



Βιοϊατρική Πληροφορική

Κουμάκης Λευτέρης, Μαρία Χατζημηνά

Εισαγωγή στη Dart-Flutter

Dart

- Αντικειμενοστραφής γλώσσα
- Δημιουργήθηκε από την Google
- Παρόμοια με :
 - Swift
 - C#
 - Java
 - JavaScript
- Just-in-time compilation
 - Re-compile του κώδικα την ώρα του προγραμματισμού της εφαρμογής
- Ahead-of-time compilation
 - Ο κώδικας μεταγλωττίζεται σε native ARM κώδικα



Dart Resources

- <https://dart.dev/guides/language/effective-dart>
- <https://dart.dev/guides/language/language-tour>
- <https://flutter.dev/docs/resources/faq>
- Dart Basics: αρχείο στον φάκελο του εργαστηρίου

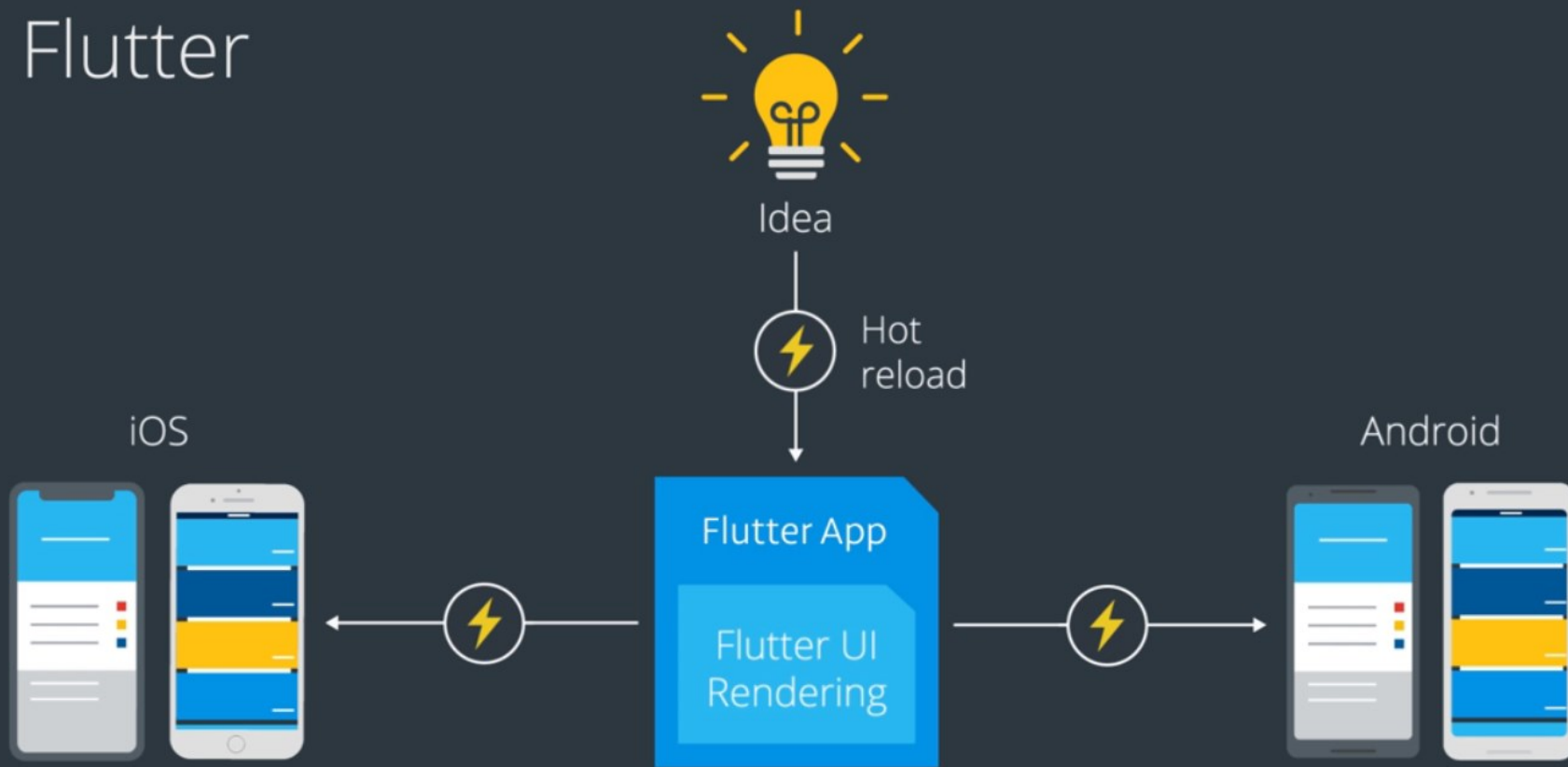
Τι είναι το Flutter;

- Το Flutter είναι framework της Google
- Γρήγορη και εύκολη δημιουργία εφαρμογών iOS και android με τον ίδιο κώδικα
- Δωρεάν
- Ανοιχτού κώδικα (open source)
- Στα 50 πιο ενεργά repositories στο GitHub
- Πληθώρα έτοιμων widget

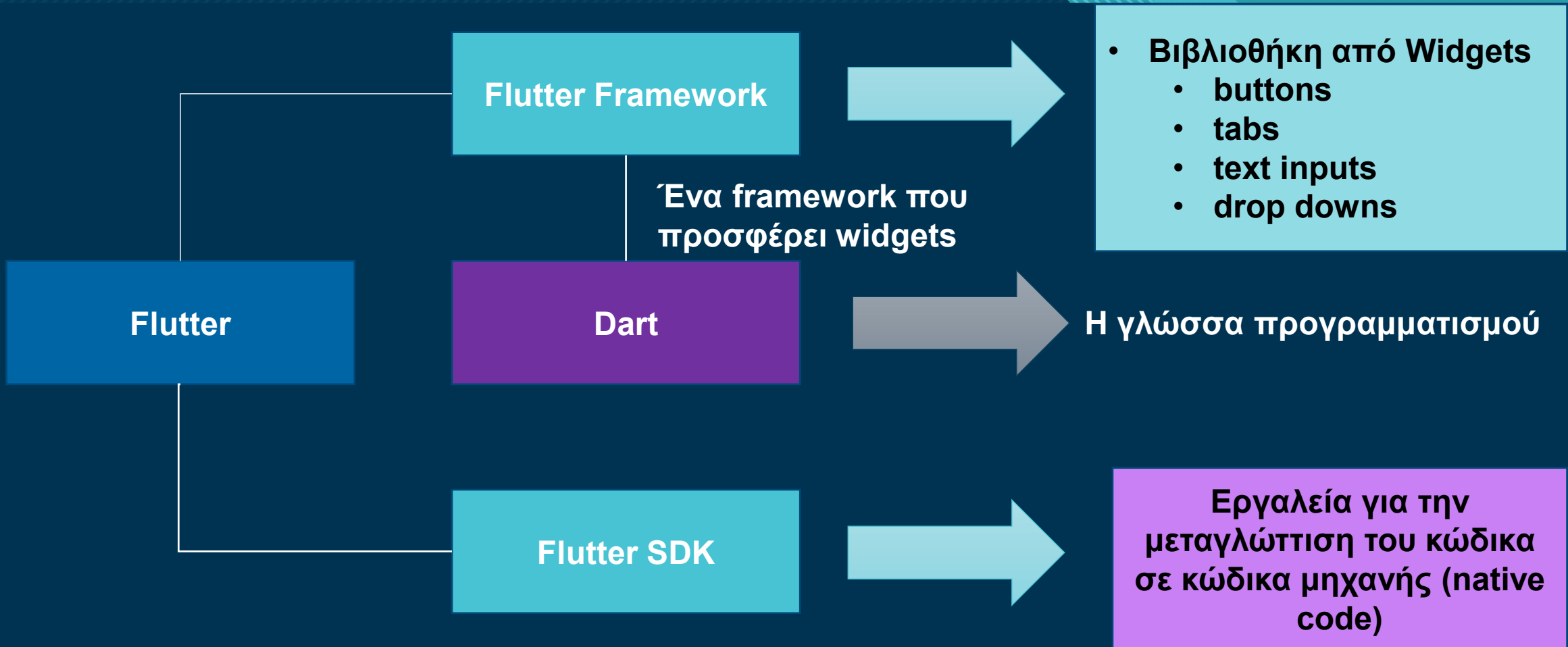


Flutter

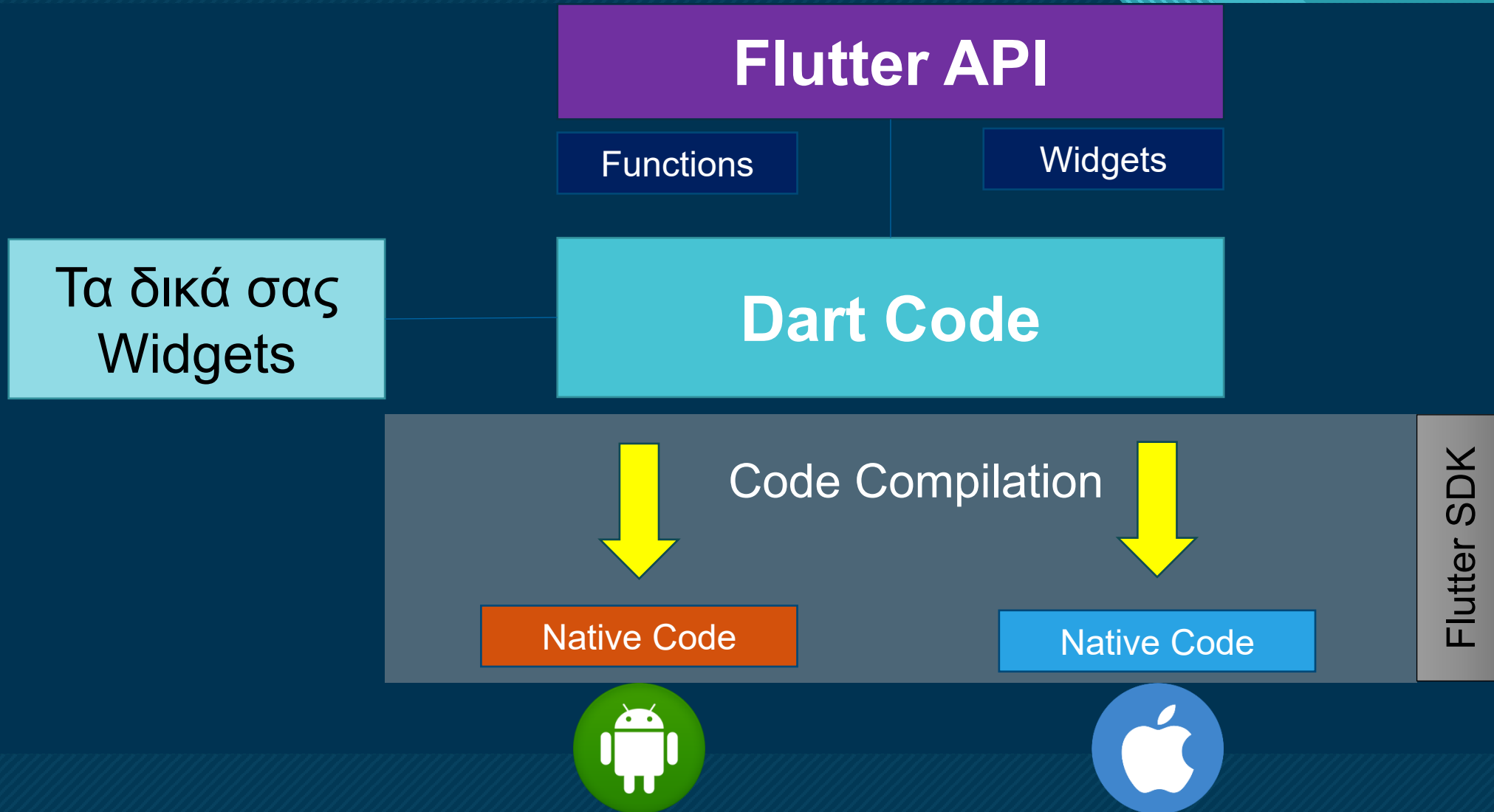
Flutter



Flutter vs Dart



Μετατροπή Flutter/Dart σε Native Application



Flutter Widgets

- Το widget είναι η περιγραφή ενός μέρους της διεπαφής χρήστη (UI)
- Τα widgets θεωρούνται η βάση του Flutter
- Κάθε εφαρμογή Flutter είναι ένα σύνολο widgets
- Στο Flutter η εφαρμογή δημιουργείται σαν μια δενδροειδής μορφή από widgets
- Δεν υπάρχουν χωριστά αρχεία για την προσαρμογή του layout
 - Όλος ο κώδικας και οι ρυθμίσεις ενός widget ορίζονται μέσα στο widget

Παράδειγμα widget

Elevated Button Widget

- `textTheme`
- `color`
- `onPressed`
- `shape`
- `etc.`

Widget

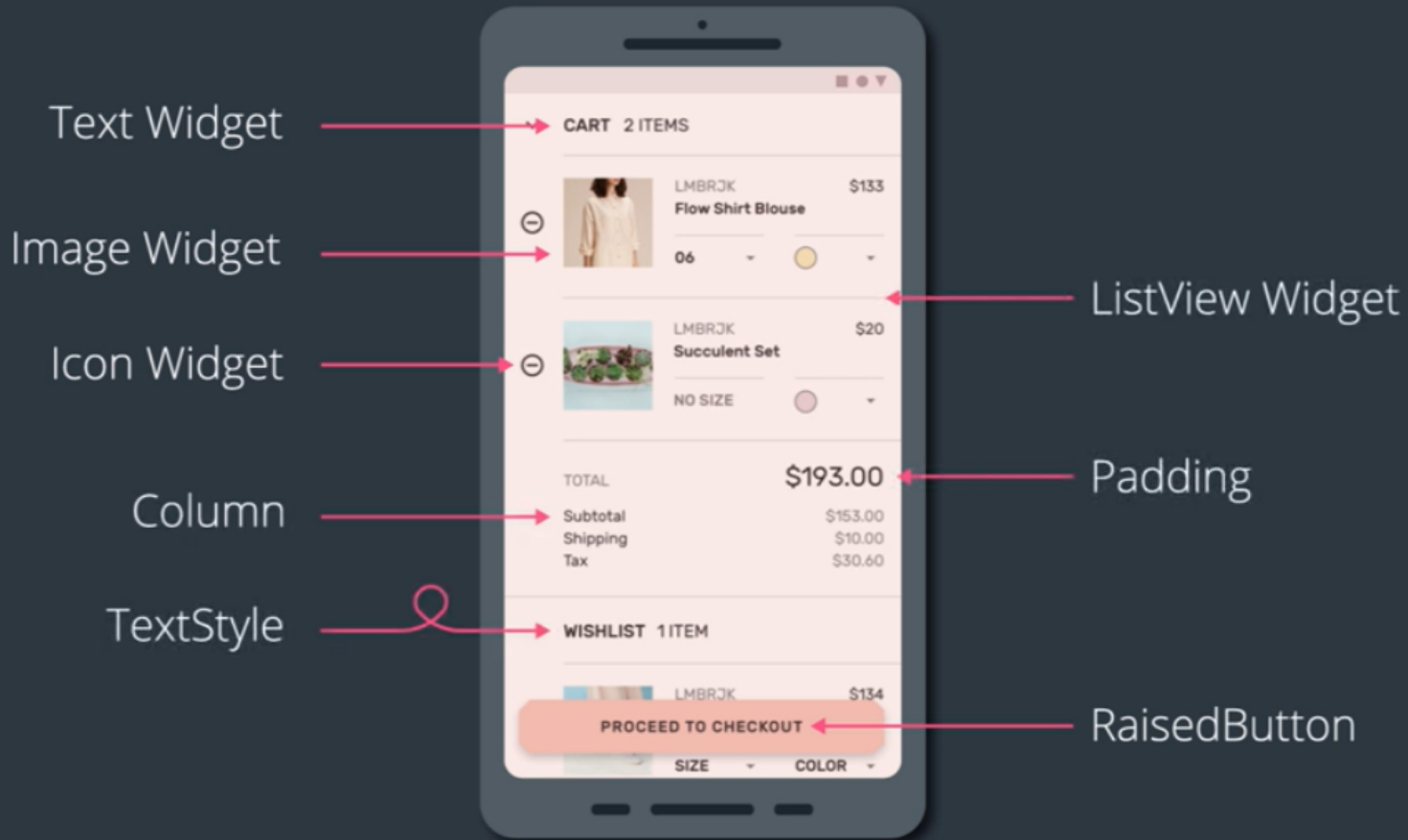


Elevated button



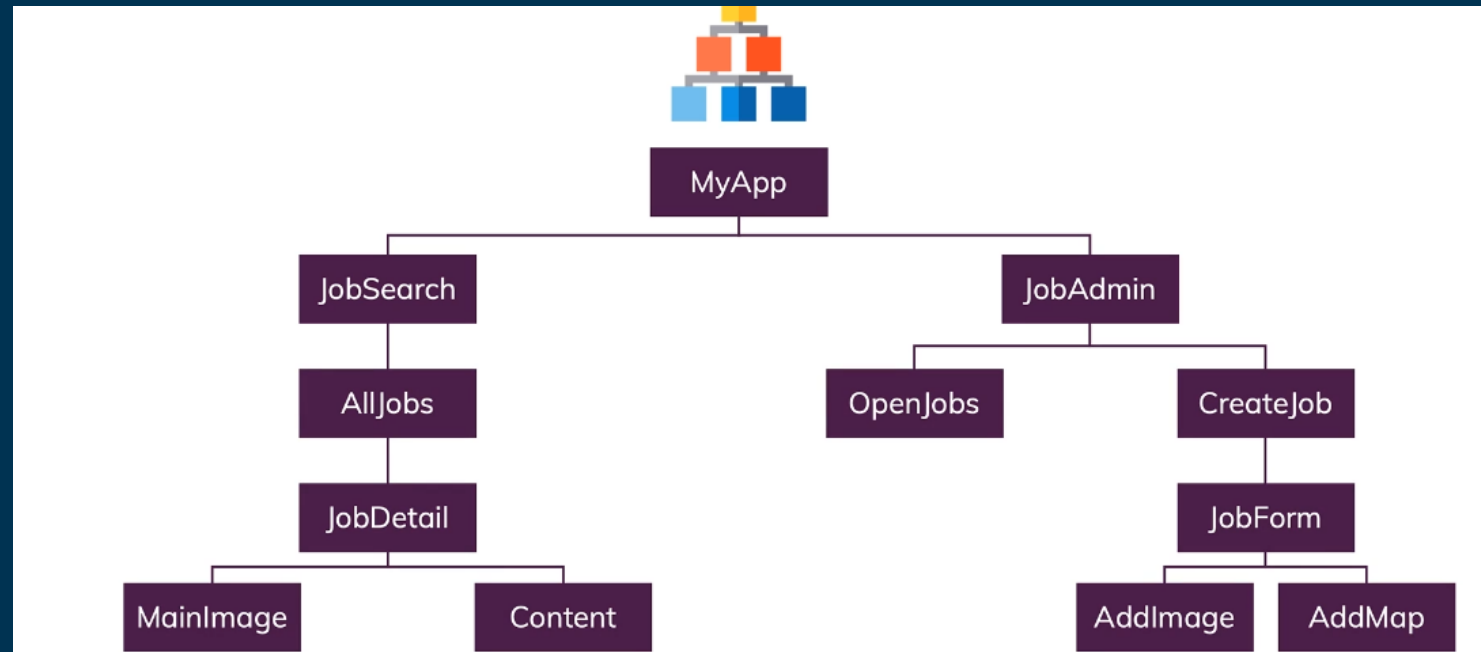
Disabled button

Widgets



Αρχιτεκτονική Flutter

- Δημιουργία widget tree για τις εφαρμογές
- Δεν υπάρχει drag-and-drop editor για εισαγωγή κουμπιών, κειμένου κτλ.



Κατηγορίες Widget

Output & Input (Visible)

RaisedButton(), Text(), Card(), ...

Drawn onto the screen: "What the user sees"

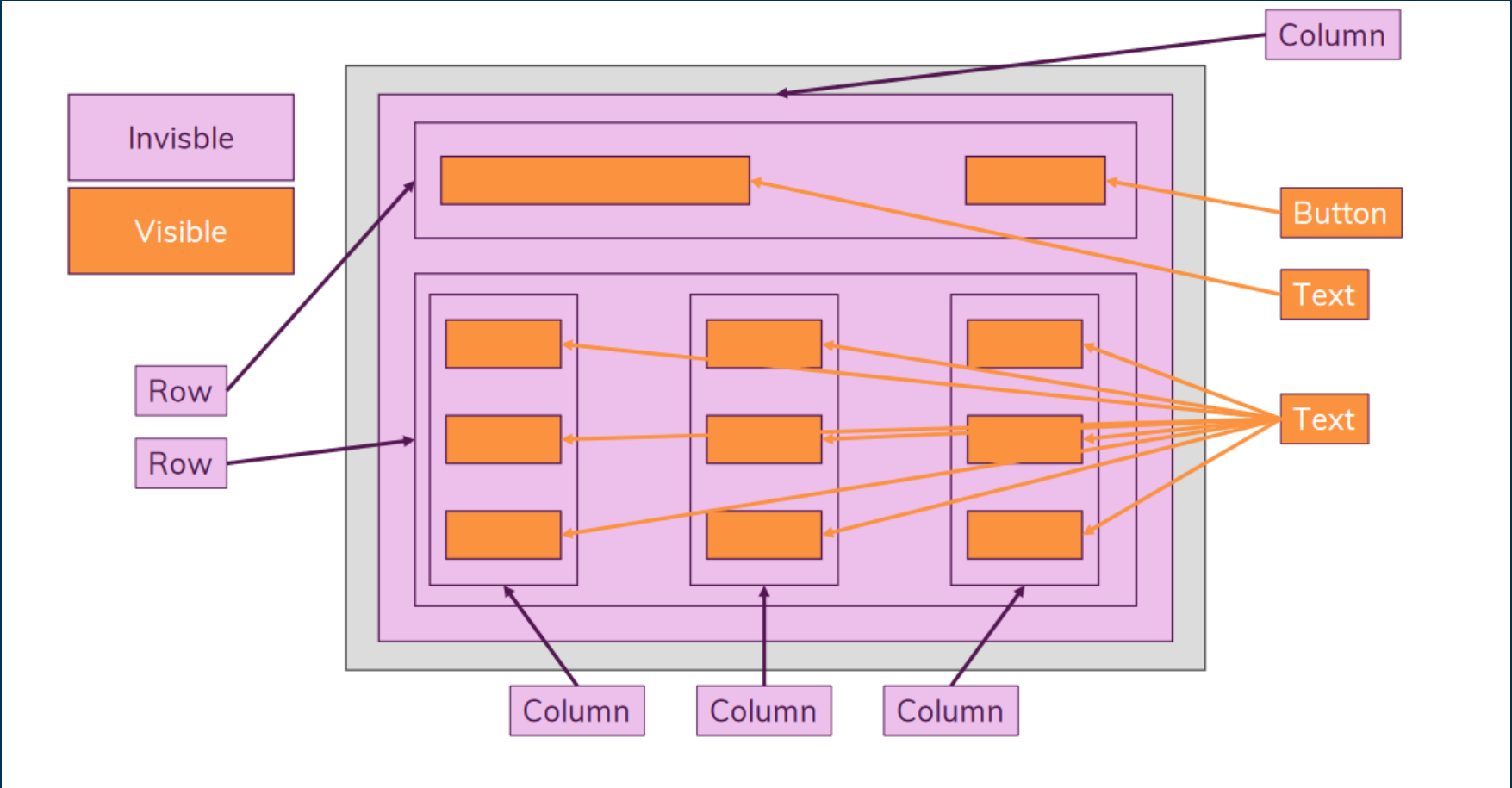
Layout & Control (Invisible)

Row(), Column(), ListView(), ...

Give your app structure and control how visible widgets are drawn onto the screen (incirectly visible)

Container()

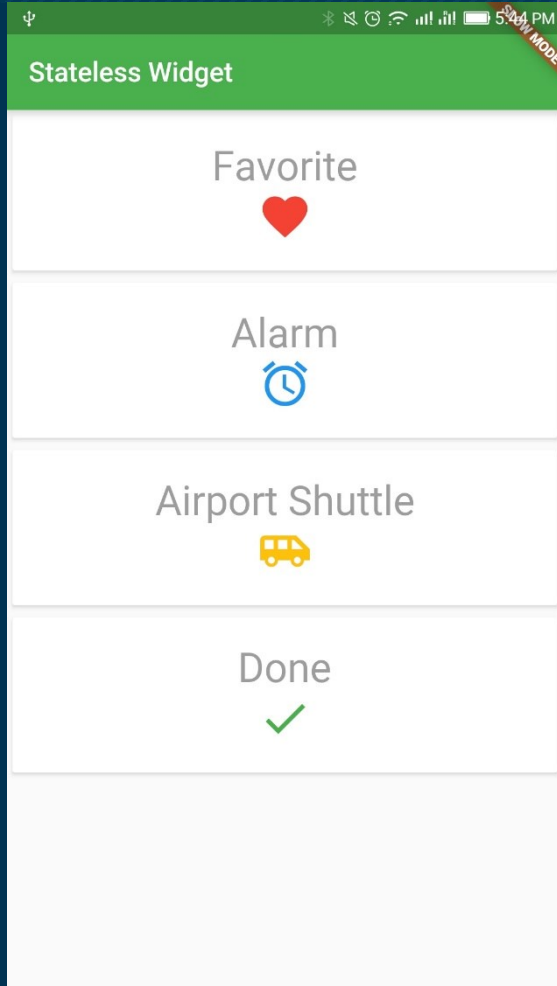
Εμφανή και μη-εμφανή widget



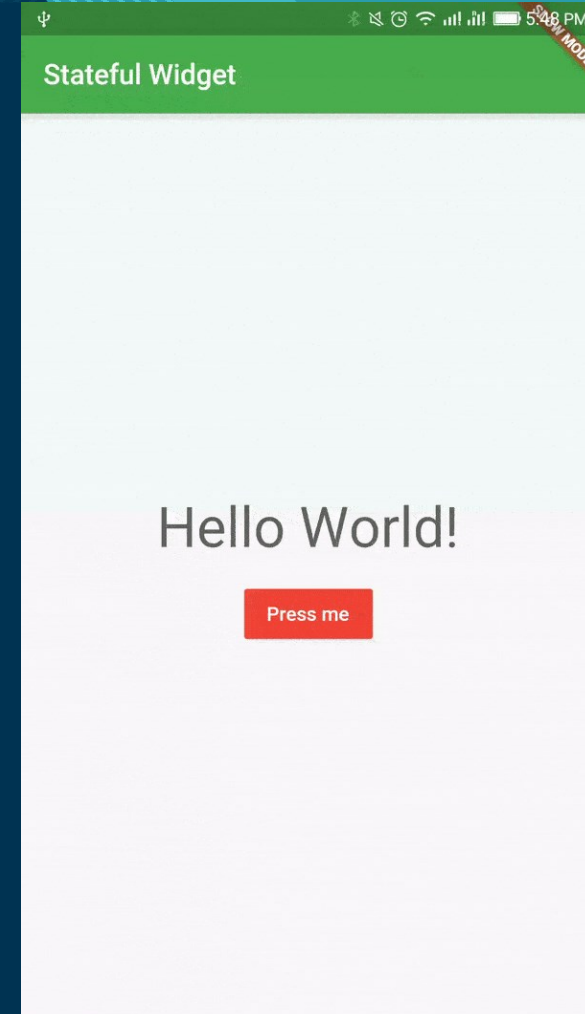
StatelessWidget και StatefulWidget

- StatelessWidget
 - Ένα αμετάβλητο widget. Όλα του τα πεδία είναι τετελεσμένα μετά την δημιουργία του
 - Ρυθμίσεις κατά την αρχικοποίηση
 - Χρώμα
 - Ύψος
 - Πλάτος
- StatefulWidget
 - Ένα widget που δημιουργεί μια κατάσταση και είναι ευμετάβλητο

Stateless vs Stateful

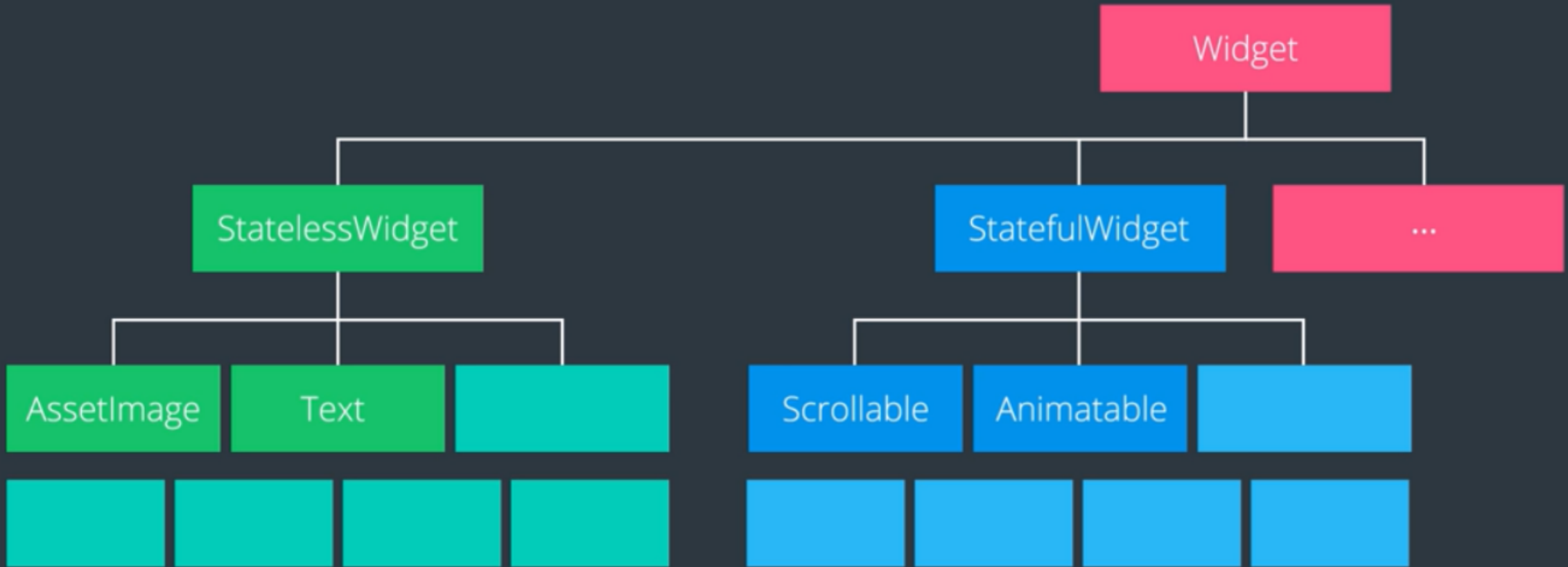


Stateless



Stateful

Widgets



Useful Flutter Dev Tools

- Βοηθάνε στην ανάπτυξη και τον εντοπισμό σφαλμάτων
- Hot reload
 - Το UI ανανεώνεται αμέσως όταν γίνεται κάποια αλλαγή (just-in compiler)
- Widget-inspector
 - Παρόμοιο με την κονσόλα του Chrome/Firefox
- Code auto-formatter
 - Reformat Code with dartfmt

Dart Basics

- Dart pad: <https://dartpad.dev/>
- Στις ονομαστικές παραμέτρους αν δεν επιτρέπεται να είναι null η τιμή τους (*null safety*) προσθέτουμε την λέξη “required” ή δίνουμε default τιμή

```
DartPad <> New Pad ⌂ Reset ≡ Format ⬇ Install SDK withered-su
```

```
class Person {  
    String name = 'Maria';  
    int age = 34;  
}  
  
double addNumbers( double num1,double num2){  
    //print(num1 + num2);  
    return num1+num2;  
}  
  
void main() {  
    var p1 = Person();  
    print(p1);  
    print(p1.age);  
    var p2 = Person();  
    p2.name= 'Stella';  
    print(p2.name);  
    print(addNumbers(1,2.6));  
    var firstResult = addNumbers(1,2);  
    print (firstResult);  
    print (firstResult+1);  
    print("hello");  
}
```

▶ RUN

```
Console  
Instance of 'Person'  
34  
Stella  
3.6  
3  
4  
hello
```

Documentation

```
DartPad <> New Pad ⌂ Reset ≡ Format ⬇ Install SDK
```

```
1 class Person {  
2     var name;  
3     var age;  
4     Person({String inputName="Not null value", required int age}){  
5         name=inputName;  
6         this.age=age;  
7     }  
8 }  
9  
10 void main() {  
11  
12     // var p1 = Person('Maria',34); Ετσι θα καλουσαμε τον constructor  
13     //χωρις ονομαστικες μεταβλητες. Αν ελειπαν τα {}  
14     var p1 = Person(inputName:"Maria",age:35);  
15     print(p1.name);  
16     var p2 = Person(inputName:"Elsa"); //με τις ονομαστικες μεταβλητες  
17     //δεν χρειαζεται να παρουν τιμη ολες οι μεταβλητες  
18  
19 }  
20
```

Null safety Flutter

- Θα πρέπει να εγκαταστήσετε την έκδοση **flutter/dart 3.24.0** και μετα
- Εργασίες με κώδικα παλιότερων εκδόσεων και Null safety προβλήματα θα θεωρηθεί **αντιγραφή και δεν θα διορθωθεί**

Εγκατάσταση VS Code χωρίς Android studio + Emulator

- Αναλυτικές οδηγίες:
 - <https://abp.io/docs/latest/framework/ui/react-native/setting-up-android-emulator>
- Online Flutter Compiler:
 - <https://flutlab.io/>

Installation

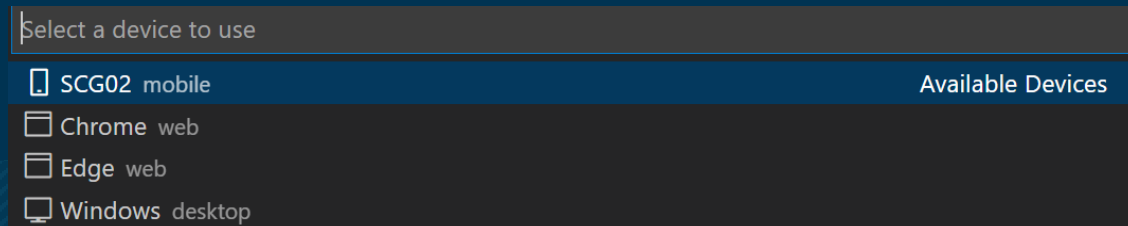
- Install Visual Studio Code <https://code.visualstudio.com/>
- Install flutter <https://docs.flutter.dev/get-started/editor?tab=vscode>
- Emulator: [Bluestacks](#)
- Connect Bluestacks emulator with VSCode (next slide)

Connect BlueStacks emulator VSCode

- In bluestacks emulator select Settings → Advanced
- Preferences : Enable Android Debug Bridge
- Port Number(in this example is 51614) but every time changes
- Write in command line in “platform-tools” path*:

```
C:\Users\admin\AppData\Local\Android\platform-tools>adb connect localhost:51614  
connected to localhost:51614
```

- The emulator will be shown in VSCode



Settings

Performance

Display

Graphics

Audio

Gamepad

Preferences

Device

Shortcuts

User data

Advanced

About

Application Binary Interface (ABI) ?

x86 32-bit, x86 64-bit, ARM 32-bit, ARM 64-bit

Android Debug Bridge (ADB)

Connect to Android at 127.0.0.1:51614

Turn off ADB after debugging. Leaving it on can compromise the security of your system.

Input debugging

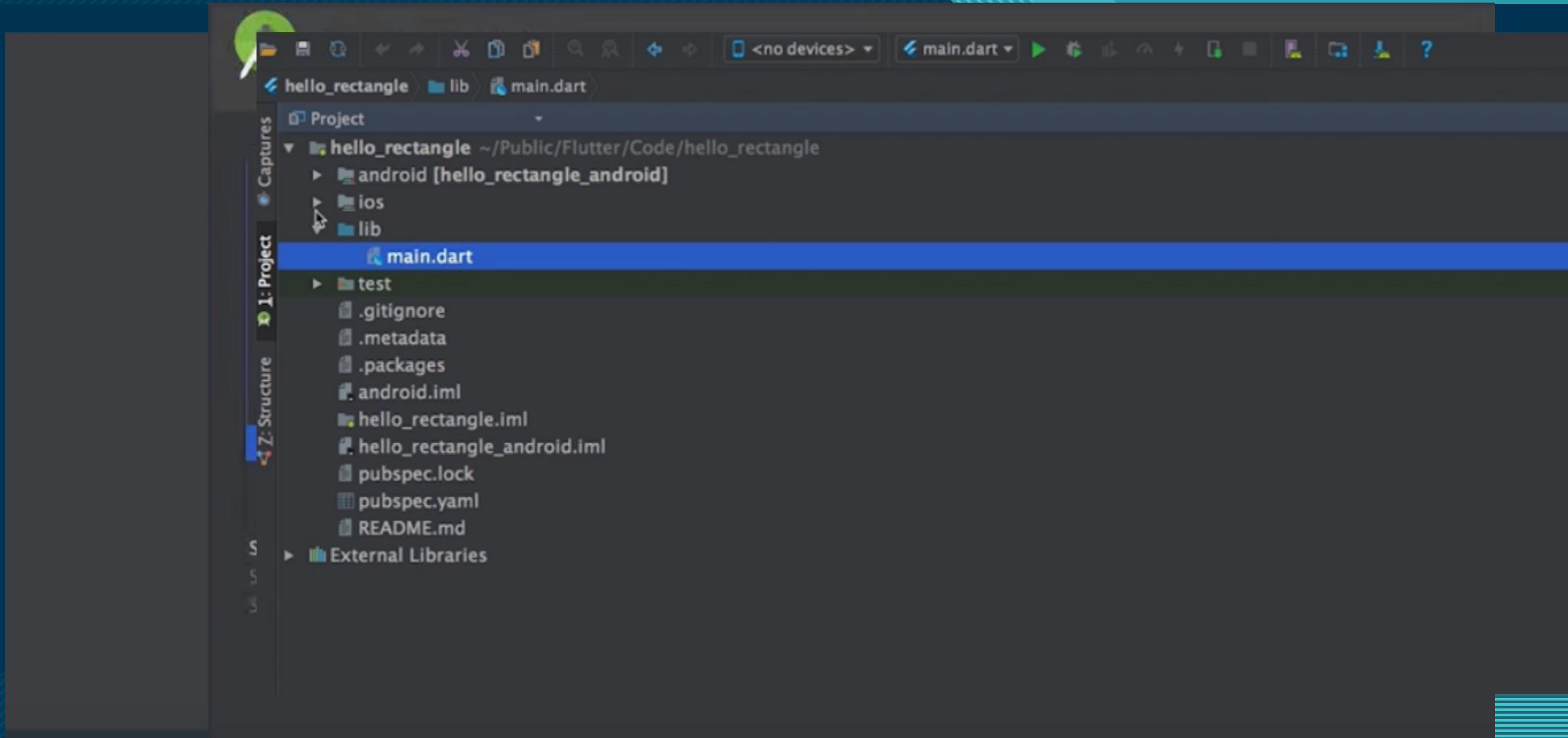
Show visual feedback for taps

Show pointer location for current touch data

Enabling this setting can have adverse effects on your gameplay. Turn it off after debugging.

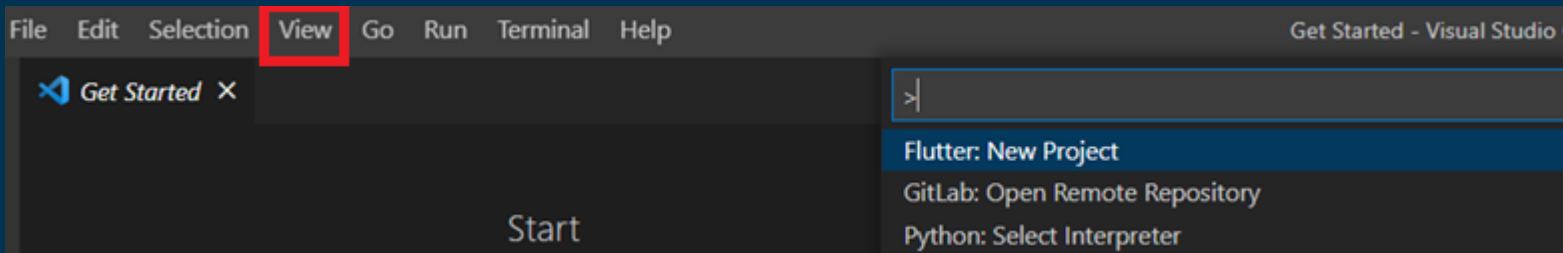
If “AppData” folder is not shown enable “hidden items” in your user folder

Create Flutter Project - Android Studio

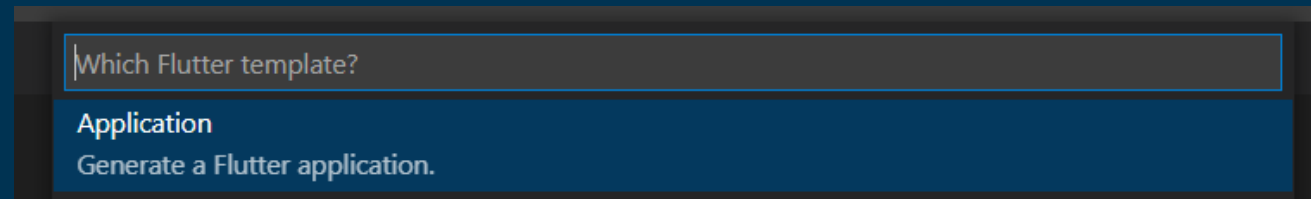


Create Flutter Project - VS Code

1. View → Command Palette → Flutter: New Project

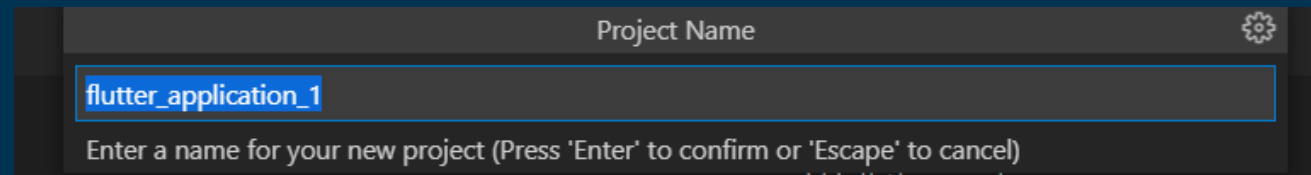


2. Επιλέγετε “Application”



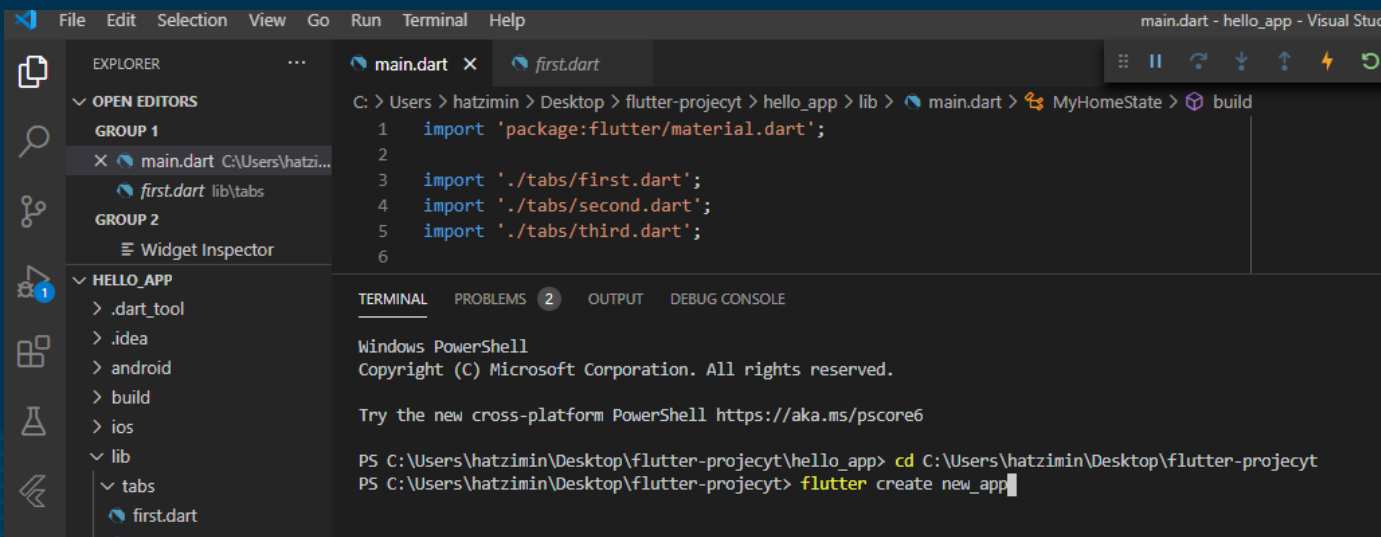
3. Επιλέγετε τον φάκελο που θέλετε να αποθηκευτεί το project

4. Δίνετε τίτλο στο project



Create Flutter Project - VS Code - COMMAND LINE

- Δημιουργείται ένα φάκελο που θέλετε να αποθηκεύονται τα προτζεκτ
- Ανοίγεται το VS Code και στο terminal πηγαίνετε στον φάκελο που δημιουργήσατε και γράφετε “ flutter create your_app_name “
- Από το menu του VS Code επιλέγετε file→open folder→τον φάκελο σας



The screenshot shows the Visual Studio Code interface. The Explorer view on the left shows a project structure with folders for .dart_tool, .idea, android, build, ios, lib, and tabs. The main editor shows a Dart file with the following code:

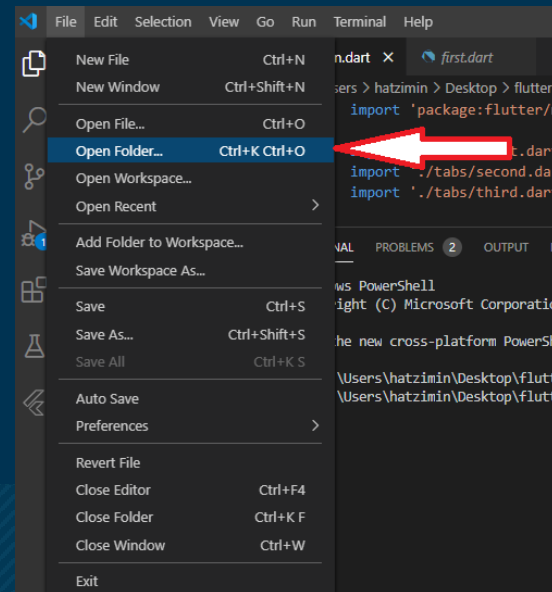
```
C:\Users\hatzimin\Desktop\flutter-projecyt\hello_app\lib> MyHomeState > build
1 import 'package:flutter/material.dart';
2
3 import './tabs/first.dart';
4 import './tabs/second.dart';
5 import './tabs/third.dart';
6
```

The Terminal view at the bottom shows the following commands and output:

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

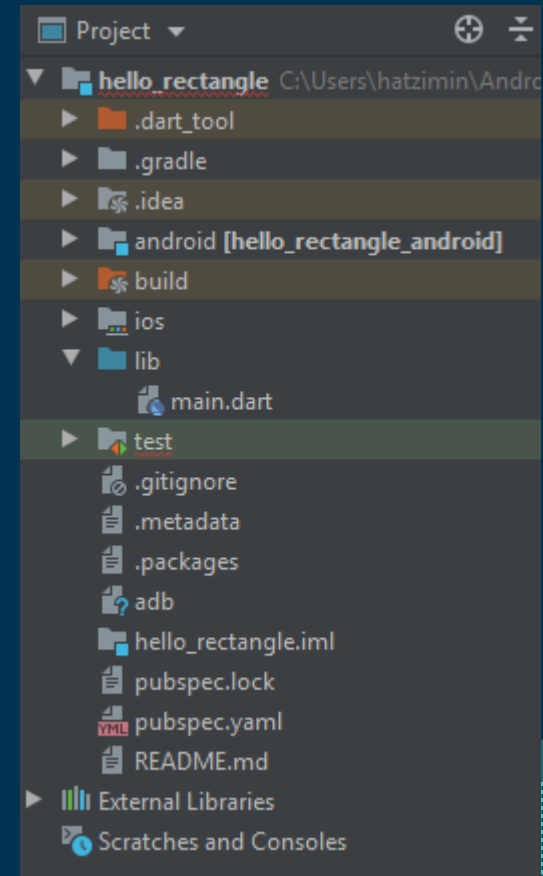
Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\hatzimin\Desktop\flutter-projecyt\hello_app> cd C:\Users\hatzimin\Desktop\flutter-projecyt
PS C:\Users\hatzimin\Desktop\flutter-projecyt> flutter create new_app
```



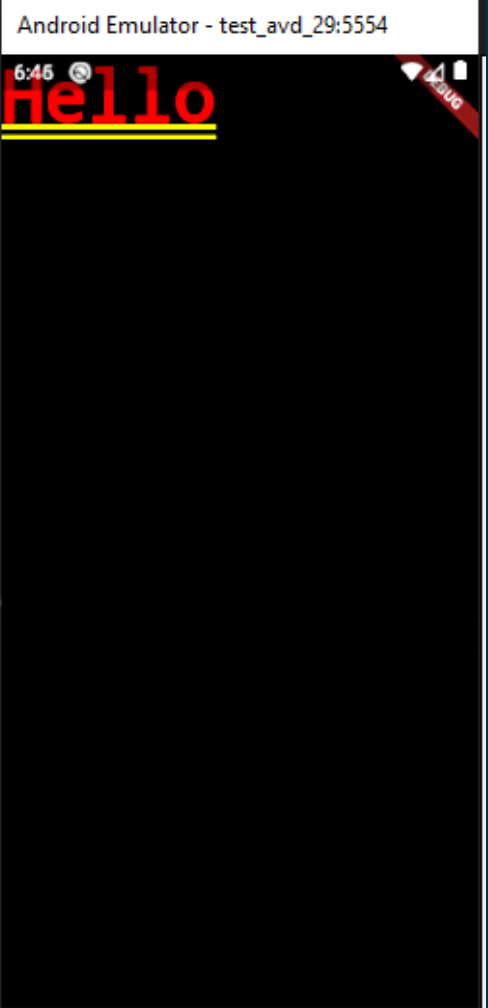
Φάκελοι και Αρχεία

- 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
1 import 'package:flutter/material.dart';
2
Run | Debug
3 void main() { //η main() είναι η πρώτη συνάρτηση που καλεί το flutter
4
5   runApp(
6     MyApp()); //Η συνάρτηση που καλεί όλη την εφαρμογή και εδώ συγκεκριμένα την κλάση MyApp()
7 }
8
9 class MyApp extends StatelessWidget {
10   @override
11   Widget build(BuildContext context) { //override build μεθοδος και επιστρέφει ένα widget
12
13     return MaterialApp( //to MaterialApp είναι το widget που συνδυάζει όλα τα widget και δημιουργεί τη
14
15       home: Text(
16         'Hello'), //To text widget που δίνουμε κείμενο για να τυπωθεί στην οθόνη. // Text
17         //Βάζουμε κομματάκι μετά από κάθε widget για να είναι καλύτερο το format του κώδικα
18     ); // MaterialApp
19   }
20 }
21
22
23
24
25
```



The image shows a side-by-side view of a code editor and an Android emulator. The code editor on the left displays the Dart code for a simple Flutter application. The code defines a `main` function that calls `runApp` with a `MyApp` widget. The `MyApp` class extends `StatelessWidget` and overrides the `build` method to return a `MaterialApp` widget with a `Text` widget containing the word "Hello". The Android emulator on the right shows the rendered output of this code: a black screen with the word "Hello" in red text, underlined in yellow. The emulator's status bar at the top shows the time as 6:45 and the device name as "test_avd_29:5554".

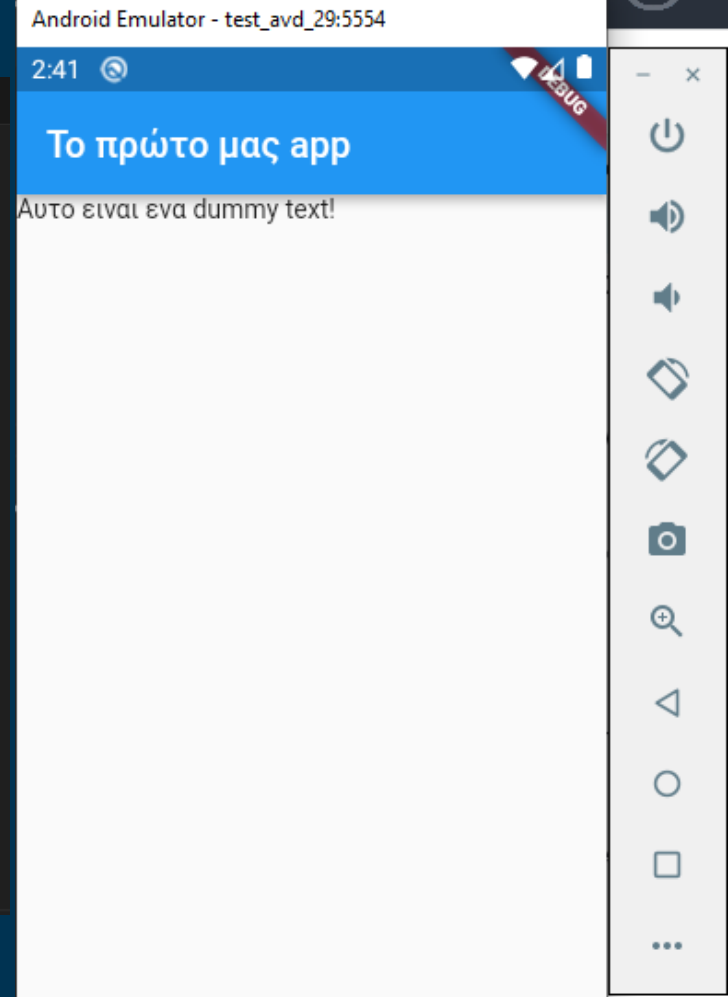
Τι είναι το MaterialApp;

- 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

App with styling

```
EXPLORER
  OPEN EDITORS 1 unsaved
    main.dart lib
  FLUTTER_APPLICATION_1
    .dart_tool
    .idea
    android
    build
    ios
    lib
    main.dart
    linux
    macos
    test
    web
    windows
    .gitignore
    .metadata
    ! analysis_options.yaml
    flutter_application_1.iml
    pubspec.lock
    ! pubspec.yaml
    README.md

main.dart
lib > main.dart > ...
1 import 'package:flutter/material.dart'; //το πακετο που εμπεριεχει ολα τα βασικα widgets
2
Run | Debug | Profile
3 void main() {
4   runApp(
5     MyApp()); //Η συναρτηση που καλει ολη την εφαρμογη και εδω συγκεκριμενα την κλαση MyApp()
6 }
7
8 class MyApp extends StatelessWidget {
9   @override //το override το βαζουμε για να δηλωσουμε στο flutter οτι δεν κανουμε override την build κατα λαθος
10  Widget build(BuildContext context) { //η build συναρτηση που πρεπει να κανουμε override
11
12    return MaterialApp(
13      home: Scaffold( //το Scaffold widget που δημιουργει καποια βασικα στοιχεια μιας εφαρμογης κινητου
14
15        appBar: AppBar(
16          backgroundColor: Colors.blue, //χωρις χρωμα δεν φαινεται η μπαρα
17          title: Text('Το πρώτο μας app'), //text widget σαν input στο title
18        ), // AppBar
19        body: Text(
20          'Αυτο είναι ένα dummy text!',
21          style: TextStyle(color: Colors.purple, fontSize: 25), //styling για το κειμενο που εχουμε βαλει στο body
22        ), //text widget σαν input στο body // Text
23      ), // Scaffold
24    ); //βαζουμε κομμα μετα απο καθε widget για να ειναι καλυτερο το format του κωδικα // MaterialApp
25  }
26 }
```

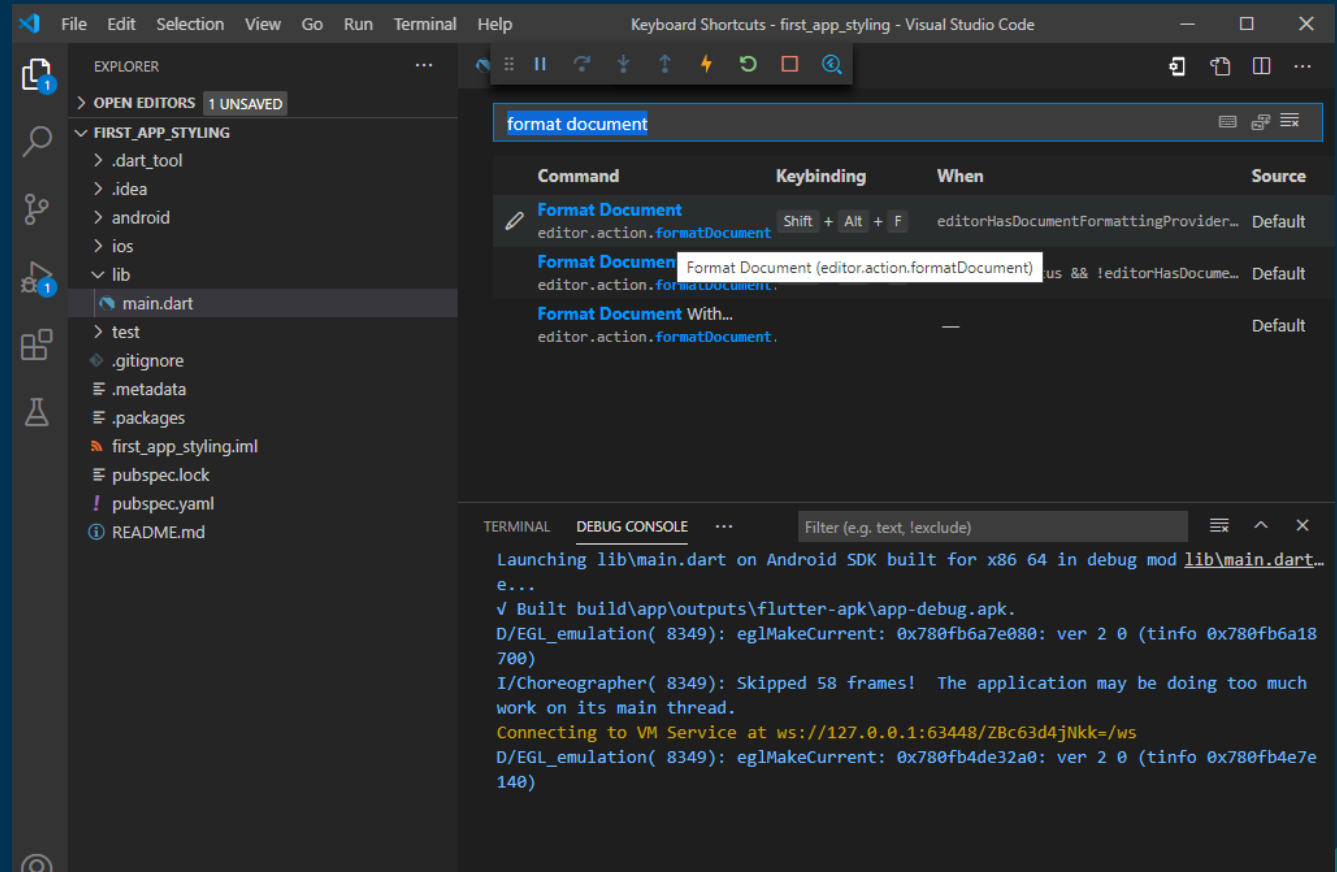


Scaffold and Container Widgets

- 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>

Format Code

- Format Code
 - File → preferences → keyword shortcuts
 - Ψάχνουμε “format document”
 - Βλέπουμε την συντόμευση και την καλούμε στο αρχείο μας



Εξτρά βοηθητικό υλικό

Εγκατάσταση

- Android studio
- Emulator:
 - <https://developer.android.com/studio/run/managing-avds>
- Οδηγίες για VSCode, Android studio
 - <https://flutter.io/using-ide>
 - <https://docs.flutter.dev/tools/vs-code>
- Set up environment
 - <https://flutter.io/setup>

Useful Links

- Official Flutter Docs: <https://flutter.io/docs/>
- macOS Setup Guide: <https://flutter.io/setup-macos>
- Windows Setup Guide: <https://flutter.io/setup-windows>
- Linux Setup Guide: <https://flutter.io/setup-linux>
- Visual Studio Code: <https://code.visualstudio.com/>
- Visual Studio Code Flutter Extension: <https://marketplace.visualstudio.com/items?itemName=Dart-Code.flutter>
- Android Studio: <https://developer.android.com/studio/>

Παραδείγματα Widget

TASK	WIDGET
I want to make a volume control for my app	Slider
I have a large number of photos that I want the user to scroll through, efficiently	GridView
I'm making a form and want a user to select all the types of music they listen to, from a list	Checkbox
I want the user to choose the month of their birthday from a list	Dropdown
I want to make a nice semi-transparent lightbox that overlays my app screen	Opacity

Παραδείγματα Widget

TASK	WIDGET
All I want to do is center this image on the screen!	Center
I have three Containers, and I want them to always each take up a third of the screen.	Flex
I want the Submit button to always be at the top right of the screen.	Align
I want a sticky footer at the bottom of my app that is always visible, laying "on top of" the other widgets on the page	Stack
I want a Widget to perform an action when I swipe left	GestureDetector