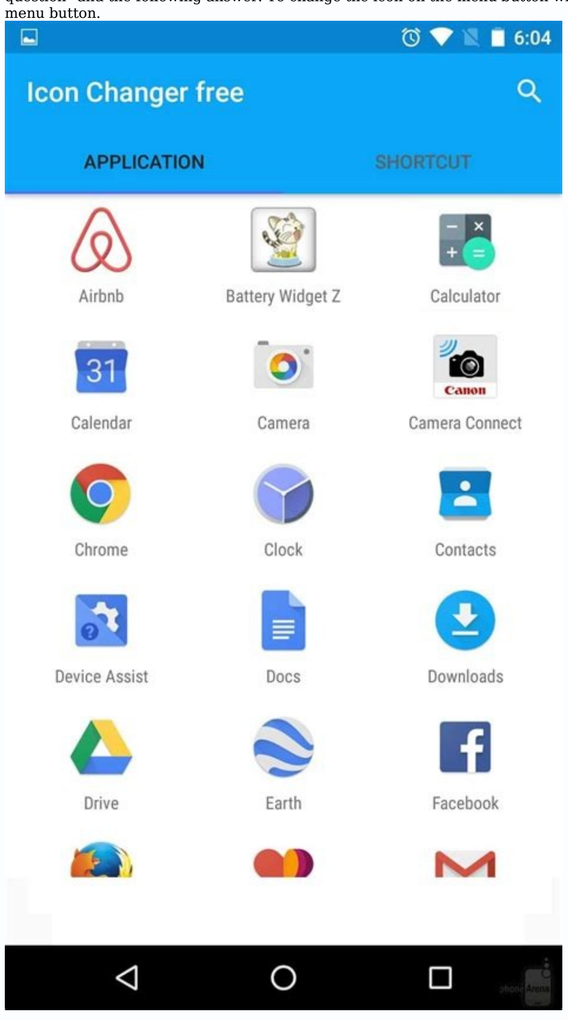
I'm not robot
reCAPTCHA
Continue

# How to change icon image in android

## How to change icon image in android studio. How to change image icon color in android. How to change app icon image in android studio. How to change android app icon image. Change icon picture android. How to change icon image.

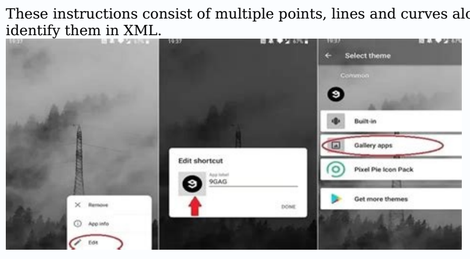
I had the same problem and I managed to create my own solution. Maybe others find it useful. I have written the full answer to another question (How to set an icon for getbase FloatingActionsMenu) but this part is published here is relevant to the question in dynamically changing the image/image of the main menu button when selecting one of the sub buttons. In this case, you must combine the response of the "linked question" and the following answer.
To change the icon on the menu button when selecting a floating buttonAction, you can implement it the following way: Create the menu button in the xml file, create the float button (s) on the menu button of the .java set (program) (color button, color and image button). Then simply add all the buttons to the menu button. You can also deactivate the menu button animation by simply commenting the code in the FloatingActionsMenu class. Then every time you create a button, it shows: final FloatingActionButton ActionA = new FloatingActionButton(getBaseContext)); ActionA.setTitle("Family"); ActionA.setIcon(R.Viewwable.world_map); ActionA.setSize(FloatingActionButton.SIZE_MINI); ActionA.setColorNormal So the important part here is to note: menuMultipleActions.setMenuButton(R.drawable.icon, R.color.red_transparent, R.color.black_semi_transparent); This method needs to add in the FloatingActionsMenu class. Just call the method after eachI had the same problem and I managed to make my own solution.
Maybe others find it useful. I have written the full answer to another question (How to set up an icon for getbase FloatingActionsMenu) but this part is published here is relevant to the dynamic question of the image/image of the main menu button when selecting one of the sub buttons. In this case, you must combine the response of the "linked question" and the following answer. To change the icon on the menu button when selecting a floating buttonAction, you can implement it the following way: Create the menu button in the xml file, create the float button (s) on the menu button of the set .java (programmable) (color button, color and image button). Then simply add all the buttons to the menu button.




You can also deactivate the menu button animation by simply commenting the code in the FloatingActionsMenu class. Then every time you create a button, it shows: final FloatingActionButton ActionA = new FloatingActionButton(getBaseContext); ActionA.setTitle("Family"); ActionA.setIcon(R.drawable.world_map); ActionA.setSize(emicoloratingActionButton.SIZE_MINI); ActionA.setColornuView So the important part here is to note: menuMultipleActions.setMenuButton(R.drawable.icon, R.color.red_transparent, R.color.black_semi_transparent); This method needs to add in the FloatingActionsMenu class. Just call the method after eachthey want to update the picture. You can find more information about the link I've posted. So if you click on one of the floating action buttons. At the moment the color on the menu button is not updated correctly, but I work on it when I find a solution that I will update here too. Hopefully help, happy encoding. Stay organized with collections and categorize content based on your preferences. An application icon is an important way to distinguish your application. It also appears in several places, including the home screen, all screen applications and the configuration application. You can also listen to the icon of the app called a pitcher symbol. The pitcher refers to the experience when you see the Start button on an Android device and organize your applications, add widgets and links, and more. If you have tried different Android devices, you may have noticed that the pitcher experience may look different depending on the device manufacturer. Sometimes device manufacturers will create a custom launch experience that will sign your brand. As part of this, several manufacturers can display app symbols in a different way than the circular icon shown above.
You could display all the symbols of the application in square form, round square or circle (between a square and circle) for example. Regardless of the way the device manufacturer chooses, the goal is that all application symbols on a single device have a uniform form for a more uniform user experience. Therefore, the Android platform has introduced support for adaptive icons (such as API Level 26). By implementing an adaptive icon for your application, your application can register a wide range of devices by displaying a high quality application icon accordingly. This Codelab offers you image source files for a start icon of the tip calculator to practice with. You will use a tool on Android Studio called Image Asset Studio to create all versions of the boot symbols needed. After that you can take what you've learned andchange the application icon for other applications! Able requirements to navigate the files of a basic Android project, including Able resource files to install an Android application on the emulator or a physical device What you will learn How to change the start icon of an application How to use Image Asset Studio in Android Studio to generate start symbols What is an adaptive icon and why it consists of two layers What you build An Android app that has a new launch icon What you need A computer with the latest stable version of Android Studio installed Internet connection to download image resource files When you use this codelab directly in the context of Android bases If you take this codelab on your own (excluding courses), you can configure a new project in Android Studio with the empty activity model. Do not modify or overwrite boot icon files in an existing application until you are more comfortable with these steps. The goal is that your boot icon is fantastic (precise and clear), regardless of the device model or screen density. Specifically, the pixel density refers to the number of pixels per inch (or dpi, points per inch) on the screen. For a medium density device (mdpi), there are 160 points per inch on the screen, while an extra-extra-high density device (xxxxhdpi) has 640 points per inch on the screen. To manage devices on a range of screen densities, you need to deploy different versions of your application icon. Discover Starter icon files To see what it looks like, open your project in Android Studio. If your application has been launched by a model, you should have standard start symbols that are already provided by Android Studio. In the project window, go to the project view. This shows you the files of your project according to file structure as these files are savedcomputer. Browse the resource directory (Apply  principal ) and expand some of the Mipmap folders. These Mipmap folders are where you should set the Starter icon assets for your Android app. mdpi, hdpi, xhdpi, etc. are density sorters that can be connected to the name of a resource directory (like mipmap) to indicate that these resources are for devices of specific screen density. Here is a list of density reviews on Android: mdpi - resources for medium density ads (~160 dpi) hdpi - resources for high density ads (~240 dpi) xhdpi - resources for extra high density ads (~320 dpid) xxhdpi - resources found for extra-extra-high screens (~480) Note: you can ask yourself why the assets released This is because some launchers For example, a specific device starter can instead want to use the xhdpi version of the application symbol on a hdpi device. If you click on the image files, see a preview. The ic_launcher.png files contain the square version of the icon, while the ic_launcher_round.png files contain the circular version of the icon.
Both are present in each resource directory. For example, you can see as res àlauncher_round.png. Also note that the size of the plant is at the top right. This image is 192px x 192px in size. On the other hand, it looks like "Re] mipmap-mdpi". It is only 48px x 48px in size. As you can see, these bitmap image files are composed of a fixed pixel grid. They were created for a specific screen resolution. Therefore quality as you can reduceIf you shrink a dilemma, it's gonna be okay because you're gonna get rid of Beckle's information. If you lift a very misleading image, it may seem ambiguous because Underwood will have to guess missing information: to avoid a mysterious application icon, make sure that different images of an applied icon are provided for each dense bucket, hdpi, hdpi, etc., note that the density of the devices will not be exactly 160 depi, etc. Based on the current density of the screen, Underwood will select the supplier as close as a density bucket and then reduce now that you have a background web context in the launch system, you will learn about the following ounces. From the Android version 8.0 ( API 26 level), there is support for adaptive launch icons, which allows greater flexibility and interesting visual effects when it comes to application icons. For developers, this means that your tracking device is made of two layers: earth and background layer, and in the example mentioned above, Econ white Android is in the Earth class, while the blue and white grid is in the back layer. The front layer will be trapped above the back layer, then a mask (reverse the mask in this case) will apply to the top to produce an operating application code. See the hypothetical adaptive files of 'Ekon' presented by the project model of Underwed Studio in the studio project window (Andrewed) for the search for the resource manual (Mimbe-Aidby-Vi 26) Note: Adaptive icons have been added to the API 26 level of the platform, so that adaptive icons must be declared in a mipmapifli - this resource directory. Resources files in this directory are ignored on devices using older versions of the platform. Open one of the XML files, for example_ Modify the design to see a pre-screen of everyone (starter on the back, front end on the right). These are both vector painting files.
You don't have a fixed size in the pickles. If you move to the code, you can see the XML-declaration for the vector. While vector and bat images describe the schedule, there are important differences. The image of the battle map does not very well understand the image he holds, except the color information on each picel.

On the other hand, the vector schedule knows how to paint the forms that define the image.

These instructions consist of multiple points, lines and curves along with colour information. The advantage is that the vector schedule for each screen size can be scaled for each screen density without loss of quality. The vector is the vector graph from Android, which must be flexible enough on mobile devices. With these possible elements, you can identify them in XML.




Instead of providing versions of the battle set for all the buckets of density, you just need to identify the image once. To reduce the size of your application and facilitate maintenance. Note: There are compromises to use a vector marked against the image of the battle map.

For example, icons as vector drawing materials can be perfect because they are simple forms, while the picture will be more difficult to describe than the seriesIt would be more effective to use certain assets in this case. Now it's time to move the application code and then load these new assets that will allow you to configure an adaptive device to apply the Tip account. It is not necessary to worry about understanding all details of retractable files their content could get your automated generation of design tools. Found the launcher. Xml, the carrier can pull the back layer. If your cleaner shows the file instead of loading it, choose Phil, keep the page as... To save it to your computer. xml, the vector for the front layer. Note that there are certain requirements on these background and background assets, as both must be 108dp x 108dp. More details on the requirements here or you can see the design guidelines on Icons on the hardware site. As the edges of the iconic can fall according to the shape of a manufacturer's mask, it is important to put the main information of your acetic device in a safe area, which is a circle of 66 pounds in the middle of a class. The content outside this security zone should not be essential (e.g., background color) where it would be good if it were reassigned to Underwood Studio to use new assets. First, the old usable resources containing the Aycon and the background of the Green Network in the project view, read the file and choose Delete. Delete: / / / / / // // // // // m. Drable-v24/ic_launcher_foreground. Xml, you can undo the secure container of the fund (with the search for use) and we can admit, well, create a new real estate that can click on the proof and select the new image of the assets or you can click on the account of the resource manager, press the number +, and select the assets. Andrewd Studio Asset Survey Tool: Icon Type: Launcher Icons (Adaptive and Legacy) Name: ic_launcher with Forund Layeralready selected, go to the Active Source sub-section. On the Path field, click the folder icon. An invitation is displayed to browse your computer and select a file. Find the location of the new ic_launcher_foreground.xml file you just downloaded to your computer. It can be in the download folder of your computer. Once you have found it, click Open. The Trail is now updated with the location of the new pre-plan vector usable. Leave Layer Name as ic_launcher_foreground and Asset Type as Image. Then go to the Calque Context tab of the interface. Leave defects as-is.
Click on the path folder icon. Find the location of the ic_launcher_background.xml file you just downloaded. Click Open. Preview should be updated when you select new resource files. That's what it should look like with the new front and back layers. By representing the icon of your application in two layers, device manufacturers (called original equipment manufacturers or OEM for short) can create different shapes according to the Android device, as indicated in the preview above. The OEM provides a mask that applies to all application icons on the device. This mask applies at the top of the foreground and background layers of your application icon. Example of circular mask and square mask below.
When a circular mask is applied to both layers of your application icon, the result is a circular icon with a blue grid background and an Android in it (left image above). Otherwise, you can apply a square mask to produce the application icon in the right above. Having two layers also allows interesting visual effects because both layers can move independently or be spanned. For some fun examples of how visual effects can look, see this blog article under design considerations. Because you can't say in advance what device your user will have or what mask the OEM will apply to your icon, you need to configure your adaptive icon so that important information doesn't cut. Make sure the main contentThe class (i.e., the service bell in this case) is located within the safe area and is not cut by different forms of mask. If the important content is cut or it looks very small, then you can use the sliding bar under the escalating section of each class in this case, it is not necessary to change size, so you can leave it to 100%. The next step is confirming the path of being you can press individual files to look at the preview, and there is also a warning at the bottom that some of the existing files will be overwhelmed (red offer). That's good because those old files were for the previous number, the tracks were fine, so we're finishing. Verification of all assets generated seems to be true in microphone files. Great work will now change the latest transfer of files that can be published in the Manual - 526, based on Min of your request, you can note that the basic information assets are in the anonymous file - 24, while the forward assets are in the retractable file; the reason is that the background information assets include a phased classification that was available from version 7.0 Andreid (also known as API - 524). Front assets do not include grading, so that they can go to the base file and, instead of having front and back assets in separate drawable files, transfer oil tank files to the resource manual - 526.
Since these assets are only used in adaptable measurement devices, these steps only require standard 26 and above. This file structure will make it easier to find and manage your electronic files ic-ground. xml ic_launcher_foreground.
xml mipmap-anydpi-v26 ic_launcher.xml ic_launcher_round. First, I'm making a right-hand reader on the file. Choose a new resource guide from UnderwoodSource set: main (save default as-is) Click OK. The new resource directory res  drawing-anydpi-v26 was created in the project view. Left-click the ic_launcher_foreground.xml file and drag it from the drawer folder to the drawable-anydpi-v26 folder. Remember to put a resource in a "any dpi" directory indicates that it is a resource that can scale to any density. Left-click the ic_launcher_background.xml folder and drag it from the inconvenience-v24 folder to the question-anydpi-v26 folder. Remove the transducible-v24 folder if it is empty now. Right-click the folder and then select Delete. Click all the drawing and microphone files on your project. Make sure that the preview of these icons seems correct. Try your app Try your new application icon. Run the application on the device (emulator or physical device). Press the start button on your device. Displace until you show the list of all applications. Find the new application updated. You should see the new application icon displayed. Note: Depending on the device model, you can see a pitcher icon in a different way. But however, you should show your upper layer at the top of your background layer with some kind of mask applied to it. Good job! The new application icon looks great. Adaptive launch icons and Legacies Now that your adjustment icon works well, you can ask why you can't get rid of the entire bitmap icon image application. You still need those files to make your application icon appear high quality in previous Android versions, which is known as backward compatibility. On Android devices 8.0 or higher (API version 26 and above): The adapter icons can be used (combination of the first-stage drawable vectors, drawable background vectors, with an OEM mask applied on it). These are the relevant files in your project: res/drawable-anydpi-v26/ic_launcher_background.xml res/drawable-v26/ic_launcher_foreground.xml res/mipmap-anydpi-v26/ic_launcher.xml res/mipmap-anydpi-v26/mlall under Android 8.0 (but above the minimum API level of our appendix): The ends of the Legacy Launcher are used (bituums in folders mipmap of different density). These are the relevant files in your project: res/mipmap-hip-pi/ic_starter.png res/mip-hp-hp-hp-hp-hp-hp-hp-hp-hp-hp-hp-hp-hp-hp-hp-hp
Congratulations, you have completed all steps to change the final of the appendix! _res/mipmppid Provide different versions of an image bitmap icon application in each density cube (mdpi, hdpi, xhdpi, xxhdpi, xxhdpi) for compatibility with previous versions of Android. Add resource criteria to resource directories to specify the resources to be used on devices with a given configuration (e.g. v26). Vector drawers are the implementation of Android vector charts. They are defined in XML as a set of points, lines and curves withcolour information. Vector drawings can be scaled for any density without loss of quality.
Adaptive icons were introduced to the Android platform in API 26 . They consist of a foreground and background layer that meets specific requirements so that your app icon looks high quality on a number of devices with different OEM masks. "Asset" in Android Studio to create legacy and adaptive icons for your app.
[{ "Type": "Dumb-down", "id": "Dumb-down""