Introduction et codelab Flutter

GDG Nantes

Qui sommes-nous?

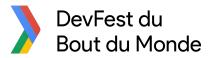


Horacio Gonzalez

Spaniard lost in Brittany, developer, dreamer and all-around geek



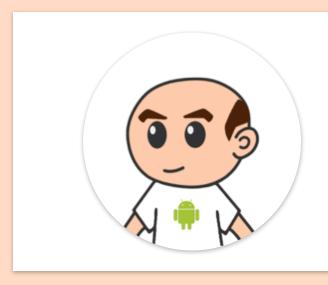






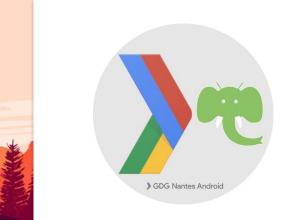


Qui sommes-nous?



Pierre Tibulle

Developer, Jobcrafter, Maker and sketchnoter!







Qui sommes-nous?



Edouard Marquez

Freelance développeur Android et Flutter





@g123k



@FlutterFrance



edouard@marquez.cool



Cofondateur du Flutter Paris



Avant de commencer...

https://flutter.dev/docs/get-started/install

https://flutter.dev/docs/get-started/editor

https://flutter.dev/docs/get-started/test-drive

En résumé, qu'est-ce que Flutter?



OpenSource

Toutes les couches



Multiplateforme

Android (4.1+) & iOS (8.0+)



Code natif

= performances



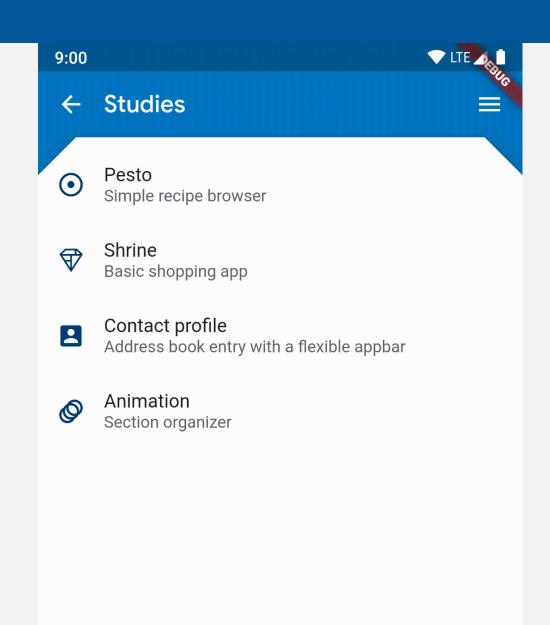
Dart

Langage de programmation



Inspiré du web

Hot Reload, concepts...



Les bases



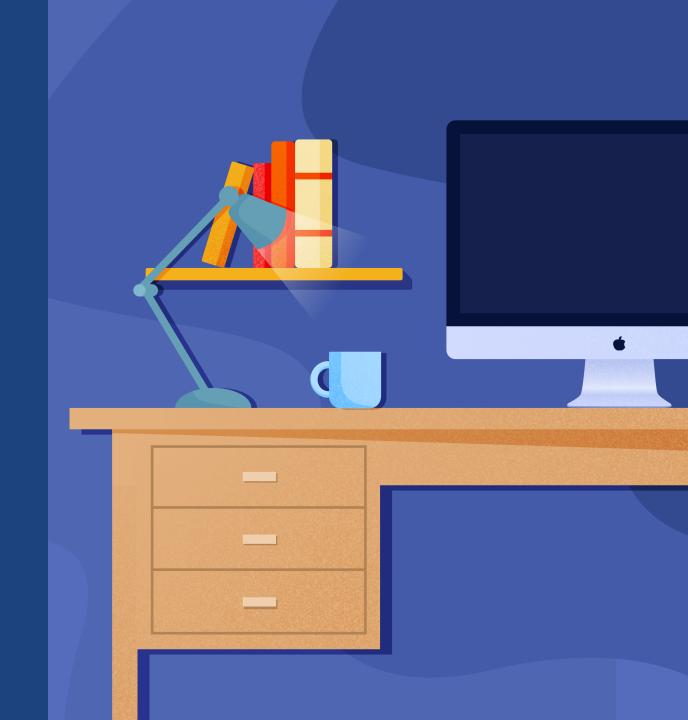
L'architecture de Flutter



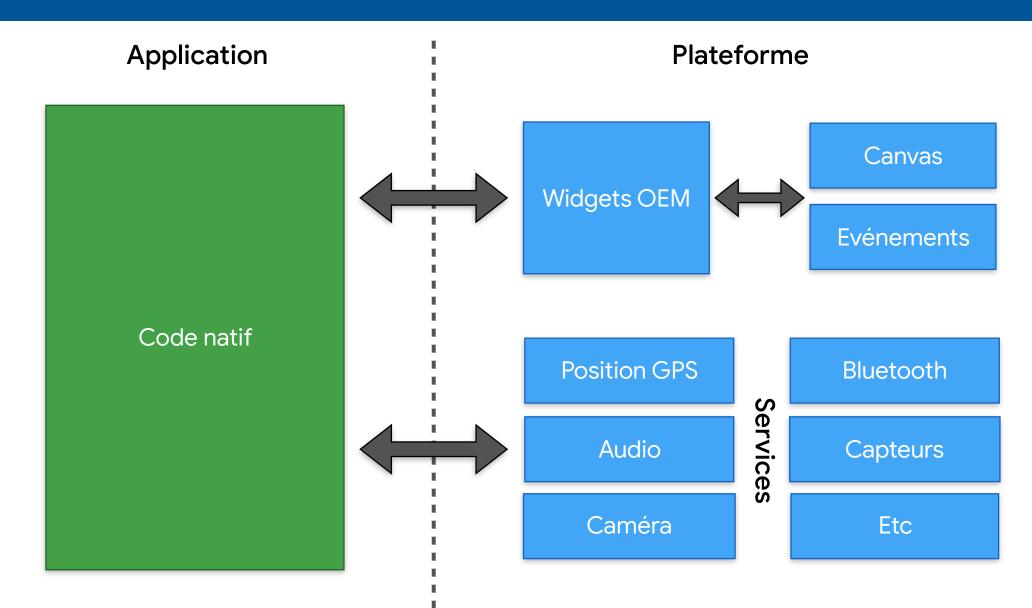
Le langage Dart



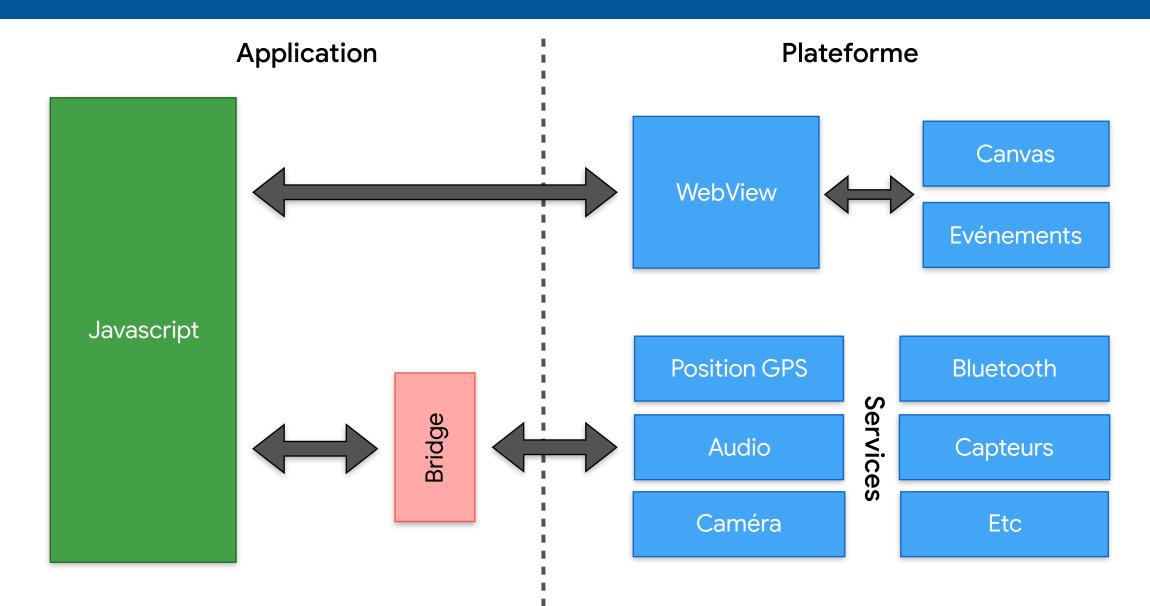
Les widgets



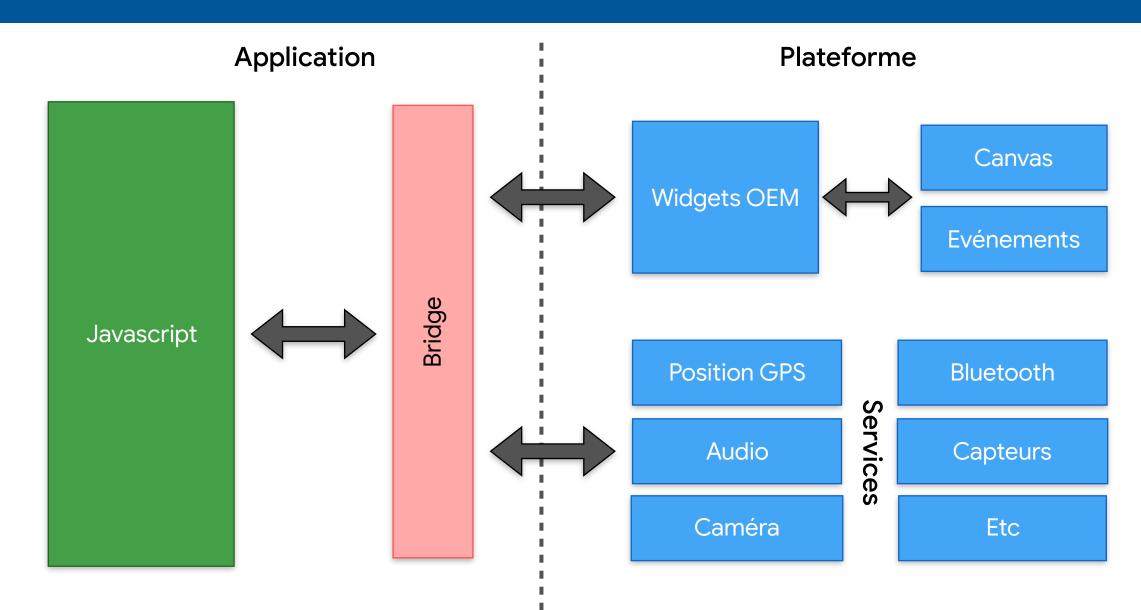
Le paradigme : application native



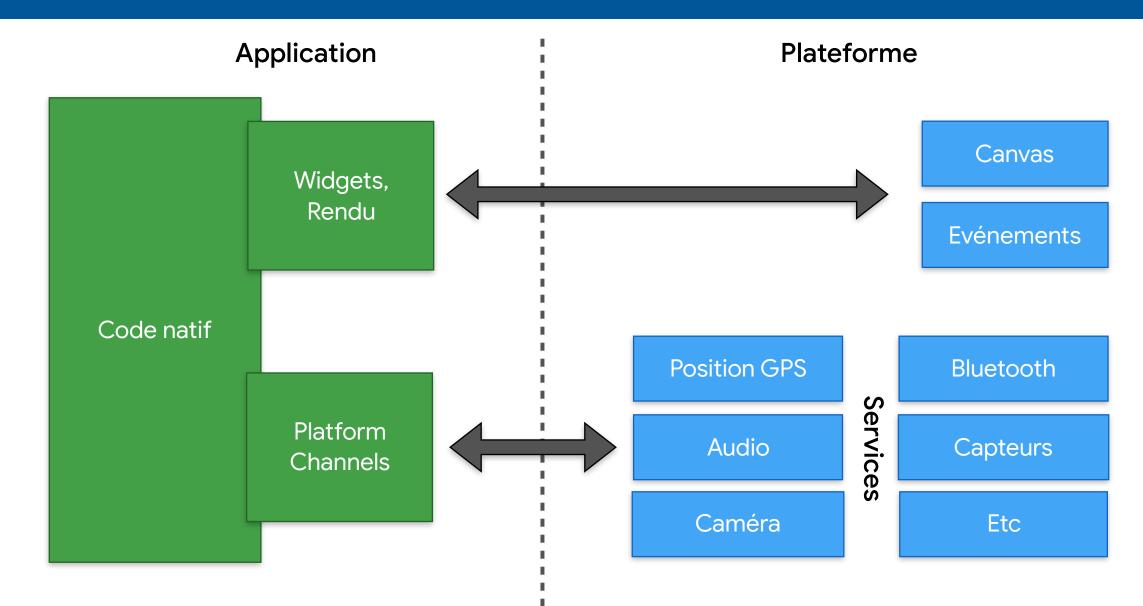
Le paradigme : WebView (Ionic, Cordova...)



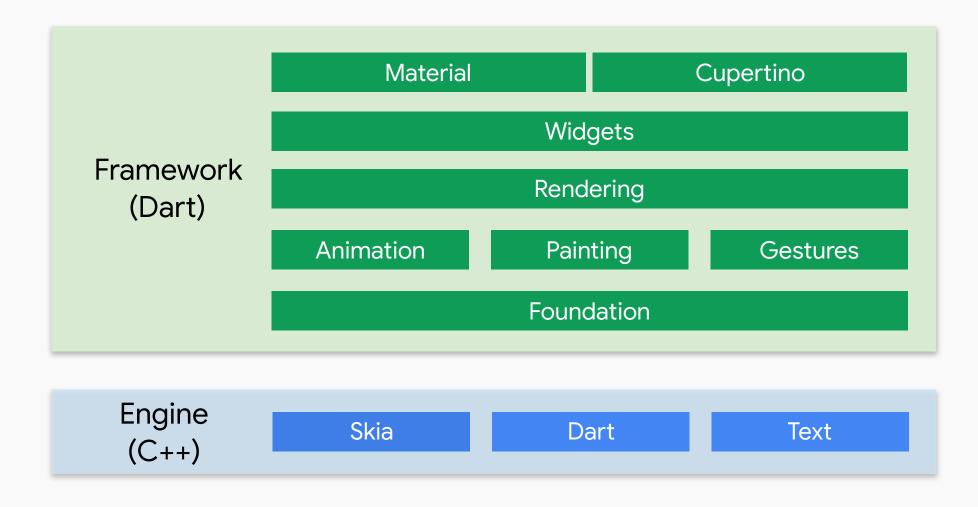
Le paradigme : type React Native



Le paradigme : Flutter



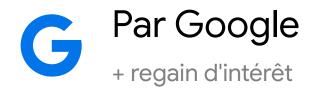
La stack Flutter





Le langage Dart

Le language Dart





Multiplateforme

Cli, web et Flutter



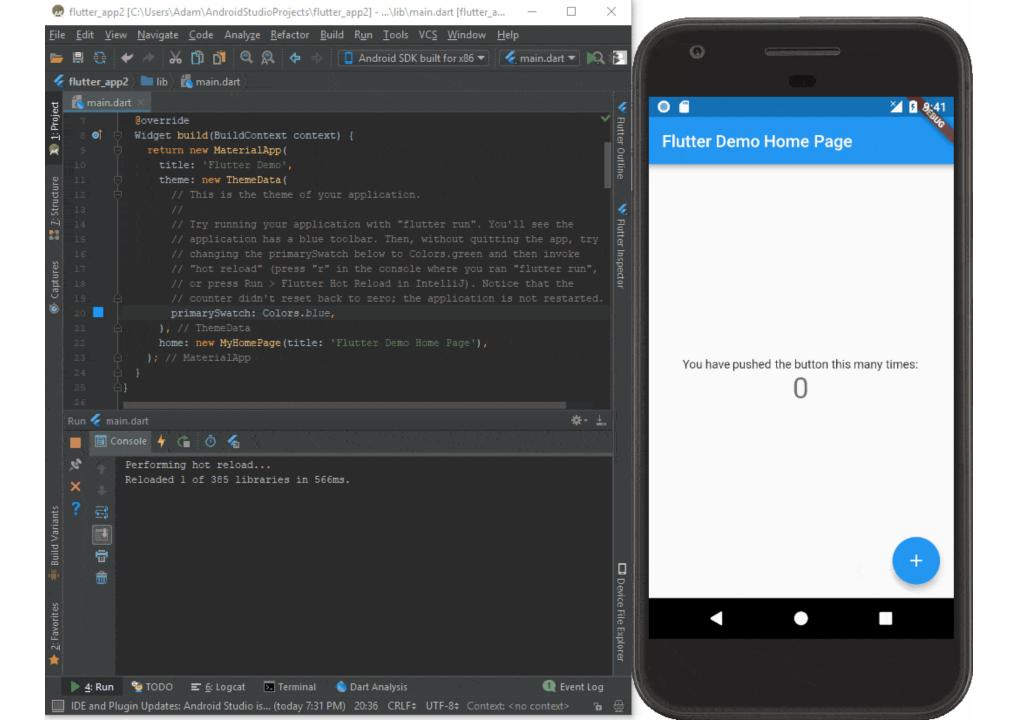






Syntaxe familière

Apprentissage court



```
class Fruit {
  final String name;
 Fruit(this.name);
 @override
 String toString() => '$name';
void myTest() {
 var fruits = [
   Fruit('Apple'),
   Fruit('Banana'),
   Fruit('Peach')
  ];
  fruits.forEach((fruit) {
   print(fruit);
 });
```

```
import 'package:http/http.dart' as http;
Future<String> getRequestAsync() async {
  var url = "https://www.google.fr";
 try {
   var response = await http.get(url);
   return response.body;
  } catch (err) {
   print(err);
   return null;
Future<String> getRequestWithoutAsyncKeyword() {
 var url = "https://www.google.fr";
  return http.get(url).then((resp) {
   return resp.body;
 }).catchError((err) {
   print(err);
   return null;
 });
```



Les widgets

"

Tout est un widget





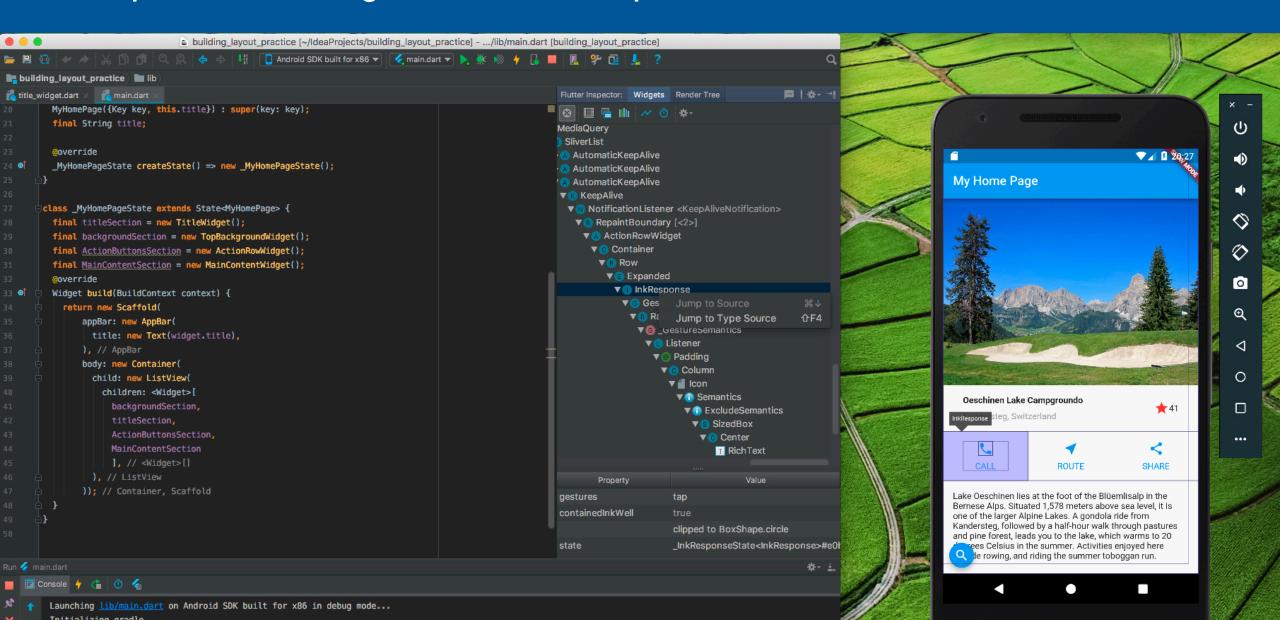
Composition over inheritance

Quizz

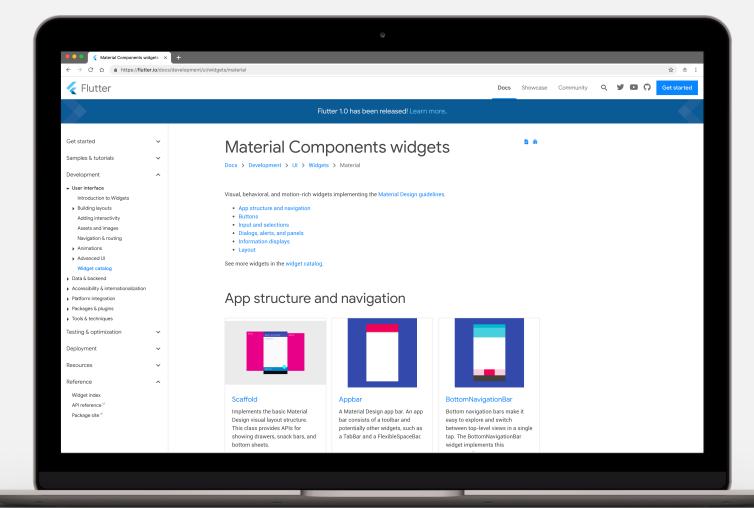
Comilien y a-t-il de vidgets affichés sur cet écran?

You have pushed the button this many times:

Comprendre les widgets : le Flutter Inspector à la rescousse



Liste des widgets

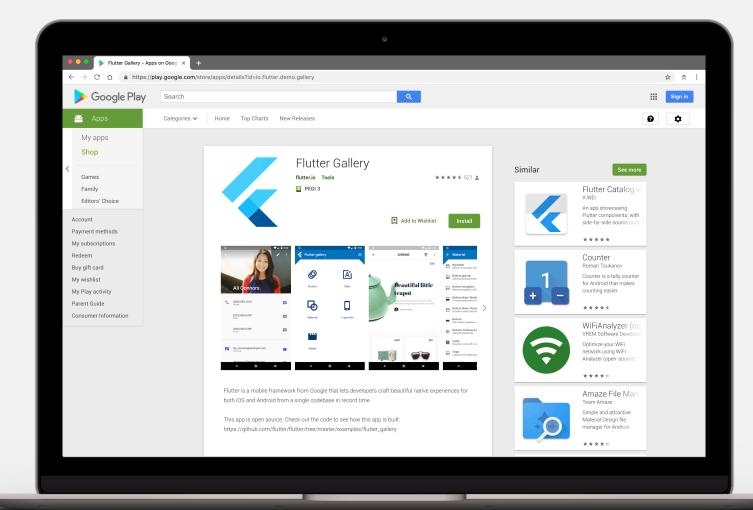




flutter.dev

Widgets catalog

Application Flutter Gallery

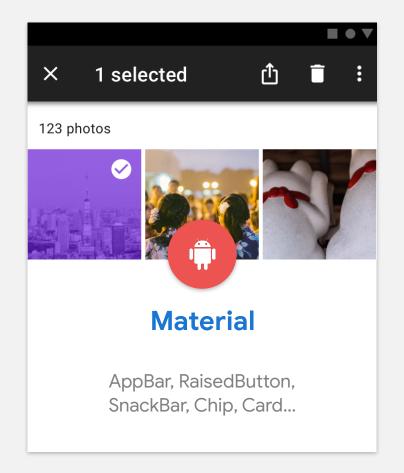


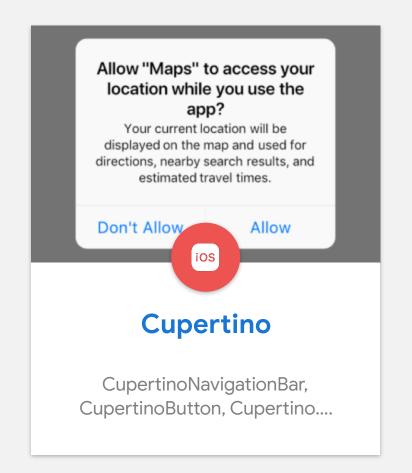


Flutter Gallery Google Play uniquement

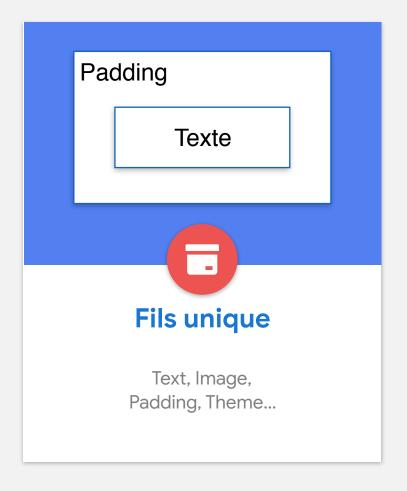
Liste des widgets







Liste des layouts / containers





```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return <a href="Center">Center</a>(
      child: Column(
        mainAxisSize: MainAxisSize.min,
         children: <Widget>[
           Icon(Icons.people, size: 40.0),
           Text('Bonjour Nantes'),
```



Avant Dart 2.3

```
• • •
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    List<Widget> widgets = [];
    if (myCondition) {
      widgets.add(const Icon(Icons.people, size: 40.0));
    if (myOtherCondition) {
      widgets.add(const Text('Bonjour Nantes'));
    return Material(
      child: Center(
        child: Column(
            mainAxisSize: MainAxisSize.min,
            children: widgets
```

Depuis Dart 2.3

```
class MyWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Material(
      child: Center(
        child: Column(
          mainAxisSize: MainAxisSize.min,
          children: <Widget>[
            if (myCondition) const Icon(Icons.people, size: 40.0),
            if (myOtherCondition) const Text('Bonjour Nantes')
```



Stateless / Statefull

Quelle différence entre StatelessWidget et StatefulWidget?





```
class MyWidget extends StatelessWidget {
   @override
   Widget build(BuildContext context) {
        return Text('Bienvenue !');
```

```
class MyWidget extends StatefulWidget {
   @override
    State createState() {
        return MyWidgetState();
```

StatefulWidget -> State

```
• • •
class MyWidgetState extends State<MyWidget> {
  int items = 0;
  @override
  Widget build(BuildContext context) {
    return Column(children: <Widget>[
      Text('Vous avez $items item(s)'),
      RaisedButton(
        child: Text('Ajouter un item au panier'),
        onPressed: _onPressed)
    ]);
  void _onPressed() {
    setState() { items++; }
```



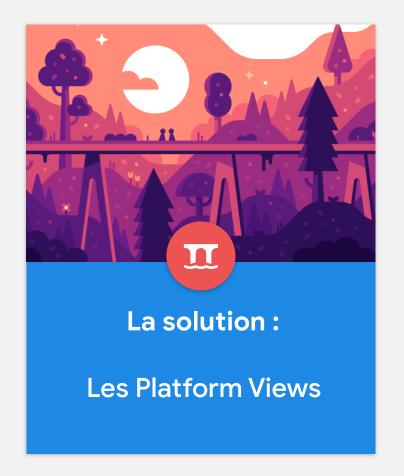
Les platform views

Flutter dessine tout ce qui est à l'écran

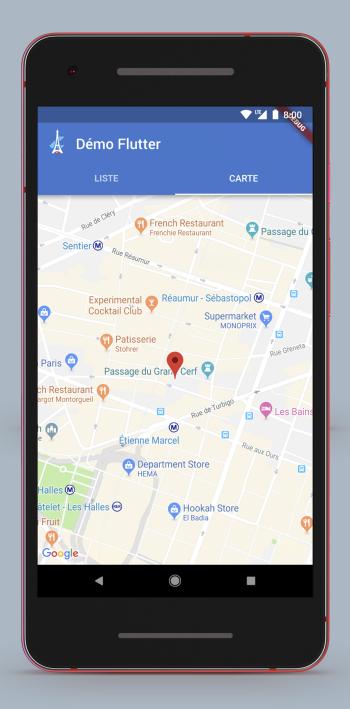
Flutter doit parfois faire des compromis







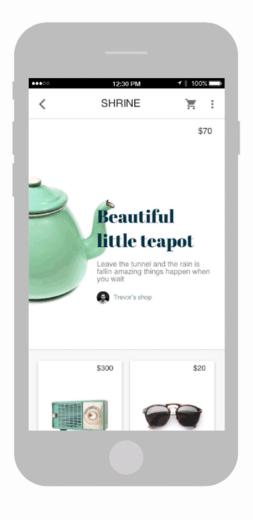
```
class MyMapWidget extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return SizedBox.expand(
        child: GoogleMap(
      onMapCreated: _onMapCreated(),
      options: GoogleMapOptions(
          myLocationEnabled: false,
          rotateGesturesEnabled: false,
          scrollGesturesEnabled: true,
          zoomGesturesEnabled: true)
    ));
  void _onMapCreated(GoogleMapController controller) {
    controller.addMarker(MarkerOptions(
        position: LatLng(48.864716, 2.349014),
        icon: BitmapDescriptor.defaultMarker));
```

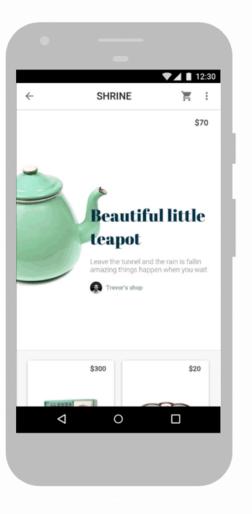




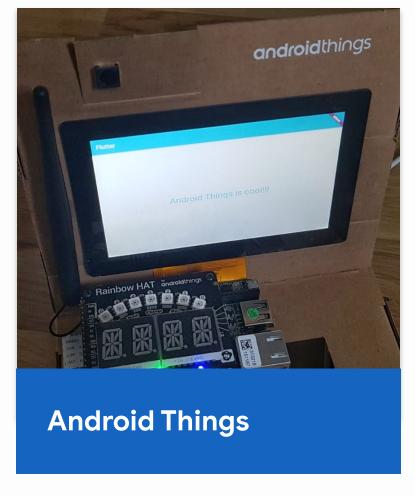
Les plateformes supportées

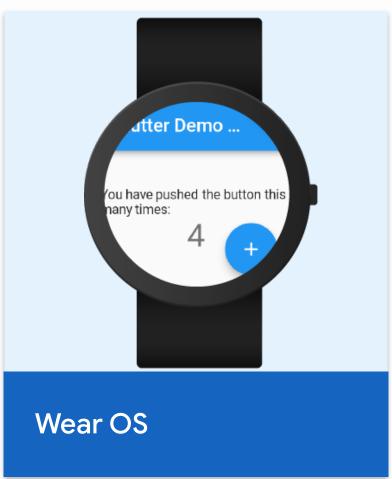
Côté mobile : Android et iOS





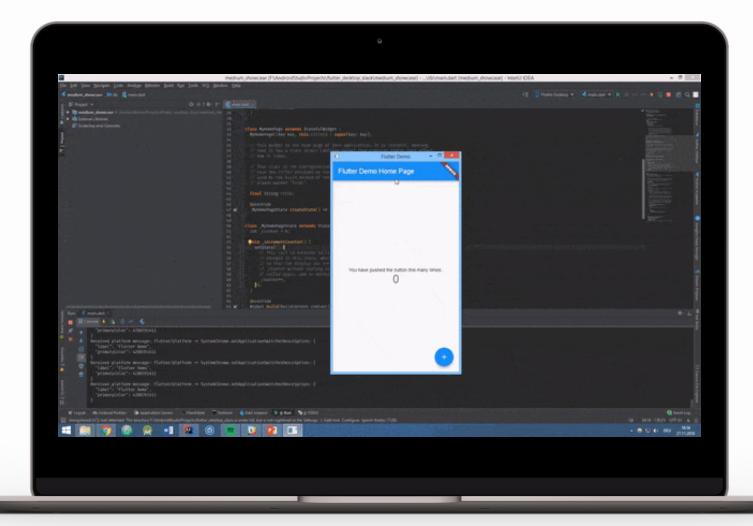
Et plus généralement, tout ce qui fait tourner des apps Android







Sur les ordinateurs





Desktop Embedding for Flutter

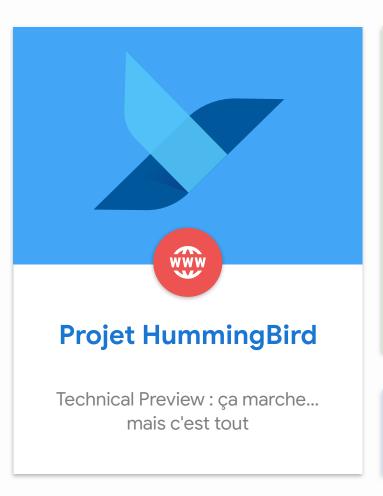
google/flutter-desktop-embedding

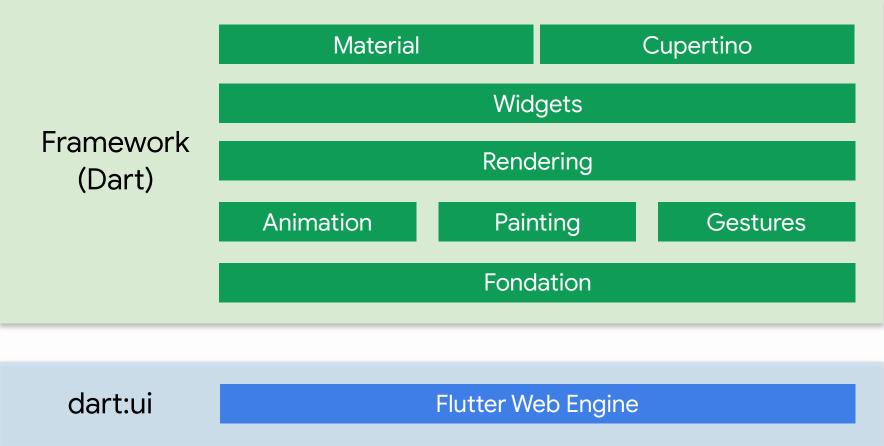


Go Flutter desktop embedder

Drakirus/go-flutter-desktop-embedder

Et même pour le web!







Quelques conseils

Un plugin pour IntelliJ Idea (Rainbow Brackets), Visual Studio Code (Bracket Pair Colorizer)

```
amain.dart
       import 'package:flutter/material.dart';
       void main() => runApp(new MyApp());
       class MyApp extends StatelessWidget {
         @override
        Widget build(BuildContext context) {
           return new MaterialApp(
             title: 'Flutter Demo'.
             theme: new ThemeData(
              primarySwatch: Colors.blue,
             home: new MyHomePage(title: 'Flutter Demo Home Page'),
           ); // MaterialApp
      class MyHomePage extends StatefulWidget {
        MyHomePage({Key key, this.title}) : super(key: key);
         final String title;
         @override
         _MyHomePageState createState() => new _MyHomePageState();
      class _MyHomePageState extends State<MyHomePage> {
         int _counter = 0;
        void _incrementCounter() {
          setState(() {
             _counter++;
           });
```

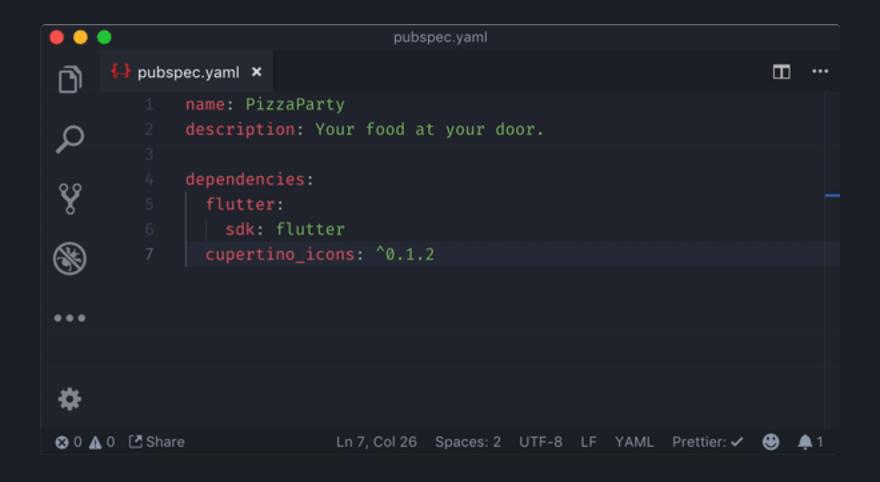
```
tamain.dart
      import 'package:flutter/material.dart';
      void main() => runApp(new MyApp());
      lagclass MyApp \mathsf{extends} StatelessWidget \{
        @override
        Widget build(BuildContext context) {
          return new MaterialApp(
            title: 'Flutter Demo',
            theme: new ThemeData
              primarySwatch: Colors.blue.
           ), // ThemeData
            home: new MyHomePage(title: 'Flutter Demo Home Page'),
          ); // MaterialApp
     MyHomePage({Key key, this.title}) : super(key: key);
        final String title;
        @override
        _MyHomePageState createState() => new _MyHomePageState();
      class _MyHomePageState extends State<MyHomePage> {
        int counter = 0:
        void _incrementCounter() {
          setState(() {
            _counter++;
```

Raccourcis (live templates)

Stful: Créer un StatefulWidget

Stless: Créer un StatelessWidget

Pubspec Assist (VS Code uniquement): trouver facilement des dépendances



Flutter Pub Version Checker (IntelliJ uniquement)

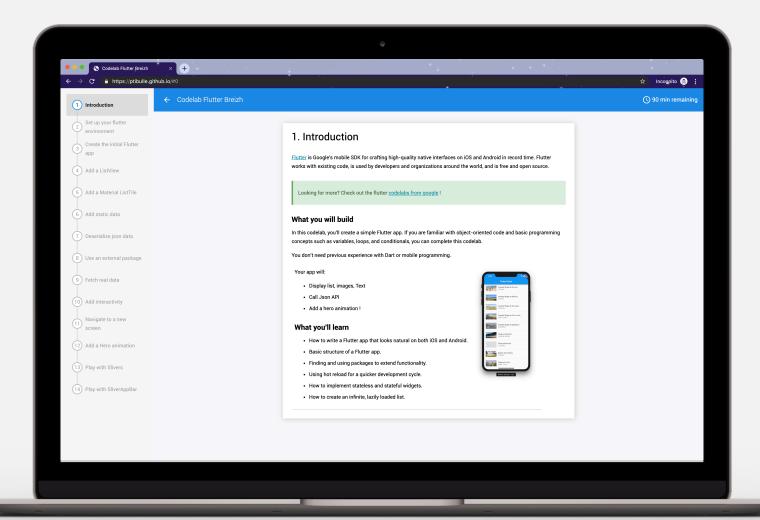
```
environment:
  sdk: ">=2.1.0 <3.0.0"
dependencies:
  flutter:
    sdk: flutter
  flutter_localizations:
                                                                                             dev_dependencies:
    sdk: flutter
                                                                                               flutter_test:
                                                                                                 sdk: flutter

glutter_redux: ^0.5.3
                                                                                               test: ^1.5.1+1
  flutter_redux_dev_tools: ^0.2.6+3
                                                                                               $\text{guild_runner: \gamma1.4.0}
  shared_preferences: ^0.4.2
                                                                                               json_serializable: ^2.3.0
  cuparting icons: ^0 1 2
                                                                                              Version 2.3.0 is different than the latest 3.0.0 more... (%F1)
Version 0.4.2 is different than the latest 0.5.2 more... (無F1)
                                                                                              r riid roccoming addition is apoditive to rea
                                                                                             flutter:
  flutter_test:
                                                                                               uses-material-design: true
    sdk: flutter
                                                                                               assets:
                                                                                                 - assets/
flutter:
                                                                                               fonts:
  uses-material-design: true
```



3.. 2.. 1!

C'est à vous!





ptibulle.github.io
Codelab

Liens utiles



Slack francophone

slack.flutter-france.fr



YouTube Flutter Paris

youtube.flutter.paris



Cours sur Udacity

bit.ly/study-jam-udacity



Codelabs officiels

codelabs.google.com



Awesome Flutter

Solido/awesome-flutter



Newsletter

flutterweekly.net

Les plateformes



La communication
Dart <-> plateforme



Intégrer Flutter à une app existante



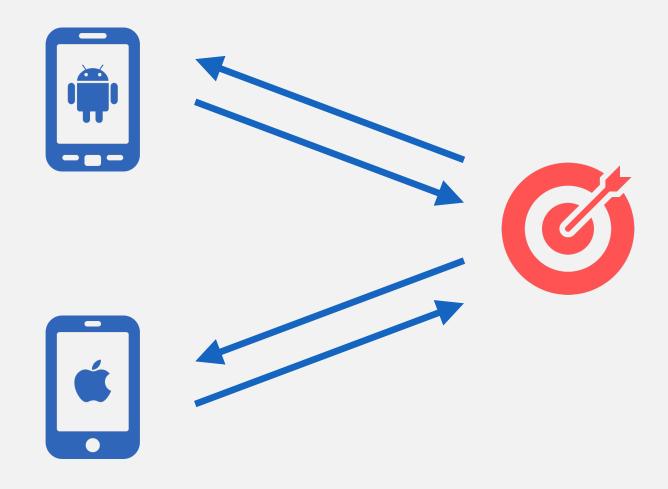
Les platform views



Les plateformes supportées



La communication entre code Dart et les plateformes



MethodChannel: transmettre une information

Code Flutter

```
static const MethodChannel channel = const MethodChannel('g123k.example/battery');
Future<String> getBatteryLevel() {
  try {
      int level = await channel.invokeMethod('getBatteryLevel');
      return 'Battery level $level';
  } on PlatformException {
      return 'Unknown battery level';
```

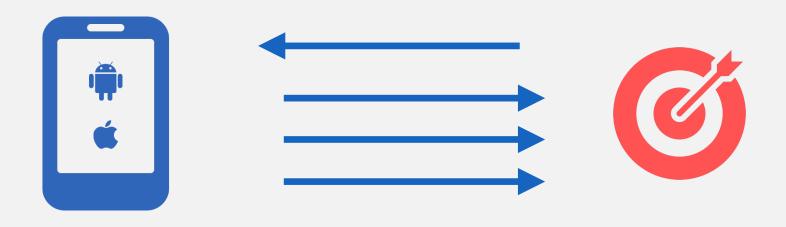
Code Android (Java)

```
new MethodChannel(getFlutterView(), "g123k.example/battery")
  .setMethodCallHandler(new MethodCallHandler() {
    public void onMethodCall(MethodCall call, Result result) {
      if (call.method.equals("getBatteryLevel")) {
        int batteryLevel = getBatteryLevel();
        if (batteryLevel != −1) {
            result.success(batteryLevel);
        } else {
            result.error("UNAVAILABLE", "Battery level not available.", null);
      } else {
          result.notImplemented();
```

Code iOS (Objective C)

```
[batteryChannel setMethodCallHandler:^(FlutterMethodCall* call, FlutterResult result) {
  if ([@"getBatteryLevel" isEqualToString:call.method]) {
    int batteryLevel = [self getBatteryLevel];
    if (batteryLevel == -1) {
        result([FlutterError errorWithCode:@"UNAVAILABLE"
       message:@"Battery info unavailable" •details:nil]);
   } else {
        result(@(batteryLevel));
 } else {
    result(FlutterMethodNotImplemented);
}];
```

La communication entre code Dart et les plateformes



EventChannel: transmettre un flux d'informations



Mélanger natif/Flutter

Possibilité d'intégrer Flutter à une application existante













Mélanger natif/Flutter

S'intégrer à une app existante



Points d'entrée Flutter

On peut lancer une zone du code