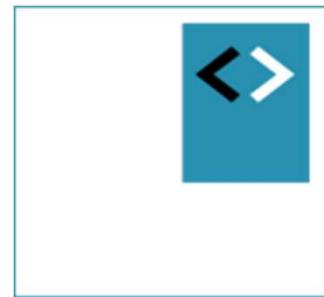




# **aariXa Academie:**

# **Flutter – introduXion**



Peter Kassenaar

[info@kassenaar.com](mailto:info@kassenaar.com)



# Peter Kassenaar

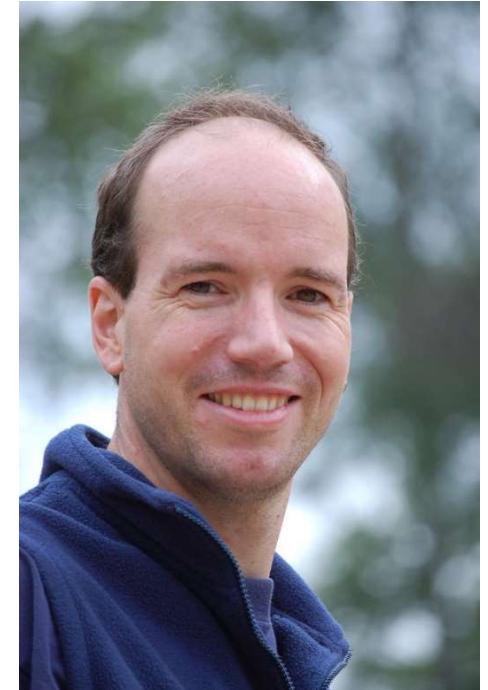


- Trainer, auteur, developer – since 1996
- Specialty: *JavaScript and cross-platform apps*
- React, Vue.js, Angular, TypeScript, JavaScript, Cordova, Flutter

[www.kassenaar.com](http://www.kassenaar.com)

[info@kassenaar.com](mailto:info@kassenaar.com)

Twitter: [@PeterKassenaar](https://twitter.com/PeterKassenaar)



oberon interactive



## aariXa academie



**Flutter** 25/03/2021

Flutter is een Mobile UI Framework van Google, dat wordt gebruikt voor het maken van apps voor iOS en Android. Je maakt met Flutter native apps. Het zijn dus geen apps die draaien in een webview van het betreffende platform (zoals Ionic of PhoneGap/Cordova).

Het kenmerk van Flutter is dat er een single code base is. Apps schrijf je in de eveneens door Google ontwikkelde programmeertaal Dart. Je schrijft hiermee zowel de lay-out als de programmalogica. Jouw Dart-code wordt



# Agenda

## 1. "Theory" *19:00*

- What is Flutter, Landscape, Features, pro- and cons
- Tooling: Android Studio, Emulator, Dart
- Architecture of Flutter-applications
- *Widgets* and Flutter Layout system

\* \* \* \* ca. 20:15 Break \* \* \*

## 2. "Hands-on" – until ca. *21:30*

- *Live coding – demo application*
- Working with external data and API's
- Building complete applications

## 3. Q & A

## Q & A



*Do you have any **questions**? Ask them  
via the **chat box**!*

After the presentation we'll look into the  
topics/questions

[github.com/PeterKassenaar/aarixa](https://github.com/PeterKassenaar/aarixa)

The screenshot shows a GitHub repository page for the user 'PeterKassenaar' with the repository name 'aarixa'. The page includes a navigation bar with links for 'Pull requests', 'Issues', 'Marketplace', and 'Explore'. Below the navigation bar, there are buttons for 'Unwatch', 'Star', and 'Fork'. The main content area displays repository statistics: 1 commit, 1 branch, 0 packages, 0 releases, 1 contributor, and an MIT license. A list of files is shown, including '.gitignore', 'LICENSE', 'README.md', and another 'README.md' file containing the text 'aarixa'. The repository description is 'Slides and sample application on the Aarixa Academy React - March 2020'.

OctoTree >

PeterKassenaar / aarixa

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

Slides and sample application on the Aarixa Academy React - March 2020

Manage topics

1 commit 1 branch 0 packages 0 releases 1 contributor MIT

Branch: master New pull request Create new file Upload files Find file Clone or download

PeterKassenaar Initial commit Latest commit 49c3358 10 days ago

.gitignore Initial commit 10 days ago

LICENSE Initial commit 10 days ago

README.md Initial commit 10 days ago

README.md

aarixa

Slides and sample application on the Aarixa Academy React - March 2020



# What is Flutter?

What can you create using Flutter?

# What is Flutter?



A screenshot of the official Flutter website. The header features the Flutter logo, navigation links for Docs, Showcase, and Community, and social media icons for GitHub, Twitter, YouTube, and Gitter. A prominent blue banner at the top encourages users to "Sign up now for Flutter Engage, happening worldwide on 3/3!" and provides a link to open job listings. Below the banner, a large central image shows a smartphone displaying a vibrant landscape scene from a game. Overlaid on the phone screen is a code snippet showing asset paths: "assets: - images/abaaba.jpeg - images/fern.jpeg". To the left of the phone, there's a wavy line icon and a blue box containing the text "Create faster apps". To the right, there's a circular progress bar with the text "60 fps" and a button labeled "showDialog()". The bottom of the page has a light blue footer bar.

<https://flutter.dev/>



*"Flutter is Google's UI toolkit for building beautiful, natively compiled applications for mobile, web, and desktop from a single codebase."*





# Flutter 2 – March 2021



Our goal is to *fundamentally shift* how developers think about building apps, starting not with the platform you're targeting but rather with the experience you want to create.

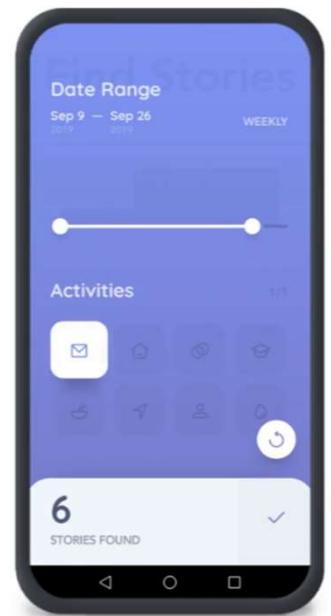
Flutter allows you to handcraft **beautiful** experiences where your brand and design comes to the forefront. Flutter is **fast**, compiling your source to machine code, but

<https://developers.googleblog.com/2021/03/announcing-flutter-2.html>

# Flutter features



- Fast Development
  - Integrated Development Environment, Emulators
  - Hot reload
  - Fully customizable UI-widgets
- Expressive and Flexible UI
  - Create UI from code, flexible designs
- Native performance
  - Dart as a programming language
  - High performance on rendering, scrolling, navigation, animation



# Flutter Pro's



- Fast development
- Multi-platform
- Single code base
- Cross Platform
- Rich UI system
- Native performance
- Developed and Backed by Google



# Flutter Cons



- No *platform-aware* widgets
  - Build UI twice if you want to specifically target both mobile platforms optimally
- Learn a new language (Dart)
- Relatively new – smaller community
- Not the smallest footprint for apps
- Rapid changes between versions
- Developed and backed by Google



# More information



We use cookies to enhance your experience. Read more about cookies in our [privacy policy](#). [Agree](#)

 Future Mind      Portfolio    Services    Solutions    Estimate project    Blog    Career    [Contact](#)

June 2, 2020    6 min. read    #Mobile Development

## Pros & Cons of Flutter Mobile Development

 Wojciech Rozwadowski





<https://www.futuremind.com/blog/pros-cons-flutter-mobile-development>



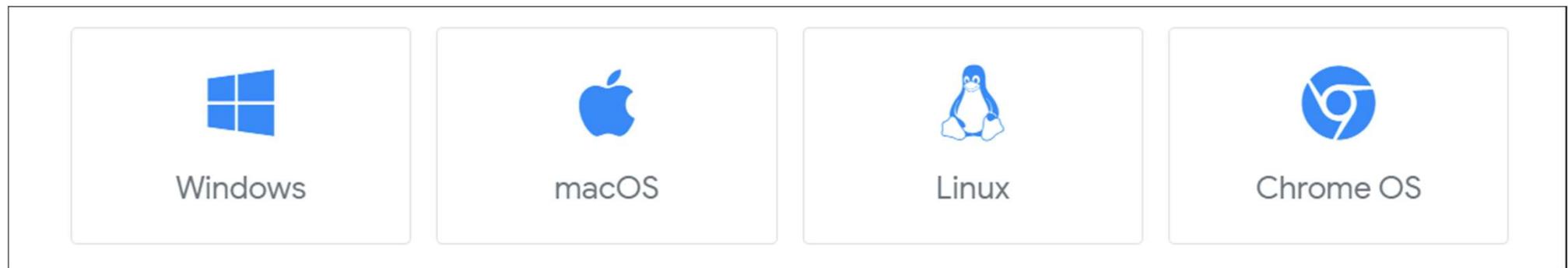
# Flutter requirements

Prerequisites, hardware and software

# Flutter Requirements - Hardware



- PC or Mac
  - Optional: physical phone/tablet to test application
- iOS applications?
  - You'll need a Mac + XCode



# Flutter Requirements - software



- Editor
  - Android Studio, Visual Studio Code + Flutter plug-ins
  - OR: IntelliJ Community/Ultimate
- Flutter SDK



## Get the Flutter SDK

1. Download the following installation bundle to get the latest stable release of the Flutter SDK:

[flutter\\_windows\\_1.22.6-stable.zip](#)

For other release channels, and older builds, see the [SDK releases](#) page.

# Android Emulator – AVD Manager



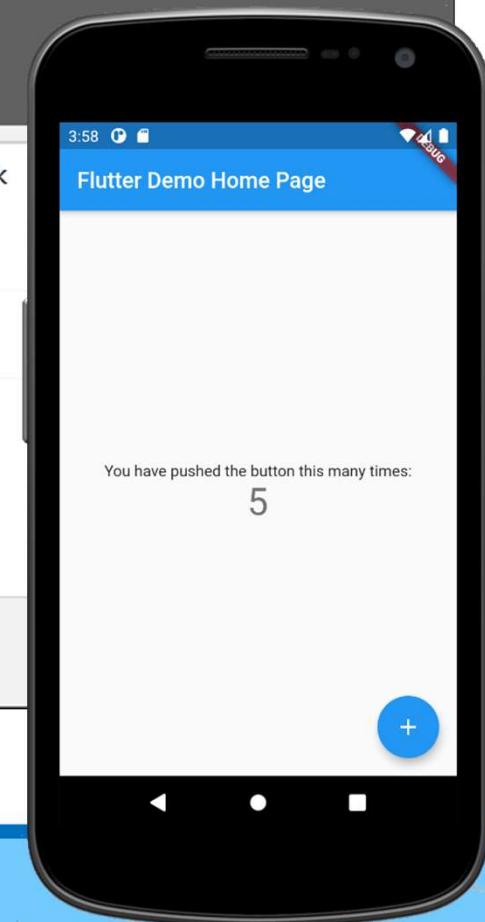
Android Virtual Device Manager

## Your Virtual Devices

Android Studio

Type	Name	Play Store	Resolution	API	Target	CPU/ABI	Size on Disk
Smartphone	Galaxy Note 4		720 × 1280	30	Android 5.0	x86	12 GB
Smartphone	Nexus S A		480 × 800	30	Android 5.0	x86	10 GB

+ Create Virtual Device...



A screenshot of a Flutter application running on an Android emulator. The app displays a counter with the value '5'. The top status bar shows the time as 3:58 and battery level. The bottom navigation bar has three icons. A blue circular button with a '+' sign is visible at the bottom right of the screen.

Flutter Demo Home Page

You have pushed the button this many times:

5



# Installing Flutter

Prerequisites, Installation of hardware and software

# Installation Guides per platform



Flutter

Docs Showcase Community Get started

Sign up now for Flutter Engage, happening worldwide on 3/3!

Interested in working on Flutter? See our [open job listings](#).

Get started

- 1. Install
- 2. Set up an editor
- 3. Test drive
- 4. Write your first app
- 5. Learn more

From another platform?

- Flutter for Android devs
- Flutter for iOS devs
- Flutter for React Native devs
- Flutter for web devs
- Flutter for Xamarin.Forms devs
- Introduction to declarative UI
- Dart language overview
- Building a web app

## Install

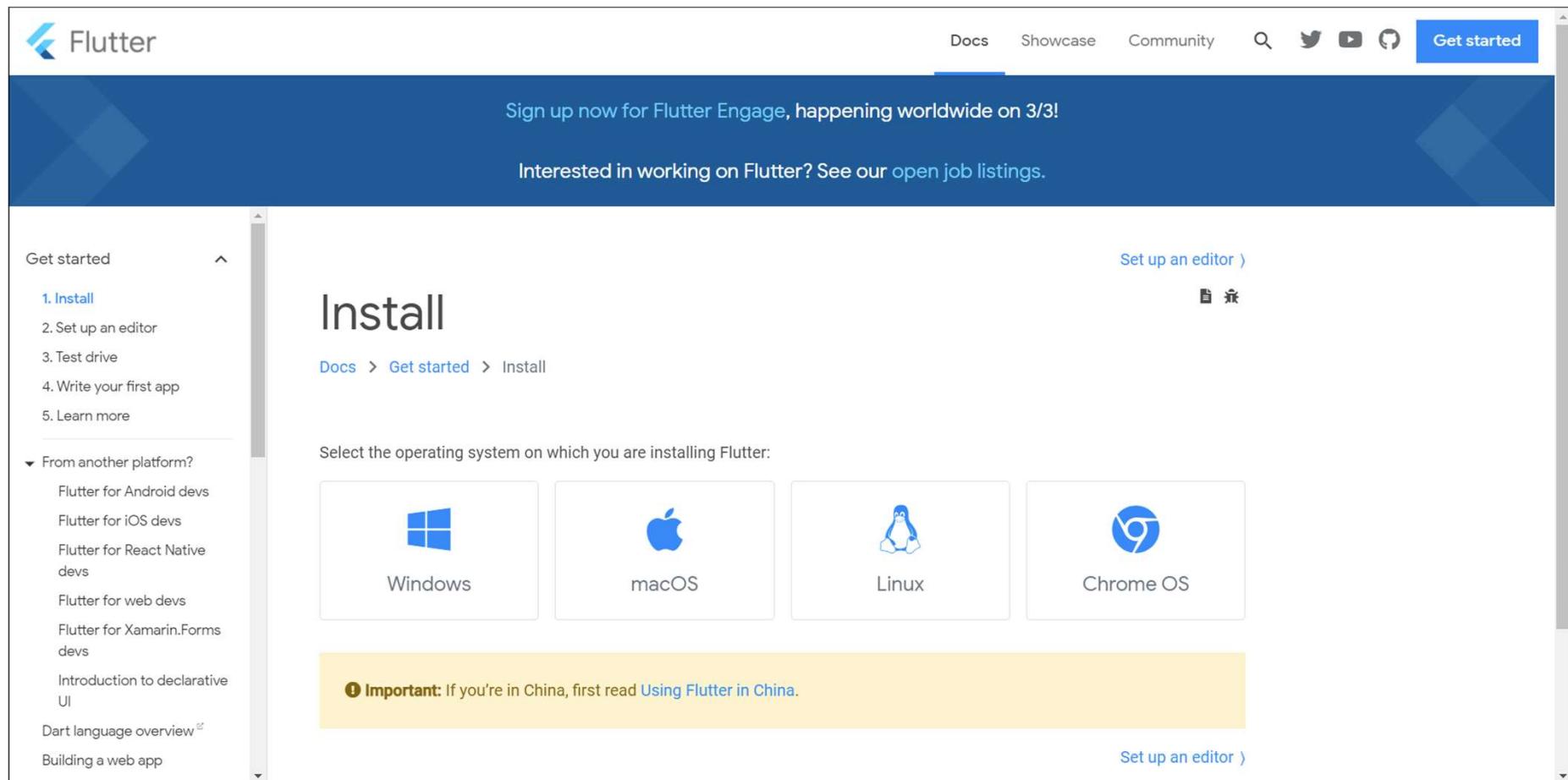
Docs > Get started > Install

Select the operating system on which you are installing Flutter:

Windows   macOS   Linux   Chrome OS

**Important:** If you're in China, first read [Using Flutter in China](#).

[Set up an editor](#)



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

# Installation



See official Docs, or blog post

Otherwise:

<https://www.kassenaar.com/blog/post/2020/10/29/Kennismaken-met-Flutter.aspx>

<https://www.youtube.com/watch?v=75iIR9OVAZ8>

A screenshot of a blog post on Peter Kassenaar's website. The post is titled "Kennismaken met Flutter" and is dated October 29. It discusses the author's experience with Flutter during the coronavirus lockdown. The post includes a summary, a bulleted list of frameworks, and a conclusion. To the right of the post is a sidebar featuring a photo of the author, his biography, and social media links. The top navigation bar includes links for the general site, blog homepage, archive, and newsletter sign-up.

# Official documentation



Flutter

Docs Showcase Community Get started

Migrate your packages to null safety!

Get started

- 1. [Install](#)
- 2. Set up an editor
- 3. Test drive
- 4. Write your first app
- 5. Learn more

From another platform?

- Flutter for Android devs
- Flutter for iOS devs
- Flutter for React Native devs
- Flutter for web devs
- Flutter for Xamarin.Forms devs
- Introduction to declarative UI
- Dart language overview ↗
- Building a web app

Samples & tutorials

Development

- User interface
- Data & backend
- Accessibility & internationalization
- Platform integration
- Packages & plugins
- Add Flutter to existing app

Install

Docs > Get started > Install

Select the operating system on which you are installing Flutter:

- Windows
- macOS
- Linux
- Chrome OS

**Important:** If you're in China, first read Using Flutter in China.

Set up an editor

flutter-dev@ • terms • brand usage • security • privacy • español • 社区中文资源 • 한국어 • We stand in solidarity with the Black community. Black Lives Matter.

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



# Your First App

Creating and running your first Flutter application

# Quickly start a new app



Create New Flutter Project X

### New Flutter Project

The dialog shows four project templates:

- Flutter Application**: Selected. It features a green Android icon with a white 'F' and a teal header bar.
- Flutter Plugin**: It features a green Android icon with a white puzzle piece and a teal header bar.
- Flutter Package**: It features a green Android icon with a white Dart icon and a teal header bar.
- Flutter Module**: It features a green Android icon with a white square grid icon and a teal header bar.

Select an "Application" when building for end users.  
Select a "Plugin" when exposing an Android or iOS API for developers.  
Select a "Package" when creating a pure Dart component, like a new Widget.  
Select a "Module" when creating a Flutter component to add to an Android or iOS app.

Previous Next Cancel Finish

# Requirements for new apps



- Always a good idea to get started with the **default app**, then **delete everything** you don't need
- We need:
  - Project **name** – the name of your app
    - Always `lowercase_with_underscores` (no capitals, no dashes, no numbers)
  - **SDK Path** – the location of your SDK (like `c:\src\flutter`)
  - Project **location** – personal preference (`\Documents`, or `\Desktop`)
  - **Description** – a useful description of your app

# Your first Flutter App



Create New Flutter Project

## New Flutter Application

Configure the new Flutter application

**Project name**  
first\_flutter\_app

**Flutter SDK path**  
c:\src\flutter

**Project location**  
C:\Users\Gebruiker\Desktop

**Description**  
Peter's first sample Flutter application

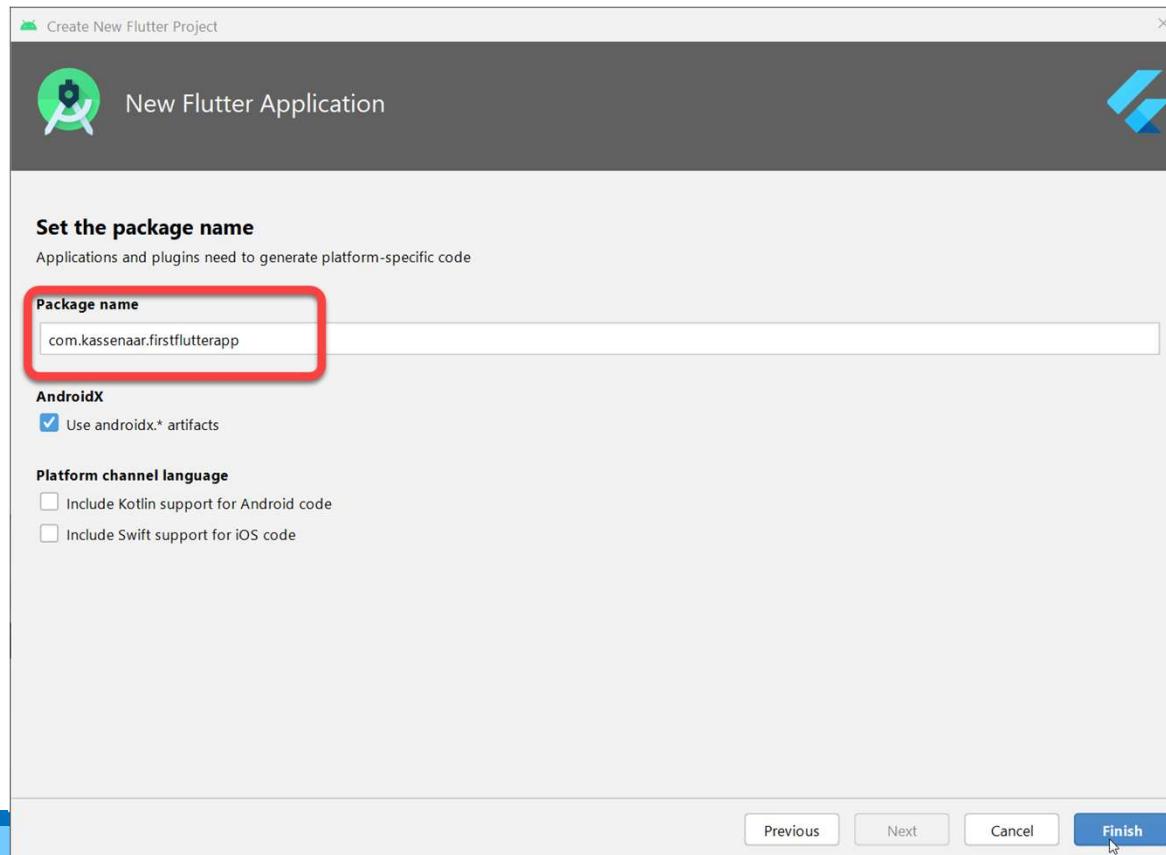
Create project offline

Previous **Next** Cancel Finish

# Kotlin/Swift support not necessary



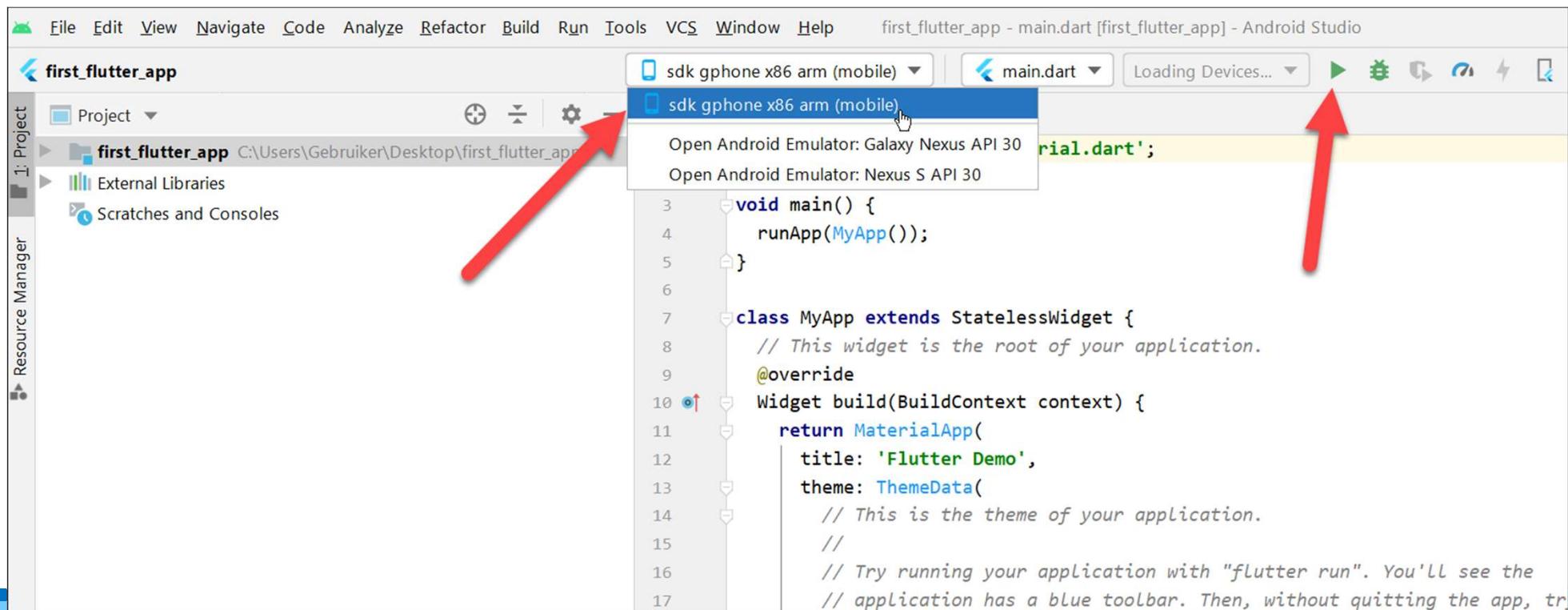
- Create a unique package name
- Typically: a domain name in reverse order



# Running the sample app



- Select an installed emulator or physical device
- Click the Run button (Shift + F10)



# Play around



- First time compilation will take a while
- Test the counter in the sample application

The image shows a screenshot of an Android Studio interface. On the left, the 'Run' tab is open, showing the path 'main.dart'. Below it, the log output for a debug build is displayed:

```
Launching lib\main.dart on sdk gphone x86 arm in debug mode...
Running Gradle task 'assembleDebug'...
Built build\app\outputs\flutter-apk\app-debug.apk.
Installing build\app\outputs\flutter-apk\app.apk...
Waiting for sdk gphone x86 arm to report its views...
Debug service listening on ws://127.0.0.1:50138/smG3y7anoyE=/ws
Syncing files to device sdk gphone x86 arm...
```

On the right, a virtual Android device displays the 'Flutter Demo Home Page'. The screen shows the text 'Flutter Demo Home Page' at the top, followed by 'You have pushed the button this many times:' and the number '5'. At the bottom right is a blue circular button with a white plus sign '+', used for incrementing the counter.

# Folder structure



- You – generally – **don't need** to manually adjust  
\.idea, \android, \build and \ios
- You **may remove** the contents of the \test folder  
(when not using unit tests)
- You'll work inside the \lib folder
  - main.dart is the startup file
- Other (top level) files are mainly for **configuration**
  - You'll hardly ever touch them

# General structure



The screenshot shows the Android Studio interface with a Flutter project named "first\_flutter\_app". A red box highlights the Project and Resource Manager panes on the left, which contain the project's directory structure and files. The main pane displays the content of the "main.dart" file, which defines the application's entry point and root widget.

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

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

class MyApp extends StatelessWidget {
  // This widget is the root of the application.
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(...),
      home: MyHomePage(title: 'F'),
    ); // MaterialApp
  }
}

class MyHomePage extends StatefulWidget {
  class _MyHomePageState extends S
}
```

# Default code – with the comments removed



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

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

class MyApp extends StatelessWidget {
  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(
        primarySwatch: Colors.blue,
        visualDensity: VisualDensity.adaptivePlatformDensity,
      ),
      home: MyHomePage(title: 'Flutter Demo Home Page'),
    );
}
...
...
```

Study the default code. It has useful comments.

# What can we learn from the default code?



*Architecture of all Flutter apps*

# Structure



- On top: import the packages you need
- void main() is the startup function.
- It calls runApp() to run the app
- runApp calls the MyApp() class to build the UI
- MyApp itself is a **Widget**. It returns a **Widget Tree**
  - *Widget Trees* describe the **behavior** and **layout** of your app
- Widgets are always **classes**
  - **Custom** classes
  - **Default** classes / **Widgets**

# Classes



- StatelessWidget
  - Have no data (or 'state')
  - Always return a `build()` function...
  - ...which should be of type `Widget`
  - `@override` is mostly **optional** in newer versions of Dart
- So: `Widget build(BuildContext context) {...}`



- StatelessWidget
  - Can have data or state
  - Actually consists of two classes
  - We'll look into that later!

```
17
18     }
19     // Our own class, extending the base StatelessWidget class
20     class HomeAsync extends StatelessWidget {
21         @override
22         @+  _HomeAsyncState createState() => _HomeAsyncState();
23     }
24
25     class _HomeAsyncState extends State<HomeAsync> {...}
150
```

Two red arrows point to specific parts of the code in the screenshot. One arrow points to the line 'class HomeAsync extends StatelessWidget {', and another arrow points to the line 'class \_HomeAsyncState extends State<HomeAsync> {...}'.

# So, when collapsed, the structure is like



```
main.dart ×  
1 import 'package:flutter/material.dart';  
2  
3 void main() {  
4   runApp(MyApp());  
5 }  
6  
7 class MyApp extends StatelessWidget {  
8   // This widget is the root of your application.  
9   Widget build(BuildContext context) {  
10     return MaterialApp(  
11       title: 'Flutter Demo',  
12       theme: ThemeData(...), // ThemeData  
13       home: MyHomePage(title: 'Flutter Demo Home Page'),  
14     ); // MaterialApp  
15   }  
16 }  
17  
18 }  
19 }  
20  
21 class MyHomePage extends StatefulWidget {...}  
22  
23 class _MyHomePageState extends State<MyHomePage> {...}  
24  
25 |
```

A red arrow points from line 9 down to the start of the `MyHomePage` class definition at line 21. The code block is enclosed in a red rectangle.



# **Starting from scratch**

Building the UI from scratch – based on the Default  
Starter App

# Working with Widget Trees



- Starting from the Default App and **deleting** (most of) the code
- Open `main.dart`
- Remove all classes and empty `runApp()`
- Your app will **not work** at the moment:

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

void main() {
  runApp();
}
```

# Using MaterialApp()



- MaterialApp() is often the **start widget** of your Widget Tree
- It adheres to the Google Material Design Specs
  - <https://material.io/design>
- You configure it with *properties*, for instance home:
- But: you **can not** simply type some text to be shown on the home page

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

void main() {
  runApp(MaterialApp(
    home: 'Hello Flutter!'
  ));
}
```

Invalid!

# Using Widgets

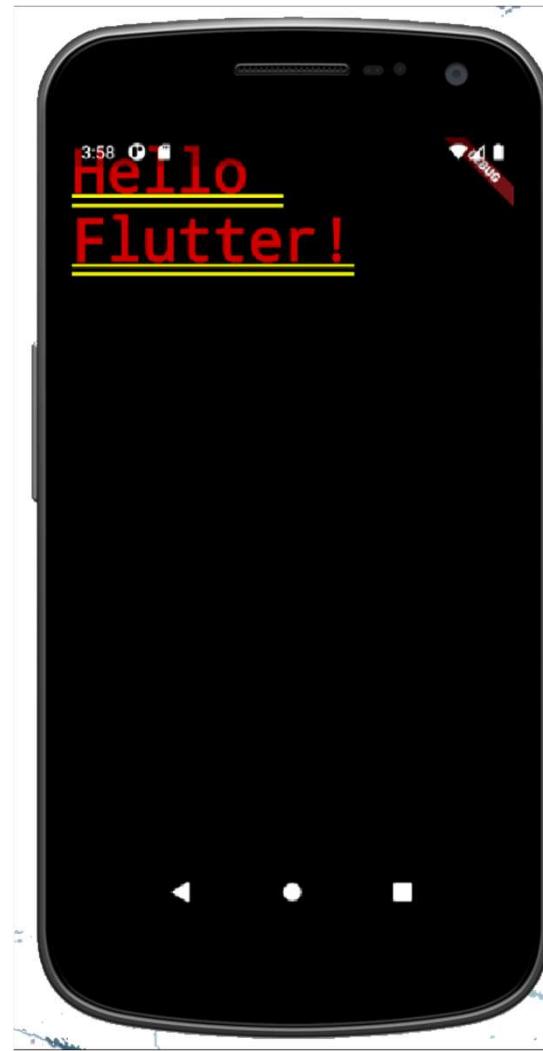


- *Rule* – ALL content **MUST** be wrapped **inside a Widget**
- There are **thousands** of predefined widgets
- You can build/compose **custom widgets** yourself
- For instance: wrap text inside a `Text()` Widget
- But – **EVERY piece of content** is some form of widget or property

```
void main() {  
  runApp(MaterialApp(  
    home: Text('Hello Flutter!'),  
  ));  
}
```

Valid!

# Result – ugly but working





# Layout widgets

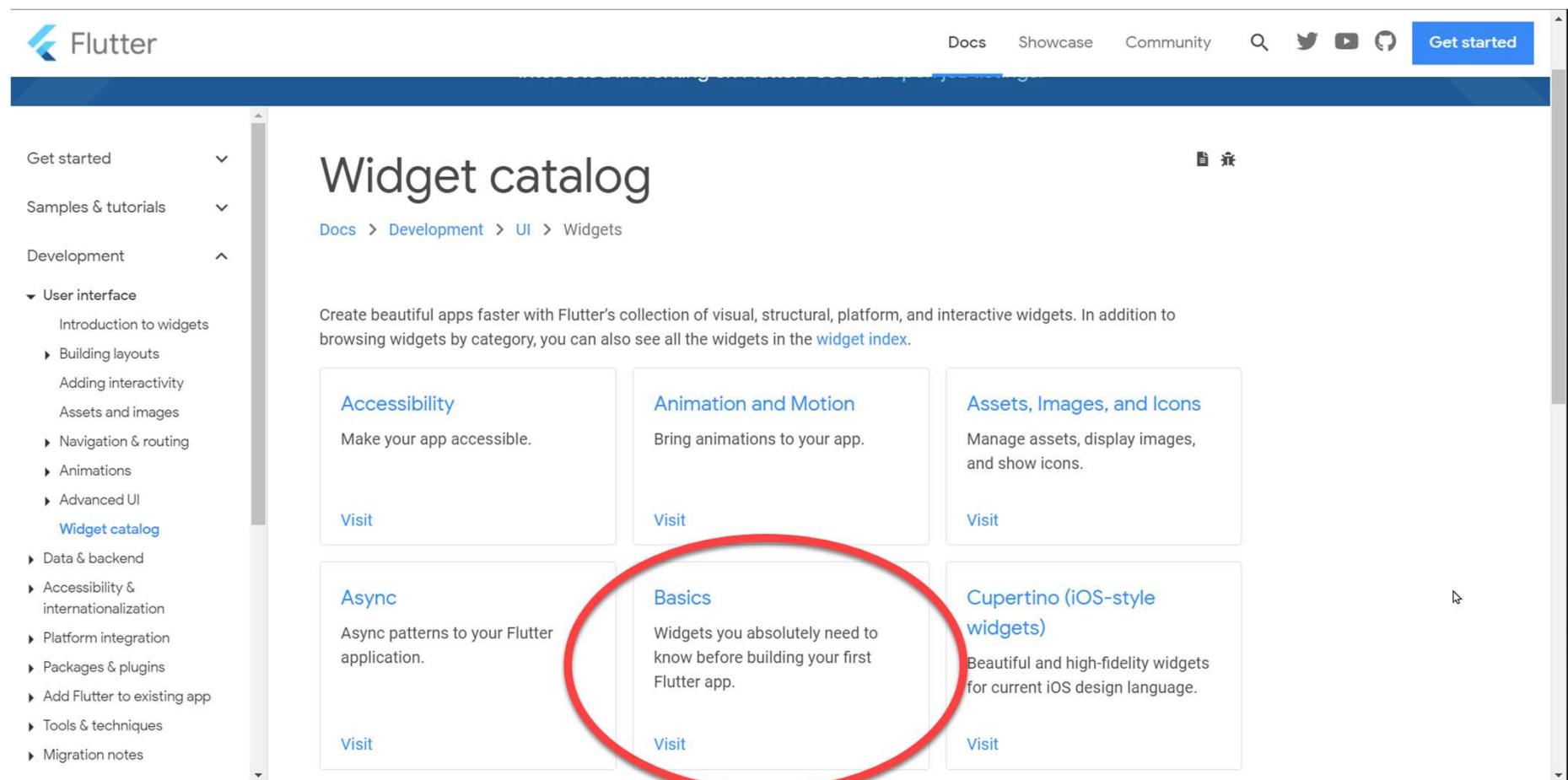
Default widgets to create the lay-out of your app.

# Creating the structure of your lay-out



- Scaffold()
- AppBar()
- Themes, fonts & colors
- Images
- Icons
- Rows, Columns

# The Widget Catalog



A screenshot of the Flutter documentation website showing the Widget catalog page. The page has a dark blue header with the Flutter logo, navigation links (Docs, Showcase, Community), and social media icons. A sidebar on the left contains a navigation tree for 'Development' and 'UI'. The main content area features a title 'Widget catalog' and a breadcrumb trail 'Docs > Development > UI > Widgets'. Below this, a paragraph explains the purpose of the catalog. The page is divided into several sections with cards: 'Accessibility', 'Animation and Motion', 'Assets, Images, and Icons', 'Async', 'Basics' (which is circled in red), and 'Cupertino (iOS-style widgets)'. Each card has a brief description and a 'Visit' button.

Flutter

Docs Showcase Community

Get started

Get started

Samples & tutorials

Development

User interface

- Introduction to widgets
- Building layouts
- Adding interactivity
- Assets and images
- Navigation & routing
- Animations
- Advanced UI
- Widget catalog**

Data & backend

Accessibility & internationalization

Platform integration

Packages & plugins

Add Flutter to existing app

Tools & techniques

Migration notes

## Widget catalog

Docs > Development > UI > Widgets

Create beautiful apps faster with Flutter's collection of visual, structural, platform, and interactive widgets. In addition to browsing widgets by category, you can also see all the widgets in the [widget index](#).

**Accessibility**  
Make your app accessible.  
[Visit](#)

**Animation and Motion**  
Bring animations to your app.  
[Visit](#)

**Assets, Images, and Icons**  
Manage assets, display images, and show icons.  
[Visit](#)

**Async**  
Async patterns to your Flutter application.  
[Visit](#)

**Basics**  
Widgets you absolutely need to know before building your first Flutter app.  
[Visit](#)

**Cupertino (iOS-style widgets)**  
Beautiful and high-fidelity widgets for current iOS design language.  
[Visit](#)

<https://flutter.dev/docs/development/ui/widgets>

# Learn the Basic widgets first

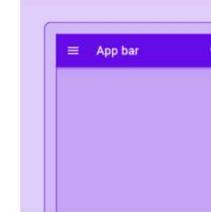
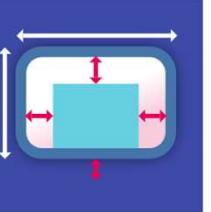


## Basic widgets

Docs > Development > UI > Widgets > Basics

Widgets you absolutely need to know before building your first Flutter app.

See more widgets in the [widget catalog](#).

 <b>Appbar</b> A Material Design app bar. An app bar consists of a toolbar and potentially other widgets, such as a TabBar and a FlexibleSpaceBar. <a href="#">Documentation</a>	 <b>Column</b> Layout a list of child widgets in the vertical direction. <a href="#">Documentation</a>	 <b>Container</b> A convenience widget that combines common painting, positioning, and sizing widgets. <a href="#">Documentation</a>
 <b>FlutterLogo</b> The Flutter logo, in widget form. This widget respects the IconTheme. <a href="#">Documentation</a>	 <b>Icon</b> A Material Design icon. <a href="#">Documentation</a>	 <b>Image</b> A widget that displays an image. <a href="#">Documentation</a>



# Using Scaffold()

Wrapping your content inside the `Scaffold()` widget for better lay-out options



# Scaffold class

Flutter > material > Scaffold class

Search API Docs

CLASSES

- AboutDialog
- AboutListTile
- AbsorbPointer
- Accumulator
- Action
- ActionChip
- ActionDispatcher
- ActionListener
- Actions
- ActivateAction
- ActivateIntent
- AlertDialog
- Align
- Alignment
- AlignmentDirectional
- AlignmentGeometry
- AlignmentGeometryTween
- AlignmentTween
- AlignTransition
- AlwaysScrollableScrollPh...
- AlwaysStoppedAnimation

CONSTRUCTORS

- Scaffold

PROPERTIES

- appBar
- backgroundColor
- body
- bottomNavigationBar
- bottomSheet
- drawer
- drawerDragStartBeh...
- drawerEdgeDragWid...
- drawerEnableOpenD...
- drawerScrimColor
- endDrawer
- endDrawerEnableOp...
- extendBody
- extendBodyBehindA...
- floatingActionButton
- floatingActionButton...
- floatingActionButton...
- hashCode

## Scaffold class

Implements the basic material design visual layout structure.

This class provides APIs for showing drawers, snack bars, and bottom sheets.

To display a snackbar or a persistent bottom sheet, obtain the `ScaffoldState` for the current `BuildContext` via `Scaffold.of` and use the `ScaffoldState.showSnackBar` and `ScaffoldState.showBottomSheet` functions.

Interactive App    Sample code

This example shows a `Scaffold` with a `body` and `FloatingActionButton`. The `body` is a `Text` placed in a `Center` in order to center the text within the `Scaffold`. The `FloatingActionButton` is connected to a callback that increments a counter.

Sample Code

Flutter 1.22.5 • 2020-12-10 22:48 • 7891006299 • stable

<https://api.flutter.dev/flutter/material/Scaffold-class.html>

# Using Scaffold()



- The Scaffold() widget implements the basic Material Design visual lay-out structure
- It can hold other widgets like
  - AppBar()
  - Drawer()
  - SnackBar()
  - BottomSheet(), and many more
- It has properties like
  - backgroundColor:...
  - body: ...
  - ...

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

void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text('Hello Flutter'),
        centerTitle: true,
      ),
    )
  ));
}
```

# Inside Scaffold()



- Again: **don't** just type text or add images inside a Scaffold Widget
  - Everything must be **wrapped** inside its own widget
- In the previous example `appBar: AppBar(...):`
  - `appBar:` is the property
  - `AppBar(...)` is the widget
- The AppBar Widget has a **property** `title` which holds a content **Widget** `Text(...)` as its value
- It has a **property** `centerTitle:` whose value is `true`

*“Every Flutter App  
is composed as a  
tree of widgets”*

# Adding a body



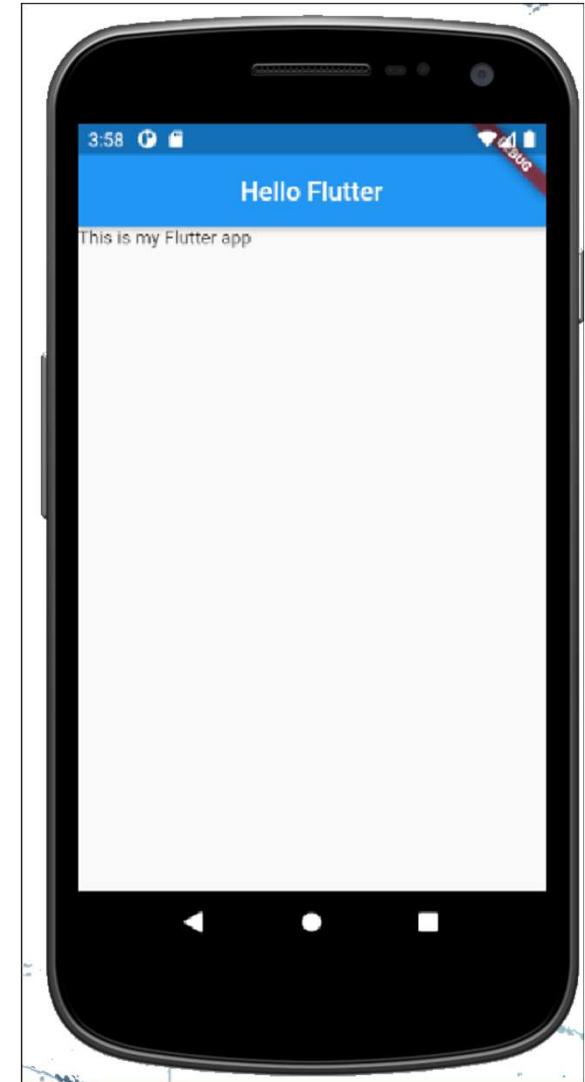
- All content below the `AppBar()` goes into the `body:` property
- This – again – is also a tree of Widgets.
- Convention:
  - Properties start with a `lowercase letter` and then `camelCase`
  - Widgets start with an `uppercase letter` and then `CamelCase`
  - Like in `appBar: AppBar()`
- A widget *always* is followed by `(...)`,

# Expanding the app



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

void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: Text('Hello Flutter'),
        centerTitle: true,
      ),
      body: Text('This is my Flutter app'),
    )
  ));
}
```



# Some background information



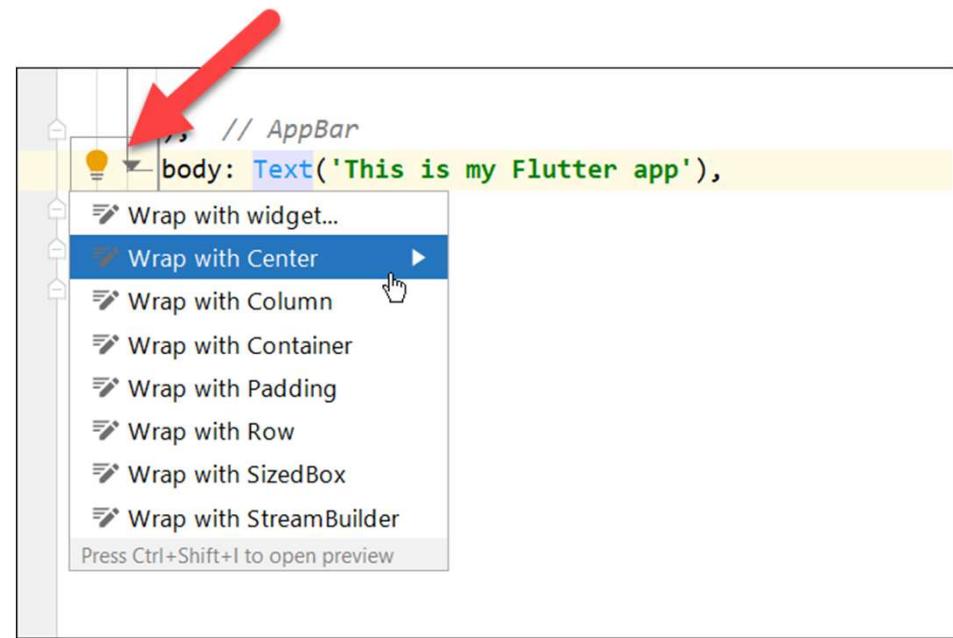
- Old: using the `new` keyword:
  - `title: new Text('Hello Flutter'),`
  - This still works, but is `not mandatory` anymore
- New: using `anonymous instances`
  - `title: Text('Hello Flutter'),`
- Android Studio places `//comment` if closing a Widget.
  - These are `not part` of the code
- Flutter refreshes its widget tree at `60fps`

# Tips



- Use **Ctrl+” “ (space)** to activate IntelliSense in Android Studio
- Use the lightbulb (**Alt+Enter**) to quickly wrap Widgets
- Use **Ctrl+Q** to activate Help, documentation, options

```
name: Scaffold(  
    appBar: AppBar(  
        title: new Text('Hello Flutter'),  
        centerTitle: true,  
  
        backgroundColor: ,           Color  
        leading: ,                  Widget  
        elevation: ,                double  
        actions: [],                 List<Widget>  
        actionsIconTheme: ,          IconThemeData  
        automaticallyImplyLeading: ,  bool  
        bottom: ,                   PreferredSizeWidget  
        bottomOpacity: ,             double  
        brightness: ,               Brightness  
        excludeHeaderSemantics: .   bool
```



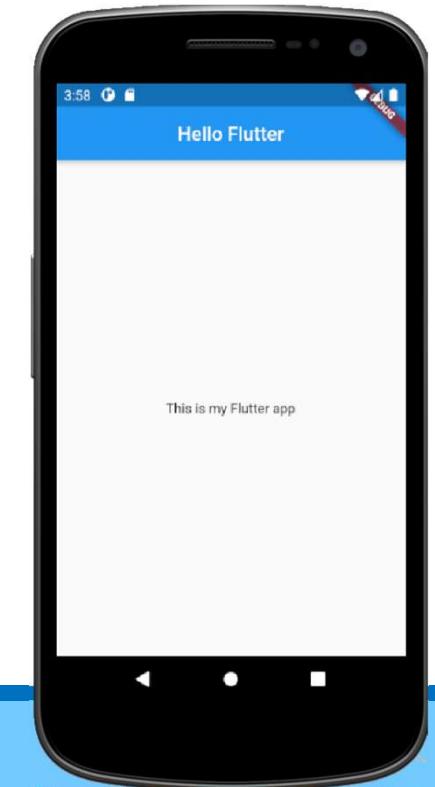
# The Center() Widget



- Use `Center()` to (duhhh) center items in their parent widget
- `Center()` has a property `child` that contains the child widget(s)

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

void main() {
  runApp(MaterialApp(
    home: Scaffold(
      appBar: AppBar(
        title: new Text('Hello Flutter'),
        centerTitle: true,
      ),
      body: Center(
        child: Text('This is my Flutter app')
      ),
    ),
  );
}
```



# FloatingActionButton()

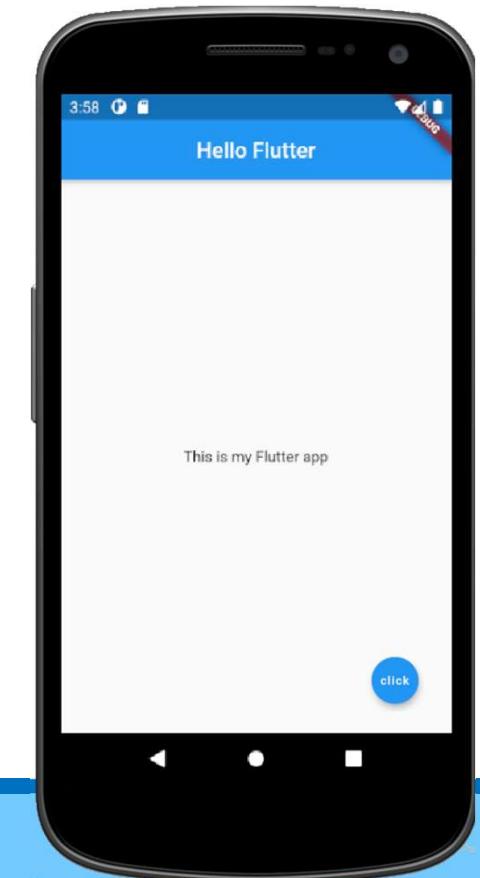


- Every Scaffold() can have 1 (one)

FloatingActionButton(...)

- The contents of the FAB are other Widgets
- They go inside the child: property

```
void main() {  
  runApp(MaterialApp(  
    home: Scaffold(  
      ...  
      floatingActionButton: FloatingActionButton(  
        child: Text('click'),  
      ),  
    ),  
  );  
}
```



# Mandatory properties



- Android Studio might be complaining that `FloatingActionButton` is missing a required `property onPressed`
- This function is called if the button is pressed
- Just add an empty function for now, to get rid of the error message. We're dealing with it later

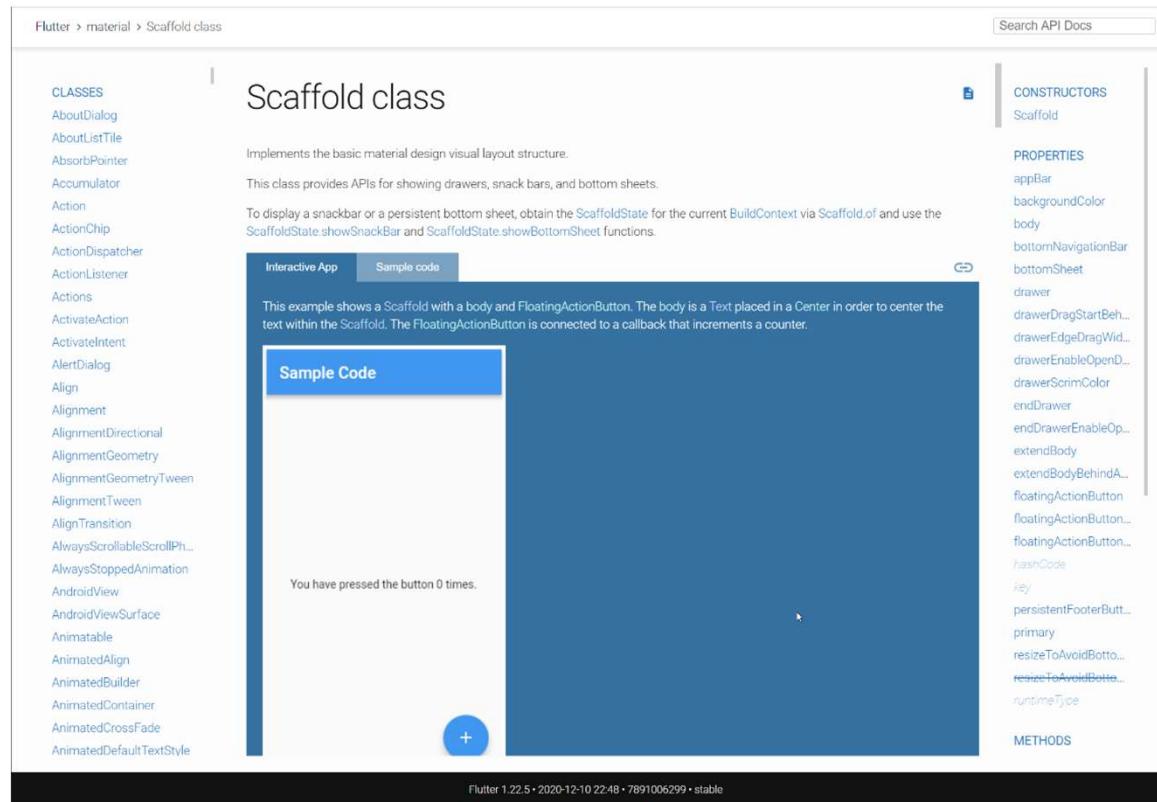
```
floatingActionButton: FloatingActionButton(  
    onPressed: (){}, // or: null  
    child: Text('click'),  
,
```

# More on Scaffold() and other widgets



- See official documentation at

[api.flutter.dev/flutter/material/Scaffold-class.html](https://api.flutter.dev/flutter/material/Scaffold-class.html)



The screenshot shows the official Flutter API documentation for the `Scaffold` class. The page is titled "Scaffold class" and includes the following sections:

- CLASSES**: A sidebar listing various Flutter classes like `AboutDialog`, `AboutListTile`, `AbsorbPointer`, etc.
- Scaffold class**: The main content area. It describes the class as implementing basic material design visual layout structure and providing APIs for drawers, snack bars, and bottom sheets. It includes a note about obtaining `ScaffoldState` via `Scaffold.of`.
- Interactive App** and **Sample code** buttons: The **Sample code** button is currently selected, showing a screenshot of a Flutter application. The screenshot displays a blue header bar labeled "Sample Code". Below it is a white content area with the text "You have pressed the button 0 times." To the right of the content area is a floating action button (FAB) represented by a blue circle with a white plus sign.
- CONSTRUCTORS**: A list containing the `Scaffold` constructor.
- PROPERTIES**: A list of properties including `appBar`, `backgroundColor`, `body`, `bottomNavigationBar`, `bottomSheet`, `drawer`, `drawerDragStartBeh...`, `drawerEdgeDragWid...`, `drawerEnableOpenD...`, `drawerScrimColor`, `endDrawer`, `endDrawerEnableOp...`, `extendBody`, `extendBodyBehindA...`, `floatingActionButton`, `floatingActionButton...`, `floatingActionButton...`, `hashCode`, `key`, `persistentFooterButt...`, `primary`, `resizeToAvoidBotto...`, `resizeToAvoidBotto...`, and `runtimeType`.
- METHODS**: A section currently empty.

At the bottom of the page, a footer notes "Flutter 1.22.5 • 2020-12-10 22:48 • 7891006299 • stable".

**aariXa**

> ict and business optimization

[www.aarixa.be](http://www.aarixa.be)

**Break... .**

Cont'd: 20:30 hr

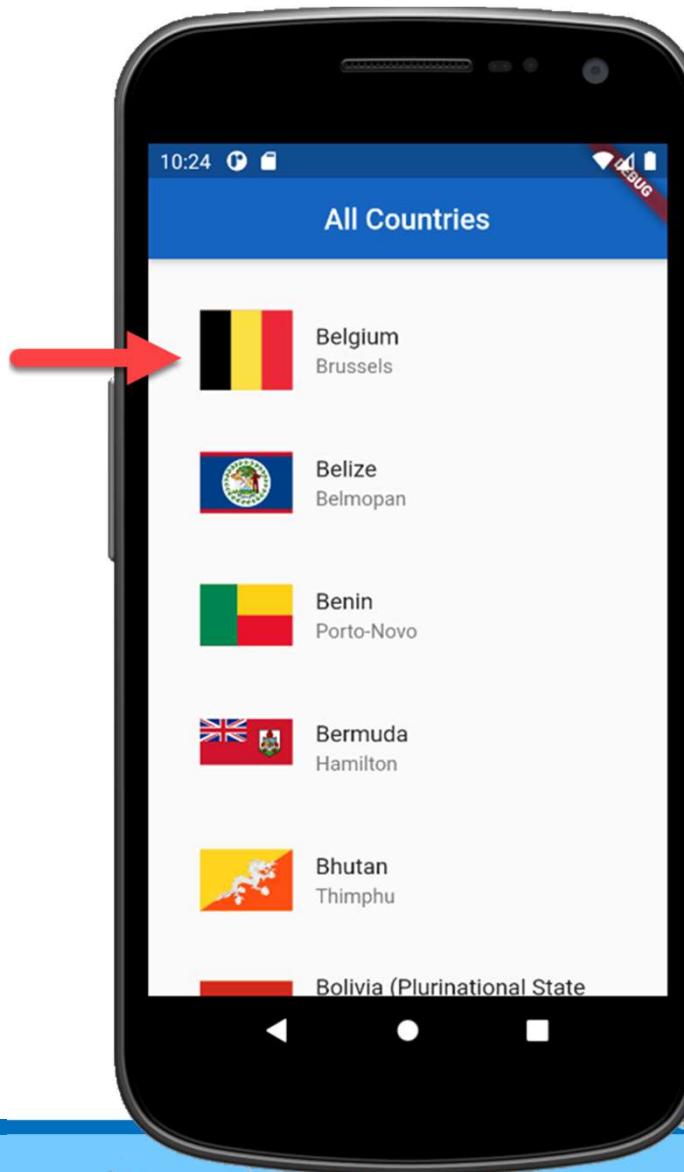




# Let's build a more complex app

Creating a Country Picker

# What are we going to build



# Using this API



REST Countries | View on GitHub | Issues Star 1,840

**REST COUNTRIES**  
Donate! Users Stay up-to-date

**API ENDPOINTS**  
All Name Full Name Code List of codes Currency Language Capital city Calling code Region Regional Bloc Response Example

**FILTER RESPONSE**

**SOURCES**

**SIMILAR PROJECTS**

**LICENSE**

**REST COUNTRIES**  
Get information about countries via a RESTful API  
build unknown

**DONATE!**  
The restcountries project has been acquired by apilayer, one of the leading providers of API microservices. We will keep supporting restcountries and providing it as a free solution for developers. We will finance this project fully and have turned off the donations feature.

**USERS**  
RESTCountries has over 1200 users, including:  
TTÜ  
Spotify International Pricing Index  
Gorillaz  
Wanderlust

<https://restcountries.eu/>

# Requirements



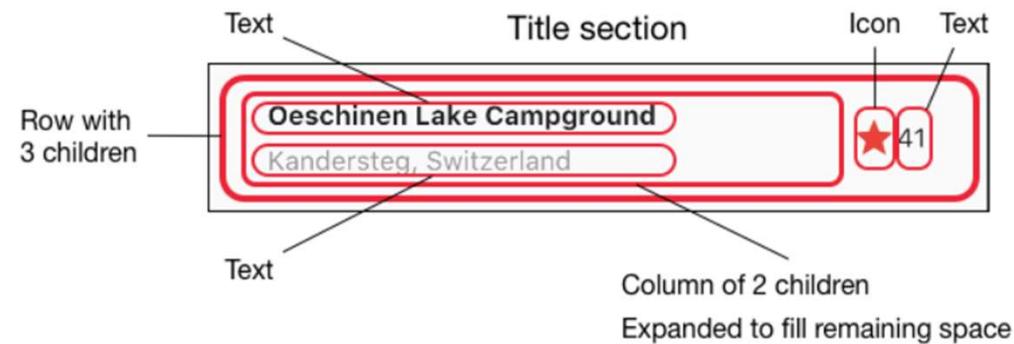
- Communicate over `http` with RestCountries API
  - Show data in a scrollable list
  - Show Flag (`.svg`) icons of a country
  - Show Country Name and Capital.
- 
- **Optional** – routing to details
  - **Optional** – search for specific country name

# Steps

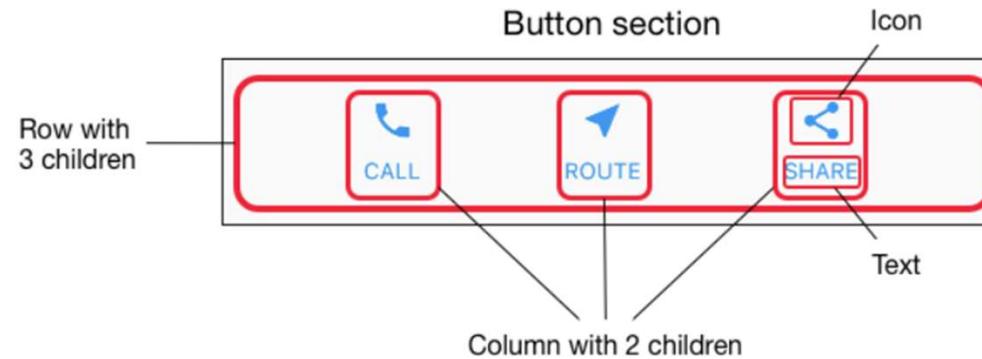


1. Design the UI
2. Create widgets and basic User Interface
3. Set state in the widgets
4. Add http for external API communication
5. Run the app!

# Development of the UI - wireframes



The second row, called the Button section, also has 3 children: each child is a column that contains an icon and text.



<https://flutter.dev/docs/development/ui/layout/tutorial>



# **Summary – used the following widgets**



- **Scaffold** and **AppBar()**
- **Column()**
- **Expanded()**
- **ListView()** and **ListView.builder()**
- **Padding()**
- **ListTile()**
- **SvgPicture()**
- **Text()**



# Flutter Next Steps

More information on Flutter

# Learn the widgets



Flutter > material > AboutDialog class

CLASSES

- AboutDialog
- AboutListTile
- AbsorbPointer
- Accumulator
- Action
- ActionChip
- ActionDispatcher
- ActionListener
- Actions
- ActivateAction
- ActivateIntent
- AlertDialog
- Align
- Alignment
- AlignmentDirectional
- AlignmentGeometry
- AlignmentGeometryTween
- AlignmentTween
- AlignTransition
- AlwaysScrollableScrollPh..

**AboutDialog class**

An about box. This is a dialog box with the application's icon, name, version number, and copyright, plus a button to show licenses for software used by the application.

To show an `AboutDialog`, use `showAboutDialog`.

The screenshot shows a blue arrow pointing from the text "Widget of the Week" to the word "AboutDialog". The "AboutDialog" text is overlaid on a play button icon.

<https://api.flutter.dev/>

# Stackoverflow tags



stack overflow Products [flutter] 261 3 4 +10 ?

Home PUBLIC Stack Overflow Tags Users FIND A JOB Jobs Companies TEAMS What's this? Free 30 Day Trial

## Questions tagged [flutter]

Ask Question Watch tag Ignore tag Learn more... Improve tag info Top users Synonyms

73,753 questions Newest Active Bountied 6 Unanswered More Filter

1 question with new activity

How do i navigate from one activity to another in flutter  
Hi I am pretty new to flutter and software development in general I can't seem to understand how do i navigate from one activity to another. What i really want is when i press the login button it ...  
0 votes 0 answers android flutter dart 3 views asked 1 min ago Rishali Siddiqui 29 6

Issues getting multiple values from List of Maps  
I'm trying to get the "alarm" array values from the following map. { "time": 12122, "alarm": [ { "title": "BPM LOW", "...  
0 votes 0 answers json flutter sorting dart hashmap 4 views asked 4 mins ago Michael Knight 15 1 6

GestureDetector OnTap Delay on Android Studio / Flutter / Nexus 6 API 30  
Think about extremly simple flutter application that contains one Container widget on the screen and GestureDetector on it. main goes to myApp, main and myApp widgets are in the same dart file. myApp ...  
1 answer android-studio flutter gesturedetector 10 views asked 17 mins ago Nurol Alacaatli 35 4

disabled double tap on better\_player in flutter

The Overflow Blog  
Fulfilling the promise of CI/CD Podcast 305: What does it mean to be a "senior" software engineer

Featured on Meta  
2020: a year in moderation Should the [which] tag be burninated?

Hot Meta Posts  
Closing an old upvoted question as duplicate of new unvoted questions

Custom Filters Create a custom filter

Watched Tags edit angular javascript typescript

Ignored Tags Add an ignored tag

Related Tags dart x 34083

<https://stackoverflow.com/questions/tagged/flutter>

# YouTube



YouTube NL

Zoeken

☰

Home

Trending

Abonnementen

Bibliotheek

Geschiedenis

Je video's

Je films

Later bekijken

Meer weergeven

ABONNEMENTEN

- The Net Ninja
- NOS Sport
- Scott Hanselman
- Global Cycling Net...
- A&P-REACTS
- WatchMojo.com

Introducing Flutter

ALLES AFSPELEN

Flutter by Google

84 video's • 385.146 weergaven • Laatst geüpdate op 12 nov. 2019

Get started at <https://flutter.io> today.

Google Developers

Introducing Flutter

1 1:53

Alibaba used Flutter to build 50+ million user Xianyu app (Flutter Developer Story)

2 2:51

Google Developers

Hamilton app built with Flutter and featured on iOS and Android (Flutter Developer Story)

3 2:37

Google Developers

Abbey Road Studios (Flutter Developer Story)

4 3:16

Google Developers

Flutter Live Keynote Recap

5 11:09

Google Developers

Flutter Release Preview 2

6 3:01

Google Developers

Building your first Flutter Widget

7 10:05

Google Developers

<https://www.youtube.com/playlist?list=PLOU2XLYxmsIJ7dsVN4iRuA7BT8XHzGtCr>

# Firebase



Firebase Products Use Cases Pricing Docs Community Support Search English Go to console

## Documentation

Overview Guides Reference Samples Libraries

Guides

Get started with Firebase

- Add Firebase to an app
- Add Firebase to a game
- Add Firebase to a server
- Use Firebase with a framework

Flutter

Set up projects programmatically

Manage your Firebase projects

Platforms and frameworks

Prototype and test with Emulator Suite

Analytics

Extensions

BUILD

Authentication

Google is committed to advancing racial equity for Black communities. [See how.](#)

Firebase > Docs > Guides

Add Firebase to your Flutter app

iOS Android Send feedback

Follow this guide to add Firebase products to a Flutter app.

**Note:** Firebase supports frameworks like Flutter on a best-effort basis. These integrations are not covered by Firebase Support and may not have full feature parity with the [official Firebase SDKs](#).

iOS Android

Getting started with Firebase on Flutter

Firecasts For Firebase Developers

Prerequisites

- Install your preferred [editor or IDE](#).

Table of contents

Prerequisites

Step 1: Create a Firebase project

Step 2: Register your app with Firebase

Step 3: Add a Firebase configuration file

Step 4: Add FlutterFire plugins

Try out an example app with Analytics

Next steps

<https://firebase.google.com/docs/flutter/setup>

# Q & A



[github.com/PeterKassenaar/aarixa](https://github.com/PeterKassenaar/aarixa)



## PETER KASSENAAR

Pro-level front-endtrainingen. 20+ jaar ervaring.

### Introductie StudieMix

StudieMix - jouw balans tussen zelfstudie en persoonlijke feedback

Liggen bij jou ook vaak de dagen dat je e-learnings trainingen op de plank? Is het lastig om online te blijven leren? Wij kunnen je helpen!

StudieMix lessen zijn gebaseerd op front-end webdevelopment. In een gestructureerde periode van zes weken helpen we een goed beginnend programmeur te worden. Samen met een kickoff leer je wanneer het jou uitkomt. Samen met een programma met vast programma haalt hij de indeling en



<http://kassenaar.pages.co/studiemix>

[www.kassenaar.com](http://www.kassenaar.com)