Βιοϊατρική Πληροφορική

Κουμακης Λευτερης, Μαρία Χατζημηνά

Τι είναι το MaterialApp;

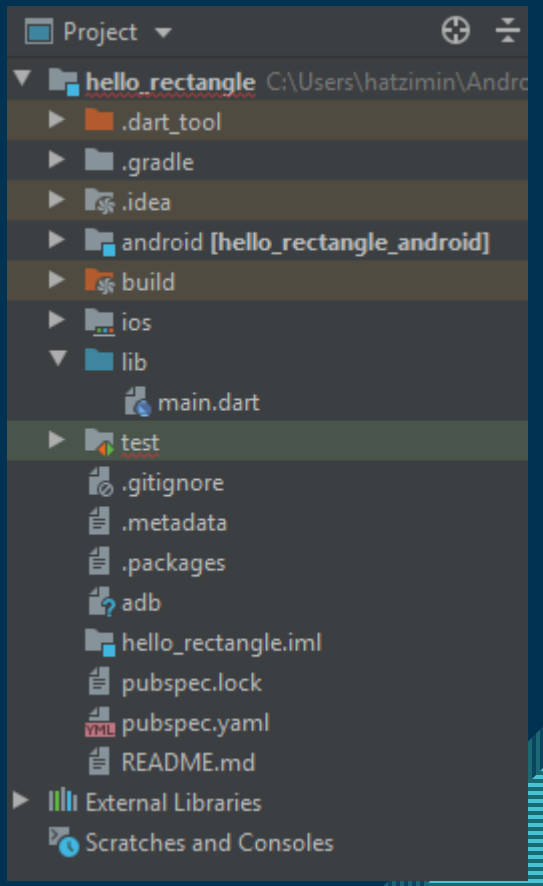
- Core component που μας δίνει όλα τα άλλα στοιχεία/widgets που προσφέρει το Flutter SDK όπως:
  - Scaffold
  - AppBar
  - Text widget
  - Dropdownbutton widget
- Αρχικό σημείο της εφαρμογής
- Πληροφορεί το flutter ότι θα ακολουθήσουμε τους κανόνες του Material Design
- Basic Attributes:
  - title: Κείμενο που εμφανίζεται σαν τίτλος
  - home: η πρώτη σελίδα που θα εμφανιστεί στον χρήστη

Scaffold and Container Widgets

- Scaffold widget:
  - Δίνει πολλές έτοιμες ιδιότητες όπως:
    - AppBar – Η μπάρα που εμφανίζεται στο πάνω μέρος της εφαρμογής
    - Body
    - Bottom Navigation
  - Το scaffold δίνει μια βασική εμφάνιση στην εφαρμογή που ακολουθεί τους κανονισμούς του Material Design που δημιουργήθηκε από την Google
  - Scaffold Class: https://api.flutter.dev/flutter/material/Scaffold-class.html
  - App Samples: https://flutter.dev/docs/catalog/samples/Scaffold
- Container widget:
  - Βασικό widget που εμπεριέχει άλλα widgets
  - Χρησιμοποιείται για να διακοσμήσει τα child widgets του χρησιμοποιώντας ιδιότητες όπως:
    - borders
    - padding
    - alignment
    - height
    - width
  - Container Class: https://api.flutter.dev/flutter/widgets/Container-class.html

Φάκελοι και Αρχεία

- Φάκελος android: Ένα android project με τον μεταγλωττισμένο κώδικα του Flutter (δεν μετατρέπουμε/χρησιμοποιούμε τον φάκελο)
- Φάκελος build: εμπεριέχει το output του Flutter project. Δημιουργείται και διαχειρίζεται από το Flutter SDK (δεν μετατρέπουμε/χρησιμοποιούμε τον φάκελο)
- Φάκελος iOS: Αντίστοιχο με τον Android φάκελο. (δεν μετατρέπουμε/χρησιμοποιούμε τον φάκελο)
- Φάκελος lib: Ο φάκελος που εισάγουμε/δημιουργούμε όλα τα Dart αρχεία
- Φάκελος test: Ο φάκελος που χρησιμοποιείται για την δημιουργία test για την εφαρμογή μας (δεν θα χρησιμοποιηθεί στο μάθημα)

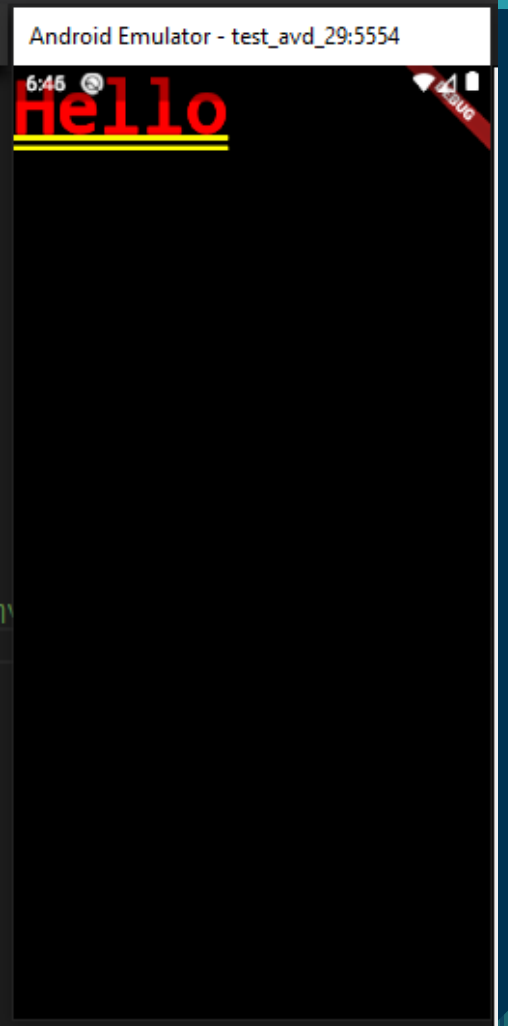


Simple App

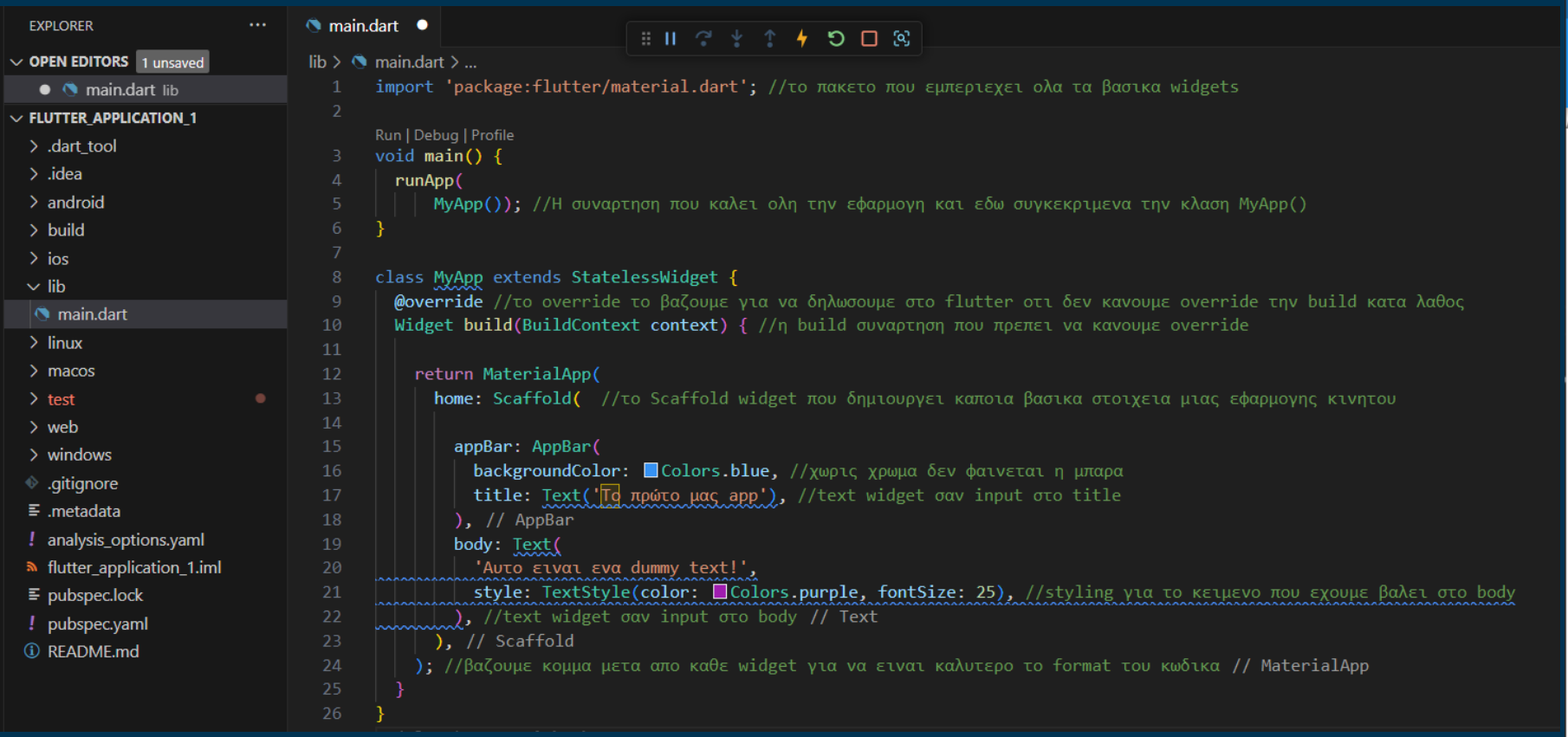
main.dart

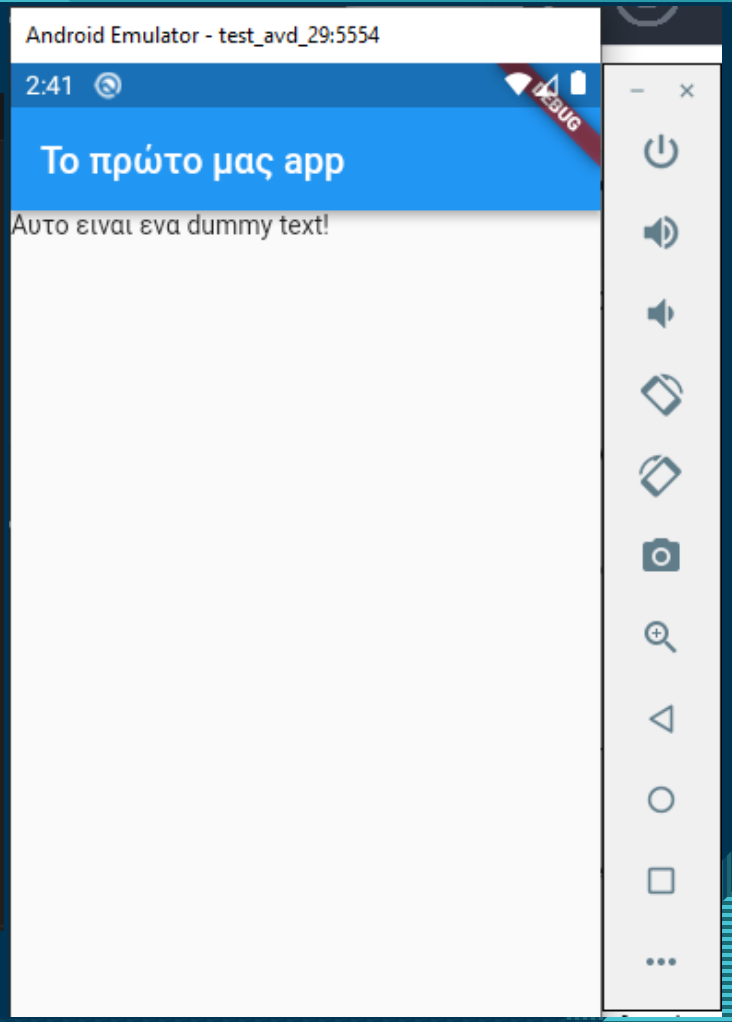
lib > main.dart > MyApp > build

```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





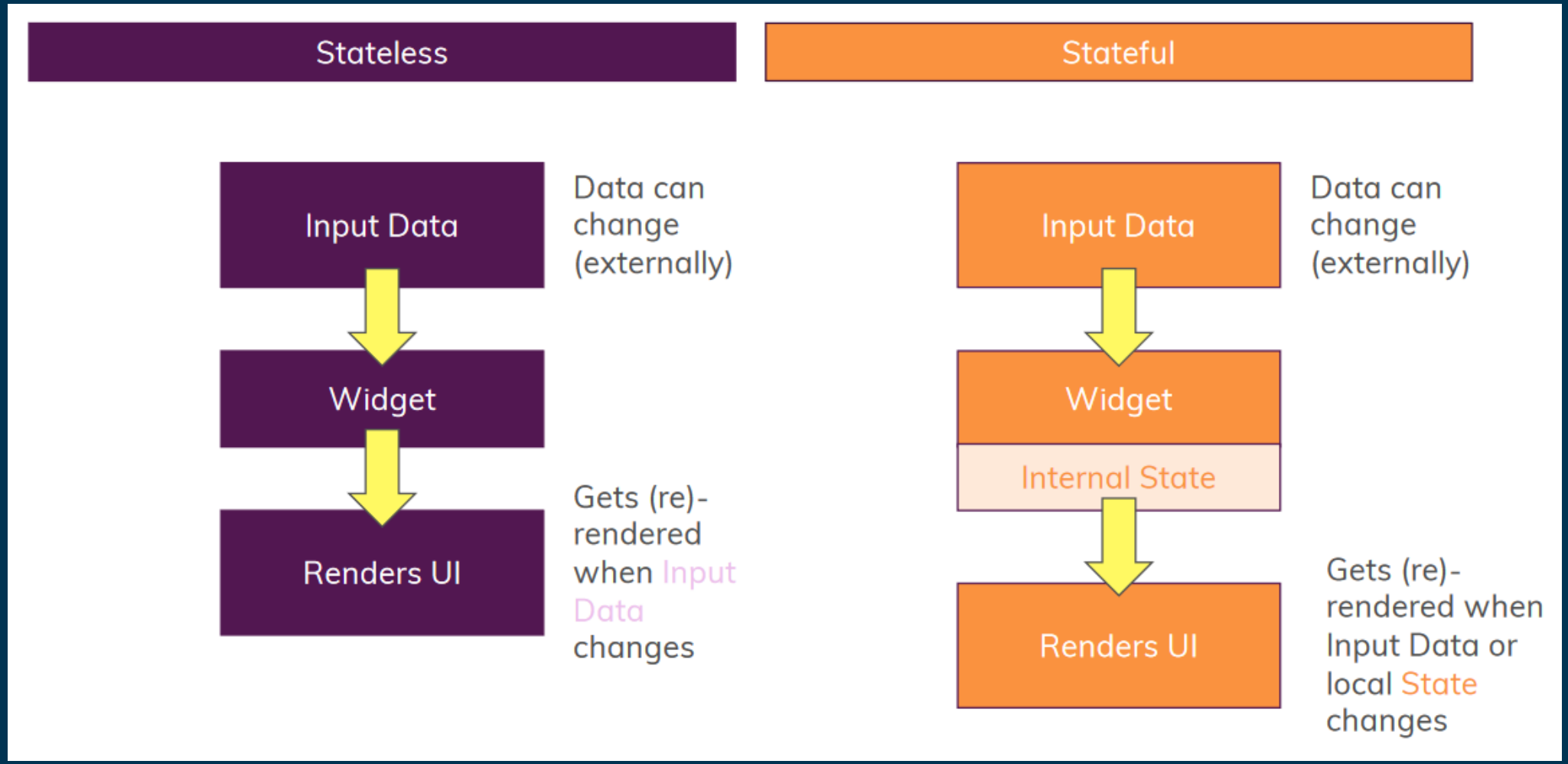
Ερωτήσεις

- Ποιος είναι ο ρόλος της runApp()
  - Η συνάρτηση δημιουργεί το κεντρικό widget και καλεί την build συνάρτηση του widget
- Τι κάνει η build() μέθοδος;
  - Επιστρέφει το widget tree που θα πρέπει να εμφανιστεί στην οθόνη του κινητού
- Τι είναι τα Widgets;
  - Είναι τα βασικά δομικά στοιχεία του Flutter για την δημιουργία του user interface

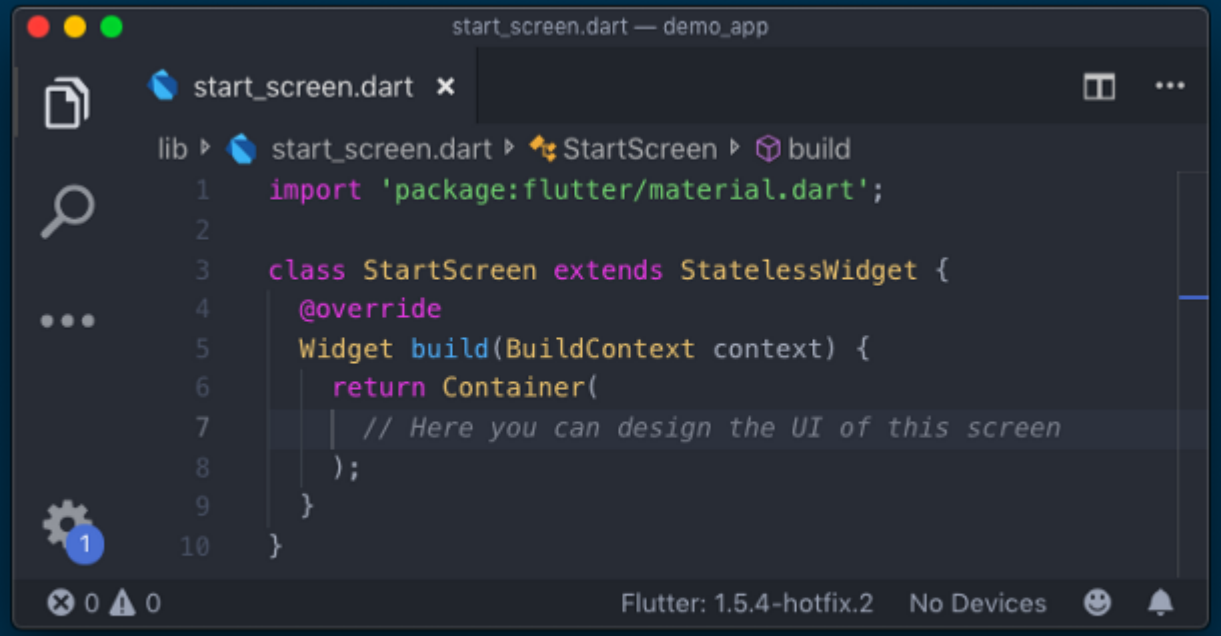
Ερωτήσεις

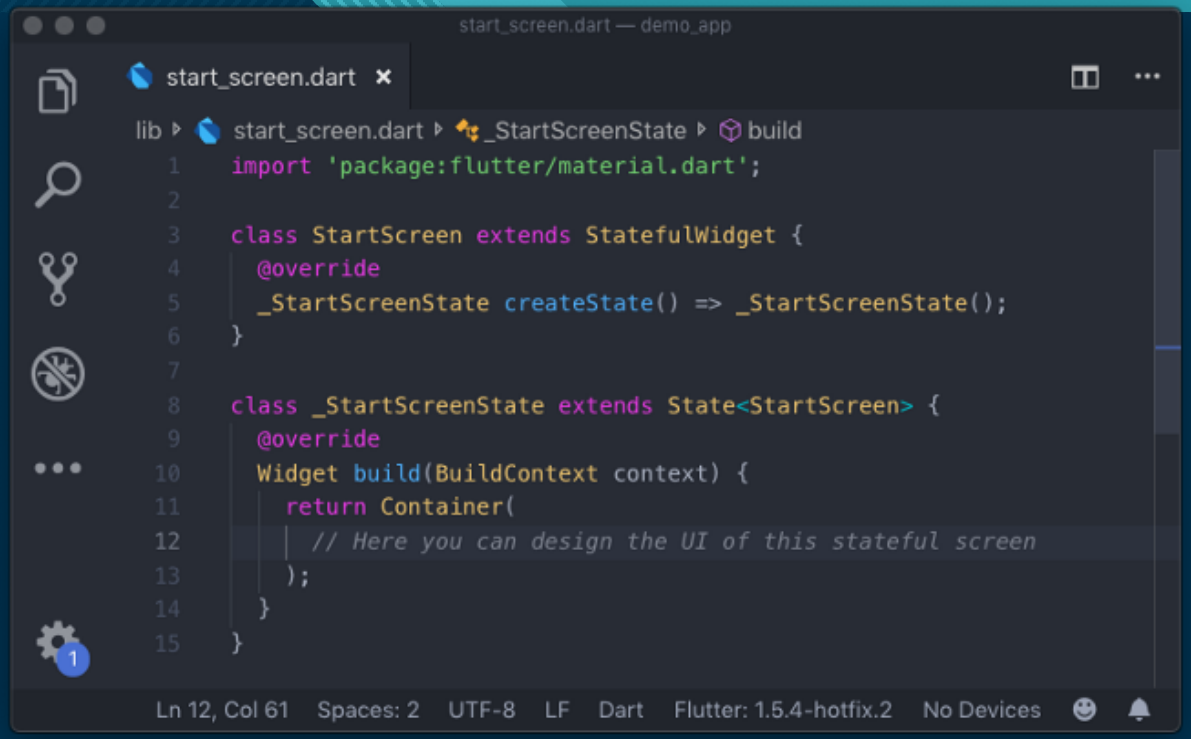
- Ποιος είναι ο ρόλος ενός Widget;
  - Η διεπαφής χρήστη(UI) της εφαρμογής σας δημιουργείται από Widgets.
- Τι περιγράφει ο όρος "Widget tree";
  - Το "Widget tree" σημαίνει ότι δημιουργείτε το περιβάλλον χρήστη σας με έναν συνδυασμό (nested) Widget.
- Γιατί χρειαζόμαστε τα Widgets;
  - Τα widget είναι το βασικό δομικό στοιχείο με το οποίο δημιουργείτε πλούσιες διεπαφές χρήστη στο Flutter.
- Πώς συνδυάζονται τα Widgets;
  - Περνάμε Widgets στους constructors άλλων Widgets ώστε να επιτύχουμε το επιθυμητό αποτέλεσμα.
- Τι κάνει ένα αντικείμενο Dart να γίνει ένα Widget;
  - Extending StatelessWidget / StatefulWidget και η δημιουργία μιας build() μεθόδου.

Stateless vs Stateful



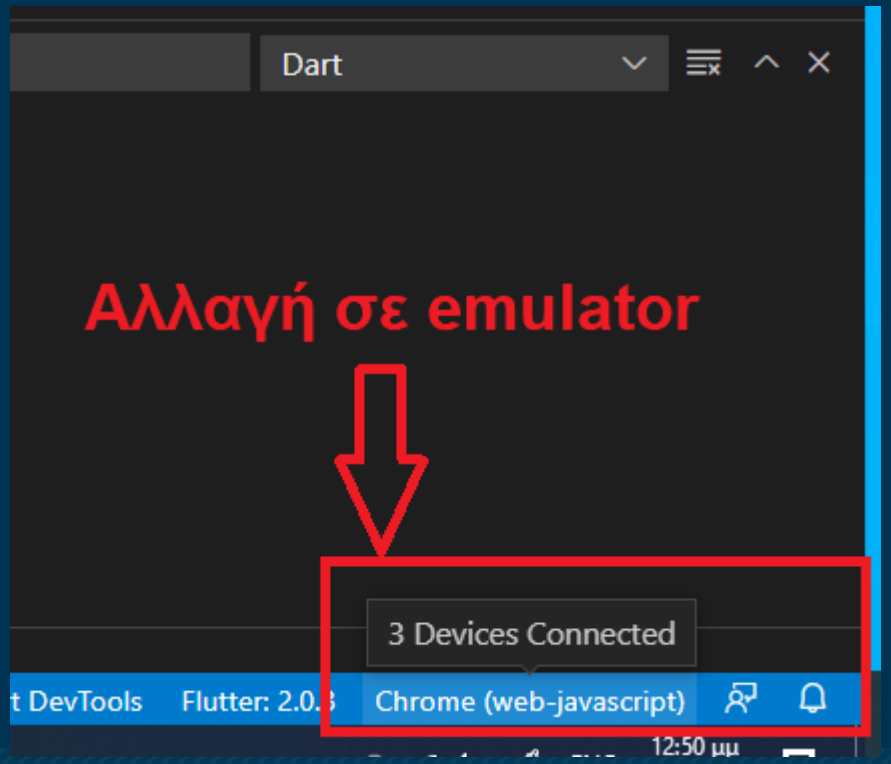
Stateless vs Stateful



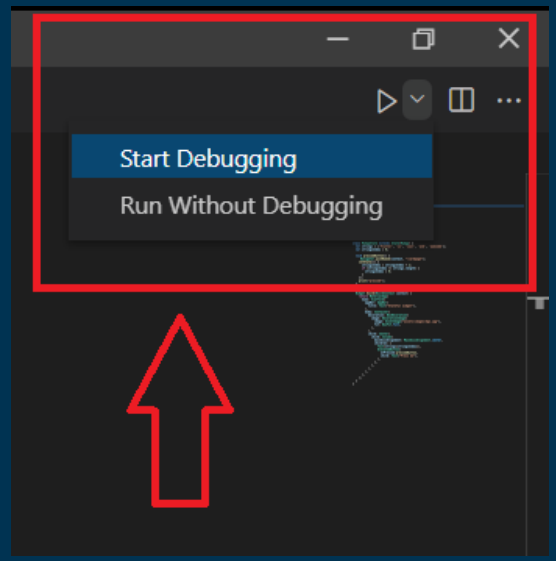


Run project

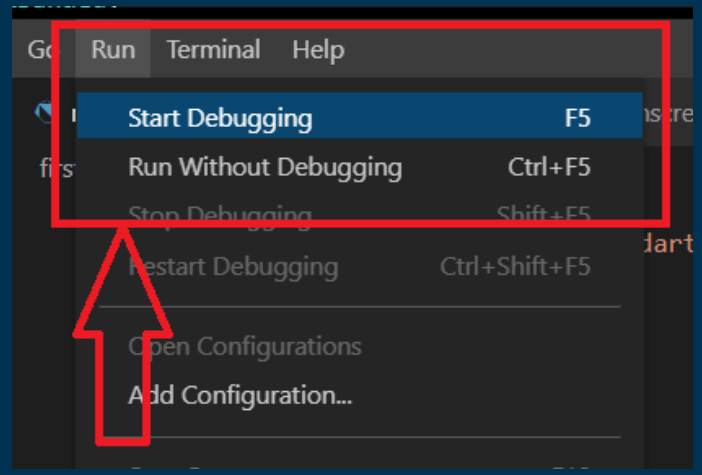
- 1 βήμα – Επιλογή emulator



- 2 βήμα – Run code



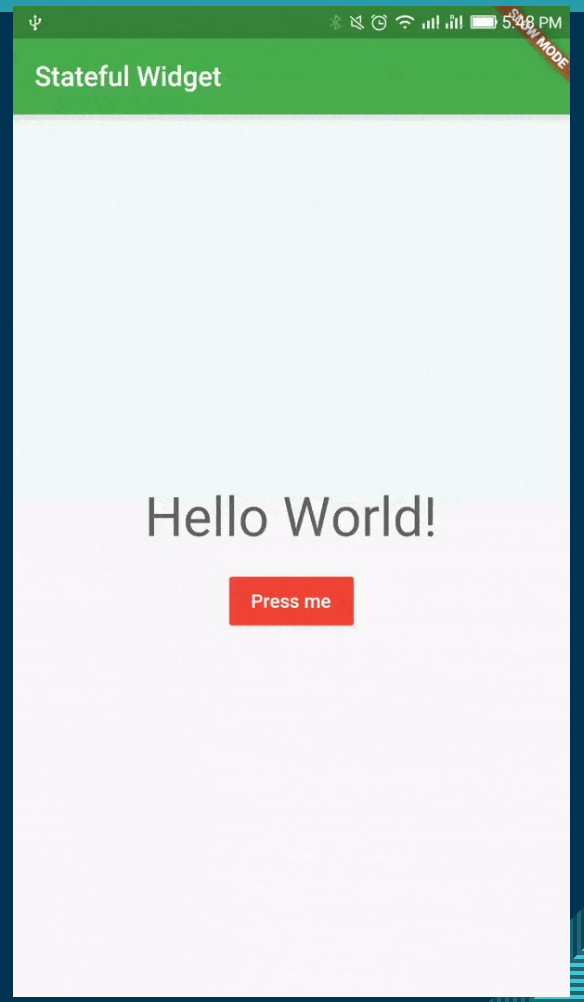
ή



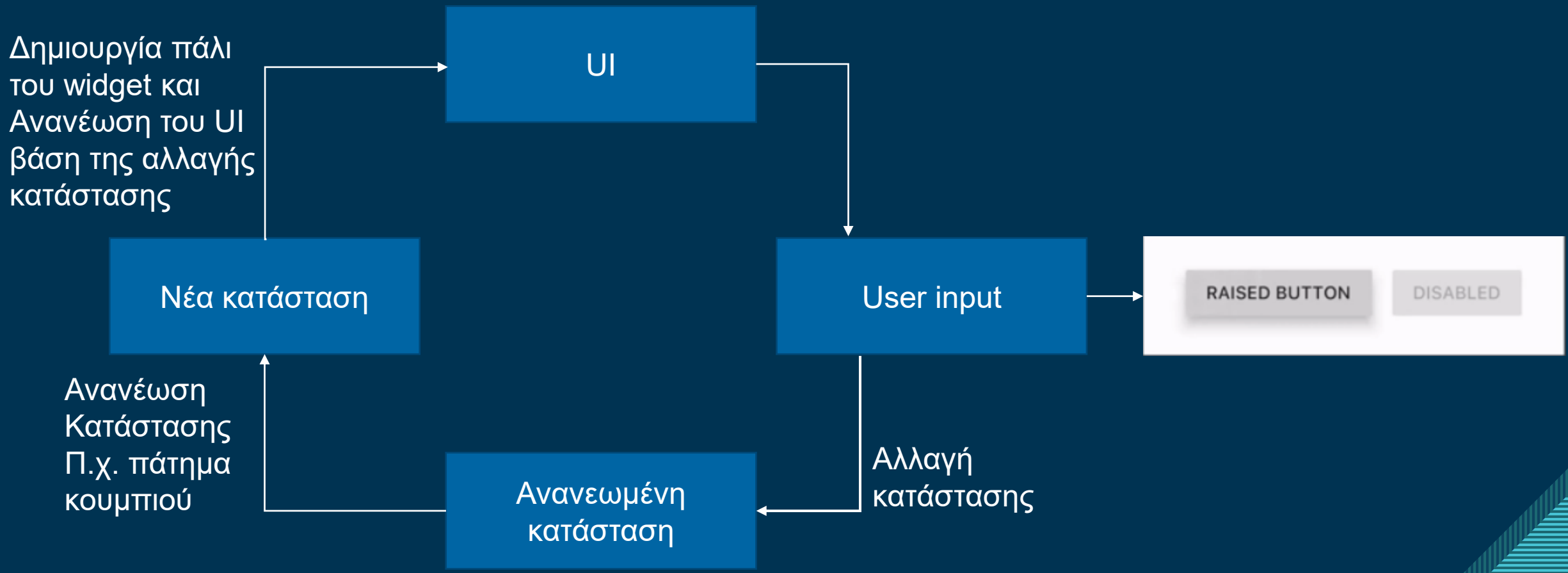
Παραδείγματα Flutter

```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



Stateless vs Stateful

- Τα Stateless Widgets δεν έχουν μέθοδο για να ξανασχεδιάσουν (re-render) τους εαυτούς τους αν τα εσωτερικά τους δεδομένα αλλάξουν
- Τα Stateless Widgets δεν έχουν εσωτερική κατάσταση (internal state) όπως φαίνεται και από το όνομα τους.
- Εάν ένα Stateless Widget ξανασχεδιαστεί, τότε ολόκληρο το widget σχεδιάζεται εκ νέου
- Τα Stateful widgets μπορούν να ξανασχεδιαστούν όταν τα εσωτερικά ή εξωτερικά τους δεδομένα αλλάξουν
- Η μέθοδος setState () στα Stateful widgets ξανασχεδιάζει εκ νέου στοιχεία που σχετίζονται μόνο με την αλλαγή κατάστασης. Δεν ξανασχεδιάζει ξανά ολόκληρο το widget αλλά μόνο τα απαραίτητα μέρη του.

Stateful Widgets

- https://flutter.io/tutorials/interactive/#stateful-stateless
- **setState()** είναι μια ειδική συνάρτηση που είναι διαθέσιμη μόνο στα Stateful Widgets η οποία ανανεώνει την κατάσταση του widget και ξανασχεδιάζει τα στοιχεία που επηρεάζονται από την αλλαγή κατάστασης.
- Ορίζουμε τα δεδομένα που θέλουμε να ανανεωθούν μέσα στο setState() **[εμείς χρησιμοποιήσαμε το stringIndex]**
- Στο παράδειγμα μας μόνο το Text() θα ξανασχεδιαστεί γιατί μόνο αυτό το widget αλληλοεπιδρά με την αλλαγή τιμής string**Index**.

Ερωτήσεις

- Ποια είναι η διαφορά μεταξύ StatelessWidget και StatefulWidget
  - Το StatelessWidget δεν μπορεί να ξανατρέξει την build() μέθοδο όσο τρέχει η εφαρμογή
- Γιατί πρέπει να καλέσετε το setState (() {...}) σε ένα StatefulWidget (κατά την αλλαγή ορισμένων δεδομένων);
  - Χωρίς το setState το widget δεν θα ξανατρέξει την build() οπότε οι αλλαγές δεν θα φανούν στην οθόνη.

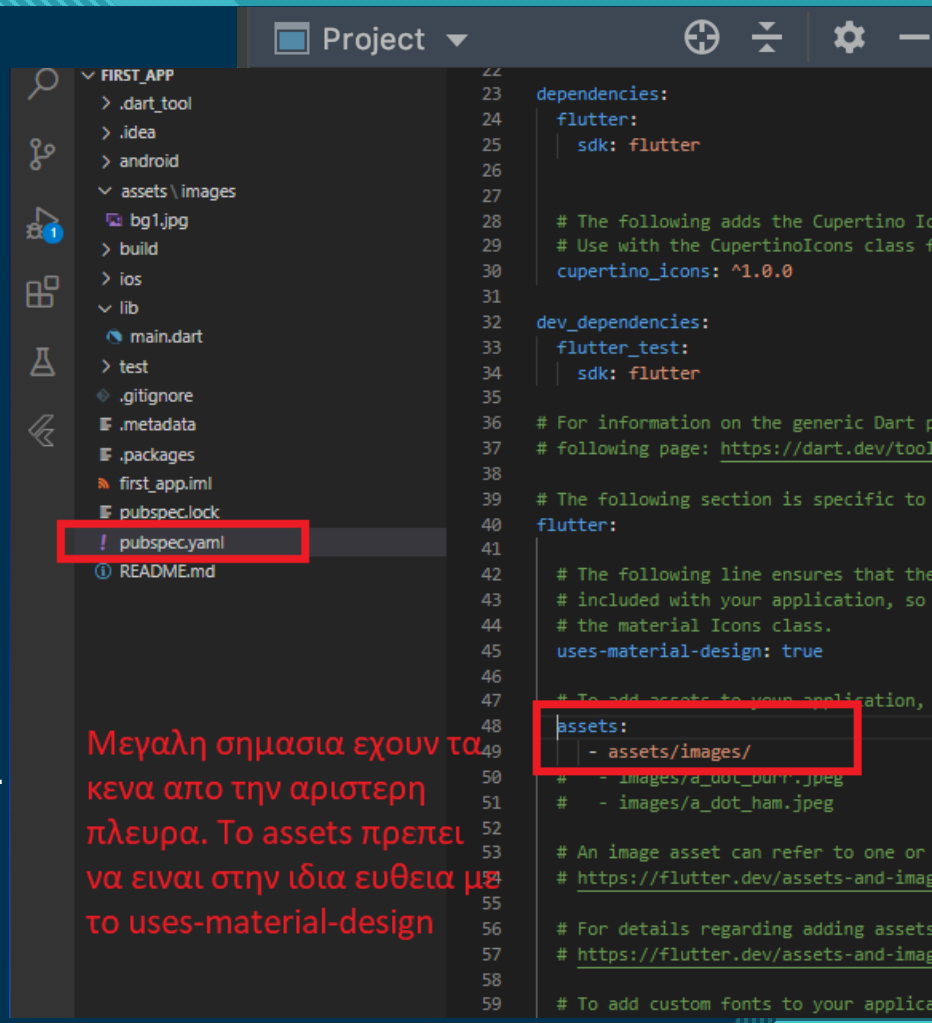
Flutter παραδείγματα

Εισαγωγή Background εικόνας

1. Δημιουργία φακέλου assets/images
   - Χρησιμοποιείται ότι ονόματα θέλετε
2. Αποθηκεύστε την εικόνα στον φάκελο images
3. Δηλώστε τον assets στο pubspec.yaml
   - Ανοίγω το αρχείο pubspec.yaml
   - Συμπληρώστε μια assets υποκατηγορία στην κατηγορία flutter με αυτό τον τρόπο:

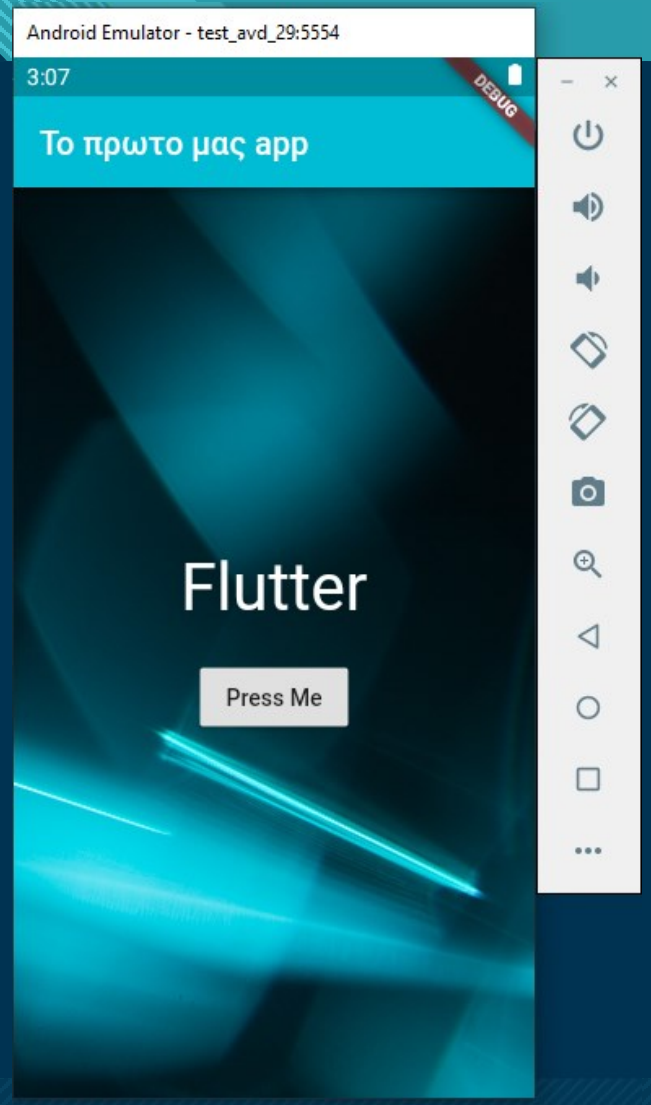
```
flutter:
  assets:
    - assets/images/
```

   Προσοχή! Έχουν σημασία τα κενά από την αριστερή πλευρά μέχρι το όνομα της υποκατηγορίας (πχ. assets) που γράφετε σε αυτό το αρχείο.
4. Εισάγετε την εικόνα στο main.dart
   - Image.asset('assets/images/bg1.jpg')



Background Example

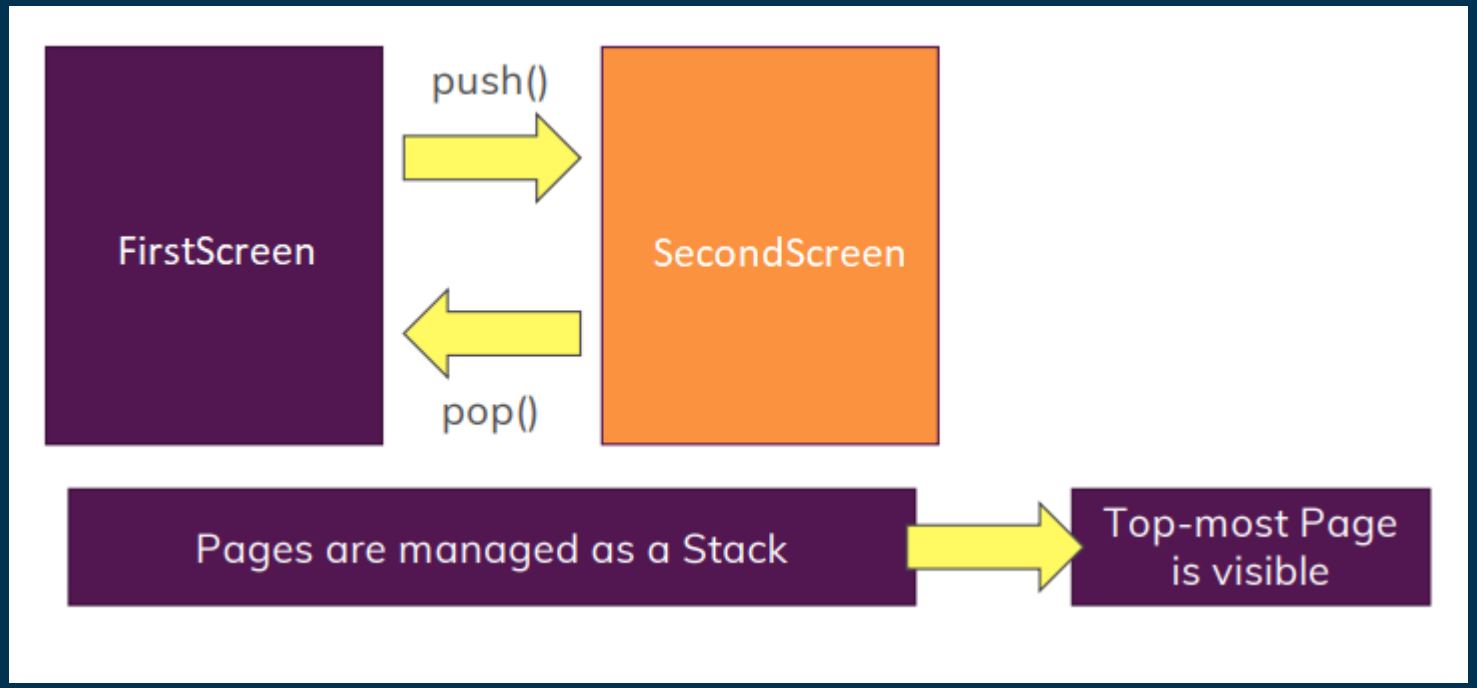
```dart
@override
Widget build(BuildContext context) {
  return MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: const Text('Stateful widget'),
        backgroundColor: Colors.cyan,
      ), // AppBar
      body: Container(
        decoration: const BoxDecoration(
          image: 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
```

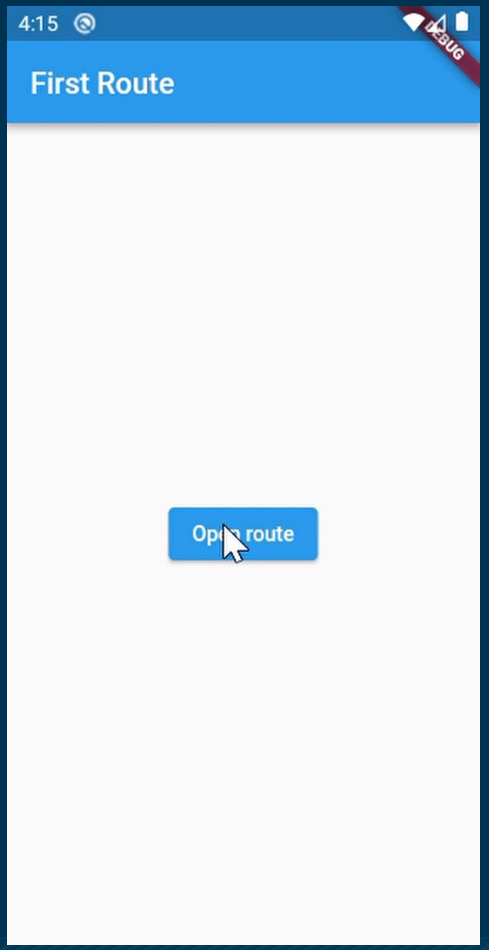


Flutter Navigation

- Χρησιμοποιούμε την κλάση Navigator
  - Η Navigator κάνει την μεταφορά από την μια οθόνη στην άλλη
  - Είναι συνδεδεμένη με την μεταβλητή context
  - Η context μεταβλητή εμπεριέχει πληροφορίες για την κεντρική κλάση widget που καλείται και σε ποιο σημείο είναι στο Widget Tree
- Οι οθόνες διαχειρίζονται σαν ένα stack
- Top-most page is visible

Flutter Navigation





Κώδικας Navigation

- 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 κραταει 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

- Στο Flutter, οι οθόνες και οι σελίδες ονομάζονται διαδρομές (routes).
  - Το “route” είναι ένα “abstraction” για μια "οθόνη" ή "σελίδα" μιας εφαρμογής
- Το Navigator είναι ένα widget που διαχειρίζεται διαδρομές (routes)

Κλάση Map

- Το Map είναι μια κλάση της dart
- Χρησιμοποιεί key-value δομή
- Τη δηλώνουμε με τις { }
- Παράδειγμα:

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

Κώδικας Navigation με 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

- Διαφορετικοί τρόποι προώθησης σε νέα σελίδα
  1. Με χρήση δυναμικού ονόματος σελίδας που έχει δηλωθεί στη main.dart (τα named routes)
     - Navigator.pushNamed(context, routePage);
  2. Με αρχικοποίηση του Widget που θέλουμε να μας κατευθύνει το πρόγραμμα
     - Navigator.of(context).push(MaterialPageRoute(builder:(context)=>CardPage()));
- Αφαίρεση της σελίδας που καλούμε τον Navigator από το Stack (από την λίστα με τις διαδρομές)
  - Navigator.of(context).pushReplacement(MaterialPageRoute(builder:(context)=>CardPage()));
  - έτσι όταν πατήσουμε επιστροφή από την καινούρια σελίδα (στο παράδειγμα μας την CardPage) πάει στην προηγουμένη σελίδα από το homepage δηλαδή στην SplashScreen
- Αφαίρεση της σελίδας που καλούμε τον Navigator από το Stack με named routes
  - Navigator.of(context).pushReplacementNamed(routePage);

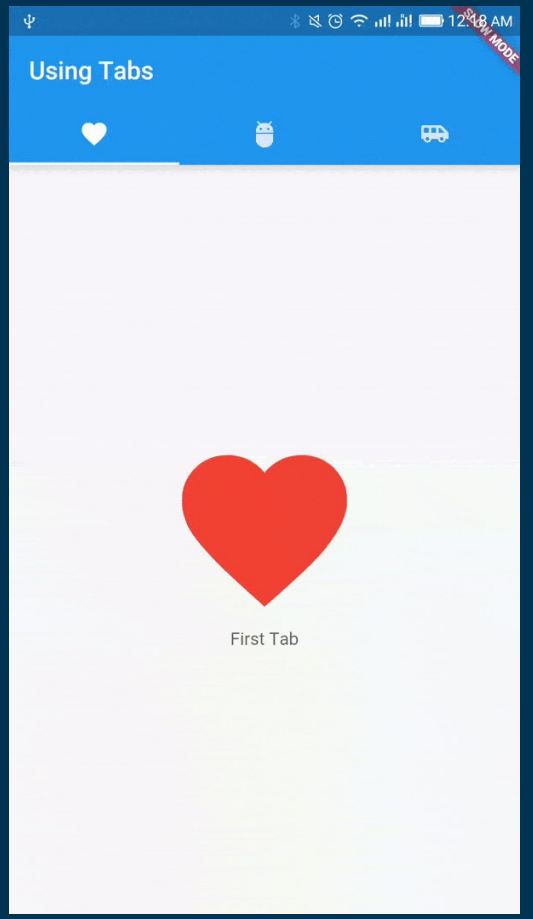
Tabs example

- 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
  }
}
```

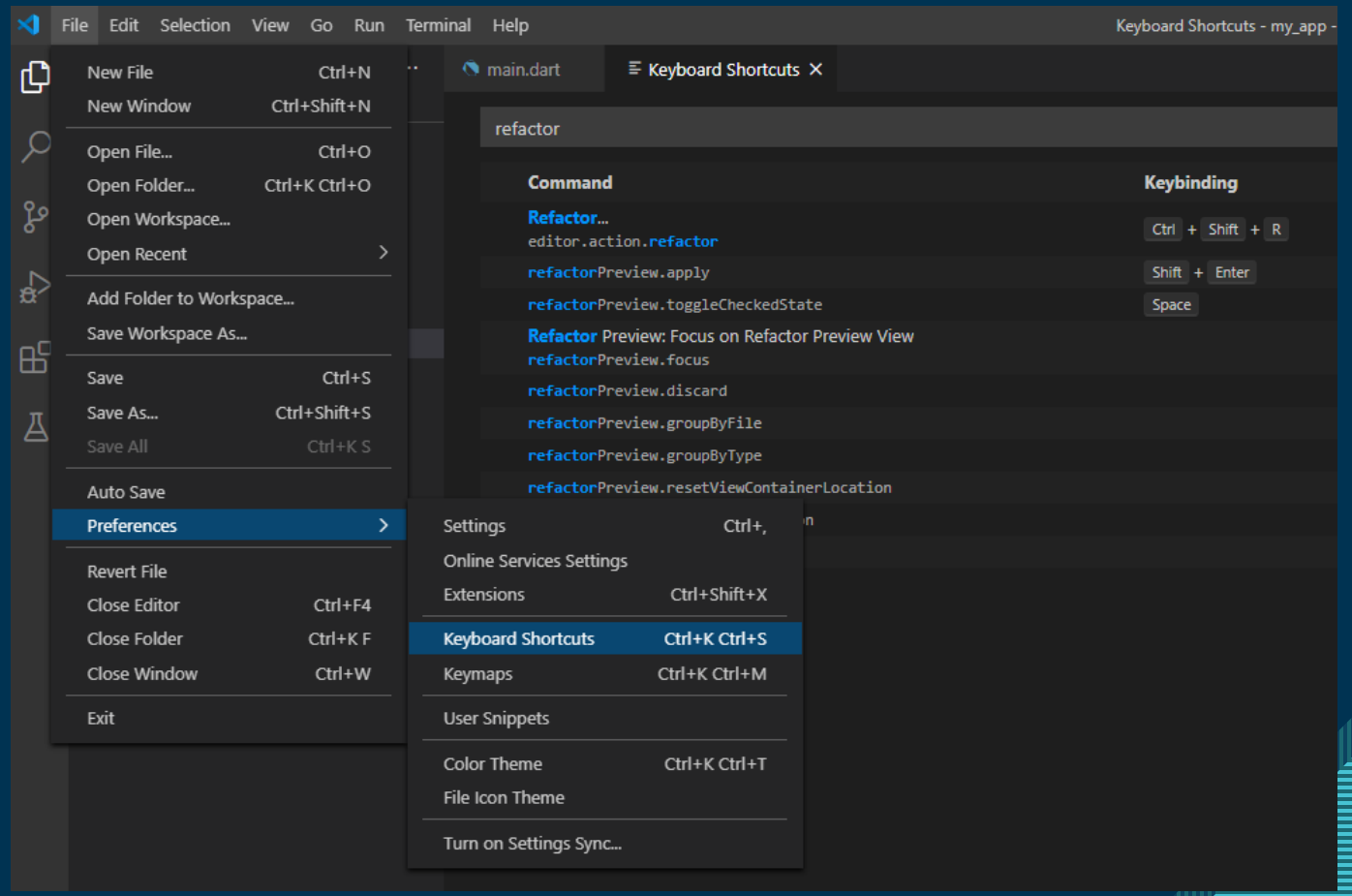
First.dart

```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

- Κάθε αρχείο στην Dart μπορούμε να το θεωρήσουμε ως κλειστό οικοσύστημα
- Διαφορετικά αρχεία μπορούν να δουλέψουν μεταξύ τους μέσω του import
- Αν προσθέσουμε ‘_’ μπροστά από κλάση, συνάρτηση και μεταβλητή μετατρέπονται σε private και μπορούν να χρησιμοποιηθούν μόνο μέσα στο αρχείο

```
8   class MyApp extends StatefulWidget {
9     @override
10    State<StatefulWidget> createState() {
11      return _MyAppState();
12    }
13  }
14
15  class _MyAppState extends State<MyApp> {
16    var _stringIndex = 0;
17
18    void _onPressedButton() {
19      //stringIndex++;
20      setState(() {
21        _stringIndex = _stringIndex + 1;
22        if (_stringIndex > 4) {
23          _stringIndex = 0;
24        }
25      });
26
27      print(_stringIndex);
28      //print('Button pressed!');
29    }
30
```