

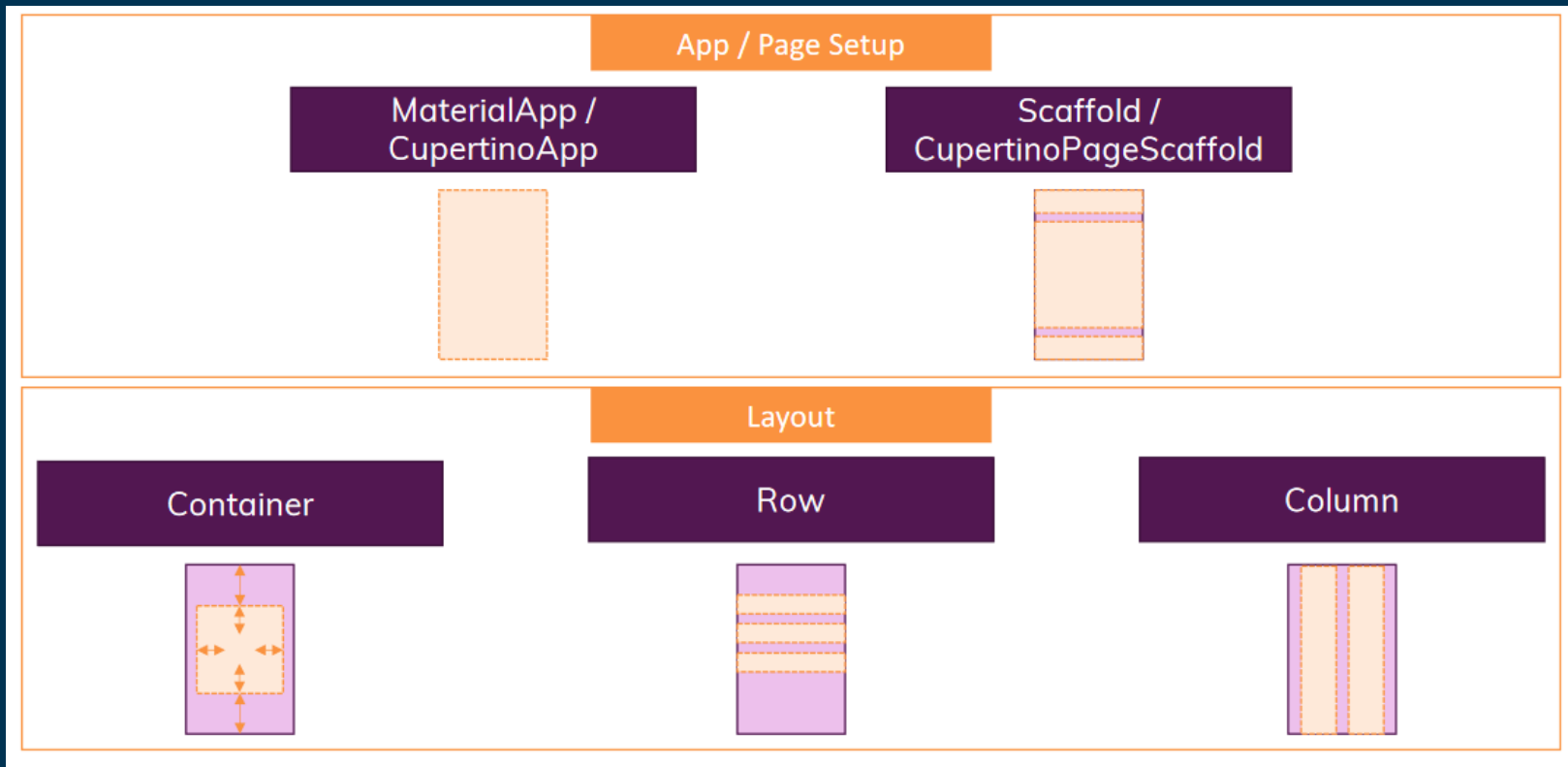


Bioinformatics

Lefteris Koumakis, Maria Chatzimina

Overview Flutter Widgets

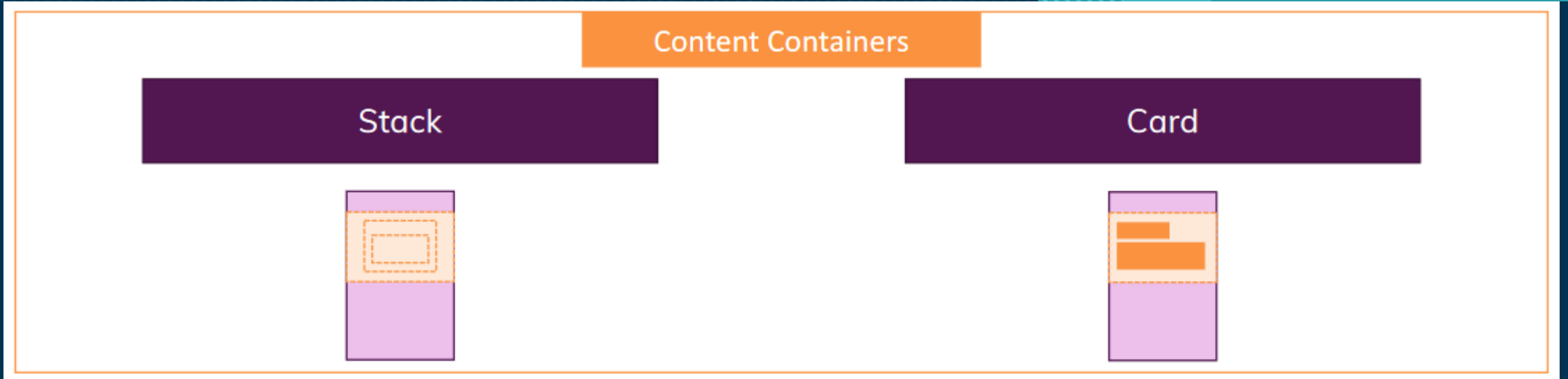
- Flutter Widgets Catalog: <https://flutter.dev/docs/development/ui/widgets>



Overview Flutter Widgets

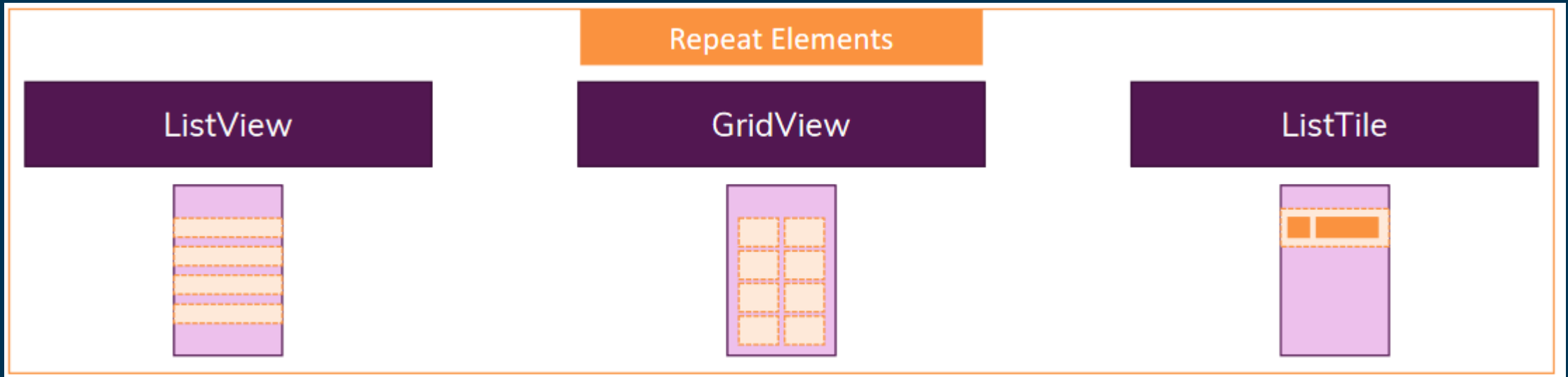
- **Container:**
 - Extremely flexible widget
 - Can be transformed based on size (width, height, max-width, max-height), style (border, color, shape, ...) and more
 - It can take a child (not necessarily) which you can also align in different ways
 - You will be using this widget quite often
- **Row / Column:**
 - You should use it if you need multiple widgets next to each other horizontally or vertically
 - Limited style options => Use with Container to apply styles
 - Its children can be aligned along the vertical and horizontal axis

Overview Flutter Widgets



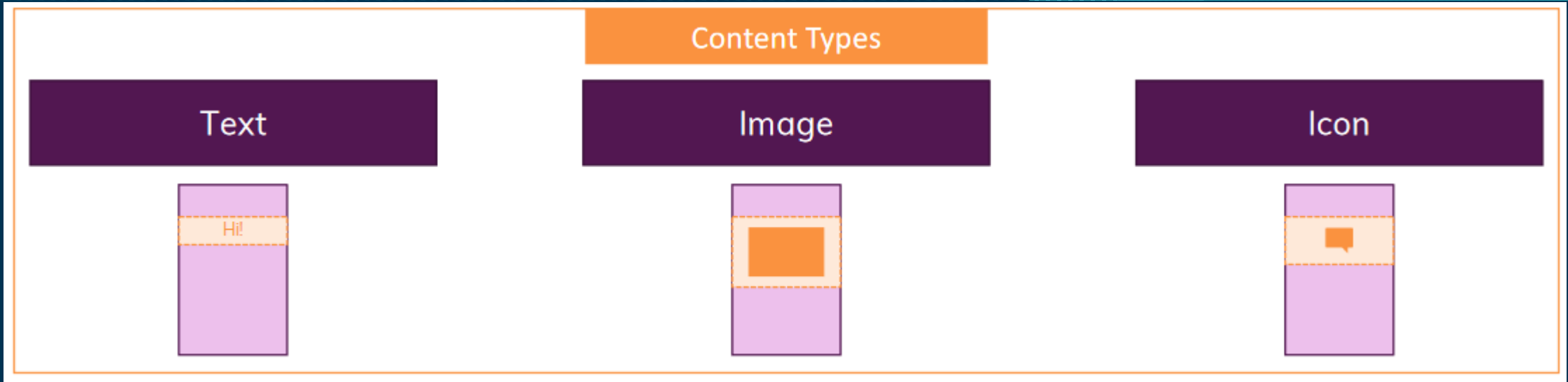
- **Stack:**
 - Used to stack objects on top of each other (along the Z axis) Widgets can overlap
- **Card:**
 - A Container with some ready-made styling (shadow, background color, rounded corners)
 - Can take a child (can be anything)

Overview Flutter Widgets



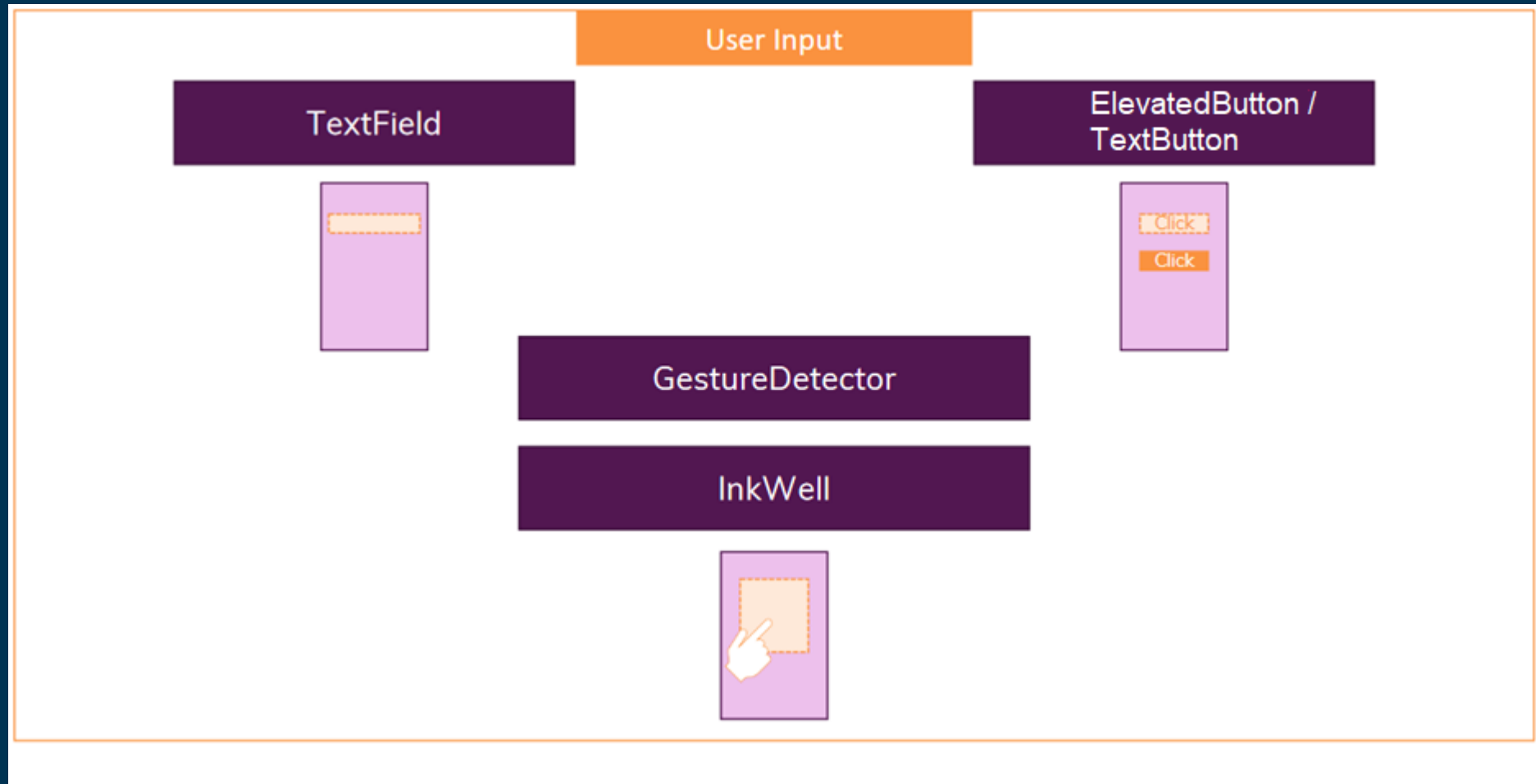
- **ListView/ GridView:**
 - Used to display lists (or grids) of objects
 - Like Column() but scrollable (column doesn't scroll)
- **ListTile:**
 - A pre-styled container / Row() that gives you a standard "list-item look"
 - Offers various widget slots (e.g. at the beginning, a title, at the end)

Overview Flutter Widgets



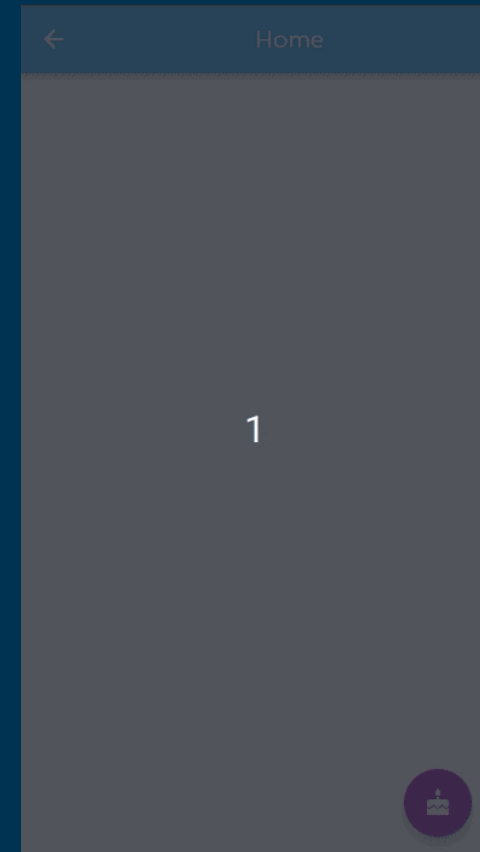
- **Text:**
 - A widget that just prints some text on the screen
 - Text can acquire styling (font family, font weight, font size etc.)
- **Image:**
 - It is used to display an image on the screen
 - Supports different sources (in-app, online, ...)
 - It can be shaped to the specific size in different ways
- **Icon:**
 - Displays an icon on the screen
 - Flutter gives many ready-made Android and iOS icons that you can use

Overview Flutter Widgets



Overview Flutter Widgets

- **TextField:**
 - Displays an editable text field where the user can enter (type) information
 - Lots of configuration options (e.g. autocorrect, error messages, labels, styles)
 - It supports different types (email, number, regular text, ...)
- **ElevatedButton/ TextButton/ IconButton:**
 - Differently styled buttons that handle the user's touch on the screen
 - A function will be executed when the user taps/touches them
 - It can be configured / customized in appearance
- **GestureDetector/ InkWell:**
 - GestureDetector allows you to wrap any widget with touch listeners (e.g. double tap, long tap)
 - InkWell does the same, but adds a visual ripple effect to the touch (the effect can be configured)



Splash Screen

- The page that will open when we run the application
- Change text in Random way
 - import 'dart:math';
 - We override init.State
- You call init.State only once even in statefulWidget
- We override init.State when we want to do initialization before the build method runs

```
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
 - Calling setState notifies the framework that the internal state of this object has changed in a way that may affect the user interface in this subtree, which causes the framework to run build to change the state of that object.
- initState
 - We override this method if we need to do some kind of initialization because unlike build() this method is called once

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

Code 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:
 - We use the named argument “elevation” for shading
- Material Widget:
 - elevation: shade
 - borderRadius: Rounded edges according to the number in parentheses
- ClipOval Widget:
 - We use it to wrap an image that we want to make it round

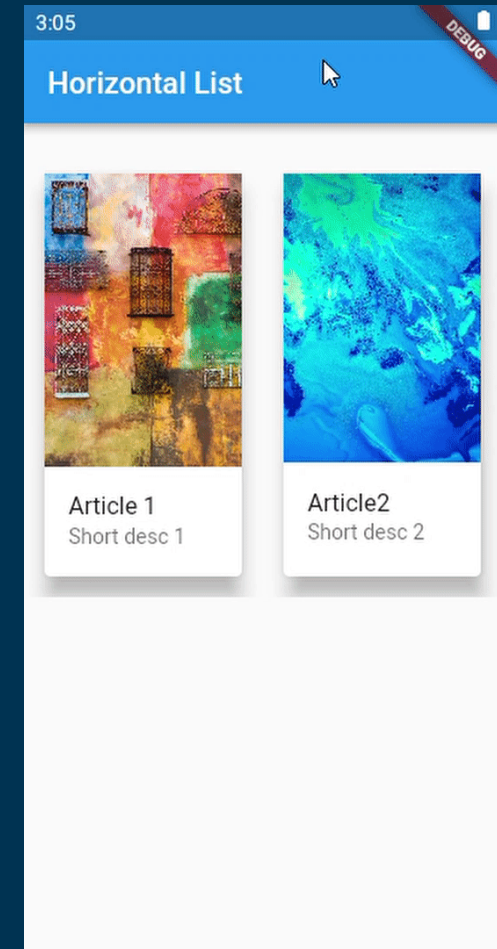
```
child: Card(  
  //το widget Card  
  color: Colors.blue,  
  elevation: 10.0, //η σκια που θελουμε να φαινεται  
  child: Container(  
    child: Column(  
      children: [  
        Padding(  
          padding: EdgeInsets.symmetric(  
            vertical: 10.0), //συμμετρικη αποσταση στον καθετο αξονα // t  
          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
 - To have horizontal scrolling we add
 - `scrollDirection: Axis.horizontal`
- SizedBox()
 - You can add space between widgets depending on the width and height you declare in Sizedbox()
 - E.g. `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()
 - We can use an online image:
 - `Image.network(URL)`
- Card()
 - We can use `Wrap()` Widget to add more than widgets since Card has one “child”

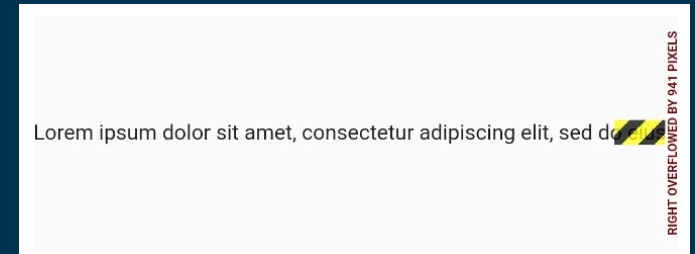
```
List<String> images = [
  "https://images.unsplash.com/photo-1503875154399-95d2b151e3b0?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=60",
  "https://images.unsplash.com/photo-1484581400079-58a319a15a2a?ixlib=rb-1.2.1&ixid=eyJhchBfawQiojIyMTIzF0&auto=format&fit=crop&w=500&q=60",
  "https://images.unsplash.com/photo-1515875294982-4796669a7932?ixlib=rb-1.2.1&ixid=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 όταν 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` - It gives us the width of the device
- `MediaQuery.of(context).size.height` - It gives us the height of the device
- We can divide or multiply them depending on what size we want to transform the widget
- Other widgets that can help with overflow:
 - `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."),  
  )  
)
```

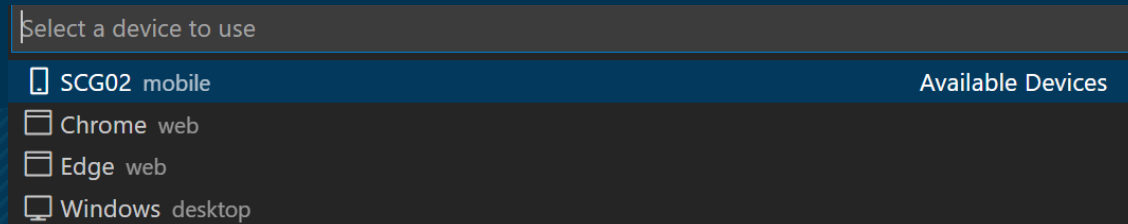
```
Flexible(  
  child:Text(  
    "Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed"  
    " do eiusmod tempor incididunt ut labore et dolore magna "  
    "aliqua. Ut enim ad minim veniam, quis nostrud "  
    "exercitation ullamco laboris nisi ut aliquip ex ea "  
    "commodo consequat."),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

Connect BlueStacks emulator VSCode

- In bluestacks emulator got to Settings → Advanced
- Depending on the port(in the example is 51614) but the number may change everytime
- Type in command line to “platform-tools” path:

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

- The emulator will show on 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.