**Animation in Android with Example**

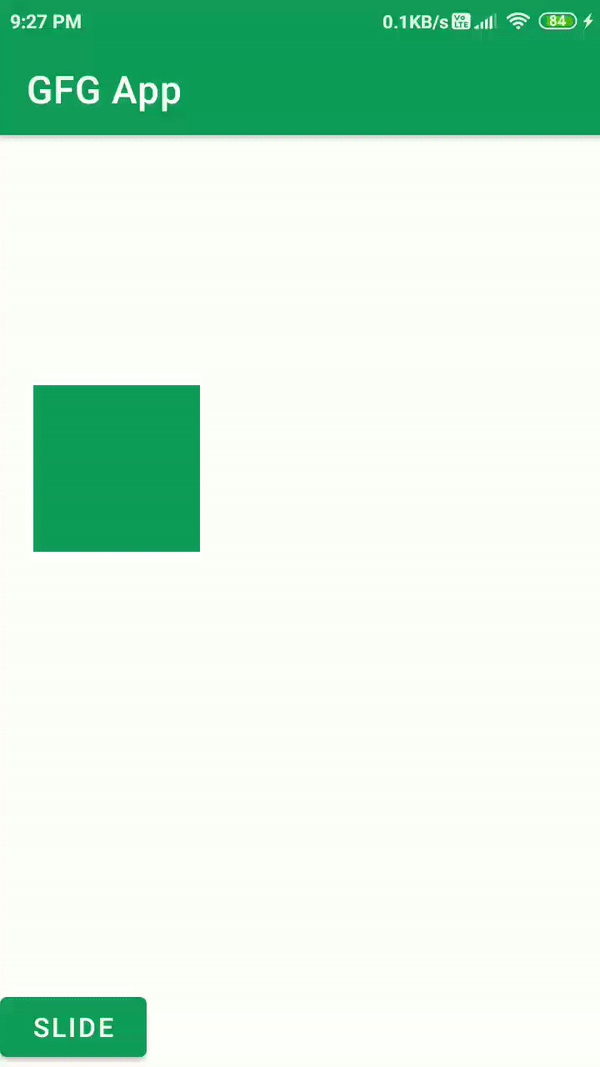
Last Updated : 06 Jan, 2025

Animation is the process of adding a motion effect to any view, image, or text. With the help of an animation, you can add motion or can change the shape of a specific view. Animation in Android is generally used to give your UI a rich look and feel. The animations are basically of three types as follows:

1. **Property Animation**
2. **View Animation**
3. **Drawable Animation**

**1. Property Animation**

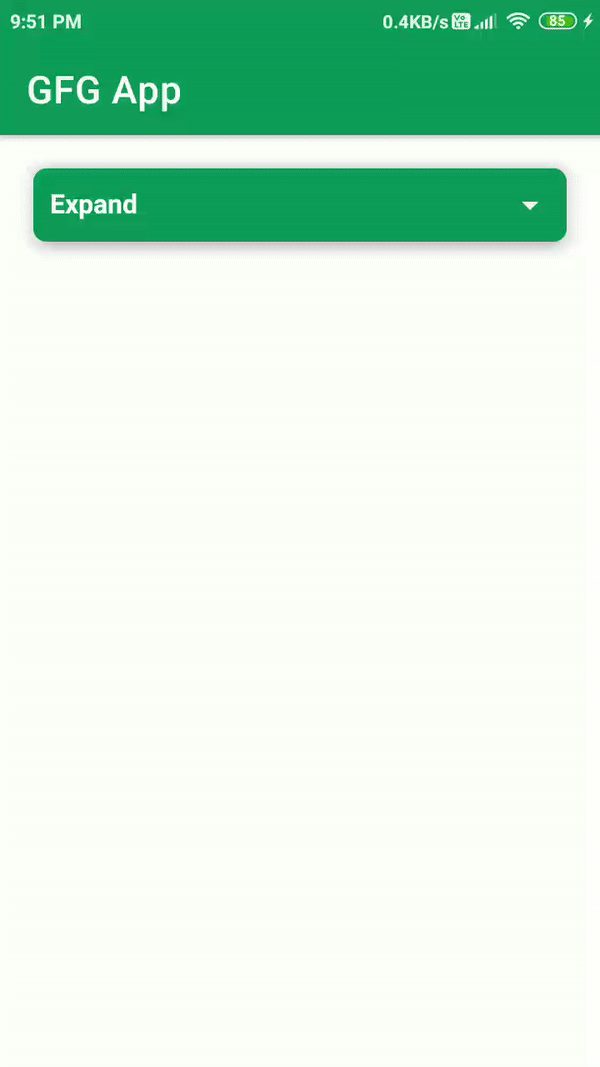
Property Animation is one of the robust frameworks which allows animation almost everything. This is one of the powerful and flexible animations which was introduced in Android 3.0. Property animation can be used to add any animation in the [CheckBox](https://www.geeksforgeeks.org/checkbox-in-kotlin), [RadioButtons](https://www.geeksforgeeks.org/android-how-to-add-radio-buttons-in-an-android-application), and widgets other than any view.



Looking to become an expert in Android App Development? Whether you're a student or a professional aiming to advance your career in mobile app development, our course, "[**Android App Development with Kotlin**](https://gfgcdn.com/tu/R4Q/)," available exclusively on GeeksforGeeks, is the perfect fit for you.

**2. View Animation**

View Animation can be used to add animation to a specific view to perform tweened animation on views. Tweened animation calculates animation information such as size, rotation, start point, and endpoint. These animations are slower and less flexible. An example of View animation can be used if we want to expand a specific layout in that place we can use View Animation. The example of View Animation can be seen in Expandable RecyclerView.



**3. Drawable Animation**

Drawable Animation is used if you want to animate one image over another. The simple way to understand is to animate drawable is to load the series of drawable one after another to create an animation. A simple example of drawable animation can be seen in many apps Splash screen on apps logo animation.



**Important Methods of Animation**

| **Methods** | **Description** |
| --- | --- |
| startAnimation() | This method will start the animation. |
| clearAnimation() | This method will clear the animation running on a specific view. |

**Example of Implementation Android Animation**

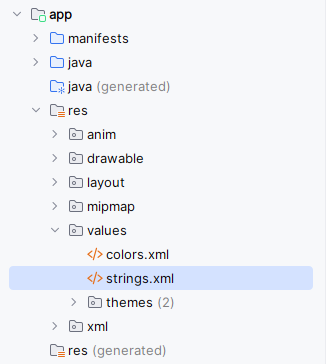
Now we will see the Simple Example to add animations to [ImageView](https://www.geeksforgeeks.org/imageview-in-kotlin). Note that we are going to implement this project using the **Java** language.

**Step 1: Create a New Project**

To create a new project in Android Studio please refer to [How to Create/Start a New Project in Android Studio](https://www.geeksforgeeks.org/android-how-to-create-start-a-new-project-in-android-studio). Note that select **Java** as the programming language.

**Step 2: Working with the strings.xml file**

Strings.xml can be found from the **app > res > values > strings.xml**.



**Below is the snippet for the strings.xml file.**

**<resources>**

**<string** name="app\_name"**>**GFG App**</string>**

**<string** name="blink"**>**BLINK**</string>**

**<string** name="clockwise"**>**ROTATE**</string>**

**<string** name="fade"**>**FADE**</string>**

**<string** name="move"**>**MOVE**</string>**

**<string** name="slide"**>**SLIDE**</string>**

**<string** name="zoom"**>**ZOOM**</string>**

**<string** name="stop\_animation"**>**STOP ANIMATION**</string>**

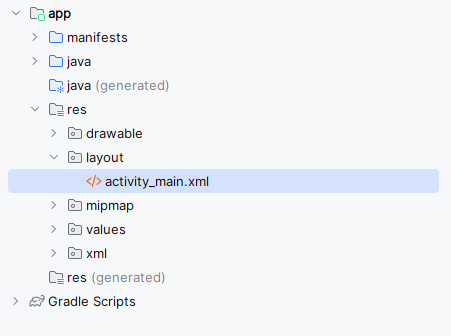
**<string** name="course\_rating"**>**Course Rating**</string>**

**<string** name="course\_name"**>**Course Name**</string>**

**</resources>**

**Step 3: Working with the activity\_main.xml file**

Create ImageView in the **activity\_main.xml** along with buttons that will add animation to the view. Navigate to the **app > res > layout > activity\_main.xml**.



Below is the code for the **activity\_main.xml** file.

<?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=".MainActivity"**>**

**<ImageView**

android:id="@+id/imageview"

android:layout\_width="200dp"

android:layout\_height="200dp"

android:layout\_centerHorizontal="true"

android:layout\_marginTop="40dp"

android:contentDescription="@string/app\_name"

android:src="@drawable/gfgimage" **/>**

**<LinearLayout**

android:id="@+id/linear1"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/imageview"

android:layout\_marginTop="30dp"

android:orientation="horizontal"

android:weightSum="3"**>**

*<!--To start the blink animation of the image-->*

**<Button**

android:id="@+id/BTNblink"

style="@style/TextAppearance.AppCompat.Widget.Button"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:padding="3dp"

android:text="@string/blink"

android:textColor="@color/white" **/>**

*<!--To start the rotate animation of the image-->*

**<Button**

android:id="@+id/BTNrotate"

style="@style/TextAppearance.AppCompat.Widget.Button"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:padding="3dp"

android:text="@string/clockwise"

android:textColor="@color/white" **/>**

*<!--To start the fading animation of the image-->*

**<Button**

android:id="@+id/BTNfade"

style="@style/TextAppearance.AppCompat.Widget.Button"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:padding="3dp"

android:text="@string/fade"

android:textColor="@color/white" **/>**

**</LinearLayout>**

**<LinearLayout**

android:id="@+id/linear2"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/linear1"

android:layout\_marginTop="30dp"

android:orientation="horizontal"

android:weightSum="3"**>**

*<!--To start the move animation of the image-->*

**<Button**

android:id="@+id/BTNmove"

style="@style/TextAppearance.AppCompat.Widget.Button"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:padding="3dp"

android:text="@string/move"

android:textColor="@color/white" **/>**

*<!--To start the slide animation of the image-->*

**<Button**

android:id="@+id/BTNslide"

style="@style/TextAppearance.AppCompat.Widget.Button"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:padding="3dp"

android:text="@string/slide"

android:textColor="@color/white" **/>**

*<!--To start the zoom animation of the image-->*

**<Button**

android:id="@+id/BTNzoom"

style="@style/TextAppearance.AppCompat.Widget.Button"

android:layout\_width="0dp"

android:layout\_height="wrap\_content"

android:layout\_margin="10dp"

android:layout\_weight="1"

android:padding="3dp"

android:text="@string/zoom"

android:textColor="@color/white" **/>**

**</LinearLayout>**

*<!--To stop the animation of the image-->*

**<Button**

android:id="@+id/BTNstop"

android:layout\_width="match\_parent"

android:layout\_height="wrap\_content"

android:layout\_below="@id/linear2"

android:layout\_marginLeft="30dp"

android:layout\_marginTop="30dp"

android:layout\_marginRight="30dp"

android:text="@string/stop\_animation" **/>**

**</RelativeLayout>**

**Step 4: Create 6 different types of animation for ImageView**

To create new animations we have to create a new directory for storing all our animations. Navigate to the**app > res > Right-Click on res >> New >> Directory >> Name your directory as “anim”**. Inside this directory, we will create our animations. For creating a new anim right click on the anim directory >> Animation Resource file and give the name to your file. Below is the code snippet for 6 different animations.

**1. Blink Animation**

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

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

**<alpha** android:fromAlpha="0.0"

android:toAlpha="1.0"

android:interpolator="@android:anim/accelerate\_interpolator"

android:duration="500"

android:repeatMode="reverse"

android:repeatCount="infinite"**/>**

**</set>**

**2. Fade Animation**

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

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

android:interpolator="@android:anim/accelerate\_interpolator"**>**

*<!-- duration is the time for which animation will work-->*

**<alpha**

android:duration="1000"

android:fromAlpha="0"

android:toAlpha="1" **/>**

**<alpha**

android:duration="1000"

android:fromAlpha="1"

android:startOffset="2000"

android:toAlpha="0" **/>**

**</set>**

**3. Move Animation**

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

**<set**

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

android:interpolator="@android:anim/linear\_interpolator"

android:fillAfter="true"**>**

**<translate**

android:fromXDelta="0%p"

android:toXDelta="75%p"

android:duration="700" **/>**

**</set>**

**4. Rotate Animation**

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

**<set**

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

**<rotate**

android:duration="6000"

android:fromDegrees="0"

android:pivotX="50%"

android:pivotY="50%"

android:toDegrees="360" **/>**

**<rotate**

android:duration="6000"

android:fromDegrees="360"

android:pivotX="50%"

android:pivotY="50%"

android:startOffset="5000"

android:toDegrees="0" **/>**

**</set>**

**5. Slide Animation**

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

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

android:fillAfter="true" **>**

**<scale**

android:duration="500"

android:fromXScale="1.0"

android:fromYScale="1.0"

android:interpolator="@android:anim/linear\_interpolator"

android:toXScale="1.0"

android:toYScale="0.0" **/>**

**</set>**

**6. Zoom Animation**

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

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

android:fillAfter="true" **>**

**<scale**

android:interpolator="@android:anim/linear\_interpolator"

android:duration = "1000"

android:fromXScal = "1"

android:fromYScale = "1"

android:pivotX = "50%"

android:pivotY = "50%"

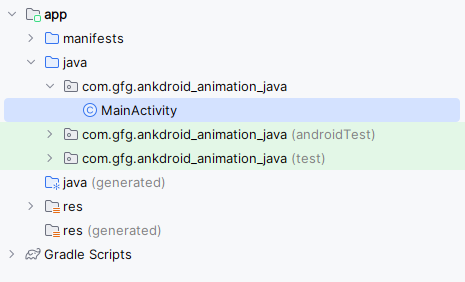
android:toXScale = "2"

android:toYScale = "2"**/>**

**</set>**

**Step 5: Working with the MainActivity.java file**

Add animation to the ImageView by clicking a specific Button. Navigate to the **app > java > your apps package name >> MainActivity.java**.



**Below is the implementation of MainActivity file:**

MainActivity.javaMainActivity.kt

**import** **androidx.appcompat.app.AppCompatActivity**;

**import** **android.os.Bundle**;

**import** **android.view.View**;

**import** **android.view.animation.Animation**;

**import** **android.view.animation.AnimationUtils**;

**import** **android.widget.Button**;

**import** **android.widget.ImageView**;

**public** **class** **MainActivity** **extends** AppCompatActivity {

*// Declare ImageView and Buttons*

ImageView imageView;

Button blinkBTN, rotateBTN, fadeBTN, moveBTN, slideBTN, zoomBTN, stopBTN;

@Override

**protected** void onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.activity\_main);

*// Initialize ImageView and Buttons using their IDs*

imageView = findViewById(R.id.imageview);

blinkBTN = findViewById(R.id.BTNblink);

rotateBTN = findViewById(R.id.BTNrotate);

fadeBTN = findViewById(R.id.BTNfade);

moveBTN = findViewById(R.id.BTNmove);

slideBTN = findViewById(R.id.BTNslide);

zoomBTN = findViewById(R.id.BTNzoom);

stopBTN = findViewById(R.id.BTNstop);

*// Set up click listener for blink button to start blink animation*

blinkBTN.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** void onClick(View v) {

*// Load blink animation and start it on the ImageView*

Animation animation = AnimationUtils.loadAnimation(getApplicationContext()

, R.anim.blink\_animation);

imageView.startAnimation(animation);

}

});

*// Set up click listener for rotate button to start rotate animation*

rotateBTN.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** void onClick(View v) {

*// Load rotate animation and start it on the ImageView*

Animation animation = AnimationUtils.loadAnimation(getApplicationContext()

, R.anim.rotate\_animation);

imageView.startAnimation(animation);

}

});

*// Set up click listener for fade button to start fade animation*

fadeBTN.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** void onClick(View v) {

*// Load fade animation and start it on the ImageView*

Animation animation = AnimationUtils.loadAnimation(getApplicationContext()

, R.anim.fade\_animation);

imageView.startAnimation(animation);

}

});

*// Set up click listener for move button to start move animation*

moveBTN.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** void onClick(View v) {

*// Load move animation and start it on the ImageView*

Animation animation = AnimationUtils.loadAnimation(getApplicationContext()

, R.anim.move\_animation);

imageView.startAnimation(animation);

}

});

*// Set up click listener for slide button to start slide animation*

slideBTN.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** void onClick(View v) {

*// Load slide animation and start it on the ImageView*

Animation animation = AnimationUtils.loadAnimation(getApplicationContext()

, R.anim.slide\_animation);

imageView.startAnimation(animation);

}

});

*// Set up click listener for zoom button to start zoom animation*

zoomBTN.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** void onClick(View v) {

*// Load zoom animation and start it on the ImageView*

Animation animation = AnimationUtils.loadAnimation(getApplicationContext()

, R.anim.zoom\_animation);

imageView.startAnimation(animation);

}

});

*// Set up click listener for stop button to clear any ongoing animation*

stopBTN.setOnClickListener(**new** View.OnClickListener() {

@Override

**public** void onClick(View v) {

*// Clear the animation on the ImageView*

imageView.clearAnimation();

}

});

}

}

***Note:*** *Drawables and strings can be found in the drawable folder and strings.xml file. Drawables can be found from the* ***app > res > drawable****.*