

```
<!-------#RainbowTextView.java-----!>
```

```
package com.example.application;

import android.content.Context;
import android.graphics.LinearGradient;
import android.graphics.Shader;
import android.util.AttributeSet;
import android.widget.TextView;
import android.animation.ValueAnimator;
import android.graphics.LinearGradient;
import android.graphics.Shader;
import android.os.Bundle;
import android.widget.TextView;
import android.view.animation.LinearInterpolator;

public class RainbowTextView extends TextView {

    private int[] colors = {
        0xFFFF0000, // Red
        0xFFFF7F00, // Orange
        0xFFFFFF00, // Yellow
        0xFF00FF00, // Green
        0xFF0000FF, // Blue
        0xFF4B0082, // Indigo
        0xFF9400D3  // Violet
    };

    public RainbowTextView(Context context) {
        super(context);
        init();
    }

    public RainbowTextView(Context context, AttributeSet attrs) {
        super(context, attrs);
        init();
    }

    public RainbowTextView(Context context, AttributeSet attrs, int defStyleAttr) {
        super(context, attrs, defStyleAttr);
        init();
    }

    private void init() {
        // Set up the gradient shader
        Shader shader = new LinearGradient(0, 0, getWidth(), 0, colors, null, Shader.TileMode.CLAMP);
        getPaint().setShader(shader);

        // Start the animation
        startAnimation();
    }

    private void startAnimation() {
        // Create a ValueAnimator to animate the gradient
        ValueAnimator animator = ValueAnimator.ofFloat(0.0f, 1.0f);
        animator.setDuration(7000); // Duration of the animation
        //animator.setRepeatMode(ValueAnimator.REVERSE);
        animator.setRepeatCount(ValueAnimator.INFINITE); // Repeat indefinitely
        animator.setInterpolator(new LinearInterpolator());
        animator.addUpdateListener(new ValueAnimator.AnimatorUpdateListener() {
            @Override
            public void onAnimationUpdate(ValueAnimator animation) {
                float value = (float) animation.getAnimatedValue();
                int startX = (int) (value * getWidth());
                int endX = (int) (startX + getWidth());

                Shader shader = new LinearGradient(
                    startX, 0, endX, 0,
                    colors, null, Shader.TileMode.REPEAT);
                getPaint().setShader(shader);
                invalidate(); // Redraw the TextView
            }
        });
        animator.start();
    }
}
```

Â

```
<!-------#activity_main.xml-----!>
```

Use RainbowTextView In activity\_main.xml

Â

```
<?xml version="1.0" encoding="utf-8"?>
<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="match_parent"
    android:layout_height="match_parent">

    <com.example.application.RainbowTextView
        android:id="@+id/rainbow_view"
        android:layout_width="match_parent"
        android:layout_height="match_parent"/>

</RelativeLayout>
```

```
<!-------#MainActivity-----!>
```

Use RainbowTextView In MainActivity

```
package com.example.RainbowTextView;
```

```
import android.os.Bundle;
import androidx.appcompat.app.AppCompatActivity;

public class MainActivity extends AppCompatActivity {
    private RainbowTextView RainbowTextView;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
        RainbowTextView = findViewById(R.id.rainbow_view); // From activity_main.xml Id

        // Without Any Xml Id By Only Java Source
        RainbowTextView rainP = new RainbowTextView(this);
        layout_name.addView(rainP); // Replace With Actual Layout Id | Layout Names($$layout_name$$)-----!

    }
}
}
}
```