



FLUTTER

SVILUPPA APP BELLISSIME E NATIVAMENTE
COMPILATE PER MOBILE, WEB E DESKTOP

Made by 

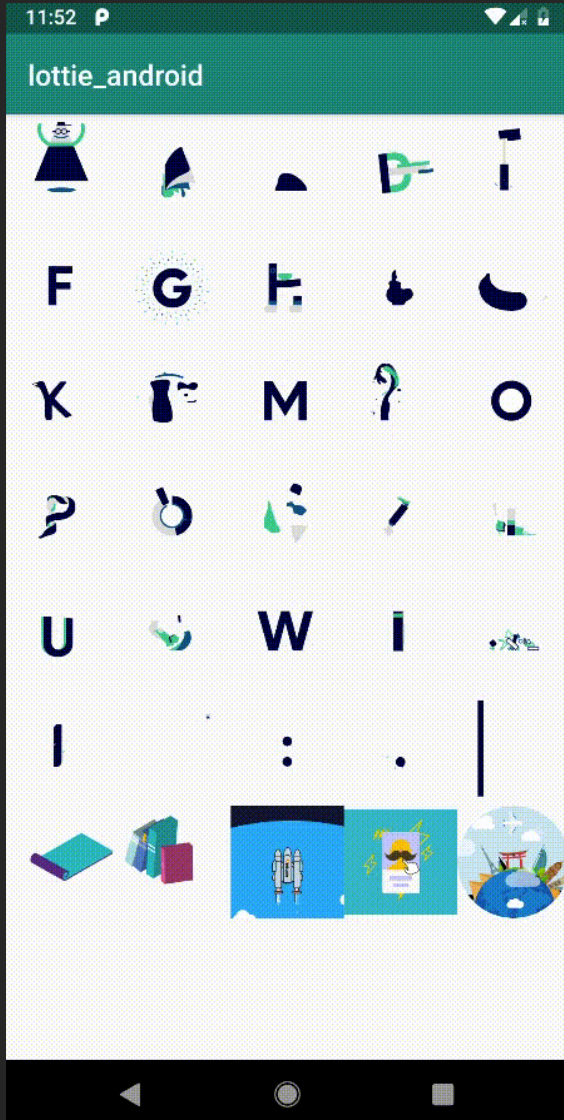
NATIVE VS HYBRID APP





FLUTTER - PERFORMANCE

NATIVE vs HYBRID APP



Android	FPS	CPU	Memory Mb	Battery mAh
Native (Android)	30	18.9	205	15.97 mAh
RN	29	15.6	280	14.80 mAh
Flutter	9	12.8	266	14.11 mWh

iOS iPhone 6s	FPS	CPU % per core in total	GPU	Memory Mb	Battery mAh
Native (iOS)	25	151	62.9	48	16.00 mAh
RN	23	72.1	65.1	134.9	18.00 mAh
Flutter	8	123	57.71	117	17.00 mAh

FLUTTER - PERFORMANCE

NATIVE vs HYBRID APP

Android	FPS	CPU	Memory Mb
Native (Android)	58	6.53	80
RN	7	8.5	424
Flutter	19	10.28	168

iOS iPhone 6s	FPS	CPU % per core in total	GPU %	Memory Mb
Native (iOS)	59	61	48.28	158
RN	59	118.6	19.8	220
Flutter	59	69	81.91	191



AZIENDE CHE UTILIZZANO FLUTTER



COS'È FLUTTER?

E' un **FRAMEWORK** open-source creato da Google per la creazione di interfacce native per iOS e Android.



SDK

L'SDK di Flutter si basa sul linguaggio di programmazione **Dart**, anch'esso sviluppato da Google.

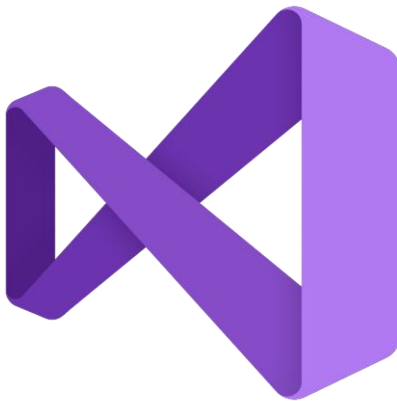


COS'È FLUTTER?

L'SDK (Software Development Kit) di Flutter è scaricabile e usufruibile da diversi ambienti di sviluppo.

Necessario:

- plugin Dart
- plugin Flutter
- variabile d'ambiente percorso SDK flutter

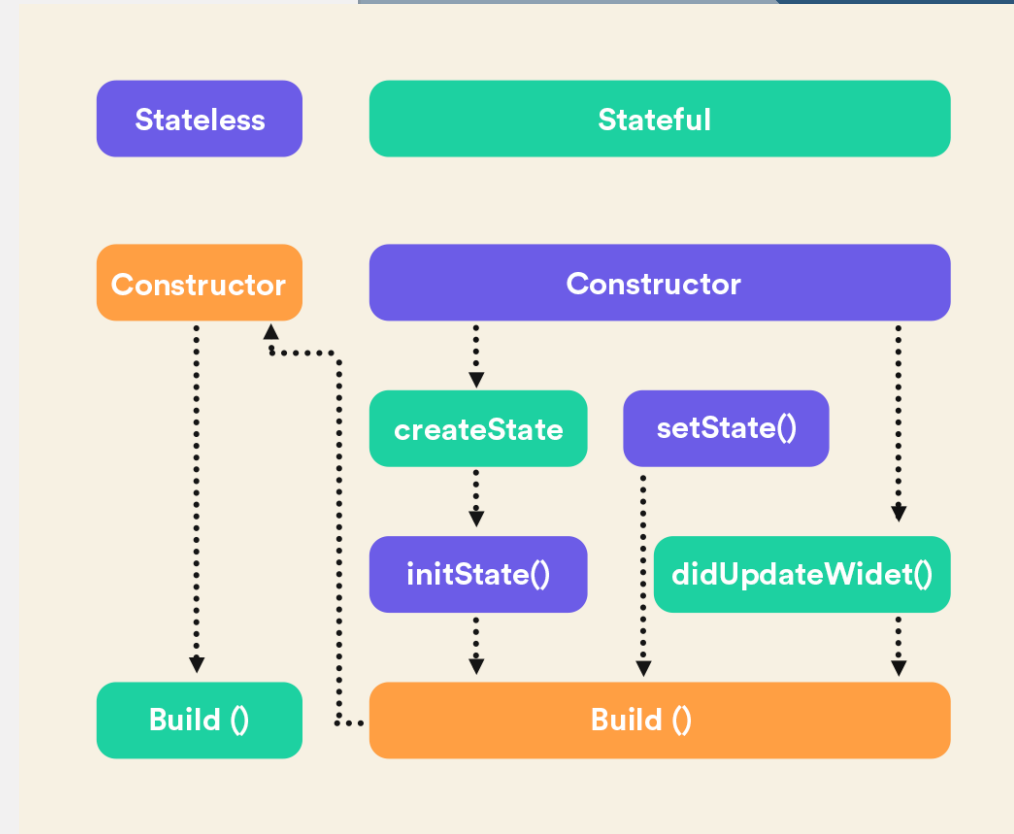
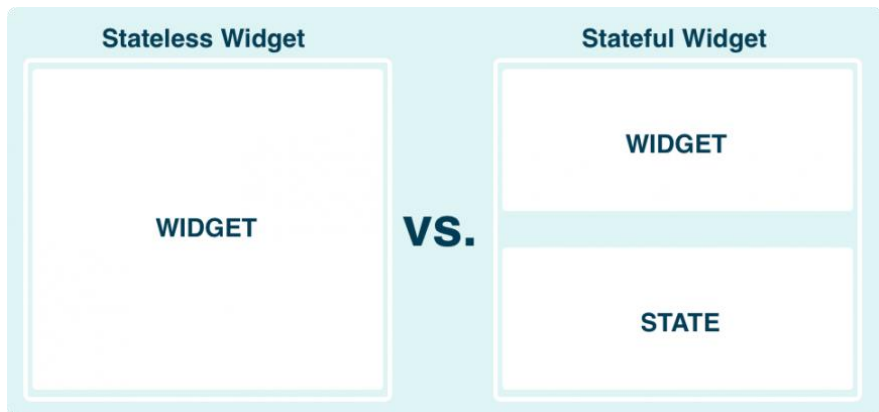


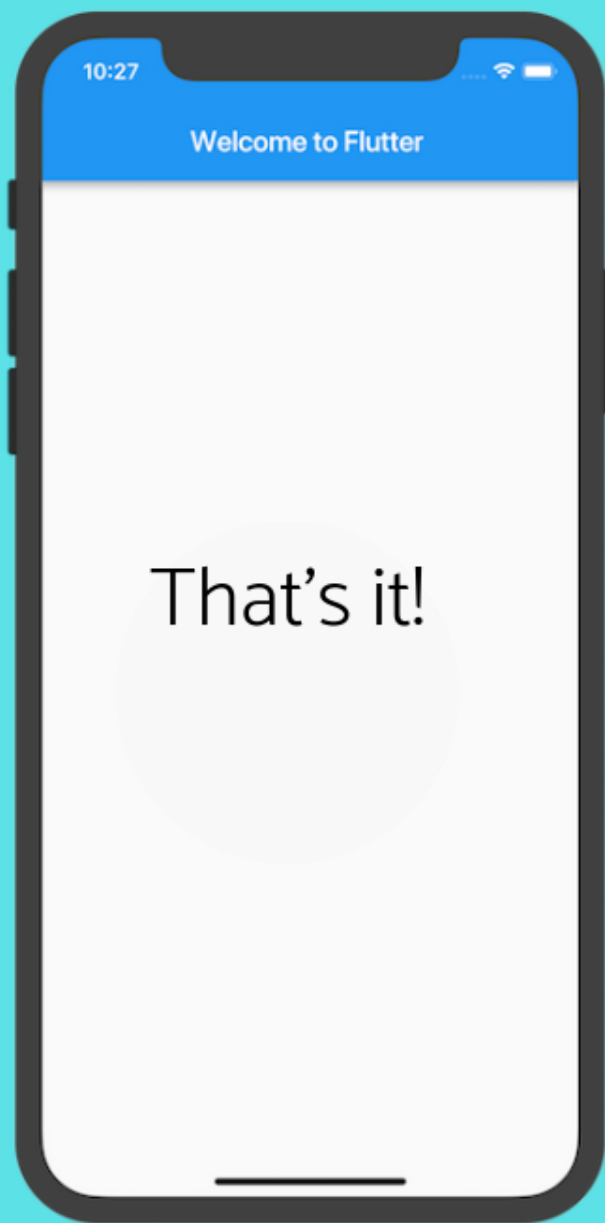
Android
Studio

TUTTO È UN WIDGET

Approccio reattivo, l'unico tuo scopo è quello definire il WIDGET e informare Flutter quando esso è cambiato.

Sarà il framework poi a decidere il modo più efficiente per aggiornare la UI tramite appositi algoritmi.





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

```
void main() => runApp(MyApp());
```

```
class MyApp extends StatelessWidget {  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      title: 'Welcome to Flutter',  
      home: Scaffold(  
        appBar: AppBar(  
          title: Text('Welcome to Flutter'),  
        ),  
        body: Center(  
          child: Text('That's it!'),  
        ),  
      ),  
    );  
  }  
}
```

COME COSTRUIRE IL CORPO DELLA TUA APP

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

Classe Scaffold – Material library
(è necessario importare la libreria `material.dart`)

Main che richiama la libreria Material e richiama la funzione dove sono presenti i widget dell'app.

```
void main() {  
  runApp(MaterialApp(home: MyApp()));  
}
```

SETSTATE

Per aggiornare variabili o aspetti grafici all'interno di un widget basterà richiamare `setState` in una funzione presente dentro una classe dove è stato creato lo stato.

```
setState(() { });
```

Al resto ci penserà il framework Flutter che andrà ad aggiornare la UI del proprio dispositivo andando a sostituire gli elementi che sono cambiati. Questa è la reale potenzialità di Flutter.

COME CREARE UNA FUNZIONE RICORRENTE

Innanzitutto è necessario importare la libreria `dart:async`

```
import 'dart:async';
```

```
void _incrementCounter() {  
  Timer.periodic(Duration(seconds:1), (timer) {  
    _counter++;  
    setState(() {  
    });  
  }); // Timer.periodic  
}
```

Per creare la funzione ricorrente dovremo richiamare:
`Timer.periodic(Duration(int), (nometimer){'codice'})`

COME IMPORTARE LE LIBRERIE ESTERNE

Cerca "nomeLibreria + flutter" su google e apri il primo risultato dal sito pub.dev

Cliccando su [Installing](#) ti verrà mostrato il nome della libreria da aggiungere al file [pubspec](#).

NB: Ricordati dopo averlo aggiunto di eseguire il comando [pub get](#) da riga di comando per aggiornare il pacchetto delle dependencies.

font_awesome_flutter 8.11.0

Published Dec 16, 2020 •  [fluttercommunity.dev](#) • Latest: [8.11.0](#) / Prerelease

[FLUTTER](#) [ANDROID](#) [IOS](#) [WEB](#)

[Readme](#) [Changelog](#) [Example](#) [Installing](#) [Versions](#) [Scores](#)

Use this package as a library

1. Depend on it

Add this to your package's pubspec.yaml file:

```
dependencies:  
  font_awesome_flutter: ^8.11.0
```


FLUTTER DOCTOR

```
Prompt dei comandi
Microsoft Windows [Versione 10.0.19042.746]
(c) 2020 Microsoft Corporation. Tutti i diritti sono riservati.

C:\Users\tomvi>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel beta, 1.25.0-8.3.pre, on Microsoft Windows [Versione 10.0.19042.746], locale en-US)
[✓] Android toolchain - develop for Android devices (Android SDK version 30.0.2)
[✓] Chrome - develop for the web
[✓] Android Studio (version 4.1.0)
[✓] IntelliJ IDEA Ultimate Edition (version 2020.3)
[✓] Connected device (2 available)

• No issues found!

C:\Users\tomvi>
```

FLUTTER BUILD APK

```
C:\Users\tomvi\AndroidStudioProjects\flutter_app>flutter build apk
```

You are building a fat APK that includes binaries for android-arm, android-arm64, android-x64.

If you are deploying the app to the Play Store, it's recommended to use app bundles or split the APK to reduce the APK size.

To generate an app bundle, run:

```
flutter build appbundle --target-platform android-arm,android-arm64,android-x64
```

Learn more: <https://developer.android.com/guide/app-bundle>

To split the APKs per ABI, run:

```
flutter build apk --target-platform android-arm,android-arm64,android-x64 --split-per-abi
```

Learn more: <https://developer.android.com/studio/build/configure-apk-splits#configure-abi-split>

```
Running Gradle task 'assembleRelease'...
```

```
Running Gradle task 'assembleRelease'... Done 41.0s
```

```
✓ Built build\app\outputs\flutter-apk\app-release.apk (15.4MB).
```

SITOGRAFIA

SITO UFFICIALE:

<https://flutter.dev/>

NATIVE vs HYBRID:

<http://www.mr-apps.com/it/blog/la-differenza-tra-app-native-app-ibride-e-web-app>

TEST PERFORMANCE:

<https://medium.com/swlh/flutter-vs-react-native-vs-native-deep-performance-comparison-990b90c11433#:~:text=Flutter%20almost%20matches%20native%20in,Native%20is%20actively%20using%20GPU>

FLUTTER SPIEGAZIONE:

<https://www.linkness.com/flutter-lo-sviluppo-delle-app-secondo-google/>

<https://www.ionos.it/digitalguide/siti-web/programmazione-del-sito-web/cose-flutter/>

RECOVERY DART:

<https://pub.dev/>

