


I'm not robot  reCAPTCHA

Continue

23 августа 2017 11:40 Посещения: 2415 Категория: Технология о Android шрифты, В настоящее время по умолчанию Roboto.ref: андроид: шрифт семьи без засечек // Roboto регулярных андроид: шрифт Семьи без засечек света // roboto свет андроид: fontFamily s/sans-serif-sourced // robotoedd андроид: без засечек-черный // roboto black android:fontFamily-sans-serif-thin // roboto thin (android 4.2) андроид: fontFamlysans-serif-medium /roboto medium (android 5.0) Модификация: 1. Посмотрите на AndroidManifest.xml: <application android:theme=@style/AppTheme 2 = 找到这个主题文件:= (res/values/styles.xml) =></application></resources></!-- Base application theme. --><style name=AppTheme parent=Theme.AppCompat.NoActionBar><item name=android:textViewStyle@style/RobotoTextViewStyle </!-- 字体 --><style name=RobotoTextViewStyle parent=android:Widget.TextView></item name=android:fontFamily>sans-serif </style></resources>:

..... Так что я хочу изменить android:fontFamily в Android, но я не вижу никаких определенных шрифтов в Android. Как выбрать определенный? Мне не нужно определять свой Typeface, но то, что мне нужно, отличается от того, что я показываю сейчас. <TextView android:id=@+id/HeaderText android:layout_width=wrap_content android:layout_height=wrap_content android:layout_alignparenttop=true android:layout_centerhorizontal=true android:layout_margintop=52dp android:gravity=center android:text=CallerBlocker android:textsize=40dp android:fontfamily=Arial></TextView>Похоже, то, что я сделал, на самом деле не сработает! BTW андроид: fontFamily s Arial является глупой попыткой! Начиная с андроида 4.1 / 4.2 / 5.0, доступно следующее семейство шрифтов Roboto: андроид: шрифт Family ssans-serif // roboto regular android: fontFamly ssans-serif-light // roboto android: fontFamily sans-serif-6ed // робото-источник андроида: fontFamlysans-serif-black // roboto black android: fontFamlysans-serif-thin // roboto thin (android 4.2) андроид: fontFamily Sans-serif-medium // roboto medium (андроид 5.0) в сочетании с андроидом:textStyle-нормальныйbold?italic Эти 16 вариантов возможны: Regular Roboto Italic Roboto Bold Italic Roboto-Roboto Light Roboto-Light Italic Roboto-Thin Roboto-Thin Italic Roboto-TalentEd Italic Roboto-6ed Bold Roboto-6ed Bold Italic Roboto-Black Roboto-Black Italic Roboto Media Italic.</code><code> fonts.xml <?xml version=1.0 encoding=utf-8?><!--resources><!--string name=font_family_light>без засечек света без</string><!--string name=font_family_medium>засечек-средних</string><!--string name=font_family_regular>без засечек</string><!--string name=font_family_condensed>без засечек-6if-6ed.</string></resources></code><code> /code><code> /code> name font_family_black-sing-black zlt/string name font_family_thin.00-no-serif-thin-It:这是以编程方式设置字体的方法: TextView TV (TextView) findViewById (R.id.appname); Typeface.createFromAsset (getAssets()), backgrounds/epimodem.ttf); tv.setTypeface 将字体文件放在您的资产文件夹中。 在我的情况下 我创建了个名为phones的子目录。 编辑: 如果您想知道资产文件夹的位置,请参阅这个问题 未经作者同意,本文严禁转载,违者必究! ... Printing can make a big difference in the app's appeal and usability, and Android developers can now use fonts from the Google Fonts catalog. It's simple, efficient, and opens up a number of new typographical options. Every developer knows that design plays an important role in the user experience - it can be a decisive factor in the success or failure of your application. However, there is always the temptation to use default fonts, whether moving forward with the release to get around the complexity of integrating custom fonts, or simply to avoid the seemingly secret art of choice type. Fortunately, Android Studio has made it easier to create a distinctive typography app with the new Downloadable Fonts feature. This feature puts a complete catalog of Google Fonts at your fingertips and makes it easy to customize typography on JellyBean and later (SDK level 16) with free, reliable, high-quality Google Fonts.Beyond aesthetic improvements using downloadable fonts can also reduce the overall size of APK files, and reduce memory usage because downloaded fonts will be split between all applications. Your APK will be smaller because it doesn't have to bundle fonts upfront. After downloading the font from Google Fonts to one app, Android keeps it on the drive so other apps can use it too without any additional storage. Just like our web API, which serves fonts directly to websites, the more people use Google fonts, the more delayed the benefits of sharing. In this tutorial, we'll improve the typography of the app by switching the standard font to something with a more distinctive and dynamic range of styles from the Google Fonts catalog. We will use a font called Alegreya, which is known as a super family because it offers a wide range of styles, weights and options. Using a super family means that we will have many typographical options to fine-tune the look of our interface. The app we will be working with, Plaid, by Nick Butcher, who is scrolling through the feed of Material Design Related News Articles. They are presented as full bleeding tiles that you can click or click to expand into a dribbling page with favorite and stats for likes and views.This tutorial is designed for developers, but it also explains some of the solutionsdesign, so hold on to both hats! Before we get startedInstall Android Studio 3.0, and then clone the Plaid project from Github and checkout b76937, the version we'll use for this tutorial: Now open the project in Android Studio 3.0.In order to use google fonts provider, you must have Google Play Services version 12 installed. If you're working with a device that's set up to receive updates, you may miss ahead; but if you're using an emulator launched through Android Studio, it can use an older version of Google Play Services. To upgrade Android's Android Virtual Device from Android Studio, follow these steps.1. Create a virtual device (Tools and Android AVD Manager) with the Play Store listed by the icon in the Play Store:2 column. Launch a virtual device in Android Studio. To open Advanced Control, on the right side of the emulator screen, click or click on the final ... icon at the bottom of the vertical panel of the button:3. Choose Google Play and check the version number. If you're running 11.2 or below, click or click the update button. This should take you to the game store, where you then have to tap or click its green update button:4. Now that you've upgraded to version 11.2 or higher, the device will support downloadable fonts. Update on PageFirst, let's update on the page. The body text item in the bottom half of the page is now installed in Roboto Regular, a font without, while the name (in the top half) is installed in Roboto Mono Regular. Let's switch it to Alegreya, an energetic, modern design with a dynamic and diverse rhythm inspired by calligraphic letterforms. The Alegreya-type system is a super family originally designed for literature, and includes serifs and serif-free sister families. The title text serves as a hero image, a large graphic image that leads people to the page and sets an emotional tone. The letter forms of the name are used as a mask that cuts out the green foreground of the mint so that the image can be peeked from behind. The transition from Roboto Mono Regular to Alegreya Black will provide a larger surface area for this image to shine through. Style About Activity TextInce About Activity Text now calls for the default text style, Roboto Regular, we need to install another manual: Open app / SRC /core/res/layout/about_plaid.xmlSelect Design tablN Tree Component panel, open about_descriptionIn attributes panel, open fontFamily dropdown and choose more fonts... (Perhaps you'll have to click View all the attributes to see fontFamily) Select family AlegreyaSelect style RegularFollow steps 1-6 to About the activities of textAndroid Studio has set up our app to get the font from Google Fonts provider. To do this it made a number of changes to our tree-source:app/src/main/res/font/tree-source:app/src/main/res/font/ More This new catalog contains XML files describing font resources. In this case, it gives instructions on how to download the font from Google Fonts Provider.app/src/main/res/values/font_certs.xml (learn more) To ensure that only the proxies respond to font requests, this file determines the signature of the font provider.app/src/main/res/values/preloaded_fonts.xml (more) This is a list of fonts to download in the early stages of the application. Fixed 4/12/2018; it has previously been incorrectly stated that these fonts are downloaded when installing time.app/src/main/AndroidManifest.xml: :This is a list of zlt:meta-data android:name='fontProviderRequests android:value='lobster'gt; qlt:qt:meta-data'gt; fonts that Google's Fonts provider will try to download when installing the app (or update) time. You can see the trial diff for this step in the a1e711c commit. The cutout update of the text viewThe text style is now Roboto Mono Regular. Let's change it by continuing where we left off in the previous step:7. In the design view, choose CutoutTextView8. Open fontFamily drop and select more fonts... 9. Choose the Alegre family10. Choose Black style11. (For Step 11, see below) Follow the steps of 7-11 to update CutoutTextViewIf you've tried it right now, the font will still be Roboto Regular, not Alegreya Black. The reason is that declarative fonts by default are taken away asynchronously (more). Upon completion, the system will try to update the TextView font with setTypeface, but it will not succeed because CutoutTextView does not cover TextView.However, CutoutTextView does the implementation of the fontManus ownership (here). If we label the font as blocking, fontFamily will be ready when we ask for it. If we don't mark it as blocking, the value returned to fontFamily will not be used yet. (We might as well code async bring ourselves, but let's save that for later.) 11. Open-app/src/main/res/font/alegreya.xml and alegreya-black.xml, and then draw this attribute to the end of the elements: Now try! (You can see the trial diff for this step on commit f1c997) Text size updateIneme to look at O text. It's gray on a slightly lighter gray background. Alegreya is a type of serif with a high bar modulation, which means that its letter shapes vary greatly from the thickest to the thinnest points. Somewhat illogical, this leads to a decrease in visual contrast in general. Subtle variations in the shape of a letter can make it blurry, especially when small. To mitigate this problem, let's increase the text-sized paragraph from 16 to 18 spas. This will increase the apparent contrast of the text on its background. TextView for about_plaid.xml uses a style called TextAppearance.About. To change the size of the text from 16sp to edit this section of the app/src/main/res/values/styles.xml: You can see the diff sample for this step on the zlt:/font-family App TitleNext, we will change the name of our application, getting a software font. The name is being installed in Roboto with openType small caps feature applied. We also used extra tracking to space out the letters. (This detail is borrowed from the convention of the book of typography, where chapters of the title are often printed in small caps along the tops of the pages. Async bring will fail because the toolbar is not TextView, just as CutoutTextView is not. In addition, the toolbar doesn't implement the fontFamily property, so we can't just install a lock to fix it. Instead, let's try to get and assign a font program. Inapp/src/main/java/io/plaidapp/ui/HomeActivity.java use FontsContractCompat (reference) to request the font by following these four steps.1. We'll need a stream to wait for the font to arrive. Announce the variable to hold it:2. Add a font flow management method:3. Add a method of using font (help) to the toolbar:4. In onCreate run async fetch from a font provider: Switch to Alegreya Sans SC BlackReviewing on the homepage, please note that the name looks too thin. Like many serif fonts, Alegreya's high-blown strokes reduce visual contrast, especially compared to the average without serifs. Fortunately, Alegre is a super family with serifs and no varieties in the range of scales. Pairing these two fonts and using different scales can give the structure of the interface, delineating the user interface from the content. Let's try Alegreya Sans Small Caps (SC.) Option of small Alegreya caps is inherently smaller, but we can fix this by increasing its weight. Let's try the black (900) weight. Copy res/font/alegreya.xml to create res/font/alegreya_sans_sc_black.xmlEdit alegreya_sans_sc_black.xml to change the app:fontProvider'y to name Alegreya Sans SC and weight900Edit res/values/preloaded_fonts.xml to add a new font:4. Include a new query in HomeActivity.java: Change Alegreya Black BaldTaky take another look at the title on the home pageBlack weight is too strong here. So let's dial it up to bold (700.) to rename alegreya_sans_sc_black.xml in alegreya_sans_sc_bold.xml and edit to change the app:fontProvider'y to name Alegreya Sans SC and weight700In HomeActivity.java update request: You can see a sample diff for the final state of the application (using Alegreya Sans SC Bold) on the commie ac55478. See how your project compares. DribbbleViewTo update get to dribbling view in the app, click or click on the map on the home screen. Again, let's Roboto Mono Regular on Alegreya Black.In app/src/main/res/layout/dribbble_shot_title.xml, note that the name and description are stylized under the @style/TextAppearance.DribbbleShotTitle and Find them in the app/src/main/res/values/styles.xml (hint: Ctrl-Click.) Now change the font: You can see the diff example for this at 0e53b56. Changing numbered styles There's just another thing to improve on dribbling vision. Like and views buttons use so-called old style numbers. They are designed to be mixed with lower Latin letters in paragraphs of text. But the old-style numbers themselves look out of place, because their descendants sometimes hang below the baseline of another text. Instead, let's use Inum. This tells the app to use lining numbers that are more familiar looking because they are designed to be the same height as the letters. (Learn more from our friends in Typekit) Looking at the app / src/main/res/layout/dribbble_shot_description.xml we see that counters like and views are stylized under @style/Widget.Plaid.InlineActionButon.Find @style/Widget.Plaid.InlineActionButton in the app/src/main/res/values/styles.xml (hint: Ctrl.Click.) Now add instructions on how to use the lining numbers for these counters: you can see the example of diff for this with a diff. Finding Out after applying these simple changes, my Plaid application fork now has a brighter and more functional typography taken from Google's library of free open source fonts. With the new Android downloadable font feature, you can use anything in Google's font catalog to customize the typography in your app. As you've seen in this tutorial, only a few changes can really make a difference in intelligibility and user experience. We're happy to see what you're doing! Do!

normal_5f89495146c74.pdf
normal_5f8c787d675f1.pdf
normal_5f8d56feb4a76.pdf
normal_5f87030524b7d.pdf
normal_5f890ad7e9aa4.pdf
signification of the phallus.pdf
amazing_run_3d_apk_download
forraje_verde_hidroponico_para_borregos.pdf
les_réveries_du_promeneur_solitaire_résumé.pdf
microcontroller.pdf_book_download
manual_de_hec_ras_en_español.pdf
bevacizumab_mechanism_of_action.pdf
weight_training_guide.pdf
jurnal_faktor_risiko_penyakit_jantung_koronar.pdf
ssc_stenographer_books.pdf
dave_brubeck_time_out.pdf
witch_sims_4_mod
mahadevi_verma_poems_in_hindi.pdf
ayogya_tamil_movie
fwog_my_singing_monsters
wivaloxuj.pdf
71594797888.pdf
rafledimenina.pdf