Flutter

Mobile Application Development 2020

Munich University of Applied Sciences

A presentation by Mathias Weinstabl and Tobias Stadler

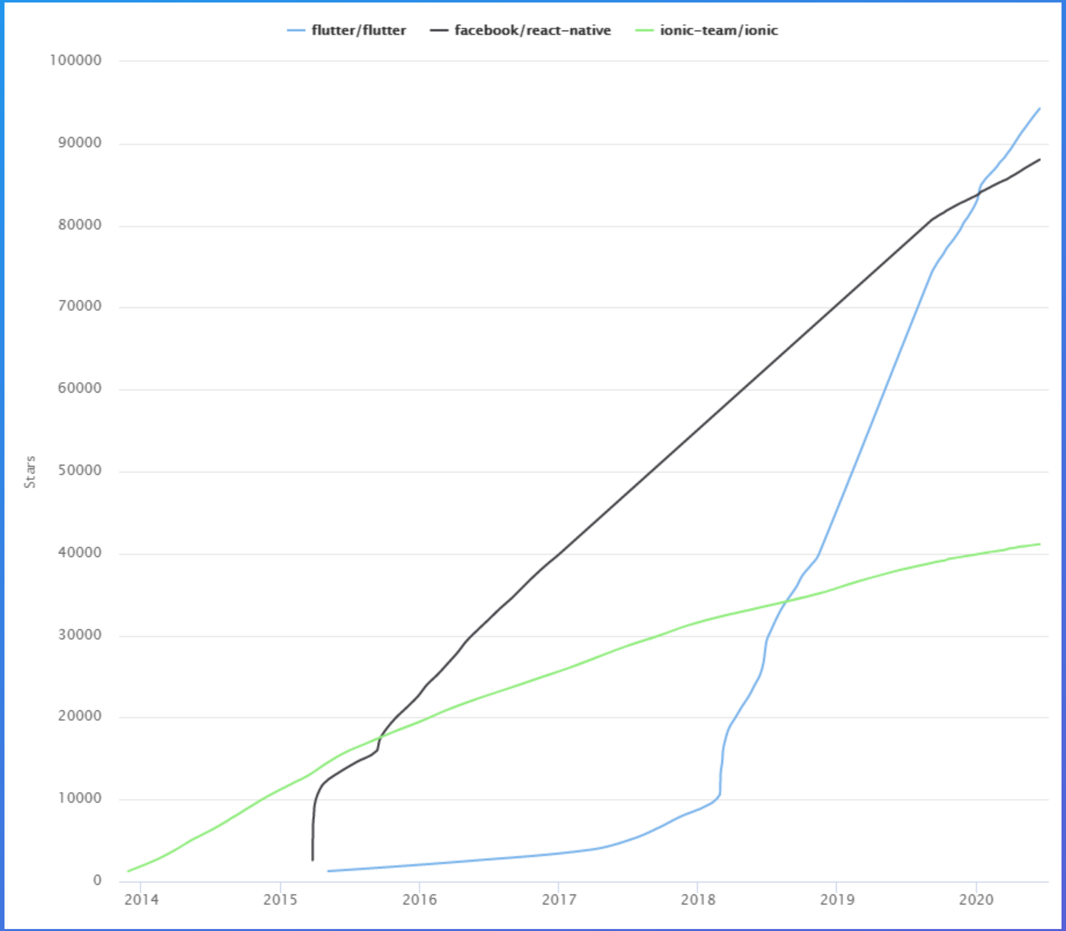
What is Flutter?

- “Flutter is Google’s UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase.”
- Developed and maintained by Google
- Based on the Programming Language Dart
- Open Source
- First Alpha Released May 2017

Flutter

Adaptation

- Used by Big Companies like Google, Ebay, Alibaba and Tencent
- Growing faster than other Frameworks
- Over 10000 flutter packages released on pub.dev
- Multiple Editors like VSCode and Android Studio supported
- Very Active Development



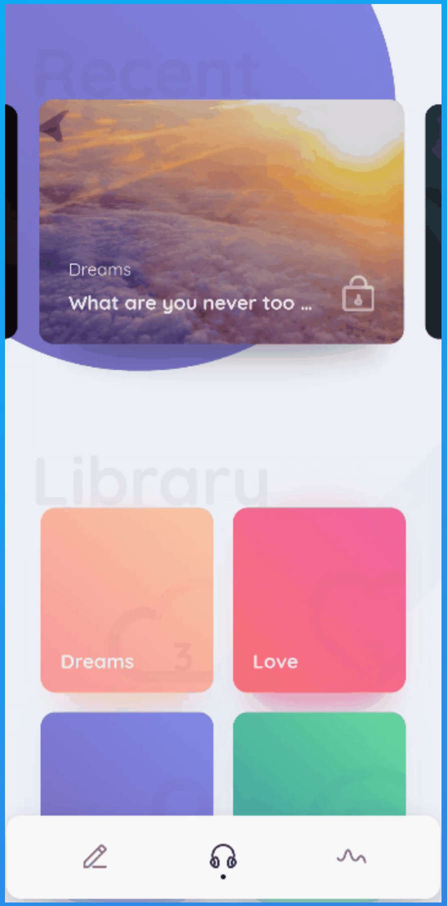
Flutter

What is Flutter?

	Android Studio	Flutter	React Native	PWA
Platforms	Android	**Android, iOS, Web, (Desktop)**	Android, iOS	Everything with a Browser
Programming Languages	Kotlin, Java	Dart	JavaScript	JavaScript
UI Generation	Layout Editor/XML	Declarative Dart Code	HTML/CSS	HTML/CSS
UI Rendering	Native UI Components	Own 2D Renderer	Native UI Components	Browser
Performance	Very Good	**Very Good**	Good	Okay
App Size Overhead	500 KB	**5 MB**	7 MB	~100 KB
Hardware APIs	Excellent	Very Good	Very Good	Good
Extensibility	Okay	**Very Good**	Very Good	Very Good

Flutter

Rendering



- Flutter renders UI with it’s own 2D Rendering Engine written in C++
- 60 FPS even on old devices
- Full control over what is rendered
- Widgets look exactly the same across Platforms
- Not using any native OS UI-Components
- Big Library of Material and Cupertino Widgets (iOS) included

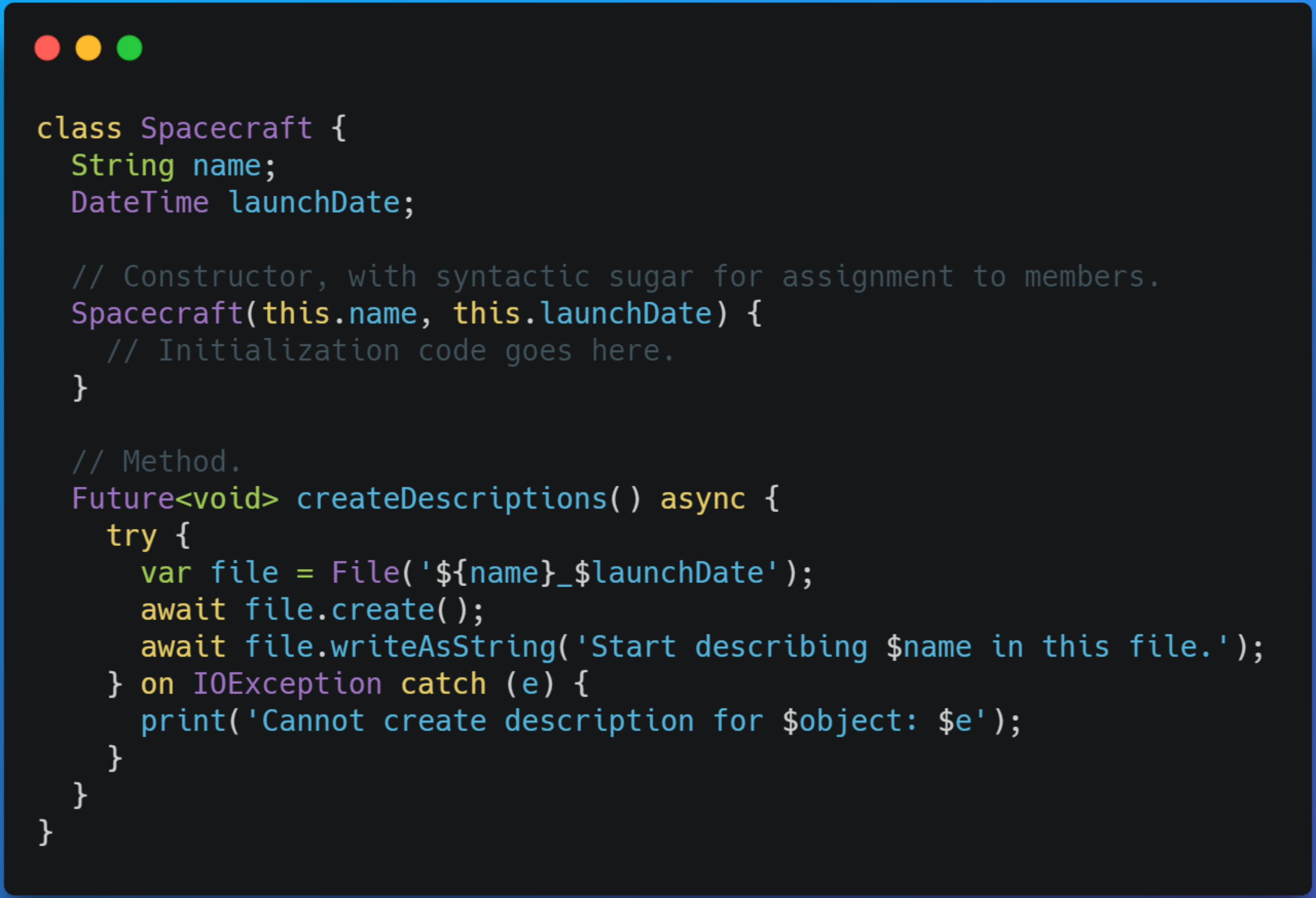
Flutter

Dart

- Object-oriented, garbage-collected language with C-style syntax
- Statically typed
- Inheritance: single inheritance, mixins, interfaces, abstract classes
- Strong support for concurrency and reactive programming (Futures and Streams)
- Compilation to JavaScript for web
- Just-in-time compilation for Hot-Reload during development
- Ahead-of-time compilation to native code when building apps
- Modern package manager. Packages available over pub.dev

Flutter

Dart

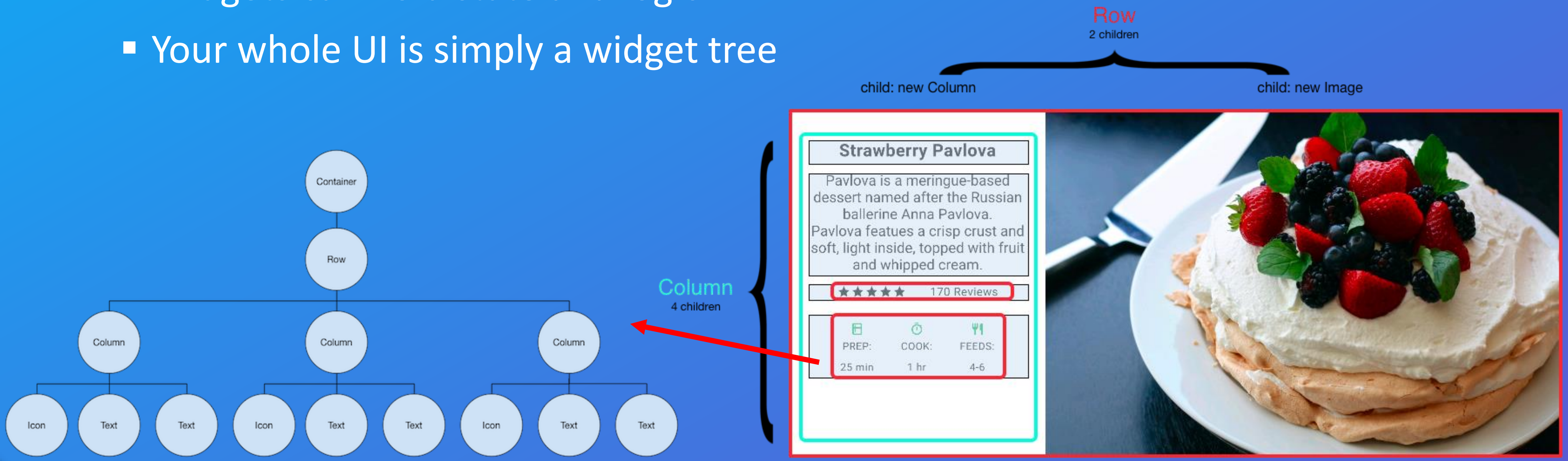


```
class Spacecraft {
  String name;
  DateTime launchDate;

  // Constructor, with syntactic sugar for assignment to members.
  Spacecraft(this.name, this.launchDate) {
    // Initialization code goes here.
  }

  // Method.
  Future<void> createDescriptions() async {
    try {
      var file = File('${name}_$launchDate');
      await file.create();
      await file.writeAsString('Start describing $name in this file.');
    } on IOException catch (e) {
      print('Cannot create description for $object: $e');
    }
  }
}
```

Flutter

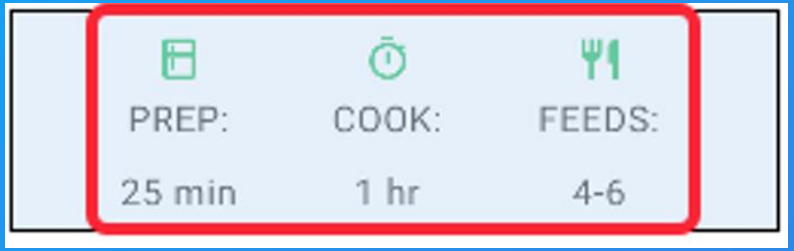
Everything is a Widget!

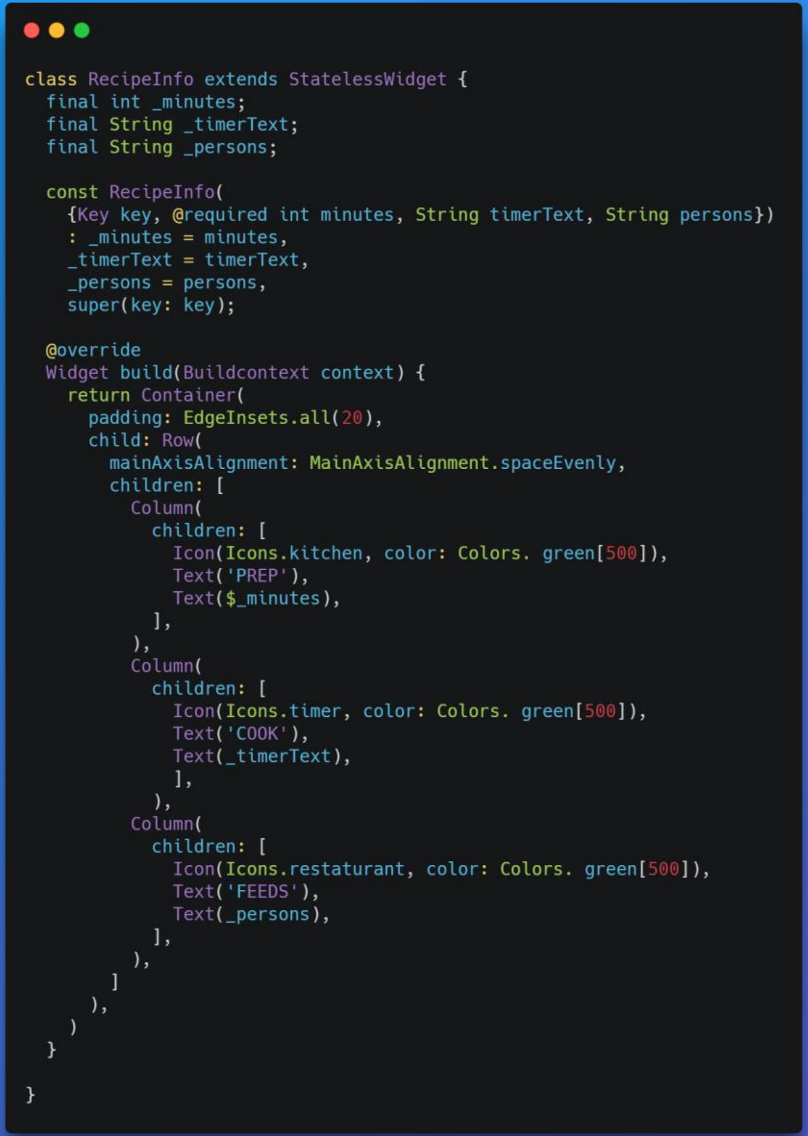
- Everything you draw in Flutter is a Widget, including Appbars and pages. There are no special citizens.
- Widget can control the Layout of it‘s children
- Widgets can hold state and logic.
- Your whole UI is simply a widget tree



Pavlova is a meringue-based dessert named after the Russian ballerine Anna Pavlova. Pavlova featues a crisp crust and soft, light inside, topped with fruit and whipped cream.

You can build your own Widgets

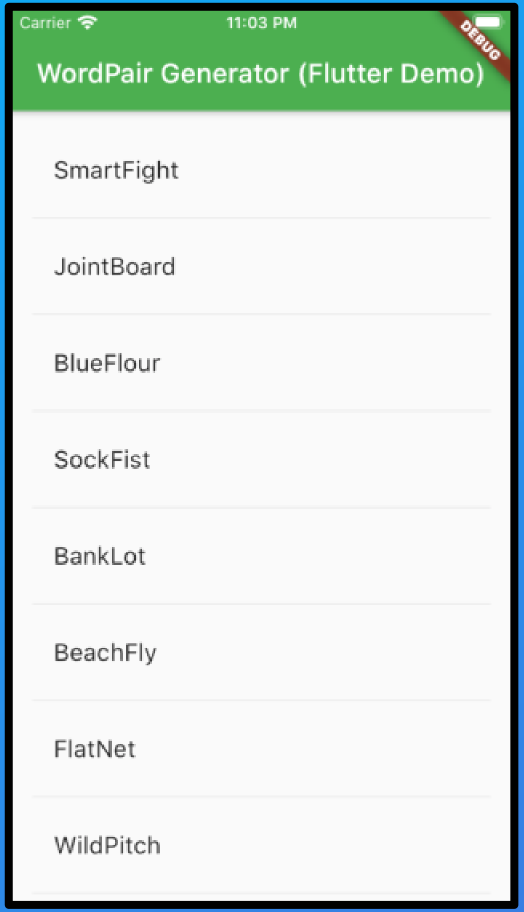




```
class RecipeInfo extends StatelessWidget {
  final int _minutes;
  final String _timerText;
  final String _persons;

  const RecipeInfo(
    {Key key, @required int minutes, String timerText, String persons})
    : _minutes = minutes,
    _timerText = timerText,
    _persons = persons,
    super(key: key);

  @override
  Widget build(Buildcontext context) {
    return Container(
      padding: EdgeInsets.all(20),
      child: Row(
        mainAxisAlignment: MainAxisAlignment.spaceEvenly,
        children: [
          Column(
            children: [
              Icon(Icons.kitchen, color: Colors. green[500]),
              Text('PREP'),
              Text($_minutes),
            ],
          ),
          Column(
            children: [
              Icon(Icons.timer, color: Colors. green[500]),
              Text('COOK'),
              Text(_timerText),
            ],
          ),
          Column(
            children: [
              Icon(Icons.restaurant, color: Colors. green[500]),
              Text('FEEDS'),
              Text(_persons),
            ],
          ),
        ]
      ),
    )
  }

}
```

Flutter

Code Example (WordPair Generator)


Carrier 11:03 PM

WordPair Generator (Flutter Demo)

SmartFight

JointBoard

BlueFlour

SockFist

BankLot

BeachFly

FlatNet

WildPitch

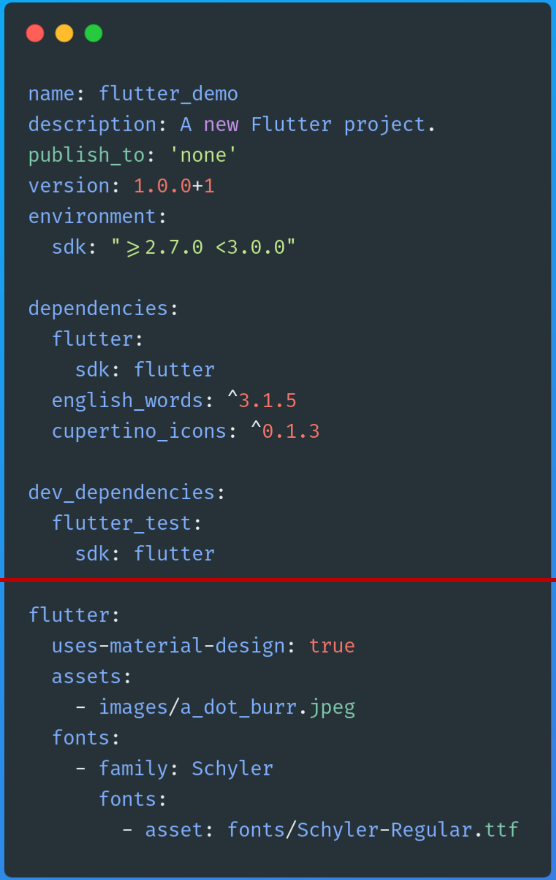
- Very simple WordPair Generator app
- Has an infinite scrolling list

https://github.com/devtobi/flutter-demo

- The project is based on a CodeLab by Google

Flutter

Code Example (Basic pubspec.yaml)

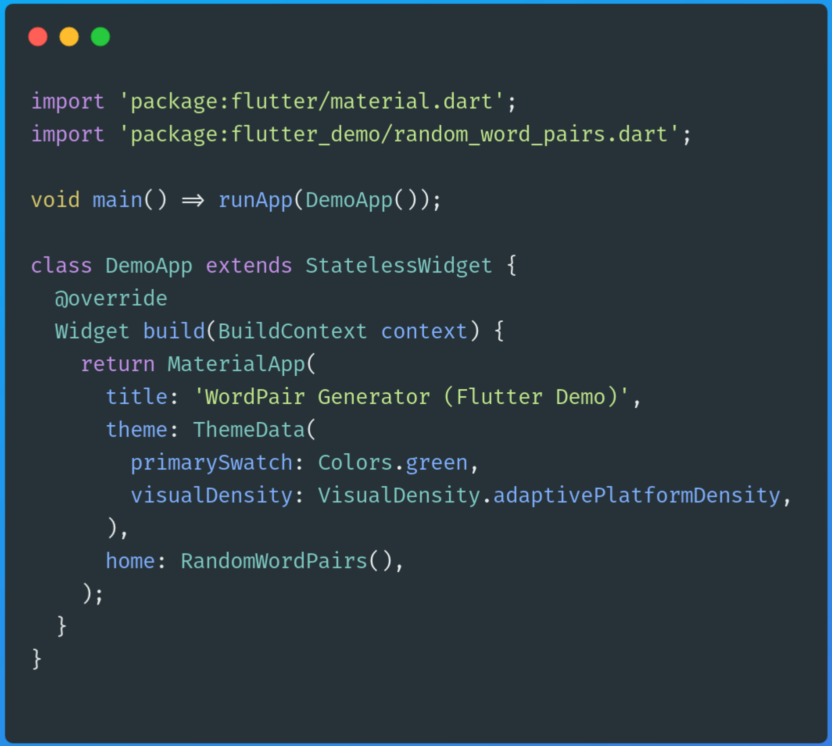


```yaml
name: flutter_demo
description: A new Flutter project.
publish_to: 'none'
version: 1.0.0+1
environment:
  sdk: ">=2.7.0 <3.0.0"

dependencies:
  flutter:
    sdk: flutter
  english_words: ^3.1.5
  cupertino_icons: ^0.1.3

dev_dependencies:
  flutter_test:
    sdk: flutter

flutter:
  uses-material-design: true
  assets:
    - images/a_dot_burr.jpeg
  fonts:
    - family: Schyler
      fonts:
        - asset: fonts/Schyler-Regular.ttf
```

- Manages project **meta data** and **dependencies**
- Used in every Dart project
- **dependencies** are part of the final app
- **dev_dependencies** help in the development process (e.g code generators, linters…)

- Flutter specific configuration section
- Define **assets, fonts** and much more
- **plugins** can use this section for configuration

Code Example (main Widget)

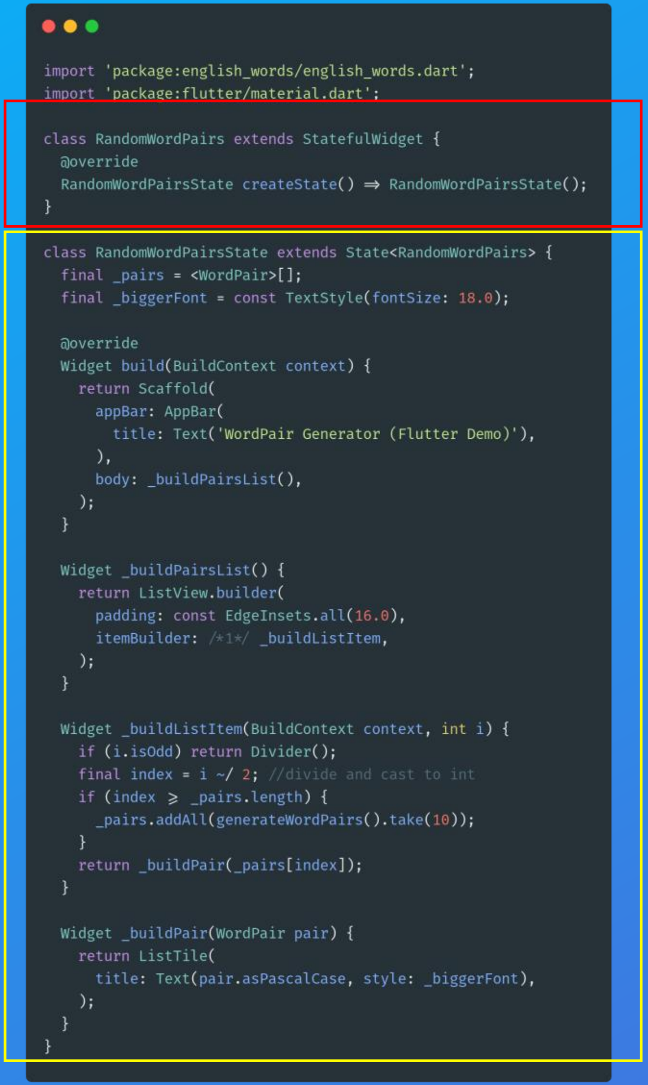


```dart
import 'package:flutter/material.dart';
import 'package:flutter_demo/random_word_pairs.dart';

void main() ⇒ runApp(DemoApp());

class DemoApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'WordPair Generator (Flutter Demo)',
      theme: ThemeData(
        primarySwatch: Colors.green,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: RandomWordPairs(),
    );
  }
}
```

- main()-method as **entry point** for the **dart program**
- **runApp()** is a Flutter specific function that accepts a widget
- Flutter is **widget based** and uses a **widget hierarchy**
- **Root widget** of every app is **MaterialApp/CupertinoApp**
- Widgets can be **stateless** or **stateful**
- Widgets itself consist of widgets

Flutter

Code Example (WordPairs Widget)



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

class RandomWordPairs extends StatefulWidget {
  @override
  RandomWordPairsState createState() ⇒ RandomWordPairsState();
}

class RandomWordPairsState extends State<RandomWordPairs> {
  final _pairs = <WordPair>[];
  final _biggerFont = const TextStyle(fontSize: 18.0);

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('WordPair Generator (Flutter Demo)'),
      ),
      body: _buildPairsList(),
    );
  }

  Widget _buildPairsList() {
    return ListView.builder(
      padding: const EdgeInsets.all(16.0),
      itemBuilder: /*1*/ _buildListItem,
    );
  }

  Widget _buildListItem(BuildContext context, int i) {
    if (i.isOdd) return Divider();
    final index = i ~/ 2; //divide and cast to int
    if (index ⩾ _pairs.length) {
      _pairs.addAll(generateWordPairs().take(10));
    }
    return _buildPair(_pairs[index]);
  }

  Widget _buildPair(WordPair pair) {
    return ListTile(
      title: Text(pair.asPascalCase, style: _biggerFont),
    );
  }
}
```

- Stateful widgets are split in two classes – the **widget itself** and the corresponding **state class** (which extends the generic state class)
- The widget has a **createState(**)– method
- State contains **variables** that can **change over the widgets lifetime** (if the UI should respond to **the state change** the variable change must be **wrapped in setState()**)
- State is automatically preserved throughout app lifecycle
- **Split the build function** into more methods to make it **more readable**

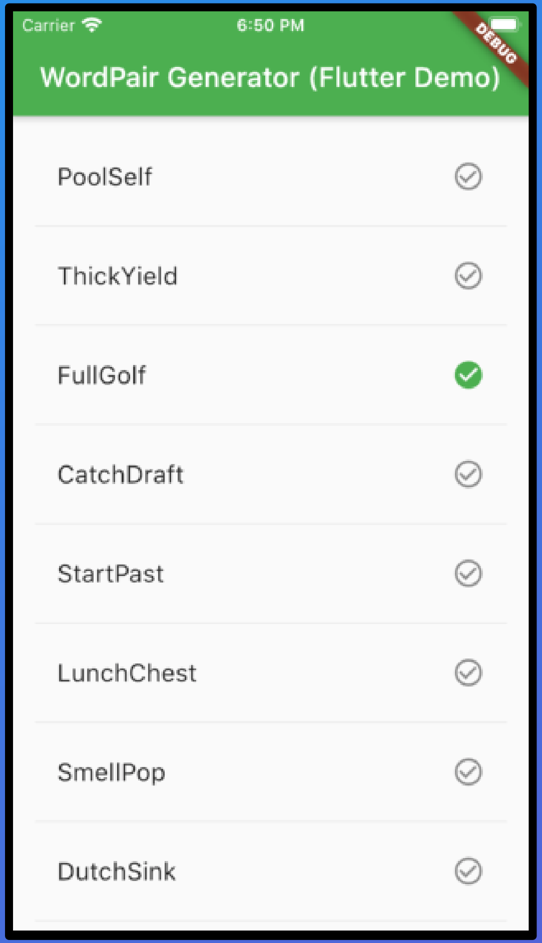
Flutter

Live Coding

What we gonna add:

- **Checkmark icon** on the right side of each item
- **Change state** of the WordPairs widget on **item click**

  -> In this particular case the **icon changes** its **appearance**


Carrier 6:50 PM

WordPair Generator (Flutter Demo)

PoolSelf ✓

ThickYield ✓

FullGolf ✓

CatchDraft ✓

StartPast ✓

LunchChest ✓

SmellPop ✓

DutchSink ✓

Flutter

Summary

- Flutter is constantly growing in popularity since its release
- Flutter is a cross-platform framework that can compile to iOS, Android, Desktop or Web
- Flutter is highly extendable due to the vast amount of existing Dart packages
- Dart has developer friendly features like hot reloading
- Everything in Flutter is a widget
- Downside is the overhead in application size
- Native development only makes sense when highest performance is needed or extremely special sensor data has to be retrieved

Sources

- https://dart.dev
- https://flutter.dev
- https://codelabs.developers.google.com/codelabs/first-flutter-app-pt1/index.html

Flutter