Biomedical Informatics

Koumakis Lefteris, Maria Chatzimina

What is MaterialApp?

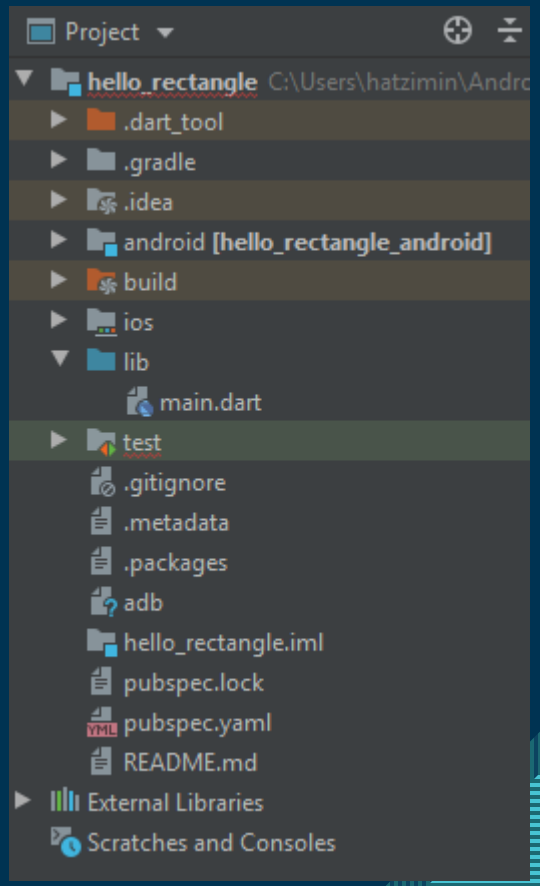
- **MaterialApp** is a core component that provides all the other elements (widgets) offered by the **Flutter SDK**, such as:
  - Scaffold
  - AppBar
  - Text widget
  - DropdownButton widget
- It is the **entry point of the application** and informs Flutter that the application will follow the **Material Design** guidelines.
- **Basic Attributes**
  - **title:** Text that appears as the application title
  - **home:** The first screen that will be shown to the user

Scaffold and Container Widgets

- **Scaffold widget**
- The **Scaffold** widget provides several built-in properties such as:
  - **AppBar** – the bar displayed at the top of the application
  - **Body**
  - **Bottom Navigation**
  - Scaffold provides a basic structure for the application that follows the **Material Design guidelines created by Google**.
  - Scaffold Class: https://api.flutter.dev/flutter/material/Scaffold-class.html
  - App Samples: https://flutter.dev/docs/catalog/samples/Scaffold
- Container widget:
  - The **Container** widget is a basic widget that can contain other widgets.
  - It is used to decorate its **child widgets** using properties such as:
    - borders
    - padding
    - alignment
    - height
    - width
  - Container Class: https://api.flutter.dev/flutter/widgets/Container-class.html

Folders and Files

- **android folder**
  An Android project containing the compiled Flutter code
  (we do not modify or use this folder directly).
- **build folder**
  Contains the output of the Flutter project.
  It is generated and managed by the **Flutter SDK**
  (we do not modify or use this folder).
- **iOS folder**
  Similar to the Android folder
  (we do not modify or use this folder).
- **lib folder**
  The folder where we create and add all **Dart files**.
- **test folder**
  The folder used for creating **tests** for the application
  (it will not be used in this course).

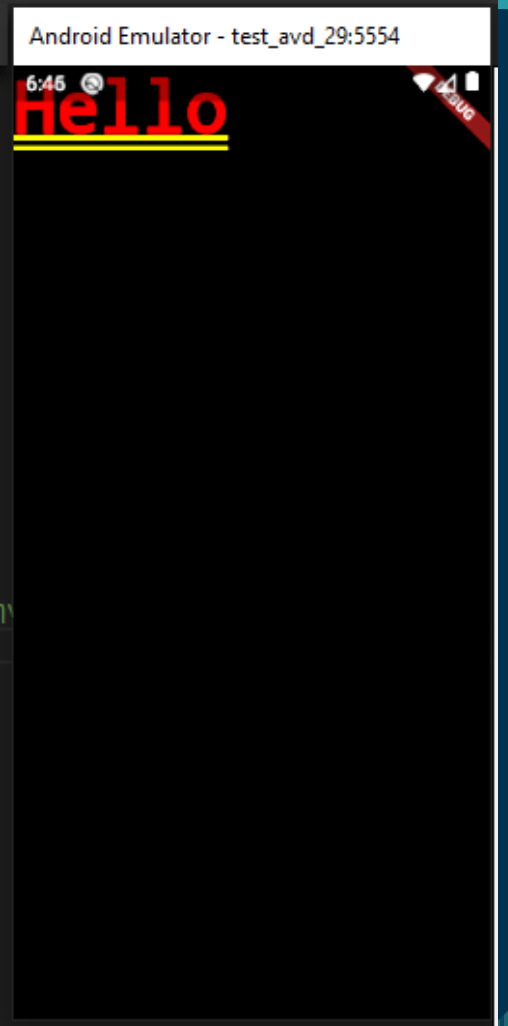


Simple App

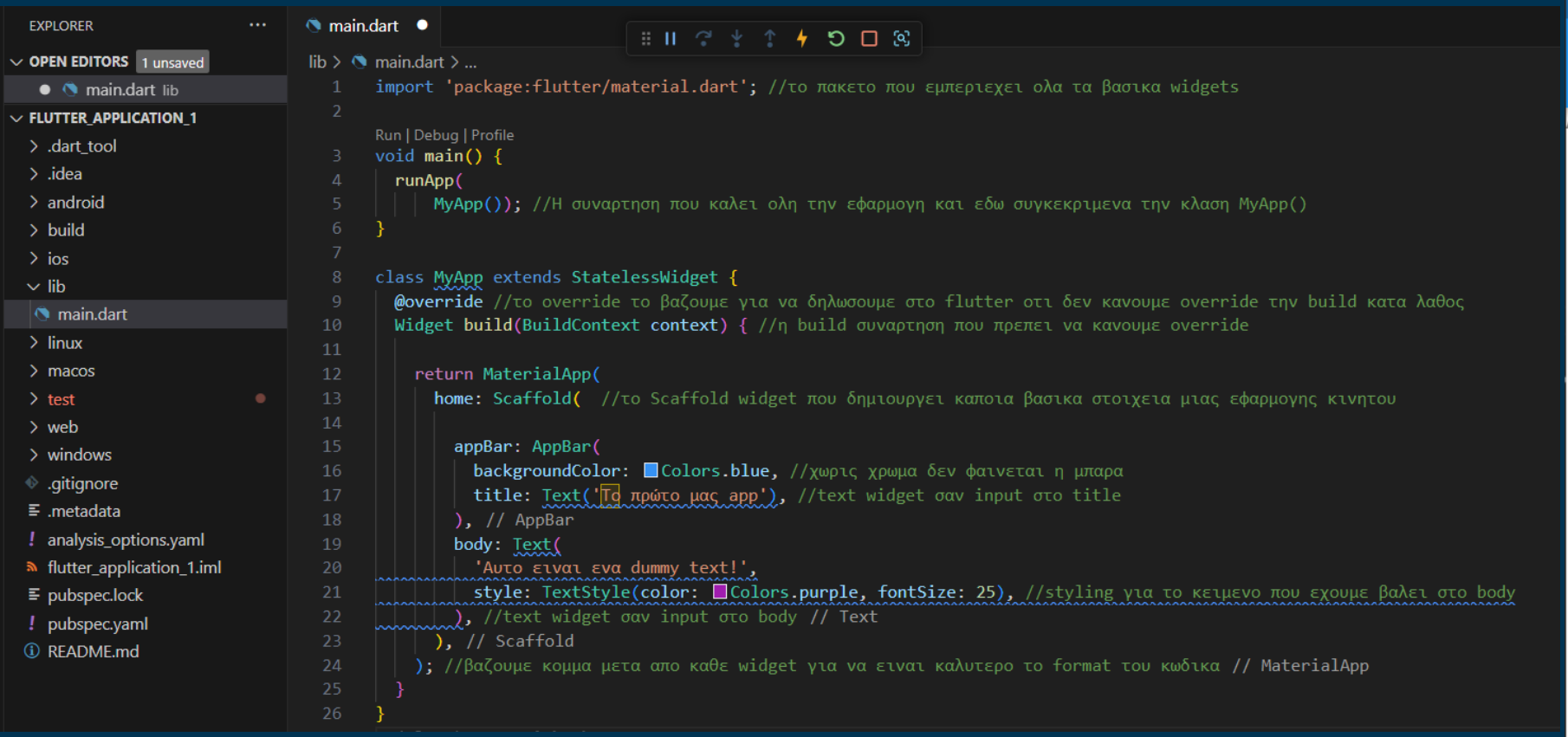
main.dart

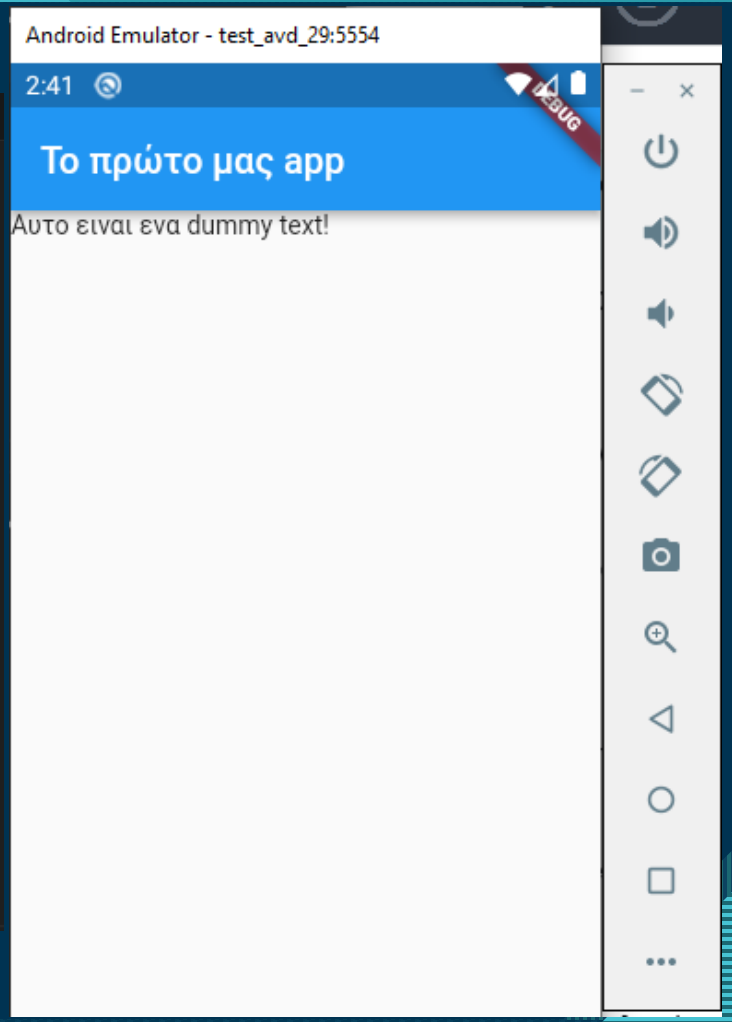
lib > main.dart > MyApp > build

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

Run | Debug
void main() { //η main() ειναι η πρωτη συναρτηση που καλει το flutter

  runApp(
      MyApp()); //Η συναρτηση που καλει ολη την εφαρμογη και εδω συγκεκριμενα την κλαση MyApp()
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {  //override build μεθοδος και επιστρεφει ενα widget

    return MaterialApp( //το MaterialApp ειναι το widget που συνδυαζει ολα τα widget και δημιουργει την

      home: Text(
          'Hello'), //Το text widget που δινουμε κειμενο για να τυπωθει στην οθονη. // Text
      //Βαζουμε κομμα μετα απο καθε widget για να ειναι καλυτερο το format του κωδικα
    ); // MaterialApp
  }
}
```



App with styling





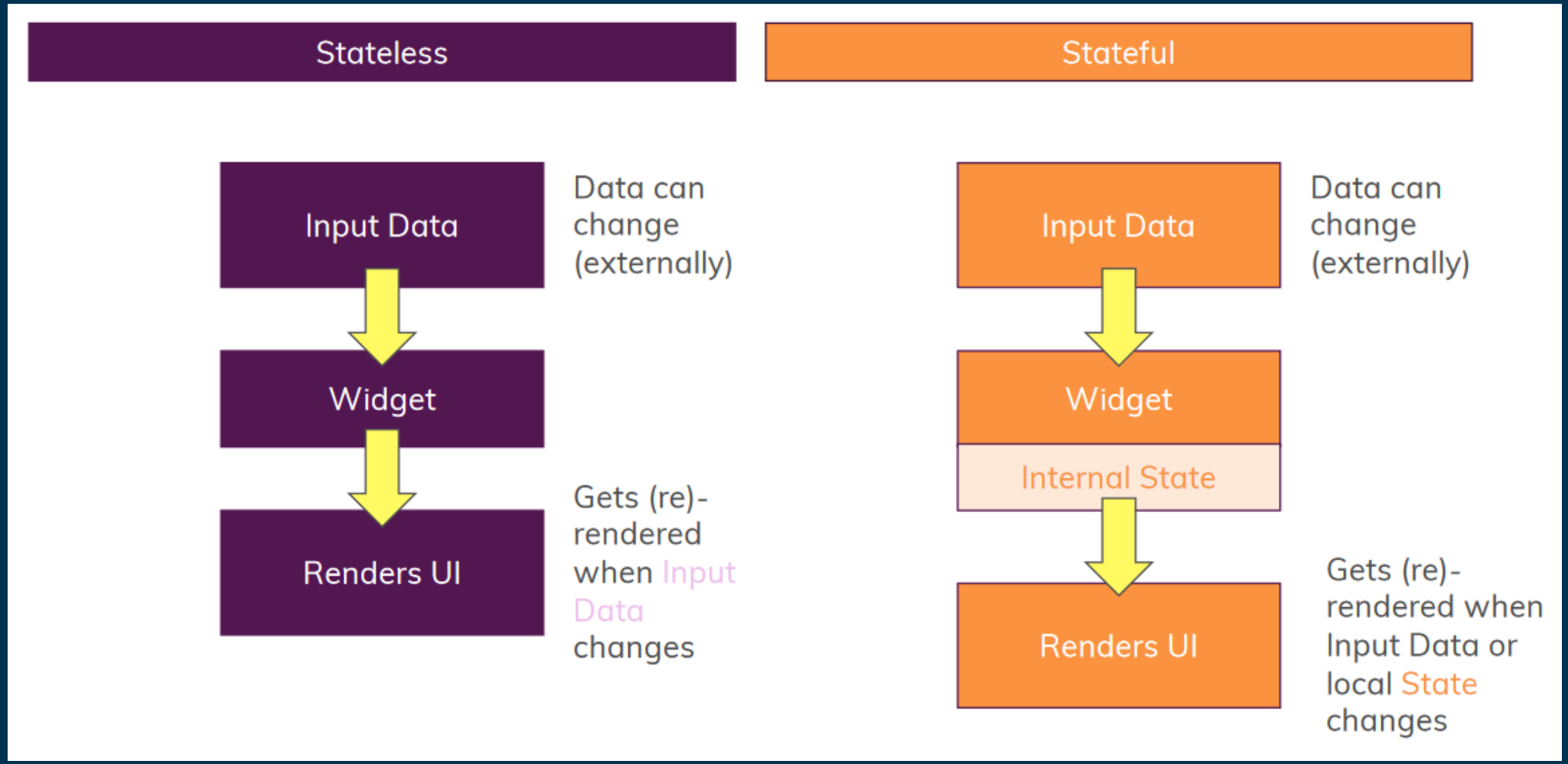
Questions

- What is the role of runApp()?
  - This function creates the main widget and calls the **build()** function of the widget.
- What does the build() method do?
  - It returns the **widget tree** that should be displayed on the mobile screen.
- **What are Widgets?**
  - They are the basic building blocks of Flutter used to create the **user interface**.

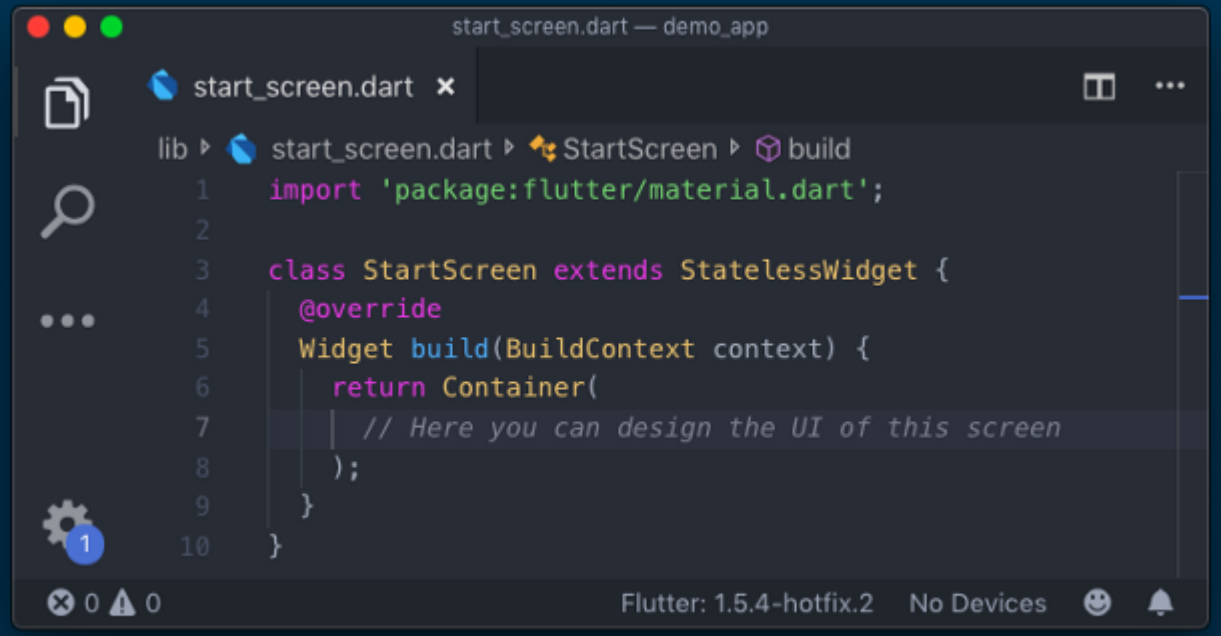
Ερωτήσεις

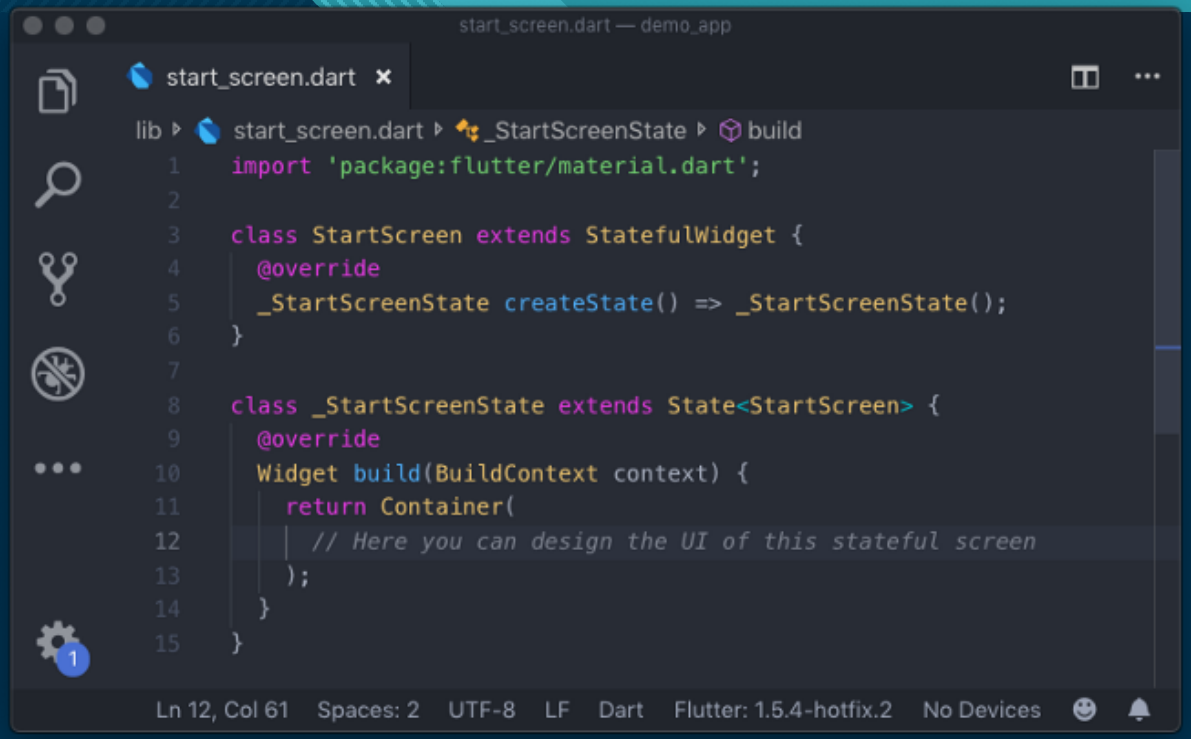
- **What is the role of a Widget?**
  - The **user interface (UI)** of your application is built using widgets.
- **What does the term "Widget tree" describe?**
  - The widget tree means that the UI is built by combining **nested widgets**.
- **Why do we need Widgets?**
  - Widgets are the fundamental components used to build **rich user interfaces in Flutter**.
- **How are Widgets combined?**
  - Widgets are passed into the constructors of other widgets to achieve the desired result.
- **How does a Dart object become a Widget?**
  - By extending **StatelessWidget** or **StatefulWidget** and implementing a **build() method**.

Stateless vs Stateful



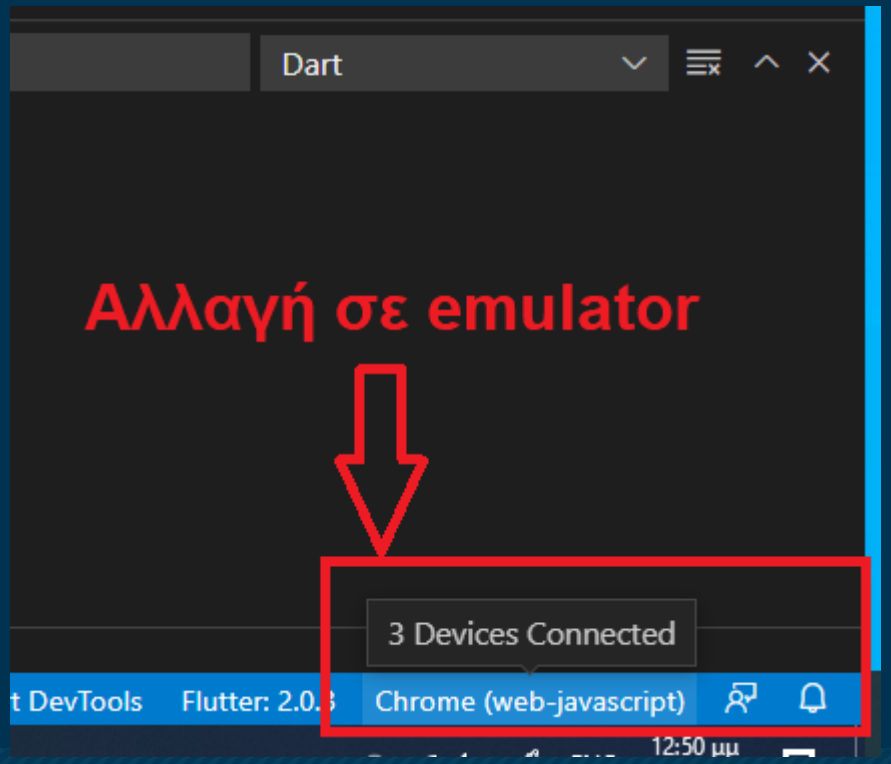
Stateless vs Stateful



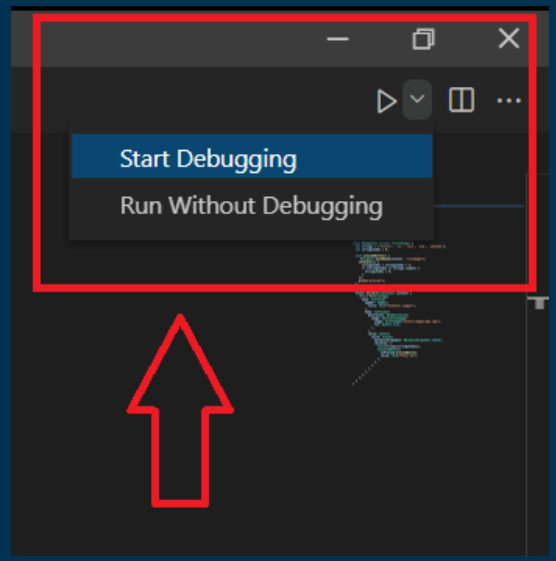


Run project

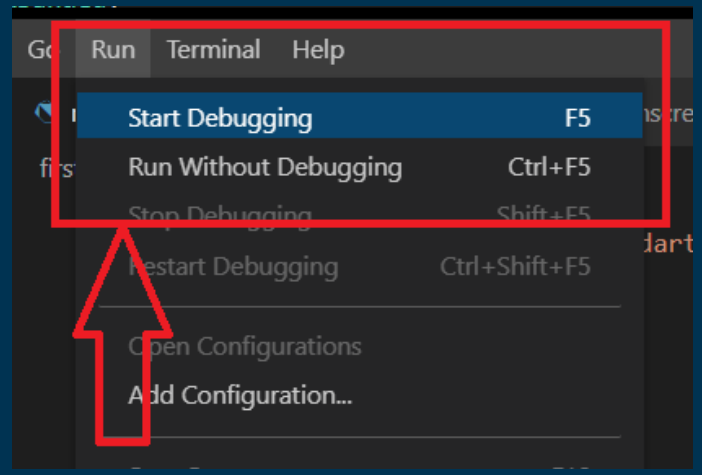
- 1 βήμα – Select an **emulator**



- 2 βήμα – Run code



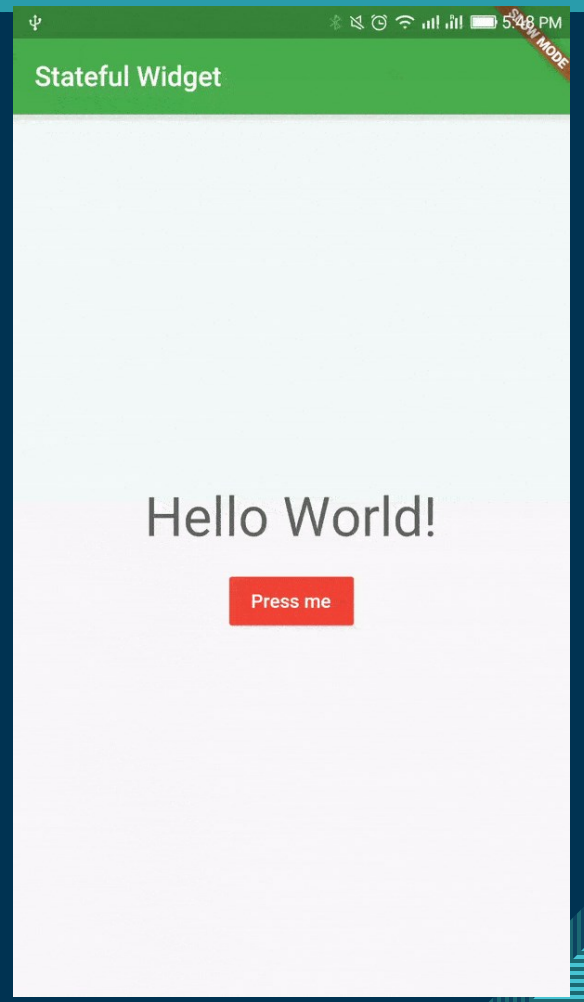
ή



Flutter Examples

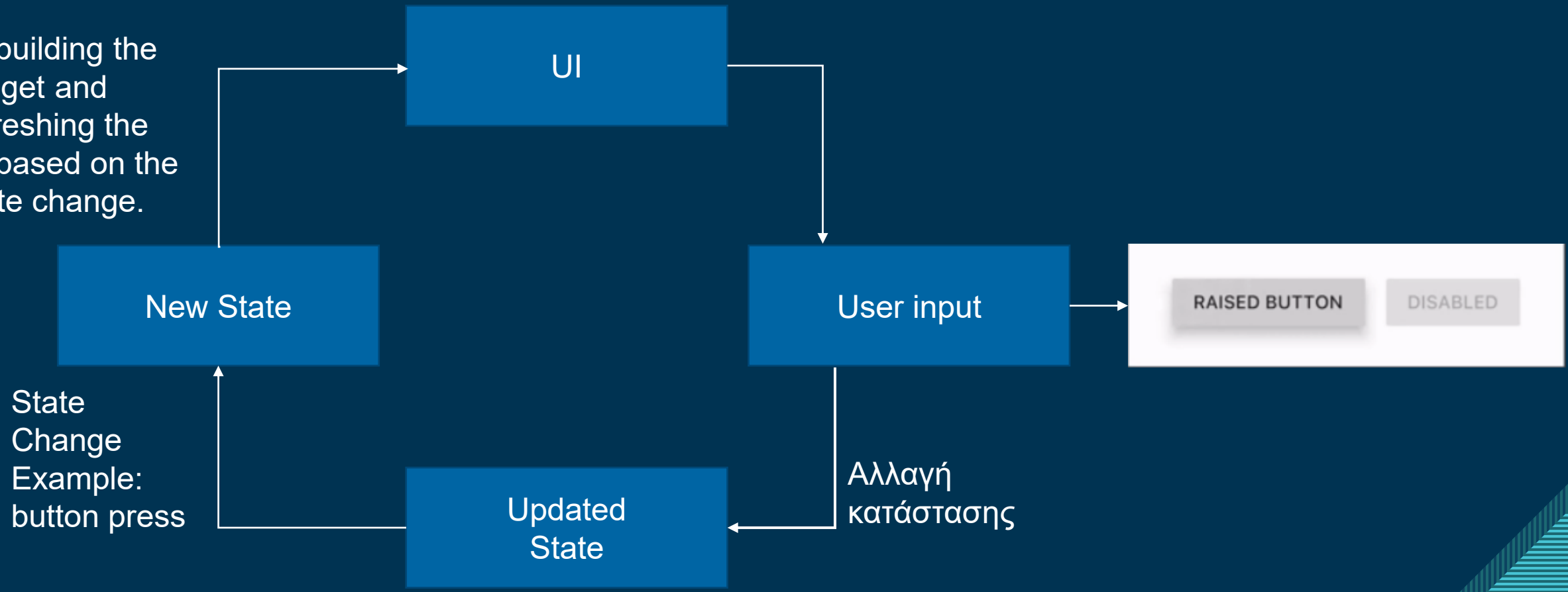
```dart
import 'package:flutter/cupertino.dart';
import 'package:flutter/material.dart';

Run | Debug
void main() {
  runApp(MyApp()); //η ετοιμη συναρτηση που τρεχει την εφαρμογη μας
}

class MyApp extends StatefulWidget {
  @override
  State<StatefulWidget> createState() {
    return MyAppState();   //συνδεουμε το MyApp με το State παρακατω
  }
}

class MyAppState extends State<MyApp> { //συνδεουμε το State με το widget MyApp
  var stringIndex = 0;

  void onPressedButton() {
    //stringIndex++;
    setState(() {
      stringIndex = stringIndex + 1;
      if (stringIndex > 4) {  //ελεγχω να μην ξεπεραστει ο αριθμος στοιχειων της λιστας
        stringIndex = 0;
      }
    });

    print(stringIndex);
    //print('Button pressed!');
  }

```

```dart
@override
Widget build(BuildContext context) {
  return MaterialApp(
  home: Scaffold(
      appBar: AppBar(
        title: const Text('Stateful widget'),
        backgroundColor: Colors.cyan,
      ), // AppBar
      body:Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(dynamicStrings[questionIndex],
                  style: const TextStyle(fontSize: 30)), // Text
              const Padding(padding: EdgeInsets.all(10)),
              ElevatedButton(
                  onPressed: pressedButton,
                  style: ElevatedButton.styleFrom(
                  primary: Colors.grey, //χρωμα κουμπιου
                  onPrimary: Colors.white),//χρωμα κειμενου κουμπιου
                  child: const Text('Press me'),
              ), // ElevatedButton
              //Διαφορετικοί τρόποι να ορισεις συναρτησεις στο onPressed
              //onPressed: ()=> print("You pressed the button")
              //onPressed: (){
              //
              //
              //print("You pressed the button")
              //}

              ],
            ), // Column
      ), // Center
  ), // Scaffold
  ); // MaterialApp
```



Widget lifecycle

Rebuilding the widget and refreshing the UI based on the state change.



Stateless vs Stateful

- **Stateless Widgets**
  - Do not have a method to re-render themselves when their internal data changes.
  - They do not have an **internal state**, as the name suggests.
  - If a Stateless Widget is rebuilt, the **entire widget is rebuilt**.
- **Stateful Widgets**
  - Can be rebuilt when their internal or external data changes.
  - The **setState()** method triggers the rebuilding of elements affected by the state change.
  - It does not rebuild the entire widget, only the necessary parts.

Stateful Widgets

- https://flutter.io/tutorials/interactive/#stateful-stateless
- **setState() is a special function available only in Stateful Widgets.**
- **It updates the state of the widget and rebuilds the UI elements affected by the state change.**
- **We define the data that needs to be updated inside the setState() block(in our example we used stringIndex).**
- **Only the Text() widget will be rebuilt because it interacts with the change of the stringIndex value.**

Questions

- What is the difference between StatelessWidget and StatefulWidget?
  - A StatelessWidget cannot re-run the build() method during the execution of the application.
- Why do you need to call setState(() { ... }) in a StatefulWidget when data changes?
  - Without calling setState, the widget will not re-run the build() method, and the changes will not appear on the screen.

Flutter Examples

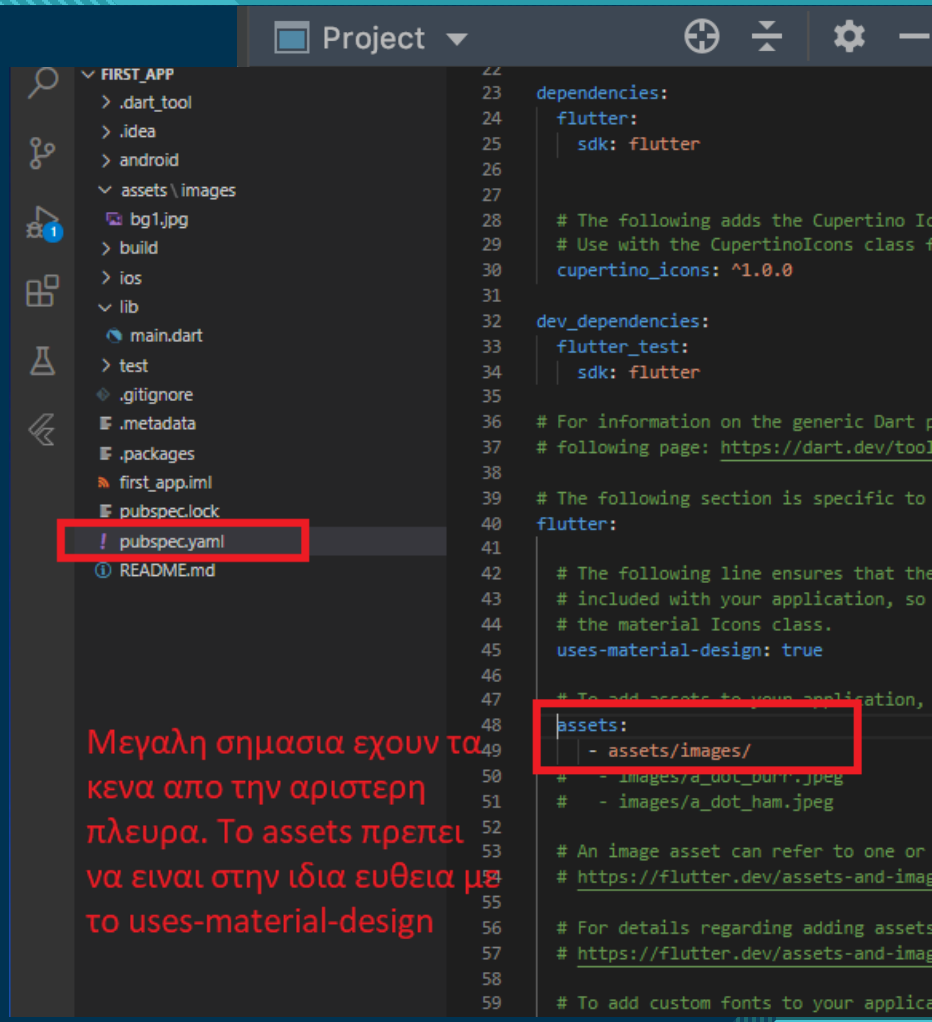
Adding a Background Image

1. Create a folder **assets/images**
   - Use any folder names you want
2. Save the image inside the **images** folder
3. Declare the assets in **pubspec.yaml**
   - Open the file **pubspec.yaml**
   - Add an **assets** subcategory inside the **flutter** section like this:

```
flutter:
  assets:
    - assets/images/
```

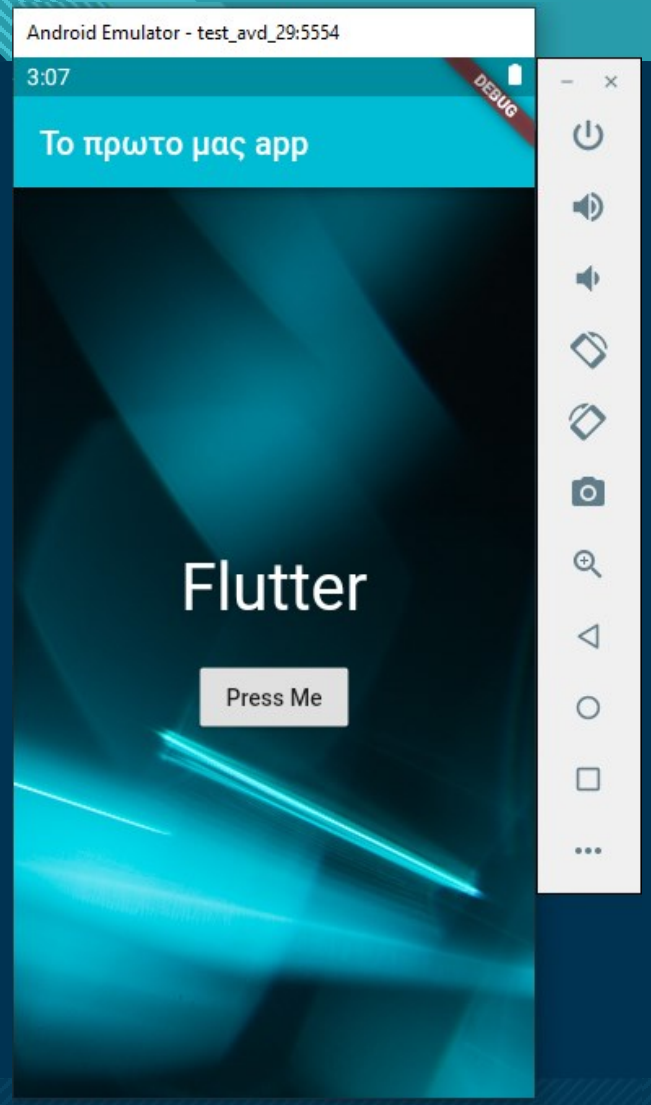
⚠ Important: indentation (spaces on the left) is important in this file.

4. Insert the image in main.dart:
   - Image.asset('assets/images/bg1.jpg')



Background Example

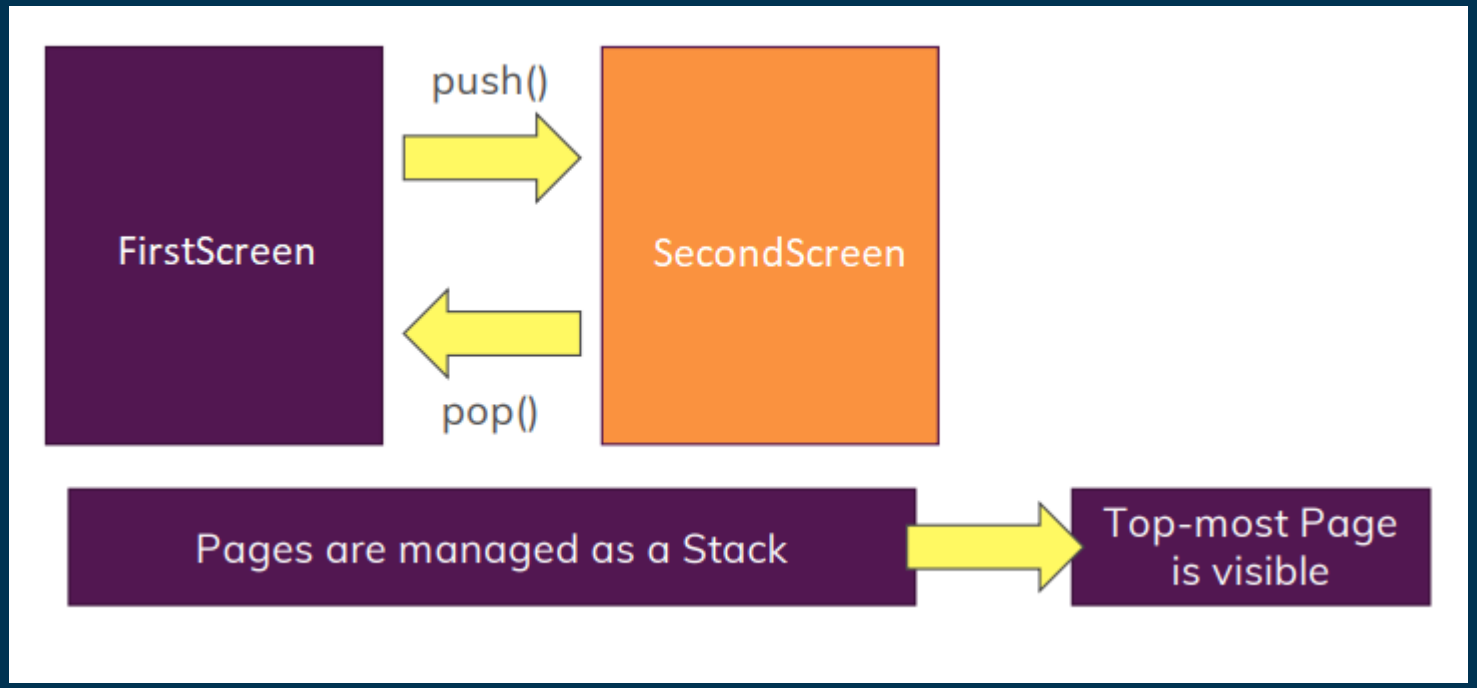
```dart
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: const Text('Stateful widget'),
        backgroundColor: Colors.cyan,
      ), // AppBar
      body: Container(
        decoration: const BoxDecoration(
          image: DecorationImage(
              image: AssetImage('assets/images/bg1.jpg'), fit: BoxFit.cover), // DecorationImage
        ), // BoxDecoration
        child: Center(
          child: Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text(dynamicStrings[questionIndex],
                  style: const TextStyle(fontSize: 30)), // Text
              const Padding(padding: EdgeInsets.all(10)),
              ElevatedButton(
                onPressed: pressedButton,
                style: ElevatedButton.styleFrom(
                    primary: Colors.grey, //χρωμα κουμπιου
                    onPrimary: Colors.white), //χρωμα κειμενου κουμπιου
                child: const Text('Press me'),
              ), // ElevatedButton
              //Διαφορετικοί τρόποι να ορισεις συναρτησεις στο onPressed
              //onPressed: ()=> print("You pressed the button")
              //onPressed: (){
              //
              //
              //print("You pressed the button")
              //}
            ],
          ), // Column
        ), // Center
      ), // Container
    ), // Scaffold
```

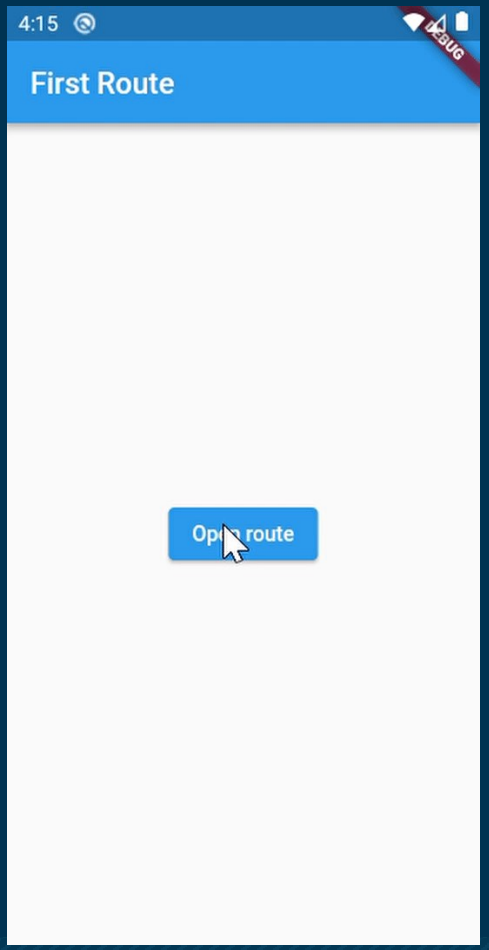


Flutter Navigation

- We use the **Navigator class**.
  - Navigator allows navigation **from one screen to another**.
  - It is connected to the **context variable**.
  - The **context variable** contains information about the widget class that is called and its position in the **Widget Tree**.
- Screens are managed as a **stack**.
- The **top-most page is visible**.

Flutter Navigation





Flutter Navigation Code

- main.dart

main.dart > FirstRoute > build

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

Run | Debug | Profile
void main() {
  runApp(MaterialApp(
    title: 'Named Routes',
    initialRoute: '/',
    routes: {
      '/': (context) => FirstRoute(),
      '/second': (context) => SecondRoute(),
    },
  )); // MaterialApp
}

class FirstRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) { //το context κρατάει διαφορες πληροφοριες για το widget
  //και του συγκεκριμενου widget
    return Scaffold(
      appBar: AppBar(
        backgroundColor: Colors.blue,
        title: Text("First Route"),
      ), // AppBar
      body: Center(
        child: ElevatedButton( //αλλος τυπος κουμπιου και το RaisedButton θα δουλευε
          child: Text('Open route'),
          onPressed: () {
            Navigator.push( //οταν θελουμε να παμε σε καινουρια σελιδα χρησιμοποιουμε push
              context,
              MaterialPageRoute(builder: (context) => SecondRoute()),
            );
          },
        ), // ElevatedButton
      ), // Center
    ); // Scaffold
  }
}
```

- secondscreen.dart

lib > secondscreen.dart > SecondRoute > build

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

class SecondRoute extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Second Route"),
      ), // AppBar
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            Navigator.pop(context);//οταν θελουμε να επιστρεψουμε στην προηγουμενη
            //σελιδα χρησιμοποιουμε το pop
          },
          child: Text('Go back!'),
        ), // ElevatedButton
      ), // Center
    ); // Scaffold
  }
}

```

Named Routes

- In Flutter, screens and pages are called **routes**.
  - A **route** is an abstraction for a screen or page of an application.
- The **Navigator** is a widget that manages routes.

Map Class

- The **Map** is a class in **Dart**.
- It uses a **key-value structure**.
- It is declared using **{ }**
- Example:

```
var questions = [
  {
    'questionText': 'What\'s your favorite color?',
    'answers':'black',
  },
{
    'questionText': 'What\'s your favorite animal?',
    'answers': 'cat',
  },
];
```

Navigation Code with Named Routes

- main.dart

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

Run | Debug
void main() {
  runApp(MaterialApp(
    title: 'Named Routes Demo',
    //  Ξεκινα την εφαρμογη απο το name route "/" σε αυτη την περιπτωση
    // η εφαρμογη ξεκιναει στο FirstScreen widget.
    initialRoute: '/',
    routes: {
      // Οταν κανουμε πλοηγηση στο route "/" , γινεται build το FirstScreen widget.
      '/': (context) => FirstScreen(),
      // Οταν κανουμε πλοηγηση στο route "/second", γινεται το SecondScreen widget.
      '/second': (context) => SecondScreen(),
    },
  )); // MaterialApp
}

class FirstScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('First Screen'),
      ), // AppBar
      body: Center(
        child: ElevatedButton(
          child: Text('Launch screen'),
          onPressed: () {
            // Πηγαινε στην δευτερη σελιδα χρησιμοποιωντας το ονομα της (named route)
            Navigator.pushNamed(context, '/second');
          },
        ), // ElevatedButton
      ), // Center
    ); // Scaffold
  }
}
```

- secondscreen.dart

```dart
class SecondScreen extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text("Second Screen"),
      ), // AppBar
      body: Center(
        child: ElevatedButton(
          onPressed: () {
            // Επεστρεψε στην πρωτη σελιδα με το να αφαιρεσεις το τωρινο route
            // απο την λιστα(stack).
            Navigator.pop(context);
          },
          child: Text('Go back!'),
        ), // ElevatedButton
      ), // Center
    ); // Scaffold
  }
}
```

Navigator

- Different ways to navigate to a new page
  1. Using a dynamic route name declared in main.dart (named routes):
     - Navigator.pushNamed(context, routePage);
  2. Using widget initialization:
     - Navigator.of(context).push(MaterialPageRoute(builder:(context)=>CardPage()));
- Removing the current page from the stack:
  - Navigator.of(context).pushReplacement(MaterialPageRoute(builder:(context)=>CardPage()));
  - When pressing back, it goes to the previous page from the homepage (SplashScreen).
- Using named routes to remove the page:
  - Navigator.of(context).pushReplacementNamed(routePage);

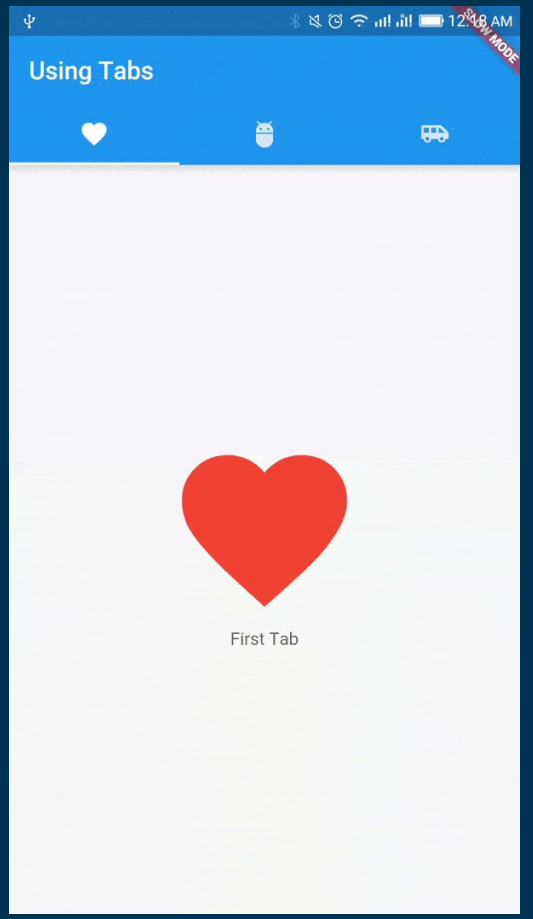
Tabs example

- Icons: https://api.flutter.dev/flutter/material/Icons-class.html

```dart
import 'package:flutter/material.dart';
import './tabs/first.dart'; //import τις σελίδες που φτιαξαμε
import './tabs/second.dart';
import './tabs/third.dart';

Run | Debug
void main() {
  runApp(MaterialApp(home: TabsScreen()));
}

class TabsScreen extends StatefulWidget {
  @override
  TabsScreenState createState() => TabsScreenState();
}

class TabsScreenState extends State<TabsScreen> {
  @override
  Widget build(BuildContext context) {
    return DefaultTabController(
      length: 3,
      child: Scaffold(
        appBar: AppBar(
          // Title
          title: Text("Using Bottom Navigation Bar"),
          // Set the background color of the App Bar
          backgroundColor: Colors.blue,
          bottom: TabBar(
            tabs: [
              Tab(icon: Icon(Icons.favorite)), //ετοιμα icons
              Tab(icon: Icon(Icons.adb)),
              Tab(icon: Icon(Icons.airport_shuttle)),
            ],
          ), // TabBar
        ), // AppBar
        //στο body καλω τις τρεις σελίδες που εφτιαξα. Αλλα τις καλω σαν κλασεις/αντικειμενα
        body: TabBarView(children: [FirstTab(), SecondTab(), ThirdTab()]),
      ), // Scaffold
    ); // DefaultTabController
  }
}
```

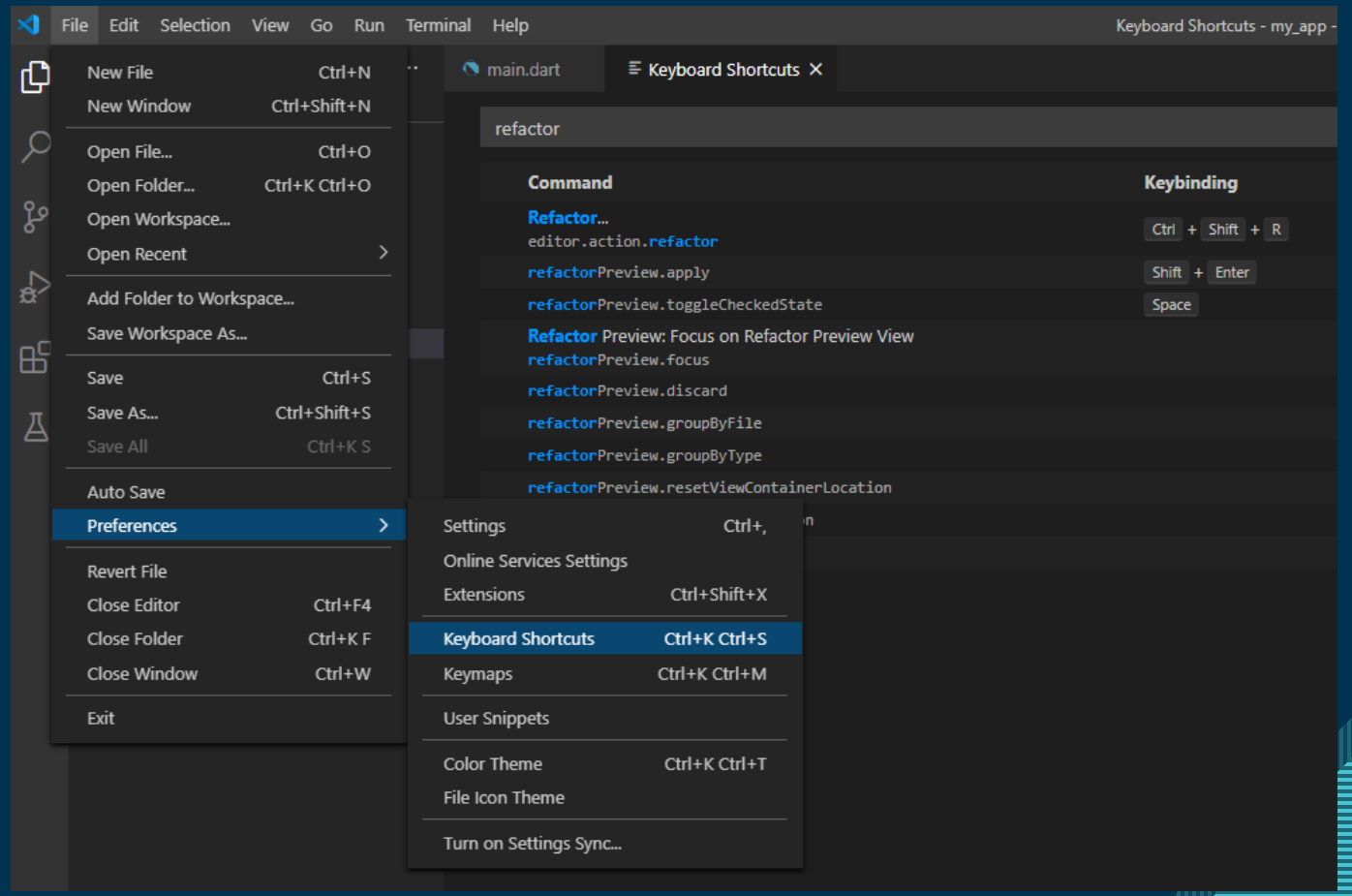
First.dart

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

class FirstTab extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Container(
      child: Center(
        child: Column(
          // center the children
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Icon(
              Icons.favorite,
              size: 160.0,
              color: Colors.red,
            ), // Icon
            Text(
              "First Tab",
              style: TextStyle(color: Colors.white),
            ) // Text
          ], // <Widget>[]
        ), // Column
      ), // Center
    ); // Container
  }
}
```



Refactor code - VS Code steps

- Refactor StatelessWidget class σε StatefulWidget class
- File→Preferences→Keyboard Shortcuts→Γράφουμε refactor



Private Properties

- Each **Dart file** can be considered a **closed ecosystem**.
- Different files can work together through **import**.
- If we add _ **(underscore)** before a class, function, or variable, it becomes **private** and can only be used within that file.

```dart
class MyApp extends StatefulWidget {
  @override
  State<StatefulWidget> createState() {
    return _MyAppState();
  }
}

class _MyAppState extends State<MyApp> {
  var _stringIndex = 0;

  void _onPressedButton() {
    //stringIndex++;
    setState(() {
      _stringIndex = _stringIndex + 1;
      if (_stringIndex > 4) {
        _stringIndex = 0;
      }
    });

    print(_stringIndex);
    //print('Button pressed!');
  }

```