Internet of Things

Lab1

Introduction to Flutter

Grading scheme

- Final examination: 40%
- Lab – 30% (average of lab activity, hw) – min. grade 5 required
- Project – 30% - min. grade 5 required

Grades: https://docs.google.com/spreadsheets/d/1h-MISNrNQCNIUIkFUzWtNb9QOZrV-VnwZpx6nLksw8o/edit#gid=0

If you have any issues:

ciungan_alexandra@yahoo.ro

vlad.garmacea@gmail.com

Flutter

- Open-source, portable UI toolkit, created by Google
- Frontend development framework
- Used to develop multi-platform applications from a single codebase
  - Android
  - iOS
  - Linux
  - macOS
  - Windows
  - Google Fuchsia
- Flutter apps use Dart programming language

Installation

- For the installation, please, follow the steps inside this tutorial: https://docs.flutter.dev/get-started/install

- You will need to install it for at least Web

- Please, be aware, that, for Windows, additional paths may need to be set up (for example, in system variables path C:\Windows\System32\WindowsPowerShell\v1.0 and in user variables path C:\Program Files\Git\bin\git.exe ; C:\Program Files\Git\cmd ; C:\Windows\System32)

Widgets

Central class hierarchy in the Flutter framework

All the instances of a widget are immutable

Describe what their view should look like given their current configuration and state

Types:

- Stateful widgets
- Stateless widgets

Basic widgets:

- Text
- Row, Column
- Stack
- Container

Stateless vs Stateful Widgets

State = information that can be read synchronously when the widget is built and might change during the lifetime of the widget

Stateless

- Does not require a mutable state
- They are used when the part of the UI does not depend on other parts

Stateful

- Has a mutable state
- They are used when the UI might change multiple times over time

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

Run | Debug | Profile
void main() {
  runApp(const MyApp());
}

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ), // ThemeData
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    ); // MaterialApp
  }
}
```

```
class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});
  final String title;

  @override
  State<MyHomePage> createState() => _MyHomePageState();
}

class _MyHomePageState extends State<MyHomePage> {
  int _counter = 0;

  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }
```

```
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text(
              'You have pushed the button this many times:',
            ), // Text
            Text(
              '$_counter',
              style: Theme.of(context).textTheme.headline4,
            ), // Text
          ], // <Widget>[]
        ), // Column
      ), // Center
      floatingActionButton: FloatingActionButton(
        onPressed: _incrementCounter,
        tooltip: 'Increment',
        child: const Icon(Icons.add),
      ),  // FloatingActionButton
    ); // Scaffold
  }
}
```

Basic widgets

Text: Lets you create a run of styled text within your application

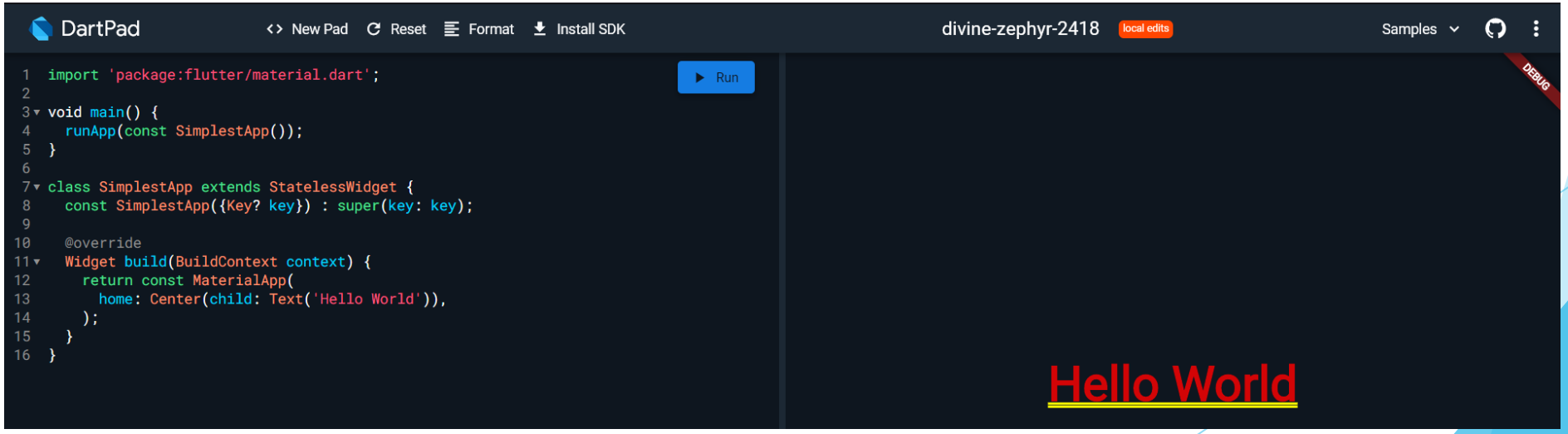
Row, Column: Flex widgets that let you create flexible layouts in both the horizontal (Row) and vertical (Column) directions

Stack: Instead of being linearly oriented (either horizontally or vertically), a Stack widget lets you place widgets on top of each other in paint order. You can then use the Positioned widget on children of a Stack to position them relative to the top, right, bottom, or left edge of the stack.

Container: Lets you create a rectangular visual element. A container can be decorated with a BoxDecoration, such as a background, a border, or a shadow. A Container can also have margins, padding, and constraints applied to its size.

Text Widget

- Import the package necessary for building flutter apps
- Main function->runApp function is called by passing a widget as a parameter (making it the root of the app)
- SimplestApp - user defined class that inherits Stateless Widget (also has a constructor with 1 optional parameter – key)
- @override - marks an instance member as overriding a superclass member with the same name.
- build - method responsible for creating the components
- MaterialApp – predefined class used to access all the other components of Flutter SDK
- Center – center the content both vertically and horizontally



TextField Widget

- Scaffold – predefined class used under MaterialApp, giving basic functionalities
- Padding – widget used to inset its child by the given padding
- Column – aligns the children by column

DartPad | New Pad | Reset | Format | Install SDK | divine-zephyr-2418 local edits | Samples

Run

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp( home: MyApp(),));
}
class MyApp extends StatelessWidget {
@override
Widget build(BuildContext context) {
  return Scaffold(
      body: Padding(
          padding: EdgeInsets.all(30),
          child: Column(
            children: <Widget>[
              Padding(
                padding: EdgeInsets.all(30),
                child: TextField(
                  decoration: InputDecoration(
                    border: OutlineInputBorder(),
                    labelText: 'First TextField',
                  ),
                ),
              ),
              Padding(
                padding: EdgeInsets.all(30),
                child: TextField(
                  obscureText: true,
                  decoration: InputDecoration(
                    border: OutlineInputBorder(),
                    labelText: 'Second TextField',
                  ),
                ),
              ),
            ],
```

DEBUG

First TextField

I can write in it

Second TextField

Row, Column Widget
Container Widget

- AppBar – used to create a fixed widget at the top of the screen
- Container – used to surround a child with padding, then add additional constraints to it
- BoxDecoration – adds decoration to a container

DartPad New Pad Reset Format Install SDK divine-zephyr-2418 local edits Samples

```dart
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp( home: MyApp(),));
}
class MyApp extends StatelessWidget{
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Row Example"),
      ),
      body: Row(
          mainAxisAlignment: MainAxisAlignment.spaceEvenly,
          children:<Widget>[
            Container(
              margin: EdgeInsets.all(10.0),
              padding: EdgeInsets.all(10.0),
              decoration:BoxDecoration(
                  borderRadius:BorderRadius.circular(10),
                  color:Colors.pink
              ),
              child: Text("Row1",style: TextStyle(color:Colors.blueAccent,fontSize:20),),
            ),
            Container(
              margin: EdgeInsets.all(10.0),
              padding: EdgeInsets.all(10.0),
              decoration:BoxDecoration(
                  borderRadius:BorderRadius.circular(10),
                  color:Colors.red
              ),
              child: Text("Row2",style: TextStyle(color:Colors.white,fontSize:20),),
            ),
            Container(
              margin: EdgeInsets.all(10.0),
              padding: EdgeInsets.all(10.0),
              decoration:BoxDecoration(
                  borderRadius:BorderRadius.circular(10),
                  color:Colors.purple
              ),
              child: Text("Row3",style: TextStyle(color:Colors.black,fontSize:20),),
            )
```

Run

Row Example

Row1 Row2 Row3

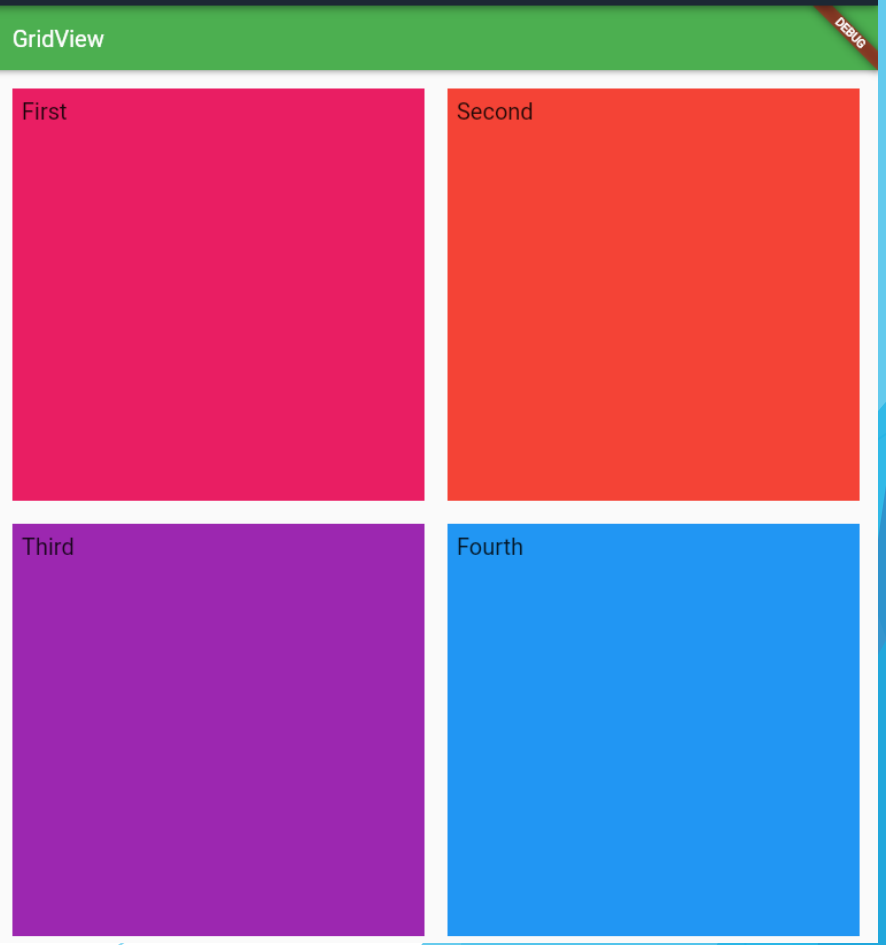
GridView Widget

- Gridview.extent – creates a scrollable 2D array of widgets
- EdgeInsets – specify offsets in terms of visual edges, left, top, right and bottom

DartPad New Pad Reset Format Install SDK divine-zephyr-2418 local edits Samples

Run

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    home: FirstRoute(),
  ));
}
class FirstRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("GridView"),
        backgroundColor: Colors.green,
      ),
      body: Center(
          child: GridView.extent(
            primary: false,
            padding: const EdgeInsets.all(16),
            crossAxisSpacing: 20,
            mainAxisSpacing: 20,
            maxCrossAxisExtent: 400.0,
            children: <Widget>[
              Container(
                padding: const EdgeInsets.all(8),
                child: const Text('First', style: TextStyle(fontSize: 20)),
                color: Colors.pink,
              ),
              Container(
                padding: const EdgeInsets.all(8),
                child: const Text('Second', style: TextStyle(fontSize: 20)),
                color: Colors.red,
              ),
              Container(
                padding: const EdgeInsets.all(8),
                child: const Text('Third', style: TextStyle(fontSize: 20)),
                color: Colors.purple,
              ),
              Container(
                padding: const EdgeInsets.all(8),
                child: const Text('Fourth', style: TextStyle(fontSize: 20)),
                color: Colors.blue,
```



Stack Widget

- Positioned – controls where a child is positioned
- Allign – used to set the alignment of the child (in this case, the text in the top widget)

DartPad New Pad Reset Format Install SDK divine-zephyr-2418 local edits Samples

Run

```
1   import 'package:flutter/material.dart';
2   void main() {
3     runApp(MaterialApp( home: MyApp(),));
4   }
5   class MyApp extends StatelessWidget {
6 @override
7  Widget build(BuildContext context) {
8    return MaterialApp(
9      home: Scaffold(
10       appBar: AppBar(
11         title: Text("Stack Widget"),
12       ),
13       body: Center(
14         child: Stack(
15           fit: StackFit.passthrough,
16           children: <Widget>[
17             Container(
18               height: 300,
19               width: 400,
20               color: Colors.purple,
21               child: const Align(
22                 alignment: Alignment(0.2,0.6),
23                 child: Text('Top Widget',
24                   style: TextStyle(color: Colors.white, fontSize: 20),
25                 ),
26               ),
27             ),
28             Positioned(
29               top: 30,
30               right: 20,
31               child: Container(
32                 height: 100,
33                 width: 150,
34                 color: Colors.blue,
35                 child: Center(
36                   child: Text('Middle Widget',
37                     style: TextStyle(color: Colors.white, fontSize: 20),
38
39                 ),
40               ),
41             ),
42             Positioned(
43               top: 100,
44               left: 240,
45               child: Container(
46                 height: 100,
47                 width: 150,
48                 color: Colors.pink,
49                 child: Center(
50                   child: Text('Bottom Widget',
51                     style: TextStyle(color: Colors.white, fontSize: 20),
52                 ),
53               ),
54
```

Stack Widget

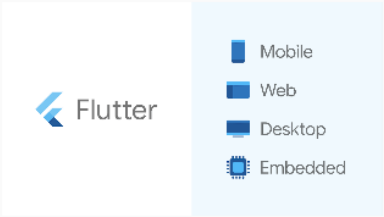
Adding images

- Image.network – widget used to display an image from a link

DartPad | New Pad | Reset | Format | Install SDK | divine-zephyr-2418 local edits | Samples

```
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    home: FirstRoute(),
  ));
}

class FirstRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        body: Center(
          child: Column(
            children: <Widget>[
              Image.network(
                  'https://1.bp.blogspot.com/-ak_aEW48mmA/XNCDH1UYfaI/AAAAAAAAHeg/NUh
                  height: 500,
                  width: 250
              ),
              Text(
                  'This is an image from a url.',
                  style: TextStyle(fontSize: 20.0),
              )
            ],
          ),
        ),
      ),
    );
  }
}
```



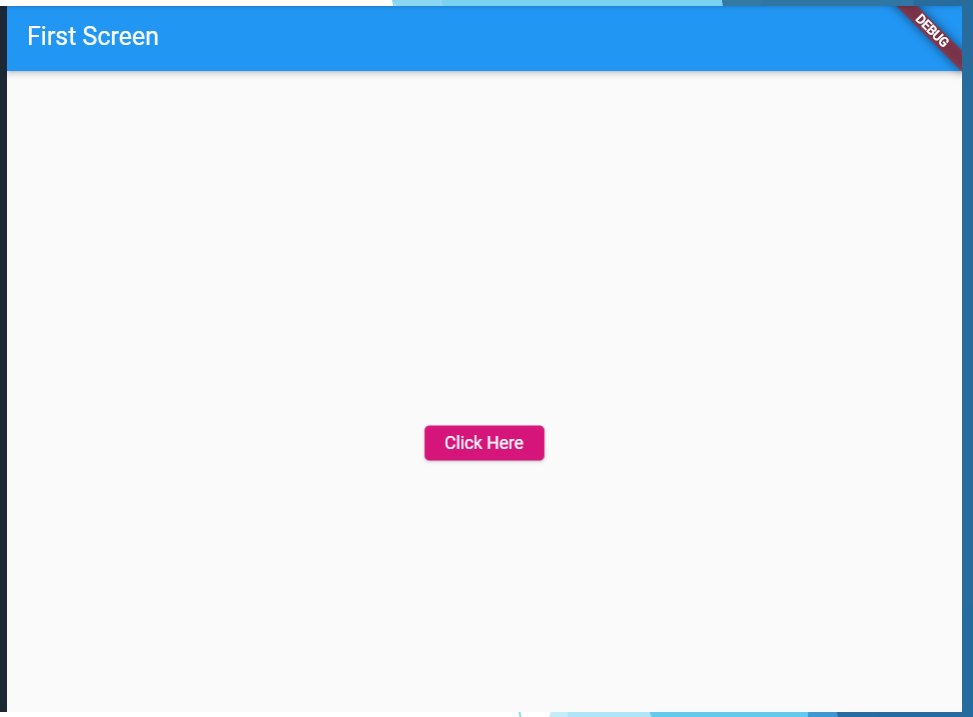
This is an image from a url.

Navigation

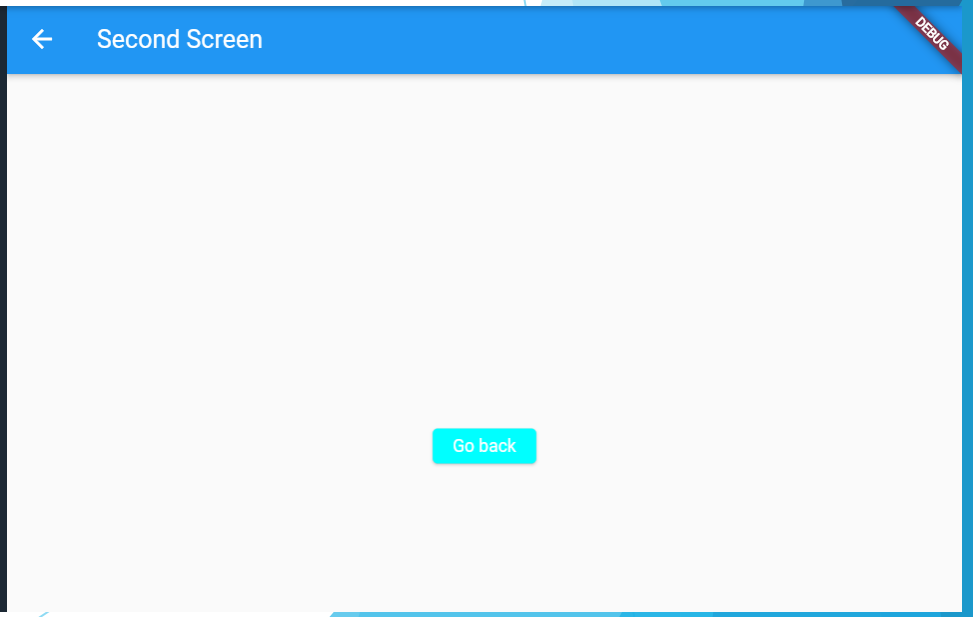
- ElevatedButton - used to create personalized buttons
- onPressed() - adds functionality if the button is pressed
- Navigator - widget used to navigate through pages
- push() - method used to push a route to the navigator
- pop() - method used to pop the top-most route

```flutter/material.dart
import 'package:flutter/material.dart';
void main() {
  runApp(MaterialApp(
    home: FirstRoute(),
  ));
}

class FirstRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Screen'),
      ),
      body: Center(
        child: ElevatedButton(
          child: Text('Click Here'),
          style: ElevatedButton.styleFrom(
           backgroundColor : Color.fromARGB(225,219,0,115)),
          onPressed: () {
            Navigator.push(
              context,
              MaterialPageRoute(builder: (context) => SecondRoute()),
            );
          },
        ),
      ),
    );
  }
}
```



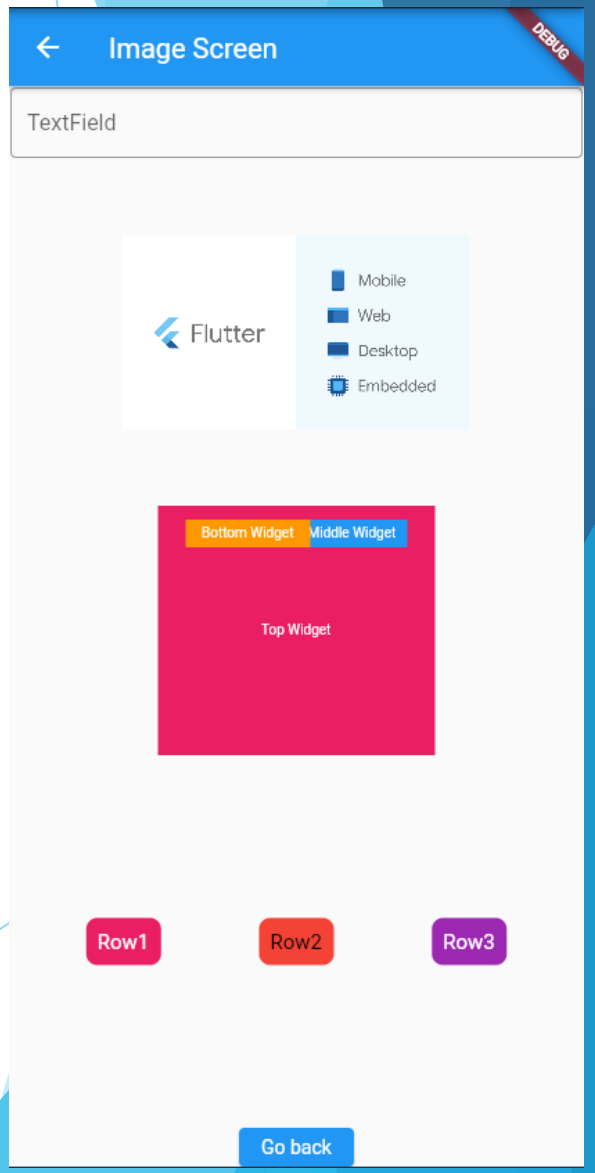

```flutter/material.dart
class SecondRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Second Screen"),
      ),
      body: Center(
        child: ElevatedButton(
          style: ElevatedButton.styleFrom(
           backgroundColor : Color.fromARGB(255,0,255,255)),
          onPressed: () {
            Navigator.pop(context);
          },
          child: Text('Go back'),
        ),
      ),
    );
  }
}
```



Exercises

- Go to dartpad.dev and create 2 navigable pages:
  - Home page has:
    - At least 1 Text Widget
    - A grid with at least 6 different elements
    - A button to go to the Image page

    *(Hint: you may put these elements inside a Column; you can put GridView as a child inside a new Expanded class because you may need to constrain the height of GridView in order to fit inside Column)
  - Image page has:
    - At least 1 TextField Widget
    - An image
    - A stack of at least 3 elements
    - A row with at least 3 elements
    - Button to return to Home page



*The images are just an example; you can create any pages you wish as long as you follow the requirements

Exercises- example how to start

```
1  import 'package:flutter/material.dart';
2  void main() {
3    runApp(MaterialApp(
4      home: FirstRoute(),
5    ));
6  }
7
8  class FirstRoute extends StatelessWidget {
9    @override
10   Widget build(BuildContext context) {
11     return Scaffold(
12       appBar: AppBar(
13         title: Text('Home Screen'),
14       ),
15
16               body: Center(
17           child: Column(
18             children: <Widget>[
19               Element1-Text,
20           Element2-Expanded (child: GridView, children:Container),
21           Element3-ElevatedButton
22
23
24             ],
25         ),
26
27         ),
28     );
29   }
30 }
31
```

```
32 class SecondRoute extends StatelessWidget {
33   @override
34   Widget build(BuildContext context) {
35     return Scaffold(
36       appBar: AppBar(
37         title: Text("Image Screen"),
38       ),
39         body: Center(
40           child: Column(
41             children: <Widget>[
42         Element1 - Padding (TextField),
43         Element2 - Image,
44         Element3 - Stack,
45         Element4 - Expanded (child: Row, children: Container)
46         Element5 - Elevated Button
47
48             ],
49         ),
50
51         ),
52
53     );
54   }
55 }
```

Homework

- Complete the installation for at least Web (All the examples in the next labs are for web, Flutter SDK version `3.0.0`. If you use a different version, then you might encounter issues with dependencies provided in the lab)
- Create a new application, that we will use in the next labs:
  - Create a Log in page (email and password)
  - Create a Registration page (at least email and password)
  - Create a “Products” page that will display 2 products with image and name
  - Conditions:
    - The pages should be navigable
    - Log in and Registration pages should contain at least 2 TextField and 2 buttons
- All hw will be checked during the next lab