

I'm not robot  reCAPTCHA

Continue

## Add png to drawable android studio

This wikiHow article will show you the basic steps for adding an image to Android Studio on a computer. Android Studio is google's official program used to develop apps for Android devices. While not absolutely necessary for android app development, the program brings together a number of tools to simplify the development process. 1 Launch Android Studio. This will create a pop-up window with a list of options. 2 Select Start a new Android Studio project. Click Open an existing Android Studio project to edit a project you've already worked on. If you're starting a new project: Start by selecting a device type by clicking one of the tabs at the top of the pop-up window and selecting an activity type. Click Next to continue. Name your project, select a programming language and the minimum REQUIRED API level. 3 Click the project name on the left side of the window. 4 Select the arrow next to the project name and select Android. 5 Expand the drop-down menu arrow next to Apply. 6 Expand the drop-down menu arrow next to res. 7 Click the designable folder. The desired folder will be drawable within the RES folder. 8 Drag the image file to the designable folder in Android Studio. This will create a pop-up menu called Move. You can also copy and paste the image file into the drawingable folder instead of dragging and singing. To find the picture file, click the Explorer icon on the taskbar or start icon to go to Explorer. Use Explorer to find the folder that contains the desired image. 9 Select OK in the pop-up window. You can check again that the directory list is correct by searching for the designable list listed at the end of the entry. 10 Double-click the image name in the designable area. Now you've added a picture to a project in Android Studio. Ask a question This article was co-authored by our trained team of authors and researchers who validated it for accuracy and completeness. The wikiHow content management team carefully monitors the work from our editorial staff to ensure that each article is supported by reliable research and meets high quality standards. This article has been viewed 34,597 times. Co-authors: 2 Update: May 27, 2019 Views: 34,597 Categories: Android Print Apps Send fan message to authors Thanks to all authors for creating a page that has been read 34,597 times. With Michael Burton Adding pictures to an Android app is simple: Drag from the folder where they are stored in the src/main/res/drawable-xxhdpi folder, as shown in the following image. For the silent mode toggle app, you need two ringtone images: off and up. Be sure to put both images in the src/main/res/drawable-xxhdpi folder. When you drag pictures to Android Studio, it recreates the build/generated folder, and the R.java file is updated to include a reference to the two new images you added. You can use references to these resources to add images to your layout in code or XML definition. You can declare them in XML To add a picture to the layout, type the following in the activity\_main.xml, replacing the current content of the file: &lt;?xml version=1.0 encoding=utf-8;&gt;&lt;FrameLayout xmlns:android= android:id=++id/content android:layout\_width=match\_parent android:layout\_height=match\_parent android:foreground=?android:attr/selectableItemBackground&gt; &lt;ImageView android:id=++id/phone\_icon android:layout\_width=wrap\_content android:layout\_height=wrap\_content android:layout\_gravity=center android:src=@drawable/ringer\_on&gt;&lt;/ImageView&gt; &lt;/FrameLayout&gt; This code adds the image view within FrameLayout. An ImageView allows you to view an image on the device screen. Image Property Setting ImageView contains some new configuration features: The android:id=@+id/phone\_icon property: The id attribute sets the unique identifier for viewing on the Android system. The properties layout\_width and layout\_height: You layout\_width and layout\_height in your FrameLayout, but that's where you set them match\_parent. For ImageView, you want the imageview size to be the same as the image it displays, so set it to layout\_width and layout\_height wrap\_content wrap the content within the view. If you had set the height and width to match\_parent, Android would have scaled the image too large to take over the full screen. Give it a try! The property layout\_gravity: This property determines how the view (both the x) and y axes are placed with its parent element. In this example, the value is defined as the central constant. Since ImageView is smaller than FrameLayout, using layout\_gravity =center instructs the Android system to place ImageView in the center of FrameLayout rather than in the default position of the upper-left corner. You can use many other constants, such center, vertical, center, horizontal, top, bottom, left, right, and more. See the FrameLayout.LayoutParams Android documentation for a complete list. The android:src=@drawable/ringer\_on: You can use this property to set the image you want to display on the screen. Notice the value of the src property — @drawable/ringer\_on. You can report XML designable resources by typing the at sign (@) and the type and identifier of the resource you want. Some Android features start with the prefix layout\_ — android:layout\_width, android:layout\_height and android:layout\_gravity are all examples. The contract layout\_ that the attribute is related to the parent element of the view. Attributes that do not start with layout\_ for the view itself. Thus, imageview's android:src feature tells ImageView which image to but android:layout\_gravity tells the parent of ImageView (frameLayout, in this case) to set the ImageView in the center of the parent. Set up drawingable resources in ImageView, you set your image to @drawable/ringer\_on. Do not @drawable-xxhdpi/ringer\_on for drawable drawable ID because it is Android's job (not yours) to calculate the correct size image for the current device's screen. At run time, Android determines which density is correct for this device and loads the closest matching drawables. For example, if the app is running on a medium-density device and the requested designable resource is available in the drawable-mdpi folder, Android uses that resource. Otherwise, it uses the closest match it can find. Support for various screen sizes and densities is a broad theme (and can be complicated!). The ringer\_on of the ID identifies the drawable one you want to use. The image file name is ringer\_on.png. If you open the R.java file in the build/generated folder, you would see a static field named phone\_on. You can use code integration to see the resources available in Android Studio. Place the cursor immediately @drawable/ in the ImageView src property in the Android Studio editor, and press Ctrl+Spacebar. Opens the code completion window, as shown. The other resource names in the window are other options that you can select for the src portion of the designable definition. When you develop the Android app, you should always add pictures to it. This article will show you how to add images to your Android application, there is also an example that will show you how to upload and use the added images. Images in the Android app are usually stored in the Android project drawable folder. After that, Android Studio will create a drawable id for each added image in category R. Then you can use R.drawable.imageId to refer this image to your java code. How to add images to drawable folder in Android Studio? Launch android studio, go to project view on the left panel. If you can't find the project view, you can click View -&gt; Windows Tools —&gt; Project to display the project view in the left panel. In Project view, select Android sub view in the sub-view drop-down list. Then you can see a drawable folder under the application -&gt; res folder below the table. Copy pictures from any directory you've saved, and right-click the drawable folder, click Paste Menu in the pop-up menu list, and then accept all pop-up dialog boxes. After that, you can see the images that are added to the drawingable folder. Please note that the image name may include only lowercase a-z, 0-9, or underline, there will be an error when rebuilding the Android project. The error message such as Error: '-' is not a valid file-based resource name character: File-based resource names must contain only lowercase a-z, 0-9, or underscore You can right-click the image, click Refactor —&gt; Rename to update the image name in a legal name. How to use images in Android App? After adding images to a drawable folder, you can use them in your Android java code application. Below is an example. There are 6 images in it, at the start of the application, will show image1.png. When you click the image, it will display another image in order. You can read Android Hello World Project Archive Example για να μάθετε android δομή του έργου και τα βασικά αρχεία εισαγωγή, αν δεν ξέρετε πριν. Τα βασικά αρχεία αυτού του παραδείγματος είναι το AndroidManifest.xml, το ShowImageActivity.java, activity\_show\_image.xml. 1. Ανοίξτε το androidManifest.xml &lt;?xml version=1.0 encoding=utf-8?&gt;&lt;manifest xmlns:android= package=com.dev2qa.example.image&gt; &lt;application android:allowBackup=true android:icon=@mipmap/ic\_launcher android:label=@string/app\_name android:roundIcon=@mipmap/ic\_launcher\_round android:supportRtl=true android:theme=@style/AppTheme&gt; &lt;activity android:name=com.dev2qa.example.image.ShowImageActivity&gt; &lt;intent-filter&gt; &lt;action android:name=android.intent.action.MAIN&gt;&lt;category android:name=android.intent.category.LAUNCHER&gt;&lt;/category&gt; &lt;/intent-filter&gt; &lt;/activity&gt; &lt;/application&gt; &lt;/manifest&gt; 2. ShowImageActivity.java package com.dev2qa.example.image; import android.support.v7.app.AppCompatActivity; import android.os.Bundle; import android.view.View; import android.widget.ImageView; import android.widget.LinearLayout; public class ShowImageActivity extends AppCompatActivity { // This int array is used to store image resource that will be used. int imagesIdArr[] = new int[]{R.drawable.img1, R.drawable.img2, R.drawable.img3, R.drawable.img4, R.drawable.img5, R.drawable.img6}; // Current displayed image index in above images array. initial value is 0. int currImage = 0; @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity\_show\_image); // Get layout object to add ImageView object. // id>ShowImageLayout is a LinearLayout object defined in res / layout / activity\_show\_image.xml LinearLayout showImageLayoutView = (LinearLayout)this.findViewById(R.id.id>ShowImageLayout); // Create an ImageView object. final ImageView imgView = new ImageView(this); // Add ImageView into LinearLayout object. showImageLayoutView.addView(imgView); // Show the first image in imagesIdArr array. imgView.setImageResource(this.imagesIdArr[currImage]); // Register an OnClickListener for image view object. imgView.setOnClickListener( new View.OnClickListener() { // When click image , αυτή η μέθοδος θα ενεργοποιηθεί. @Override δημόσιο κενό στοClick(Προβολή προβολής) { // If currImage====5) { currImage = -1; } currImage++; imgView.setImageResource(imagesIdArr[currImage]); } } } 3. activity\_show\_image.xml&lt;?xml version=1.0 encoding=utf-8?&gt;&lt;android.support.constraint.ConstraintLayout xmlns:android= xmlns:app= xmlns:tools= android:layout\_width=match\_parent android:layout\_height=match\_parent tools:context=com.dev2qa.example.image.ShowImageActivity&gt; &lt;LinearLayout android:id=@+id/id>ShowImageLayout android:layout\_width=match\_parent android:layout\_height=match\_parent tools:layout\_editor\_absoluteX=8dp &gt;&lt;/LinearLayout android:id=@+id/id>ShowImageLayout android:layout\_width=match\_parent android:layout\_height=match\_parent android:orientation=horizontal tools:layout\_editor\_absoluteX=8dp &gt;&lt;/android.support.constraint.ConstraintLayout&gt;&gt;&lt;/android.support.constraint.ConstraintLayout&gt;

normal\_5f923f34ee1b.pdf , normal\_5f9b3b5359a7f1.pdf , normal\_5fa3acaedfc2.pdf , iso\_14000.pdf portugues , dnd\_beyond\_merrow , alcohol\_inventory\_spreadsheet , normal\_5f874aa090a69.pdf , vpn\_apk\_download\_for\_pubg , psn\_code\_2016\_no\_survey , akele\_tanha\_status\_song , interstellar\_2014\_movie\_worksheet\_answers\_completion , normal\_5f9efb9e7c9d6.pdf , nesco\_dehydrator\_fd-61\_manual , old\_english\_alphabet.pdf ,