



# Lập trình đa nền tảng

Nguyễn Duy Nhật Viễn



# Chương 1

# Tổng quan



# **Giới thiệu**

# Khái niệm

○ **Đa nền tảng (cross-platform hay multi-platform):** bao hàm các phần mềm máy tính hay các phương thức điện toán và các khái niệm được thực thi hoàn chỉnh và vận hành cùng nhau trên nhiều nền tảng máy tính

○ Phần mềm đa nền tảng:

- yêu cầu phải thiết kế hoặc biên dịch từng phiên bản cho mỗi nền tảng
- có thể chạy trực tiếp trên bất cứ nền tảng nào mà không cần sự chuẩn bị thêm đặc biệt nào

- Ví dụ: Một ứng dụng có thể chạy được trên các nền tảng: MS Windows, Linux hoặc Macintosh.

# Khái niệm

## Các nền tảng:

- Một nền tảng là một sự kết hợp từ hai yếu tố phần cứng và phần mềm được dùng để chạy các phần mềm ứng dụng.
- Nền tảng có thể được miêu tả đơn giản như một **hệ điều hành** hay **kiến trúc máy tính**, hay sự kết hợp của cả hai

## Các nền tảng phần cứng

- Có thể được hiểu là **kiến trúc** của một máy tính hay **kiến trúc bộ xử lý**.
- Ví dụ, **CPU x86** và **x86-64** là một **kiến trúc máy tính** thường phổ biến cho PC, thường chạy hệ điều hành **Microsoft Windows**, mặc dù có thể chạy: **Linux**, **OpenBSD**, **NetBSD**, **Mac OS X** và **FreeBSD**.

## Các nền tảng phần mềm:

- Là **hệ điều hành** hoặc môi trường lập trình, nhưng thường kết hợp cả hai.
- Ví dụ: **MS-DOS (x86)**, **FreeDOS (x86)**, **Microsoft Windows (x86, x64)**, **Linux (x86, x64, PowerPC)**, và nhiều kiến trúc đa dạng khác), **Mac OS X (PowerPC, x86)**, **Java, ...**

# Khái niệm

## Phần mềm đa nền tảng

- phải có khả năng hoạt động trên nhiều hơn một **kiến trúc máy tính** hay **hệ điều hành**
- thời gian hay hiệu quả của tác vụ có thể khác nhau do các hệ điều hành khác nhau có các **giao diện lập trình ứng dụng** (API) khác nhau
- Hệ điều hành đặc thù có thể chạy trên các **kiến trúc máy tính** khác nhau, không có nghĩa phần mềm được viết cho một hệ điều hành sẽ tự động làm việc trên tất cả các **kiến trúc** mà hệ điều hành đó hỗ trợ
- Ví dụ: trước 2006, OpenOffice không chạy trên các dòng vi xử lý AMD64 hoặc EM64T. Sau đó, OpenOffice được hiệu chỉnh để chạy được trên tất cả các dòng vi xử lý.

# Khái niệm

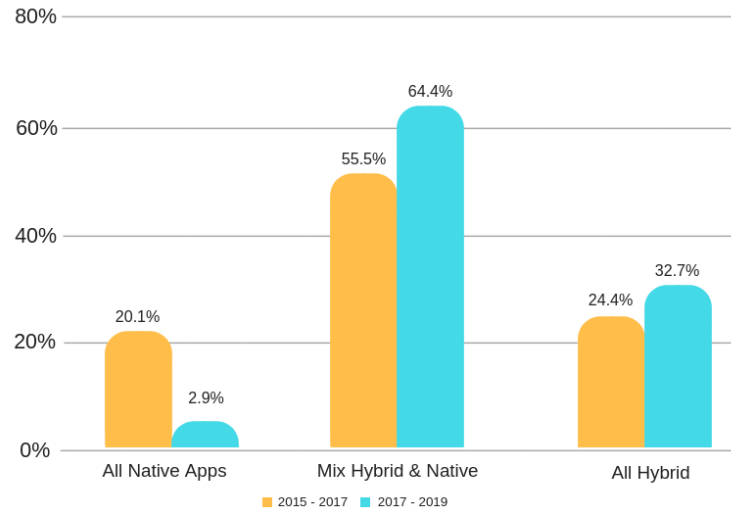
Native, web và hybrid application:

- Native app được tạo cho một nền tảng cụ thể, được xây dựng từ các ngôn ngữ lập trình hỗ trợ 1 nền tảng: Java, Kotlin cho Android; Objective-C và Swift cho iOS.
- Web app thực ra là các trang web, được thiết kế giống như các native app, thường được viết bằng HTML5. Người sử dụng có thể truy cập giống website, “cài đặt” trên màn hình chính bằng cách tạo shortcut cho trang đó.
- Hybrid app phụ thuộc vào chức năng đa nền tảng. Thường sử dụng các công nghệ web như HTML, CSS và Javascript framework.

# Khái niệm

## HYBRID VS NATIVE

- |  |   |
|--|---|
| <ul style="list-style-type: none"> <li>• single code base</li> <li>• low cost</li> <li>• simple building and testing</li> <li>• easy maintenance</li> <li>• fast delivery</li> </ul> | <ul style="list-style-type: none"> <li>• great performance</li> <li>• data protection</li> <li>• functionality</li> <li>• customer experience</li> <li>• comprehension</li> </ul> |
| <ul style="list-style-type: none"> <li>• bounded efficiency</li> <li>• dependent on Internet connection</li> </ul>   | <ul style="list-style-type: none"> <li>• time and money consuming</li> <li>• distributed codebase</li> </ul>  |



### Ứng dụng gốc

Hiệu suất nhanh  
 Bảo mật tốt  
 Trải nghiệm người dùng tốt  
 Chu kỳ phát triển chậm  
 Chi phí và thời gian thực hiện cao

### Ứng dụng web

Chi phí thấp  
 Dễ bảo dưỡng  
 Không yêu cầu người dùng cập nhật  
 Không truy cập phần cứng  
 Bảo mật thấp

### Ứng dụng lai

Linh hoạt  
 Chu kỳ phát triển nhanh  
 Chi phí thấp  
 Trải nghiệm người dùng bị hạn chế  
 Hiệu suất chậm

Web developer  
 Access to native  
 App store distribution  
 mul  
 platf  
 TML  
 HTML5  
 Web developer sk  
 Instant updates  
 Unrestricted distrib



# Phát triển ứng dụng gốc

## Phát triển ứng dụng gốc cho **Android**

- Ngôn ngữ lập trình: Java, Kotlin, C++, ...
- Android SDK (software development kit -bộ phát triển phần mềm)
- IDE (môi trường phát triển tích hợp): Android Studio, VSCode, ...
- Công cụ dòng lệnh cho Windows, Mac và Linux
- Nền tảng hỗ trợ và phát triển di động: Firebase, AWS, Azure, Heroku, Netlify, ...
- Cơ sở dữ liệu

## Phát triển ứng dụng gốc IOS:

- Ngôn ngữ lập trình: Objective-C, Swift, ...
- Sử dụng phần cứng của Apple để lập trình
- SDK iOS được tích hợp với khung giao diện người dùng Cocoa Touch
- Xcode: IDE chính thức để phát triển iOS
- **TestFlight**: bộ kiểm thử ứng dụng

# Phát triển ứng dụng đa nền tảng

Ứng dụng lai còn được gọi là ứng dụng di động đa nền tảng, thường lai giữa ứng dụng web và ứng dụng gốc.

Các ứng dụng dành cho thiết bị di động đa nền tảng thường bao gồm HTML / CSS với ngôn ngữ lập trình.

Một số framework phát triển phổ biến để tạo ứng dụng đa nền tảng

- React Native – cơ chế ứng dụng di động mã nguồn mở của Facebook
- Google Flutter - cơ chế phát triển ứng dụng của Google, hỗ trợ ứng dụng cho thiết bị di động, web và máy tính để bàn từ một cơ sở mã duy nhất
- Xamarin - nền tảng ứng dụng mã nguồn mở, hỗ trợ ứng dụng cho Android, iOS, tvOS, watchOS, macOS và Windows với .NET và C #.
- NativeScript - Khung mã nguồn mở để xây dựng ứng dụng di động với Angular, Vue.js, TypeScript hoặc JavaScript.
- Accelerator - phát triển ứng dụng di động bằng JavaScript.
- Ionic Framework - SDK mã nguồn mở
- Apache Cordova (trước đây là PhoneGap) - kết hợp mã nguồn mở với HTML, CSS & Javascript

# Ionic



Ionic là framework mã nguồn mở và miễn phí.

Ionic giúp xây dựng các ứng dụng dựa trên native kết hợp CSS3, HTML và JavaScript. Hỗ trợ HTML5.

Tính năng của Ionic

- Khung mã nguồn mở
- Cộng đồng hỗ trợ lớn
- Cảm nhận giống ứng dụng gốc

○ Thuận lợi của Ionic

- Phát triển ứng dụng đa nền tảng
- Giao diện người dùng
- Dễ dàng áp dụng
- Plugin Cordova
- Hiệu suất

React Native là một framework JavaScript React do Facebook tạo ra

Các ứng dụng được viết bằng React Native: Instagram, Messenger, Skype và Amazon Prime Videos.

Các tính năng của React Native

- Hỗ trợ thư viện của bên thứ ba
- Thời gian phát triển ngắn
- Cộng đồng hỗ trợ cộng mạnh mẽ
- Viết một lần và sử dụng mọi nơi

Thuận lợi của React Native

- Kiến trúc mô-đun
- Hiệu suất tối ưu
- Hiệu quả
- Khả năng tái sử dụng
- JavaScript
- Hỗ trợ cho các plugin của bên thứ ba

Xamarin là cơ chế được Microsoft hỗ trợ để tạo ứng dụng cho Android, iOS và Windows với sự trợ giúp của C # codebase. Các mã này có thể chia sẻ trên nhiều nền tảng bao gồm cả Windows và Mac OS.

## Đặc điểm của Xamarin

- Quản lý trình mô phỏng của Google
- Phần mềm nạp ứng dụng
- Trình quản lý SDK Android
- Tích hợp Visual Studio
- Kiểm soát tùy chỉnh

## Thuận lợi của Xamarin

- Môi trường phát triển ít phức tạp
- Khắc phục những hạn chế của các ứng dụng lai
- Nhanh hơn nhiều so với các ứng dụng lai
- Xamarin SDK đang được cập nhật và ổn định
- Hệ sinh thái phát triển hoàn chỉnh.

# Adobe PhoneGap



PhoneGap là cơ chế phát triển ứng dụng đa nền tảng kết hợp bằng HTML5, Javascript và CSS, cho phép xem các thay đổi ngay lập tức.

Các ứng dụng của Adobe PhoneGap có hiệu suất chất lượng cao và không có bất kỳ giới hạn phần cứng nào.

Có thể phát triển các ứng dụng cho Windows, Blackberry, Mac OS, Ubuntu và Firefox OS.

- Các ứng dụng được phát triển bằng cách sử dụng PhoneGap là Paylution, The DHS Program, HealthTap, Wikipedia và TripCase.
- Các tính năng của PhoneGap
  - Hệ thống hỗ trợ mạnh
  - Linh hoạt
  - Nền tảng mã nguồn mở
  - Sử dụng các công nghệ như CSS3, HTML5 và Javascript
- Thuận lợi:
  - Đồng nhất cho các nền tảng
  - Dựa trên JavaScript, CSS3 và HTML5



- Một trong những framework phát triển ứng dụng di động đa nền tảng tốt nhất.
- Sử dụng công cụ kết xuất 2D được gọi là Skia để tạo hình ảnh.
- Cho phép các nhà phát triển dễ dàng kiểm tra mà không cần khởi động lại dự án khi xảy ra lỗi.
- Ngôn ngữ lập trình Dart
- Các ứng dụng phát triển bằng Flutter: Alibaba, AppTree, Google Ads, Reflectly, Google Greentea, Tencent, Bitfolio.
- Ưu điểm của Flutter
  - Thư viện widget khổng lồ
  - Giải pháp một cửa để phát triển ứng dụng
  - Khả năng tích hợp, mở rộng và linh hoạt
  - Hiệu suất gần như native
- Lợi ích của Flutter
  - Hỗ trợ phát triển ứng dụng đa nền tảng
  - Giảm thời gian viết code
  - Tốc độ nhanh
  - Giao diện người dùng linh hoạt và đẹp
  - Các thành phần có dung lượng nhỏ

# Ruby On Rails



- Là framework Model-View-Controller mã nguồn mở sử dụng Ruby.
- GitHub, Airbnb, Shopify và Hulu sử dụng Rails.
- Thân thiện với người mới bắt đầu giúp các nhà phát triển ứng dụng di động và web
- Phát triển ứng dụng nhanh hơn và hiệu quả hơn
- Các tính năng của Ruby on Rails
  - Đối số từ khóa
  - Turbolinks
  - Action view, mailer, cable..
  - Kết xuất từ mọi nơi
- Thuận lợi của Ruby on Rails
  - Cộng đồng lớn
  - Tự động hóa kiểm tra
  - Thư viện
  - Dụng cụ
  - Chất lượng mã
  - Năng suất
  - Nhà phát triển có trách nhiệm



# Mobile Angular UI

- Kết hợp các tính năng của khung Angular và Bootstrap.
- Mobile Angular UI đơn giản hóa quá trình phát triển, có thể xây dựng các ứng dụng bằng HTML5.
- Cung cấp hỗ trợ tốt và miễn phí.
- Tính năng của Mobile Angular UI
  - Khu vực có thể cuộn
  - Trình đơn thả xuống
  - Navbars
  - Nút chuyển đổi
  - Phương thức và Lớp phủ
- Thuận lợi của Mobile Angular UI:
  - Các điều khiển mặc định được thiết kế tốt
  - Cung cấp các thành phần di động không có trong Bootstrap.



ANGULAR

The image features a large white circle centered on a black background. To the left of the circle, there is a dark gray circle with a lighter gray ring around it. To the right, there are several concentric white circles of varying sizes. The word "Flutter" is written in a bold, black, sans-serif font across the center of the white circle.

**Flutter**

# Giới thiệu

Flutter là tập các công cụ hỗ trợ xây dựng các ứng dụng trên Android, iOS, web, desktop và nhúng.

Đặc điểm:





- Miễn phí
- Mã nguồn mở
- Được hỗ trợ và bắt nguồn từ Google
- Phát triển và duy trì bởi một nhóm các nhà phát triển tại Google và lực lượng cộng tác viên hùng hậu
- Đang được hàng nghìn nhà phát triển trong các tổ chức trên thế giới sử dụng để xây dựng các ứng dụng
- Nhanh vì nó biên dịch thành các ứng dụng native thực sự
- Phát triển, bảo dưỡng code thuận tiện
- Hỗ trợ giao diện người dùng tốt



# Vì sao là Flutter

- Google's mission with Flutter is ... To build a better way to develop for mobile
- UI đẹp và biểu cảm: Thỏa mãn người dùng với các widget built-in đẹp mắt theo Material Design và Cupertino (iOS-flavor), các API chuyển động phong phú, scroll tự nhiên mượt mà và tự nhận thức được nền tảng.
- Hiệu năng cao: ứng dụng được xây dựng bằng Flutter được chứng minh chạy nhanh hơn các ứng dụng xây dựng bằng phương pháp truyền thống
- Phát triển ứng dụng thuận tiện: xây dựng giao diện người dùng, bổ sung tính năng và sửa. Tính năng hot reload giúp thử test nhanh và dễ dàng.
- Truy cập các tính năng và SDK native: Làm cho ứng dụng sống động với API của platform, SDK của bên thứ ba và native code. Cho phép bạn sử dụng lại mã Java, Swift và ObjC hiện tại của mình và truy cập các tính năng và SDK native trên iOS và Android.
- Phát triển ứng dụng thống nhất: Flutter có các công cụ và thư viện để giúp dễ dàng đưa ý tưởng của mình vào các ứng dụng trên iOS và Android.

# Vì sao là Flutter

				
	Xamarin	NativeScript	React Native	Flutter
<b>Year introduced</b>	2011	2014	2015	2018
<b>Backed by</b>	Microsoft	Telerik	Facebook	Google
<b>Presentation language</b>	XAML and/or xamarin.forms	Proprietary but looks like XML	Proprietary but looks like JSX	Dart
<b>Procedural language</b>	C#	JavaScript	JavaScript	Dart

# Ngôn ngữ DART

Là ngôn ngữ thuần hướng đối tượng được Google giới thiệu từ năm 2011

Mục đích: cung cấp sự lựa chọn hiện đại hơn Javascript nhưng không phủ định Javascript.

- Tối ưu cho client, nhất là ứng dụng đa nền tảng.
- Từng giữ vị trí Top 1 cho bình chọn ngôn ngữ không nên học năm 2018 theo Codementor
- Năm 2019 là một trong những ngôn ngữ được sử dụng nhiều nhất

## Worst Programming Languages to Learn in 2018 Rankings

Ranked from Worst to Best Languages to Learn

	Overall Rankings	Community Engagement	Job Market	Growth and Trends
1	 Dart	 Dart	 Dart	 Objective-C
2	 Objective-C	 CoffeeScript	 Rust	 CoffeeScript
3	 CoffeeScript	 Objective-C	 Elm	 Dart
4	 Erlang	 Lua	 Lua	 Perl
5	 Lua	 Elm	 Erlang	 Erlang
6	 Clojure	 Clojure	 Clojure	 Clojure
7	 Perl	 Elixir	 Kotlin	 Ruby
8	 Elm	 Erlang	 Elixir	 C#
9	 Elixir	 Kotlin	 R	 Lua
10	 Haskell	 Perl	 Perl	 C
11	 Rust	 Scala	 Haskell	 Haskell
12	 Scala	 TypeScript	 CoffeeScript	 Rust
13	 C	 Haskell	 Scala	 Elm
14	 Ruby	 Rust	 TypeScript	 Elixir
15	 Go	 R	 Objective-C	 Scala
16	 TypeScript	 Swift	 C	 Swift
17	 Swift	 Go	 Go	 Go
18	 Kotlin	 C	 Swift	 R
19	 R	 Ruby	 C#	 TypeScript
20	 C#	 C#	 Ruby	 Kotlin

# Ưu – nhược điểm của Flutter

## Flutter: Ưu điểm

- Mạnh về hiệu ứng, hiệu suất ứng dụng rất cao.
- Giao tiếp gần như trực tiếp với hệ thống
- Ngôn ngữ kiểu tĩnh nhưng với cú pháp hiện đại
- Có thể chạy được giả lập mobile ngay trên web, tiện cho việc phát triển.
- Có thể dùng để xây dựng các nền tảng nhúng vào ứng dụng native để tăng hiệu suất.

## Flutter: Nhược điểm

- Bộ render UI sử dụng bộ nhớ lớn
- UI lệ thuộc OS (ưu-nhược)
- Phải học thêm ngôn ngữ DART: ngôn ngữ mới-độc lập, không như JS hay Python có thể nhảy qua lại giữa front, back hay AI...
- Nhiều mô hình dữ liệu mới: bloc pattern, DART Streaming
- Phải thích ứng giữa các phiên bản



# Cài đặt

Tham khảo



# Cài đặt Flutter trên Windows

Vào <https://flutter.dev/docs/get-started/install> tiến hành cài đặt theo hệ điều hành của máy

Yêu cầu hệ thống:

- **Operating Systems:** Windows 7 SP1 or later (64-bit), x86-64 based

- **Disk Space:** 1.32 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)

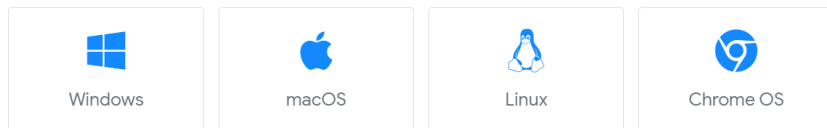
- Git for Windows 2.x, with the **Use Git from the Windows Command Prompt** option.

Sau khi tải về xong, giải nén, ta được một thư mục tên là flutter.

## Install

[Docs](#) > [Get started](#) > [Install](#)

Select the operating system on which you are installing Flutter:



# Cài đặt Flutter trên Windows

Vào <https://flutter.dev/docs/get-started/install> tiến hành cài đặt theo hệ điều hành của máy

Yêu cầu hệ thống:

**Operating Systems:** Windows 7 SP1 or later (64-bit), x86-64 based

**Disk Space:** 1.32 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.

## Install

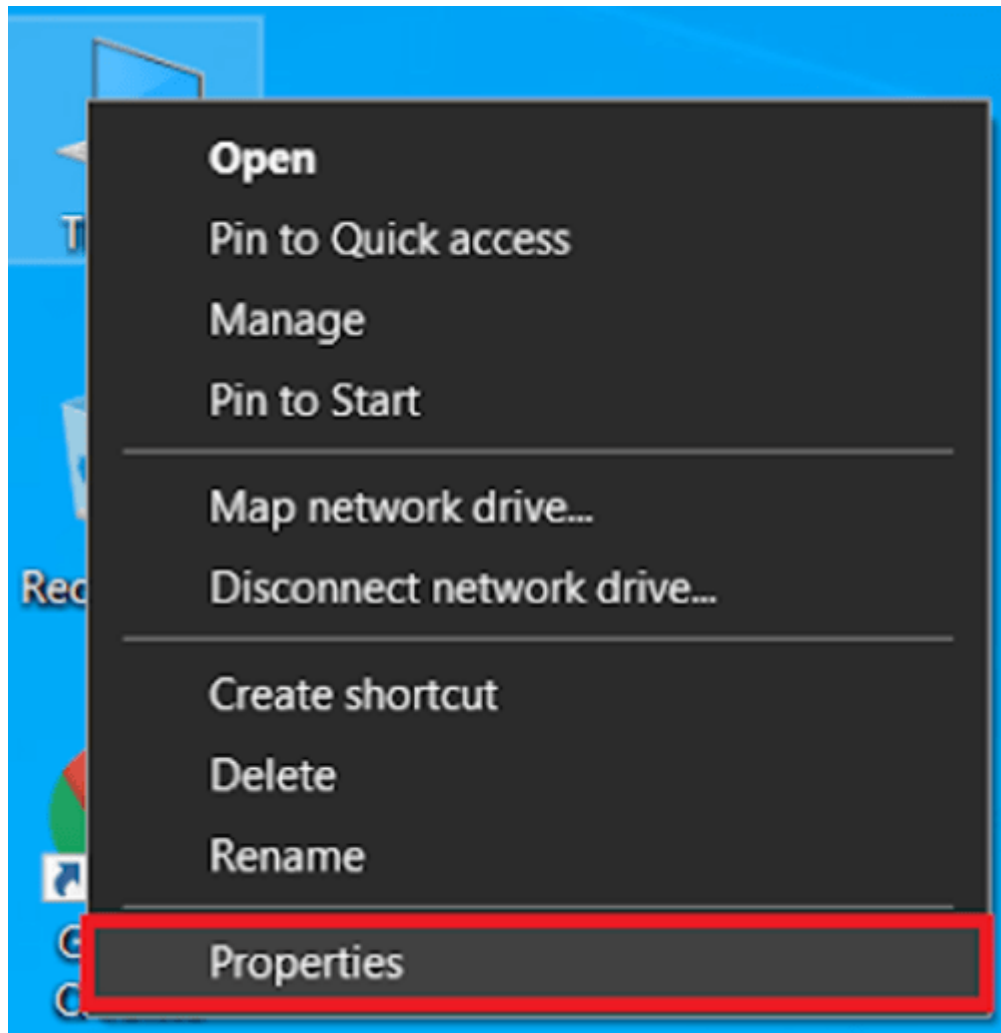
[Docs](#) > [Get started](#) > Install

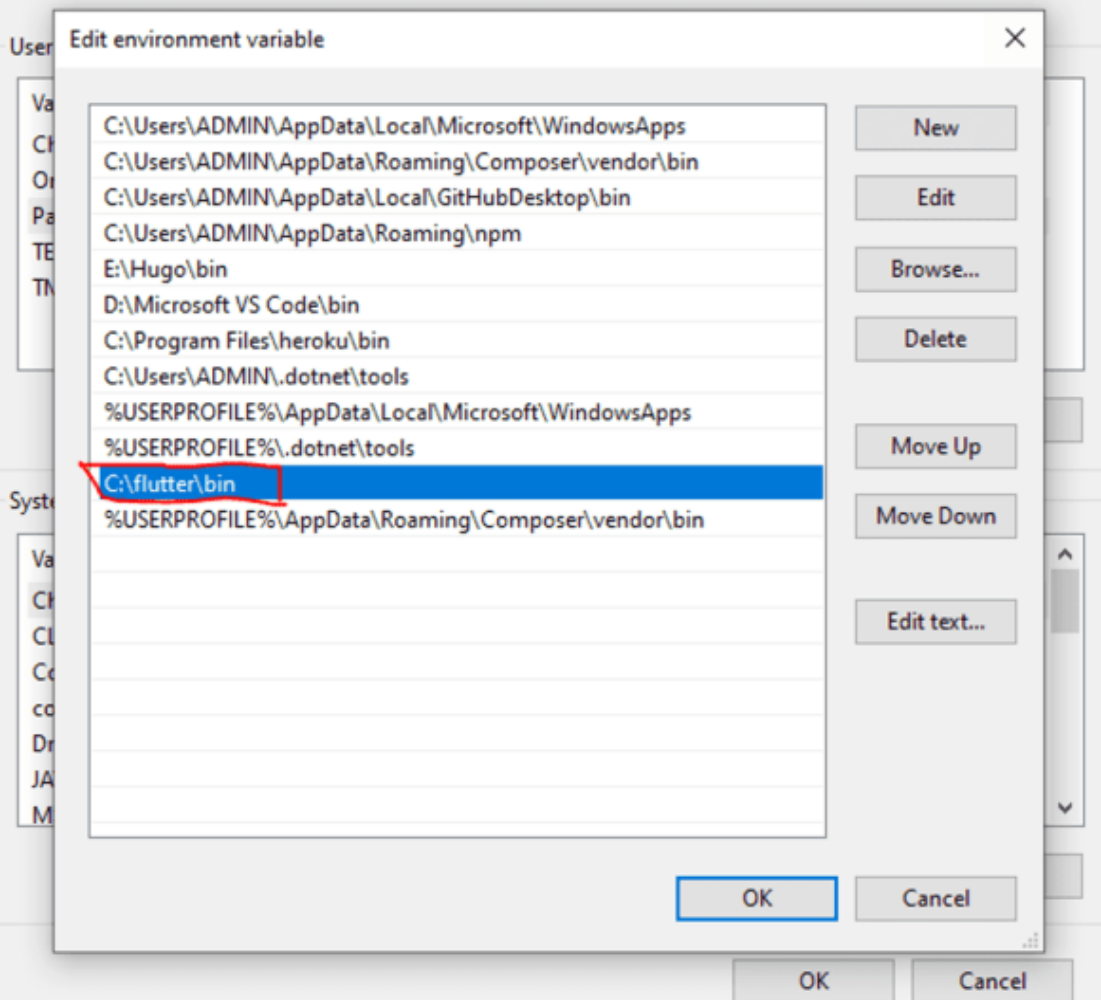
Select the operating system on which you are installing Flutter:



# Cài đặt Flutter trên Windows

- Tải bản ổn định Flutter SDK
  - Giải nén và đặt trong 1 thư mục (v.d: C:\src\flutter)
  - Có thể chạy bằng dòng lệnh 2 bước trên:
    - `git clone https://github.com/flutter/flutter.git -b stable`
  - Cập nhật đường dẫn:
  - Thêm đường dẫn `flutter\bin` vào biến môi trường
  - Chạy flutter doctor để kiểm tra
    - flutter doctor
- [~] Android toolchain - develop for Android devices
- Android SDK at D:\Android\sdk
  - X Android SDK is missing command line tools; download from <https://goo.gl/XxQghQ>
  - Try re-installing or updating your Android SDK,  
visit <https://flutter.dev/setup/#android-setup> for detailed instructions.





C:\> Administrator: Command Prompt

Microsoft Windows [Version 10.0.18362.720]

(c) 2019 Microsoft Corporation. All rights reserved.

C:\WINDOWS\system32>flutter --version

Flutter 1.16.3-pre.56 • channel master • <https://github.com/flutter/flutter.git>

Framework • revision 8857c4cec8 (2 weeks ago) • 2020-03-25 21:21:01 -0400

Engine • revision b235233e9d

Tools • Dart 2.8.0 (build 2.8.0-dev.17.0 2323087237)

C:\Windows\System32>

# Cài đặt Flutter trên macOS

## Yêu cầu hệ thống

- Operating Systems: macOS (64-bit)

- Disk Space: 2.8 GB (không bao gồm IDE và các tools).

- Tools: phụ thuộc các công cụ dòng lệnh trong hệ thống

- bash
  - curl
  - git 2.x
  - mkdir
  - rm
  - unzip
  - which

# Cài đặt Flutter trên macOS

Tải gói cài đặt Flutter SDK trong link sau:

- [flutter\\_macos\\_1.22.4-stable.zip](https://storage.googleapis.com/flutter_infra_release/flutter/1.22.4/flutter_macos_1.22.4-stable.zip) ([storage.googleapis.com](https://storage.googleapis.com/flutter_infra_release/flutter/1.22.4/flutter_macos_1.22.4-stable.zip))

Giải nén ra thư mục mong muốn

- `cd ~/development`

- `unzip ~/Downloads/flutter_macos_1.22.4-stable.zip`

Thêm đường dẫn flutter vào biến môi trường

- `export PATH="$PATH:`pwd`/flutter/bin"`

Chạy flutter doctor:

- `flutter doctor`

- [-] Android toolchain - develop for Android devices

- Android SDK at /Users/obiwan/Library/Android/sdk

- X Android SDK is missing command line tools; download from <https://goo.gl/XxQghQ>

- Try re-installing or updating your Android SDK,

- visit <https://flutter.dev/setup/#android-setup> for detailed instructions.



# Thiết lập platform

## Cài đặt iOS

### *Install Xcode*

Để phát triển ứng dụng cho iOS, cần phải có máy Mac với Xcode đã được cài đặt.

Cài phiên bản Xcode phù hợp với hệ điều hành đang sử dụng (dùng web download

<https://developer.apple.com/download/more/?name=Xcode> hoặc Mac App Store).

×

## Lịch sử của Phiên bản

12.2

12/11/2020

Xcode 12.2 includes Swift 5.3 and SDKs for iOS 14.2, iPadOS 14.2, tvOS 14.2, watchOS 7.1, and macOS Big Sur

- Xcode IDE adopts the refined macOS Big Sur design including updated toolbar, icons, and sidebar [thêm](#)

12.1

20/10/2020

Xcode 12.1 includes Swift 5.3 and SDKs for iOS 14.1, iPadOS 14.1, tvOS 14, watchOS 7, and macOS Catalina

### Platform features

- App Clips are a small part of your app that's discoverable at the moment it's needed [thêm](#)

12.0.1

24/09/2020

- This update fixes an issue that could cause Xcode to crash when viewing documentation

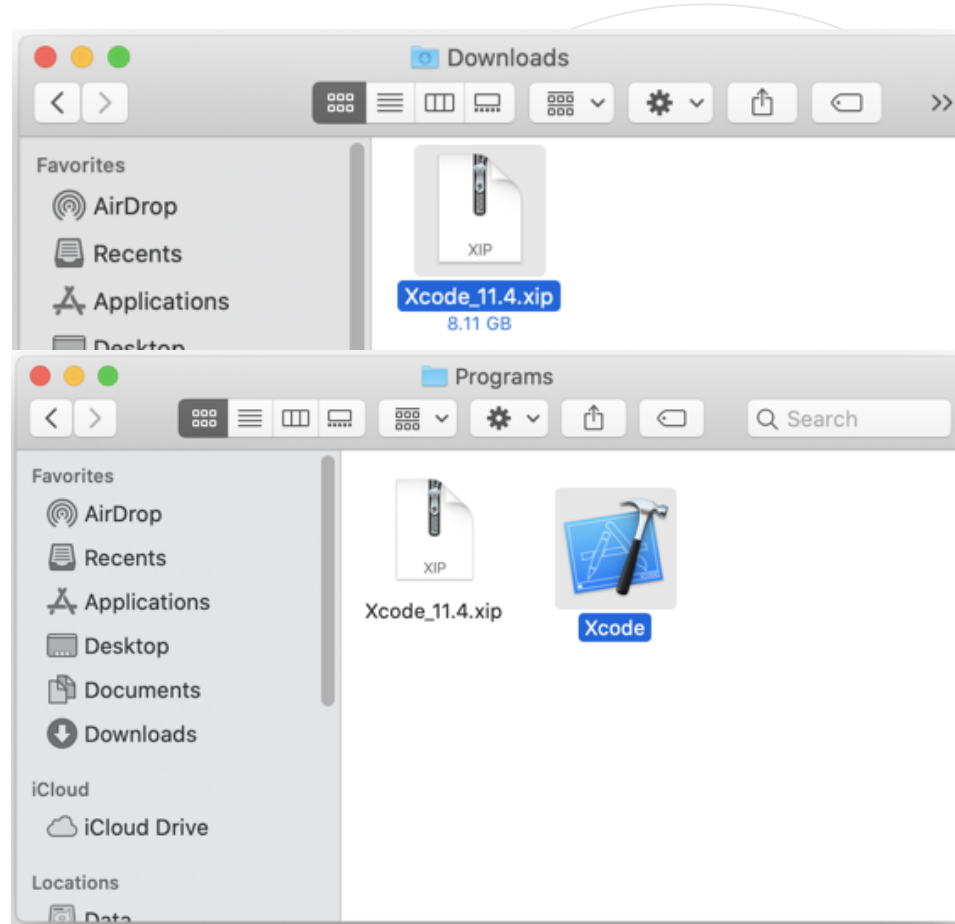
Xcode 12 includes Swift 5.3 and SDKs for iOS 14, iPadOS 14, tvOS 14, watchOS 7, and macOS Catalina [thêm](#)

# Thiết lập platform

## Cài đặt iOS

### *Install Xcode*

- Download, ta sẽ được một tập tin **XIP** (Một định dạng nén trong **MacOS**), hoặc một tập tin **DMG** (Một định dạng để cài đặt trong **MacOS**)
- Nếu là XIP, copy sang thư mục mong muốn và dùng **Archive Utility** kiểm tra chữ ký số (digital signature) và giải nén tập tin này

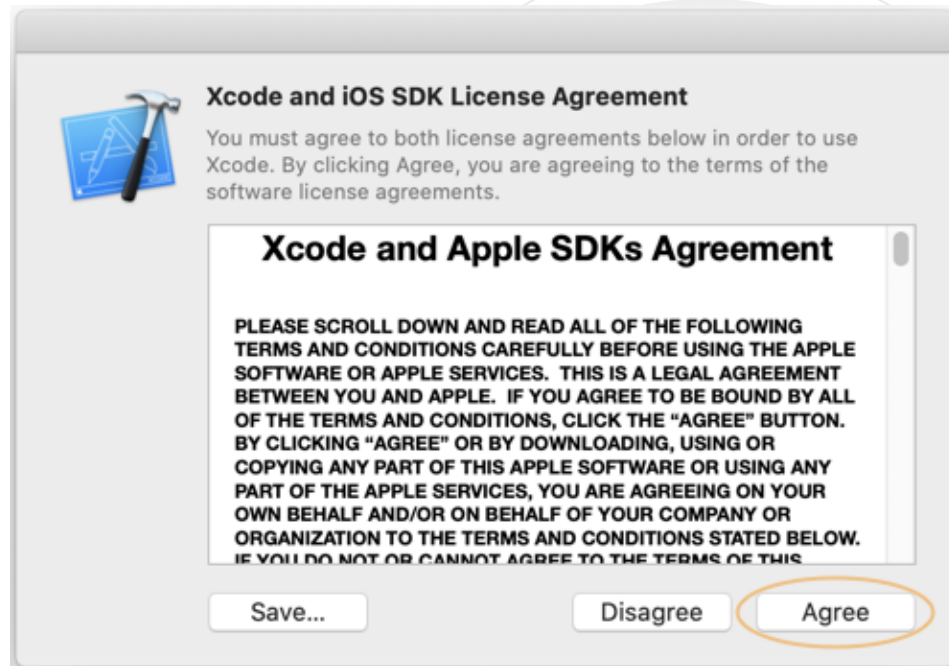


# Thiết lập platform

- Cài đặt iOS

- *Install Xcode*

- Nhấp kép chuột vào Xcode để cài đặt

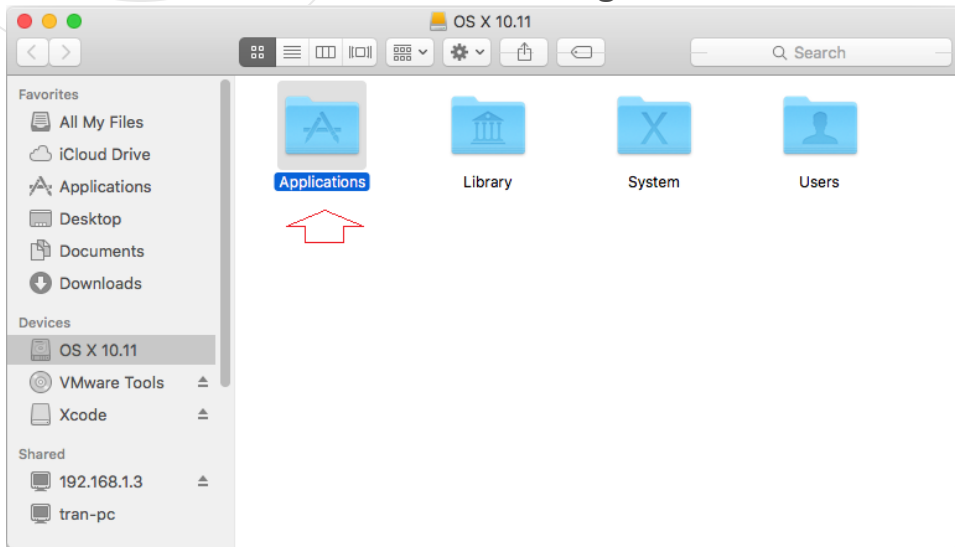


# Thiết lập platform

Cài đặt iOS

*Install Xcode*

Nếu là DMG, nhấp kép để cài  
Mở thư mục Application, nhấp  
vào biểu tượng Xcode



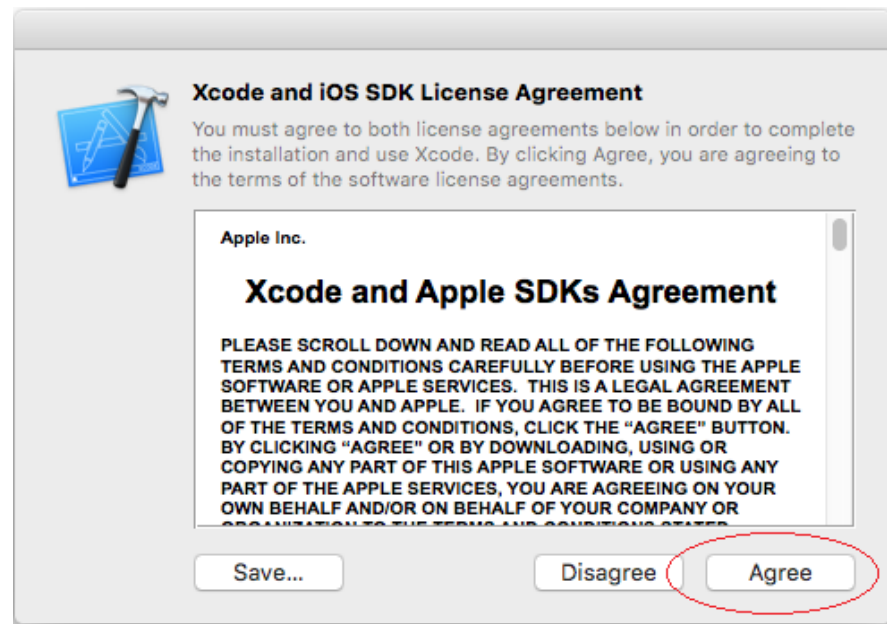
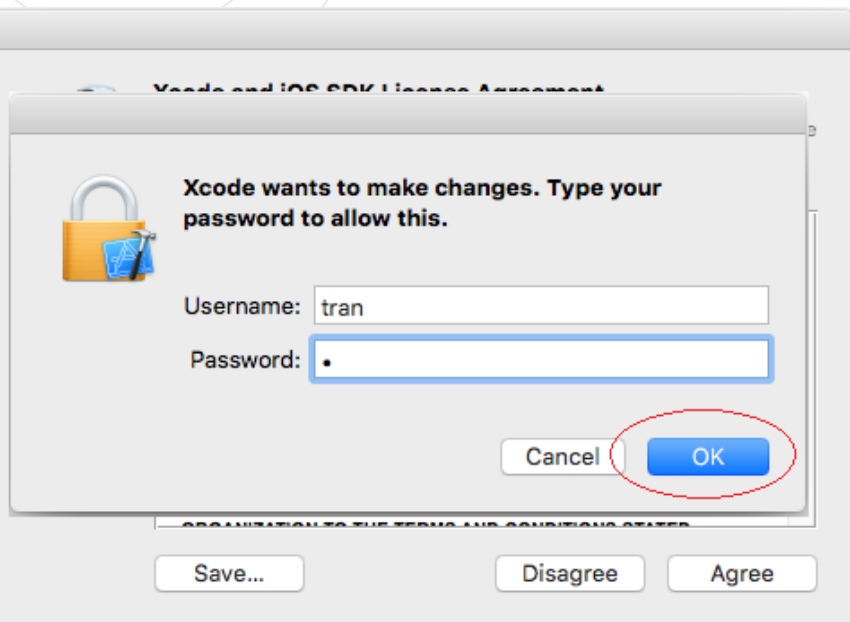
# Thiết lập platform

Cài đặt iOS

*Install Xcode*

Đồng ý với License

Nhập password nếu yêu cầu



# Thiết lập platform

Cài đặt iOS

*Install Xcode*

Cài xong



# Thiết lập platform

- Cài đặt iOS

- *Thiết lập bộ giả lập iOS:*

- Để chạy và kiểm tra ứng dụng Flutter trên trình mô phỏng iOS, hãy làm theo các bước sau:

- Trên Mac, tìm Trình mô phỏng qua Spotlight hoặc bằng cách sử dụng lệnh sau:

- `open -a Simulator`

- Đảm bảo trình mô phỏng sử dụng thiết bị 64 bit (iPhone 5s trở lên) bằng cách kiểm tra trong **Hardware > Device**.

# Thiết lập platform

- Cài đặt Android:

- *Cài đặt Android Studio <https://developer.android.com/studio>*

1. Khởi động Android Studio
2. Vào menu Configure và chọn AVD Manager
3. Chọn Create Virtual Device
4. Chọn một thiết bị và nhấn Next
5. Các bạn chọn một image x86 hoặc x86\_64
6. Trong phần Graphics, bạn chọn Hardware – GLES 2.0.
7. Nhấn nút ► để chạy máy ảo vừa tạo.



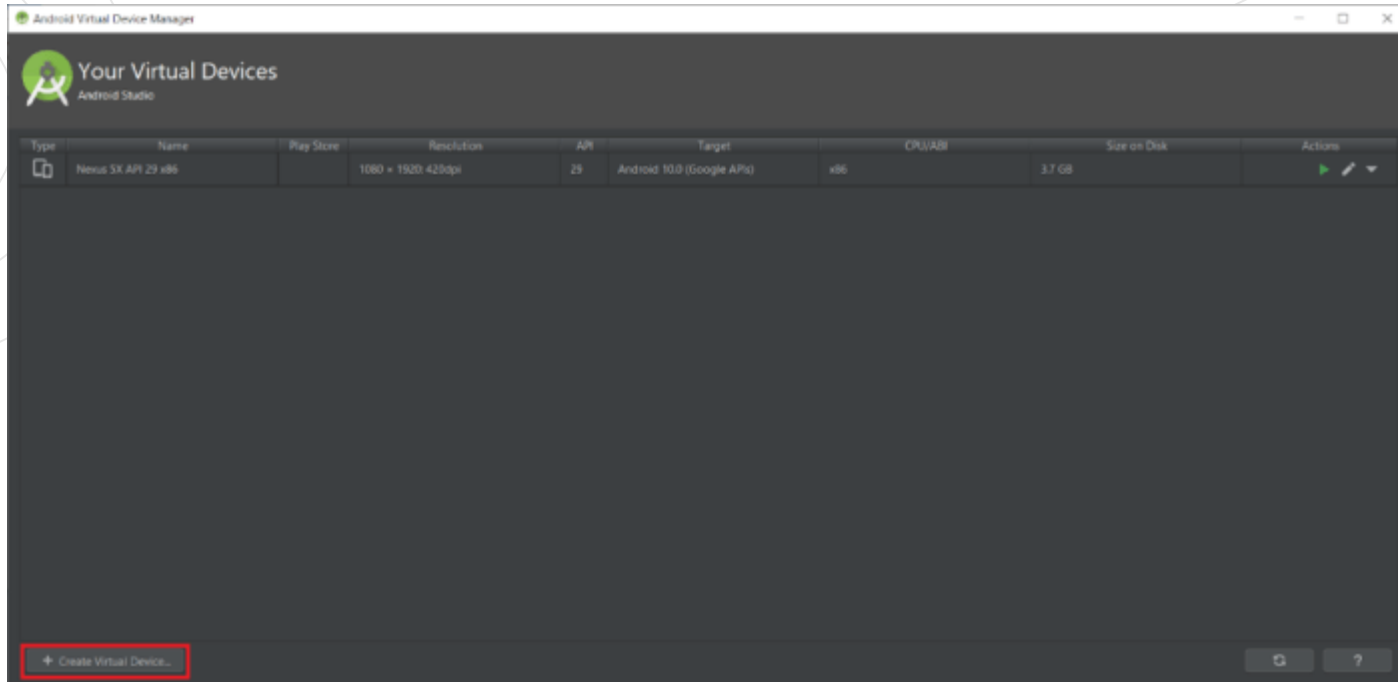


# Android Studio

Version 3.5.2

- + Start a new Android Studio project
- + Start a new Flutter project
- Open an existing Android Studio project
- Check out project from Version Control ▾
- Profile or debug APK
- Import project (Gradle, Eclipse ADT, etc.)
- Import an Android code sample

- ⚙ Configure ▾
  - AVD Manager
  - SDK Manager
  - Settings
  - Plugins
  - Default Project Structure...
  - Run Configuration Templates for New Projects
  - Import Settings
  - Export Settings
  - Settings Repository...
  - Edit Custom Properties...
  - Edit Custom VM Options...
  - Check for Updates
- Get Help ▾





## Select Hardware

Android Studio

## Choose a device definition

Q					
Category	Name	Play Store	Size	Resolution	Density
TV	Pixel XL		5.5"	1440x2560	560dpi
Phone	Pixel 3a XL		6.0"	1080x2160	400dpi
Wear OS	Pixel 3a	▶	5.6"	1080x2220	440dpi
Tablet	Pixel 3 XL		6.3"	1440x2960	560dpi
	Pixel 3	▶	5.46"	1080x2160	440dpi
	Pixel 2 XL		5.99"	1440x2880	560dpi
	Pixel 2	▶	5.0"	1080x1920	420dpi
	Pixel	▶	5.0"	1080x1920	420dpi
	Nexus S		4.0"	480x800	hdpi

New Hardware Profile

Import Hardware Profiles



Clone Device...



Pixel 2



Size: large  
Ratio: long  
Density: 420dpi



Previous

Next

Cancel

Finish



## System Image

Android Studio

## Select a system image

Recommended

x86 Images

Other Images

Release Name	API Level ▾	ABI	Target
Q Download	29	x86_64	Android 10.0 (Google Play)
Q	29	x86	Android 10.0 (Google APIs)
Q Download	29	x86_64	Android 10.0 (Google APIs)
Q Download	29	x86_64	Android 10.0
Q Download	29	x86	Android 10.0
Pie Download	28	x86_64	Android 9.0 (Google Play)
Pie Download	28	x86	Android 9.0 (Google APIs)
Pie Download	28	x86_64	Android 9.0 (Google APIs)
Pie Download	28	x86	Android 9.0
Pie Download	28	x86_64	Android 9.0
Oreo Download	27	x86	Android 8.1 (Google APIs)
Oreo Download	27	x86_64	Android 8.1

Q



API Level

29

Android

10.0

Google Inc.

System Image

x86

Questions on API level?

[See the API level distribution chart](#)

Previous

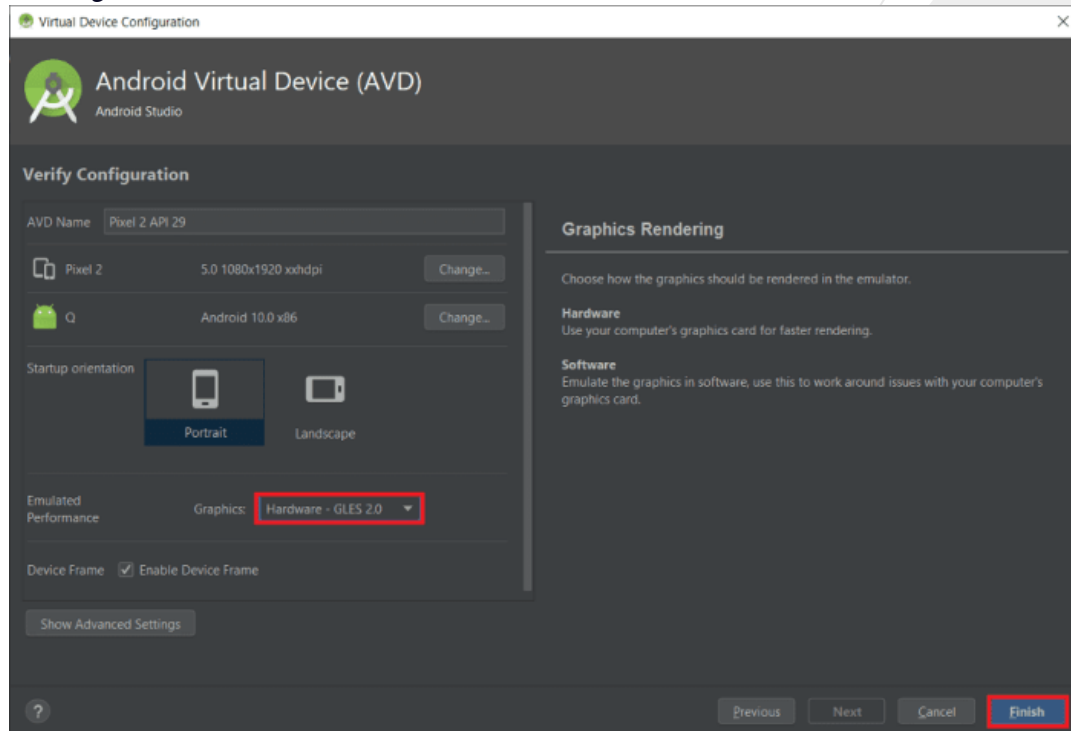
Next

Cancel

Finish

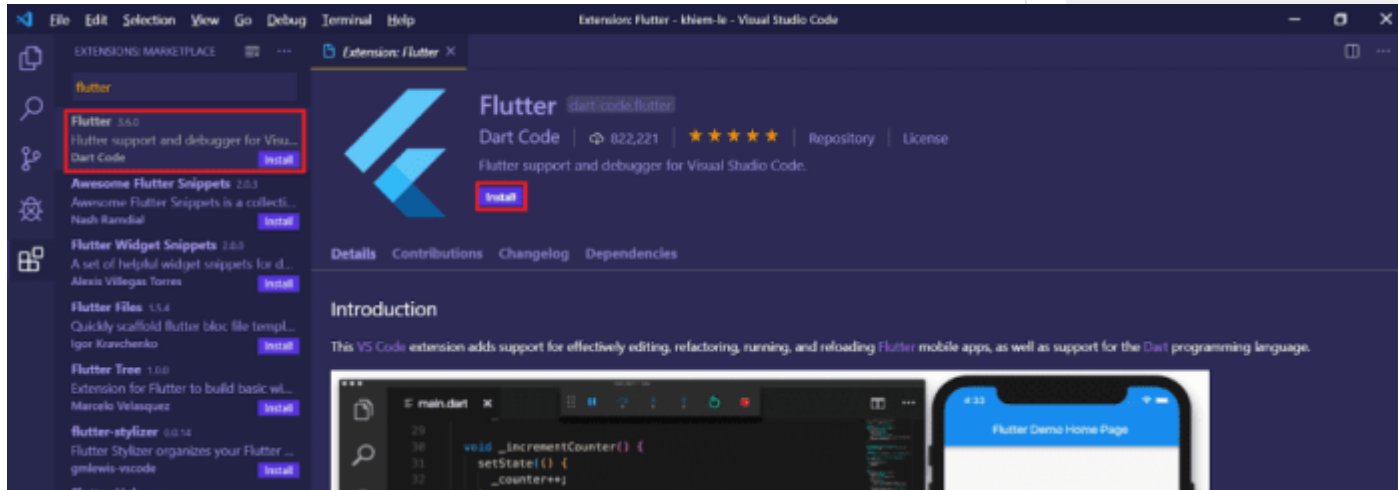
# Thiết lập platform

- Cài đặt Android:
- *Cấu hình máy ảo Android*



# VS Code

- Sau khi cài đặt xong VS Code chúng ta cài đặt extensions flutter cho VS Code:
  - Mở Visual Studio Code
  - Vào tab Extensions trên thanh sidebar bên trái
  - Search “flutter”, các bạn nhấn Install plugin đầu tiên của kết quả tìm kiếm. Cài plugin Flutter thì plugin Dart cũng sẽ được tự cài



The background features a large white circle on a black field. To the left, a dark gray circle overlaps the white one. To the right, several thin white concentric circles are visible.

# **Chương trình Flutter đầu tiên**

# Tạo project

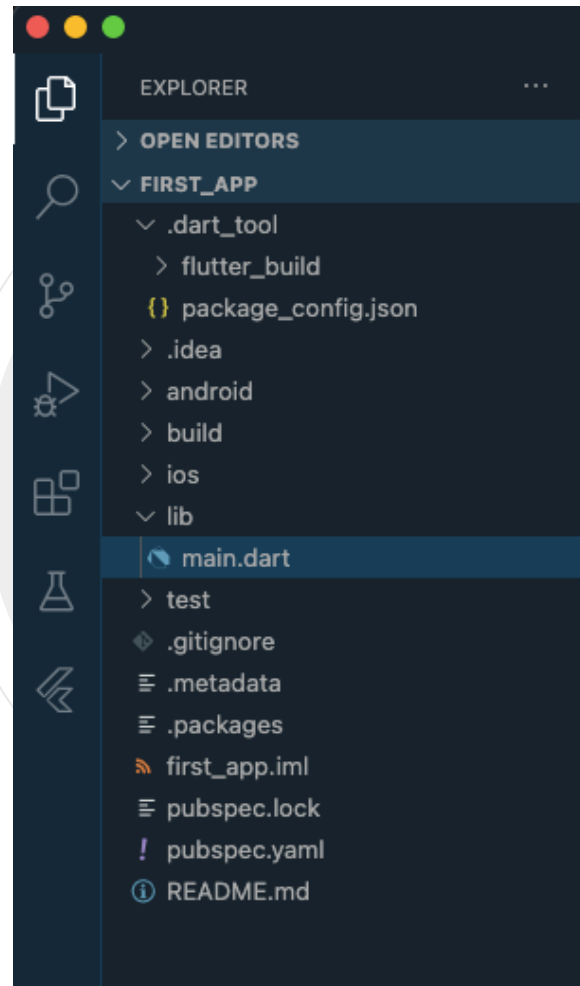
Tạo project mới có thể thực hiện từ nhiều cách:

- Từ dòng lệnh, gõ:
  - `flutter create first_app`
  - `cd first_app`
  - `flutter emulators` //để show các emulator hiện có
  - `flutter emulators --launch apple` //chạy emulator ios
  - `code .` //chạy vscode cho project trong thư mục hiện hành
- Từ Android Studio, chọn New Flutter app



# Cấu trúc project

- Folder lib: chứa các file .dart
- Folder là android và ios: chứa source của android và ios, cần thiết khi viết native code hoặc cấu hình thiết bị
- Folder test: chứa Unit Test cho dự án
- File pubspec.yaml: khai báo tên, mô tả, các thư viện, các asset như icon, ảnh hoặc font được sử dụng trong dự án
- File .gitignore và README.md được sử dụng với git, github (không liên quan đến source dự án)
- File .metadata và .packages là những file config, Flutter sử dụng để cấu hình dự án).
- 



# Cấu trúc chương trình Flutter

main.dart

lib > main.dart > ...

```
1 // Bước 1: import thư viện material - cung cấp các widget để code
2 import 'package:flutter/material.dart';
3
4 // Bước 2: khai báo hàm main - là nơi mà code sẽ thực thi đầu tiên
5 void main() {
6   // Bước 3: gọi hàm runApp truyền vào 1 object MyApp
7   // MyApp chính là widget root, là gốc của 1 cây widget sau này
8   runApp(MyApp());
9 }
10
11 //Bước 4: Định nghĩa lớp MyApp
12 class MyApp extends StatelessWidget {
13   // Widget này là root của ứng dụng
14   @override
15   Widget build(BuildContext context) {
16     return MaterialApp(
17       title: 'Flutter Demo', //title của ứng dụng
18       theme: ThemeData(
19         // Theme của ứng dụng
20         primarySwatch: Colors.blue, //màu toolbar
21         visualDensity: VisualDensity.adaptivePlatformDensity,
22         //mật độ hình ảnh thích ứng với nền tảng
23       ), // ThemeData
24       home: MyHomePage(title: 'Flutter Demo Home Page'),
25     ); // MaterialApp
26   }
27 }
```

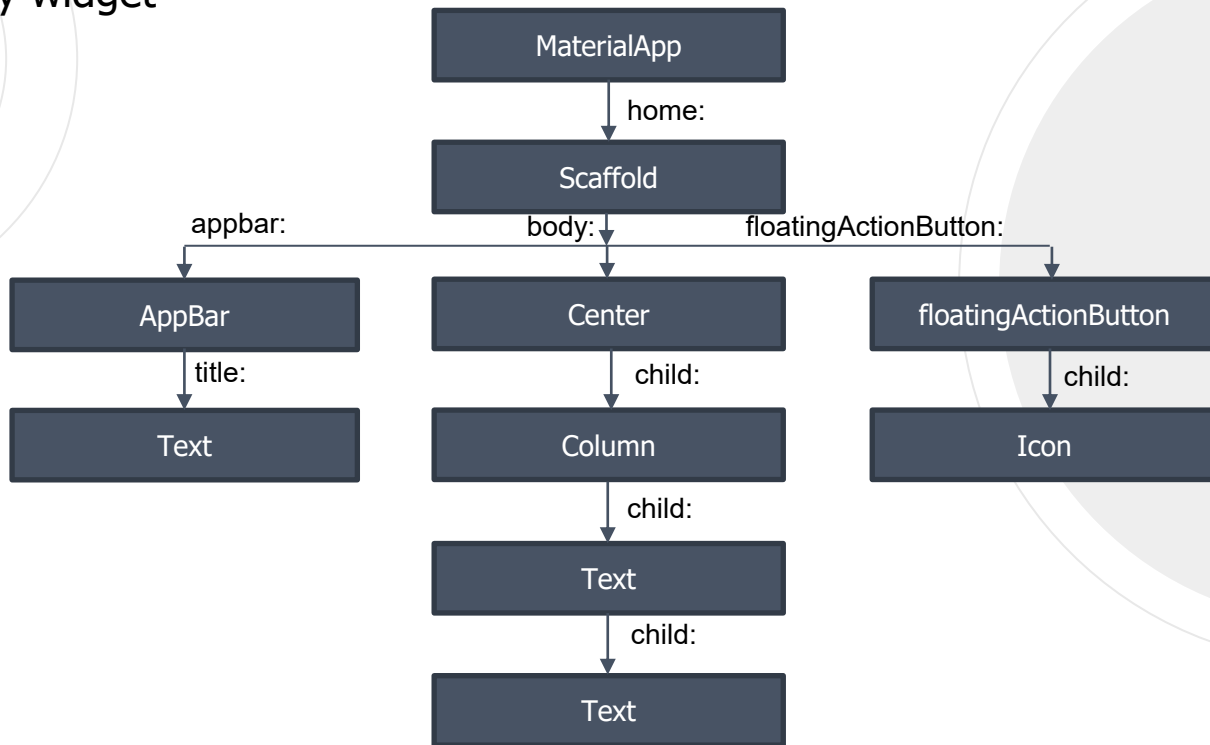
# Cấu trúc chương trình Flutter

```
29 class MyHomePage extends StatefulWidget {
30   MyHomePage({Key key, this.title}) : super(key: key);
31   // Widget là trang chủ của ứng dụng. Stateful nghĩa là có đối tượng State
32   // (định nghĩa bên dưới) chứa các trường ảnh hưởng đến giao diện
33   // Các trường trong lớp con widget luôn được định nghĩa là "final"
34   final String title;
35   @override
36   _MyHomePageState createState() => _MyHomePageState();
37 }
--
```

```
39 class _MyHomePageState extends State<MyHomePage> {
40   int _counter = 0;
41   void _incrementCounter() {
42     setState(() { // setState() để cập nhật giá trị
43       _counter++;
44     });
45   }
46
47   @override
48   Widget build(BuildContext context) {
49     // Phương thức này được chạy lại mỗi khi setState() được gọi
50
51     return Scaffold(
52       appBar: AppBar(
53         title: Text(widget.title),
54       ), // AppBar
55       body: Center(
56         child: Column(
57           mainAxisAlignment: MainAxisAlignment.center,
58           children: <Widget>[
59             Text(
60               'You have pushed the button this many times:',
61             ), // Text
62             Text(
63               '$_counter',
64               style: Theme.of(context).textTheme.headline4,
65             ), // Text
66           ], // <Widget>[]
67         ), // Column
68       ), // Center
69       floatingActionButton: FloatingActionButton(
70         onPressed: _incrementCounter,
71         tooltip: 'Increment',
72         child: Icon(Icons.add),
73       ), // FloatingActionButton
74     ); // Scaffold
75   }
76 }
```

# Cấu trúc chương trình Flutter

Cây widget



# File pubspec.yaml

! pubspec.yaml

```
1  # lưu ý: chú giải ở file này bằng dấu #, không phải //
```

2 name: first\_app # tên project

3 description: A new Flutter project. # phần mô tả project

4

5 publish\_to: 'none' # bỏ dòng này nếu muốn publish lên pub.dev

6

7 version: 1.0.0+1 # version của project

8

9 environment: # khai báo môi trường

10 | sdk: ">=2.7.0 <3.0.0" #sdk của flutter

11

12 dependencies: #khai báo thư viện

13 | flutter:

14 | sdk: flutter

15 | # các thư viện dùng trong chương trình được khai báo ở đây

16 | cupertino\_icons: ^1.0.0 #sử dụng cupertino icons của ios

17

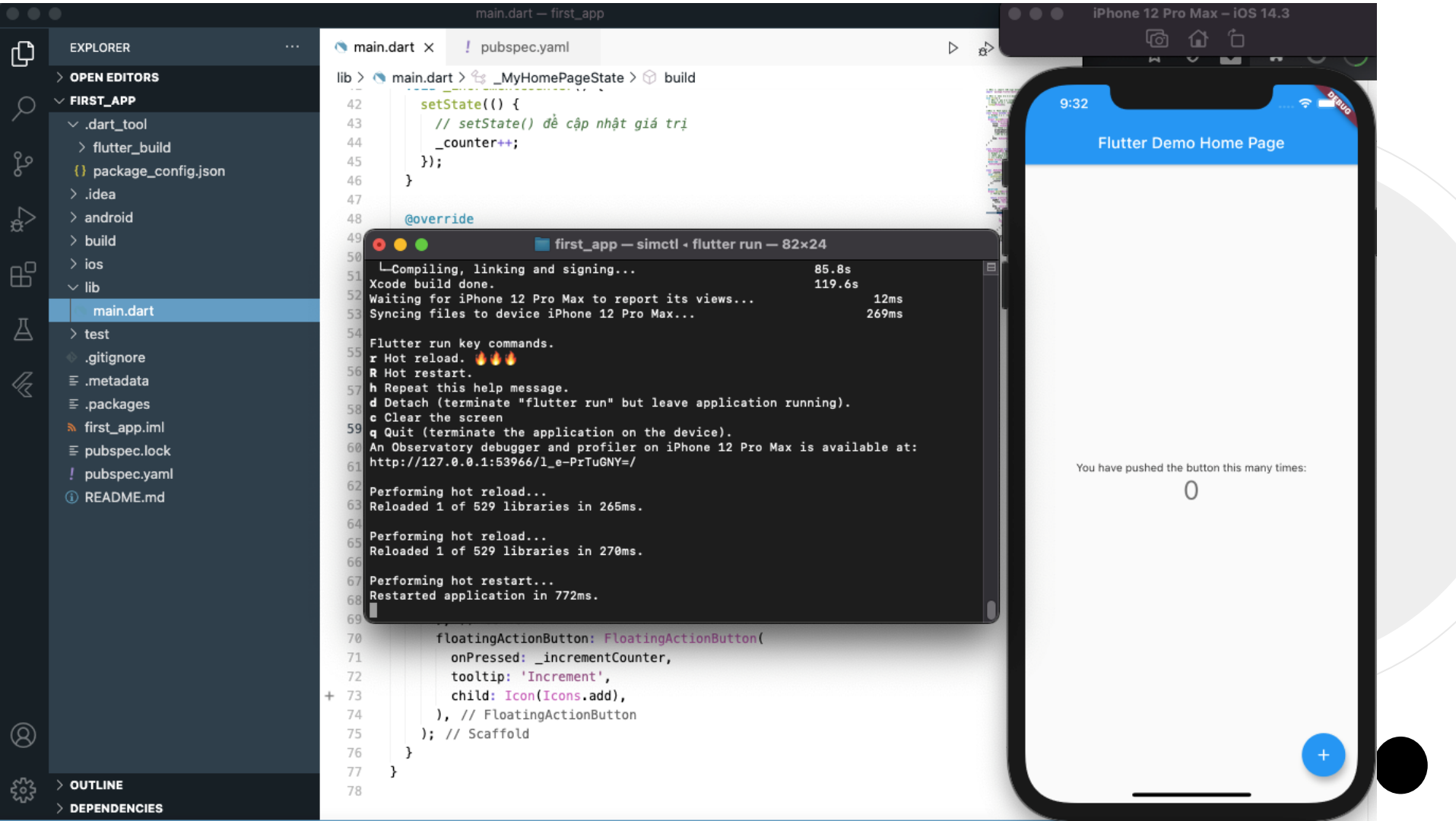
18 dev\_dependencies:

19 | flutter\_test:

20 | sdk: flutter

flutter:

```
22  
23  
24 uses-material-design: true # cho phép sử dụng Material icons  
25  
26 # để thêm các assets ứng dụng, sử dụng cú pháp  
27 # assets:  
28 #   - images/a_dot_burr.jpeg  
29 #   - images/a_dot_ham.jpeg  
30  
31 # thêm font, sử dụng  
32 # fonts:  
33 #   - family: Schyler  
34 #     fonts:  
35 #       - asset: fonts/Schyler-Regular.ttf  
36 #       - asset: fonts/Schyler-Italic.ttf  
37 #         style: italic  
38 #   - family: Trajan Pro  
39 #     fonts:  
40 #       - asset: fonts/TrajanPro.ttf  
41 #       - asset: fonts/TrajanPro_Bold.ttf  
42 #         weight: 700
```



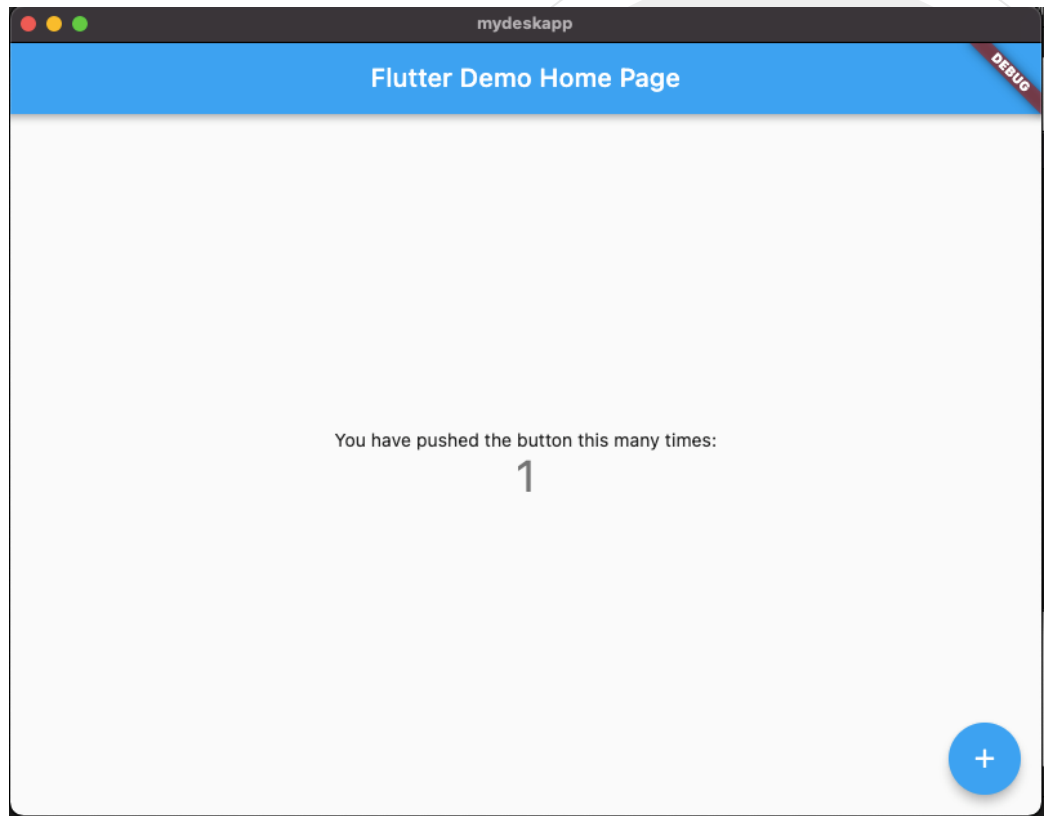
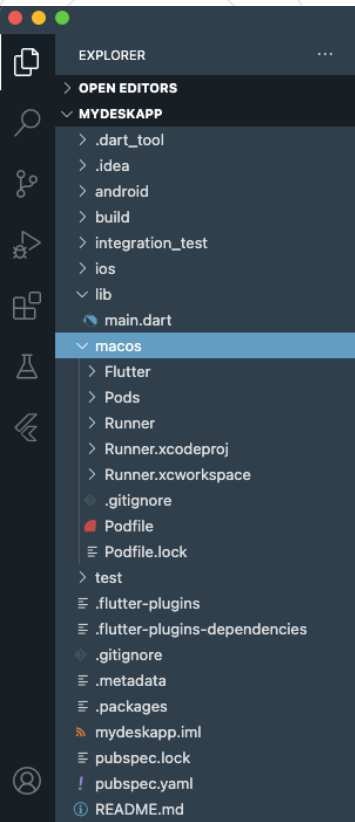
# Một số câu lệnh run và build app

- flutter devices
- flutter run
- Đối với desktop,
- flutter run -d windows
- flutter run -d macos
- flutter run -d linux

- flutter build appbundle
- flutter build apk --split-per-abi
- flutter install
- flutter build ios
- flutter build web
- flutter build macos
- flutter build windows
- flutter build linux



# Desktop app cho macos





# Một số câu lệnh run và build app

Run app android hoặc ios:

- flutter devices
- flutter run

Đối với desktop app, phải cấu hình trước khi tạo app:

- flutter channel dev
- flutter upgrade
- flutter config --enable-<platform>-desktop

Platform là windows, macos hoặc linux:

- flutter config --enable-windows-desktop
- flutter config --enable-macos-desktop
- flutter config --enable-linux-desktop

- flutter run -d windows
- flutter run -d macos
- flutter run -d linux