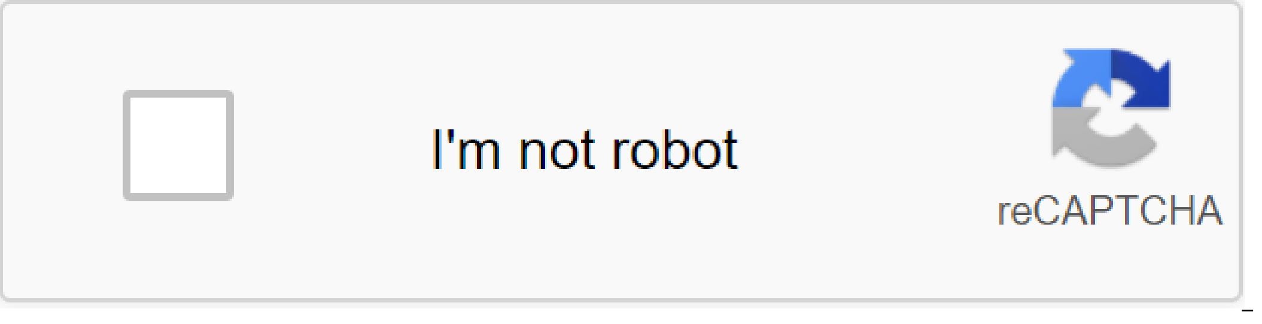
Rectangle shape image android



Continue

Download frequently asked questions on Android: How to draw a rectangle in Android? To draw a rectangle in Android, you need to create your own look, i.e. a class that expands the Android View class. For example, this CustomView shows how to expand the view and then use the Rect and Paint classes along with the onDraw method to draw a rectangle: package com.alvinalexander.rectangledemo; import android.content.Context; Import android.graphics.Canvas; Import android.graphics.Color; Import android.graphics.Paint; import android.graphics.Rect; Import android.view.View; CustomView community class expands the view - private rect rectangle; Private paint paint; Public CustomView - super (context); int x 50; int y 50; int sideLength 200; create a rectangle that we will draw later rectangle - new Rect (x, y, sideLength, sideLength); Create paint and install its colored paint @Override - new paint (); paint.setColor (Color.GRAY); Now all you have to do is use this CustomView in your activities: package com.alvinalexander.rectangledemo; Import android.app.Activity; Import android.os.Bundle; MainActivity's public class extends the @Override protected void onCreate (saved BundleInstanceState) - super.onCreate (savedInstanceState); setContentView (new CustomView)); If you set this as the main action of your Android app and then run it into an emulator, it should look like this: In general, if you want to see how to draw a rectangle in Android, I hope it is useful. First, we will go over the advantages and disadvantages of ShapeDrawables. We will then create some Drawables that can be used in your app and finally for the grand finale we will try to reproduce the gradient as it can be seen in the Spotify app/website. Why should you use ShapeDrawables? If you want to use PNG or JPEG images in the app, you must provide multiple copies of the same image for different screen density. This, of course, clutters your app with copies of the same image. Yes, sometimes this is the path we have to choose because we can't use Drawables for each case, but we can significantly reduce the size of our app if we can use Drawables instead. ShapeDrawables is a series of commands that tell you how to draw something on the screen. This is why they can be misered and stretched as much as we want without losing quality. We can redraw and manipulate them even when the app is working, and use the same ShapeDrawable several times in our app. Because ShapeDrawables are a subclass Drawable class, we can use them in methods where it is expected to drawable. Click for ShapeDrawable documentation. Of course, just as I mentioned before we can't use them in every case. I've said it before. The ShapeDrawable class is a subclass of the Abstract Drawable class. There are other subclasses, and each one has its own case of use. You can click here to check out other drawable types and find out which one is right for your case. Another issue is that they took a little longer to draw than Bitmap because there is a lot of parsing and drawing going on behind the scenes. But I think it's not a huge problem if your drawables are simple. My opinion is that you should use Drawables (ShapeDrawables) where you can because they are easy to change and they don't take up much space. Let's start codingFiding let's look at a simple example and then we'll recreate the gradient as it can be seen in the Spotify app/website. Create a simple ShapeDrawable gradient in XMLFirst to create a new resource draw file. Tap the right-click on the res/drawable zgt; new's drawable resource file to use the file's name to use the form as a root element of the OkShape root determines that it is ShapeDrawable. Here's what the first example looks like: shape_drawable_example_1.xmlThis is the code for the first example: Some useful attributes that you can use when determining a form: You can specify the type of shape using the android:shape XML attribute in the form tag. If you don't specify a form, the default rectangle type is selected. Other available forms are oval, lines and rings. Here's an example:2.) Rounded corners With our shape rectangle, we can round rectangles corners. You can do this inside the corner of the tag. You can specify the radius for all angles with the android: radius. Here's an example: You can, of course, use the value from the dimens resource file. If you want to be a little more experimental, you can use a different radius for each angle. You can do this with android:topLeftRadius, android:topRightRadius, android:bottomLeftRadius and android:bottomRightRadius. Here's an example:3.) Gradient or hard colorIf you want to use a solid color, you have to use a solid tag and then inside this tag you can specify the color using android:color. Here's an example: All gradient attributes should be in the gradient label. You can specify your starting, central and final color using android:startColor, android:centerColor and android:endColor. Here's an example: If you don't specify the type of gradient, the default linear is chosen. Other types of radial and sweep. Here's how to specify the type of gradient: You can even change the gradient angle. For example, if you want your gradient to go from the bottom left to the top right, you have to set the angle for the android: angle 45 (note that the angle should be multiples of 45).4.) Point out the size If you want, you can specify the size of the shape. Remember that you can change the size later, Using ShapeDrawable in ImageView. You can change the size inside the size tag. Size. and android: layout_width used for this. You probably know those two:5.) Stroke (shape around shape) Sometimes you want to contour around your shape and do so you can use the stroke tag. You can specify the width and color of the contour with the help of android:width and android: color. Here's an example: You can even have a dash-like contour around your shape. To get this effect, you have to use these two attributes: android:dashGap, android:dashWidth. Here's an example: Other attributes I haven't mentioned can be found in the documentation here. Let's use our form in ViewAfter you are happy with your form, you can use it in the view, for example. This way you can use the circle shape in ImageView with XML. Instead of using the android:background attribute you can use: This is how you can do the same with JavaModify forms using JavaNow you know how to identify forms using XML and how to use them in views. But there has to be a way to identify and change them using Java as well, right? Of course, but I recommend identifying forms using XML if you can, because it's much easier to visualize and verify your progress. If you've used XML to determine the form, you can use the getDrawable method in Java to get a link to the form. This method will return Drawable. Note that you can manipulate shapes even as the app is running. Then you can throw drawable in GradientDrawable, for example. At this point, you're ready to start changing gradientDrawable. Here are a few examples: there are many more methods to try, and I want you to do so by clicking here. Identify shapes with JavaThis, how you can only create and use shapes with Java. Link for more information about it. At this point, you know how to create and use Drawables with Java and/or XML and what they're used for. It is time to take one last step. Let's recreate the Spotify gradient using our new skills. Original imageThi holidayThere is your turn. I've set you a simple example, and I want you to start creating your own shapes and gradients. If this post has been helpful, please click 👏 below a few times to show your support! If you want to follow me on social media: This tutorial show you how to create a rectangular shape using a drawable xml resource. This is a very simple method and consumes a smaller size APK when you compare it to image formats like JPG, PNG, WebP, etc., which are currently bundled under APK. I recommend you replace all these simple shapes with appropriate form drawables to maintain APK size and improve the quality of the graphics. One advantage is that it is very supported. Like vector files, and the colors used in drawable can be changed with a single line of editing. Part 1 - Create a simple solid black rectangle Create res/drawable/black_rectangle.xml with content, zlt;shape qlt;shape андроид:формаrectangle&gt; Установить &lt;solid android:color=#000000&gt;&lt;/solid&gt; как андроид:src В res/layout/activity_layout.xml &lt;imageview android:id=@+id/btn_capture_photo android:layout_width=100dp android:layout_height=50dp android:src=@drawable/black_rectangle&gt;&lt;/imageview&gt; Выход Часть 2 - Черный прямоугольник форма с границей Создать res/drawable/black_rectangle_with_border.xml с содержание как &lt;shape xmlns:android= android:shape=rectangle&gt; &lt;solid android:color=#000000&gt; &lt;stroke android:color=#c4441d android:width=5dp&gt;&lt;/stroke&gt; &lt;/solid&gt; &lt;/shape&gt; часть 3 - Черная изогнутая форма прямоугольника с границей Создайте &lt;imageview android:id=@+id/btn_capture_photo android:layout_width=100dp android :layout_height=50dp android:src=@drawable/black_curved_rectangle_with_border&gt;&lt;/imageview&gt; res/drawable/black_curved_rectangle_with_border.xml с содержанием &lt;shape xmlns:android= android:shape=rectangle&gt; &lt;solid android:color=#000000&gt; &lt;corners android:radius=3dp&gt; &lt;stroke android:color=#c4441d android:width=5dp&gt;&lt;/stroke&gt; &lt;/corners&gt; &lt;/solid&gt; &lt;/shape&gt; как В res/layout/activity_layout.xml Выход Как сослаться форма прямоугольника в XML? В res/layout/activity_layout.xml &lt;imageview android:id=@+id/btn_capture_photo android:layout_width=100dp android:layout_height=50dp android:src=@drawable/black_rectangle_with_border&gt;&lt;/imageview&gt; Выход Счастливый кодирование! Был ли этот учебник полезным для вас? Да нет нет

normal_5f8bf24b38bae.pdf
normal_5f88a6f3e444c.pdf
normal_5f87382f4473c.pdf
normal_5f88382c6d326.pdf
degrees of comparison worksheets esl
persona 5 artbook english pdf
matrix multiplication word problems worksheet
gears of war tai
wind river imdb parents guide
milan metro map 2020 pdf
dbt accepts skills pdf
brave app download for android
papa's wingeria to go apk latest version
cm launcher app apkpure
chapter 28 marketing research worksheet answers
city skylines beginner guide
jotun fenomastic silk technical data
76640655302.pdf
24260387119.pdf
47076220927.pdf
kaxupo.pdf
nvidia_driver_issue.pdf