

פרק 14 - אנימציות

לאפליקציות שניצור נוכל להוסיף אנימציות.
נוכל לשנות את הדרך שבה האובייקט שלנו יופיע.
נלמד להשתמש במספר אנימציות בסיסיות:

fade in

fade out

CrossFade

Blink

Zoom in

Zoom Out

Rotate

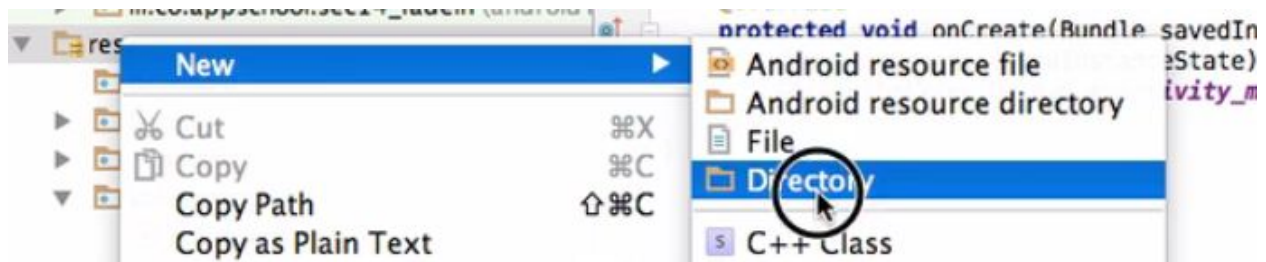
Move

Slide up

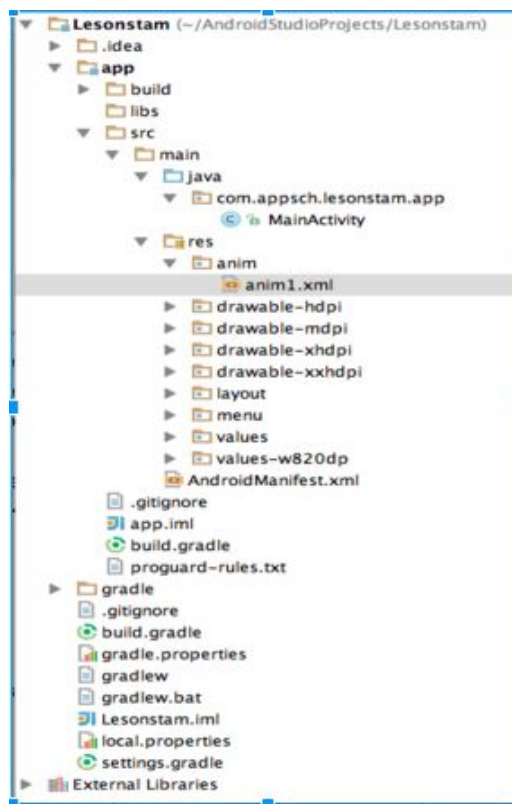
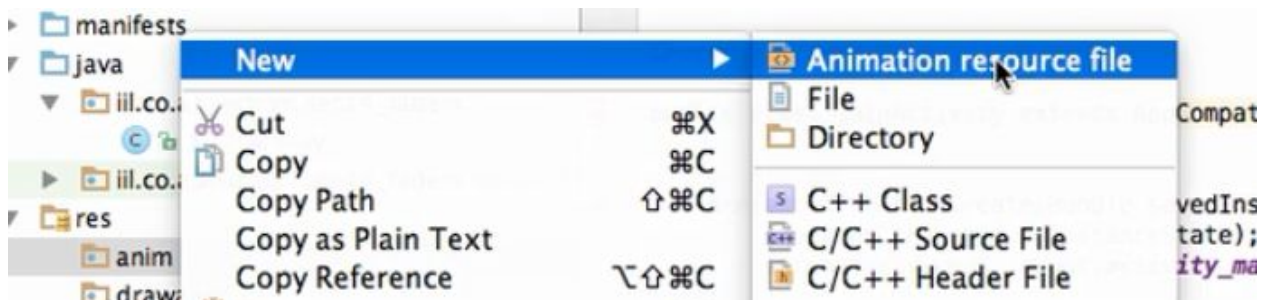
offset

הדרך להוספת אנימציה לאפליקציה:

שלב 1 - ניצור תיקיה בשם anim תחת התיקיה res:



שלב 2 - נוסיף קובץ animation לתיקיה anim, שבו נגדיר את האנימציה:



אנימציות fade in:

אנימציות fade in מבהירה את התמונה בהדרגה

1. נערוך את קובץ האנימציה שבתיקיה anim

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

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

```
<alpha
```

```
  android:duration="5000"
```

```
  android:fromAlpha="0.0"
```

```
  android:toAlpha="1.0"
```

```
>
```

```
</alpha>
```

```
</set>
```

כמה זמן תרצו שהאנימציה
תפעל (milliseconds)

רמת הבהירות של
התמונה בהתחלה

רמת הבהירות של
התמונה בסוף
(משתנה בהדרגה)

2. עיצוב MainActivity

```
<?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="iil.co.appschool.sec14_fadein.MainActivity">
```

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Fade in!" />

<Button

android:layout_width="200dp"

android:layout_height="wrap_content"

android:id="@+id/btnFadein"

android:text="Fade in"

/>

</LinearLayout>

3. טעינת האנימציה על האובייקט שנרצה

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
```

```
    Button btnFadeIn;
```

```
    Animation animFadeIn;
```

```
@Override
```

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

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    btnFadeIn=(Button)findViewById(R.id.btnFadeIn);
```

```
    btnFadeIn.setOnClickListener(this);
```

```
    animFadeIn= AnimationUtils.loadAnimation(this,R.anim.fade_in);
```

```
}
```

ניתן הפניה גם לאנימציה
לקובץ ה animation
שיצרנו בתיקיה anim

3. הפעלת האנימציה

```
@Override
```

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

```
    if(v==btnFadeIn)
```

```
    {
```

```
        btnFadeIn.startAnimation(animFadeIn);
```

```
    }
```

```
}
```

```
}
```

תוכלו להפעיל את האנימציה
על כל אובייקט שתמצאו

תכונות נוספות שניתן להוסיף לאנימציה:

android:duration - duration of animation

android:startOffset - waiting time before an animation start.

Android:interpolator - it is the rate of change in animation

android:fillAfter:

android:repeatMode - when you want the animation to repeat.

android:repeatCount - number of repeats

אנימצית FadeOut:

אנימציה שמתעמעמת עם הזמן.

1. נערוך את קובץ האנימציה שבתיקיה anim

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

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

<pre><alpha</pre>	
<pre> android:fromAlpha="1"</pre>	בהירות התמונה ההתחלתית
<pre> android:toAlpha="0"</pre>	בהירות התמונה בסוף (מתעמעמת בהדרגה)
<pre> android:duration="1000"</pre>	הזמן שתפעל האנימציה (milliseconds)
<pre> android:repeatCount="3"</pre>	מספר הפעמים שתפעל האנימציה
<pre>></pre>	
<pre></alpha></pre>	
<pre></set></pre>	

2. עיצוב MainActivity

```
<?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="il.co.appschool.sec14_fadeout.MainActivity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Fade out!" />
```

```
<Button
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:id="@+id/btnFadeOut"
```

```
    android:text="Fade out"
```

```
/>
```

```
</LinearLayout>
```

3. טעינת האנימציה על האובייקט שנרצה


```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
```

```
    Button btnFadeOut;
```

```
    Animation animFadeOut;
```

```
@Override
```

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

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    btnFadeOut=(Button)findViewById(R.id.btnFadeOut);
```

```
    btnFadeOut.setOnClickListener(this);
```

```
    animFadeOut= AnimationUtils.loadAnimation(this,R.anim.fade_out);
```

```
    animFadeOut.setAnimationListener(new FadeOut());
```

```
}
```

נוכל לממש class חדש ובו
נממש ממשק ופונקציות
לשליטה באנימציה (לא חובה)

שלב 4 - הפעלת האנימציה

```
@Override
```

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

```
    if(v==btnFadeOut)
```

```
    {
```

```
        btnFadeOut.startAnimation(animFadeOut);
```

```
    }
```

```
}
```

שלב 5 - בניית class ומימוש הממשק לשליטה באנימציה

```
public class FadeOut implements Animation.AnimationListener {
```

```
    @Override
```

```
    public void onAnimationStart(Animation animation) {
```

```
        if(animation==animFadeOut) {
```

```
            Toast.makeText(MainActivity.this, "anim start", Toast.LENGTH_LONG).show();
```

```
        }
```

```
    }
```

שליטה באנימציה
ברגע ההפעלה

```
    @Override
```

```
    public void onAnimationEnd(Animation animation) {
```

```
        if(animation==animFadeOut) {
```

```
            Toast.makeText(MainActivity.this, "anim end", Toast.LENGTH_LONG).show();
```

```
        }
```

```
    }
```

שליטה באנימציה
ברגע הסיום

```
    @Override
```

```
    public void onAnimationRepeat(Animation animation) {
```

```
    }
```

```
}
```

שליטה באנימציה כשהיא
מופעלת מספר פעמים

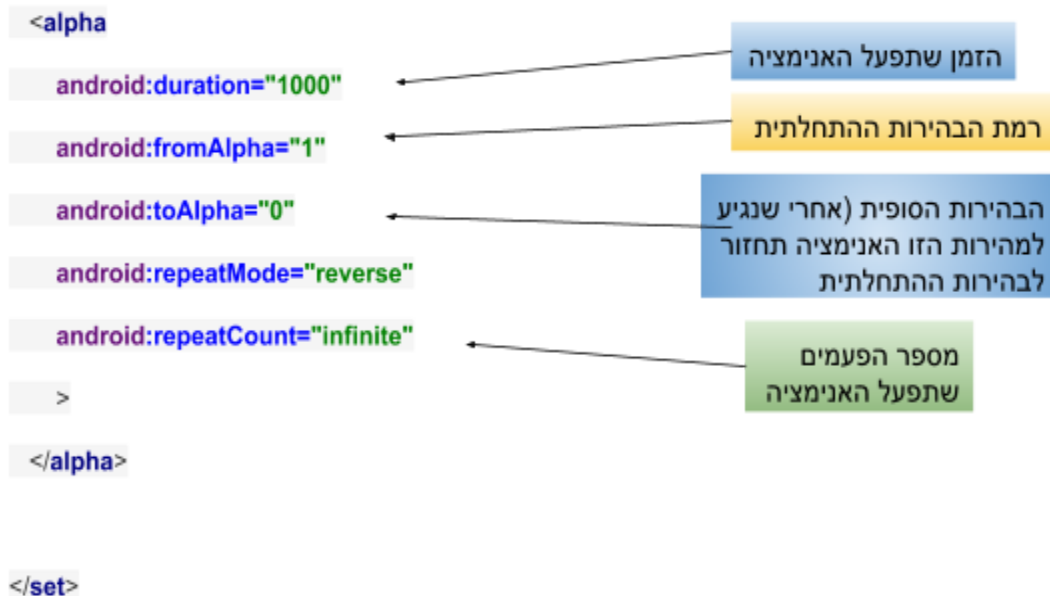
אנימצית reverse:

אנימציה שבסוף חוזרת למצבה ההתחלתי.

1. נערוך את קובץ האנימציה שבתיקיה anim

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

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



2. עיצוב MainActivity

```
<?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="iil.co.appschool.sec14_fadein.MainActivity">
```

```
<TextView  
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:text="Reverse !" />
```

```
<Button  
    android:layout_width="200dp"  
    android:layout_height="wrap_content"  
    android:text="Click me"  
    android:id="@+id/btnReverse"  
/>
```

```
</LinearLayout>
```

3. טעינת האפליקציה על האובייקט שנרצה

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
    Button btnReverse;  
  
    Animation animReverse;  
  
    @Override  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_main);  
  
  
        btnReverse=(Button)findViewById(R.id.btnReverse);  
  
        btnReverse.setOnClickListener(this);  
  
        animReverse= AnimationUtils.loadAnimation(this, R.anim.reverse);  
  
        animReverse.setAnimationListener(new FadeAnim());  
  
    }  
}
```

4. הפעלת האנימציה

```
@Override  
  
public void onClick(View v) {  
  
    if(v==btnReverse)  
  
    {  
  
        btnReverse.startAnimation(animReverse);  
  
    }  
  
}
```

5. בניית המחלקה ומימוש הממשק לשליטה באנימציה

```
public class FadeAnim extends AppCompatActivity implements Animation.AnimationListener {  
  
    @Override  
    public void onAnimationStart(Animation animation) {  
        Toast.makeText(MainActivity.this, "Animation start",  
            Toast.LENGTH_SHORT).show();  
    }  
  
    @Override  
    public void onAnimationEnd(Animation animation) {  
        if (animation == animReverse) {  
            Toast.makeText(MainActivity.this, "Animation Stopped", Toast.LENGTH_SHORT).show();  
        }  
    }  
  
    @Override  
    public void onAnimationRepeat(Animation animation) {  
    }  
}
```

אנימציות move:

אנימציה שזזה על פני המסך מצד לצד.

1. נערוך את קובץ האנימציה שבתיקיה anim:

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

```
<set xmlns:android="http://schemas.android.com/apk/res/android" android:fillAfter="true"
    android:fillEnabled="true">
```

```
<translate
```

```
    android:duration="3000"
```

```
    android:fromXDelta="1"
```

```
    android:toXDelta="200"
```

```
></translate>
```

```
</set>
```

נוכל לקבוע גם את
ערך ה y של הנקודה

נקודת ההתחלה של
האובייקט על המסך

הנקודה הסופית של
האובייקט על המסך

2. עיצוב MainActivity

```
<?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="iil.co.appschool.sec14_fadein.MainActivity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
android:text="Move !" />
```

```
<Button
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Click me"
```

```
    android:id="@+id/btnMove"
```

```
/>
```

```
</LinearLayout>
```

3. טעינת האנימציה על האובייקט שנרצה


```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
    Button btnMove;  
  
    Animation animMove;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_main);  
  
  
        btnMove=(Button)findViewById(R.id.btnMove);  
        btnMove.setOnClickListener(this);  
        animMove= AnimationUtils.loadAnimation(this,R.anim.move);  
    }  
}
```

4. נפעיל את האנימציה

```
@Override  
public void onClick(View v) {  
    if(v==btnMove) {  
        btnMove.startAnimation(animMove);  
    }  
}  
}
```

אנימציות rotate

1. נערוך את קובץ האנימציה בתיקיה anim

```

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

<set xmlns:android="http://schemas.android.com/apk/res/android" android:fillAfter="true" android:fillEnabled="true">

  <rotate
    android:fromDegrees="0"
    android:toDegrees="100"
    android:pivotX="50%"
    android:pivotY="50%"
    android:duration="3000"
  >
</rotate>
</set>

```

מרכז הכובד יהיה באמצע (יסתובב סביב האמצע)

כשנצא מהמסך האנימציה תישאר, ולא תחזור חזרה

2. עיצוב MainActivity

```

<?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="iil.co.appschool.sec14_fadein.MainActivity">

  <TextView
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:text="Move !" />

```

```
<Button
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Click me"
```

```
    android:id="@+id/btnRotate"
```

```
<ImageView
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="200dp"
```

```
    android:id="@+id/iv"
```

```
    android:background="@drawable/ic_launcher"
```

```
</LinearLayout>
```

3. טעינת האנימציה על האובייקט שנרצה

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
    Button btnRotate;  
    ImageView iv;  
    Animation animRotate;  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
  
        btnRotate=(Button)findViewById(R.id.btnRotate);  
        btnRotate.setOnClickListener(this);  
        iv=(ImageView)findViewById(R.id.iv);  
  
        animRotate= AnimationUtils.loadAnimation(this,R.anim.rotate);  
    }  
}
```

4. נפעיל את האנימציה

```
@Override  
public void onClick(View v) {  
  
    if(v==btnRotate)  
    {  
        iv.startAnimation(animRotate);  
    }  
}
```

אנימציות scale

האנימציה יכולה להקטין את האובייקט או להגדיל אותו

1. נערוך את קובץ האנימציה שבתיקה anim

```
<?xml version="1.0" encoding="utf-8"?>  
  
<set xmlns:android="http://schemas.android.com/apk/res/android" android:fillAfter="true" android:fillEnabled="true">  
  
  <scale  
  
    android:duration="3000"  
  
    android:fromXScale="1"  
  
    android:toXScale="0.1"  
  
    android:fromYScale="1"  
  
    android:toYScale="0.1"  
  
    android:pivotY="50%"  
  
    android:pivotX="50%"  
  
  >  
  
</scale>  
  
</set>
```

2. עיצוב MainActivity

```
<?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="iil.co.appschool.sec14_fadein.MainActivity">
```

<TextView

android:layout_width="wrap_content"

android:layout_height="wrap_content"

android:text="Scale !" />

<Button

android:layout_width="200dp"

android:layout_height="wrap_content"

android:text="Click me"

android:id="@+id/btnScale"

/>

<ImageView

android:layout_width="200dp"

android:layout_height="200dp"

android:id="@+id/iv"

android:background="@drawable/ic_launcher"

/>

</LinearLayout>

3. טעינת האנימציה על האובייקט שנרצה

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
    Animation scaleAnim;  
  
    Button btnScale;  
  
    ImageView iv;  
  
    @Override  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_main);  
  
  
        btnScale=(Button)findViewById(R.id.btnScale);  
  
        btnScale.setOnClickListener(this);  
  
        iv=(ImageView)findViewById(R.id.iv);  
  
        scaleAnim= AnimationUtils.loadAnimation(this,R.anim.scale);  
  
    }  
}
```

4. הפעלת האנימציה

```
@Override  
  
public void onClick(View v) {  
  
    if(v==btnScale){  
  
        iv.startAnimation(scaleAnim);  
  
    }  
  
}  
  
}
```

אנימציות slide up

סיבוב של האובייקט במקום

1. נערוך את קובץ האנימציה שבתיקיה anim

```
<?xml version="1.0" encoding="utf-8"?>  
  
<set xmlns:android="http://schemas.android.com/apk/res/android">  
  
  <scale  
  
    android:duration="3000"  
  
    android:fromXScale="1"  
  
    android:toXScale="0"  
  
    android:fromYScale="1"  
  
    android:toYScale="1"  
  
    android:pivotX="50%"  
  
    android:pivotY="50%"  
  
  >  
  
</scale>  
  
</set>
```

2. עיצוב MainActivity

```
<?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="iil.co.appschool.sec14_fadein.MainActivity">
```



```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Slide out !" />
```

```
<Button
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Click me"
```

```
    android:id="@+id/btnSlideUp"
```

```
/>
```

```
<ImageView
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="200dp"
```

```
    android:id="@+id/iv"
```

```
    android:background="@drawable/ic_launcher"
```

```
/>
```

```
</LinearLayout>
```

3. נטען את האנימציה על האובייקט שנרצה

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {  
  
    Button btnSlideUp;  
  
    ImageView iv;  
  
    Animation animSlideUp;  
  
    @Override  
  
    protected void onCreate(Bundle savedInstanceState) {  
  
        super.onCreate(savedInstanceState);  
  
        setContentView(R.layout.activity_main);  
  
  
  
        btnSlideUp=(Button)findViewById(R.id.btnSlideUp);  
  
        btnSlideUp.setOnClickListener(this);  
  
        iv=(ImageView)findViewById(R.id.iv);  
  
        animSlideUp= AnimationUtils.loadAnimation(this,R.anim.slide_up);  
  
    }  
}
```

4. נפעיל את האנימציה

```
@Override  
  
public void onClick(View v) {  
  
    if(v==btnSlideUp) {  
  
        iv.startAnimation(animSlideUp);  
  
    }  
  
}  
  
}
```

אנימציות OffSet

הפעלת אנימציות יפעלו אחת אחרי השנייה

1. נערוך את קובץ האנימציה שבתיקה anim

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

```
<set xmlns:android="http://schemas.android.com/apk/res/android" android:fillAfter="true" android:fillEnabled="true">
```

```
<rotate
```

```
    android:duration="3000"
```

```
    android:fromDegrees="0"
```

```
    android:toDegrees="360"
```

```
    android:pivotY="50%"
```

```
    android:pivotX="50%"
```

```
></rotate>
```

```
<translate
```

```
    android:duration="2000"
```

```
    android:fromXDelta="1"
```

```
    android:toXDelta="800"
```

```
    android:startOffset="3000"
```

```
>
```

```
</translate>
```

```
</set>
```

2. עיצוב MainActivity

```
<?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="iil.co.appschool.sec14_fadein.MainActivity">
```

```
<TextView
```

```
    android:layout_width="wrap_content"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Offset !" />
```

```
<Button
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="wrap_content"
```

```
    android:text="Click me"
```

```
    android:id="@+id/btnOffset"
```

```
/>
```

```
<ImageView
```

```
    android:layout_width="200dp"
```

```
    android:layout_height="200dp"
```

```
    android:id="@+id/iv"
```

```
android:background="@drawable/ic_launcher"
```

```
/>
```

```
</LinearLayout>
```

3. טעינת האנימציה על האובייקט שנרצה

```
public class MainActivity extends AppCompatActivity implements View.OnClickListener {
```

```
    Animation animOffset;
```

```
    Button btnOffset;
```

```
    ImageView iv;
```

```
@Override
```

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

```
    super.onCreate(savedInstanceState);
```

```
    setContentView(R.layout.activity_main);
```

```
    btnOffset=(Button)findViewById(R.id.btnOffset);
```

```
    btnOffset.setOnClickListener(this);
```

```
    iv=(ImageView)findViewById(R.id.iv);
```

```
    animOffset= AnimationUtils.loadAnimation(this,R.anim.offset_anim);
```

```
}
```

4. הפעלת האנימציה

```
@Override  
  
public void onClick(View v) {  
  
    if(v==btnOffset){  
  
        iv.startAnimation(animOffset);  
  
    }  
  
}  
  
}
```

תוכלו לשלב גם שני סוגי אנימציות בקובץ אנימציה אחד כך שיעבדו
במקביל:

```
<?xml version="1.0" encoding="utf-8"?>  
<set xmlns:android="http://schemas.android.  
  
    <translate  
        android:duration="3000"  
        android:fromXDelta="1"  
        android:toXDelta="200"  
        android:repeatMode="reverse"  
  
    ></translate>  
  
    <alpha  
        android:duration="5000"  
        android:fromAlpha="0"  
        android:toAlpha="1"  
        android:repeatMode="reverse"  
        android:repeatCount="5"  
  
    ></alpha>  
  
</set>
```