

Chapter 2:

Flutter installation & setup

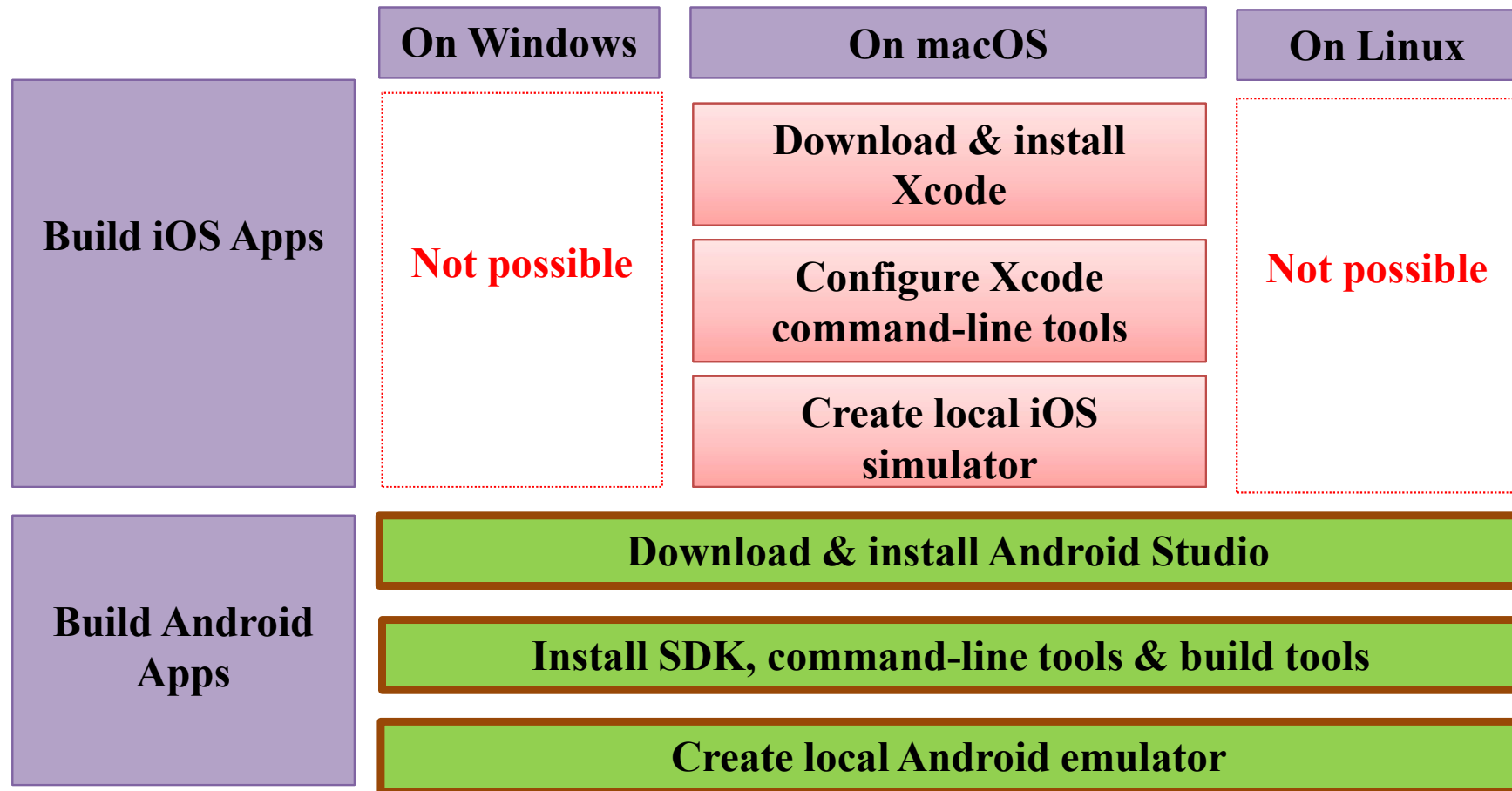
Plan

- 1. Installation overview.**
- 2. Installation on Windows.**
- 3. Installation on macOS.**
- 4. Use a real device (Android smartphone).**
- 5. Flutter project.**

Plan

- 1. Installation overview.**
2. Installation on Windows.
3. Installation on macOS.
4. Use a real device (Android smartphone).
5. Flutter project.

Flutter installation overview



- The code for all target platforms (iOS, Android, ...) can be **written** on **any** machine (Linux, Windows, MacOS), but to build **iOS** & **macOS apps** the code can **only** be tested and run **on macOS machines** (the same for Windows and Linux apps).
- On the other hand, Android and web apps can be built on **all** OS.

Flutter installation overview

- **IDE for Flutter:**
 - Visual Studio Code (VS Code).
 - Android Studio.
 - IntelliJ IDEA.
 - FlutLab.io (online IDE with limited free plan and premium paid plan) : <https://flutlab.io/pricing>.

Flutter installation overview

- **Installation steps:**

Step 0 : IDE + extensions

- a) **IDE** : for writing code.
- b) **Extensions** : add Flutter and Dart plugins to the IDE to simplify the development process.

Step 1 : Flutter SDK

- a) **Flutter SDK (UI Framework + collection of tools)** : for managing Flutter projects.
- b) **Git** : version control software, used internally by Flutter SDK.

Step 2 : Platform Tools

- a) **Android SDK** : used by Flutter SDK and needed for Android app deployment.
- b) **Xcode** : used by Flutter SDK and needed for iOS app deployment.

Step 3 : Virtual Devices

- a) **Android** : preview Flutter apps on Android Virtual Devices.
- b) **iOS** : preview Flutter apps on iOS simulators.

Plan

1. Installation overview.
- 2. Installation on Windows.**
3. Installation on macOS.
4. Use a real device (Android smartphone).
5. Flutter project.

Flutter installation on Windows

- Step 0.a. install *Android Studio* (*VS Code*).

Android setup

Note: Flutter relies on a full installation of Android Studio to supply its Android platform dependencies. However, you can write your Flutter apps in a number of editors; a later step discusses that.

The download
link

Install Android Studio

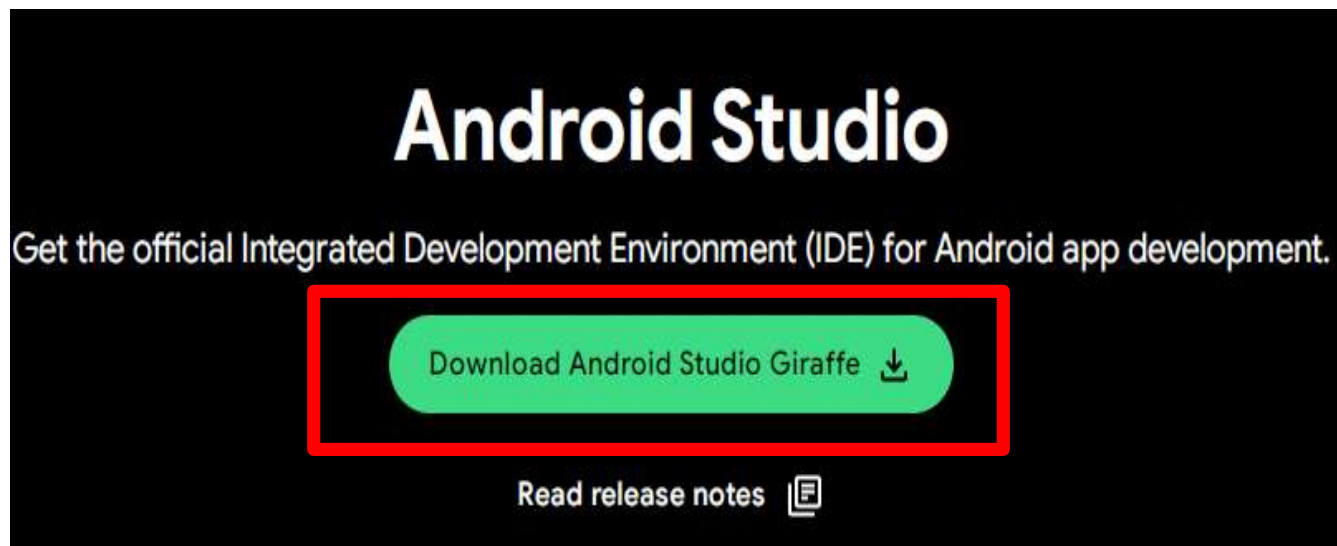
 Help

1. Download and install [Android Studio](#).
2. Start Android Studio, and go through the 'Android Studio Setup Wizard'. This installs the latest Android SDK, Android SDK Command-line Tools, and Android SDK Build-Tools, which are required by Flutter when developing for Android.
3. Run `flutter doctor` to confirm that Flutter has located your installation of Android Studio. If Flutter cannot locate it, run `flutter config --android-studio-dir=<directory>` to set the directory that Android Studio is installed to.

Flutter installation on Windows

- Step 0.a. install *Android Studio* (VS Code).

Step 0.a.1. Download *Android Studio* (*):



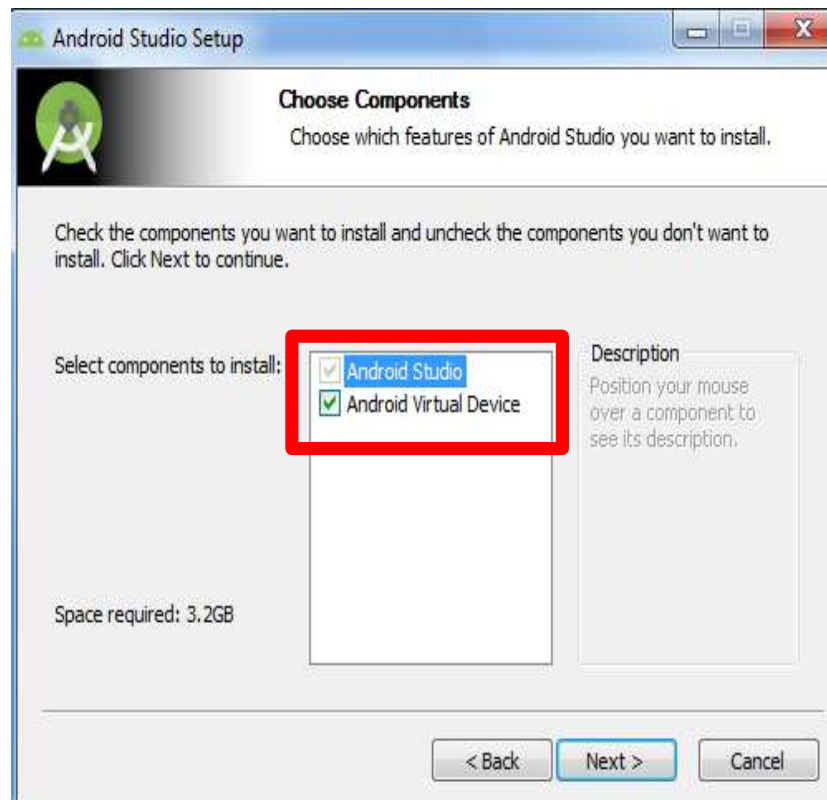
(*) : Google's official Android development environment.

Flutter installation on Windows

- Step 0.a. install *Android Studio (VS Code)*.

Step 0.a.

→ Step 0.a.2. Install Android Studio :



Flutter installation on Windows

- Step 0.a. install *Android Studio* (*VS Code*).

Step 0.a.3. Start and setup Android Studio.

The image displays three sequential screenshots of the Android Studio Setup Wizard, with red annotations highlighting key steps:

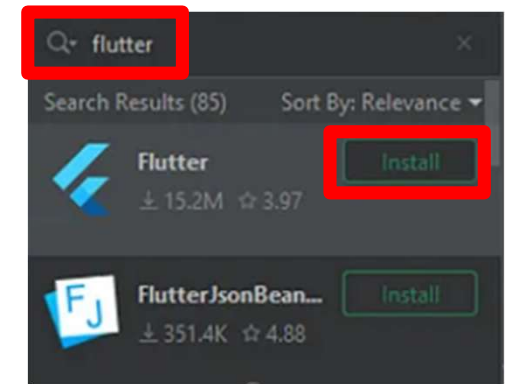
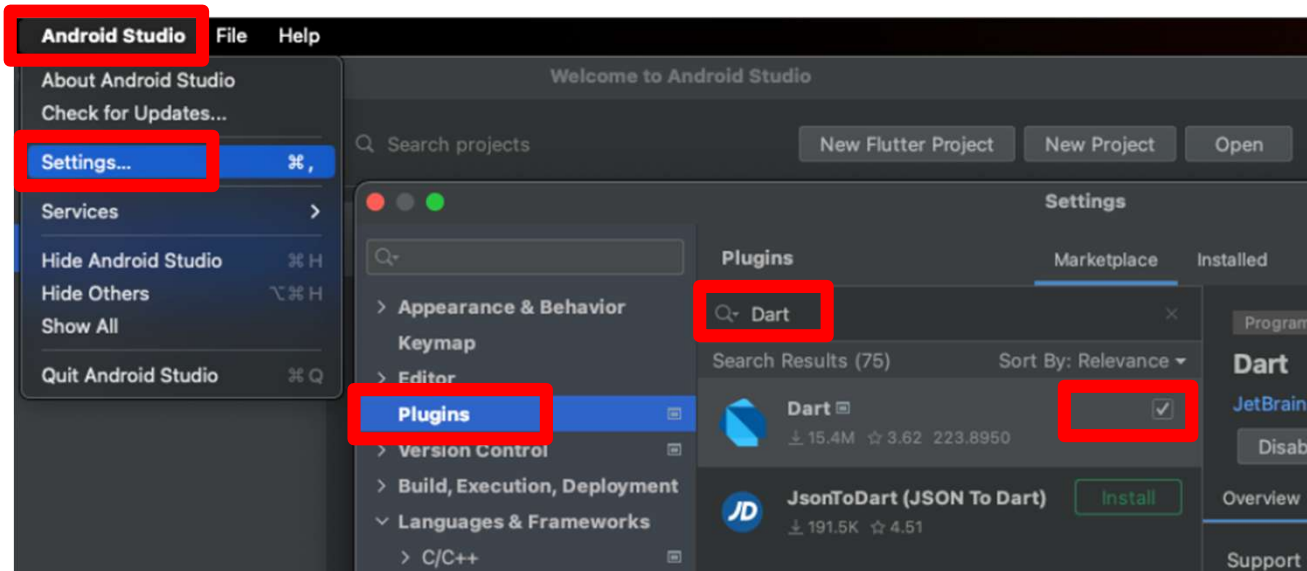
- Welcome Screen:** The first screenshot shows the 'Welcome' screen. A red arrow points from the 'Next' button to the 'Install Type' screen.
- Install Type Screen:** The second screenshot shows the 'Install Type' screen. A red box highlights the 'Custom' option, and a red arrow points from the 'Next' button to the 'SDK Components Setup' screen.
- SDK Components Setup Screen:** The third screenshot shows the 'SDK Components Setup' screen. A red box highlights the 'Android SDK' and 'Performance (Intel HAXM)' options, and a red arrow points from the 'Next' button to the 'SDK location' field.

Annotations:

- Select all components : SDK platform & Android Virtual Device (AVD).** (Red text box pointing to the 'Android SDK' and 'Performance (Intel HAXM)' options in the 'SDK Components Setup' screen)
- SDK location** (Red text box pointing to the 'Android SDK Location' field in the 'SDK Components Setup' screen)

Flutter installation on Windows

- Step 0.b. add *Flutter & Dart* plugins (extensions).

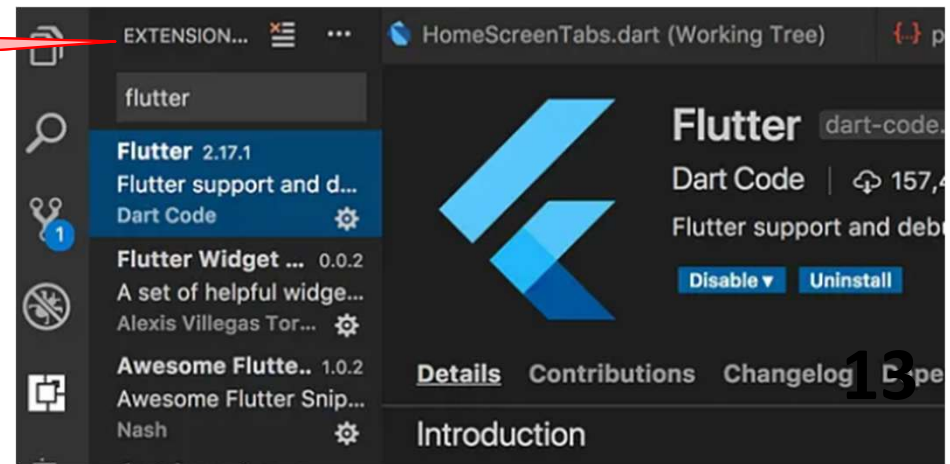


Add extensions to *Android Studio*

Flutter installation on Windows

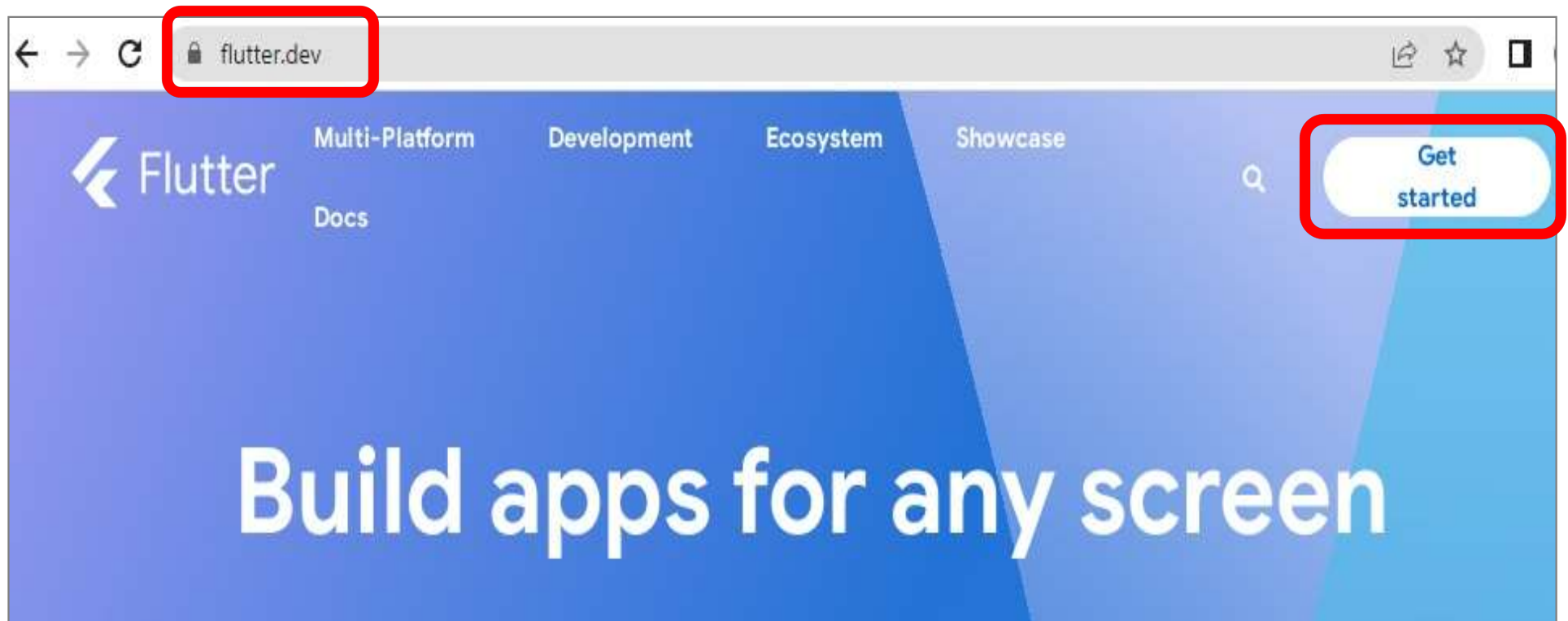
- Step 0.b. add *Flutter & Dart* plugins (extensions).

For VS Code, add the plugins from extension tab.



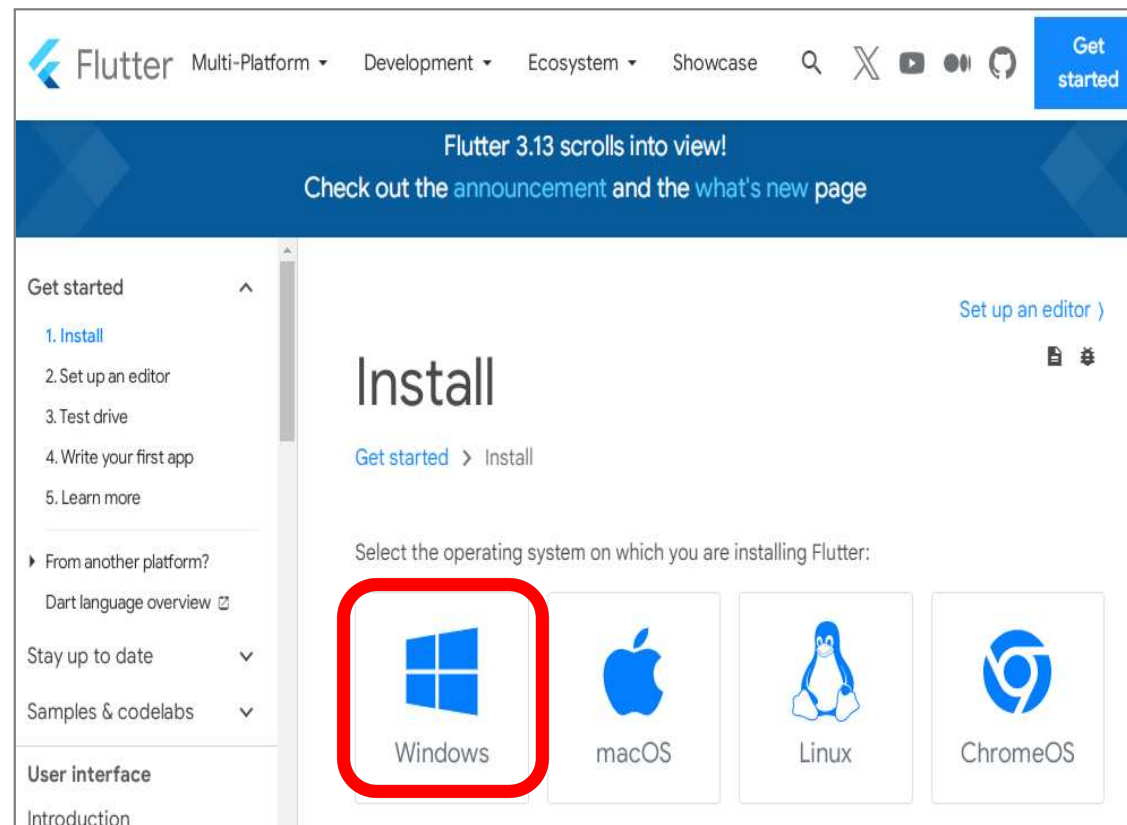
Flutter installation on Windows

- Step 1.a.1. Visit the official Flutter website *Flutter.dev* then click on *Get started* button :



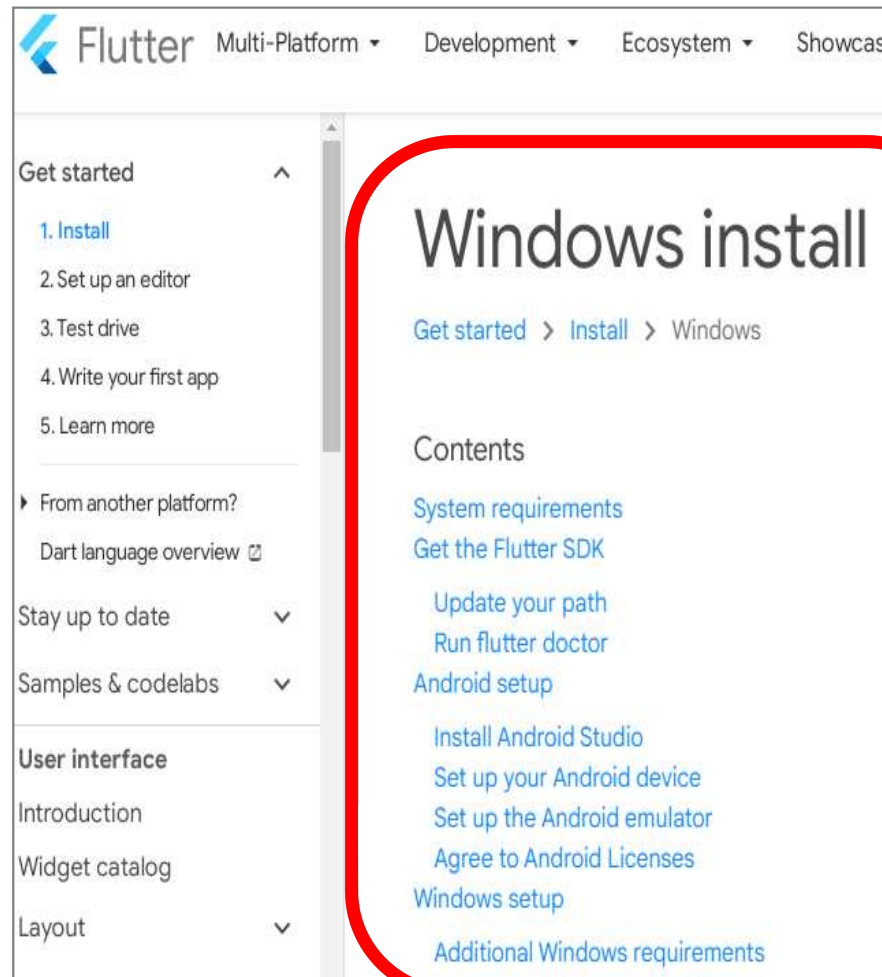
Flutter installation on Windows

- Step 1.a.2. Picking *Windows* as an option of the platform on which mobile apps will be developed :



Flutter installation on Windows

- Step 1.a.2. Picking *Windows* as an option of the platform on which mobile apps will be developed :



Flutter installation on Windows

- Step 1.a.3. System requirements: installing *Git* with the option "use Git from the Windows Command Prompt" :

System requirements

To install and run Flutter, your development environment must meet these minimum requirements:

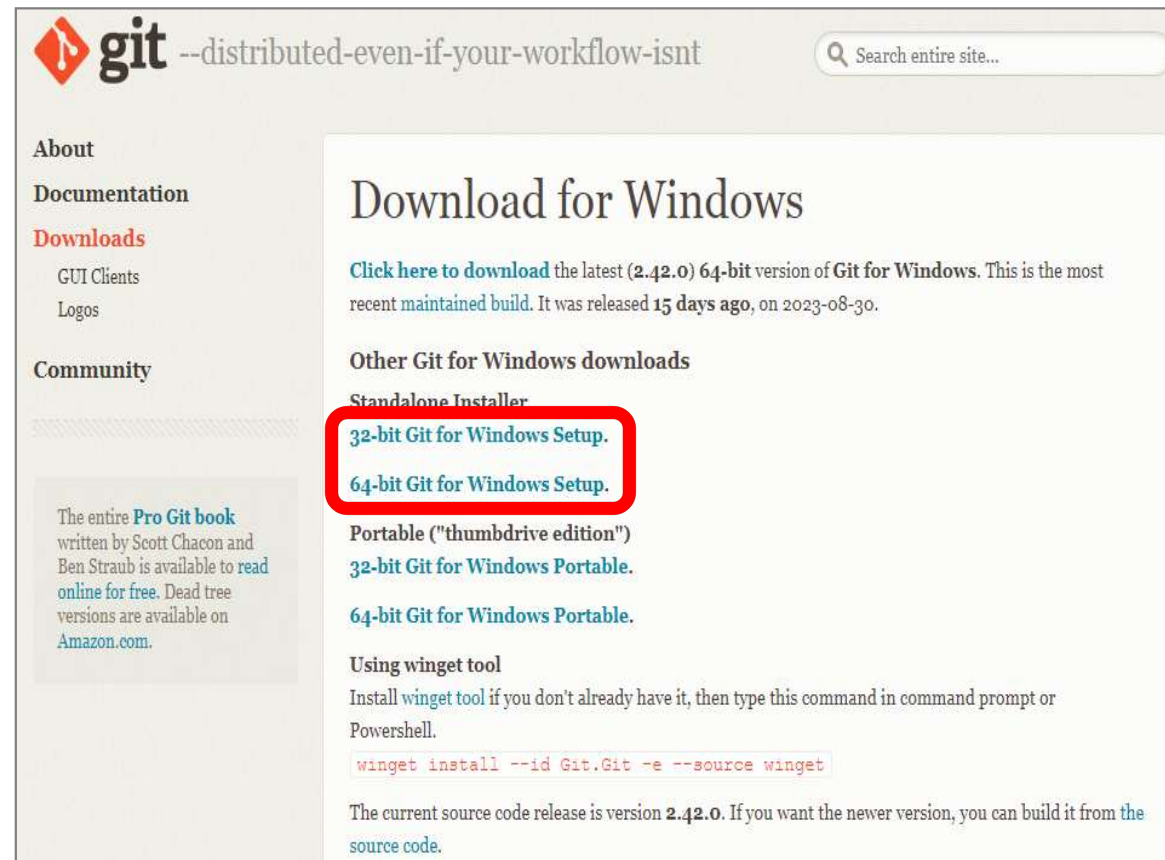
- **Operating Systems:** Windows 10 or later (64-bit), x86-64 based.
- **Disk Space:** 1.64 GB (does not include disk space for IDE/tools).
- **Tools:** Flutter depends on these tools being available in your environment.
 - [Windows PowerShell 5.0](#) or newer (this is pre-installed with Windows 10)
 - [Git for Windows 2.x](#), with the **Use Git from the Windows Command Prompt** option.

If Git for Windows is already installed, make sure you can run `git` commands from the command prompt or PowerShell.

Follow the
link to install
Git for
Windows

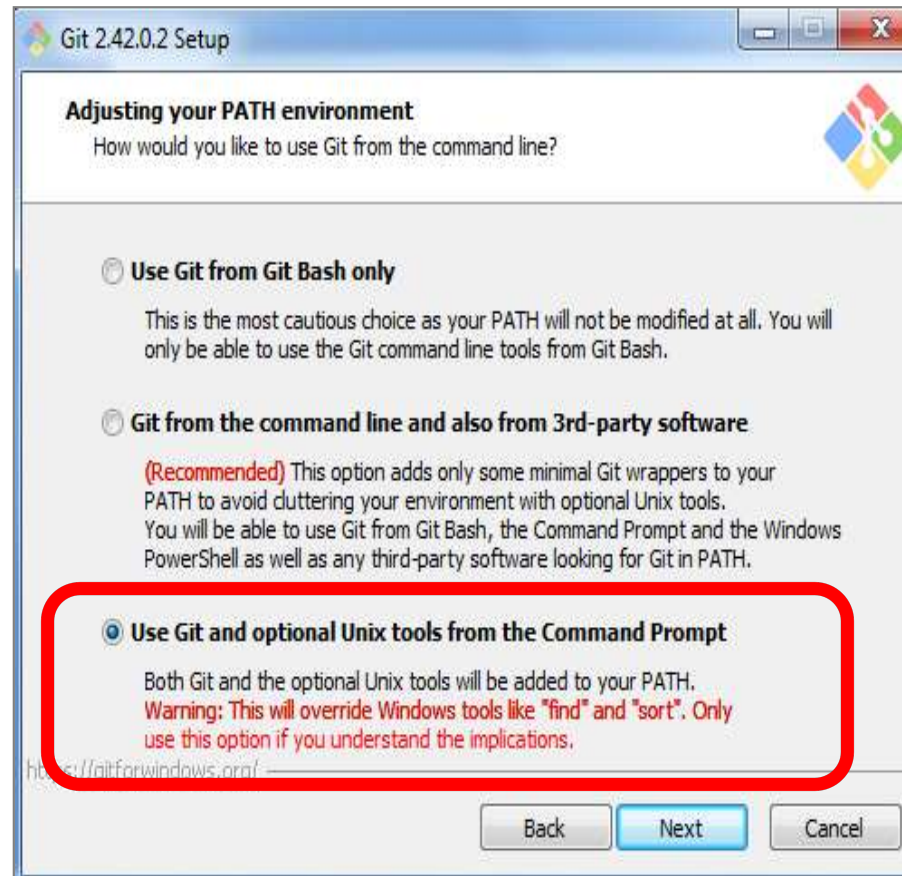
Flutter installation on Windows

- Step 1.a.3. System requirements: installing *Git* with the option "use Git from the Windows Command Prompt" :



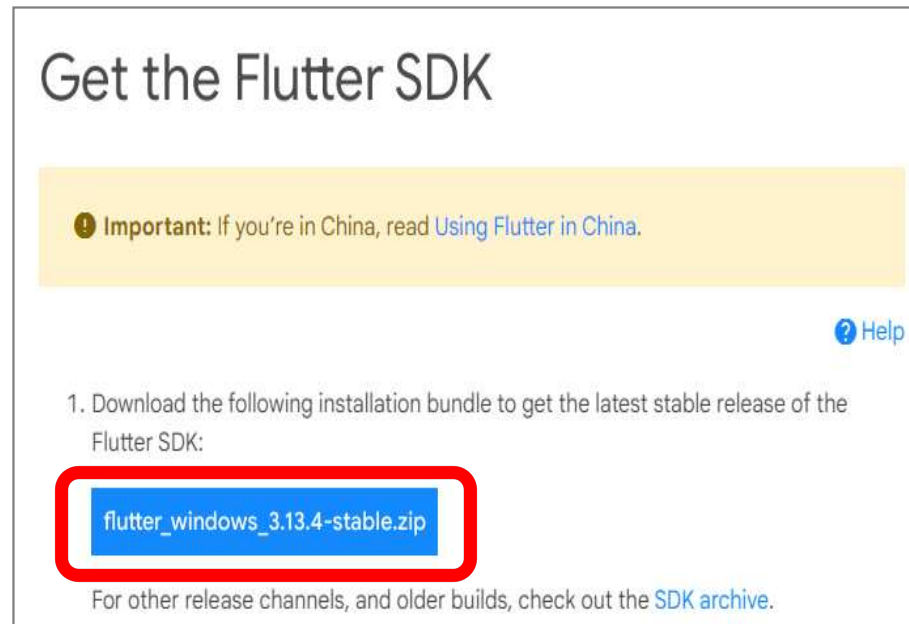
Flutter installation on Windows

- Step 1.a.3. System requirements: installing *Git* with the option "use Git from the Windows Command Prompt" :



Flutter installation on Windows

- Step 1.b.1. Installing *Flutter* itself (from the Flutter website):



- Download the zip file and unzip it.**
- Move the flutter file somewhere else:**
e.g. to **C:\src\flutter** folder.

Flutter installation on Windows

- Step 1.b.2. **Update the path** : set environment variable.

Update your path

[? Help](#)

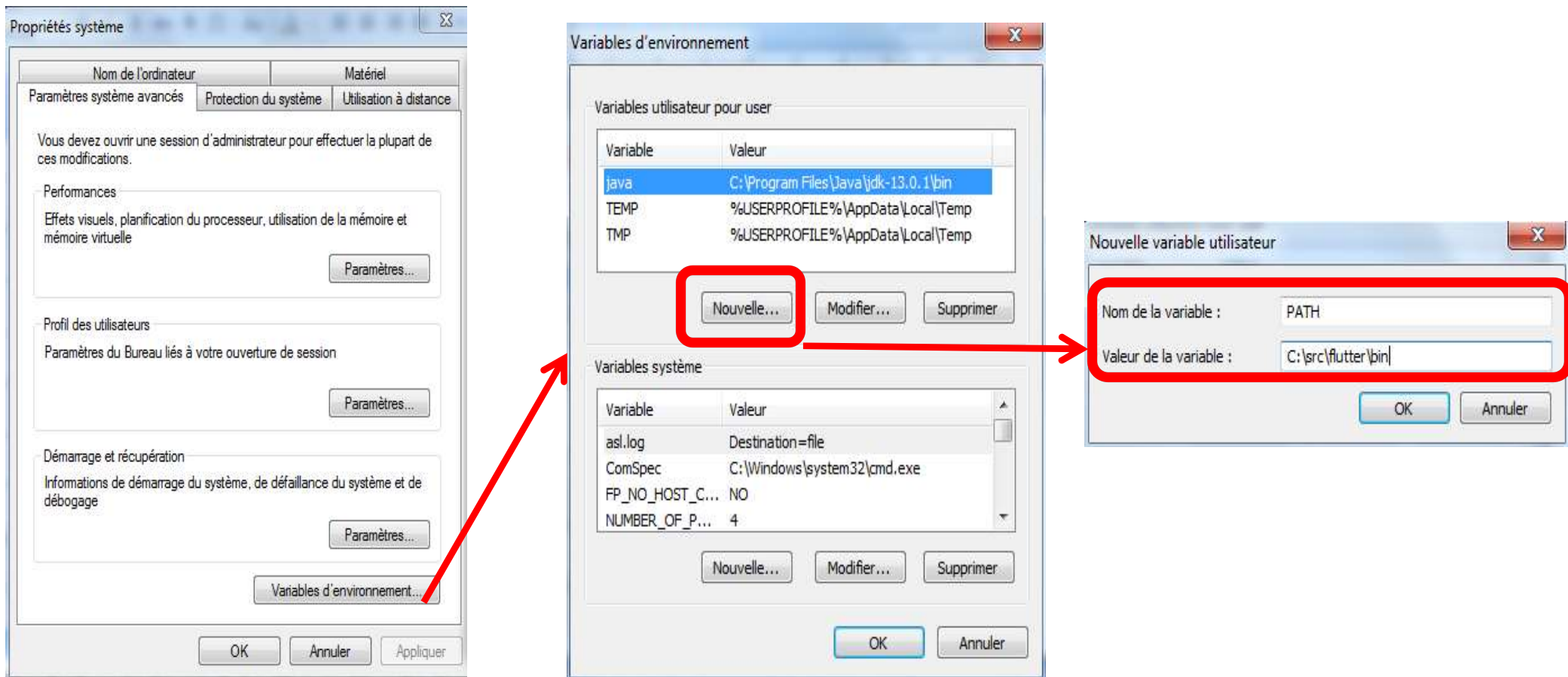
If you wish to run Flutter commands in the regular Windows console, take these steps to add Flutter to the `PATH` environment variable:

- From the Start search bar, enter 'env' and select **Edit environment variables for your account**.
- Under **User variables** check if there is an entry called **Path**:
 - If the entry exists, append the full path to `flutter\bin` using `;` as a separator from existing values.
 - If the entry doesn't exist, create a new user variable named `Path` with the full path to `flutter\bin` as its value.

You have to close and reopen any existing console windows for these changes to take effect.

Flutter installation on Windows

- Step 1.b.2. Update the path : set environment variable.



Look for "system environment variables" from the research bar of windows system.

Flutter installation on Windows

- Step 1.b.3. Check platform dependencies to complete the setup : run *flutter doctor* command.

Run flutter doctor

[? Help](#)

From a console window that has the Flutter directory in the path (see above), run the following command to see if there are any platform dependencies you need to complete the setup:

```
C:\src\flutter>flutter doctor
```

This command checks your environment and displays a report of the status of your Flutter installation. Check the output carefully for other software you might need to install or further tasks to perform (shown in **bold text**).

```
C:\Users\V
```

```
>flutter doctor
```

Welcome to Flutter! - <https://flutter.dev>

The Flutter tool uses Google Analytics to anonymously report feature usage statistics and basic crash reports. This data is used to help improve Flutter tools over time.

Flutter tool analytics are not sent on the very first run. To disable reporting, type 'flutter config --no-analytics'. To display the current setting, type 'flutter config'. If you opt out of analytics, an opt-out event will be sent, and then no further information will be sent by the Flutter tool.

By downloading the Flutter SDK, you agree to the Google Terms of Service. Note: The Google Privacy Policy describes how data is handled in this service.

Moreover, Flutter includes the Dart SDK, which may send usage metrics and crash reports to Google.

Read about data we send with crash reports:
<https://flutter.dev/docs/reference/crash-reporting>

See Google's privacy policy:
<https://policies.google.com/privacy>

Flutter installation on Windows

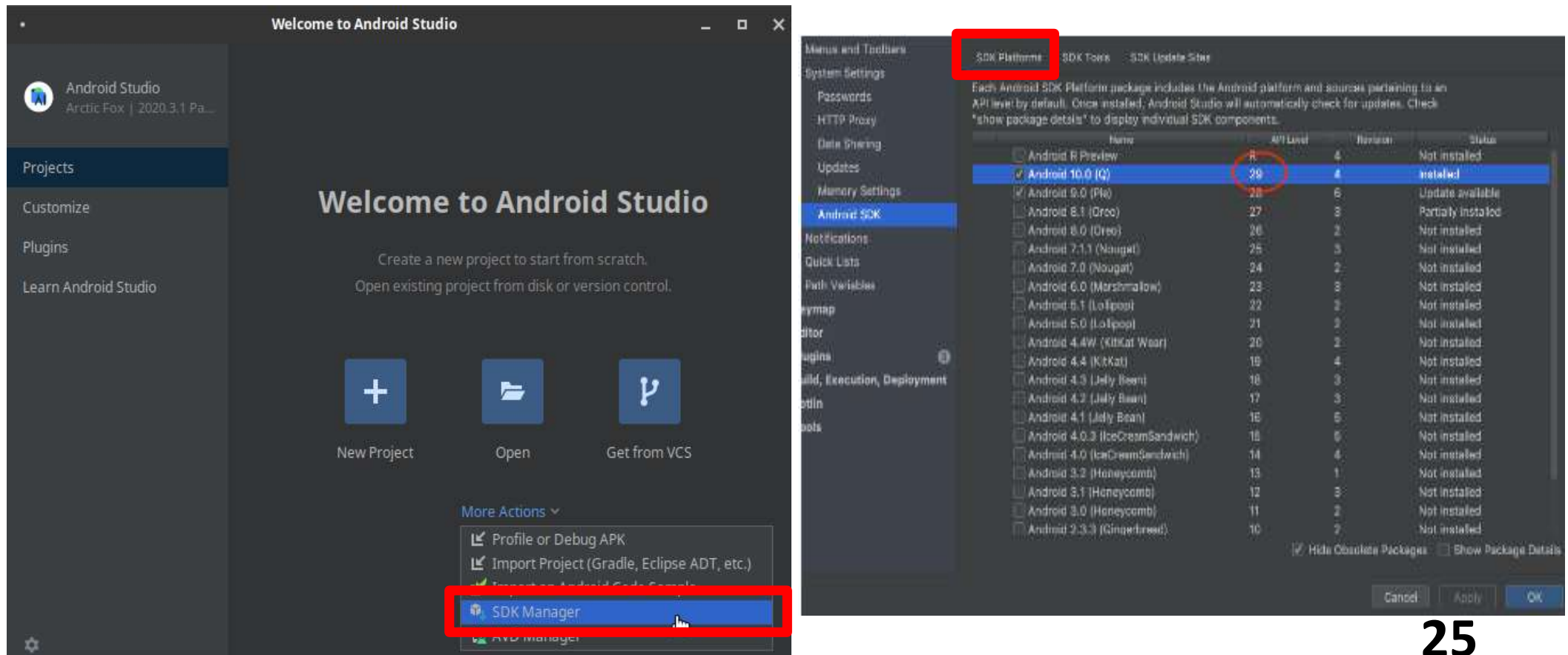
- Step 1.b.3. Check platform dependencies to complete the setup : run *flutter doctor* command.

```
test_api 0.4.16 (0.4.18 available)
test_core 0.4.20 (0.4.24 available)
vm_service 9.4.0 (11.0.1 available)
web_socket_channel 2.2.0 (2.3.0 available)
webdriver 3.0.1 (3.0.2 available)
Got dependencies in ...\.src\flutter\packages\flutter_tools\
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.7.3, on Microsoft Windows [Version 10.0.22621.1105], locale en-US)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[✗] Android toolchain - develop for Android devices
    ✗ Unable to locate Android SDK.
      Install Android Studio from: https://developer.android.com/studio/index.html
      On first launch it will assist you in installing the Android SDK components.
      (or visit https://flutter.dev/docs/get-started/install/windows#android-setup for detailed instructions).
      If the Android SDK has been installed to a custom location, please use
      `flutter config --android-sdk` to update to that location.
[✗] Chrome - develop for the web (Cannot find Chrome executable at .\Google\Chrome\Application\chrome.exe)
    ! Cannot find Chrome. Try setting CHROME_EXECUTABLE to a Chrome executable.
[✗] Visual Studio - develop for Windows
    ✗ Visual Studio not installed; this is necessary for Windows development.
      Download at https://visualstudio.microsoft.com/downloads/.
      Please install the "Desktop development with C++" workload, including all of its default components
[!] Android Studio (not installed)
[✓] Connected device (2 available)
[✓] HTTP Host Availability

Doctor found issues in 4 categories.
```

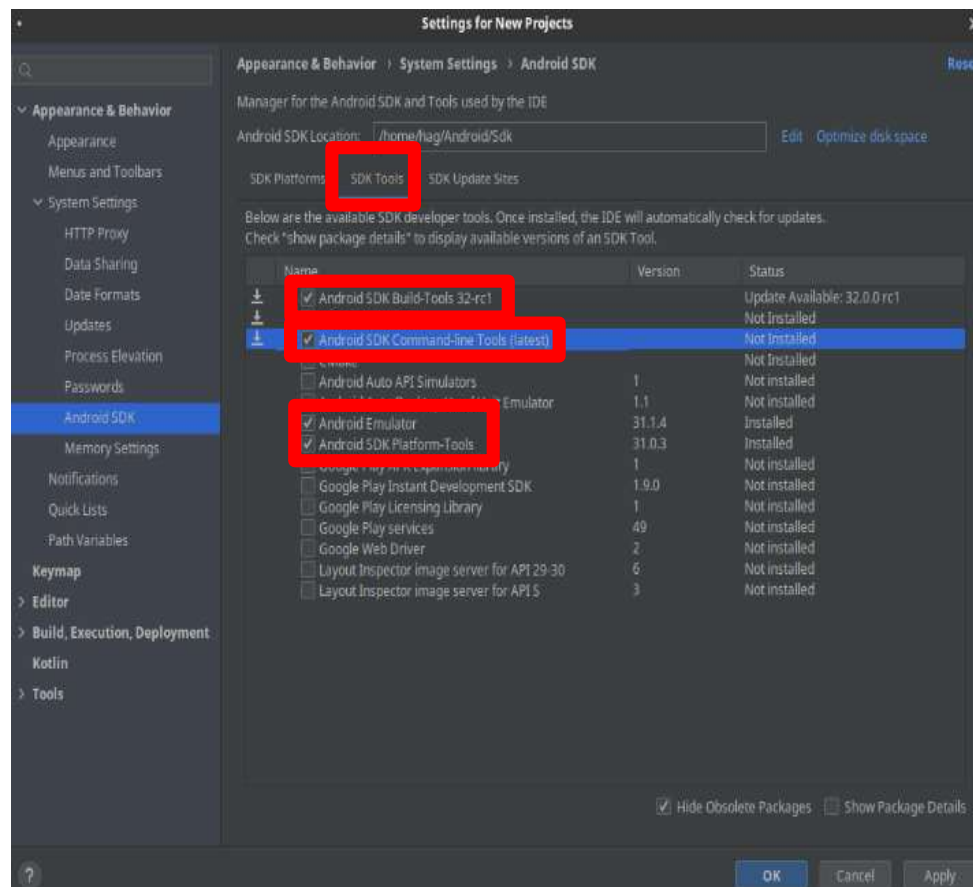
Flutter installation on Windows

- Step 2.a. Prepare the system for Android app development :
Setup the *SDK Manager* from Android Studio.



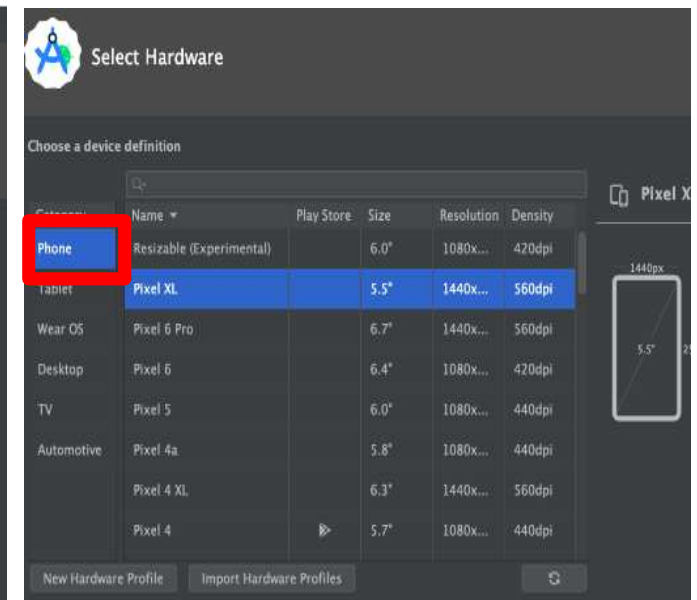
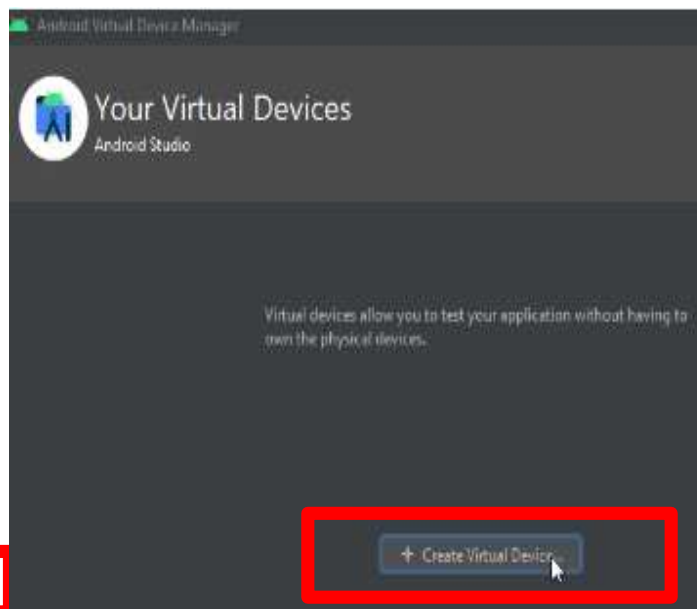
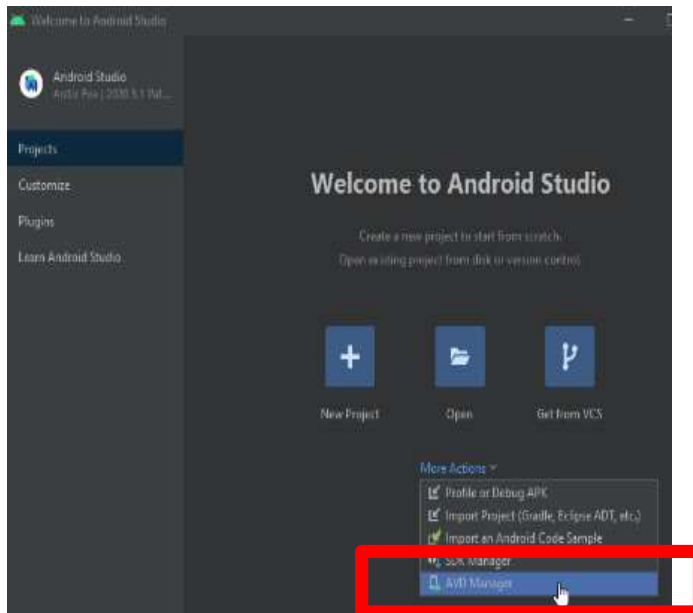
Flutter installation on Windows

- Step 2.a. Prepare the system for Android app development :
Setup *SDK Manager* from Android Studio.



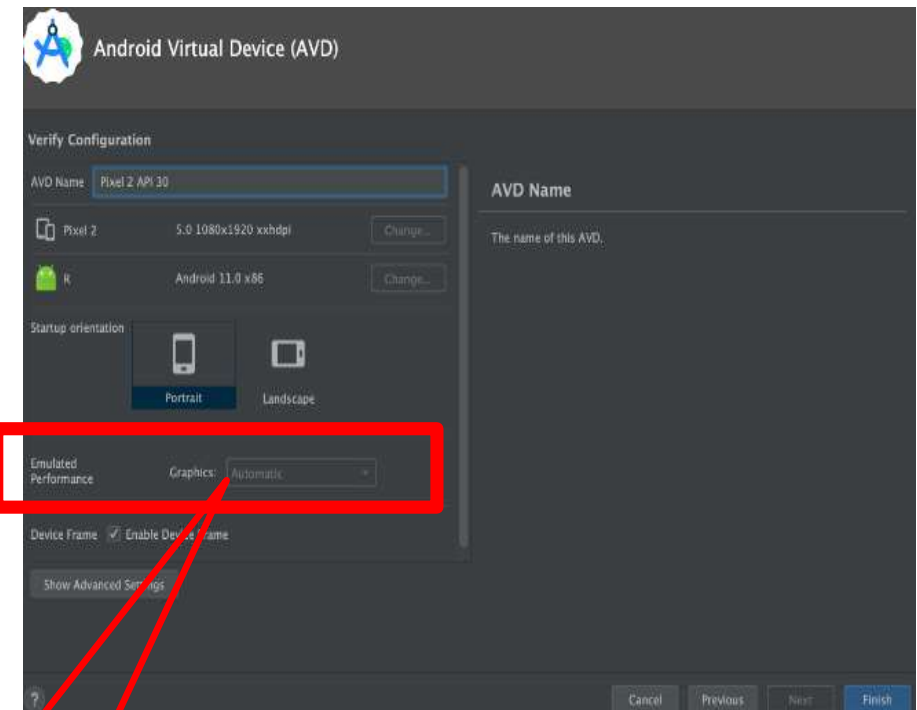
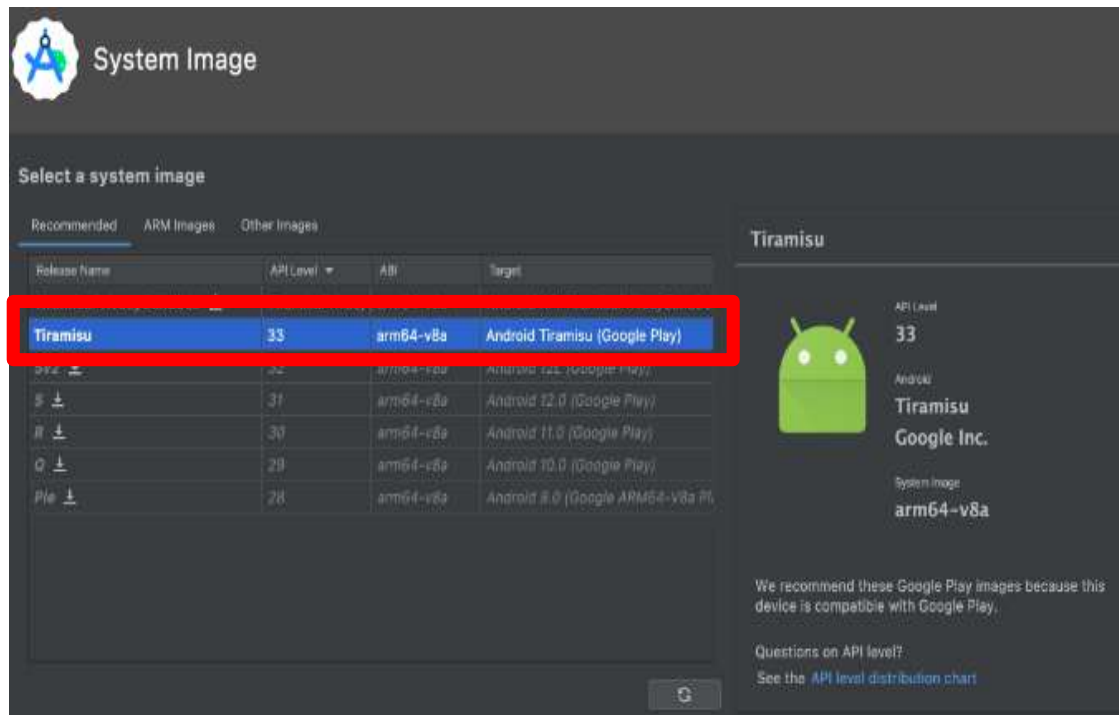
Flutter installation on Windows

- Step 3.a. Setup the *Virtual Device Manager* to create an emulator.



Flutter installation on Windows

- Step 3.a. Setup the *Virtual Device Manager* to create an emulator.



Hardware

Flutter installation on Windows

- Step 3.a. Setup the *Virtual Device Manager* to create an emulator.



- Remark: is it also possible to launch the emulator from VS Code.

Flutter installation on Windows

- For more details on
 - IDE installation of Android Studio and Visual Studio Code (including Flutter & Dart extensions) and,
 - Running Flutter project on Android emulator (AVD).
- Please get a look at the videos :
 - DemoFlutterInstallation.mp4 (*)
 - DemoFlutterInstallation2.mp4 (*)

(*) Read videos with VLC media player.



Plan

1. Installation overview.
2. Installation on Windows.
- 3. Installation on macOS.**
4. Use a real device (Android smartphone).
5. Flutter project.

Flutter installation on macOS

- **Installing Flutter on macOS:** for Flutter installation process on macOS system and running Flutter project on iOS simulator, please get a look at the video :

➤ DemoFlutterInstallation4macOS.mp4 (*)

(*) Read videos with VLC media player.



Plan

1. Installation overview.
2. Installation on Windows.
3. Installation on macOS.
- 4. Use a real device (Android smartphone).**
5. Flutter project.

Use a real device

- Running a Flutter project on an *Android smartphone*:
 - 1) Activate developer options by clicking 7 times on the build number from « settings/About phone ... »
 - 2) Activate USB debugging option on the phone from « settings/developer options ».
 - 3) Connect physical device using USB cable.
 - 4) Launch the app from the IDE (Android Studio, VS code) .

Plan

1. Installation overview.
2. Installation on Windows.
3. Installation on macOS.
4. Use a real device (Android smartphone).
- 5. Flutter project.**

Flutter project

- **Creation:** four options exist for creating a new Flutter project.
 1. From the windows prompt command:



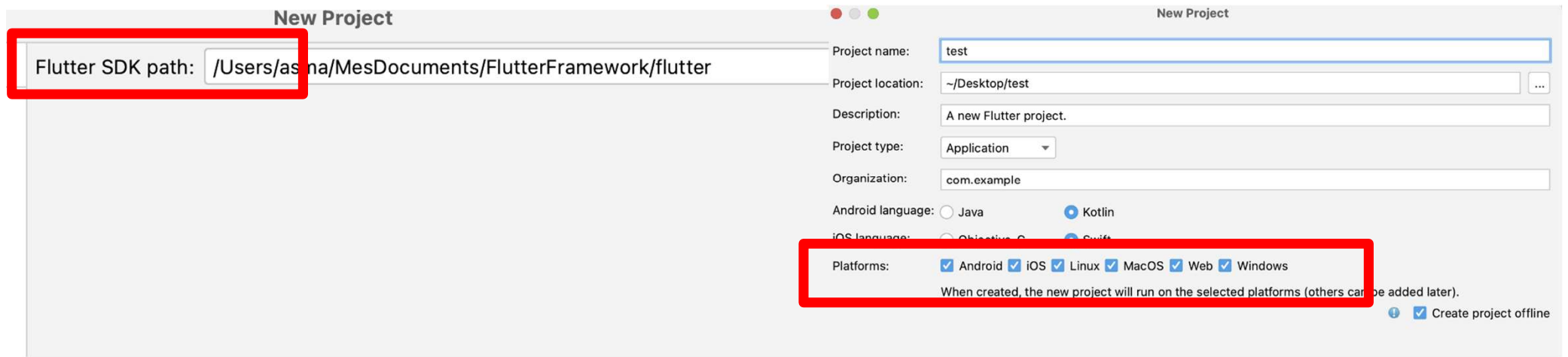
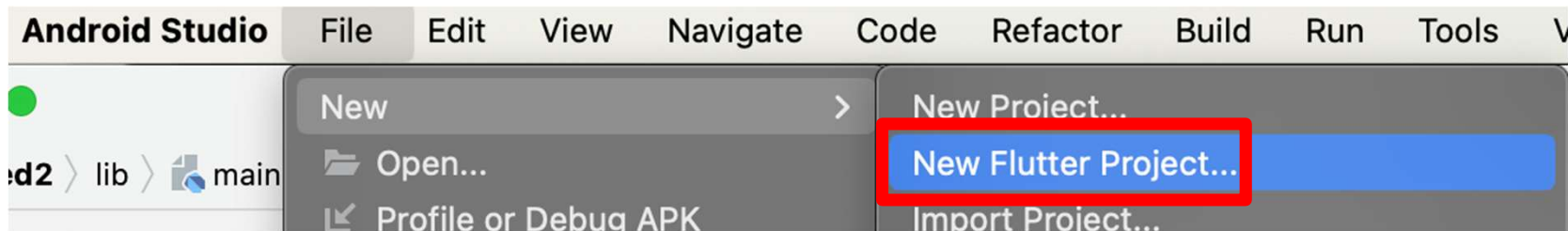
```
Administrateur : Invite de commandes
C:\Users\user\Desktop\Flutter> flutter create firstProject
```

**! Flutter
Project name
must not contain
capital letters.**

Flutter project

- **Creation:** four options exist for creating a new Flutter project.

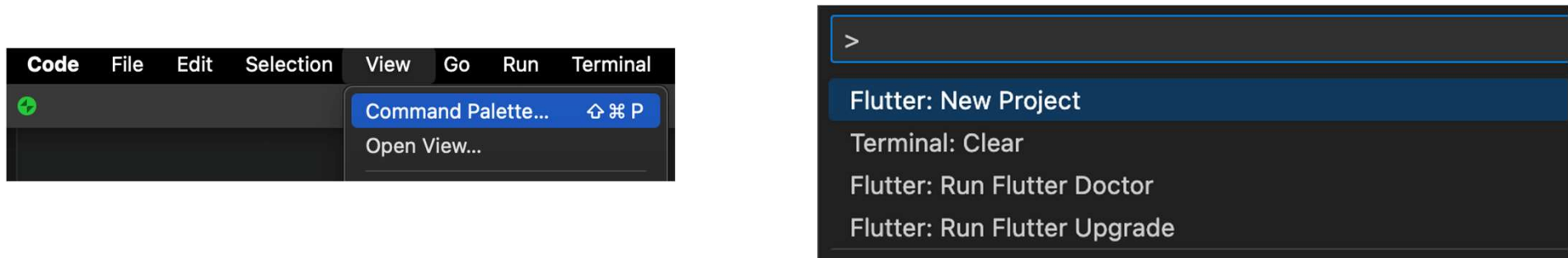
2. From Android Studio:



Flutter project

- **Creation:** four options exist for creating a new Flutter project.

3. From the VS Code command palette:



4. From the VS Code terminal:

```
MacBook-Pro-de-Asma:~ asma$ flutter create myfirst_project
Creating project myfirst_project...
Resolving dependencies in myfirst_project... (11.9s)
Got dependencies in myfirst_project.
Wrote 129 files.

All done!
You can find general documentation for Flutter at: https://docs.flutter.dev/
Detailed API documentation is available at: https://api.flutter.dev/
If you prefer video documentation, consider: https://www.youtube.com/c/flutterdev

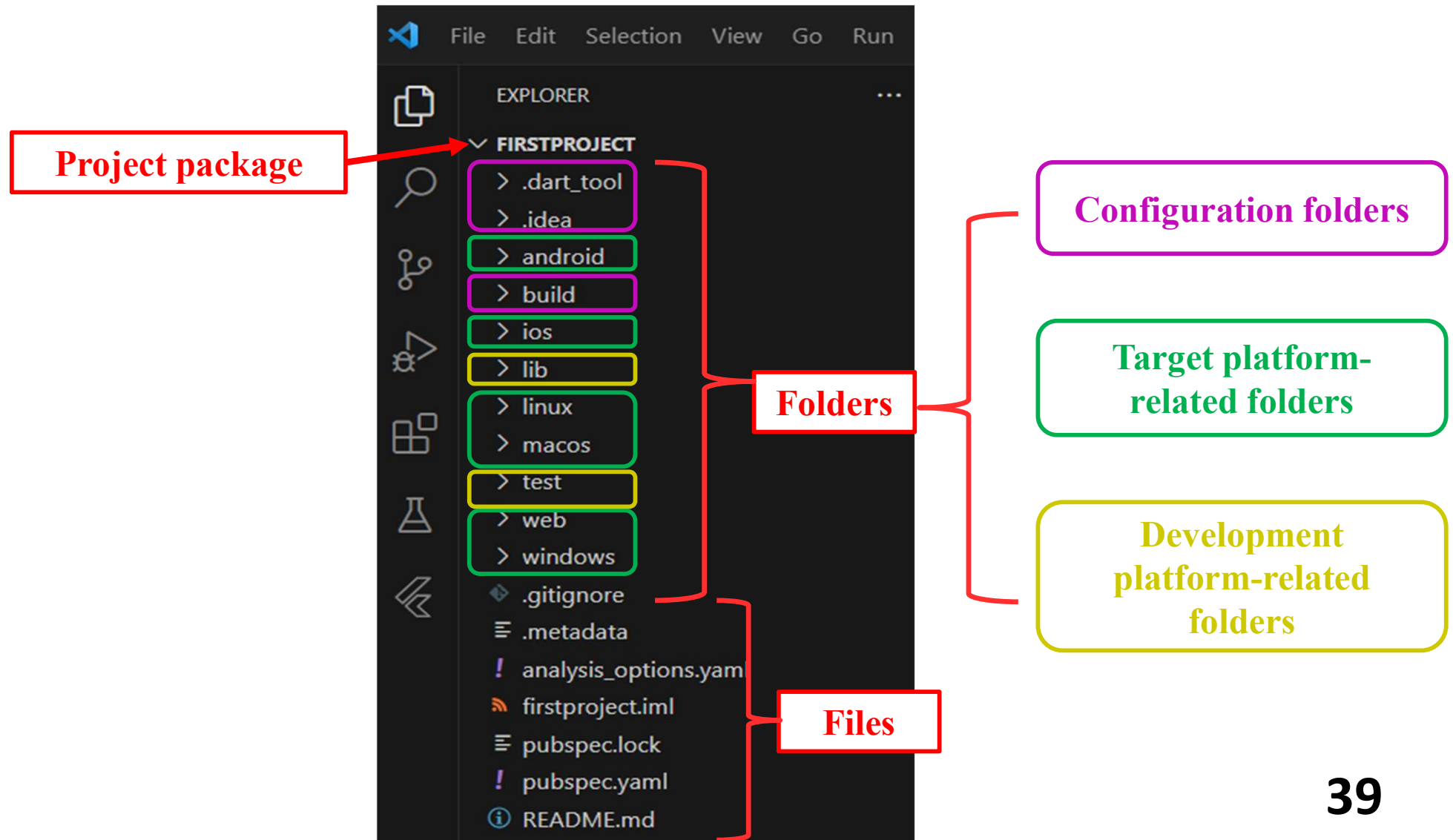
In order to run your application, type:

$ cd myfirst_project
$ flutter run

Your application code is in myfirst_project/lib/main.dart.
```

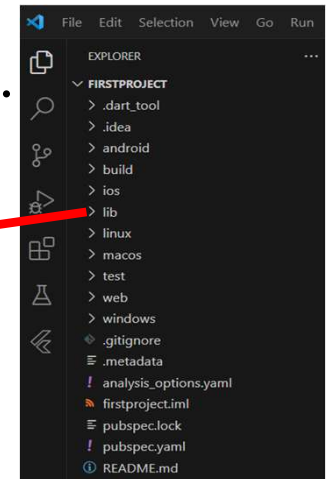
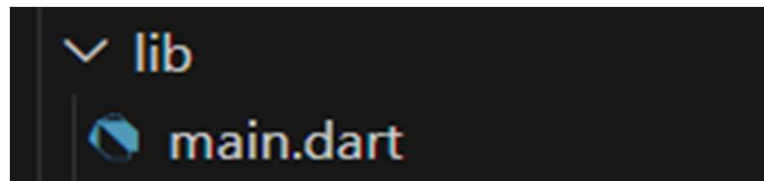
Flutter project

- Structure:



Flutter project

- **Structure: "lib folder"**
 - Contains Flutter app code (files end with .dart).
 - Contains **main.dart** (the entry point of the Flutter app).

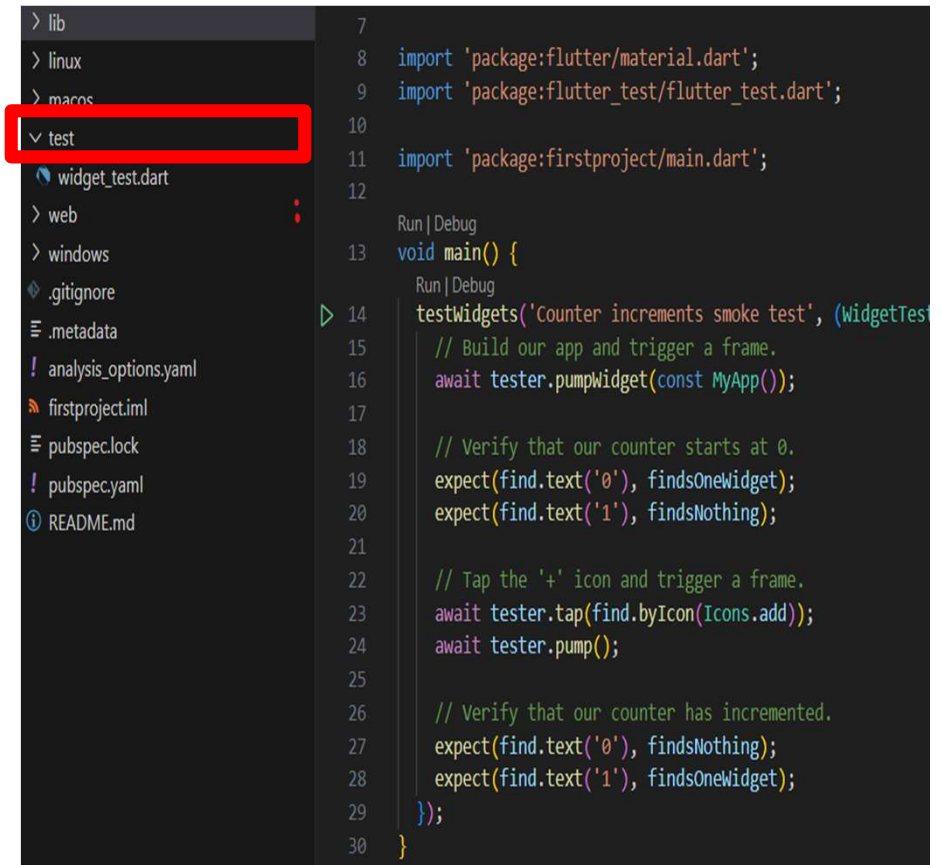


- **Example:** basic Flutter app (Hello World!)

```
import 'package:flutter/widgets.dart'; // basic set of widgets
// When Dart is running the application, it calls to the main() function
main() => runApp( // The function runApp() starts the Flutter application
  Text( // this is a widget, it renders the given text
    'Hello, World!!!', // the first argument is a text that needs to be rendered
    textDirection: TextDirection.ltr, // here we set the direction "left to right"
  ), );
```

Flutter project

- **Structure: "test folder"**
 - Contains Dart code files used to test the application code.
 - Test files are used when "flutter –test" command is run.
 - Test files should have the suffix "_test.dart".



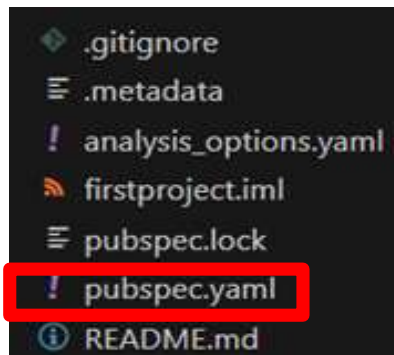
The screenshot shows an IDE with a file explorer on the left and a code editor on the right. In the file explorer, the 'test' folder is highlighted with a red rectangle. The code editor displays the contents of a test file, which includes imports for Flutter material widgets, Flutter test utilities, and the application's main.dart file. The main function contains a series of widget tests: a smoke test, a test for the initial counter value, and a test for the counter incrementing after a tap.

```
7
8 import 'package:flutter/material.dart';
9 import 'package:flutter_test/flutter_test.dart';
10
11 import 'package:firstproject/main.dart';
12
13 Run | Debug
14 void main() {
15   Run | Debug
16   testWidgets('Counter increments smoke test', (WidgetTester tester) async {
17     // Build our app and trigger a frame.
18     await tester.pumpWidget(const MyApp());
19
20     // Verify that our counter starts at 0.
21     expect(find.text('0'), findsOneWidget);
22     expect(find.text('1'), findsNothing);
23
24     // Tap the '+' icon and trigger a frame.
25     await tester.tap(find.byIcon(Icons.add));
26     await tester.pump();
27
28     // Verify that our counter has incremented.
29     expect(find.text('0'), findsNothing);
30     expect(find.text('1'), findsOneWidget);
31   });
32 }
```

Flutter project

- **Structure: "pubspec.yaml file"**

- Used to add third party packages to the project (e.g. include images into project).
- Specifies project descriptions, constraints, dependencies, version, assets, ...



Add custom fonts

```
firstproject > ! pubspec.yaml
1  name: firstproject
2  description: A new Flutter project.
3  publish_to: 'none'
4  version: 1.0.0+1
5  environment:
6    sdk: '>=3.1.3 <4.0.0'
7  dependencies:
8    flutter:
9      sdk: flutter
10   cupertino_icons: ^1.0.2
11
12  dev_dependencies:
13    flutter_test:
14      sdk: flutter
15    flutter_lints: ^2.0.0
16
17  flutter:
18    uses-material-design: true
19    assets:
20      - images/a_dot_burr.jpeg
21    fonts:
22      - family: Schyler
23        fonts:
24          - asset: fonts/Schyler-Regular.ttf
25          - asset: fonts/Schyler-Italic.ttf
26            style: italic
27      - family: Trajan Pro
28        fonts:
29          - asset: fonts/TrajanPro.ttf
30          - asset: fonts/TrajanPro_Bold.ttf
31            weight: 700
```

Flutter project

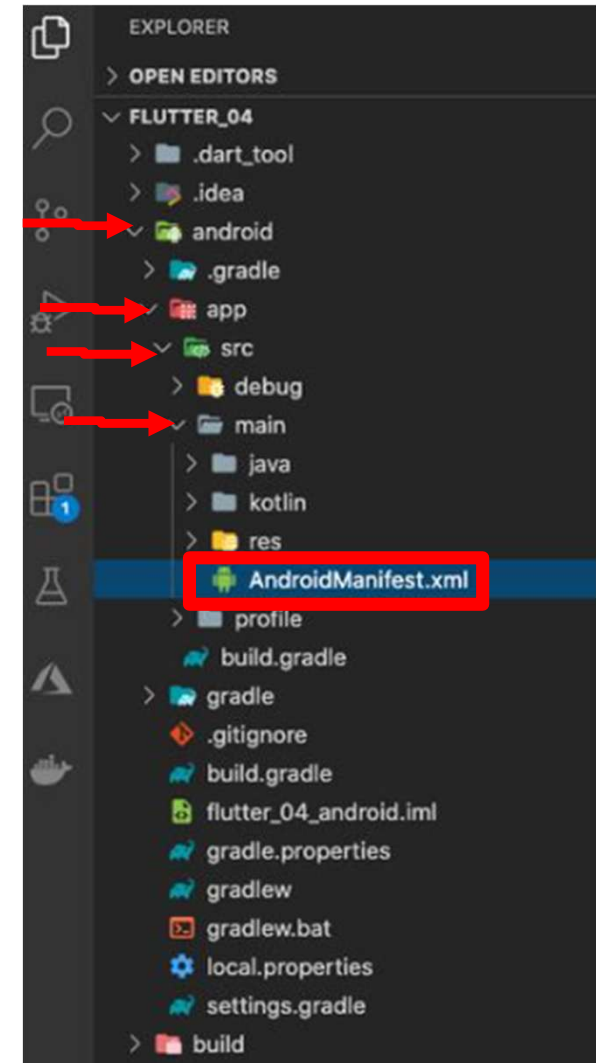
- **Structure: "AndroidManifest.xml file"**

- Describes the essential information about the app to the Android build tools, Android OS, and Google Play.
- It is used to configure the Android-specific settings of the app such as:
 - **Permissions** the app requires (camera, internet,...).

```
<uses-permission android:name="android.permission.INTERNET" />  
<uses-permission android:name="android.permission.CAMERA" />
```

- Declare **hardware** or **software** features the app requires (Bluetooth, GPS, camera,...).

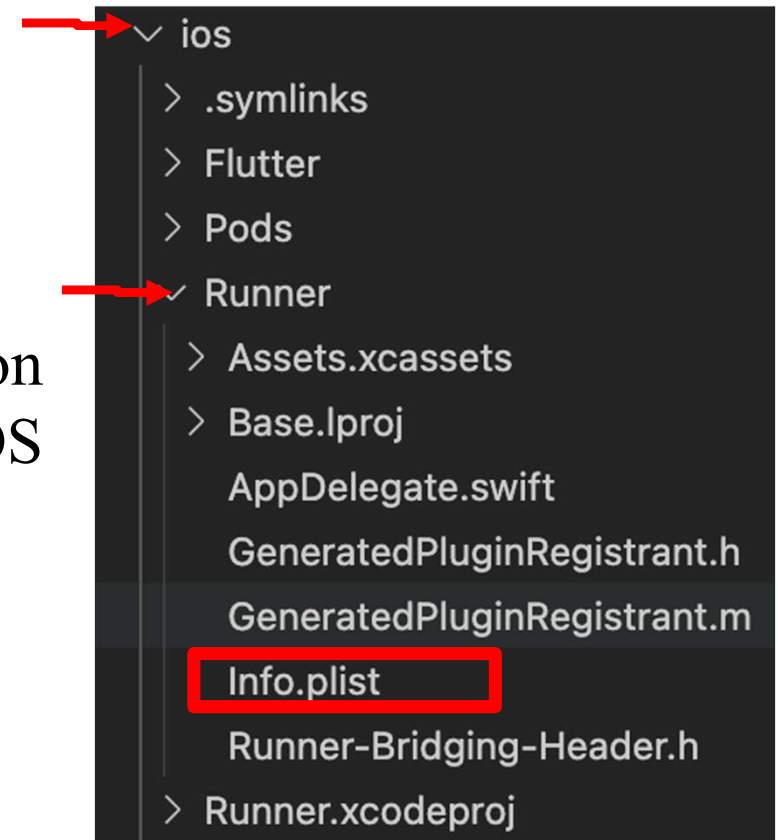
```
<uses-feature android:name="android.hardware.camera" android:required="true" />
```



Flutter project

- **Structure: "Info.plist file"**

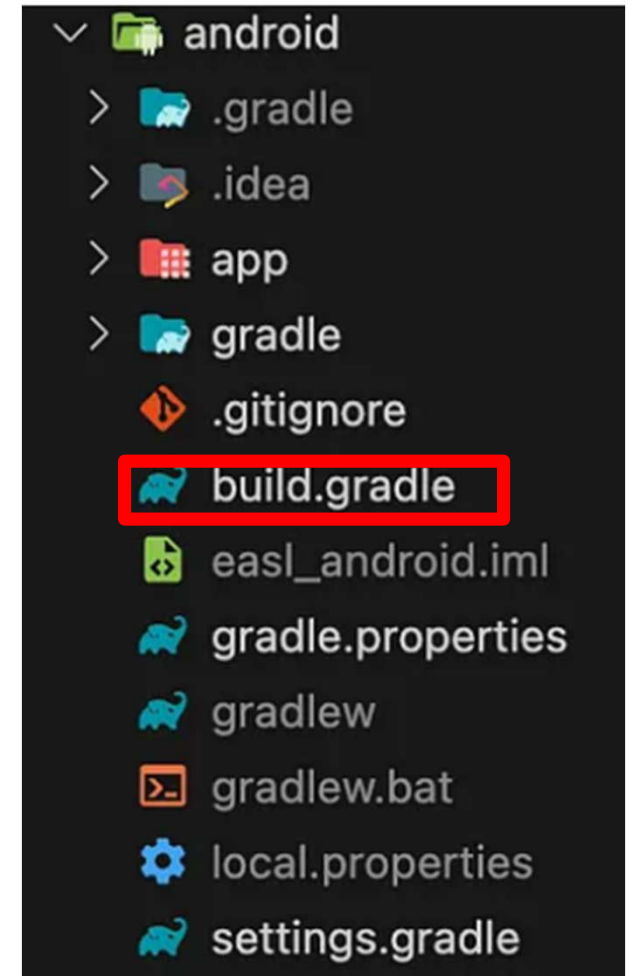
- The "*Info.plist*" file is the iOS equivalent of the *AndroidManifest.xml*.
- It contains essential configuration information about the app to the iOS build and the system.
- Defines:
 - App name and bundle identifier,
 - Version and permissions,
 - Supported device orientations,
 - Other system-related settings.



Flutter project

- **Structure: "build.gradle"**

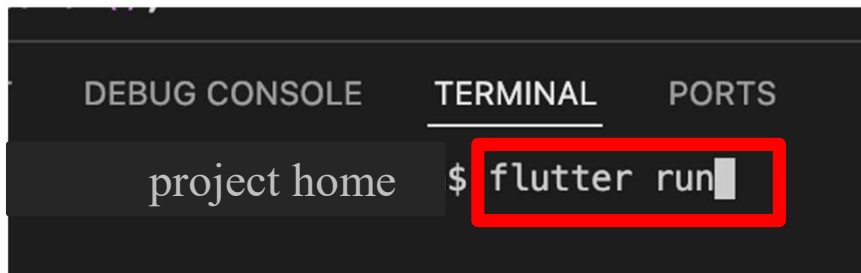
- Used for managing and building the Android-specific aspects of the Flutter app.
- It is responsible for building automatically the Android APK.
- It allows managing any native Android dependencies or plugins that the Flutter app may require on Android (e.g. `firebase_messaging` or `google_maps_flutter`).



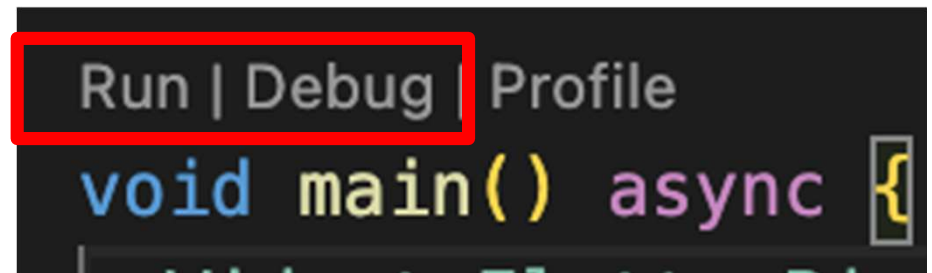
Flutter project

- **Run the app:** four options exist for running a Flutter project.

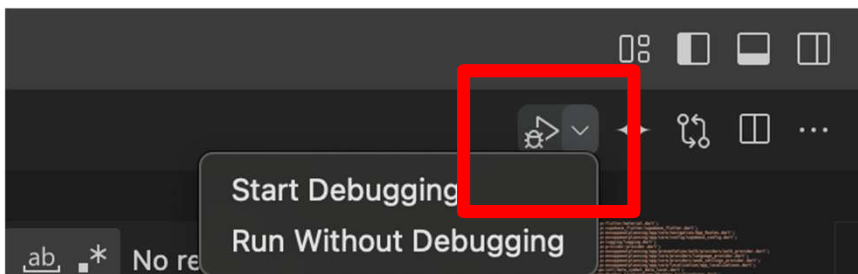
1. From the VS Code terminal:



2. From the main.dart file:



3. From VS Code menu:



4. From Android Studio:

