**Snackbar**

We can use a Snackbar to display alerts in our application

1. Var myLayout = findViewById<ConstraintLayout>(R.id.layoutId)
2. Snackbar.make(myLayout, “alert message”,Snackbar.LENGTH\_LONG).show()

**Alert :**

1. private fun customAlert(){
2. // first we create a variable to hold an AlertDialog builder
3. val dialogBuilder = AlertDialog.Builder(this)
4. // then we set up the input
5. val input = EditText(this)
6. // here we set the message of our alert dialog
7. dialogBuilder.setMessage("Enter your message:")
8. // positive button text and action
9. .setPositiveButton("Submit", DialogInterface.OnClickListener {
10. dialog, id -> messages.add(input.text.toString())
11. })
12. // negative button text and action
13. .setNegativeButton("Cancel", DialogInterface.OnClickListener {
14. dialog, id -> dialog.cancel()
15. })
16. // create dialog box
17. val alert = dialogBuilder.create()
18. // set title for alert dialog box
19. alert.setTitle("New Message")
20. // add the Edit Text
21. alert.setView(input)
22. // show alert dialog
23. alert.show()
24. }

# Shared Preferences

To use Shared Preferences, we first have to declare our Shared Preferences variable.

1. private lateinit var sharedPreferences: SharedPreferences

 Once we initialize Shared Preferences, we can load and save data.

But before we do that, let's add the following line to our strings.xml file.

1. <string name="preference\_file\_key">com.example.helloworld.PREFERENCE\_FILE\_KEY</string>

 Now we can refer to the Shared Preferences string.

1. sharedPreferences = this.getSharedPreferences(
2. getString(R.string.preference\_file\_key), Context.MODE\_PRIVATE)
3. myMessage = sharedPreferences.getString("myMessage", "").toString() // --> retrieves data from Shared Preferences
4. // We can save data with the following code
5. with(sharedPreferences.edit()) {
6. putString("myMessage", myMessage)
7. apply()
8. }