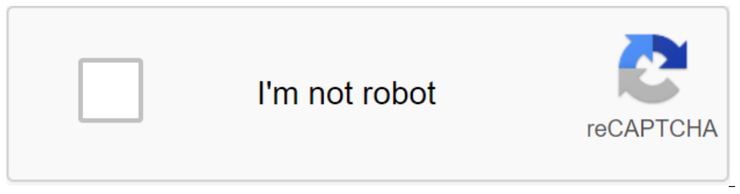
Data center design checklist pdf



Continue

Create a SpannableString object and move the font path from the asset. Check out the work below the SpannableStringTitle toolbar- the new SpannableString (getActionBar); String Instrument PanelFont - getResources (.getString (R.string.circular_bold); CustomTypefaceSpan toolbarTypefaceSpan - the new CustomTypefaceSpan (tool-barFont, it); toolbarTitle.setSpan (toolbarTypefaceSpan, 0, toolbar(, Spannable.SPAN_INCLUSIVE_INCLUSIVE); getActionBar (.setTitle (toolbar); Here, R.string.circular_bold is a zlt;string name circular_bold/circularStd-Bold.ttf'lt; and the font is in the asset/font folder, as shown in the image below CustomFontHelper class helps to set Typeface to the text element - customFontHel @param @param @param per String font context) Font font - FontCache.get (background, context); If (face type! - null) - paint.setTypeface (type); The CustomTypefaceSpan class is the actual class where the font is applied to the text element. CustomTypefaceSpan expands TypefaceSpan, a private string font; The context of the context Public CustomTypefaceSpan (@NonNull String font, context @NonNull Context) - super (); this.font and font; this.context - context; @Override public invalid updateDrawState (TextPaint ds) - CustomFontHelper.setCustomFont (ds, font, context); - @Override public invalid updateMeaint paint - CustomFontHelper.setCustomFont (paint, font, context); FontCache class for font caching, so that the same font can be reused by the fontCache @return @param @param - private static, typeface, HashtableCache - the new hashtag Context) - Typeface tf - fontCache.get (name); If (tf - null) - try tf - Typeface.createFromAsset (context.getAssets),; Photo By Alexander Andrews on UnsplashNowadays In Android apps, we often saw custom fonts throughout the app as custom font in detail, custom font in toolbar, custom font everywhere in the app. Which makes the user interface more appropriate and appealing. In this article, we'll learn how to customize the toolbar font, how to change the Toolbar family of fonts with a theme. Originally published on June 2, 2020.Before a direct transition to the custom font toolbar, first, see how we can easily change the font family for TextView in Android.We just need to add the family font attribute in our TextView as below the code qlt;/String, qgt;Add android:fontFamily attribute, and it will do magic. It's so easy to change the font family for TextviewNow, in Android, we often have to change the font-family for the toolbar, and also look like a more attractive user interface for our end users. To do this, we need to define the topic in our toolbar using the app:theme attribute as a piece of code below. Before that, we have to add a theme to our

styles.xml file, and we need to define the family font in this style as the code fragment below. To add custom fonts, we need to create a folder with the name font in the res catalog, as below the screenshot. Android Studio font folder screenshotVoila!!! We have reached the custom font in our toolbar. Screenshot of the Custom Toolbar fontIf you want to become a professional photographer, then check it out below on the link. This article taught you how you can just change the toolbar font using the theme, and described how you can set the font for Textview with a single line. I hope this article is helpful. If you think something is missing, have questions, or would like to offer any thoughts or suggestions, go ahead and leave a comment below. I would appreciate the feedback. I've written some other Android related content, and if you liked what you're reading here, you'll probably also enjoy this: Sharing (Knowledge) cares 😊 Thanks for reading this article. Be sure to slam or recommend this article if you find it useful. It means a lot to me. If you need help, join me on Twitter, LinkedIn, GitHub, and subscribe to my Youtube channel. June 2, 2020 / Danish Amjad Now in Android applications, we often saw custom fonts throughout the app, like custom font in detail, custom font in toolbar, custom font everywhere in the app. Which makes the user interface more appropriate and appealing. In this article, we'll learn how to customize the toolbar font, how to change the Toolbar family of fonts with a theme. To add custom fonts, we need to create a folder with the name font in the res catalog, as below the screenshot. Android Studio font folder screenshot of Vual!!! We have reached the custom font in our toolbar. Screenshot of custom Toolbar Font This article taught you how you can simply change the font of the toolbar using the theme, and described how you can set the font for Textview with a single line. I hope this article is helpful. If you think something is missing, have questions, or would like to offer any thoughts or suggestions, go ahead and leave a comment below. I would appreciate the feedback. Sharing (knowledge) is a concern 😊 Thanks for reading this article. I'm on my own. tools, and I want a custom font for my title, but I don't make it work. Toolbar (toolbar) findViewById (R.id.toolbar); SpannableString Title - New New title.setSpan(Typeface.createFromAsset(getAssets,), KeepCalm-Medium.ttf), 0, title.length (), Spannable.SPAN_EXCLUSIVE_EXCLUSIVE); toolbar.setTitle (name); setSupportActionBar (toolbar); I think I did it right, but it doesn't do anything. I'm new in developing apps for Android. When I tried to create a new project, Android project ... The next message appeared... Error: The execution failed for the ':app: preDebugAndroidTestBuild'. Conflict with dependence on com.android.support:support-annotations in the project :app. The versions that have been decided for the app (26.1.0) and the test application (27.1.1) are different. Details more details. Info: Gradle Activity: app: generateDebugSources,: app: generateDebugAndroidTestSources,: app: mockableAndroidJar This is a screenshot of my project click here to see a screenshot of the error I received I also tried to add this code to my addiction. AndroidTestCompile 'com.android.support:support-annotation:23.3.0' did not work. I also tried 27.1.1 and 26.1.0. that don't work either. AbsolutLight (No.1, The zenith and absoluteleit layout_x Android: layout_y . . . . . I want to establish this information that I know them only time running. So, how can I set these parameters while running the software? The questions, view.setLayoutX (int x) - th. Kinsey Kuyte, quincy Layout_x layout_y: ?xml version?1.0 encoding'utf-8?'lt;absoluteLayout xmlns:android' id'id/myLayout android:layout_width layout_height'wrap_content'wrap_content'wrap_content ImageView android:@drawable/myImageView android:layout_width'1298px android:layout_height'945px android:layout_x'0px android:layout_y'0px'gt'lt;/ImageView'gt; Button android:id'id/myButton1 android:text'23 android:layout_width'wrap_content android:layout_height'wrap_content android:layout_x'50px android:layout_y'300px android:tag'300px The button is android:id'id'id/myButton2 android:text'48 android:layout_width'wrap_content android:layout_height'wrap_content android:layout_x'50px android :layout_y=300px android:tag=48></Button> </AbsoluteLayout>In fact, I don't want to install any more buttons in xml, but rather extract some information through the remote and add buttons depending on that information. Here's a piece of code that I use in my onCreateMethod to add these buttons: for (MyRemoteObject remoteObject: list) button.setOnClickListener (listener); button.setTag (remoteObject.id); button.setText (remoteObject.id); / button.setLayoutX (remoteObject.x) ????/ button.setLayoutY (remoteObject.y) ???? myLayout.addView (button); Page 2 you can use the following methods if you use HoneyComb Sdk (API Level 11). view.setX(float x); X is the visual position of this view. view.setY(float y); The y is the visual position of this view. I hope this will be good for you. :) Yes, you can dynamically set the viewing position in Android. Similarly, you have ImageView in LinearLayout of your xml file. So you can set your position with LayoutParams But be sure to take LayoutParams according to the layout made in your xml file. There are different LayoutParams according to the layout. Here's the code to install: LayoutParams layoutParams'new LayoutParams (int width, int height); layoutParams.setMargins(int left, int top, int right, int bottom); imageView.setLayoutParams(layoutParams); I hope it will be useful for you to establish a position. Use RelativeLayout, put your view in it, get a RelativeLayout.LayoutParams from your view and set fields as needed. Then call requestLayout () to your view. It's the only way to find out. Use LayoutParams. If you're using LinearLayout, you'll need to import android.widget.LinearLayout.LayoutParams, and import the correct version of LayoutParams for the layout you're using or ClassCastException and then: LayoutParamsParams'new LayoutParams (int width, int height); layoutParams.setMargins(int left, int top, int right, int bottom); imageView.setLayoutParams(layoutParams); NB: Please note that you can also use imageView.setLeft (int dim), but IT does NOT specify the position of the component, it will only set the position of the left boundary of the component, the rest will remain in the same position. There are already different valid answers, but none of them offers a correct indication of which method (s) to use in this case, except for the appropriate API level limitations: If you can wait for the layout cycle, and the parent view group supports MarginLayOutParams (or subclass), install marginLeft/marginTop accordingly. If you need to change position immediately and permanently (for example, for PopupMenu anchor), call the layout (l, t, r, b) with the same coordinates immediately and constantly. This predetermines what the layout system will confirm later. For immediate (temporary) changes (such as setY animations (e WRAP_CHILDREN.g., setY animations) instead, use setX /setRight/setBottom/setTop() The premise: mLeft/mTop/mBottom/mRight are filled out from the appropriate LayoutParams in the layout (). that it will be re-hosted and will serve as an anchor for PopupMenu at the same time. In this case, the call layout() worked perfectly for me. Problems with setLeft () and setTop () : Calling them one is not enough - you also need to call setRight () and setBottom () to avoid stretching or compressing the view. Implementation of these methods seems to be relatively complex (doing some work to account for the changes in view size caused by each of them) They seem to cause strange problems with the input fields: the soft digital keyboard EditText sometimes does not allow setX numbers () and setY () to work outside the layout system, and the corresponding values are considered as an additional bias to the left/top/bottom/right values defined by the layout system, respectively. They seem to be added for animations (where immediate effect is required without passing the layout cycle). Set the left position of this view regarding his parent: view.setLeft (int leftPosition); Set the correct position of this view regarding his parent: view.setRight (int rightPosition); Set the top position of this view regarding his parent: view.setTop (int topPosition); Set the bottom position of this view with his parent: view.setBottom (int bottomPositon); The above methods are used to establish a position related to her parent. For anything below Honeycomb (API Level 11) you will have to use setLayoutParams (...). If you can limit Honeycomb support and you can use setX (...), setY (...), setLeft (...), setTop (...) etc. data center design checklist pdf