

Flutter Fundamentals Module - Introduction

Peter Kassenaar –
info@kassenaar.com

Peter Kassenaar

- Trainer, author, developer – since 1996
- Specialty: "*Everything JavaScript*"
- JavaScript, ES6, Angular, NodeJS, TypeScript, jQuery, React, Vue, etc.

www.kassenaar.com

info@kassenaar.com

[BlueSky](#)



Rijksoverheid





Sr. frontend developer ProductIP, Ede, NL, (50%)

The screenshot shows the homepage of the ProductIP website. At the top, there is a navigation bar with links for Home, News, Events, Services, Pricing, Contact, Jobs, Log In, and Register Now. The Register Now button is highlighted in green. To the right of the navigation bar are icons for user profile, search, and a blue decorative element. Below the navigation bar, five large, metallic spheres are suspended by thin wires, arranged horizontally. To the right of these spheres is a vertical sidebar with icons for a star (10), a magnifying glass, and a person. The main content area features a paragraph about regulatory and compliance information, followed by the tagline "Knowledge • SaaS • Impact" and a sub-tagline "Legal advice is everywhere. Now you can translate this into impact". At the bottom left is a circular logo with a fingerprint pattern. A horizontal banner at the very bottom contains the text "Building a future proof business with an operation and supply chain in...".

ProductIP provides relevant regulatory and compliance information for you to make mission critical product and supply chain business decisions

Knowledge • SaaS • Impact

Legal advice is everywhere. Now you can translate this into impact

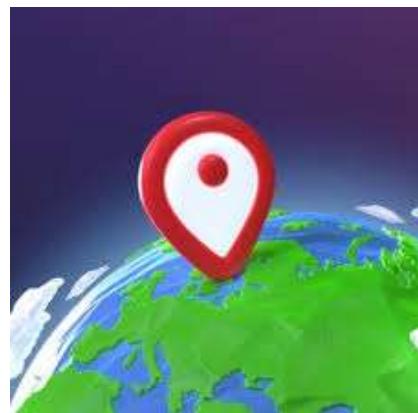
www.productip.com



peterkassenaar



PeterKassenaar



PeterKassenaar



pkas06

Mijn Vijfhart - slides

The screenshot shows the Mijn Vijfhart IT-Opleidingen website interface. At the top, there is a red header bar with the logo 'vijfhart IT-OPLEIDINGEN'. Below the header, a navigation bar includes links for 'HOME', 'REACT FUNDAME...', 'CATALOGUS', 'PETER KASSENA...', and 'HELP'. The main content area features a woman's face in a blue-toned photograph. Below the photo, there are buttons for 'TERUG NAAR BEGELEIDEN' and 'STOP MET MEELOOPEN'. The title 'React Fundamentals' is displayed, along with a progress bar showing '0% ACTIVITEITEN' and a link 'Meer resultaten >'. On the left, a sidebar lists 'Wat kun je hier verwachten?' with items like 'Uitleg online leeromgeving' and 'Persoonlijke instellingen aanpassen', each with a 'START' button. On the right, a sidebar titled 'AGENDA' shows sections for 'PLANNING', 'BERICHTEN', and 'DISCUSSIE', with links to 'Open agenda', 'Berichten & notificaties', and 'Recente reacties: Geen recente reacties.'

Github – latest changes

The screenshot shows a GitHub repository page for the 'maritiem' repository, which is public and owned by PeterKassenaar. The repository has one branch ('main') and no tags. The README.md file has been updated twice in the last two minutes. The repository has 0 stars, 1 watching, and 0 forks. It includes sections for Releases (no releases published) and Packages (no packages published). The repository description is "Slides and sample code on the training Flutter Fundamentals, Maritieme IT, 2025".

PeterKassenaar / maritiem

Type / to search

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

maritiem Public

main 1 Branch 0 Tags

Go to file Add file Code

PeterKassenaar Update README.md e983f3b · now 2 Commits

.gitignore Initial commit 2 minutes ago

LICENSE Initial commit 2 minutes ago

README.md Update README.md now

README MIT license

Octotree >

maritiem

Slides and sample code on the training Flutter Fundamentals, Maritieme IT, 2025

Links

- General Flutter Fundamentals Repo: <https://github.com/PeterKassenaar/flutter-fundamentals>
- ...

About

Slides and sample code on the training Flutter Fundamentals, Maritieme IT, 2025

Readme

MIT license

Activity

0 stars

1 watching

0 forks

Releases

No releases published

Create a new release

Packages

No packages published

Publish your first package

github.com/PeterKassenaar/maritiem

github.com/PeterKassenaar/flutter-fundamentals

The screenshot shows the GitHub repository page for `flutter-fundamentals`. The repository is public and owned by `PeterKassenaar`. The main page displays the repository's structure, recent commits, and a brief description.

Repository Structure:

- `main` branch (selected)
- 2 Branches
- 0 Tags

Recent Activity:

Commit	Message	Date
<code>examples</code>	added routing examples	15 hours ago
<code>.gitignore</code>	Updated .gitignore	5 years ago
<code>LICENSE</code>	Initial commit	5 years ago
<code>README.md</code>	Initial commit	5 years ago

Files:

- `README`
- `MIT license`

Description:

Example code on the training Flutter Fundamentals, by internationally acclaimed trainer Peter Kassenaar

Repository Statistics:

- Pin: 1
- Unwatch: 1
- Fork: 0
- Star: 0

About:

Example code on the training Flutter Fundamentals, by internationally acclaimed trainer Peter Kassenaar

Links:

- Readme
- MIT license
- Activity
- 0 stars
- 1 watching
- 0 forks

Releases:

No releases published

[Create a new release](#)

Packages:

No packages published

About you...



Tell us a little bit about yourself



Knowledge of **programming**, **apps**, other (mobile) frameworks?

Tell us a little bit about your **role** and your **projects**.

What are your **expectations** of this course?

Material



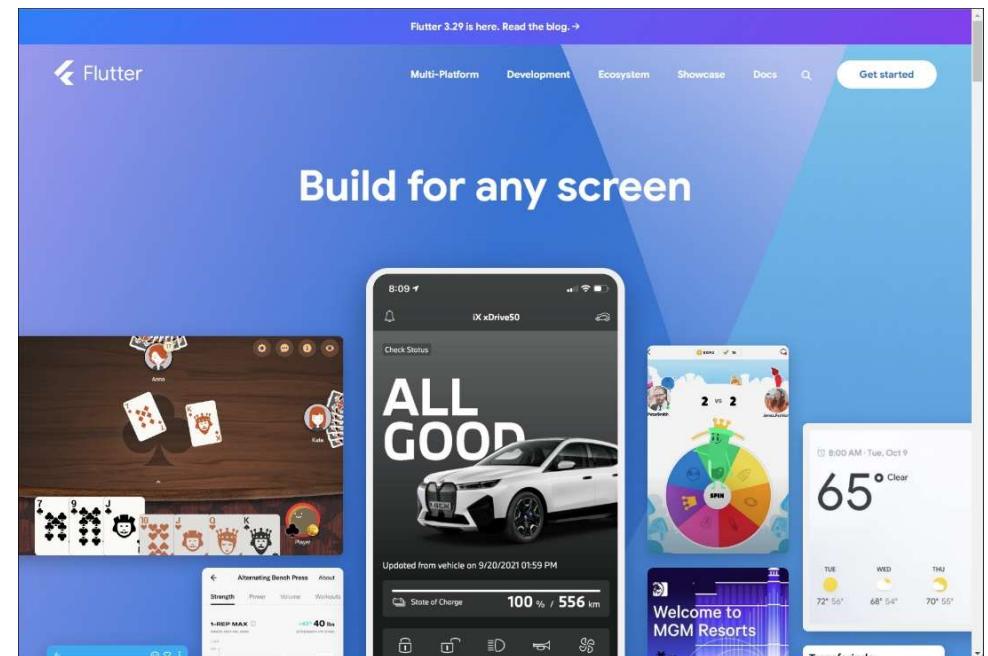
Software (IntelliJ + libs + Flutter)

Hardware (target device [phone, laptop, Windows, Linux])

Handouts (PDF, Github)

Workshops (In Presentations)

Websites (online)



<https://flutter.dev/>

Contents at a glance



- Day 1 – Flutter introduction + Lay-outs
- Day 2 – State Management + communication
- Day 3 – Various advanced concepts + publishing

Agenda - details



- Introduction – overview of the Flutter landscape
- Flutter tooling – installation
- Hello World –the structure and architecture of Flutter apps.
- A Dart Primer
- Zooming in on Flutter:
 - Components – Stateless vs. Stateful

Agenda – cont'd



- **Widgets** - introduction
- The Flutter **layout system** - `Scaffold()` and more
 - Using Images and assets
 - More layout widgets
 - `Button()`, `Icon()`, `Container()`, `Padding()`
 - `Row()`, `Column()`, properties, children and more
- **Working with data**
 - `ListView()`, `Card()`, state in components

Agenda – cont'd



- State management in various ways
- Communicating with external API's
- gRPC communication + alternatives
- Using Flutter Router
 - Navigating through your app using `TabBar()` and `Drawer()`
- Gestures, Custom Themes, Custom Plugins
- Deploy your app to production
 - Creating a production build
 - Publishing to jFrog Artifactory

Labs and example code



1. Labs/Exercises

- In the PDF's and in the Github-repo. But: feel free to deviate.
Adapt to suit your own needs! (hobby, work, current projects)

2. Example code

- Executions of the exercises, small projects
- Work in progress – let me know of additions/errors!
- github.com/PeterKassenaar/flutter-fundamentals

Advance warning



SORRY

*You're NOT going to be a Flutter
expert in 3 days...*

(hey, I don't consider myself an expert!)

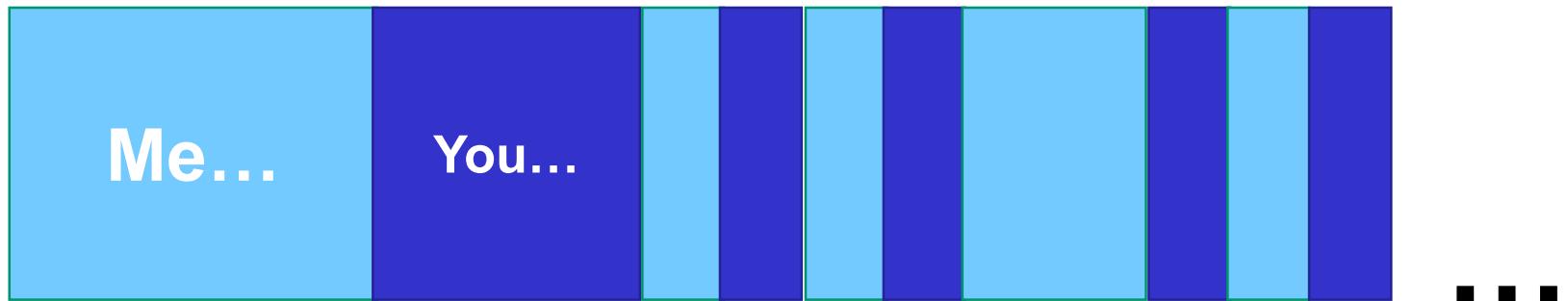
But, you'll have:

General Goals



1. An overview of the **structure and architecture** of Flutter applications
2. A **basic understanding** of some of the main widgets.
You can then Google for other widgets yourself
3. Some **hands-on experience** in creating Flutter Applications, Widgets and Logic

Overall process



Questions?



What is Flutter?

General introduction - What can you create using Flutter?

What is Flutter?

Flutter 3.29 is here. Read the blog. →

Flutter Multi-Platform Development Ecosystem Showcase Docs Get started

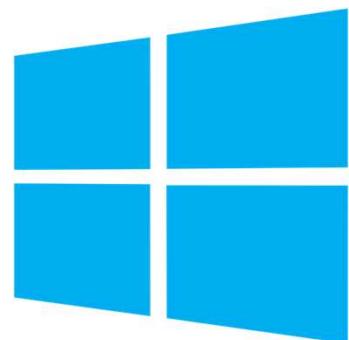
Build for any screen

The image shows a collection of screenshots from different Flutter-based apps. At the top left is a card game interface with four players (Alice, Cleo, Anna, Kate) and their hands. Next is a car status app for an 'iX xDrive50' showing a white car, battery level at 100% (556 km range), and vehicle stats. Below that is a roulette wheel game with two players, Peter Smith and James Parker, and a 'SPIN' button. To the right is a weather app for San Francisco showing a clear sky at 65°. Further down is a fitness app showing a bar chart for 'Alternating Bench Press' with a total weight of 40 lbs. At the bottom left is a video player showing a person's face. On the far right is a travel app showing a map from San Francisco to another location.

<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 3.29 – February 2025



A screenshot of a Medium.com post. The header features a large blue heart-shaped button with the Flutter logo and the text "Flutter 3.29". To the left of the button is a small blue bird icon. The Medium.com interface is visible at the top, including the "Medium" logo, a search bar, and user account options.

<https://medium.com/flutter/whats-new-in-flutter-3-29-f90c380c2317>

Google Flutter blog



Medium  Search  Write  Sign up  Sign in 

What's new in Flutter 3.29

Enhancing Performance and Fidelity Across Platforms

 Kevin Chisholm · [Follow](#)
Published in [Flutter](#) · 9 min read · 5 days ago

 2.9K  23   

Introduction

Dive into Flutter 3.29! This release refines development and boosts performance, with updates to Impeller, Cupertino, DevTools and more. With 104 unique authors contributing this release cycle, Flutter 3.29 showcases the community's dedication. Let's explore what's new!

Framework

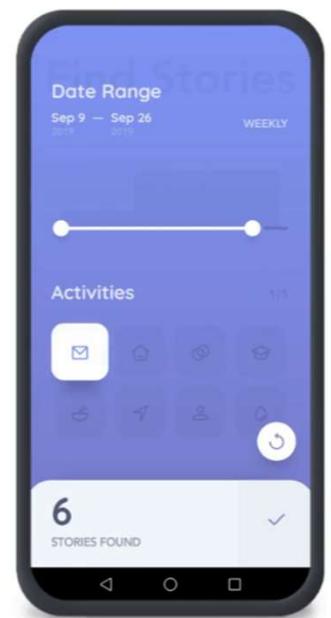
Cupertino updates

<https://medium.com/flutter/whats-new-in-flutter-3-29-f90c380c2317>

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 the programming language
 - High performance on rendering, scrolling, navigation, animation



Flutter Pro's



- Fast development
- Multi-platform
- Single code base (mostly!)
- 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

The screenshot shows a website header for 'COGNITEQ' with navigation links for 'INDUSTRIES', 'EXPERTISE', 'SERVICES', 'PORTFOLIO', and 'ABOUT US'. A blue 'CONTACT US' button is also visible. The main content features a large, bold, blue title: 'FLUTTER PROS AND CONS: IS IT A GOOD CHOICE FOR YOUR APP?'. Below the title is a light blue sidebar containing the 'COGNITEQ' logo and a small image of a smiling man wearing glasses. At the bottom right of the sidebar is a grey scroll bar with up and down arrows.

<https://www.cogniteq.com/blog/flutter-pros-and-cons-it-good-choice-your-app>



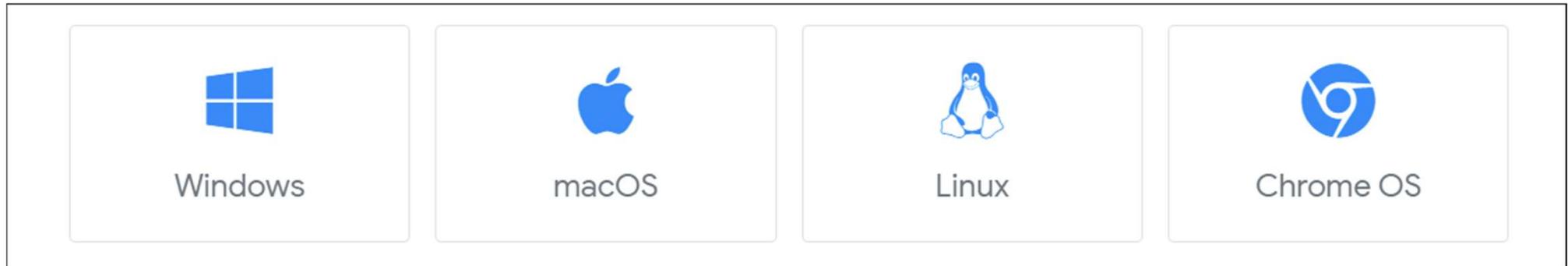
Flutter requirements

Prerequisites, hardware and software

Flutter Requirements - Hardware



- PC, Linux or Mac
 - Optional: physical phone/tablet to test application when building mobile
- iOS applications? - You'll need a Mac + Xcode
- Windows? - You'll need Visual Studio + C++



Flutter Requirements - software



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



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

[flutter_windows_3.29.0-stable.zip](#)

For other release channels, and older builds, check out the [SDK archive](#).

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



Installing Flutter

Prerequisites, Installation of hardware and software

Official documentation



Choose your first type of app

Get started > Install > Windows



Android
Recommended



Web



Desktop

Your choice informs which parts of Flutter tooling you configure to run your first Flutter app. You can set up additional platforms later. *If you don't have a preference, choose [Android](#).*

Developing in China

If you want to use Flutter in China, check out [using Flutter in China](#). If you're not developing in China, ignore this notice and follow the other instructions on this page.

如果你正在中国的网络环境下配置 Flutter, 请参考 [在中国网络环境下使用 Flutter 文档](#).

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

Windows – recommended order

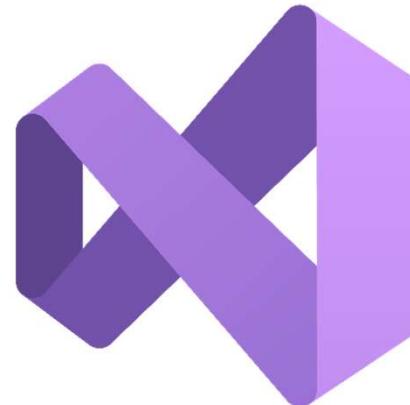


1. Install Visual Studio
2. Install IntelliJ
3. Install Flutter plug-in for IntelliJ
4. Install Flutter SDK
5. Update Windows PATH variable
6. Run `flutter doctor`, fix any possible problems

Install



- Step 1, 2 (installation VS, IntelliJ), already taken care of!
- If you do this yourself: downloading + installing takes a *lot of time*
 - think: +2 hrs (!)

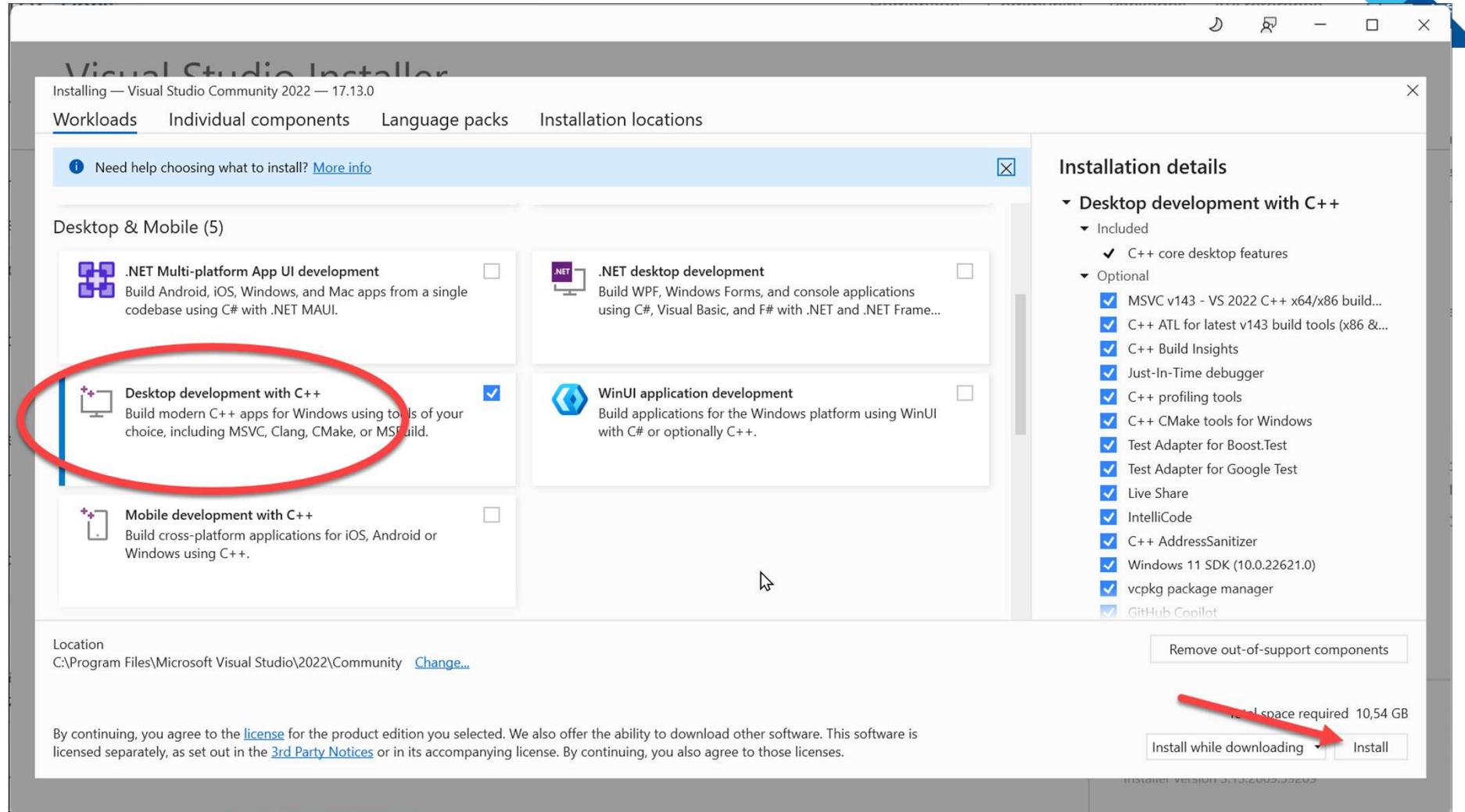


1/6. Installing Visual Studio 2022



- Visual Studio Community Edition will do
- Select Desktop Development with C++

*"you **must** install Visual Studio (not just VS Code) to develop Windows desktop apps with Flutter. This is because Flutter for Windows relies on native C++ build tools that come with MSBuild, the Windows SDK, and the necessary compiler toolchains"*



Installing Visual Studio



Visual Studio Installer

Installed Available

 Visual Studio Community 2022

Downloading and verifying: 879 MB of 2,93 GB (9 MB/sec)

29% 

Installing: package 170 of 448

15% 

Microsoft.VisualStudio.MinShell.Shared.Msi

Start after installation

[Release notes](#)

Will take some time...



Visual Studio Installer

Installed Available

All installations are up to date.



Visual Studio Community 2022

17.13.0

Powerful IDE, free for students, open-source contributors and individuals

[Release notes](#)

[Modify](#)

[Launch](#)

Done installing

Visual Studio has been successfully installed. We recommend rebooting soon to clean up any temporary files.



[OK](#)

2/6. IntelliJ Community Edition (free)



JETBRAINS

Developer Tools Team Tools Education Solutions Support Store

IntelliJ IDEA JetBrains IDEs Coming in 2025.1 What's New Features ▾ Resources Pricing Download

Other Versions

Version 2024.3 2024.3.3 ▾

IntelliJ IDEA Ultimate	IntelliJ IDEA Community Edition	
2024.3.3 - Linux aarch64 (tar.gz)	2024.3.3 - Linux aarch64 (tar.gz)	Version: 2024.3.3 (Release notes) Build: 243.24978.46 Released: 12 February 2025
2024.3.3 - Linux x86_64 (tar.gz)	2024.3.3 - Linux x86_64 (tar.gz)	Major version: 2024.3 Released: 13 November 2024
2024.3.3 - Windows ARM64 (exe)	2024.3.3 - Sources Archive (zip)	IntelliJ IDEA Ultimate third-party software IntelliJ IDEA Community Edition third-party software
2024.3.3 - Windows x64 (exe)	2024.3.3 - Windows ARM64 (exe)	
2024.3.3 - Windows x64 ZIP Archive (zip)	2024.3.3 - Windows x64 (exe)	
2024.3.3 - macOS (dmg)	2024.3.3 - Windows x64 ZIP Archive (zip)	
2024.3.3 - macOS Apple Silicon (dmg)	2024.3.3 - macOS (dmg)	
	2024.3.3 - macOS Apple Silicon (dmg)	

A red oval highlights the "IntelliJ IDEA Community Edition" section in the table.

<https://www.jetbrains.com/idea/download/other.html> |

Always check the docs (AI is behind!)



Flutter 3.29 is here with a bouquet of performance and fidelity improvements for your apps! Learn more

Flutter Docs

Get started

Set up Flutter (circled in red)

Learn Flutter

Stay up to date

App solutions

User interface

Introduction

Widget catalog

Layout

Adaptive & responsive design

Design & theming

Interactivity

Assets & media

Navigation & routing

Animations & transitions

Start building Flutter native desktop apps on Windows

Get started > Install > Windows > Make Windows desktop apps

Verify system requirements

To install and run Flutter, your Windows environment must meet the following hardware and software requirements.

Hardware requirements

Your Windows Flutter development environment must meet the following minimal hardware requirements.

Requirement	Minimum	Recommended
x86_64 CPU Cores	4	8
Memory in GB	8	16

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

Installing IntelliJ Community Edition



The screenshot shows the IntelliJ IDEA Community Edition Setup window. The title bar reads "IntelliJ IDEA Community Edition Setup". The main area is titled "Installing" with the sub-instruction "Please wait while IntelliJ IDEA Community Edition is being installed." Below this, a progress bar indicates the extraction of "intellij.fullLine.go.jar", with a green bar partially filled. A "Show details" button is located below the progress bar. At the bottom are "< Back" and "Next >" buttons.

Completing IntelliJ IDEA Community Edition Setup

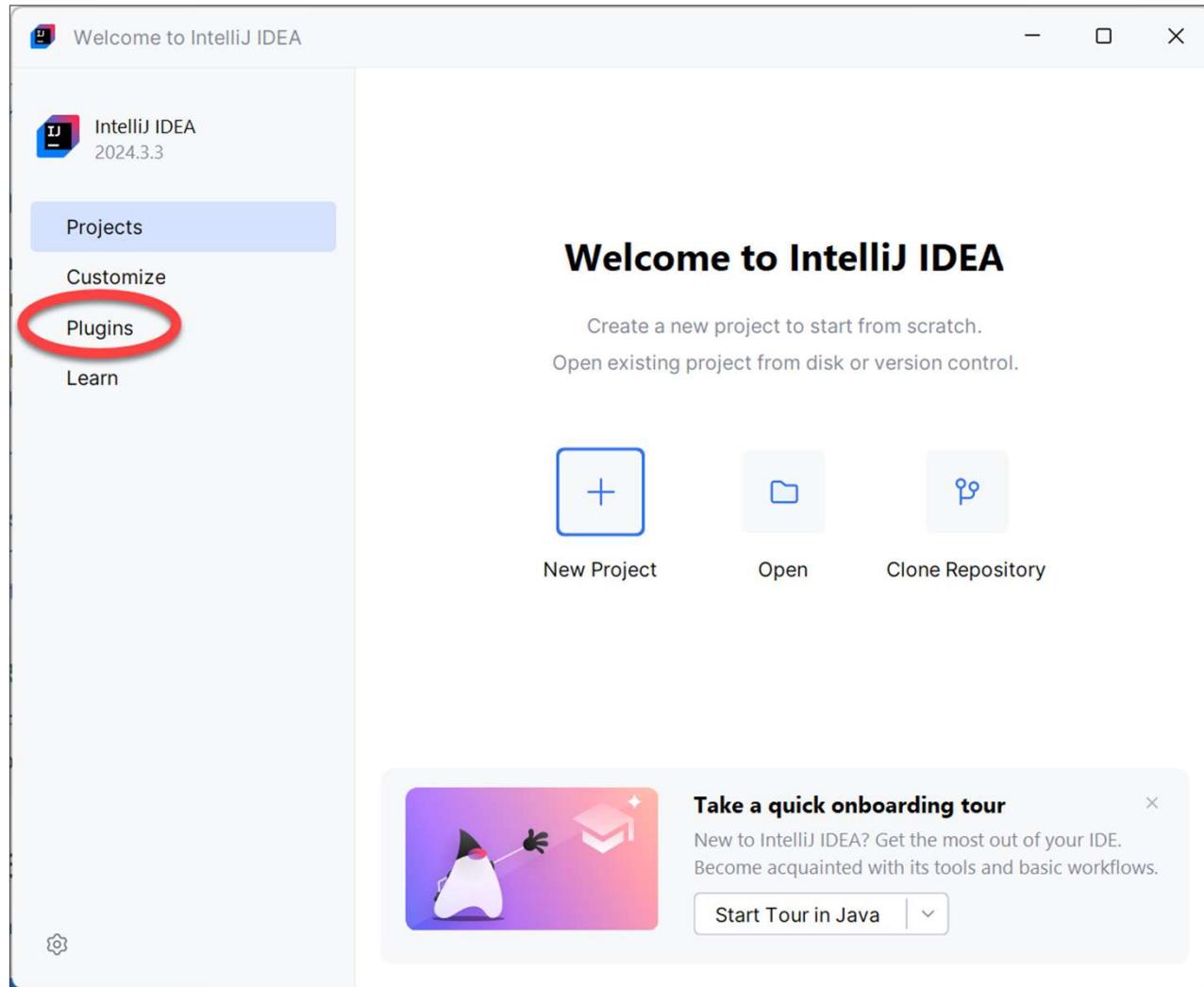
IntelliJ IDEA Community Edition has been installed on your computer.

Click Finish to close Setup.

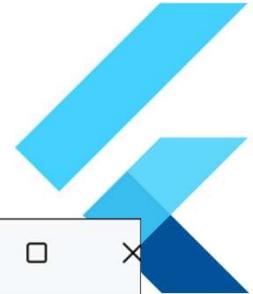
Run IntelliJ IDEA Community Edition

< Back **Finish** Cancel

3/6. Install Flutter plugin



Install plugin in IntelliJ



The screenshot shows the IntelliJ IDEA interface with the Marketplace tab selected. A red circle highlights the search bar containing the text 'flutter'. Below the search bar, the results list shows four items:

- Flutter** by Google (20.3M downloads, 4.01 rating) - An 'Install' button is highlighted with a red arrow pointing to it.
- FlutterJsonBeanFa...** by ruiyui (412K downloads, 4.89 rating)
- FlutterAssetsGener...** by 蔡锐 (115.9K downloads, 4.78 rating)
- flutter_add_image** by jun7572 (52.1K downloads, 4.71 rating)

On the right side, the 'Flutter' plugin page is displayed with tabs for Programming Language, Reviews, Additional Info, Overview, and What's New. A red arrow points to the 'Install' button on this page.

Install Plugin, Ignore warning, Restart IDE

4/6. Install Flutter SDK



- Add Flutter to Command Line, update Path and more.
- Following the procedure from the docs:
<https://docs.flutter.dev/get-started/install/windows/desktop>
 - Adapt to YOUR operating system!
 - Again: read the docs.
- Important: extract to a directory not containing special characters or spaces
 - Suggestion: ...%userprofile%/dev/
 - We will use this one. But any other folder will work

Download flutter_xxx_stable.zip



Install the Flutter SDK

To install the Flutter SDK, you can use the VS Code Flutter extension or download and install the Flutter bundle yourself.

Use VS Code to install

Download and install

Download then install Flutter

To install Flutter, download the Flutter SDK bundle from its archive, move the bundle to where you want it stored, then extract the SDK.

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

flutter_windows_3.29.0-stable.zip



For other release channels, and older builds, check out the [SDK archive](#).

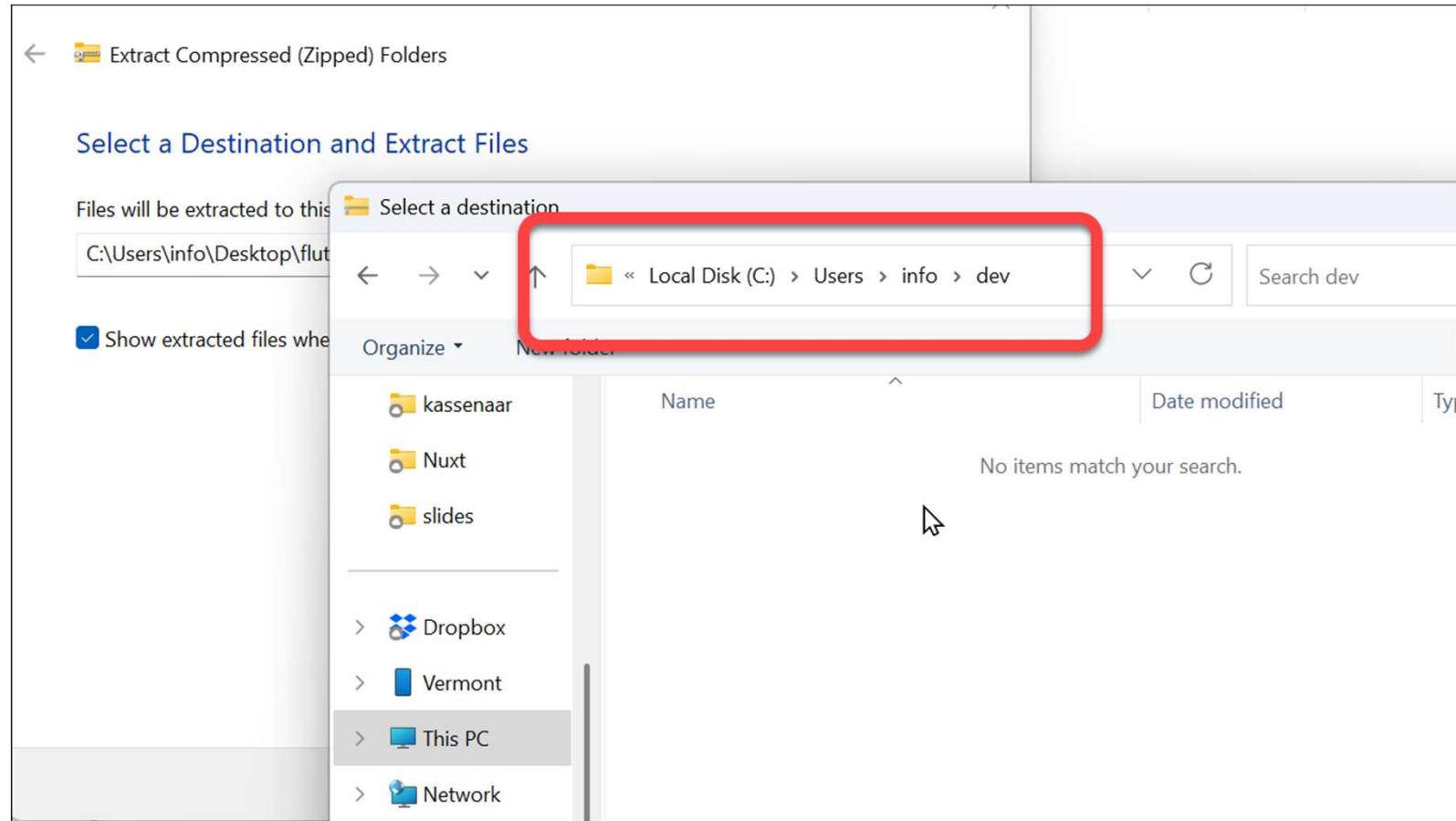
The Flutter SDK should download to the Windows default download directory: `%USERPROFILE%\Downloads`.

If you changed the location of the Downloads directory, replace this path with that path. To find your Downloads directory location, check out this [Microsoft Community post](#).

2. Create a folder where you can install Flutter.

Consider creating a directory at `%USERPROFILE%` (`C:\Users\{username}`) or `%LOCALAPPDATA%` (`C:\Users\{username}\AppData\Local`).

Unzip in %UserProfile% /dev directory



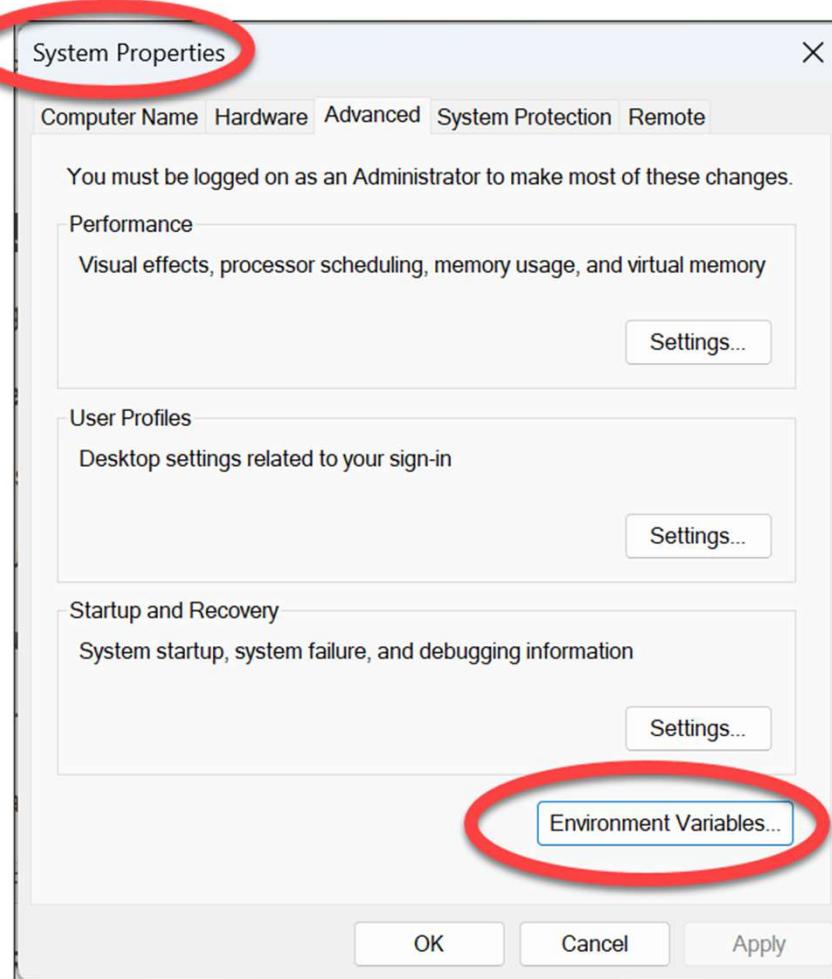
This is on MY computer. Update for your own pc!

C:\Users\info\dev

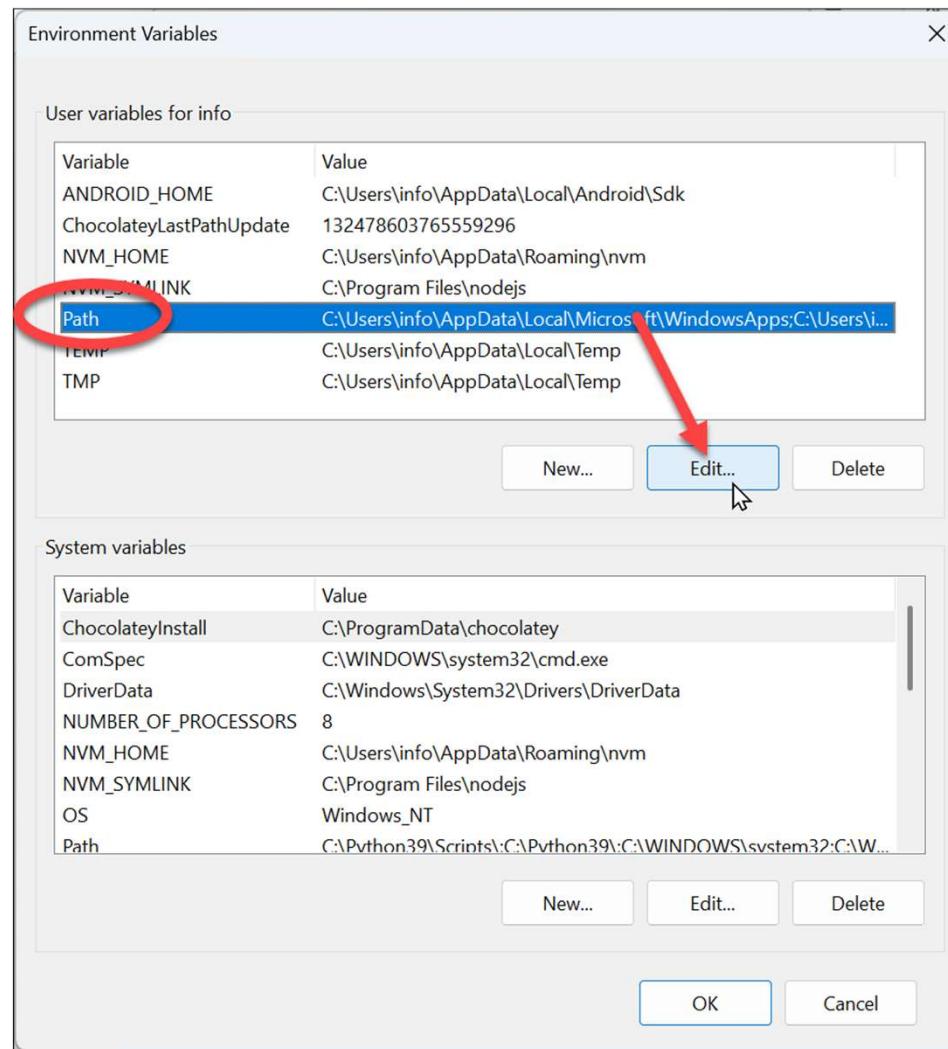
5/6. Windows: update PATH.



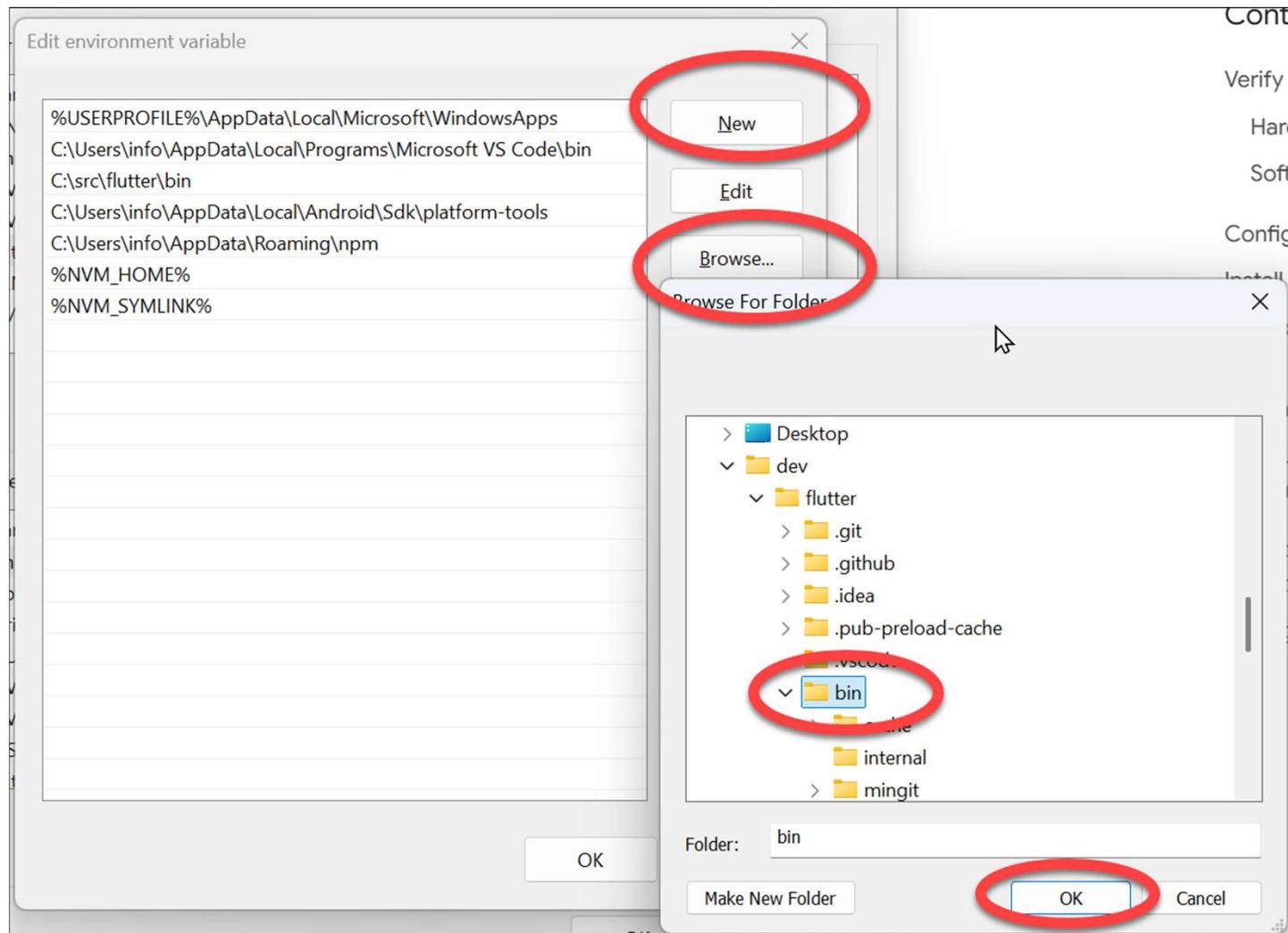
3 steps to update your
Environment Variables



Step 2/3

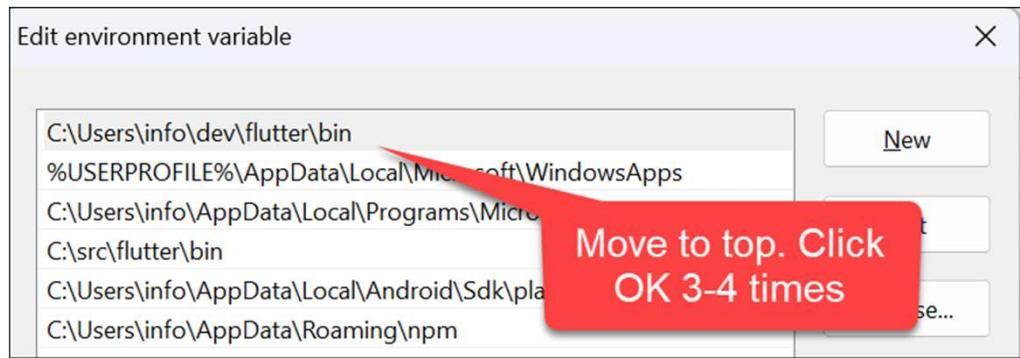


Step 3/3



Add %userprofile%/dev/flutter/bin to Path

Just to make sure: place on top



To enable changes, *close and reopen* any existing command prompts and PowerShell instances.

6/6. Check: flutter doctor



- Flutter doctor is a [diagnostics tool](#)
- Check if Flutter is OK, devices, and more

```
PS C:\Users\info> flutter doctor
Building flutter tool ...
Running pub upgrade ...
Resolving dependencies ... (2.0s)
Downloading packages ... (10.8s)
```

All OK. No need for Android development ATM

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/psv

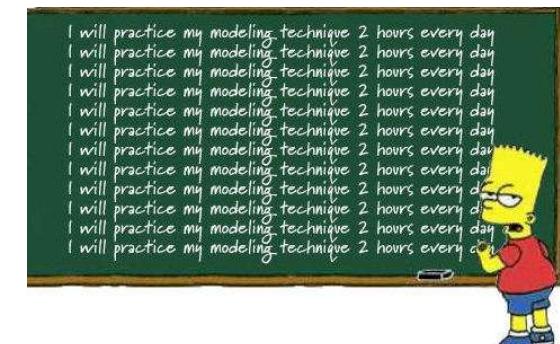
PS C:\Users\info> flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.29.0, on Microsoft Windows [Version 10.0.26100.0])
[✓] Windows Version (11 Pro 64-bit, 24H2, 2009)
[✗] Android toolchain - develop for Android devices
    X ANDROID_HOME = C:\Users\info\AppData\Local\Android\Sdk
        but Android SDK not found at this location.
[✓] Chrome - develop for the web
[✓] Visual Studio - develop Windows apps (Visual Studio Community 2022 17.13.0)
[!] Android Studio (not installed)
[✓] IntelliJ IDEA Community Edition (version 2024.3)
[✓] VS Code (version 1.94.2)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 2 categories.
PS C:\Users\info> |
```

Workshop



- Make sure **all prerequisites** are installed on your machine
- Install:
 - IntelliJ Flutter Plugin (via Marketplace)
 - Flutter SDK
 - Update Path (when applicable on Windows machines)
 - Run `flutter doctor`
- Fix any problems if they arise, help colleagues



Checkpoint



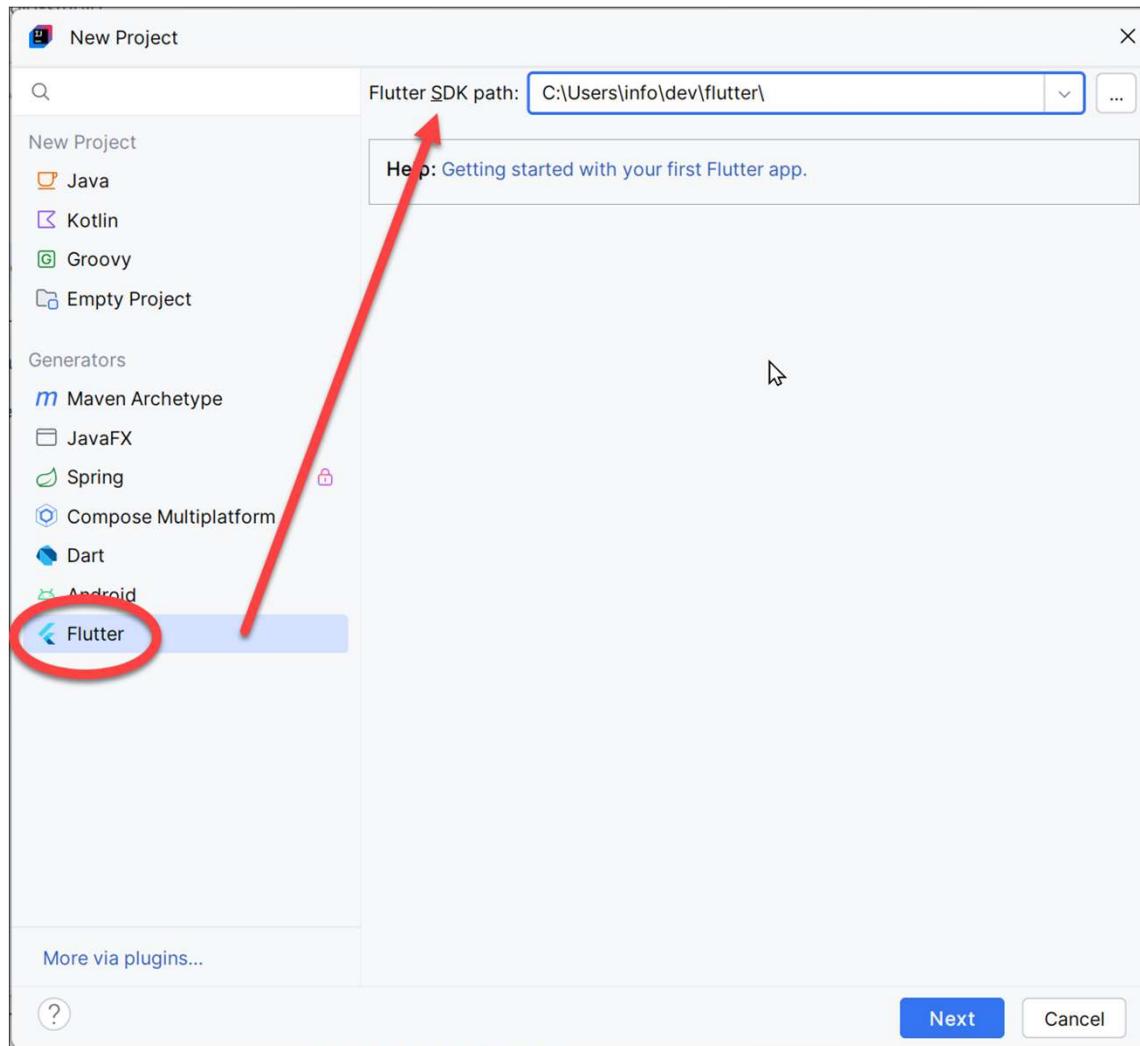
- You know what Flutter is and what it is used for
- You know the prerequisites for installing Flutter
- You can update the PATH (on Windows) to look for the Flutter SDK.
- You can describe the role of flutter doctor.



Your First App

Creating and running your first Flutter application

Creating your First Flutter project



IntelliJ New Project Dialog

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, preferably no numbers)
 - **SDK Path** – the location of your SDK (like `c:\users\<name>\dev\flutter`)
 - Project **location** – personal preference (`\Documents`, or `\Desktop`)
 - **Description** – a useful description of your app. No special chars!

Your first Flutter App



New Project

Project name: **first_flutter_app** (circled in red)

Project location: ~\Desktop\first_flutter_app

Description: Peter's sample first Flutter project.

Project type: Application

Organization: com.kassenaar

Android language: Java Kotlin

Platforms: Android iOS Linux MacOS Web Windows

When created, the new project will run on the selected platforms (others can be added later).

Help: Getting started with your first Flutter app.

Project type: 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.

Create project offline

> More Settings

?

Previous Create Cancel

Create a unique Organization name

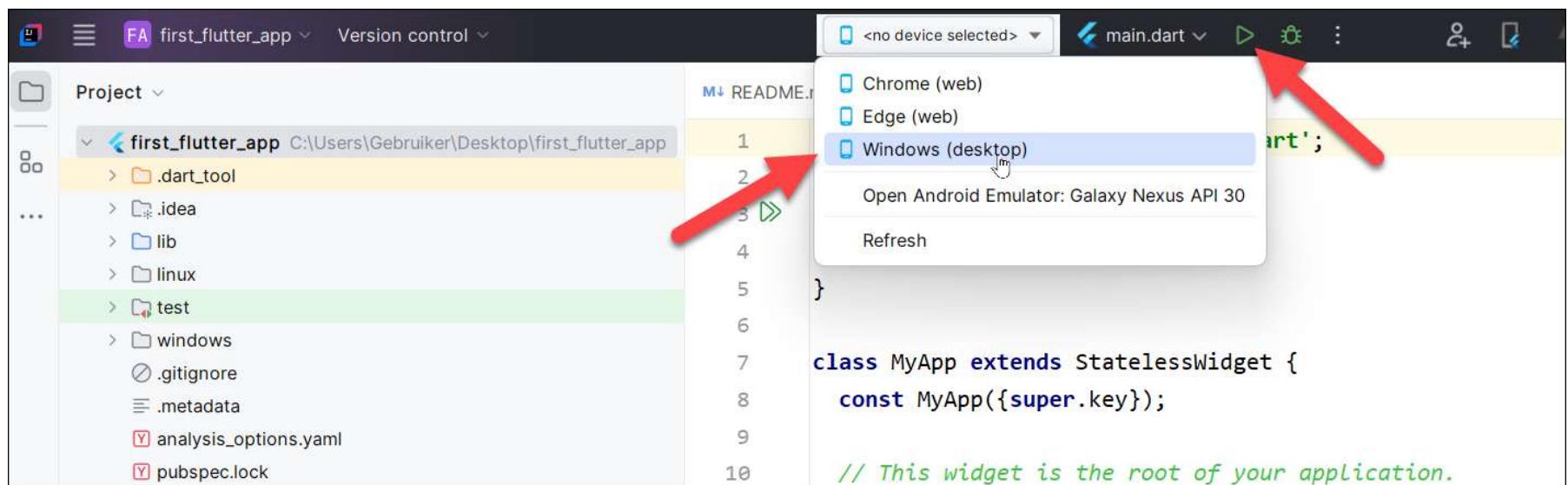
Typically: a domain name in reverse order.

Doesn't really matter at this point

Running the sample app



- Select an installed browser 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 the Android Studio IDE. On the left, the 'Run' tab is selected, showing the file 'main.dart'. The 'Console' tab is open, displaying the output of the build process:

```
Launching lib\main.dart on Windows in debug mode...
Building Windows application...
Built build\windows\x64\runner\Debug\first_flutter_app.exe
Debug service listening on ws://127.0.0.1:63069/5emRVUBNcVM=/ws
Syncing files to device Windows...
```

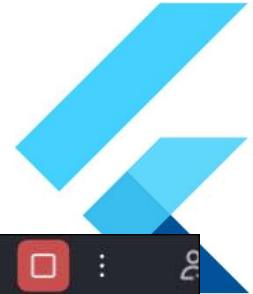
On the right, the 'first_flutter_app' window is displayed, showing the 'Flutter Demo Home Page'. The page displays the text 'You have pushed the button this many times:' followed by the number '4'. In the bottom right corner, there is a purple floating action button with a white plus sign (+).

Folder structure



- You – generally – **don't need** to manually adjust `\.idea`, `\build` and `\.dart_tool`
- 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 a development environment with a project structure on the left and a code editor on the right.

Project Structure: The left pane displays the file system structure of the project "first_flutter_app". A red box highlights the following files and folders:

- first_flutter_app (containing .dart_tool, .idea, build, lib/main.dart, linux, test, windows, .gitignore, .metadata, analysis_options.yaml, pubspec.lock, pubspec.yaml, README.md)
- External Libraries
- Scratches and Consoles

Code Editor: The right pane shows the content of the main.dart file.

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

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

class MyApp extends StatelessWidget {
  const MyApp({super.key});

  // This widget is the root of your application.
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Flutter Demo',
      theme: ThemeData(...), // ThemeData
      home: const MyHomePage(title: 'Flutter Demo'),
    );
}
}
```

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'),
    );
  }
}
...
```

(Yours might be slightly different,
due to updates)

Study the default code. It has useful comments.

What can we learn from the default code?

Architecture of all Flutter apps



- 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) {...}`
- StatefulWidget
 - Can have data or state
 - Consists of actually two classes
 - We'll look into that later!

So, when collapsed, the structure is like



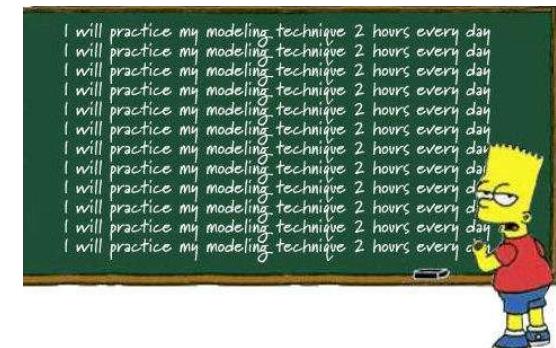
A screenshot of a code editor showing the `main.dart` file. The code defines a `MyApp` class that returns a `MaterialApp` widget with a `MyHomePage` as its home screen. A red arrow points from the `MyApp` class down to the `MyHomePage` class, which is highlighted with a red box. The code is as follows:

```
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
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
```

Workshop



- Create a new blank Starter App
- Start in the emulator or on your device
- Make yourself familiar with the folder- and file structure of the app.
- Replace some text inside main.dart (for instance the title and text on the floatingActionButton or on the page)
- Read the comments in the MyHomePage class and try to change some colors of the theme
- Code example: `../_100-default-app`



Checkpoint



- You can create new Flutter apps
- You know the default files and their function of apps
- You know `main.dart` is the startup file of the application
- You'll mostly work in `./lib` and subfolders
- You can make simple updates to the default provided application