Flutter

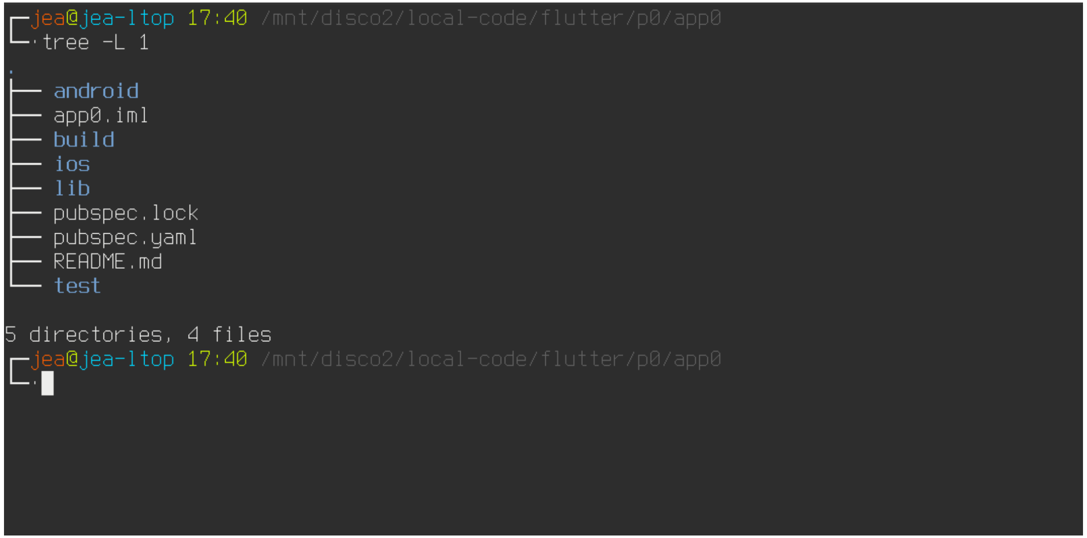
BUILD ONCE
DEPLOY EVERYWHERE!

MAIN FEATURES

- UI Framework: One Codebase, many Os’s.
- Mobile: Ios, Android
- Desktop: Linux, Web, MacOs, Windows
- Reactive Programming Paradigm
- Modern
- Supported by Google
- Awesome interfaces/GUI
- High Performace
- Used by Alibaba, Tencent, Square, Ebay, BMW, CapitalOne, Nubank
- Stock of Widgets (stateful)
- Dart Programming Language

APP TREE

Flutter create <app name>



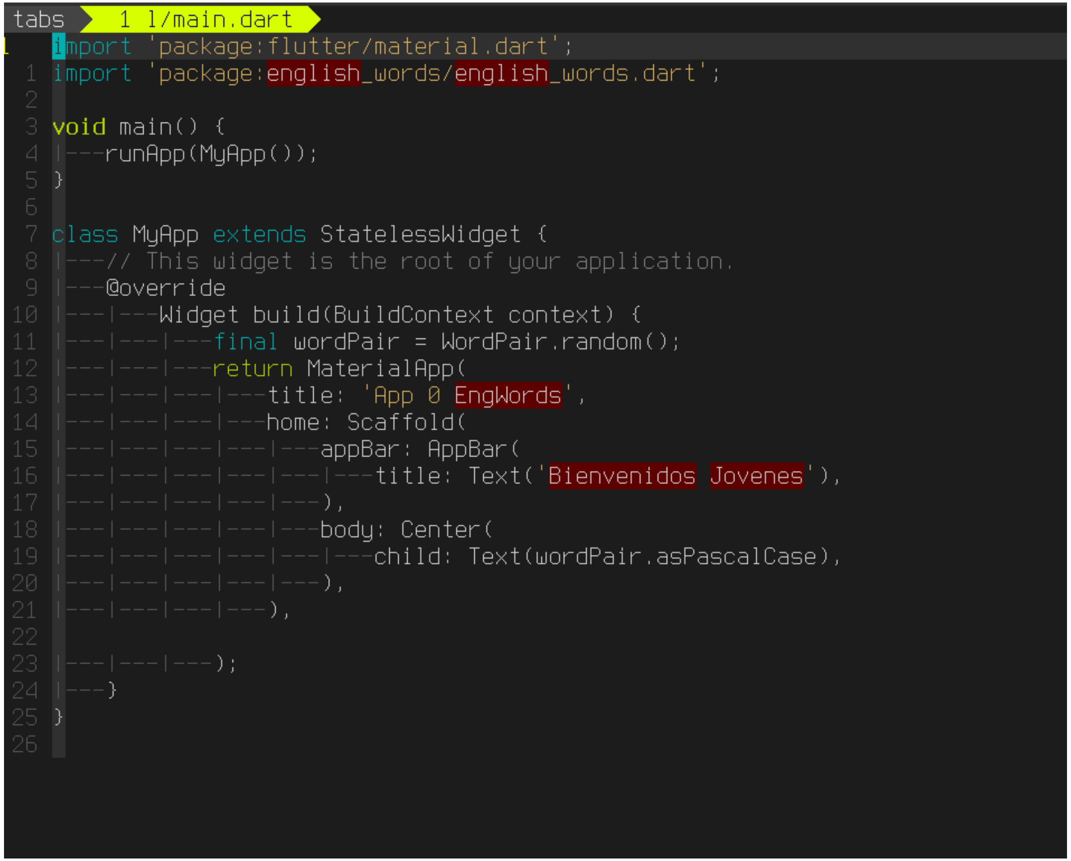
```
┌─jea@jea-ltop 17:40 /mnt/disco2/local-code/flutter/p0/app0
└─› tree -L 1
.
├── android
├── app0.iml
├── build
├── ios
├── lib
├── pubspec.lock
├── pubspec.yaml
├── README.md
└── test

5 directories, 4 files
┌─jea@jea-ltop 17:40 /mnt/disco2/local-code/flutter/p0/app0
└─› █
```

Importar paquetes

Ejecucion app

Widget



```
tabs   1 l/main.dart
import 'package:flutter/material.dart';
import 'package:english_words/english_words.dart';

void main() {
|---runApp(MyApp());
}

class MyApp extends StatelessWidget {
|---// This widget is the root of your application.
|---@override
|---|---Widget build(BuildContext context) {
|---|---|---final wordPair = WordPair.random();
|---|---|---return MaterialApp(
|---|---|---|---title: 'App 0 EngWords',
|---|---|---|---home: Scaffold(
|---|---|---|---|---appBar: AppBar(
|---|---|---|---|---|---title: Text('Bienvenidos Jovenes'),
|---|---|---|---|---),
|---|---|---|---|---body: Center(
|---|---|---|---|---|---child: Text(wordPair.asPascalCase),
|---|---|---|---|---),
|---|---|---|---),

|---|---|---);
|---}
}
```

Widget

Widget

Widget

WIDGETS

Everything is a widget!

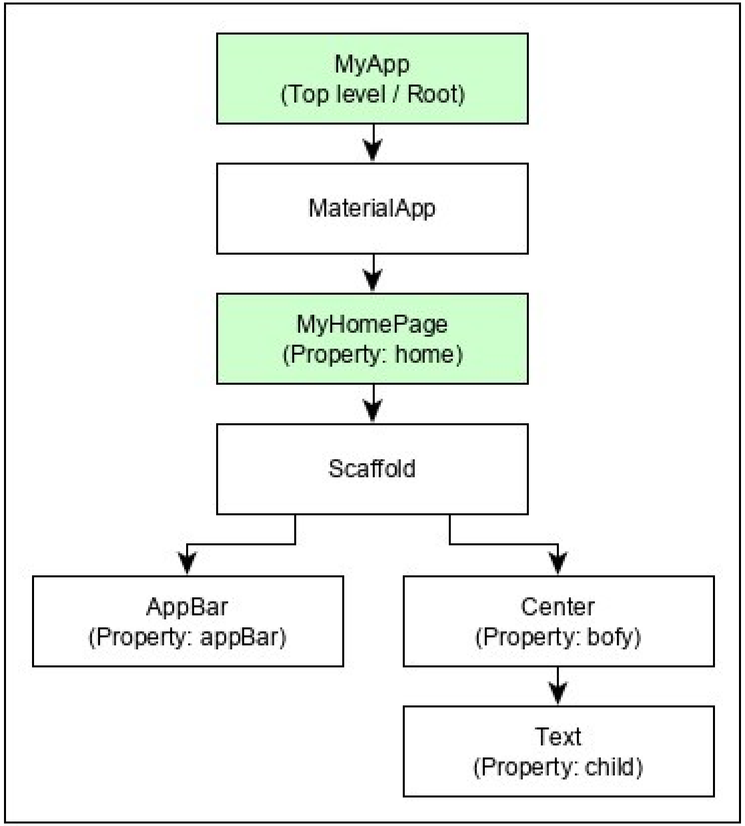
Widget

User Interface Components → Lego pieces for UI

State

Give a state, when changes, render just necessary changes. (Reactive Programming). *StatefulWidget*

Widgets compose widgets.



DART

```
void main() {
    print(“Hola Luke soy Darth Vader, tu padre”);
}
// variables
final name = “Darth”;
const lastName = “Vader”;
final occupation = “bad guy”;
var age = 175;
var state = “alive” // 1 // true

// data Types
Numbers : 55, 175.05
Strings : “Soy un Sith”
Booleans : true, false
Lists : var enemies = [‘yoda’, ‘Obi Wan Kenobi’, ‘Sidious’];
Maps : var sons = {‘name’: ‘Luke Skywalker’, ‘name’: ‘Princess Leia’};
```

DART

```
void main() {
    print(“Hola Luke soy Darth Vader, tu padre”);
}
// loops
for(var i = 1; i < = 10; i++) {
    print(i);
}

// functions
void main() {
    sumar(5,3);
}

void sumar(int a, int b) {
    int c;
    c = a + b;
    print(c);
}
```

DART O.O.P.

```
class Warrior {
    String nombre;

// getter
    String get warrior_name {
        return name;
    }

// setter
    void set warrior_name(String nombre) {
        this.nombre = nombre;
    }
// funcion o metodo
    void fight() {
        print(nombre)
    }
}
```

DART O.O.P.

```
void main {
    // crear objeto
    Warrior sith = new Warrior();
    sith.nombre = “Darth Vader”;
    sith.fight();
}
```