Flutter: Toast, Snack bar, Alert Dialog:

Toast Notification:

Flutter Toast is also called a **Toast Notification** message. It is a very small message which mainly popup at the bottom of the device screen. It will disappear on its own after completing the time provided by the developers. A developer mostly used the toast notification for showing feedback on the operation performed by the user.

we need to import **fluttertoast** library in Flutter. Flutter provides several properties to the user for showing the toast message, which is given below:

Property	Description
msg	String(Required)
toastlength	Toast.LENGTH_SHORT or Toast.LENGTH_LONG
gravity	ToastGravity.TOP or ToastGravity.CENTER or ToastGravity.BOTTOM
timeInSecForIos	It is used only for Ios (1 sec or more)
backgroundColor	It specifies the background color.
textColor	It specifies text color.
fontSize	It specifies the font size of the notification message.

FlutterToast.cancel(): This function is used when you want to cancel all the requests to show message to the user.

The following steps are needed to show toast notification in Flutter:

- Create a Flutter Project
- Add the Flutter Toast Dependencies in project
- Import the fluttertoast dart package in library
- Implement the code for showing toast message in Flutter

Adding dependencies for flutter toast:

Open the project in **VSCode** and navigate to the **lib** folder. In this folder, open the **pubspec.yaml** file. Here, we need to add the flutter toast library in the dependency section and then click on **get package** link to import the library in your **main.dart** file.

pubspec.yaml

```
1. dependencies:
2.   flutter:
3.     sdk: flutter

4.   fluttertoast: ^8.0.8
```

It ensures that while adding the dependencies, you have **left two spaces** from the left side of a fluttertoast dependency. The fluttertoast dependency provides the capability to show toast notification in a simple way. It can also customize the look of the toast popup very easily.

OR,

Run this command:

With Flutter:

```
$ flutter pub add fluttertoast
```

Before implementing flutter toast notification, import the package:

```
import 'package:fluttertoast/fluttertoast.dart';
```

Now we can write the code for flutter toast.

```
1. Fluttertoast.showToast(
2. msg: 'This is toast notification',
3. toastLength: Toast.LENGTH_SHORT,
4. gravity: ToastGravity.BOTTOM,
5. timeInSecForIos: 1,
6. backgroundColor: Colors.red,
7. textColor: Colors.yellow
8. );
```

Flutter SnackBar:

- Widget used in showing the lightweight message that briefly informs the user when certain actions occur (temporarily display a pop-up message).
- Displays the message for a very short period.
- By default, snackbar displays at the bottom of the screen.

SnackBar Properties

The following are the important properties of the snack bar used in the Flutter:

Attribute Names	Descriptions
content	It is the main content of the snack bar, which is actually a text widget.
duration	It is used to specify how much time the snack bar should be displayed.
action	It is used to take action when the user taps on the snack bar. It cannot be dismissed or cancel. We can only undo or redo in the snack bar.
elevation	It is z-coordinate where the snack bar is placed. It is used to control the shadow size below the snack bar.
shape	It is used to customize the shape of a snack bar.
behavior	It is used to set the location of the snack bar.
bagroundcolor	It specifies the background of the snack bar.

animation	It defines the exit and entrance of the snack bar.

main.dart | container.dart | snack.dart

lib > screen > snack.dart > _SnackExampleState

```dart
import 'package:flutter/material.dart';

class SnackExample extends StatefulWidget {
  const SnackExample({Key? key}) : super(key: key);

  @override
  _SnackExampleState createState() => _SnackExampleState();
}

class _SnackExampleState extends State<SnackExample> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: ElevatedButton(
        onPressed: () {
          final snackBar = SnackBar(
            duration: const Duration(seconds: 5),
            content: const Text('Example'),
            action: SnackBarAction(
                label: 'Undo',
                onPressed: () {
                  //code for snackbar acitons
                }), // SnackBarAction
          ); // SnackBar
          ScaffoldMessenger.of(context).showSnackBar(snackBar);
        },
        child: const Text('show snackbar'),
      ), // ElevatedButton
    ); // Scaffold
  }
}

```

Alert Dialog and confirmation Dialog:

Alert Dialog box informs the user about the situation that requires acknowledgment. Alert Box is a prompt that takes user confirmation. The very basic use of the alert box is used when we close the app, usually, we are notified with a prompt whether you want to exit or not. That's an alert box.

The below-added code shows how to perform alert Dialog box in flutter. I have used a button (Elevated Button in flutter) to trigger the alert dialog box. In its on the pressed property, we have to use the *showDialog* widget of flutter. It takes context and a builder. In builder, we provide the *AlertDialog* widget with **title**, **content**(Description of a title), and **actions** (Yes or no buttons), and our alert dialog box is ready to use.

Key Properties of Alter Dialog Box:

- **actions:** The set of actions that are displayed at the bottom of the dialog
- **title:** The title of the dialog is displayed in a large font at the top of the dialog.
- **content:** This gives a message/ description about the title which you have provided to the Alert Dialog box.
- **backgroundColor:** It provides the background color to the widget which is being used in.
- **elevation**: Elevation provided height to the widget, It gives a default shadow to the widget.

Flutter provides its own show Dialog widget which is used to show Dialog box.

```dart
class _AlertExampleState extends State<AlertExample> {
  //functions for alert dialog
  showAlertDialog(BuildContext context) {
    //set up the buttons
    Widget okbtn = ElevatedButton(
      onPressed: () {},
      child: const Text('ok'),
    ); // ElevatedButton
    Widget cancelbtn = ElevatedButton(
      onPressed: () {
        //to dismiss dialog
        Navigator.of(context).pop();
      },
      child: const Text('Cancel'),
    ); // ElevatedButton
    // set up the AlertDialog
    AlertDialog alert = AlertDialog(
      backgroundColor: Colors.teal,
      title: const Text('AlertDialog'),
      content: const Text('would you like to continue?'),
      actions: [
        okbtn,
        cancelbtn,
      ],
    ); // AlertDialog
    //show the Dialog
    showDialog(
      context: context,
      builder: (BuildContext context) {
        return alert;
      },
    );
  }
```

```dart
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Alert Example'),
      ), // AppBar
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            showAlertDialog(context);
          },
          child: const Text('show'),
        ), // ElevatedButton
      ), // Center
    ); // Scaffold
  }
}
```

References Git: https://github.com/samsunk/alert_example.git