Βιοϊατρική Πληροφορική

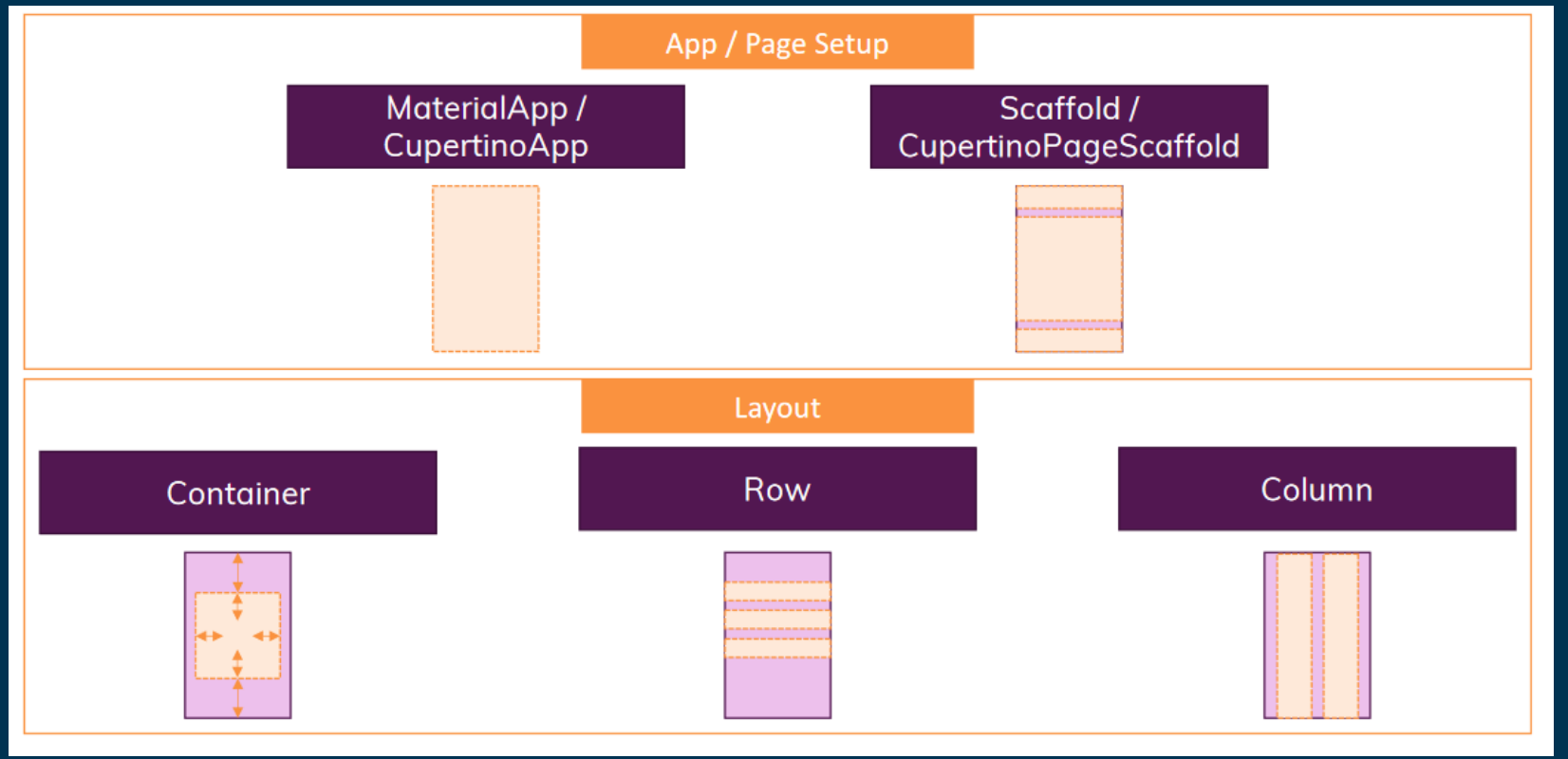
Λευτέρης Κουμάκης,
Μαρία Χατζημηνά

Εργαστήριο 3

Email: ddk8@edu.hmu.gr

Overview Flutter Widgets

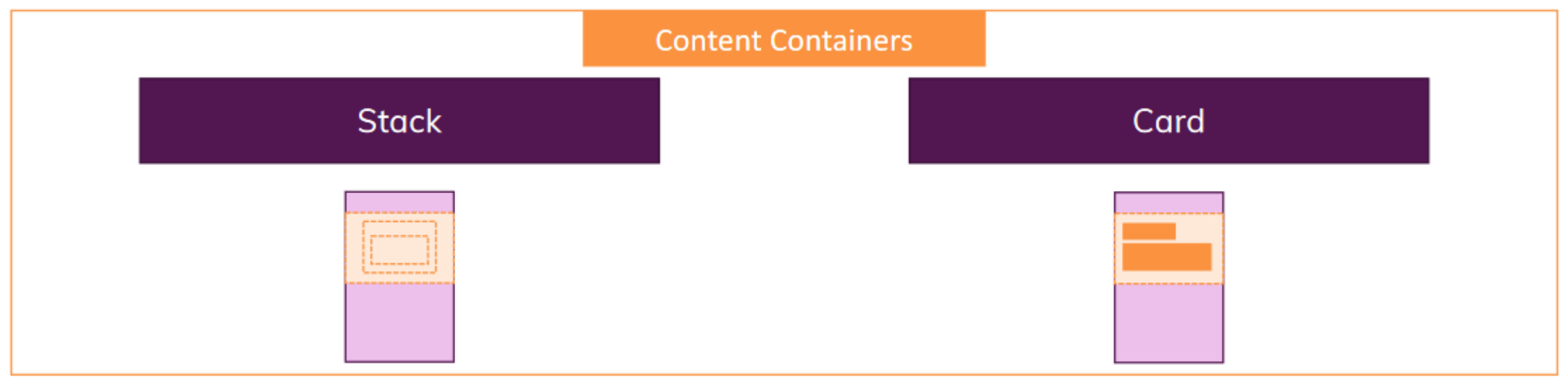
- Flutter Widgets Catalog: https://flutter.dev/docs/development/ui/widgets



Overview Flutter Widgets

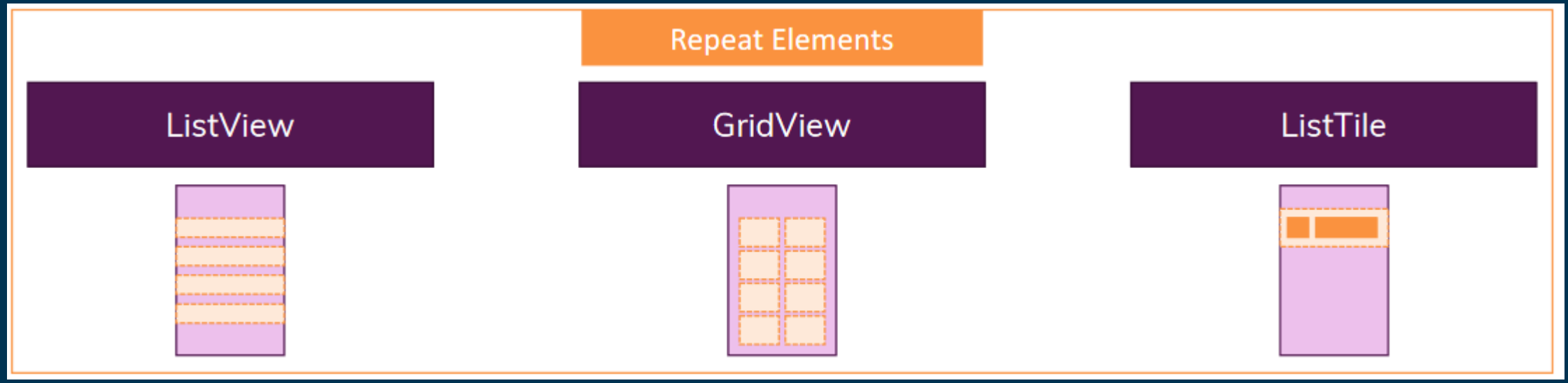
- Container:
  - Εξαιρετικά ευέλικτο widget
  - Μπορεί να μετασχηματιστεί βάση μεγέθους (πλάτος, ύψος, μέγιστο πλάτος, μέγιστο ύψος), στυλ (περίγραμμα, χρώμα, σχήμα, ...) και άλλα
  - Μπορεί να πάρει ένα child (όχι απαραίτητα) το οποίο μπορείτε επίσης να ευθυγραμμίσετε με διαφορετικούς τρόπους
  - Θα χρησιμοποιείτε αυτό το widget αρκετά συχνά
- Row / Column:
  - Πρέπει να το χρησιμοποιήσετε εάν χρειάζεστε πολλά widget το ένα δίπλα στο άλλο οριζόντια ή κάθετα
  - Περιορισμένες επιλογές στυλ => Χρησιμοποιήστε το με Container για να εφαρμόσετε στυλ
  - Τα παιδιά του (children) μπορούν να ευθυγραμμιστούν κατά μήκος του κάθετου και οριζόντιου άξονα

Overview Flutter Widgets



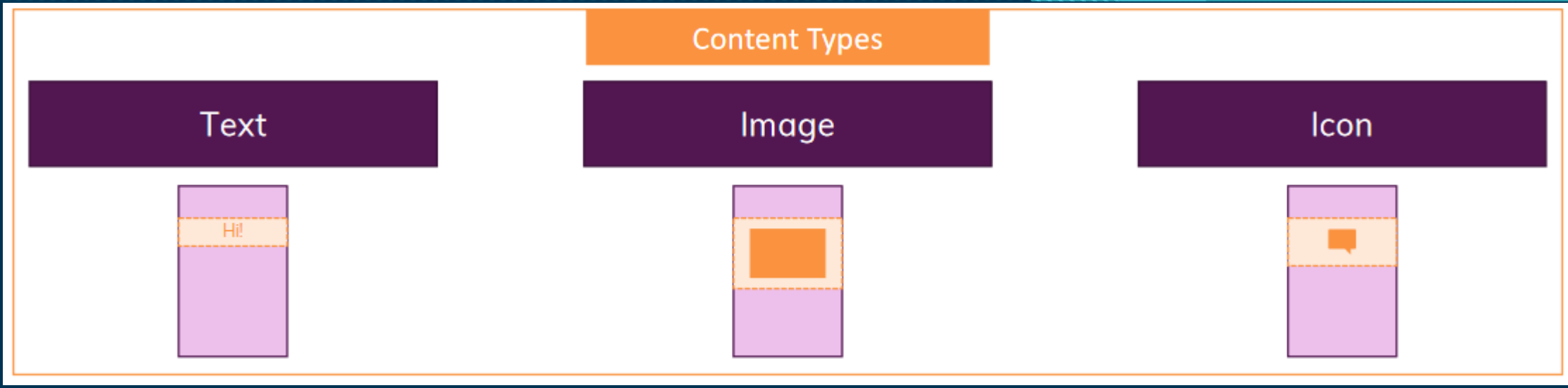
- Stack:
  - Χρησιμοποιείται για την τοποθέτηση αντικειμένων το ένα πάνω στο άλλο (κατά μήκος του άξονα Z)
  - Τα widget μπορούν να αλληλεπικαλύπτονται
- Card:
  - Ένα Container με κάποιο έτοιμο στυλ (σκιά, χρώμα φόντου, στρογγυλεμένες γωνίες)
  - Μπορεί να πάρει ένα child (μπορεί να είναι οτιδήποτε)

Overview Flutter Widgets



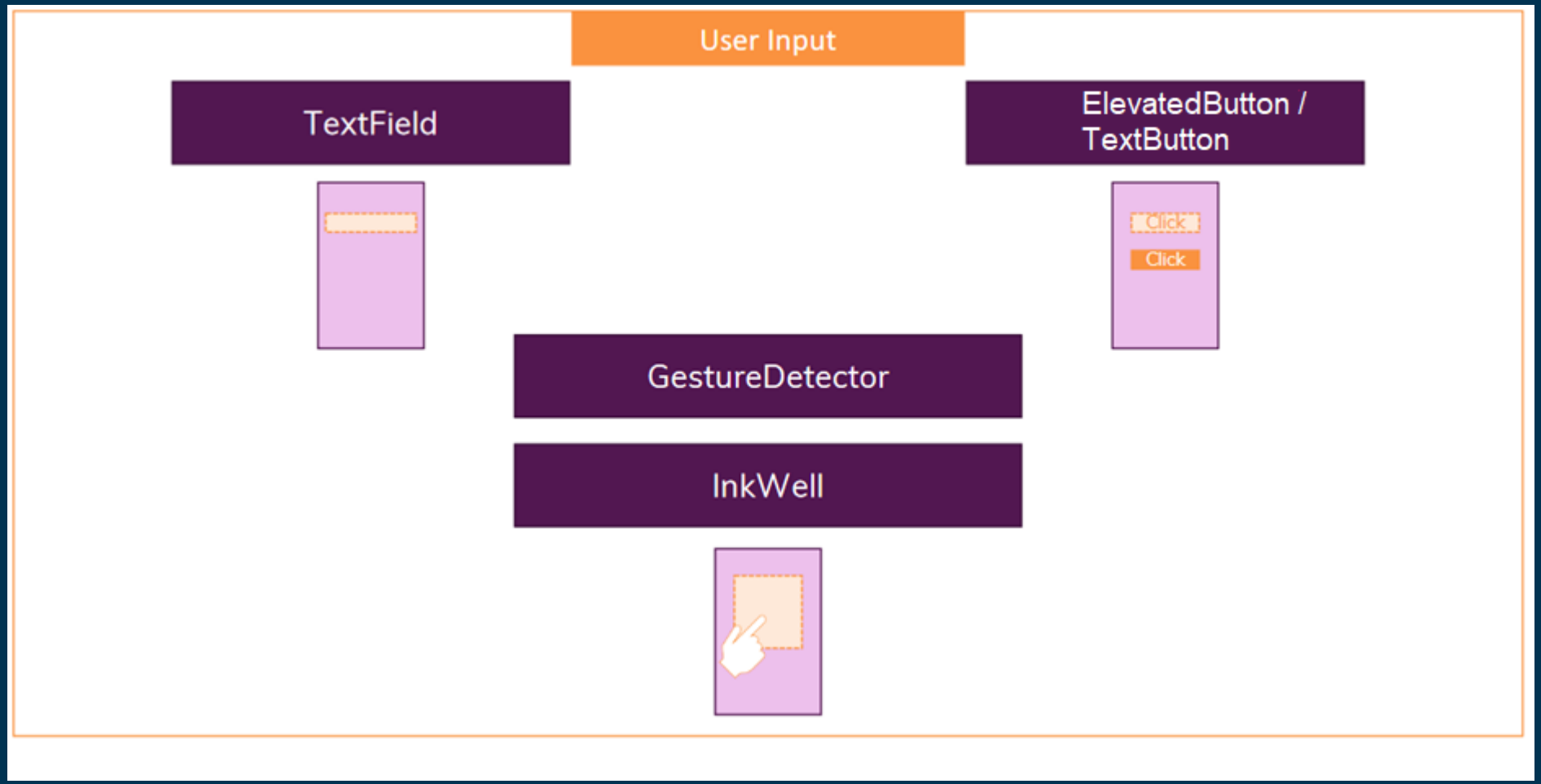
- ListView/ GridView:
  - Χρησιμοποιείται για την εμφάνιση λιστών (ή πλεγμάτων/grid) αντικειμένων
  - Σαν Column () αλλά με δυνατότητα κύλισης (η στήλη δεν έχει)
- ListTile:
  - Ένα pre-styled container / Row() που σου δίνει ένα τυπικό “list-item look”
  - Προσφέρει διάφορες υποδοχές για widgets (π.χ. στην αρχή, έναν τίτλο, στο τέλος)

Overview Flutter Widgets



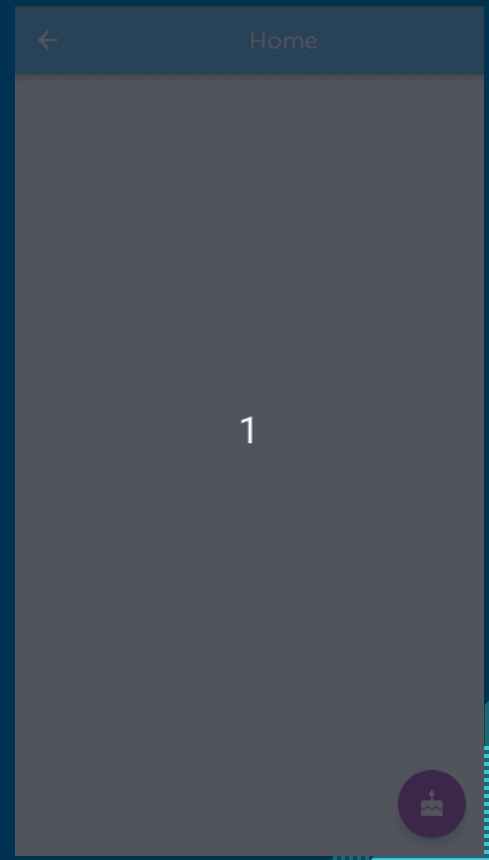
- Text:
  - Ένα widget που απλώς τυπώνει κάποιο κείμενο στην οθόνη
  - Το Text μπορεί να αποκτήσει styling(font family, font weight, font size etc.)
- Image:
  - Χρησιμοποιείται για την εμφάνιση μιας εικόνας στην οθόνη
  - Υποστηρίζει διαφορετικές πηγές (μέσα στην εφαρμογή, online τοποθεσία, ...)
  - Μπορεί να διαμορφωθεί ώστε να έχει το συγκεκριμένο μέγεθος με διαφορετικούς τρόπους
- Icon:
  - Εμφανίζει ένα εικονίδιο στην οθόνη
  - Το Flutter δίνει πολλά έτοιμα εικονίδια Android και iOS τα οποία μπορείτε να χρησιμοποιήσετε
  - Υπάρχει επίσης ένα widget IconButton() σε περίπτωση που χρειάζεστε ένα κουμπί με ένα εικονίδιο

Overview Flutter Widgets



Overview Flutter Widgets

- TextField:
  - Εμφανίζει ένα επεξεργάσιμο πεδίο κειμένου όπου ο χρήστης μπορεί να εισάγει (πληκτρολογήσει) πληροφορίες
  - Πολλές επιλογές διαμόρφωσης (π.χ. αυτόματη διόρθωση, μηνύματα σφάλματος, ετικέτες, στυλ)
  - Υποστηρίζει διαφορετικά είδη (email, αριθμός, κανονικό κείμενο, ...)
- ElevatedButton/ TextButton/ IconButton:
  - Κουμπιά με διαφορετικό στυλ που χειρίζονται το άγγιγμα του χρήστη στην οθόνη
  - Μια συνάρτηση θα εκτελεστεί όταν θα τα πατήσει/αγγίξει ο χρήστης
  - Μπορεί να διαμορφωθεί / προσαρμοστεί εμφανισιακά
- GestureDetector/ InkWell:
  - Το GestureDetector σας επιτρέπει να τυλίξετε οποιαδήποτε widget με touch listeners (π.χ. διπλό πάτημα, παρατεταμένο πάτημα)
  - Το InkWell κάνει το ίδιο, αλλά προσθέτει ένα οπτικό εφέ κυματισμού στο άγγιγμα (το εφέ μπορεί να διαμορφωθεί)



Splash Screen

- Η σελίδα που θα ανοίγει όταν τρέχουμε την εφαρμογή
- Αλλαγή κειμένου με Random τρόπο
  - import 'dart:math';
  - Κάνουμε override την init.State
- Η init.State καλείτε μόνο μια φορά ακόμα και στα statefulWidgets
- Κάνουμε override την init.State όταν θέλουμε να κάνουμε κάποια αρχικοποίηση πριν να τρέξει η build

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

class SplashScreen extends StatefulWidget {
  @override
  _SplashScreenState createState() => _SplashScreenState();
}

class _SplashScreenState extends State<SplashScreen> {
var yourList = ["first item", "second item", "third item"];//το κειμενο του splash screen\
  int randomIndex=0;
  //δημιουργει ενα τυχαιο αριθμο απο το length της λιστας
  @override
  void initState() {
    super.initState();
    randomIndex = Random().nextInt(yourList.length); //πριν την build οριζουμε το στοιχειο
    Timer(Duration(seconds: 3), () {
      //μετα απο τρια δευτερολεπτα καλειται ο Navigator
      Navigator.pushNamed(
          context, '/homepage'); // εδω χρησιμοποιουμε το named route που εχου-
      //με δηλωσει στην main.dart

      // Navigator.of(context).pushReplacement(MaterialPageRoute(
      //  builder: (context) => HomePage(),
      //));  //διαφορετικος τροπος χωρις named route. Απλα καλεις το widget που θες
    }); // Timer
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      backgroundColor: Colors.indigo,
      body: Center(
        child: Text(yourList[randomIndex],
            style: TextStyle(
              fontSize: 45,
            )), // TextStyle // Text
      ), // Center
    ); // Scaffold
  }
}
```

setState - initState

- setState
  - Η κλήση του setState ειδοποιεί το framework ότι η εσωτερική κατάσταση αυτού του αντικειμένου έχει αλλάξει με τρόπο που μπορεί να επηρεάσει τη διεπαφή χρήστη σε αυτό το υπό-δέντρο, γεγονός που προκαλεί το framework να εκτελέσει την build για να αλλάξει την κατάσταση του συγκεκριμένου αντικειμένου.
- initState
  - Κάνουμε override αυτήν τη μέθοδο αν χρειαστεί να κάνουμε κάποιου είδος αρχικοποίησης επειδή σε αντίθεση με την build() αυτή η μέθοδος καλείται μία φορά

```
@override
void initState() {
  super.initState();
  //ο κωδικας που θελετε να τρεξετε

}
```

Κώδικας Stateless και Stateful

Stateless

```
class MyWidget extends StatelessWidget {
  const MyWidget({Key? key}) : super(key: key);

  @override
  Widget build(BuildContext context) {

  }
}
```

Stateful

```
class MyWidget extends StatefulWidget {
  const MyWidget({Key? key}) : super(key: key);

  @override
  State<MyWidget> createState() =>
_MyWidgetState();
}

class _MyWidgetState extends
State<MyWidget> {
  @override
  Widget build(BuildContext context) {

  }
}
```

Card Widget

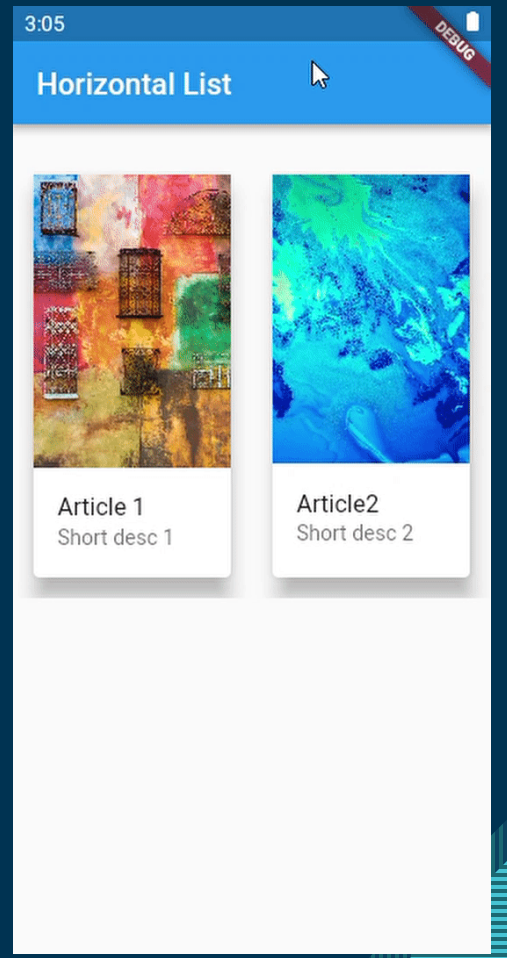
- Card Widget:
  - Χρησιμοποιούμε το name argument elevation για την σκίαση
- Material Widget:
  - elevation: σκίαση
  - borderRadius: Στρογγυλεμένες άκρες ανάλογα τον αριθμό στην παρένθεση
- ClipOval Widget:
  - Το χρησιμοποιούμε για να κάνουμε wrap μια εικόνα που θέλουμε να είναι στρογγυλή

```
child: Card(
  //το wdiget Card
  color: Colors.blue,
  elevation: 10.0, //η σκια που θελουμε να φαινεται
  child: Container(
    child: Column(
      children: [
        Padding(
          padding: EdgeInsets.symmetric(
              vertical: 10.0), //συμμετρικη αποσταση στον καθετο αξονα // E
          child: Material(
            //χρησιμοποιουμε το material για την σκιαση κτλ
            elevation: 5.0, //σκια
            borderRadius:
                BorderRadius.circular(100), //στρογγυλεμενες ακρες
            child: Container(
              height: 200,
              width: 200,
              child: ClipOval(
                //στρογγυλη εικονα
                child: Image(
                  fit: BoxFit.contain, //να γεμιζει η εικονα τον χωρο
                  image: AssetImage(imagepath), //δυναμικο path εικονας
                ), // Image
              ), // ClipOval
            ), // Container
          ), // Material
        ), // Padding
        Center(
          child: Text(
            langname, //δυναμικο κειμενο εικονας
            style: TextStyle(
                fontSize: 24,
                color: Colors.white,
                fontWeight: FontWeight.bold), // TextStyle
          ), // Text
        ), // Center
      ],
    ), // Column
  ), // Container
), // Card
```

Horizontal List View

- ListView
  - Για να έχουμε οριζόντια κύλιση βάζουμε
    - scrollDirection: Axis horizontal
- SizedBox()
  - Μπορείς να προσθέσεις κενό ανάμεσα σε widget ανάλογα το width και height που δηλώνεις στο Sizedbox()
  - π.χ. SizedBox(width:50)

```
body: Container(
  margin: EdgeInsets.symmetric(vertical: 20.0),
  height: 300.0,
  child: ListView(
    scrollDirection: Axis.horizontal,
    children: <Widget>[
      Container(
        width: 160,
        child: customcard(
            text[0], images[0], shortDescription[0], '/article1'),
      ), // Container
      Container(
        width: 160,
        child: customcard(
            text[1], images[1], shortDescription[1], '/article1'),
      ), // Container
      Container(
        width: 160,
        child: customcard(
            text[2], images[2], shortDescription[2], '/article1'),
      ), // Container
      Container(
        width: 160,
        child: customcard(
            text[3], images[3], shortDescription[3], '/article1'),
      ), // Container
      Container(
        width: 160,
        child: customcard(
            text[4], images[4], shortDescription[4], '/article1'),
      ), // Container
    ], // <Widget>[]
  ), // ListView
), // Container
```



Card and Image Widget

- Image()
  - Μπορούμε να βάλουμε εικόνα από link με:
    - Image.network(URL)
- Card()
  - Μπορούμε να χρησιμοποιήσουμε το Wrap() Widget για να εισάγουμε περισσότερα από ένα child σε μια κάρτα

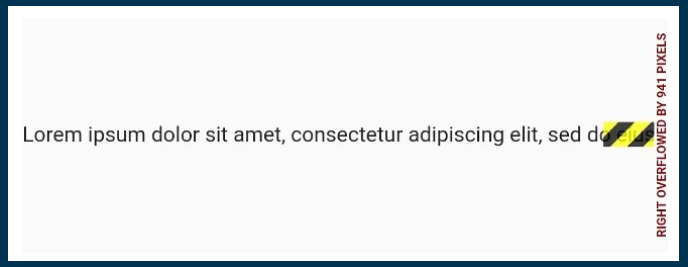
```
List<String> images = [
  "https://images.unsplash.com/photo-1503875154399-95d2b151e3b0?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  "https://images.unsplash.com/photo-1484581400079-58a319a15a2a?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjIxMTIzfQ&auto=format&fit=crop&w=500&q=60",
  "https://images.unsplash.com/photo-1515875294982-4796669a7932?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=500&q=60",
  "https://images.unsplash.com/photo-1528155124528-06c125d81e89?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjEyMDd9&auto=format&fit=crop&w=500&q=60",
  "https://images.unsplash.com/photo-1542052722982-1c9f552a534b?ixlib=rb-1.2.1&ixid=eyJhcHBfaWQiOjF9&auto=format&fit=crop&w=634&q=80",
];

Widget customcard(
    String text, String imagepath, String shortDescription, String route) {
  return Padding(
    padding: EdgeInsets.all(10.0),
    child: InkWell(
      onTap: () {
        Navigator.of(context).pushNamed(route);

        print(
            'card tapped'); //τυπώνεται οταν πατιεται η καρτα λογω του InkWell
      },
      child: Card(
        elevation: 10.0, //η σκια που θελουμε να φαινεται
        child: Wrap(
          children: [
            Image.network(imagepath),
            ListTile(
              title: Text(text),
              subtitle: Text(shortDescription),
            ), // ListTile
          ],
        ), // Wrap
      ), // Card
    ), // InkWell
  ); // Padding
```

Overflow error

- MediaQuery.of(context).size.width - Μας δίνει το πλάτος της συσκευής
- MediaQuery.of(context).size.height - Μας δίνει το ύψος της συσκευής
- Μπορούμε να τα διαιρούμε ή να τα πολλαπλασιάζουμε ανάλογα τι μέγεθος θέλουμε να έχει το widget που φτιάχνουμε
- Άλλα widgets που μπορούν να βοηθήσουν:
  - Flexible()
  - Expanded()




```
Expanded(
  child:Text(
    "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed"
    " do eiusmod tempor incididunt ut labore et dolore magna "
    "aliqua. Ut enim ad minim veniam, quis nostrud "
    "exercitation ullamco laboris nisi ut aliquip ex ea "
    "commodo consequat."),
)
```

```
Flexible(
  child:Text(
    "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed"
    " do eiusmod tempor incididunt ut labore et dolore magna "
    "aliqua. Ut enim ad minim veniam, quis nostrud "
    "exercitation ullamco laboris nisi ut aliquip ex ea "
    "commodo consequat."),
)
```

Connect BlueStacks emulator VSCode

- Στον emulator του bluestacks πάμε Settings→ Advanced
- Ανάλογα το port (εδώ είναι το 51614) αλλά κάθε φορά αλλάζει αυτός ο αριθμός, γράφουμε στο command line στο μονοπάτι που είναι το “platform-tools”:

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

- Και θα εμφανιστεί στο VSCode