

# | Welcome to the dart side



**Mais Alheraki** | ميس الحراكي  
Flutter GDE & software engineer

## *10 months of Flutter!*

**Week #1** Introduction

**Week #2** Welcome to The Dart Side

**Week #3** Basic Widgets & Material Design

**Week #4** Make Your First Flutter App

**Week #5** Using Packages to Extend Functionality

**Week #6** Flutter & Backend

**Week #7** State Management

**Week #8** Creating a Chat App

**Week #9** Navigation

**Week #10** Testing Your Widgets



## *For absolute beginners*

1. We will learn basic things any Flutter developer should know
2. This Course is just the first step
3. Along the way you will have a lot of resources to support your learning journey



## *Where can I find all the resources?*

Don't worry, links can get lost, I know!

For that I have this one link for you, where I will put all the resources including the slides.

**[fairybits.com/growwgoogle-flutter](https://fairybits.com/growwgoogle-flutter)**

Bookmark it!



## *Further support*

Tag me on Twitter @pr\_Mais for any issue you face and I will help.

**1 rule of thumb:** public tweets are better as it gives the opportunity for more people to learn from your question!



*Let's Flutter*

# | What is Dart?





Dart is a client-optimized language for fast apps  
on any platform

[dart.dev](https://dart.dev)



1

## Optimized for UI

Built for the needs of UI creation





# 2

## Hot reload

See changes instantly, no need to recompile



3

## **Fast on all platforms**

Compiles directly to ARM and x64 machine code. Compiles to JavaScript on web.



**Dart** is the power that stands  
behind **Flutter**

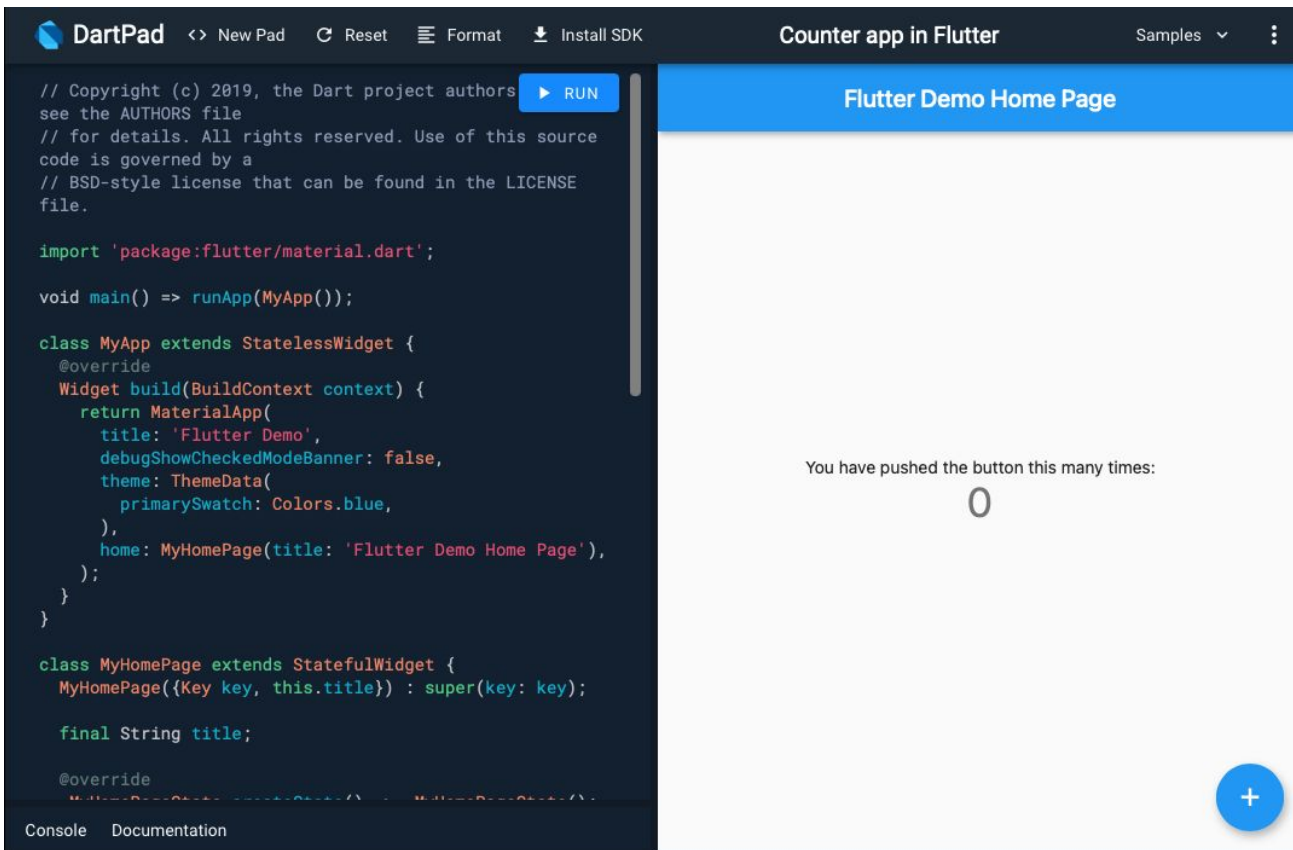


*Let's Flutter*

# | Where to start



## 1. Try it on DartPad.dev



The screenshot shows the DartPad web interface. The top bar includes the DartPad logo, navigation links (New Pad, Reset, Format, Install SDK), the title "Counter app in Flutter", and a "Samples" dropdown menu. The left pane displays Dart code for a counter app, with a "RUN" button. The right pane shows the app's output, which includes a blue header "Flutter Demo Home Page" and a counter display showing "0". A blue circular button with a "+" sign is visible in the bottom right corner of the right pane.

```
// Copyright (c) 2019, the Dart project authors.
// see the AUTHORS file
// for details. All rights reserved. Use of this source
// code is governed by a
// BSD-style license that can be found in the LICENSE
// file.

import 'package:flutter/material.dart';

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

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      debugShowCheckedModeBanner: false,
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    );
  }
}

class MyHomePage extends StatefulWidget {
  MyHomePage({Key key, this.title}) : super(key: key);

  final String title;

  @override
  State<MyHomePage> createState() => MyHomePageState();
}
```

Flutter Demo Home Page

You have pushed the button this many times:

0

Console Documentation



## 2. Install

*If you will use Flutter, Dart comes with it, so once you install Flutter you actually have installed Dart as well.*



*Let's Flutter*

# | Let's Learn Dart



1. Dart built-in types and generics
2. Classes and inheritance
3. Async
4. Isolates
5. Null safety





## Built-in Types

Everything in Dart is an Object

Literals: used to initialize any of the built-in types.

```
final x = 13; //integer
final y = 1.5; //double
final sentence = "This is a String"; //string
final condition = true;
```

- ❏ numbers
- ❏ strings
- ❏ booleans
- ❏ lists (also known as *arrays*)
- ❏ sets
- ❏ maps
- ❏ runes (for expressing Unicode characters in a string)
- ❏ symbols



## Built-in Types

Initialize a type using a constructor



*//Lists*

```
final names = ['Ahmad', 'Ayman', 'Sara']; //List
final names = []; //Empty list
final names = List(); //Depecated in null-safety
final names = List.filled(1, 0, growable: true); //Will be explained in null-safety section
```

*//Maps*


```
final users = {};
final users = Map();
final users = {'name': User()};
final users = <String, User>{};
```



## Generics

Generics makes your code safer

It's often used with collections and functions



```
var names = List<String>();  
names.addAll(['Seth', 'Kathy', 'Lars']);  
names.add(42); // Error
```



## Classes

### Class Sample

1. How to define a class
2. Class members (properties, methods, constructors)
3. Enums
4. Extension methods



*Let's Flutter*

**| Go Further**



Your FIRST destination to learn Dart:

[dart.dev/guides](https://dart.dev/guides)

Null Safety:

[dart.dev/null-safety](https://dart.dev/null-safety)



*Let's Flutter*

# Thank you!



*Upcoming: Basic Widgets and Material Design*



[fairybits.com](https://fairybits.com)



[@pr\\_Mais](https://twitter.com/pr_Mais)



[/in/maisalheraki](https://in.linkedin.com/in/maisalheraki)