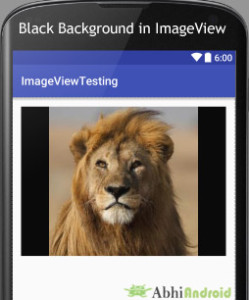
android:id="@+id/simpleImageView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:src="@drawable/lion"

**android:background="#000"**/><!--black color in background of a image view-->



**In Java:**

We can also set the background at run time programmatically in java class. In below example code we set the black color in the background of a image view.

/\*Add in Oncreate() funtion after setContentView()\*/

ImageView simpleImageView=(ImageView) findViewById(R.id.simpleImageView);

simpleImageView.setBackgroundColor(Color.BLACK);//set black color in background of a image view in java class

**4. padding:**padding attribute is used to set the padding from left, right, top or bottom of the Imageview.

* **paddingRight:** set the padding from the right side of the image view**.**
* **paddingLeft:** set the padding from the left side of the image view**.**
* **paddingTop:** set the padding from the top side of the image view**.**
* **paddingBottom:** set the padding from the bottom side of the image view**.**
* **padding:** set the padding from the all side’s of the image view**.**

Below is the example code of padding attribute in which we set the 30dp padding from all the side’s of a image view.

<ImageView

android:id="@+id/simpleImageView"

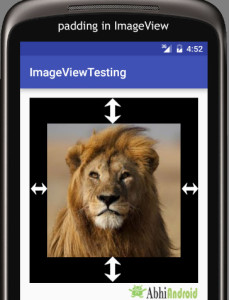
android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:background="#000"

android:src="@drawable/lion"

**android:padding="30dp"**/><!--set 30dp padding from all the sides-->



**5. [scaleType](http://abhiandroid.com/ui/scaletype-imageview-example.html" \t "_self):**scaleType is an attribute used to control how the image should be re-sized or moved to match the size of this image view. **The value for scale type attribute can be fit\_xy, center\_crop, fitStart etc.**

Below is the example code of scale type in which we set the scale type of image view to fit\_xy.

<ImageView

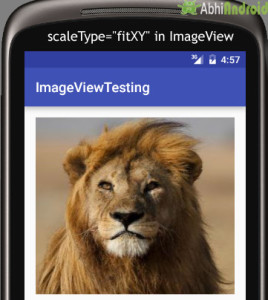
android:id="@+id/simpleImageView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:src="@drawable/lion"

**android:scaleType="fitXY"**/><!--set scale type fit xy-->



**Let’s we take an another example of scale type to understand the actual working of scale type in a image view.**

In below example code we set the value for scale type “fitStart”  which is used to fit the image in the start of the image view as shown below:

<ImageView

android:id="@+id/simpleImageView"

android:layout\_width="fill\_parent"

android:layout\_height="wrap\_content"

android:src="@drawable/lion"

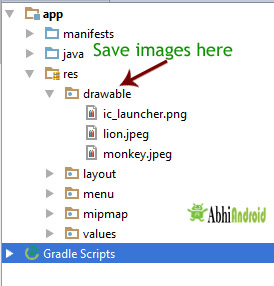
**android:scaleType="fitStart"**/><!--set scale type fit start of image view-->

**Step 1:** Create a new project and name it ImageViewExample.

In this step we create a new project in android studio by filling all the necessary details of the app like app name, package name, api versions etc.

Select File -> New -> New Project and Fill the forms and click "Finish" button.

**Step 2:** Download two images lion and monkey from the web. Now save those images in the drawable folder of your project.



**Step 3:** Now open res -> layout -> activity\_main.xml (or) main.xml and add following code:

In this step we add the code for displaying an image view on the screen in a relative layout. **Here make sure you have already saved two images name lion and monkey in your drawable folder.**

<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"

android:paddingBottom="@dimen/activity\_vertical\_margin"

android:paddingLeft="@dimen/activity\_horizontal\_margin"

android:paddingRight="@dimen/activity\_horizontal\_margin"

android:paddingTop="@dimen/activity\_vertical\_margin"

tools:context=".MainActivity">

<ImageView

android:id="@+id/simpleImageViewLion"

android:layout\_width="fill\_parent"

android:layout\_height="200dp"

android:scaleType="fitXY"

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

<ImageView

android:id="@+id/simpleImageViewMonkey"

android:layout\_width="fill\_parent"

android:layout\_height="200dp"

android:layout\_below="@+id/simpleImageViewLion"

android:layout\_marginTop="10dp"

android:scaleType="fitXY"

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

</RelativeLayout>

**Step 4:** Now open app -> java -> package -> MainActivity.java and add the following code:

In this step we add the code to initiate the image view’s and then perform click event on them.

package example.abhiandriod.imageviewexample;

import android.graphics.Color;

import android.support.v7.app.AppCompatActivity;

import android.os.Bundle;

import android.view.Menu;

import android.view.MenuItem;

import android.view.View;

import android.widget.ImageView;

import android.widget.Toast;

public class MainActivity extends AppCompatActivity {

@Override

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

ImageView simpleImageViewLion = (ImageView) findViewById(R.id.simpleImageViewLion);//get the id of first image view

ImageView simpleImageViewMonkey = (ImageView) findViewById(R.id.simpleImageViewMonkey);//get the id of second image view

simpleImageViewLion.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Toast.makeText(getApplicationContext(), "Lion", Toast.LENGTH\_LONG).show();//display the text on image click event

}

});

simpleImageViewMonkey.setOnClickListener(new View.OnClickListener() {

@Override

public void onClick(View view) {

Toast.makeText(getApplicationContext(), "Monkey", Toast.LENGTH\_LONG).show();//display the text on image click event

}

});

}

}

**Output:**

Now start AVD in Emulator and run the App. You will see the images of Lion and Monkey displayed on screen. Click on any Animal image and his name will appear on Screen. We clicked on Lion.

