



Day 2

Flutter Development

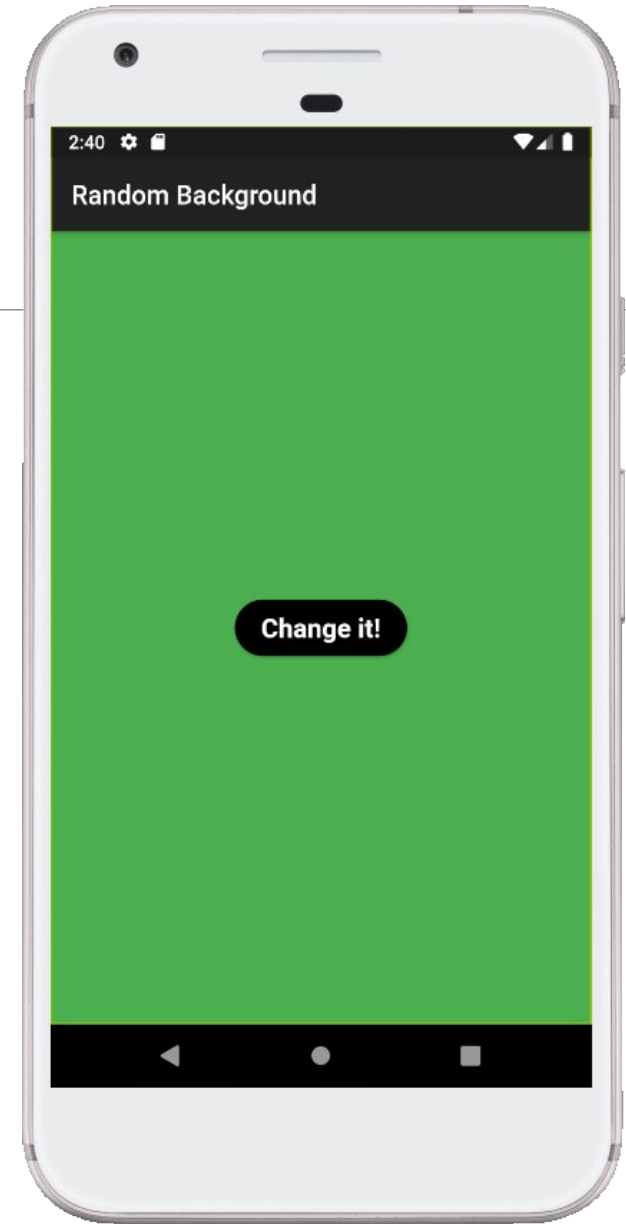
MARLON I. TAYAG





Background Changer App

1. Create an app that will randomly change its background color when pressing a button
2. User MaterialApp and Dart:Math library





1. Import Library and main()

```
1  import 'package:flutter/material.dart';
2  import 'dart:math';
3
4
5  void main() {
6    |   runApp(myApp());
7  }
```



2. Create **myApp** stateless widget

```
9  class myApp extends StatelessWidget {
10    const myApp({Key key}) : super(key: key);
11
12    @override
13    Widget build(BuildContext context) {
14      return MaterialApp(
15        title: 'BG Changer',
16        debugShowCheckedModeBanner: false,
17        theme: ThemeData.dark(),
18        home: Scaffold(
19          appBar: AppBar(
20            title: Text('Random Background'),
21          ), // AppBar
22          body: HomePage(),
23        ), // Scaffold
24      ); // MaterialApp
25    }
26  }
```



3. Create **HomePage** stateful widget

```
28   class HomePage extends StatefulWidget {  
29     HomePage({Key key}) : super(key: key);  
30     _HomePageState createState() => _HomePageState();  
31   }  
32
```



4. Create the `_HomePageState` state widget

```
28 class HomePage extends StatefulWidget {
29   HomePage({Key key}) : super(key: key);
30   _HomePageState createState() => _HomePageState();
31 }
32
33 class _HomePageState extends State<HomePage> {
34   var colors = [
35     Colors.black,
36     Colors.blue,
37     Colors.green,
38     Colors.red,
39     Colors.pink,
40     Colors.orange,
41     Colors.amber
42
43   ];
44
45   var currentColor = Colors.white;
46
47   void setRandomColor () {
48     var rnd = Random().nextInt(colors.length);
49     setState(() {
50       currentColor = colors[rnd];
51     });
52   });
53
54 }
```



4. Create the _HomePageState state widget (cont'd ..)

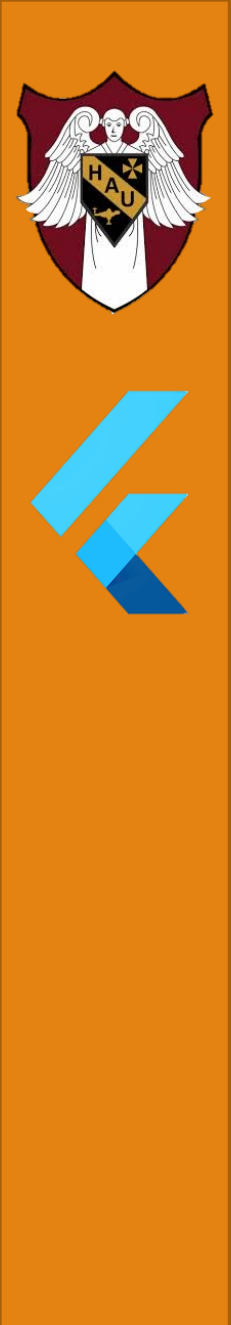
```
55
56
57 @override
58 Widget build(BuildContext context) {
59   return Container(
60
61     color: currentColor,
62     child: Center(
63       child: RaisedButton(
64         color: Colors.black,
65         padding: EdgeInsets.fromLTRB(20.0, 10.0, 20.0, 10.0),
66         child: Text('Change it!', style: TextStyle(fontSize: 20, fontWeight: FontWeight.bold)
67
68           ), // Text
69         onPressed: setRandomColor,
70         shape: RoundedRectangleBorder(
71           borderRadius: BorderRadius.circular(30.0)
72         ), // RoundedRectangleBorder
73       ), // RaisedButton
74     ), // Center
75   ); // Container
76 }
77 }
```



Activity Challenge

1. Automatically change the color of the text based on the background color whenever the user click the button



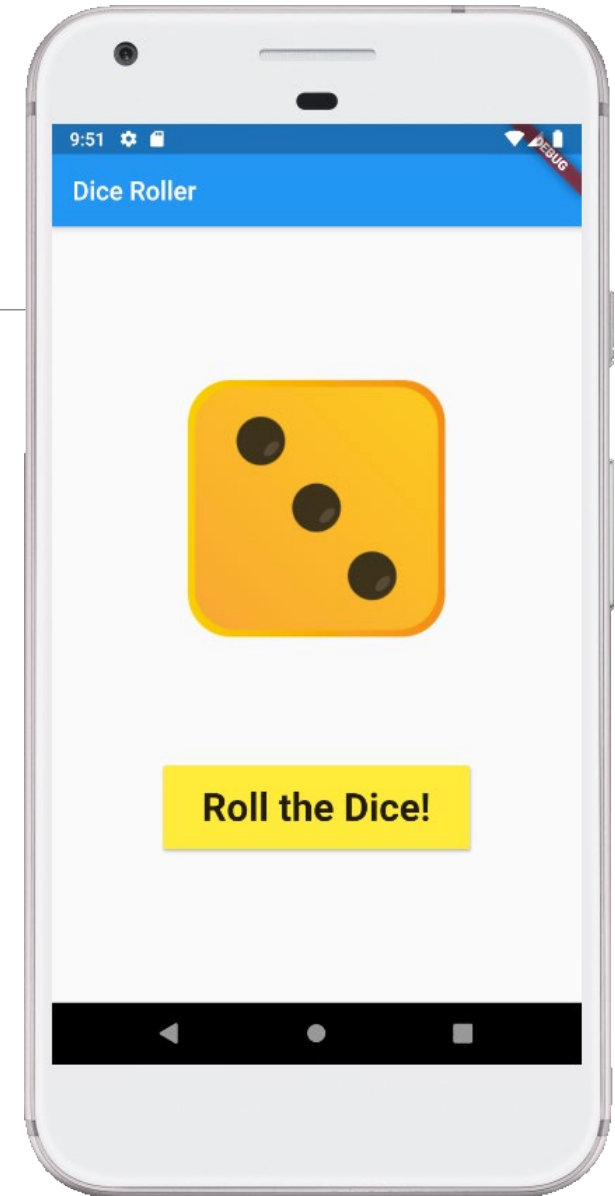


Background Changer Code



Dice Roller App

1. Load image to **image** folder on the project
2. Edit **pubspec.yaml** and add images to assets
3. Create the logic part of the Dice Roller (**homepage.dart**)
4. Create the design and layout of Dice Roller (**homepage.dart**)





1. Add images

1. Create a folder on the project name image, add all dice image on the folder

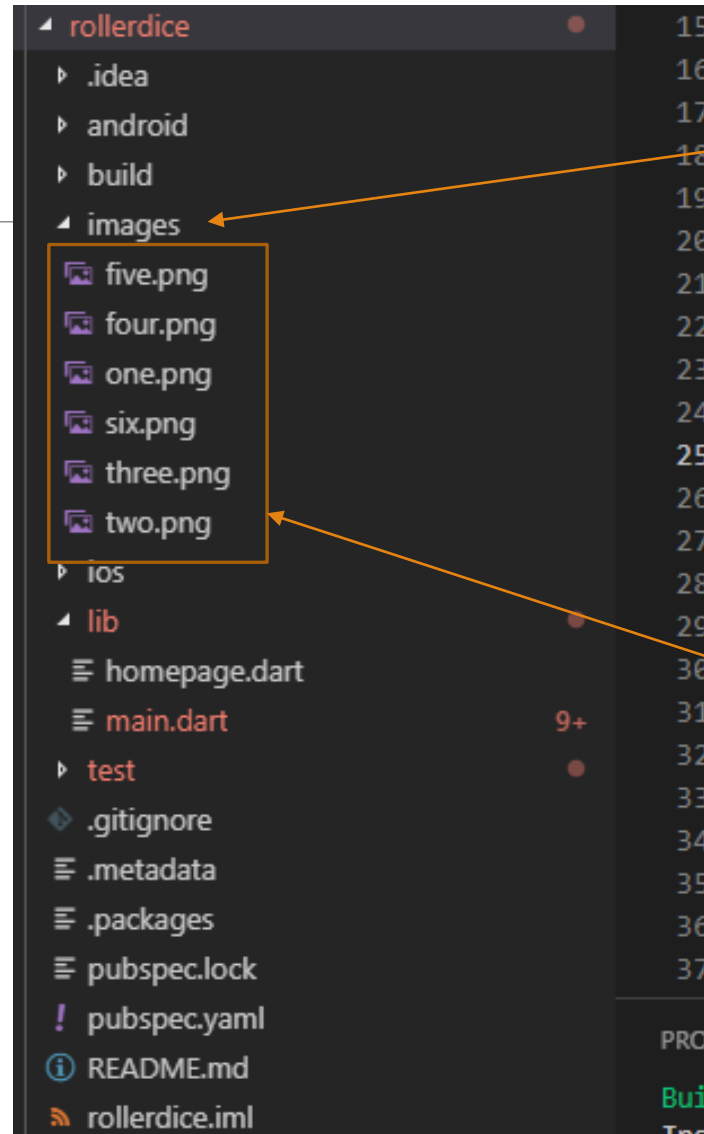


Image folder

Images



2. Edit `pubspec.yaml`

1. Edit `pubspec.yaml` and add the list of images on the `assets`

```
36 flutter:
37
38   # The following line ensures that the Material Icons font is
39   # included with your application, so that you can use the icons in
40   # the material Icons class.
41   uses-material-design: true
42
43   # To add assets to your application, add an assets section, like this:
44   assets:
45     - images/one.png
46     - images/two.png
47     - images/three.png
48     - images/four.png
49     - images/five.png
50     - images/six.png
```

Note: Be careful of the indentation



3. Import library and create `main()` , `diceApp` stateless widget (`main.dart`)

```
1  import 'package:flutter/material.dart';
2  import 'homepage.dart';
3
4
5
6  void main() => runApp(diceApp());
7
8  class diceApp extends StatelessWidget {
9    const diceApp({Key key}) : super(key: key);
10
11    @override
12    Widget build(BuildContext context) {
13      return MaterialApp(
14        title: 'Dice Rooler',
15        theme: ThemeData(primarySwatch: Colors.blue),
16        home: HomePage(),
17      );
18    }
19  }
20
21 }
```



4. Create the **HomePage** stateful widget (`homepage.dart`)



1. Import the following library
 1. `material.dart`
 2. `dart:math`
2. Create the stateful widget

```
1  import 'package:flutter/foundation.dart';
2  import 'package:flutter/material.dart';
3  import 'dart:math';
4
5  class HomePage extends StatefulWidget {
6    HomePage({Key key}) : super(key: key);
7
8    _HomePageState createState() => _HomePageState();
9  }
```



5. Create the _HomePage state (homepage.dart)

```
11 class _HomePageState extends State<HomePage> {  
12  
13     AssetImage one = AssetImage("images/one.png");  
14     AssetImage two = AssetImage("images/two.png");  
15     AssetImage three = AssetImage("images/three.png");  
16     AssetImage four = AssetImage("images/four.png");  
17     AssetImage five = AssetImage("images/five.png");  
18     AssetImage six = AssetImage("images/six.png");  
19  
20     AssetImage diceImage;  
21  
22     @override  
23     void initState() {  
24         super.initState();  
25         setState() {  
26             diceImage = one;  
27         }  
28     };  
29 }  
30
```



6. Create the _HomePage state (homepage.dart) cont'd ...

```
31
32   void rollerDice() {
33     var random = (1 + Random().nextInt(6));
34     AssetImage newImage;
35     switch (random) {
36       case 1:
37         newImage = one;
38         break;
39       case 2:
40         newImage = two;
41         break;
42       case 3:
43         newImage = three;
44         break;
45       case 4:
46         newImage = four;
47         break;
48       case 5:
49         newImage = five;
50         break;
51       case 6:
52         newImage = six;
53         break;
54     }
55
56     setState(() {
57       diceImage=newImage;
58     });
59   }
60 }
```



```

61 @override
62 Widget build(BuildContext context) {
63   return Scaffold(
64     appBar: AppBar(
65       title: Text('Dice Roller'),
66     ),
67     body: Container(
68       child: Center(
69         child: Column(
70           mainAxisAlignment: MainAxisAlignment.center,
71           children: <Widget>[
72             Image(
73               image: diceImage,
74               width: 200.0,
75               height: 200.0,
76             ),
77             Container(
78               margin: EdgeInsets.only(top: 100.0),
79               child: RaisedButton(
80                 color: Colors.yellow,
81                 padding: EdgeInsets.fromLTRB(30.0, 15.0, 30.0, 15.0),
82                 child: Text('Roll the Dice!', style: TextStyle(fontWeight: FontWeight.bold, fontSize: 30.0)),
83                 onPressed: rollerDice,
84               ),
85             ),
86           ],
87         ),
88       ),
89     ),
90   );
91 }
92
93
94
95
96

```

7. Create the
HomePage state
 (homepage.dart)
 cont'd ...



Activity Challenge

1. Modify the roller dice app, adding another dice when the button roll dice is clicked.
2. Show the total value of the two dice in a text
3. Change the launcher icon of the Dice Roller App





Dice Roller Code

[main.dart](#)

[homepage.dart](#)



Using Android Studio for Flutter Dev





LoadingImage App

1. Create from an that will load an image from an Asset folder
2. Use a placeholder image while the image is being loaded from the internet
3. Use Material Library
4. This is a widget used to show an image. When displaying an image, you specify the image source in the constructor:
 - a. image provider
 - b. asset,
 - c. network,
 - d. file,
 - e. memory





1. Get Loading Image

1. Download: <https://digitalsynopsis.com/wp-content/uploads/2016/06/loading-animations-preloader-gifs-ui-ux-effects-10.gif>
2. Create new folder 'assets' in your project.
3. Rename image file to 'loading.gif'.
4. Copy image file into 'assets' folder in your project.



2. Include the Loading Image in Your Project as an Asset



1. Edit the `pubspec.yaml` file and change the lines below from:

```
# To add assets to your application, add an assets section, like  
this: # assets:  
# - images/a_dot_burr.jpeg  
# - images/a_dot_ham.jpeg
```

To

```
assets:  
- assets/loading.gif
```



3. Import Library and create **main()**

```
1  import 'package:flutter/material.dart';  
2  
3  
4  void main() => runApp(LoadingImageApp());  
5
```




4. Create the **LoadingImageApp** stateless widget



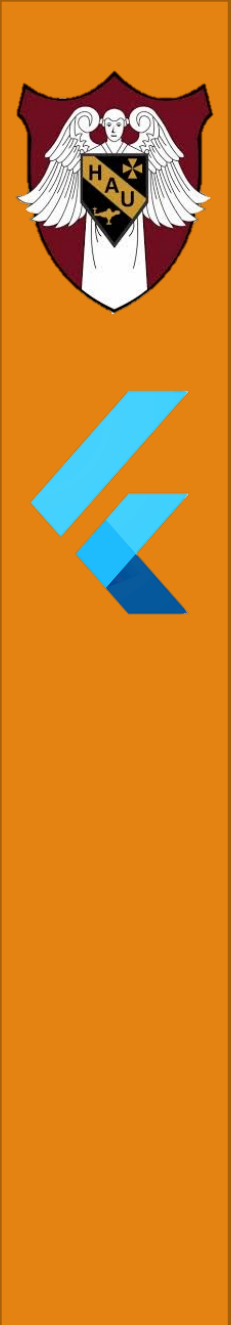
```
6  class LoadingImageApp extends StatelessWidget {  
7  
8      @override  
9  
10     Widget build(BuildContext context) {  
11         // TODO: implement build  
12  
13         return MaterialApp(  
14             title: 'Image',  
15             debugShowCheckedModeBanner: false,  
16             theme: ThemeData(  
17                 primarySwatch: Colors.blue,  
18             ), // ThemeData  
19             home: HomeWidget(),  
20  
21         ); // MaterialApp  
22     }  
23 }
```



4. Create the **HomeWidget** stateless widget

```
class HomeWidget extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    // TODO: implement build  
  
    return Scaffold (  
      appBar: AppBar(  
        title: Text('Image'),  
      ), // AppBar  
      body: Center(  
        child: FadeInImage.assetNetwork(placeholder: 'assets/loadingimg.gif', image: 'http://archivision.com/educational/samples/files/1A2-F-P-I-2-C1_L.jpg')  
      ), // Center  
    ); // Scaffold  
  }  
}
```

Link for the image: http://archivision.com/educational/samples/files/1A2-F-P-I-2-C1_L.jpg



LoadingImage Code

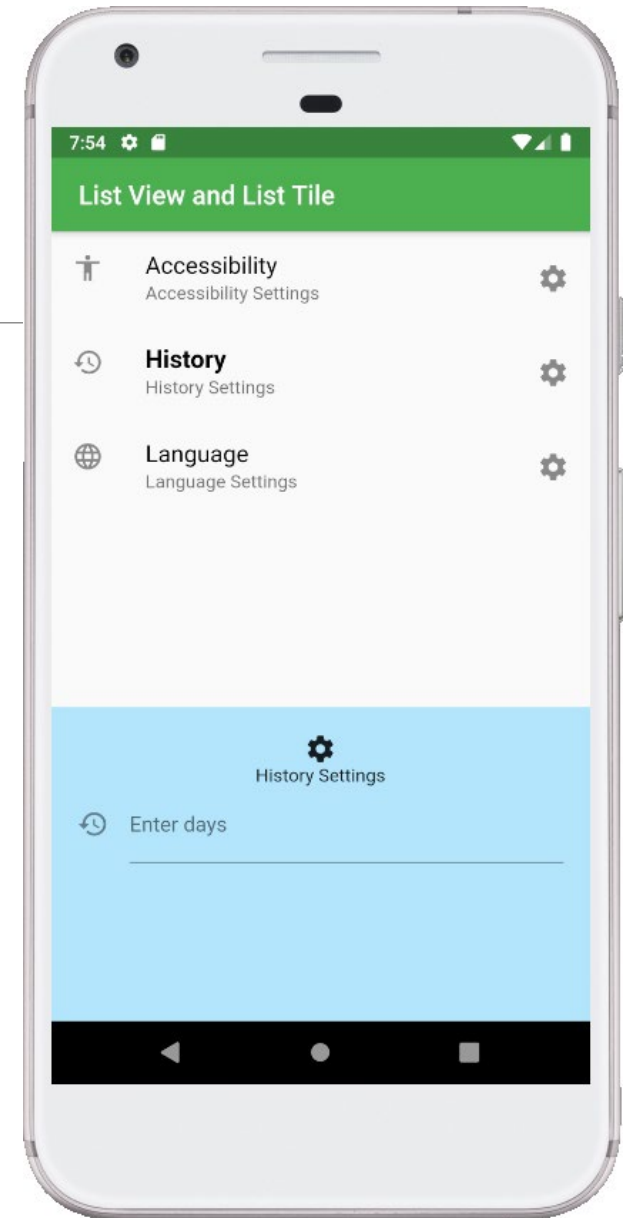


Working With ListView and ListTile



A list tile contains one to three lines of text optionally flanked by icons or other widgets, such as check boxes. So, you can have text in the middle and a widget on each side.

Many people combine ListViews and ListTiles together because ListTiles are great for building great-looking selection lists.





1. Import and create **main()**

```
1  import 'package:flutter/material.dart';  
2  
3  void main() =>runApp(ListViewListTileApp());
```



2. Create `ListViewListTileApp` stateless widget



```
5 class ListViewListTileApp extends StatelessWidget {  
6   const ListViewListTileApp({Key key}) : super(key: key);  
7  
8   @override  
9   Widget build(BuildContext context) {  
10    return MaterialApp(  
11      debugShowCheckedModeBanner: false,  
12      title: 'List View and List Tile',  
13      theme: ThemeData(  
14        primarySwatch: Colors.green,  
15      ), // ThemeData  
16      home: HomeWidget(title: 'List View and List Tile'),  
17    ); // MaterialApp  
18  }  
19 }
```



3. Create the **HomeWidget** stateful widgets

```
21 class HomeWidget extends StatefulWidget {  
22   HomeWidget({Key key, this.title}) : super(key: key);  
23   final String title;  
24  
25   _HomeWidgetState createState() => _HomeWidgetState();  
26 }  
27
```

4. Create `_HomeWidgetState` state widget

```
28 class _HomeWidgetState extends State<HomeWidget> {
29
30   int selectedIndex = 0;
31   static const TEXT_STYLE_NORMAL = const TextStyle(color: Colors.black, fontSize: 18.0, fontWeight: FontWeight.normal);
32   static const TEXT_STYLE_SELECTED = const TextStyle(color: Colors.black, fontSize: 19.0, fontWeight: FontWeight.bold);
33
34   final TextFormField _fontSizeTextField = TextFormField(
35     |   |   |   decoration: InputDecoration(icon: const Icon(Icons.format_size),
36     |   |   |   hintText: 'Font Size', labelText: 'Enter Font Size'), // InputDecoration
37   |   |   |
38   ); // TextFormField
39
40   final TextFormField _historyTextFormField = TextFormField(
41     |   |   |   decoration: InputDecoration (icon: const Icon(Icons.history),
42     |   |   |   hintText: 'Days', labelText: 'Enter days'), // InputDecoration
43     |   |   |   keyboardType: TextInputType.number,
44   ); // TextFormField
45
46   final TextFormField _languageTextFormField = TextFormField(
47     |   |   |   decoration: InputDecoration (icon: const Icon(Icons.language),
48     |   |   |   hintText: 'Language', labelText: 'Enter your language'), // InputDecoration
49   ); // TextFormField
50
51   select(index) {
52     |   |   |   setState(() {
53     |   |   |   |   selectedIndex = index;
54     |   |   |   |   });
55   }
```

Constant
variable for text
color

TextFormField
for selected list

setState



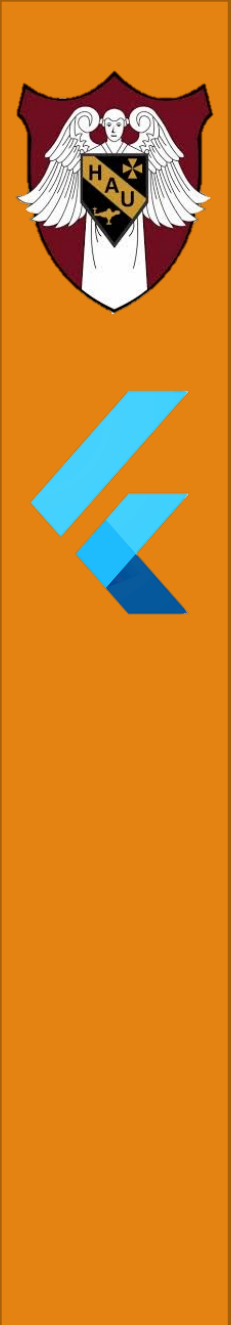
HomeWidgetState state widget (cont'd)

```
57 @override
58 Widget build(BuildContext context) {
59
60   final ListTile accessibilityListTile = ListTile (
61     leading: Icon(Icons.accessibility), title: Text('Accessibility',
62     style: selectedIndex == 0 ? TEXT_STYLE_SELECTED : TEXT_STYLE_NORMAL,
63     ), subtitle: const Text ('Accessibility Settings'), trailing: Icon(Icons.settings), onTap: () => select(0), // Text
64   ); // ListTile
65
66   final ListTile historyListTile = ListTile (
67     leading: Icon(Icons.history), title: Text('History',
68     style: selectedIndex == 1 ? TEXT_STYLE_SELECTED : TEXT_STYLE_NORMAL,
69     ), subtitle: const Text ('History Settings'), trailing: Icon(Icons.settings), onTap: () => select(1), // Text
70   ); // ListTile
71
72
73   final ListTile languageListTile = ListTile (
74     leading: Icon(Icons.language), title: Text('Language',
75     style: selectedIndex == 2 ? TEXT_STYLE_SELECTED : TEXT_STYLE_NORMAL,
76     ), subtitle: const Text ('Language Settings'), trailing: Icon(Icons.settings), onTap: () => select(2), // Text
77   ); // ListTile
78
79
80   final String selectionTitle = (selectedIndex == 0 ? 'Accessibility': selectedIndex == 1 ? "History" : "Language") + " Settings";
81   final TextFormField selectionTextFormField = selectedIndex == 0 ? _fontSizeTextFormField : selectedIndex == 1 ? _historyTextFormField : _languageTextFormField;
```



HomeWidgetState state widget (cont'd)

```
82
83     return Scaffold(
84
85       appBar: AppBar(
86         title: Text(widget.title),
87       ), // AppBar
88       body: ListView(
89         children: <Widget>[
90           accessibilityListTile,
91           historyListTile,
92           languageListTile
93         ], // <Widget>[]
94
95       ), // ListView
96       bottomSheet: Container(
97         color: Color(0xFFB3E5FC),
98         padding: EdgeInsets.all(20.0),
99         child: Container(
100           constraints: BoxConstraints(maxHeight: 200.0),
101           child: Column(
102             children: <Widget>[
103               Icon(Icons.settings),
104               Text(selectionTitle),
105               Expanded(
106                 child: selectionTextFormField,
107               ) // Expanded
108             ], // <Widget>[]
109           ), // Column
110         ), // Container
111       ), // Container
112     ); // Scaffold
113   }
114 }
```



ListViewListTile Code



Workshop Challenge

Create an app that computes
Dog age in Human years

<https://pets.webmd.com/dogs/how-to-calculate-your-dogs-age>

Size of Dog	Small Miniature Pinscher 20 lbs. or less	Medium Schnauzer 21-50 lbs.	Large Great Dane More than 50 lbs.
Age of Dog	Age in Human Years		
1 Year	15	15	15
2	24	24	24
3	28	28	28
4	32	32	32
5	36	36	36
6	40	42	45
7	44	47	50
8	48	51	55
9	52	56	61
10	56	60	66
11	60	65	72
12	64	69	77
13	68	74	82
14	72	78	88
15	76	83	93
16	80	87	120

SOURCES:
Purina: "Your Dog's Age in Human Years" and "Caring For Your Older Dog."
National Pet Wellness Month: "Pet Age Calculator" and "Pet Aging Chart!"
Humane Society: "How to Determine a Cat's or Dog's Age."
©2015 WebMD, LLC. All rights reserved.



Building Release APK



Inside the app folder type

- flutter clean, then
- flutter build apk

The release APK for your app is created at `<app dir>/build/app/outputs/apk/release/`



Thank You Very Much!!!