FLUTTER

In-depth analysis of a cross-platform framework

History

The first version was “Sky” presented in 2015

Flutter 1.0 was released on December, 4th, 2018

INTRODUCTION

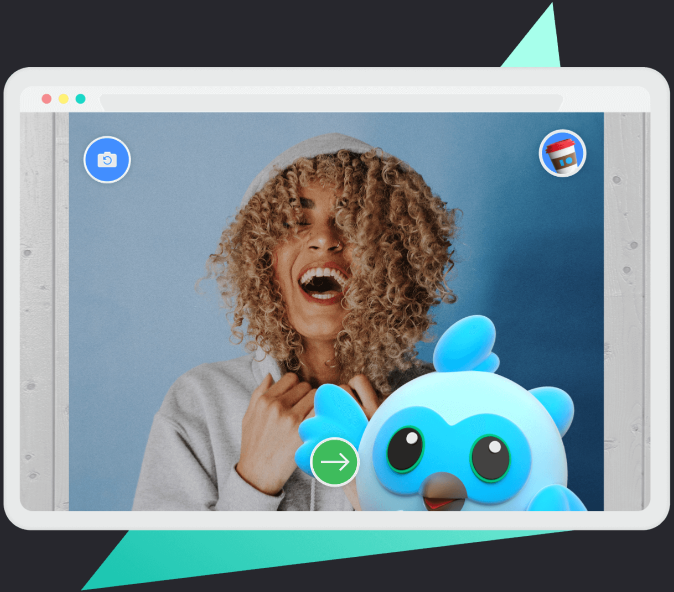
Flutter is an SDK for mobile devices, developed by Google, for the development of native application for iOS and Android starting from a uniquecodebase

CROSS-COMPILED Approach

Application written in **Dart**

INITIAL SUPPORTED PLATFORMS

OTHER SUPPORTED PLATFORMS



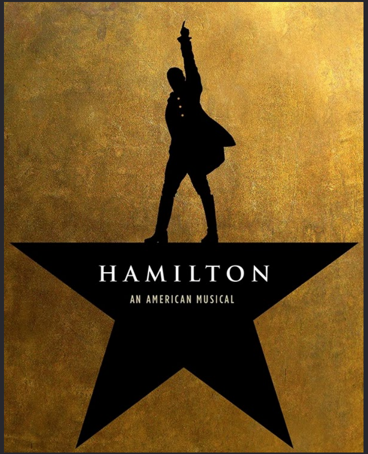
SHOWCASE

Google Ads

Alibaba.com™

APPTREE

Greentea



Abbey Road Studios

reflectly

MINDFULNESS JOURNAL

MAIN CHARACTERISTICS

- Fast development
- Expressive and flexible UI
- Native performances

FAST DEVELOPMENT

- **Hot reload:** it allows to build and reload the code during runtime
  - Stateful
- Pre-defined Widgets

EXPRESSIVE AND FLEXIBLE UI

- Personalized user experience thanks to the enormous amount of widget with **material design** and **Cupertino** style

NATIVE PERFORMANCES

- Native apps
- Widgets incorporate all the main characteristics of different platforms (e.g., scrolling, icons, fonts)

PROs & CONs

- Free e opensource
- Single codebase
- Easy setup
- Hot reload
- Widgets
- Native performances
- Plugins for IDE
- Documentation

- ~~Available only for mobile~~
- ~~Low number of libraries~~
- Difficult to create animations
- Need to know Dart

FLUTTER GUIDELINES

- Control
- Performances
- Fidelity

ACCESSIBILITY

Components to support accessibility:

- Big fonts
- Screen reader
- Contrast

COMMUNITY

- Github
- Stack Overflow
- Google groups
- Youtube
- Slack
- Twitter
- Medium
- Meetup

Official website with:

- Cookbook
- Codelabs
- Tutorials

DART

DART LANGUAGE

It is a programming language, object oriented, used to develop web, server, desktop and mobile applications, developed by Google (first name was Dash)

DART – SUPPORTED TYPES

- Numbers (int or double, num subtypes)
- Strings (String)
- Booleans (bool)
- enum
- List
- Sets
- Maps
- Runes (to use Unicode characters in a string)
- Symbols
- Generics(ex: List<type> o List<dynamic>)

VARIABLES

Each variable points to an object and stores a reference

```
var name = 'Bob';    String name = 'Bob';
```

Variables have a default null value if not initialized

```
int lineCount;
```

Identifiers can start with letters or _, and the name can have both and contain numbers

CONSTANTS

It is possible to define constants variables using final or const

```
final name = 'Bob'; // type determined by compiler
final String nickname = 'Bobby';
```

Instance variable can be only final

The keyword const can be used even for values

```
final bar = const [];
const baz = []; // equivalent to `const []`
```

LIBRARIES AND VISIBILITY - 1

Every Dart app is a library

It is possible to use libraries for code modularity

```
import 'dart:html';
```

Lazy loading for libraries

```
import 'package:greetings/hello.dart' deferred as hello;
```

LIBRARIES AND VISIBILITY - 2

Keywords show and hide:

```
import 'package:lib1/lib1.dart' show foo;
import 'package:lib2/lib2.dart' hide foo;
```

Identifiers starting with _ are visible only inside the library

STATEMENT FOR FLOW CONTROL

```
if (isRaining()) {
    ...
} else if
(isSnowing()) {
    ...
} else {
        ...
}
```

```
for (var i=0; i<5; i++) {
    print(i)
}

while (!isDone()) {
    doSomething();
}

do {
    printLine();
} while(!atEndOfPage());
```

```
switch(expression) {
  case 'A':
    ...
    break;
  case 'B':
        ...
    break;
  default:
        ...
  }
```

EXCEPTIONS

Exceptions are not managed

```
try {
    breedMoreLlamas();
} on OutOfLlamasException { // a specific exception
    buyMoreLlamas();
} on Exception catch (e) { // all the exceptions
    print('Unknown exception: $e');
}
```

INHERITANCE

Classes can inherit from other classes but only once (single-inheritance)

Keywords abstract, extends, implements, @override

```
class TV {
  void turnOn() {
    _illuminateDisplay();
    _activateIrSensor();
  }
}
```

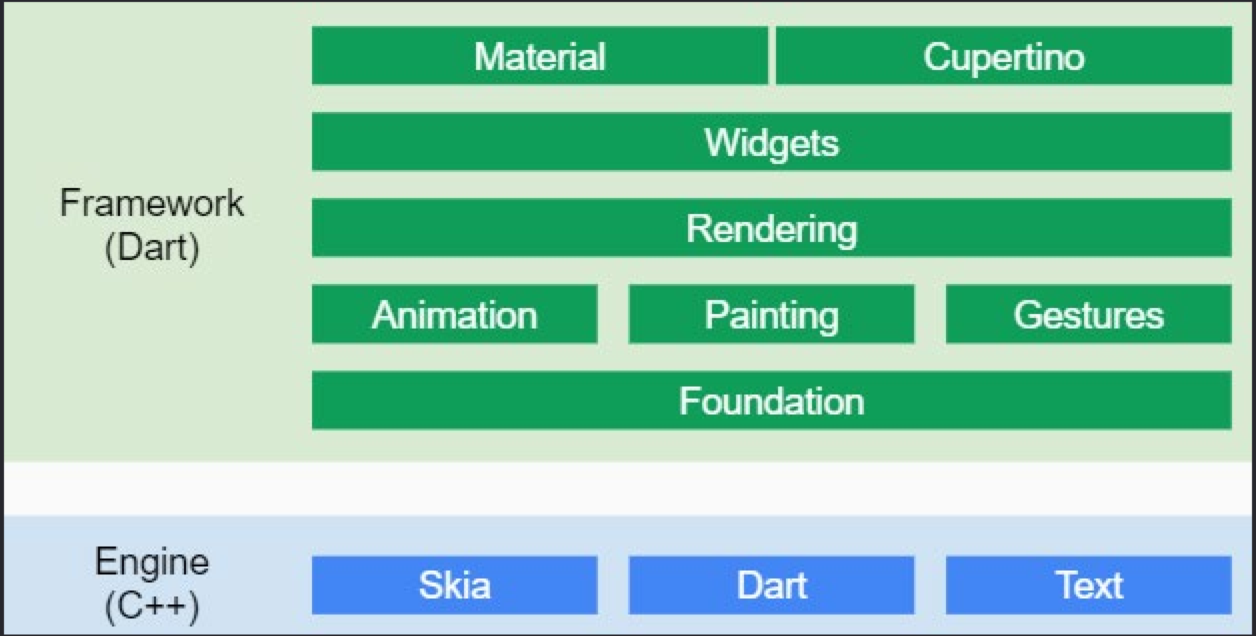
```
class SmartTV extends TV {
  void turnOn() {
    super.turnOn();
    _bootNetworkInterface();
  }
}
```

DART CODE COMPILATION

- Dart code can be compiled in different ways
  - just-in-time (JIT)
  - ahead-of-time (AOT)
    - Makes framework **cross-compiled**

ARCHITECTURE

FLUTTER SDK COMPONENTS



FRAMEWORK ARCHITECTURE

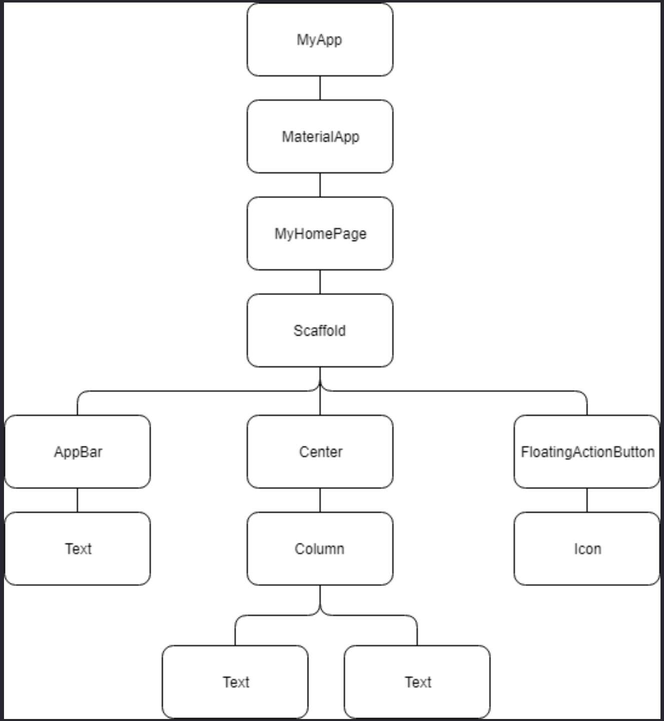
- Flutter architecture is based on the following components:
  - **Material e Cupertino** : implements widget Material (Android) and Cupertino (iOS) style
  - **Widgets** : implements generic widgets
  - **Rendering** : simplify layout management
  - **Animation** : tween and physics-based
  - **Painting, Gestures**
  - **Foundation**
  - **Dart:ui** : manage communications with the Flutter engine

WIDGET

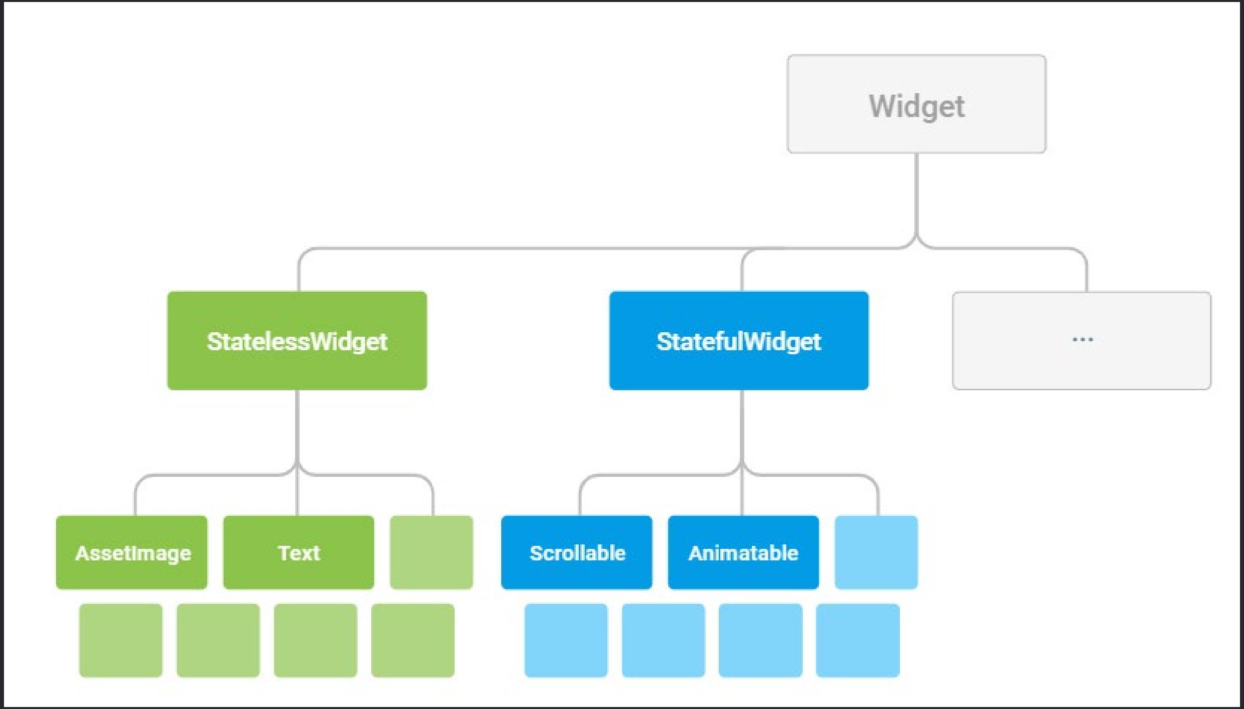
- Base components of the user interface
- Each widget is an unchangeable declaration of the user interface
- A widget can define:
  - A structural element (button, menu, ...)
  - A style element (font, ...)
  - An aspect of the layout (padding, ...)
- Define as hierarchy based on composition
- Allow to manage events

WIDGET BUILDING

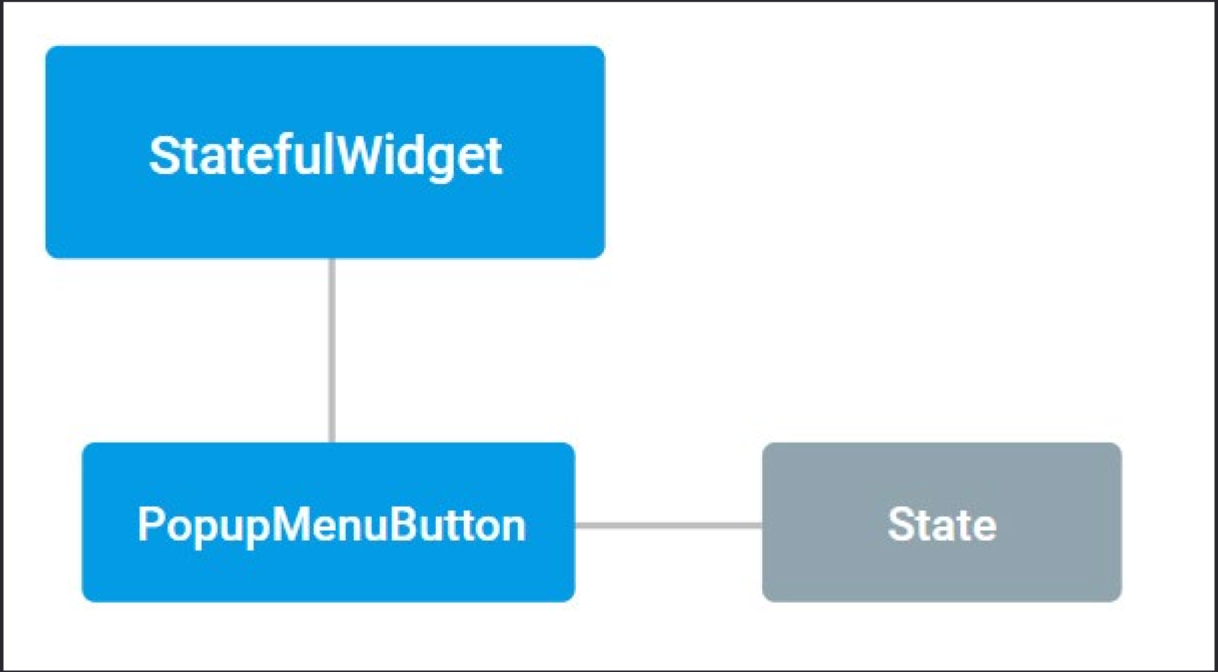
- build() method
- widgets tree definition



WIDGET : STATEFUL AND STATELESS



STATEFUL WIDGET



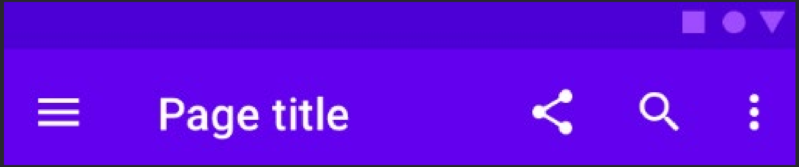
Imported methods:

- createState()
- setState()

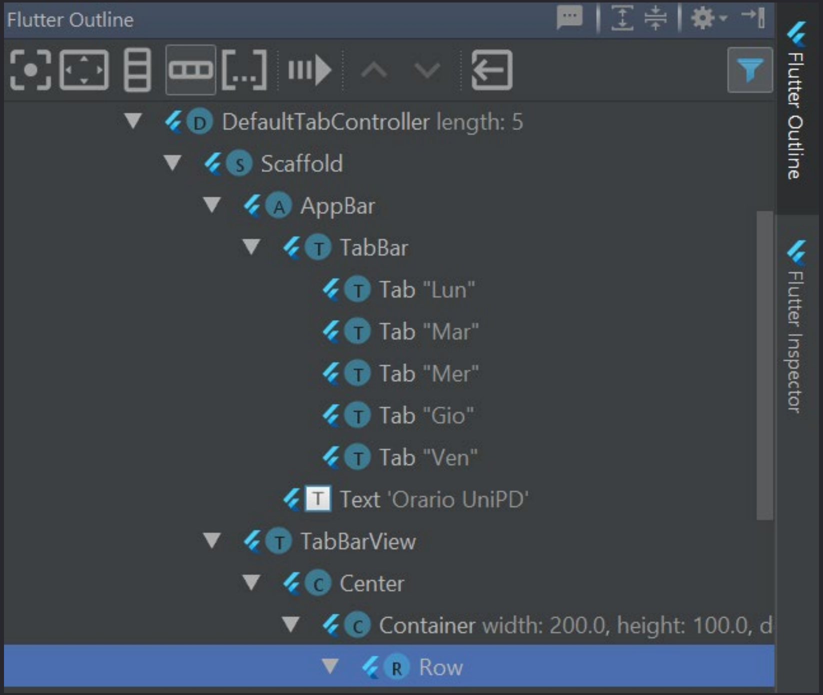
WIDGETS EXAMPLE

Flutter has a set of base widgets, the most used are

- Text
- Row
- Column
- Image
- RaisedButton
- AppBar

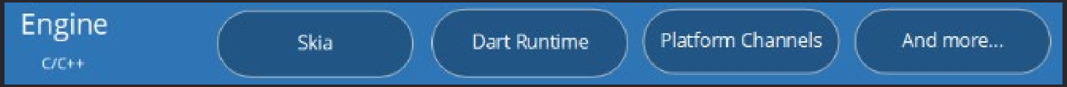


FLUTTER INSPECTOR



```
▼ DefaultTabController length: 5
  ▼ Scaffold
    ▼ AppBar
      ▼ TabBar
          Tab "Lun"
          Tab "Mar"
          Tab "Mer"
          Tab "Gio"
          Tab "Ven"
        Text 'Orario UniPD'
    ▼ TabBarView
      ▼ Center
        ▼ Container width: 200.0, height: 100.0, d
          ▼ Row
```

FLUTTER ENGINE

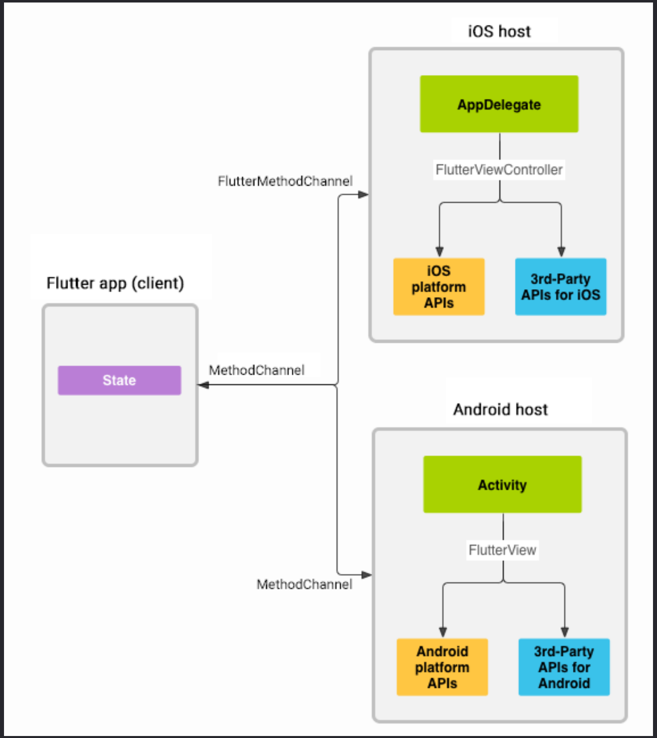


- Runtime environment written in C++
- Implements key libraries of Flutter
- Provides:
  - Dart runtime
  - Skia
  - Platform channels

PLATFORM CHANNELS

- Allow communication between Dart and specific code of each platform
- Channel types:
  - BinaryMessages
  - MessageChannel
  - MethodChannel

CODE FORKING



EXTENSIONS

- Package
- Firebase

DEVELOPMENT TOOLS CODE EXAMPLE

DEVELOPMENT TOOL

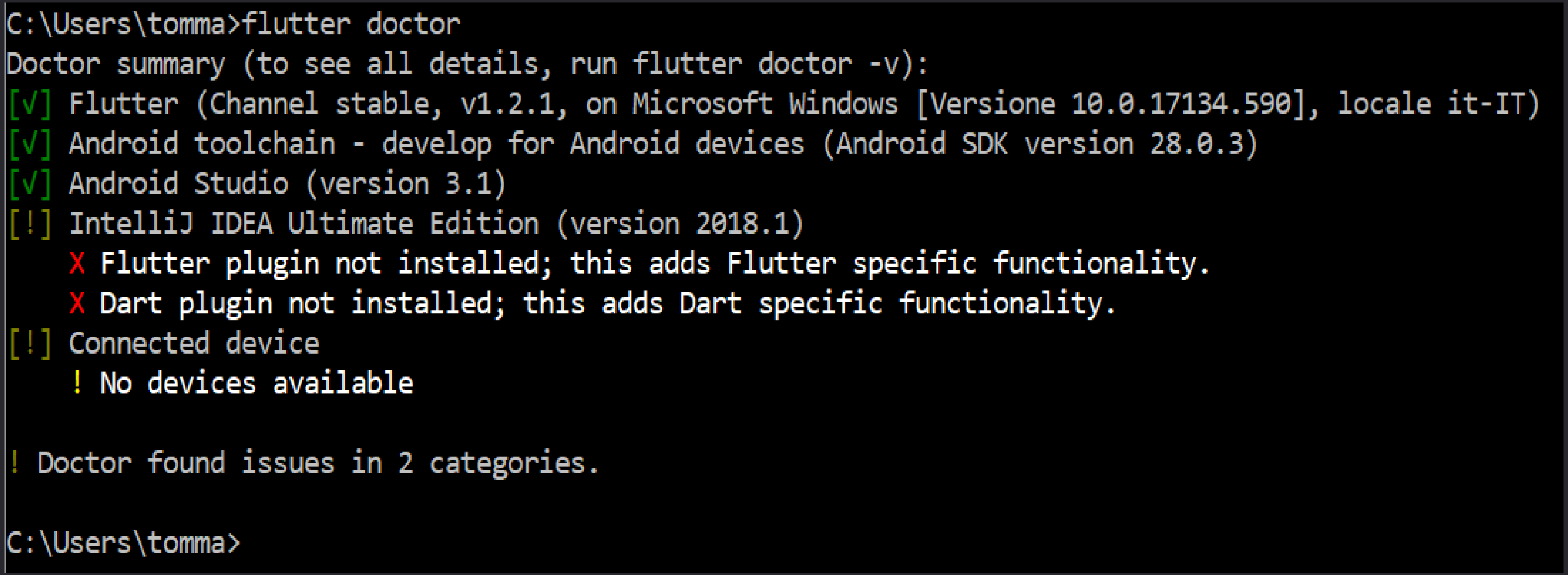
To develop Flutter applications we need:

- Flutter SDK
- An editor or IDE, suggested ones are:
  - Android Studio
  - IntelliJ IDEA
  - Visual Studio Code
- For the proposed IDE there areflutter plugins

FRAMEWORK SETUP

- It is possible to install Flutter on Windows, macOS o Linux
- Installation process:
  - SDK installation
  - PATH variable modification
  - command flutter doctor :
    - Check for missing packages

FLUTTER DOCTOR



```
C:\Users\tomma>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[√] Flutter (Channel stable, v1.2.1, on Microsoft Windows [Versione 10.0.17134.590], locale it-IT)
[√] Android toolchain - develop for Android devices (Android SDK version 28.0.3)
[√] Android Studio (version 3.1)
[!] IntelliJ IDEA Ultimate Edition (version 2018.1)
    X Flutter plugin not installed; this adds Flutter specific functionality.
    X Dart plugin not installed; this adds Dart specific functionality.
[!] Connected device
    ! No devices available

! Doctor found issues in 2 categories.

C:\Users\tomma>
```

SIMPLE PIECE OF CODE

With this simple example we will learn how to use the following components of the framework:

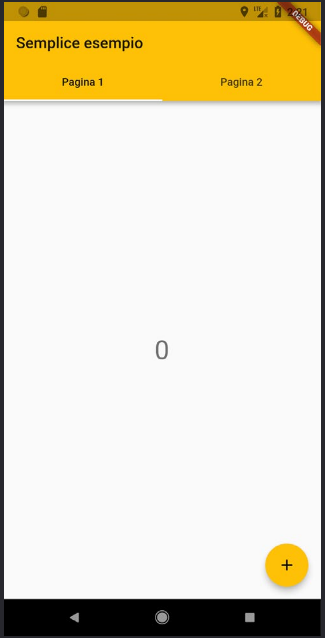
- Stateful widget
- Stateless widget
- Tabbed layout

The application has a tabbed layout with the following pages:

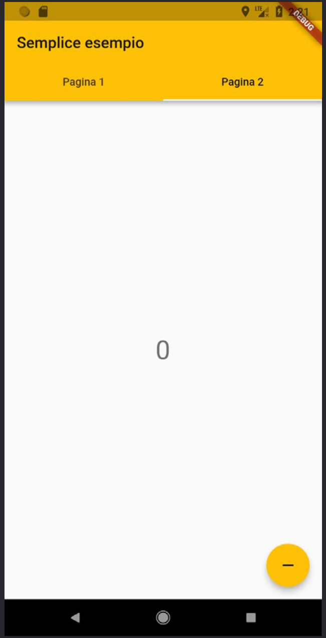
- Page 1: allows to increate a counter through button click
- Page 2: allows to decrease a counter through a button click

OUR TARGET

Page 1



Page 2



CLASSES

```
class MyApp extends StatelessWidget {...}
class FirstPage extends StatefulWidget {...}
class SecondPage extends StatefulWidget {...}
class _FirstPageState extends State<FirstPage> {...}
class _SecondPageState extends State<SecondPage> {...}
```

FIRST PAGE

```
class FirstPage extends StatefulWidget {
   FirstPage({Key key, this.title}) : super(key: key);
      final String title;
      @override
      _FirstPageState createState() =>
_FirstPageState();
}
```

STATE OF FIRST PAGE- 1

```
class _FirstPageState extends State<FirstPage> {
 int _counter1 = 0;
 void _incrementCounter() {
   setState(() {
     _counter1++;
   });
 }
```

STATE OF FIRST PAGE- 2

```
@override
 Widget build(BuildContext context) {
   return Scaffold(
     body: Center(
       child: Text(
             '$_counter1',
       ),
     ),
```

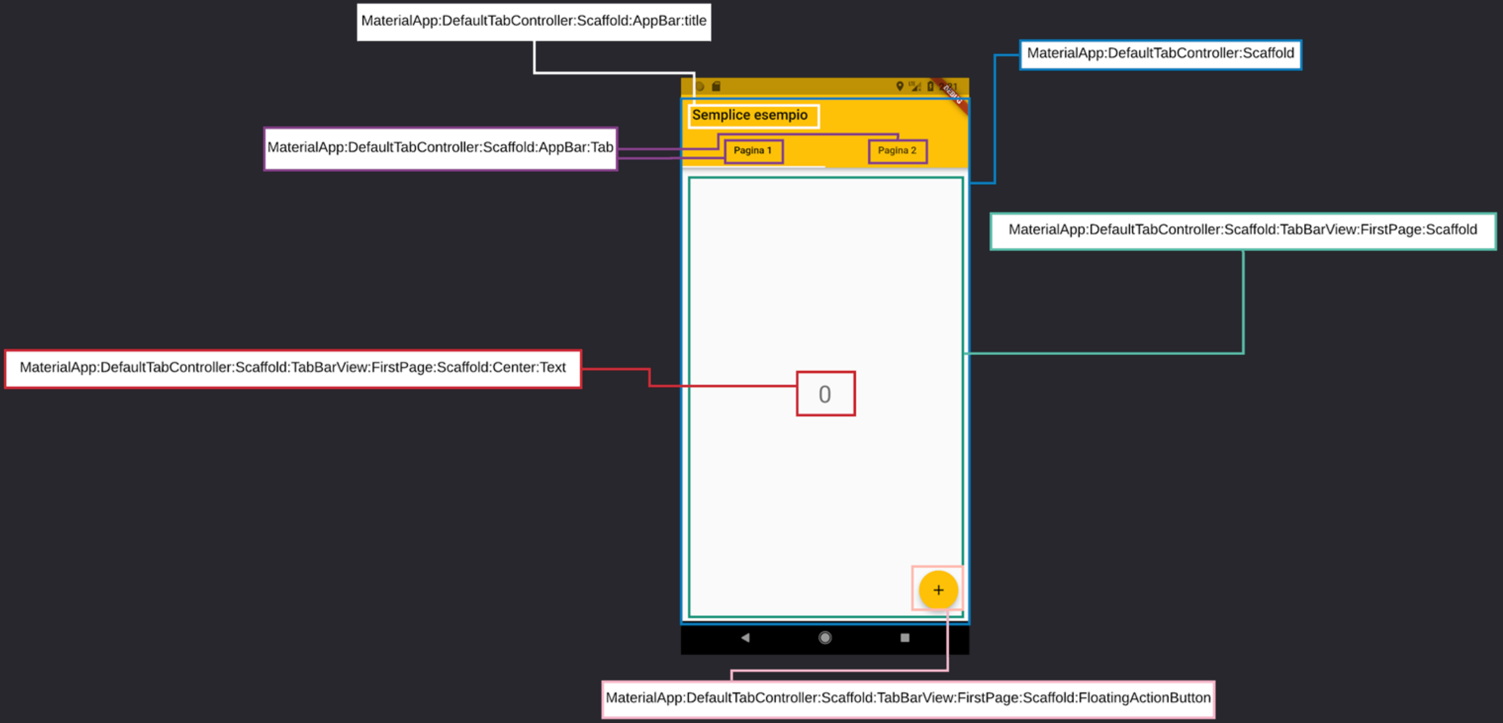
```
floatingActionButton: FloatingActionButton(
       onPressed: _incrementCounter,
       tooltip: 'Increment',
       child: Icon(Icons.add),
     ),
   );
 }
}
```

OUR APPLICATION

```
class MyApp extends StatelessWidget {
 // This widget is the root of your
application.
 @override
 Widget build(BuildContext context) {
   return MaterialApp(
     title: 'Flutter Demo',
     theme: ThemeData(
       primarySwatch: Colors.amber;
     ),
     home: DefaultTabController(
         length: 2,
```

```
child: Scaffold(
    appBar: AppBar(
        bottom: TabBar(
             tabs: [ Tab(text: ”Page 1"),
                           Tab(text: ”Page 2")]
        ),
        title: Text(”Simple example"),
     ),
     body: TabBarView(
         children: [
             FirstPage(title: ”First page"),
             SecondPage(title: ”Second page")
         ]),)),);}}
```

INTERFACE


MaterialApp:DefaultTabController:Scaffold:AppBar:title
MaterialApp:DefaultTabController:Scaffold
MaterialApp:DefaultTabController:Scaffold:AppBar:Tab
Semplice esempio
Pagina 1
Pagina 2
MaterialApp:DefaultTabController:Scaffold:TabBarView:FirstPage:Scaffold
MaterialApp:DefaultTabController:Scaffold:TabBarView:FirstPage:Scaffold:Center:Text
0
+
MaterialApp:DefaultTabController:Scaffold:TabBarView:FirstPage:Scaffold:FloatingActionButton

REFERENCES - 1

- Flutter - https://flutter.dev/
- Flutter Docs- https://docs.flutter.io/
- Dart - https://www.dartlang.org/
- Platform Channels- https://flutter.dev/docs/development/platform-integration/platform-channels
- Pro and cons of Flutter- https://hackernoon.com/flutter-pros-and-cons-for-seamless-cross-platform-development-c81bde5a4083
- Wikipedia - https://en.wikipedia.org/wiki/Flutter_(software)

REFERENCES- 2

- Flutter engine- https://github.com/flutter/engine
- Architettura Flutter - https://medium.com/flutter-community/the-layer-cake-widgets-elements-renderobjects-7644c3142401
- Flutter inspector- https://flutter.github.io/devtools/inspector
- Google SKIA- https://skia.org/