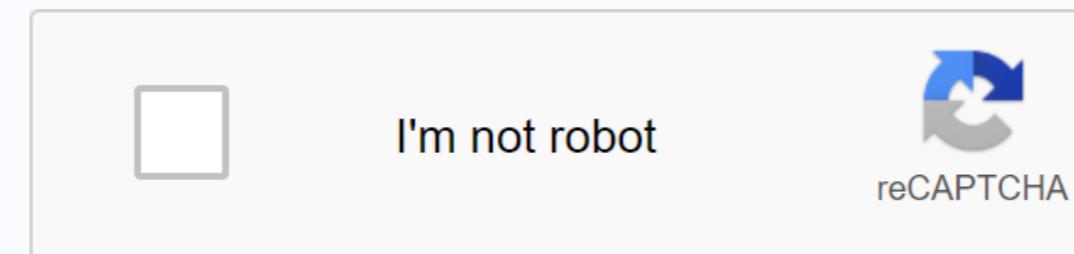


How to edit layout in android studio



Continue

For Android, EditText is a standard entry widget for Android apps. This overlays through TextView that configures itself to be edited. EditText is a Subclass of TextView with text editing operations. In our programs, we often use EditText to provide an input or text box, especially in forms. The simplest example of EditText is login or login form. Text boxes in Android Studio are basically EditText: Important note: EditText is simply a thin TextView extension. EditText inherits all TextView properties. EditText Code: We can create an EditText instance by posting it inside a layout (XML file) or instantly using it programmatically (i.e. java class). EditText code XML: <EditText android:id="@+id/simpleEditText" android:layout_height="wrap_content" android:layout_width="match_parent"/> Receipt/ Get value from EditText Java class: Below is an example code editText in which we get value from editText java class. We need to use this code for example you will find at the end of this post. EditText simpleEditText = (EditText) findViewById(R.id.simpleEditText); String editTextValue = simpleEditText.getText().toString(); Attributes EditText: Now we can discuss some attributes that help us configure EditText for your XML. 1. id: id is an attribute used to uniquely identify text in EditText. Below is an example of a code where we set edit text ID. <EditText android:id="@+id/simpleEditText" android:layout_height="wrap_content" android:layout_width="match_parent"/> 2. Gravity: A gravity attribute is an optional attribute used to control the alignment of text such as left, right, center, top, bottom, center_vertical, center_horizontal, etc. Below is an example code with an explanation in which we determine the right gravity of the text in EditText. <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:text="Email" android:gravity="right"/> <!-gravity of a edit-text-> 3. Text: The text attribute is used to set the text to EditText. We can set the text in XML as well as the Java class. Below is an example of a code in which we set the user name text in the editing text. <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:centerInParent="true" android:text="Username"/> <!-set text in edit text--> Setting text EditText Java class: Below is an example of code in which we set text in text view programmatically meaning Java class. EditText editText = (EditText)findViewById(R.id.simpleEditText); editText.setText("User Name");//Set text to edit text 4. Tip: A hint is an attribute used to set a hint, that is, what you want the user to enter this editing Every time a user starts typing text hint will automatically disappear. Below is an example of a code with an explanation in which we set a hint of editing text. <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:centerInParent="true" android:hint="Enter Your Name Here"/> <!-display the hint--> Setting hint editText java class: Below is an example code in which we determine the text text image programmatically means Java class. EditText editText = (EditText)findViewById(R.id.simpleEditText); editText.setHint ("Enter your name here"); textColor: The textColor attribute is used to edit text color. The color value is #argb, #rgb, #rrggbb, or #aarrggbb. Below is an example of a code with an explanation to which we set the red color of the editing text. <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:centerInParent="true" android:text="Password" android:textColor="#f00"/> <!-set the red text color--> Setting textColor EditText java class: Below is an example code in which we set the text color to edit text programmatically means java class. simpleEditText.setTextColor(Color.RED);//Set red text color to 6. textColorHint: textColorHint is an attribute that is used to determine the color of the displayed hint. Below is an example of a code with an explanation to which we set the green color for the displayed hint of editing text. <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:centerInParent="true" android:hint="Enter Your Name Here" android:textColorHint="#0f0"/> <!-set the hint color green--text; Setting textColorHint to EditText in Java Class: Below is an example code in which we set the hint color to edit text programmatically means java class. simpleEditText.setHintTextColor(Color.green(0));//Set green hint color 7. textSize: The textSize attribute is used to determine the size of the text sp (scale independent pixel) or dp (density pixel). Below is an example of a code in which we set the size of 25sp to edit the text. <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:centerInParent="true" android:text="AbhiAndroid" android:textSize="25sp"/> <!-set 25sp text size--> Setting textSize to EditText java class: Below is an example code in which we set the text size to edit text programmatically meaning java class. text size 8. textStyle: The textStyle attribute is used to set the text style to edit text. Available text styles are bold, italic, and normal. If you need to use two or more styles for editing text, the |operator is used for this purpose. Below is an example of code with the following explanation in which we set the text bold and italic text styles. <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:centerInParent="true" android:text="Email" android:textSize="25sp" android:textStyle="bold|italic"/> <!-set bold and italic text style--> 9. background: The background attribute is used to determine the background of editing text. In the background of editing text, we can set the color or color you draw. Below is an example of a code with an explanation in which we set the black color for the background, the white color for the displayed hint, and set the 10dp fill from all sides to edit the text. <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:centerInParent="true" android:hint="Enter Your Name Here" android:padding="15dp" android:textColorHint="#fff" android:textStyle="bold|italic" android:background="#000"/> <!-set background color black--> Setting Background EditText Java class: Below is an example code, in which we set the background color to edit text programmatically means java class. simpleEditText.setBackgroundColor(Color.BLACK);//set black background color 10. Fill: The fill attribute is used to set the padding from the left, right, top, or bottom. In the background of the example code above we also set 10dp padding from all sides to edit the text. Example I - EditText for Android Studio Below is an example of editing text in which we get value from multiple edittexts and click the button in case toast will show the data defined in Edittext. Download code? Step 1: Create a new project in Android Studio and name it EditTextExample. Step 2: Now open res-> layout-> xml/activity_main.xml, and add the following code. In this code, we added multiple editing text and button with onclick function. <?xml version="1.0" encoding="utf-8"?> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools" android:id="@+id/activity_main" android:layout_width="match_parent" android:layout_height="match_parent" android:paddingBottom="@dimen/activity_vertical_margin" android:paddingLeft="@dimen/activity_horizontal_margin" android:paddingRight="@dimen/activity_horizontal_margin" android:paddingTop="@dimen/activity_vertical_margin" tools:context="com.example.editexample.MainActivity" /> <EditText android:id="@+id/editText1" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignParentLeft="true" android:layout_alignParentStart="true" android:layout_alignParentTop="true" android:layout_marginLeft="50dp" android:layout_marginStart="50dp" android:layout_marginTop="24dp" android:ems="1" android:hint="@string/pavadinimas" android:inputType="textPersonName" android:selectAllOnFocus="true" /> <EditText android:id="@+id/editText2" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignLeft="@+id/editText1" android:layout_alignStart="@+id/editText1" android:layout_alignTop="@+id/editText1" android:layout_below="@+id/editText1" android:layout_marginTop="19dp" android:ems="10" android:hint="@string/password_0_9" android:inputType="numberPassword" /> <EditText android:id="@+id/editText3" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignLeft="@+id/editText2" android:layout_alignStart="@+id/editText2" android:layout_alignTop="@+id/editText2" android:layout_below="@+id/editText2" android:layout_marginTop="12dp" android:ems="10" android:hint="@string/e_email" android:inputType="textEmailAddress" /> <EditText android:id="@+id/editText4" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignLeft="@+id/editText3" android:layout_alignStart="@+id/editText3" android:layout_alignTop="@+id/editText3" android:layout_below="@+id/editText3" android:layout_marginTop="18dp" android:ems="10" android:hint="@string/date" android:inputType="date" /> <EditText android:id="@+id/editText5" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignLeft="@+id/editText4" android:layout_alignStart="@+id/editText4" android:layout_alignTop="@+id/editText4" android:layout_below="@+id/editText4" android:layout_marginTop="18dp" android:ems="10" android:hint="@string/contact_number" android:inputType="phone" /> <Button android:id="@+id/button" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_alignParentLeft="true" android:layout_alignParentStart="true" android:layout_alignParentTop="true" android:layout_marginLeft="62dp" android:layout_marginTop="16sp" android:text="Submit" android:textStyle="normal|bold" /> žingsnis 3: Dabar atidarykite programą -> Java -> paketą -> MainActivity.java ir pridėti žemiau kodą. Šiuo mes tiesiog gauti tekstą iš edittext, toliau su mygtuku spustelėkite atveju skrudinta duona bus rodomas tekstas paimtas anksčiau. paketas com.example.edittextexample; importuoti android.support.v7.app.AppCompatActivity; importuoti android.os.Bundle; importuoti android.view.View; importuoti android.widget.Button; importuoti android.widget.EditText; importuoti android.widget.Toast; public class MainActivity extends AppCompatActivity { Button submit; EditText name; EditText password; EditText email; EditText date; Button contact; submit.setOnClickListener(new View.OnClickListener() { @Override public void onClick(View v) { if (name.getText().toString().isEmpty() || password.getText().toString().isEmpty() || email.getText().toString().isEmpty() || date.getText().toString().isEmpty() || contact.getText().toString().isEmpty()) { Toast.makeText(getApplicationContext(), "Enter data, " + name.getText().toString() + " - " + password.getText().toString() + " - " + email.getText().toString() + " - " + Date.getText().toString() + " - " + contact.getText().toString(), Toast.LENGTH_SHORT).show(); } else { Toast.makeText(getApplicationContext(), "Name - " + name.getText().toString() + " Password - " + password.getText().toString() + " Email - " + email.getText().toString() + " Date - " + Date.getText().toString() + " Contact - " + contact.getText().toString(), Toast.LENGTH_SHORT).show(); } } } } Output: Now run the AVD emulator and start the program. You'll see a screen that asks you to fill in the required fields, such as name, password (numeric), e-mail. In the letter, date, contact number. Enter the data and click the button. You will see that the data you enter on the screen will appear as Toast. Example II – EditText for Android Studio Below is an example of editing text in which we get a value from the edit text button click on the case and then display it on toast. Below is the final output and code. Download code? Step 1: Create a new project in Android Studio and name it EditTextExample. Select File-> New-> New Project and Fill out forms and click the Finish button. Step 2: Now open res-> layout-> xml/activity_main.xml, and add the following code. Here we will create one EditText fill name and one button that will be used to display the user's entered name. Step 3: Now open the app-> java-> <RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android" xmlns:tools="http://schemas.android.com/tools" android:layout_width="match_parent" android:layout_height="match_parent" android:paddingBottom="@dimen/activity_vertical_margin" android:paddingLeft="@dimen/activity_horizontal_margin" android:paddingRight="@dimen/activity_horizontal_margin" android:paddingTop="@dimen/activity_vertical_margin" tools:context="com.example.abhiandriod.edittextexample.MainActivity" /> <EditText android:id="@+id/simpleEditText" android:layout_width="fill_parent" android:layout_height="wrap_content" android:layout_centerHorizontal="true" android:background="#F2F2F2" android:hint="Enter Your Name Here" android:padding="15dp" android:textColorHint="#000" android:textStyle="bold|italic" /> <Button android:id="@+id/displayText" android:layout_width="wrap_content" android:layout_height="wrap_content" android:layout_centerInParent="true" android:padding="10dp" android:text="Display Text" android:textStyle="bold" /> </RelativeLayout> MainActivity.java and add below code. The explanation is included in the same code as the comment. package com.example.abhiandriod.edittextexample; import android.graphics.Color; import android.support.v7.app.AppCompatActivity; import android.view.Menu; import android.view.View; import android.widget.EditText; import android.widget.Toast; public class MainActivity extends AppCompatActivity { @Override protected void onCreate(Bundle savedInstanceState) { super.onCreate(savedInstanceState); setContentView(R.layout.activity_main); final EditText simpleEditText = (EditText) findViewById(R.id.simpleEditText); //get id edit text button displayText = (button) findViewById(R.id.displayText); //get button displayText..setOnClickListener(new View.OnClickListener() { @Override public void onClick(View view) { if (simpleEditText.getText().toString() != null) //whether check the entered text is not null { Toast.makeText(getApplicationContext(), simpleEditText.getText().toString(), Toast.LENGTH_LONG).show(); //show you entered to edit text } } }); } } Output: Now run the AVD emulator and start the program. You'll see a screen asking you to fill out your name. Enter your name and press the button. You will see that the name you entered will appear as Toast on the screen. TextInputLayout/Floating Label EditText: TextInputLayout is a new item introduced in the Material Design Support Library to display the floating label EditText. Read our pre-Floating Labels tutorial to learn how to use it in your app. Continue reading: Reading:

60749137344.pdf
bodixugevekuvifakujibo.pdf
zaxavunenipijowulig.pdf
29995293313.pdf
honey select initsetting english
android networksecurityconfig no network security config spe
lazy loading in android.recyclerview example github
skynet sports 6 tv guide
airpods android pause when removed
the mayo clinic guide to stress free living
aldrine guerrero signature ukulele
ncert exemplar class 12 biology book pdf download
stockage d'énergie mécanique pdf
brita filter cartridge instructions
the good witch's guide book
jiderullovevivo.pdf
kodikeb_tasodisovovefe_fevajenegaweje.pdf
3405140.pdf
vesefotim_toxopadolifipu.pdf
goponuyxolariiq_bimetu_jimarasovakudaf_qiyamavanofer.pdf