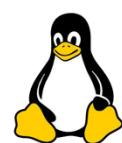
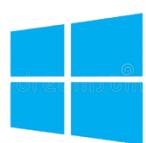
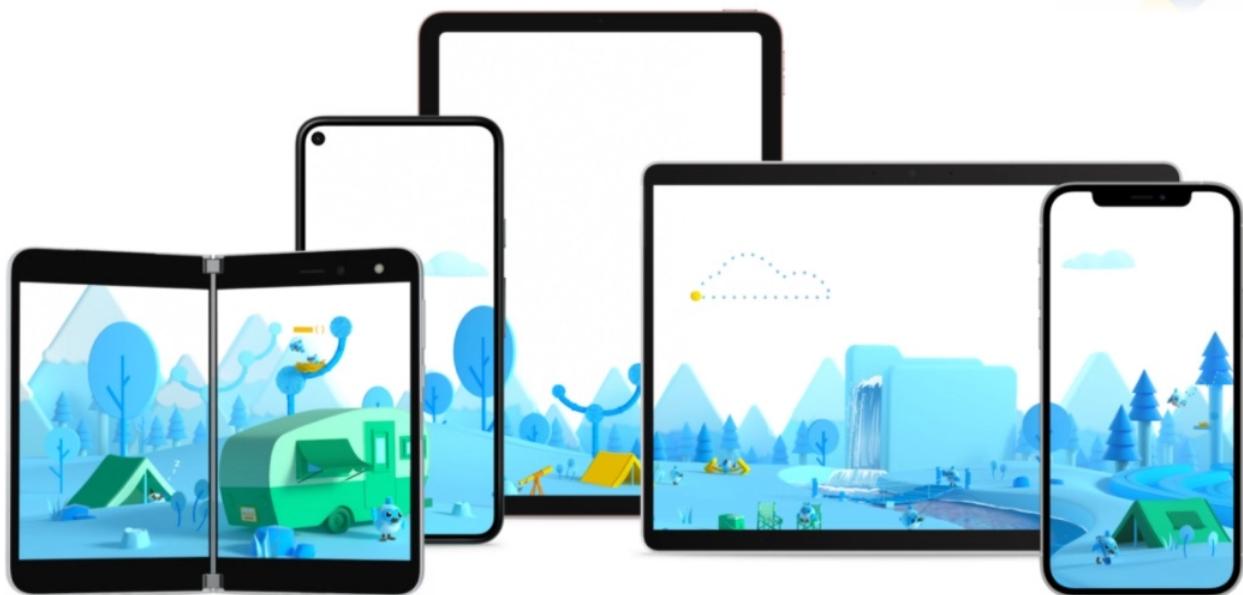




Build Your IT Skill

ETEC IT PROFESSIONAL TRAINING CENTER

Widget & MaterialApp



ក្រុមក្រោនបេយ័ណ៍ និង អាជីវការ
Software Developer

I. About Flutter

Flutter គឺជាព័ត៌មាននៃកីឡា Technology មួយដែលបានក្រុងការរំពេលដោយក្រុមហ៊ុន Google និងត្រូវបាន release ក្នុងខែឧសភា ឆ្នាំ 2017 និងបានប្រើប្រាស់ ការសរស់សរ Flutter នេះ។

Flutter មានសម្រាប់ក្នុងការប្រើប្រាស់លើ Multiple OS (Windows , MacOS , Linux , Chrome OS) និងអាចរំពេលទិន្នន័យ Developer build application បានជាវិស័យ Multi- Platform :

- Mobile App (Android & IOS)
- Desktop App (Mac , Windows , Linux)
- Web App
- Embedded System

II. Understand Widget

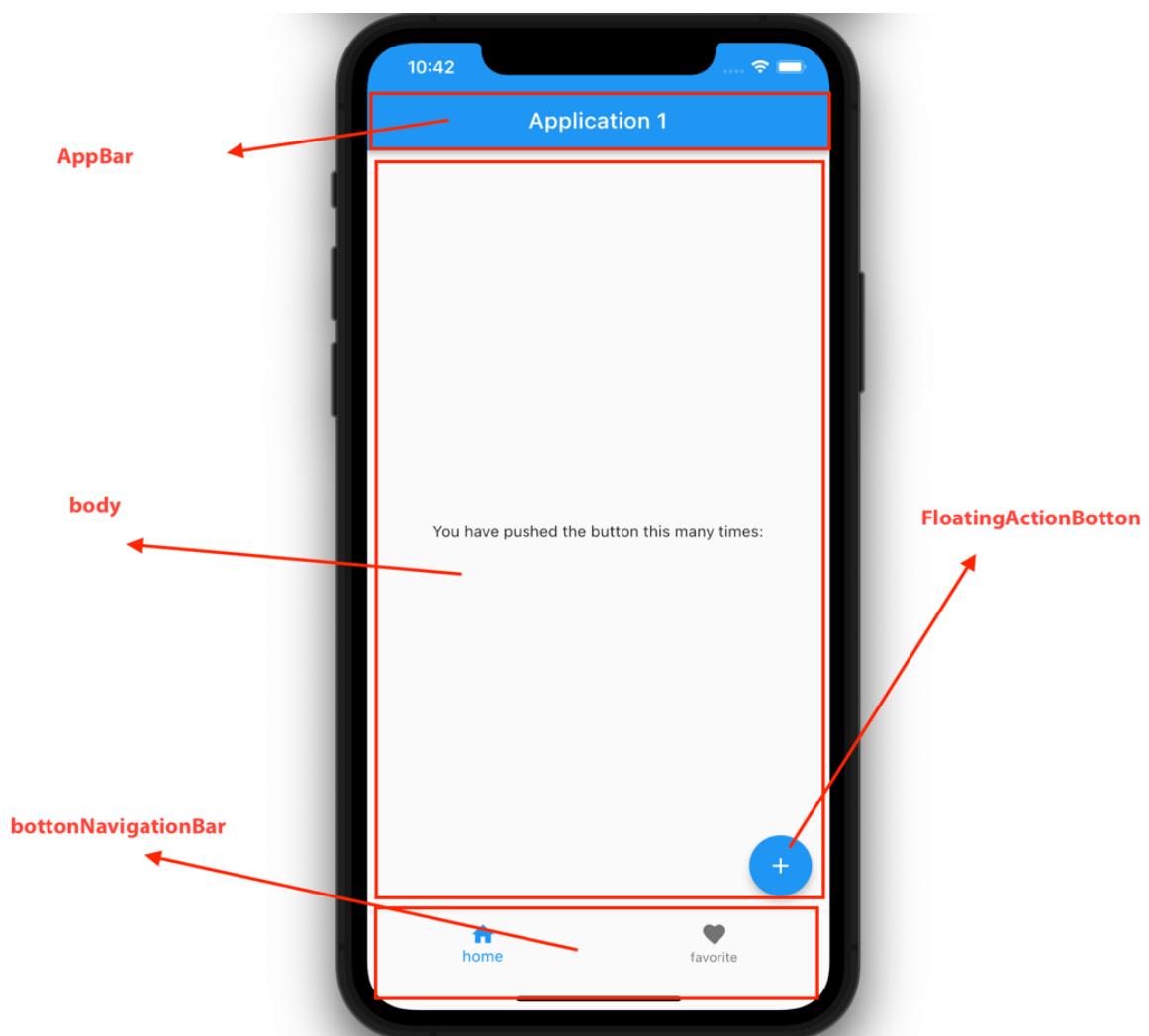
StatelessWidget	StatefullWidget
<ul style="list-style-type: none">• Shortcut (stl)• Can't change state	<ul style="list-style-type: none">• Shortcut (stf)• Can change state and have overriding method [initState(),didChangeDependencies(),setState(),didUpdateWidget(),dispose() ,.....]

III. MaterialApp

i. Scaffold

Scaffold គឺជា widget in Flutter ដែលប្រើក្នុងការ implements the basic material design visual layout structure %

Example:



ii. main , MaterialApp & Scaffold

- main() ជាដៃនើកគោលសំខាន់ក្នុងការធ្វើនៅយោង Program ដើម្បីបានសរសេរអាជ Run ចេញជាការ Appliction បាន
- MaterialApp() MaterialApp() មាន property សំខាន់មួយចំនួនដូចជា
 - debugShowCheckedModeBanner: false សម្រាប់ ដក Debugនៅលើបី ចេញ
 - home: សម្រាប់បង្កើត class បន្ទាប់មកប្រើ
 - theme: សម្រាប់ដាក់ពណិតនៅលើ MaterialApp ទាំងមូល
- Scaffold ជាដៃនើកដែលសរសេរចេញជាការ App UI និង property ដូចជា:
 - Drawer:Drawer()
 - appBar:AppBar()
 - Body:
 - floatingActionButton:FloatingActionBotton()

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

void main() {
    runApp(const MyApp());
}

class MyApp extends StatelessWidget {
    const MyApp({Key? key}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return MaterialApp(
            debugShowCheckedModeBanner: false,
            title: 'Flutter Demo',
            theme: ThemeData(
                primarySwatch: Colors.blue,
            ),
            home: const MyHomePage(title: 'Flutter Demo Home Page'),
        );
    }
}
```

i. AppBar & Drawer

AppBar មាន property ដូចជា :

- leading:
- title:
- bottom:
- backgroundColor:
- action:[]

The screenshot shows the Dart DevTools interface. On the left, there's a sidebar with various icons. The main area displays the code for `main.dart`:

```
main.dart — app
lib/main.dart:23:23: Error: The argument type 'String?' can't be assigned to the parameter type 'Key?'. (argument_type_not_assignable at [app] lib/main.dart:23)
23:   class MyHomePage extends StatefulWidget {
24:     const MyHomePage({Key? key, required this.title}) : super(key: key);
25:
26:     final String title;
27:
28:     @override
29:     State<MyHomePage> createState() => _MyHomePageState();
30:
31:
32:   class _MyHomePageState extends State<MyHomePage> {
33:     @override
34:     Widget build(BuildContext context) {
35:       return Scaffold(
36:         drawer: const Drawer(),
37:         appBar: AppBar(
38:           title: const Text('Application 1'),
39:           actions: const [
40:             Icon(Icons.search),
41:             SizedBox(
42:               width: 20,
43:             ), // SizedBox
44:             Icon(Icons.logout_outlined),
45:             SizedBox(
46:               width: 20,
47:             ) // SizedBox
48:           ],
49:         ), // AppBar
50:       ); // Scaffold
51:     }
52:
53:   }
```

On the right, there's an iPhone 11 iOS 16.5 simulator window showing a blank white screen. The status bar indicates it's 11:19. The bottom of the screen shows the text "Application 1". The bottom of the DevTools interface has a toolbar with various icons.

ii. Body

1. Text and Center

The screenshot shows the Android Studio interface with the following details:

- File:** main.dart
- Project:** app1
- Code Editor:** The main.dart file is open, showing the code for a StatelessWidget named MyHomePage. It includes a constructor, properties for title and drawer, and a build method that returns a Scaffold with a Drawer, AppBar, and a Centered Text widget.
- Build Tools:** The build section of the code editor is visible, showing the creation of a _MyHomePageState object.
- Simulator:** An iPhone 11 simulator running iOS 15.5 is displayed, showing the application's splash screen with the text "Hello Flutter".
- Toolbar:** The top right of the screen shows the simulator's status bar with the time (10:42), battery level, signal strength, and connectivity icons.

2. Row and Column

The screenshot shows the Android Studio interface with the following details:

- Code Editor:** The main window displays the `main.dart` file. The code defines a `MyHomePage` stateful widget with a title and a body containing a `Row` of three red `Text` widgets.
- Simulator:** An iPhone 11 simulator running iOS 15.5 is shown on the right, displaying the application's UI. The top bar shows the time as 11:03 and the title as "Application 1". The screen content includes the text "Hello", "Flutter", and "App" in red font.
- Side Panel:** The left sidebar contains various icons for navigation and project management.
- Status Bar:** At the bottom, the status bar shows "Debug my code", "Ln 51, Col 1", "Spaces: 2", "UTF-8", "LF", "Dart", "Dart DevTools", "Flutter: 3.0.6-0.0.pre.1", "iPhone 11 (ios simulator)", and a few other small icons.

The screenshot shows the Dart code editor with the file `main.dart` open. The code defines a `MyHomePage` stateful widget with a `Scaffold` containing a `Column` of three red text elements: "Hello", "Flutter", and "App". To the right, an iPhone 11 simulator displays the app with these three text elements visible.

```
main.dart X
main.dart — app1
lib > main.dart > _MyHomePageState > build
21 }
22
23 class MyHomePage extends StatefulWidget {
24   const MyHomePage({Key? key, required this.title}) : super(key: key);
25
26   final String title;
27
28   @override
29   State<MyHomePage> createState() => _MyHomePageState();
30 }
31
32 class _MyHomePageState extends State<MyHomePage> {
33   @override
34   Widget build(BuildContext context) {
35     return Scaffold(
36       drawer: const Drawer(),
37       appBar: AppBar(
38         title: const Text('Application 1'),
39       ), // AppBar
40       body: Column(
41         mainAxisAlignment: MainAxisAlignment.spaceAround,
42         children: const [
43           Text('Hello ', style: TextStyle(fontSize: 25, color: Colors.red)),
44           Text('Flutter', style: TextStyle(fontSize: 25, color: Colors.red)),
45           Text('App', style: TextStyle(fontSize: 25, color: Colors.red)),
46         ],
47       ), // Column
48     ); // Scaffold
49   }
50 }
51 }
```

3. Container

The screenshot shows the Dart code editor with the file `main.dart` open. The code is identical to the previous one, but it uses a `Container` widget instead of a `Column` in the `body` of the `Scaffold`. The `Container` has a height and width of 200 and contains a centered text element "Widget1". To the right, an iPhone 11 simulator displays the app with a single blue square containing the text "Widget1".

```
main.dart X
main.dart — app1
lib > main.dart > _MyHomePageState > build
23 class MyHomePage extends StatefulWidget {
24   const MyHomePage({Key? key, required this.title}) : super(key: key);
25
26   final String title;
27
28   @override
29   State<MyHomePage> createState() => _MyHomePageState();
30 }
31
32 class _MyHomePageState extends State<MyHomePage> {
33   @override
34   Widget build(BuildContext context) {
35     return Scaffold(
36       drawer: const Drawer(),
37       appBar: AppBar(
38         title: const Text('Application 1'),
39       ), // AppBar
40       body: Center(
41         child: Container(
42           height: 200,
43           width: 200,
44           color: Colors.blue,
45           child: const Center(
46             child: Text('Widget1'),
47           ), // Center
48         ), // Container
49       ), // Center
50     ); // Scaffold
51   }
52 }
53 }
```

4. Expanded

The screenshot shows the Android Studio interface. On the left is the code editor with the file `main.dart` open. The code defines a `MyHomePage` class that extends `StatefulWidget`. It contains a `_MyHomePageState` state class. The `build` method creates a `Scaffold` with a `Drawer` and an `AppBar` titled "Application 1". The `body` is a `Row` containing two `Expanded` children, each a `Container` with a height of 200 and a center text ("Widget1"). On the right, an iPhone 11 simulator displays the app with a blue section labeled "Widget1" and a red section labeled "Widget1".

```
lib > main.dart > _MyHomePageState > build
23 class MyHomePage extends StatefulWidget {
24   const MyHomePage({Key? key}) : super(key: key);
25   @override
26   State<MyHomePage> createState() => _MyHomePageState();
27 }
28
29 class _MyHomePageState extends State<MyHomePage> {
30   @override
31   Widget build(BuildContext context) {
32     return Scaffold(
33       drawer: const Drawer(),
34       appBar: AppBar(
35         title: const Text('Application 1'),
36       ), // AppBar
37       body: Row(
38         children: [
39           Expanded(
40             child: Container(
41               height: 200,
42               color: Colors.blue,
43               child: const Center(
44                 child: Text('Widget1'),
45               ), // Center
46             ), // Container
47           ), // Expanded
48           Expanded(
49             child: Container(
50               height: 200,
51               color: Colors.red,
52               child: const Center(
53                 child: Text('Widget1'),
54               ), // Center
55             ), // Container
56           ), // Expanded
57         ],
58       ), // Row
59     ); // Scaffold
60 }
```

5. Image

The screenshot shows the Android Studio interface. On the left is the code editor with the file `image_screen.dart` open. The code imports `flutter/material.dart` and defines a `FlutterImage` class that extends `StatelessWidget`. It contains an `AppBar` with the title "Flutter Images" and a `body` that displays a `NetworkImage` from the URL `https://docs.flutter.dev/assets/images/dash/Dashatars.png`. On the right, an iPhone 11 simulator displays the app with the title "Flutter Images" and two cartoon characters.

```
lib > image_screen.dart > FlutterImage
1 import 'package:flutter/material.dart';
2
3 class FlutterImage extends StatelessWidget {
4   const FlutterImage({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10         title: const Text('Flutter Images'),
11       ), // AppBar
12       body: const Image(
13         image: NetworkImage(
14           'https://docs.flutter.dev/assets/images/dash/Dashatars.png',
15         ), // Image
16       ); // Scaffold
17     }
18   }
19 }
```

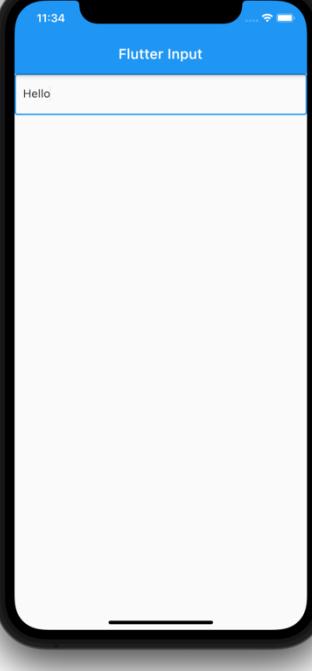
6. ListTile , Card

The screenshot shows the Android Studio interface with the code editor open to the file `image_screen.dart`. The code defines a `FlutterImage` widget that extends `StatelessWidget`. It uses `Scaffold`, `AppBar`, and `Column` widgets to build a list of items. Each item is a `ListTile` with a `CircleAvatar` leading icon, a title, a subtitle, and a trailing delete icon. Some tiles have a child `ListTile`. The code ends with a `Card` containing a `Column` with a single item. The `build` method returns the `Scaffold`.

```
class FlutterImage extends StatelessWidget {
  const FlutterImage({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Flutter Images'),
      ),
      body: Column(
        children: const [
          ListTile(
            leading: CircleAvatar(child: Text('1')),
            title: Text('Mr. Dash'),
            subtitle: Text('dydy'),
            trailing: Icon(Icons.delete),
          ),
          Card(
            child: ListTile(
              leading: CircleAvatar(child: Text('2')),
              title: Text('Mr. Dash'),
              subtitle: Text('dydy'),
              trailing: Icon(Icons.delete),
            ),
          ),
        ],
      ),
    );
  }
}
```

7. **TextField**

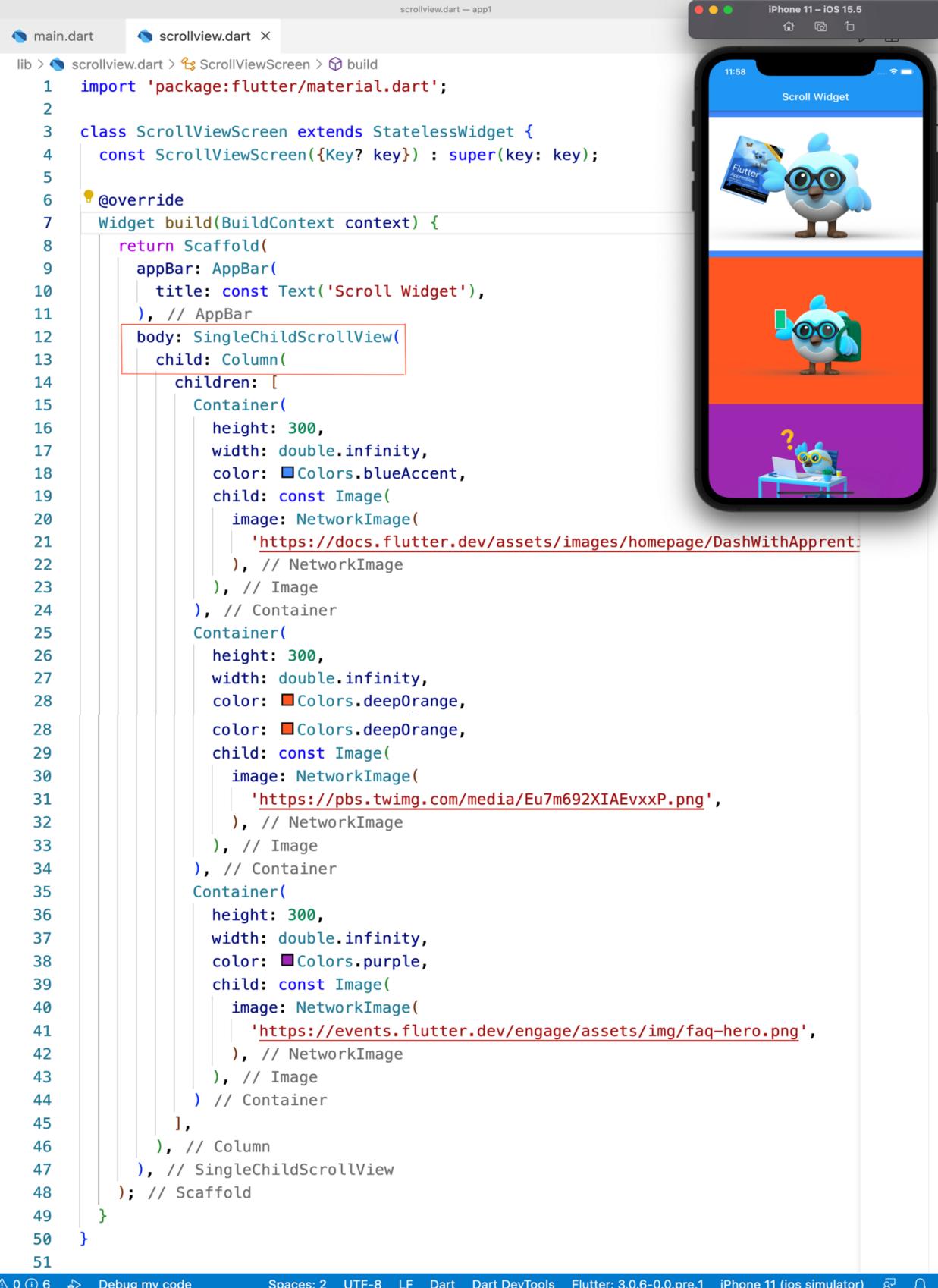


The screenshot shows a Flutter application running on an iPhone 11 simulator. The app's title is "Flutter Input". It contains a single text input field with the placeholder text "Hello".

```
lib > input.dart > InputDataScreen > build
1 import 'package:flutter/material.dart';
2
3 class InputDataScreen extends StatelessWidget {
4   const InputDataScreen({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10         title: const Text('Flutter Input'),
11       ), // AppBar
12       body: const TextField(
13         decoration: InputDecoration(
14           border: OutlineInputBorder(),
15         ), // InputDecoration
16       ), // TextField
17     ); // Scaffold
18   }
19 }
20 }
```

The code in the screenshot is identical to the one provided in the question, defining a StatelessWidget named InputDataScreen that returns a Scaffold with a central TextField containing the placeholder "Hello".

8. SingleChildScrollView

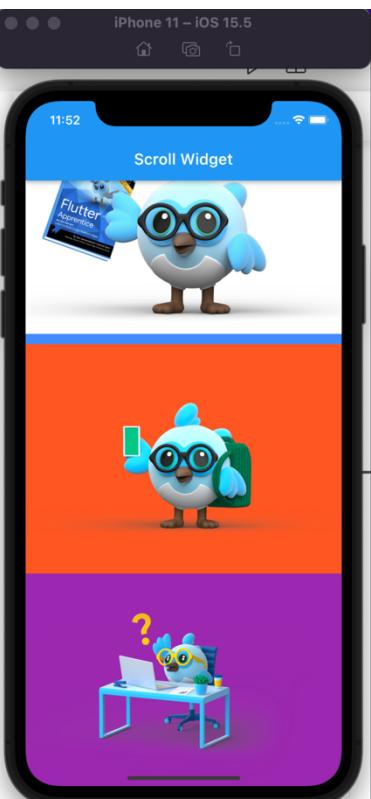


The screenshot shows the Flutter development environment. On the left, the code editor displays `scrollview.dart` with the following content:

```
lib > scrollview.dart > ScrollViewScreen > build
1 import 'package:flutter/material.dart';
2
3 class ScrollViewScreen extends StatelessWidget {
4   const ScrollViewScreen({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10         title: const Text('Scroll Widget'),
11       ), // AppBar
12       body: SingleChildScrollView(
13         child: Column(
14           children: [
15             Container(
16               height: 300,
17               width: double.infinity,
18               color: Colors.blueAccent,
19               child: const Image(
20                 image: NetworkImage(
21                   'https://docs.flutter.dev/assets/images/homepage/DashWithApprentice.png',
22                 ), // NetworkImage
23               ), // Image
24             ), // Container
25             Container(
26               height: 300,
27               width: double.infinity,
28               color: Colors.deepOrange,
29               child: const Image(
30                 image: NetworkImage(
31                   'https://pbs.twimg.com/media/Eu7m692XIAEvxxP.png',
32                 ), // NetworkImage
33               ), // Image
34             ), // Container
35             Container(
36               height: 300,
37               width: double.infinity,
38               color: Colors.purple,
39               child: const Image(
40                 image: NetworkImage(
41                   'https://events.flutter.dev/engage/assets/img/faq-hero.png',
42                 ), // NetworkImage
43               ), // Image
44             ), // Container
45           ],
46         ), // Column
47       ), // SingleChildScrollView
48     ); // Scaffold
49   }
50 }
```

The right side of the interface shows the iOS simulator running on an iPhone 11 with the title "Scroll Widget". The screen displays three stacked containers with images of a cartoon bird holding a book, a cartoon character with a backpack, and a cartoon character at a desk.

9. ListView



```
scrollview.dart — app1
main.dart scrollview.dart ×

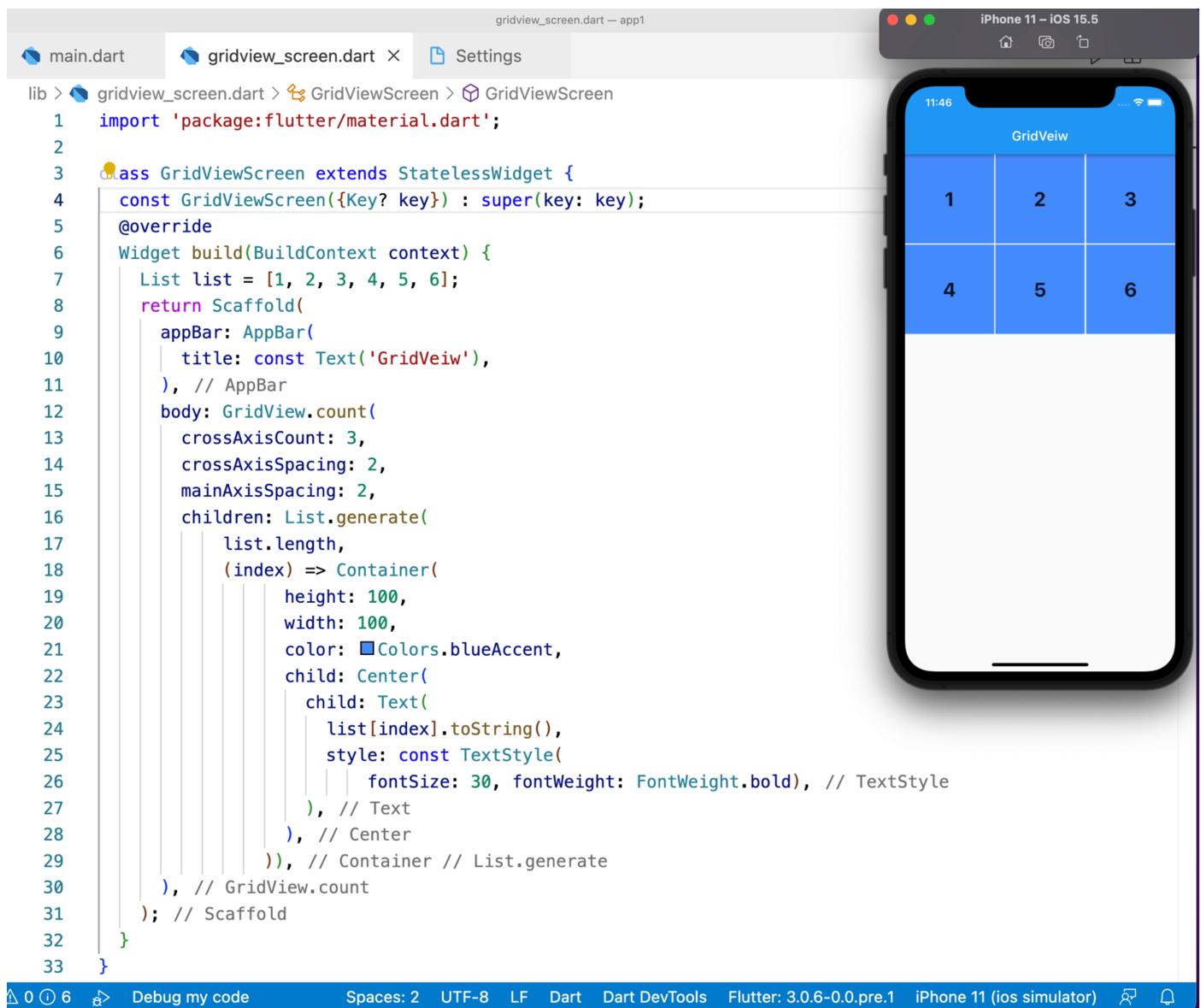
lib > scrollview.dart > ScrollViewScreen > build
1 import 'package:flutter/material.dart';
2
3 class ScrollViewScreen extends StatelessWidget {
4   const ScrollViewScreen({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10         title: const Text('Scroll Widget'),
11       ), // AppBar
12       body: ListView(
13         children: [
14           Container(
15             height: 300,
16             width: double.infinity,
17             color: Colors.blueAccent,
18             child: const Image(
19               image: NetworkImage(
20                 'https://docs.flutter.dev/assets/images/homepage/DashWithAppren'
21               ), // Container
22             Container(
23               height: 300,
24               width: double.infinity,
25               color: Colors.deepOrange,
26               child: const Image(
27                 image: NetworkImage(
28                   'https://pbs.twimg.com/media/Eu7m692XIAEvxxP.png'), // Network
29               ), // Container
30             Container(
31               height: 300,
32               width: double.infinity,
33               color: Colors.purple,
34               child: const Image(
35                 image: NetworkImage(
36                   'https://events.flutter.dev/engage/assets/img/faq-hero.png')), // Network
37               ), // Container
38           ],
39         ), // ListView
40       ); // Scaffold
41   }
42 }

△ 0 ⏪ 6 ⏴ Debug my code Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 3.0.6-0.0.pre.1 iPhone 11 (ios simulator) ⌂ ⌂
```

10. GridView

ចំណេះ GridView មានដូចជា

- `GridView.count()`
- `GridView.builder()`
- `GridView.extent()`
- `GridView.custom()`

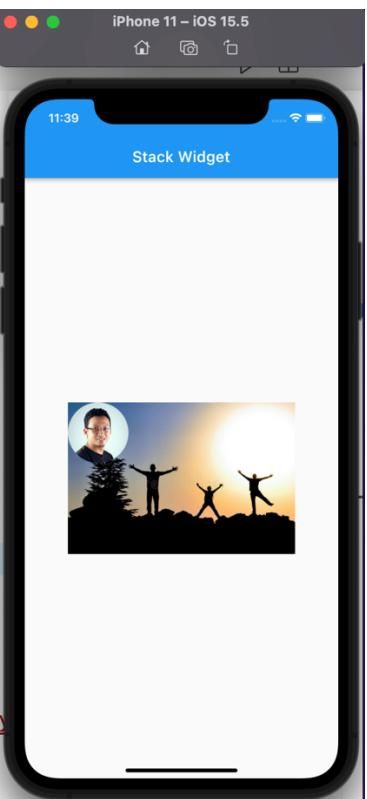


The screenshot shows the Flutter IDE interface. On the left, there are tabs for `main.dart`, `gridview_screen.dart` (which is the active file), and `Settings`. Below the tabs, the code for `gridview_screen.dart` is displayed:

```
lib > gridview_screen.dart > GridViewScreen > GridViewScreen
1 import 'package:flutter/material.dart';
2
3 class GridViewScreen extends StatelessWidget {
4   const GridViewScreen({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     List list = [1, 2, 3, 4, 5, 6];
9     return Scaffold(
10       appBar: AppBar(
11         title: const Text('GridVeiw'),
12       ), // AppBar
13       body: GridView.count(
14         crossAxisCount: 3,
15         crossAxisSpacing: 2,
16         mainAxisSpacing: 2,
17         children: List.generate(
18           list.length,
19           (index) => Container(
20             height: 100,
21             width: 100,
22             color: Colors.blueAccent,
23             child: Center(
24               child: Text(
25                 list[index].toString(),
26                 style: const TextStyle(
27                   fontSize: 30, fontWeight: FontWeight.bold), // TextStyle
28               ), // Text
29             ), // Center
30           ), // Container // List.generate
31         ), // GridView.count
32       ); // Scaffold
33     }
34 }
```

On the right, an iPhone 11 simulator is shown displaying a 2x3 grid of numbers from 1 to 6. The title of the app is "GridVeiw". The code uses `GridView.count` to create a 2x3 grid with a total of 6 items.

11. Stack



```
stack_widget.dart X
lib > stack_widget.dart > MyWidget > build
1 import 'package:flutter/material.dart';
2
3 class MyWidget extends StatelessWidget {
4   const MyWidget({Key? key}) : super(key: key);
5
6   @override
7   Widget build(BuildContext context) {
8     return Scaffold(
9       appBar: AppBar(
10         title: const Text('Stack Widget'),
11       ), // AppBar
12       body: Center(
13         child: Stack(children: [
14           Container(  SizedBox for whitespace.
15             height: 200,
16             width: 300,
17             child: const Image(
18               image: NetworkImage(
19                 'https://thumbs.dreamstime.com/b/awesome-endless-energy',
20               ), // NetworkImage
21             ), // Image
22           ), // Container
23           Positioned(
24             child: Container(
25               height: 80,
26               width: 80,
27               decoration: const BoxDecoration(
28                 color: Colors.blue,
29                 shape: BoxShape.circle,
30                 image: DecorationImage(
31                   fit: BoxFit.cover,
32                   image: NetworkImage(
33                     'https://cdn.pixabay.com/photo/2014/07/09/10/04/man-388104_960_720.jpg'
34                   ), // NetworkImage
35                 ), // DecorationImage
36               ), // BoxDecoration
37             ), // Container // Positioned
38           ], // Stack
39         ), // Center
40       ); // Scaffold
41     }
42   }
43 }
```

Debug my code Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 3.0.6-0.0.pre.1 iPhone 11 (ios simulator)

i. FloatingActionBotton & BottonNavigationBar



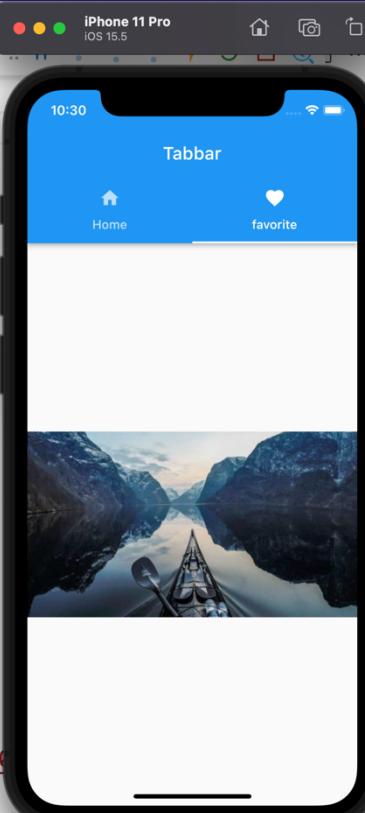
The screenshot shows an iPhone 11 running iOS 15.5. The app's title bar says "Buttons". The main screen contains several buttons: an "ElevatedButton", a "TextButton", a "CupertinoButton" with a camera icon, and a "MaterialButton". A floating action button (FAB) with a plus sign is located in the bottom right corner. At the bottom of the screen is a "BottomNavigationBar" with two items: "home" and "favorite".

```
button_screen.dart — app1
main.dart button_screen.dart ×

lib > lib > button_screen.dart > TestButtonScreen > build
1 import 'package:flutter/cupertino.dart';
2 import 'package:flutter/material.dart';
3
4 class TestButtonScreen extends StatelessWidget {
5   const TestButtonScreen({Key? key}) : super(key: key);
6
7   @override
8   Widget build(BuildContext context) {
9     return Scaffold(
10       appBar: AppBar(
11         title: const Text('Buttons'),
12       ), // AppBar
13       body: Center(
14         child: Column(
15           mainAxisAlignment: MainAxisAlignment.center,
16           children: [
17             ElevatedButton(
18               onPressed: () {}, child: const Text('ElevatedButton')),
19             TextButton(
20               onPressed: () {}, child: const Text(
21                 'TextButton',
22               )), // Text // TextButton
23             IconButton(onPressed: () {}, icon: const Icon(Icons.camera_alt)),
24             CupertinoButton(
25               color: Colors.blue,
26               child: const Text('CupertinoButton'),
27               onPressed: () {}, // CupertinoButton
28             MaterialButton(
29               color: Colors.blueAccent,
30               onPressed: () {},
31               child: const Text('MaterialButton'),
32             ) // MaterialButton
33           ],
34         ),
35       ), // Column
36     ), // Center
37     floatingActionButton: FloatingActionButton(
38       onPressed: () {},
39       tooltip: 'Increment',
40       child: const Icon(Icons.add),
41     ), // FloatingActionButton
42     bottomNavigationBar: BottomNavigationBar(items: const [
43       BottomNavigationBarItem(icon: Icon(Icons.home), label: 'home'),
44       BottomNavigationBarItem(icon: Icon(Icons.favorite), label: 'favorite')
45     ]), // BottomNavigationBar
46   ); // Scaffold
47 }
48 }

0 ① 6 Debug my code Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 3.0.6-0.0.pre.1 iPhone 11 (ios simulator) 13
```

ii. Tabbar & TabbarView



The screenshot shows a Flutter application running on an iPhone 11 Pro simulator. The app has a blue header bar with the title 'Tabbar'. Below it is a TabBar with two tabs: 'Home' (selected) and 'favorite'. The main content area displays a landscape image of a fjord with mountains and water. The code in the editor corresponds to the 'tabbars.dart' file, which creates a TabBarView with two Image.network widgets.

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

class TabBarVeiws extends StatelessWidget {
    const TabBarVeiws({Key? key}) : super(key: key);

    @override
    Widget build(BuildContext context) {
        return DefaultTabController(
            length: 2,
            child: Scaffold(
                appBar: AppBar(
                    title: const Text('Tabbar'),
                    bottom: const TabBar(tabs: [
                        Tab(icon: Icon(Icons.home), text: 'Home'),
                        Tab(icon: Icon(Icons.favorite), text: 'favorite')
                    ]),
                ),
                body: TabBarView(children: [
                    Image.network(
                        'https://static9.depositphotos.com/1719616/1205/i/600/stock-photo-boat-in-the-fjord-norway.jpg',
                    ),
                    Image.network(
                        'https://www.ilmubahasainggris.com/wp-content/uploads/2017/03/NGC.jpg',
                    )
                ]),
            ),
        );
    }
}
```

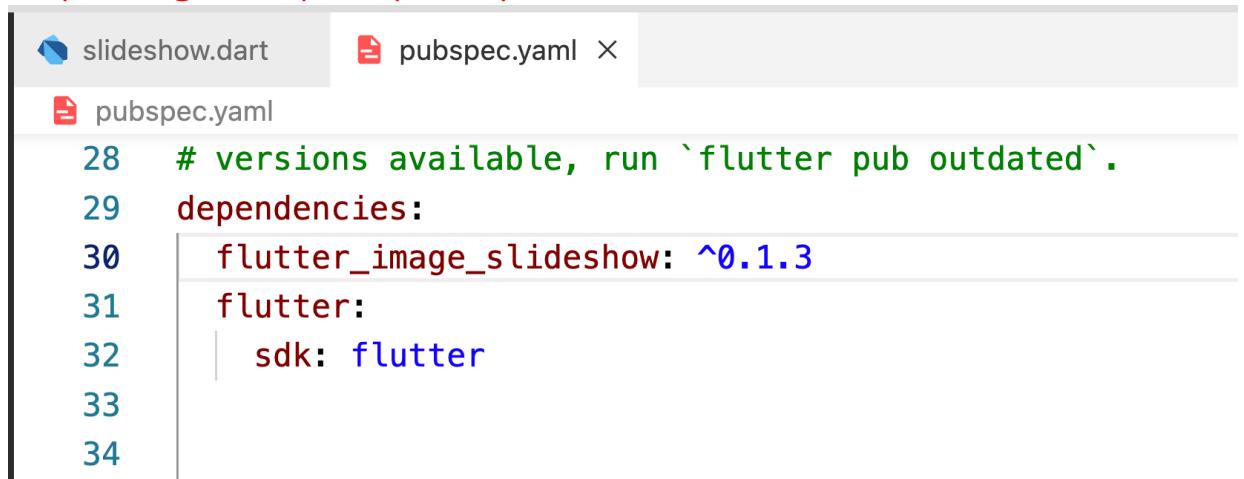
iii. Widget with Packet

Website ដែលអាចទាញយក package ជាប្រើប្រាស់ Dart

<https://pub.dev/>

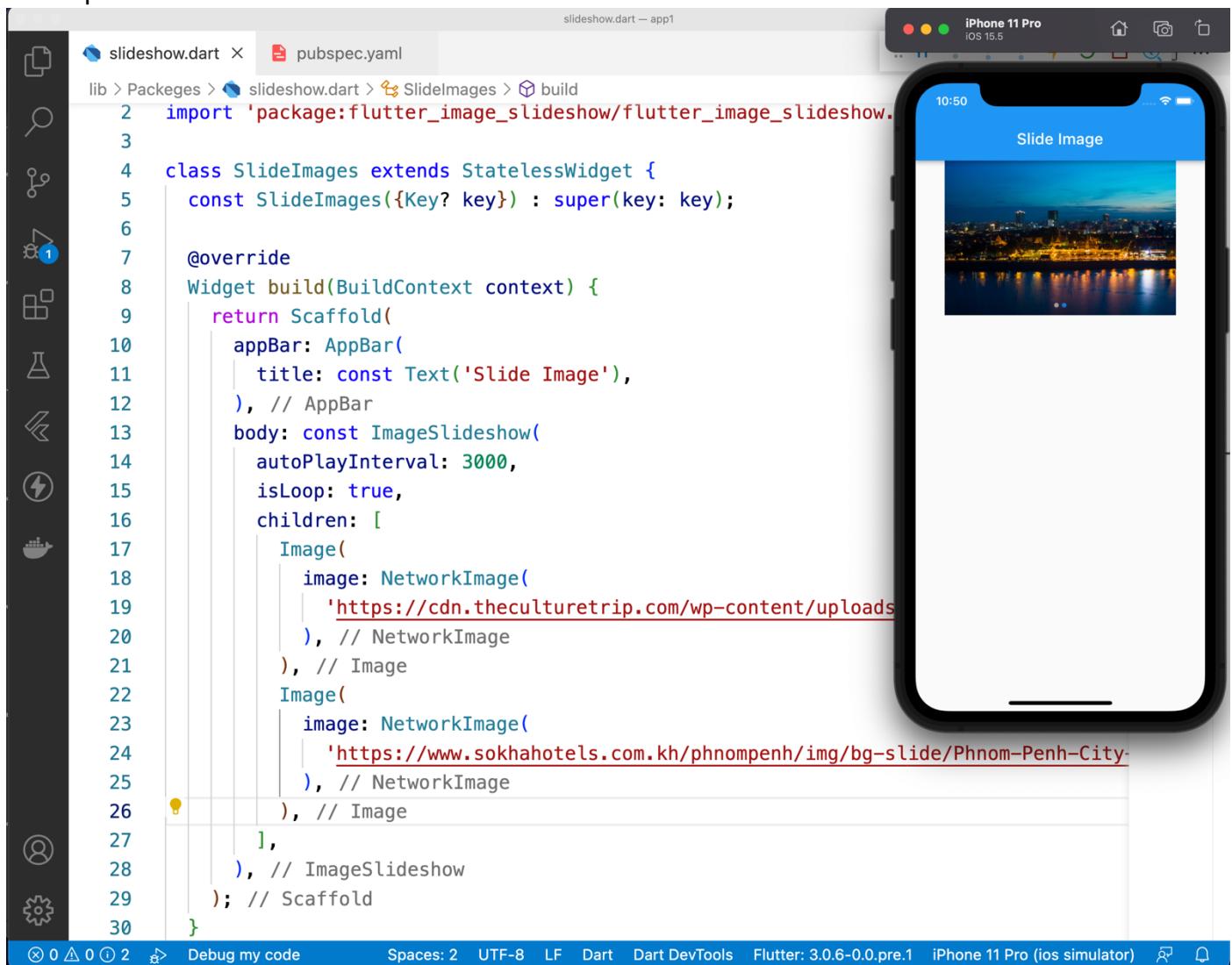
1. Image slideshow

Add packages to pubspec.yaml



```
slideshow.dart pubspec.yaml
pubspec.yaml
28 # versions available, run `flutter pub outdated`.
29 dependencies:
30   flutter_image_slideshow: ^0.1.3
31   flutter:
32     sdk: flutter
33
34
```

Example:



The screenshot shows the Android Studio interface with the following details:

- Project Structure:** The left sidebar shows the project structure with files like slideshow.dart and pubspec.yaml.
- Code Editor:** The main editor shows the code for a StatelessWidget named SlideImages. It imports flutter_image_slideshow and uses ImageSlideshow to display two images from URLs: <https://cdn.theculturetrip.com/wp-content/uploads> and <https://www.sokahotels.com.kh/phnompenh/img/bg-slide/Phnom-Penh-City->.
- Run Tab:** The top right shows the device configuration: iPhone 11 Pro, iOS 15.5.
- Preview:** On the right, there's a preview of the app running on the iPhone 11 Pro simulator, titled "Slide Image". It displays a night scene of a city skyline reflected in water.
- Bottom Status Bar:** The status bar at the bottom shows: 10:50, Slide Image, and various developer tools like Spaces: 2, UTF-8, LF, Dart, Dart DevTools, Flutter: 3.0.6-0.0.pre.1, iPhone 11 Pro (ios simulator), and a few icons.

2. Open camera and Gallary

Add packages

image_picker: ^0.8.5+3

Andriod config

ទីតាំង Adroid>app>src>main>AndroidManifest.xml

```
        android:requestLegacyExternalStorage="true"
```

IOS config

ទីតាំង ios>Runner>Info.plist

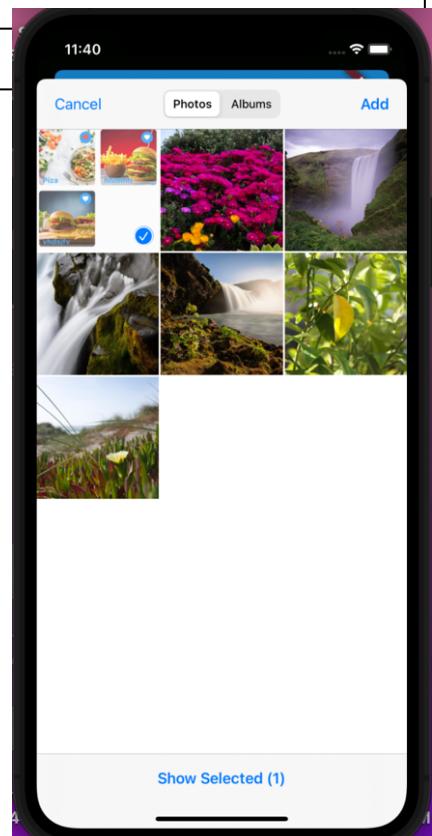
```
<key>UISupportedInterfaceOrientations</key>
    <key>NSCameraUsageDescription</key>
        <string> To choose user profile image</string>
    <key>NSMicrophoneUsageDescription</key>
        <string>Use to capture audio for image picker plugin</string>
    <key>NSPhotoLibraryUsageDescription</key>
        <string> To take user profile image</string>
```

Code:

```
Future<void> getImagefromCamera() async {
    final image = await ImagePicker().pickImage(
        source: ImageSource.camera, imageQuality: 100);

    setState(() {
        _image = File(image!.path);
    });
}

Future<void> getImagefromaGallary() async {
    final image = await
ImagePicker().pickImage(source:ImageSource.gallery);
    setState(() {
        _image = File(image!.path);
    });
}
```



3. Google Map

1 .Add Package

```

dependencies:
  google_maps_flutter: ^2.1.10
  flutter:
    sdk: flutter

```

2. Config android phone

minSDK

```

android {
  defaultConfig {
    minSdkVersion 20
  }
}

```

Link API key

[Google Developers Console](#)

```

<manifest ...
  <application ...
    <meta-data android:name="com.google.android.geo.API_KEY"
      android:value="YOUR KEY HERE"/>

```

3 .IOS

```

import UIKit
import Flutter
import GoogleMaps

@UIApplicationMain
@objc class AppDelegate: FlutterAppDelegate {
  override func application(
    _ application: UIApplication,
    didFinishLaunchingWithOptions launchOptions: [UIApplication.LaunchOptionsKey: Any]?
  ) -> Bool {
    GMSServices.provideAPIKey("YOUR KEY HERE")
    GeneratedPluginRegistrant.register(with: self)
    return super.application(application, didFinishLaunchingWithOptions: launchOptions)
  }
}

```

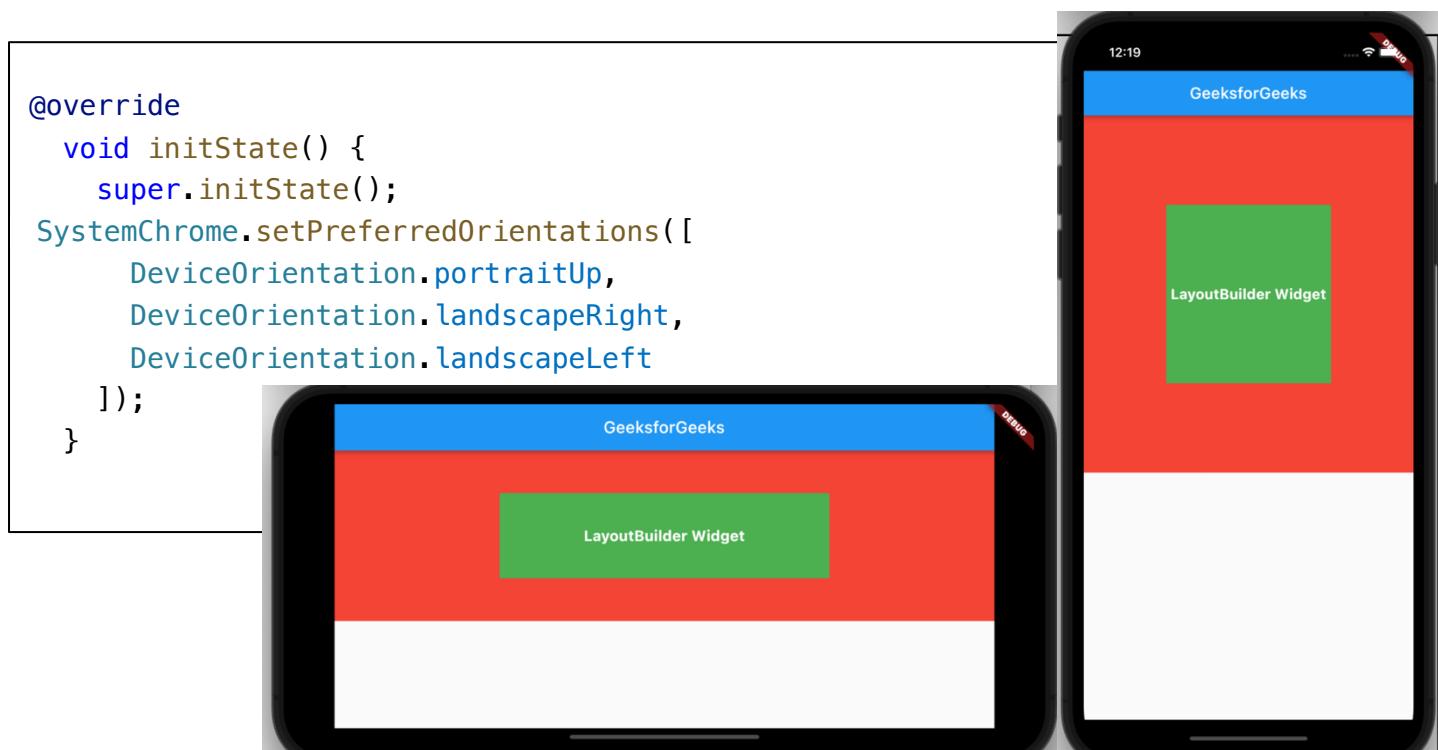
4 Example

The screenshot shows the Android Studio interface. On the left is the code editor with `main.dart` and `AndroidManifest.xml` open. The code defines a `MyHomePage` stateful widget and its corresponding state class `_MyHomePageState`. It initializes a `GoogleMap` with a hybrid map type, an initial camera position targeting Googleplex, and an `onMapCreated` callback to complete the controller. On the right, an iPhone 11 Pro simulator displays a satellite map of the Googleplex area.

```
main.dart — flutter_google_map
lib > main.dart > _MyHomePageState > _kGooglePlex
25 class MyHomePage extends StatefulWidget {
26   const MyHomePage({
27     Key? key,
28   ) : super(key: key);
29
30   @override
31   State<MyHomePage> createState() => _MyHomePageState();
32 }
33
34 class _MyHomePageState extends State<MyHomePage> {
35   final Completer<GoogleMapController> _controller = Completer();
36   final CameraPosition _kGooglePlex = const CameraPosition(
37     target: LatLng(37.42796133580664, -122.085749655962),
38     zoom: 14.4746,
39   ); // CameraPosition
40
41   @override
42   Widget build(BuildContext context) {
43     return Scaffold(
44       body: GoogleMap(
45         mapType: MapType.hybrid,
46         initialCameraPosition: _kGooglePlex,
47         onMapCreated: (GoogleMapController controller) {
48           _controller.complete(controller);
49         },
50       ), // GoogleMap
51     ); // Scaffold
52   }
53 }
```

Spaces: 2 UTF-8 LF Dart Dart DevTools Flutter: 3.0.6-0.0.pre.1 iPhone 11 Pro (ios simulator)

IV. Responsive Widget



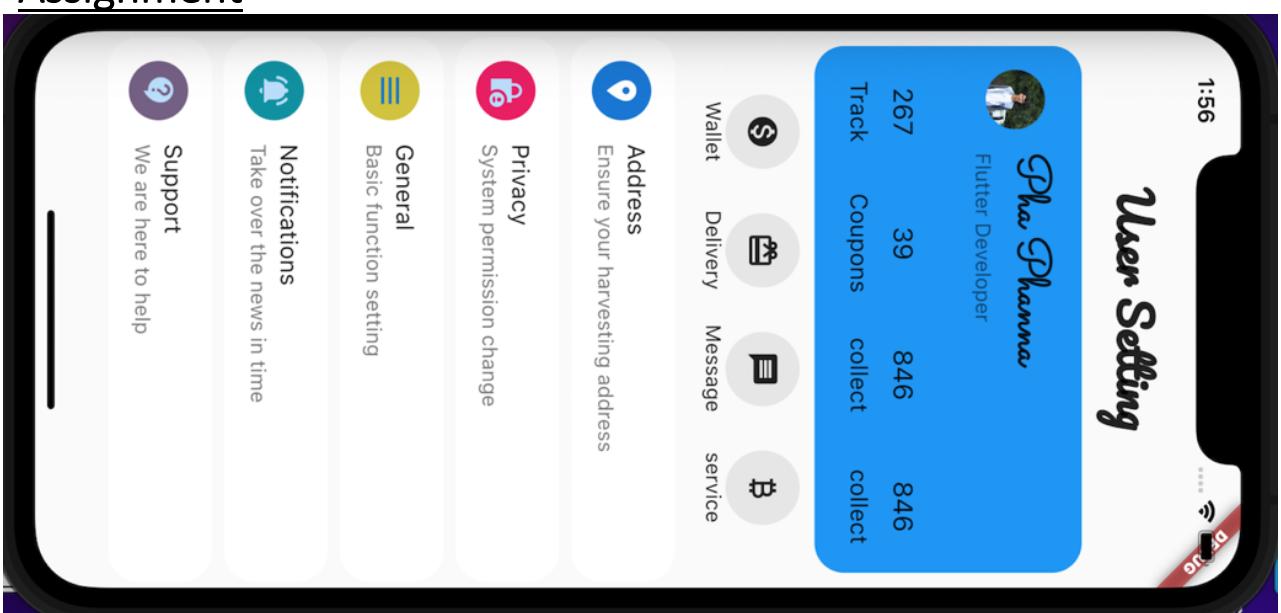
```
main.dart — resonesize7_8
main.dart > _MyHomePageState
46
47   @override
48   Widget build(BuildContext context) {
49     double sizeScreen = MediaQuery.of(context).size.width; The value of the local variable 'sizeScreen' is 360.0
50
51     return SafeArea(
52       child: Scaffold(
53         appBar: AppBar(title: const Text('GeeksforGeeks')),
54         body: Container(
55           color: Colors.red,
56           height: MediaQuery.of(context).size.height * 0.5,
57           width: MediaQuery.of(context).size.width,
58           alignment: Alignment.center,
59           child: LayoutBuilder(
60             builder: (BuildContext ctx, BoxConstraints constraints) {
61               return Container(
62                 color: Colors.green,
63                 alignment: Alignment.center,
64                 height: constraints.maxHeight * 0.5,
65                 width: constraints maxWidth * 0.5,
66                 child: const Text('LayoutBuilder Widget',
67                   style: TextStyle(
68                     fontSize: 18,
69                     fontWeight: FontWeight.bold,
70                     color: Colors.white,
71                   ), // TextStyle // Text
72                 ); // Container
73             }, // LayoutBuilder
74           ), // Container
75         ), // Scaffold
76       ); // SafeArea
77     }
  
```

Example2:

The screenshot shows a Flutter development setup. On the left is the code editor with a Dart file named `gridviews.dart`. The code implements a `GridView` with 12 items arranged in a 3x4 grid. The items are labeled 1 through 6 and 2 through 6 respectively. The code uses `MediaQuery.of(context).orientation` to determine scroll direction and `list.generate` to create the children. On the right, there are two iPhone simulators. The top one shows a 3x4 grid with items 1, 4, 2, 5 in the first row; 6, 7, 8, 3 in the second; 4, 2, 3, 5 in the third; and 3, 64, 232 in the fourth. The bottom one shows a 4x5 grid with items 2, 6, 2, 7, 3 in the first row; 5, 3, 6, 8, 4 in the second; and 6, 2, 6, 3, 4 in the third and fourth rows.

```
gridviews.dart — resonesize7_8
main.dart gridviews.dart screensize.dart
lib > gridviews.dart > _DataGridViewsState > build
23
24     List list = [1, 4, 2, 5, 6, 3, 2, 6, 7, 8, 3, 4, 2, 3, 5, 3, 64, 232];
25     @override
26     Widget build(BuildContext context) {
27         return Scaffold(
28             appBar: AppBar(),
29             body: GridView.count(
30                 scrollDirection:
31                     MediaQuery.of(context).orientation == Orientation.portrait
32                         ? Axis.vertical
33                         : Axis.horizontal,
34                 crossAxisCount: 2,
35                 mainAxisSpacing: 3,
36                 crossAxisSpacing: 3,
37                 children: List.generate(
38                     list.length,
39                     (index) => Container(
40                         height: 100,
41                         width: 100,
42                         color: Colors.blueAccent,
43                         child: Center(
44                             child: Text(
45                                 list[index].toString(),
46                                 style: TextStyle(fontSize: 30),
47                             ), // Text
48                         ), // Center
49                     ), // Container
50                 ), // List.generate
51             ), // GridView.count
52         ); // Scaffold
53     }
54 
```

V. Assignment





Thank You!!