Flutter

การพัฒนา Mobile Application ด้วย Flutter

โดย ร.ท.สมโภชน์ กุลธารารมณ์

ผู้สอน

ร.ท. สมโภชน์ กุลธารารมณ์ อาจารย์แผนกการศึกษา
กองวิทยาการศูนย์ไซเบอร์ทหารกองบัญชาการกองทัพไทย

- ประวัติการศึกษา
  - ปริญญาโท
    - วท.ม. สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ
  - ปริญญาตรี
    - วท.บ. สาขาเทคโนโลยีสารสนเทศ มหาวิทยาลัยเทคโนโลยีราชมงคลพระนค

- ประวัติการทำงาน
  - พ.ศ. 2549- 2551
    - ตำแหน่ง Developer บริษัท AISOFT (บริษัทรับพัฒนาแอพพลิเคชันด้านการท่องเที่ยว)
  - พ.ศ. 2551- 2561
    - หัวหน้างานพัฒนาซอฟต์แวร์ สำนักวิทยบริการและเทคโนโลยีสารสนเทศมหาวิทยาลัยเทคโนโลยีราชมงคลพระนคร
  - พ.ศ. 2561 – ปัจจุบัน
    - อาจารย์แผนกการศึกษา กองวิทยาการศูนย์ไซเบอร์ทหาร กองบัญชาการกองทัพไทย

Certificate

CompTIA: Classroom Trainer (CTT+), Project+, A+, Network+ , Security+

ORACLE: Oracle Database 11g Administrator Certified Associate

ADOBE: Visual Communication using Adobe Photoshop® CS6

NSTDA (สวทช): Information Technology Professionals Examination Council: ITPEC

Microsoft: Microsoft Certified Trainer, Microsoft Certified Solution Exapert

Flutter Intro

- Flutter คือ Framework ที่ใช้สร้าง UI สำหรับ Mobile Application ที่สามารถทำงานได้ทั้ง iOS และ Android โดยภาษาที่ใช้ใน Flutter นั้นจะเป็นภาษา dart ซึ่งถูกพัฒนาโดย Google และที่สำคัญคือเป็น open source ที่สามารถใช้งานได้แบบฟรี ๆ อีกด้วย

Flutter

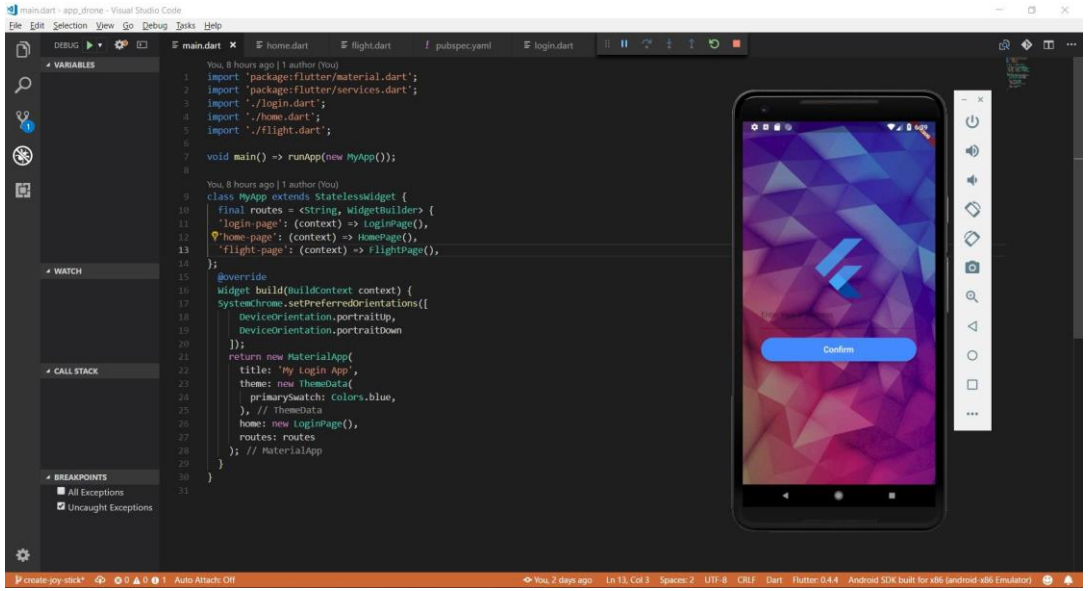
Flutter Intro (ต่อ)

- *ตัวอย่าง syntax ของภาษา dart ที่ใช้ใน Flutter ซึ่งจะมีความคล้ายกับภาษา Java เนื่องจาก dart เป็นภาษาที่รองรับ OOP และมีแนวคิดของ class และ inheritance เช่นเดียวกับภาษา Java นั่นเอง*

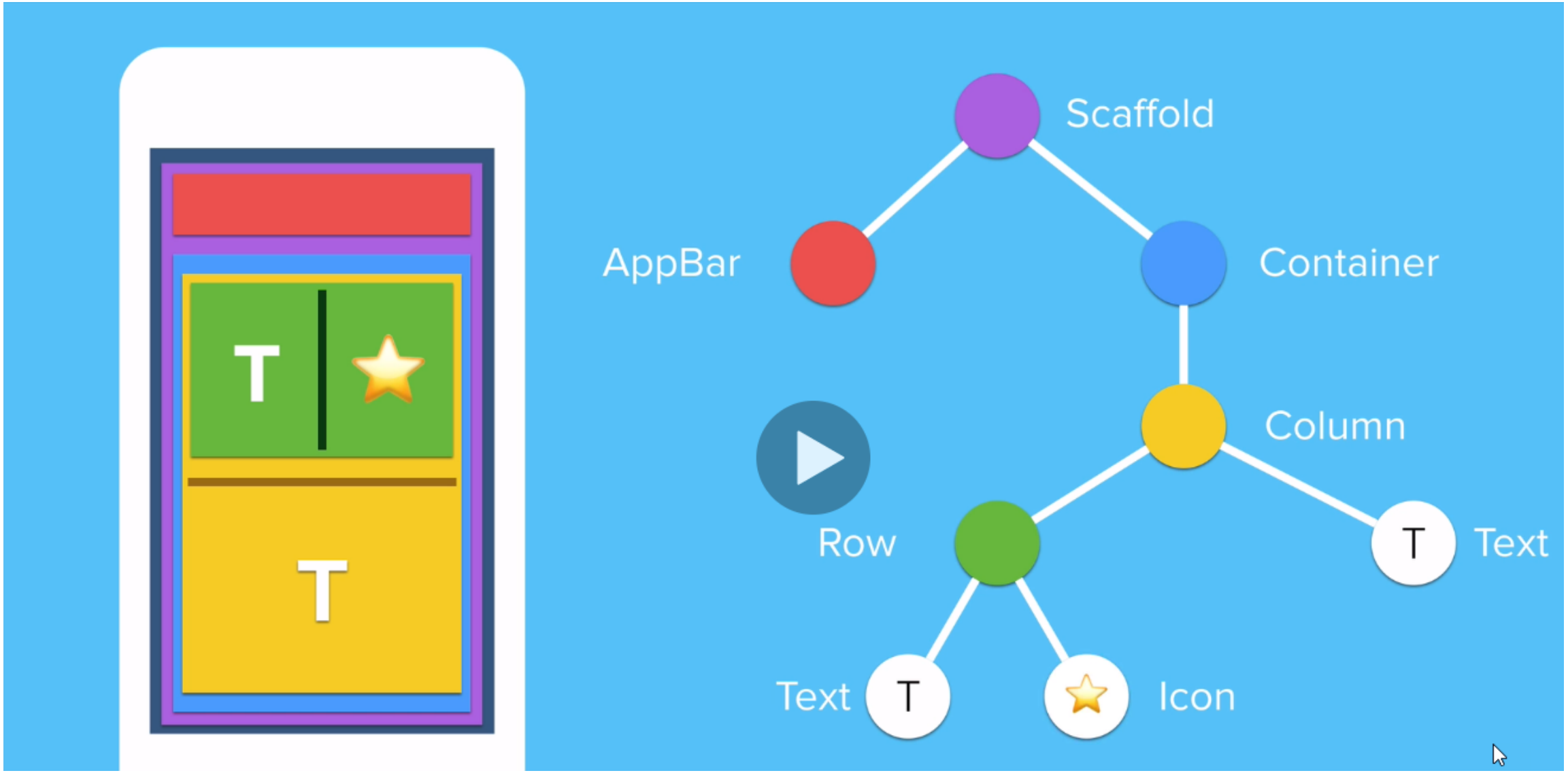
```dart
import 'package:flutter/material.dart';

void main() {
  runApp(new MaterialApp(
    home: new MyApp(),
  ));
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return new Scaffold(
      appBar: new AppBar(
        title: new Text("Example App"),
        backgroundColor: Colors.blue,
      ),
      backgroundColor: Colors.blue,
      body: new Center(
        child: new Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            new Icon(Icons.favorite, color: Colors.redAccent, size: 200.0,
            ),
          ],
        ),
      ),
    );
  }
}
```

จุดเด่นของ Flutter คืออะไร?

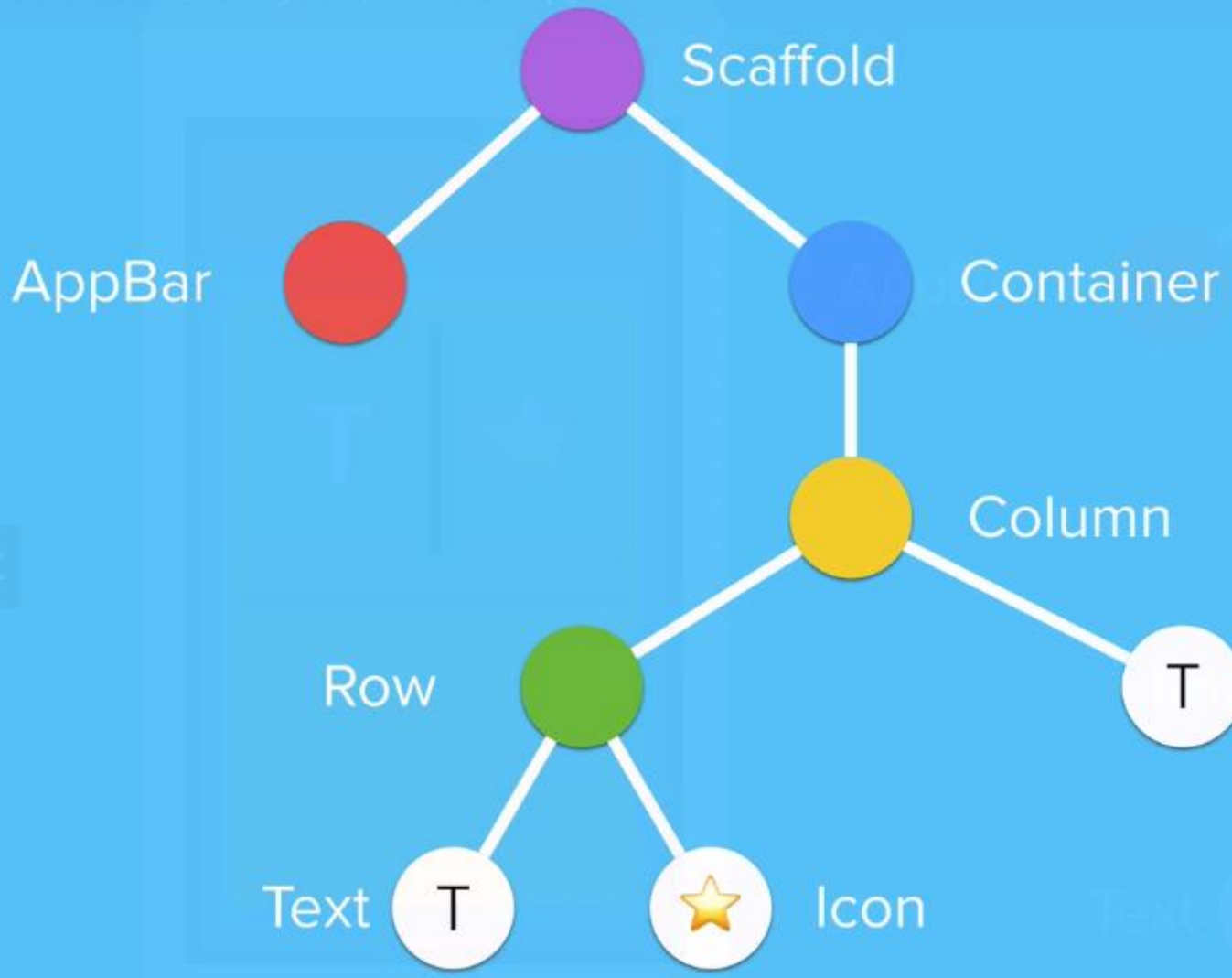
- จุดเด่นหลัก ๆ ของ Flutter คือ ระบบ Hot Reload โดยเมื่อมีการทดสอบ, การสร้าง, การ add features หรือการกระทำต่าง ๆ กับ UI จะต้องมีการ reload เพื่อให้หน้า UI update ซึ่งระบบ Hot Reload จะเข้ามาช่วยในส่วนของการ reload โดยจุดเด่นของระบบนี้คือการย่นระยะเวลาที่ใช้ในการ reload ให้เหลือเพียงเสี้ยววินาทีเท่านั้น ทำให้การพัฒนา UI ของ application มีความรวดเร็วขึ้นอย่างมาก และยังมีจุดเด่นอื่น ๆ ที่ช่วยให้การพัฒนาเป็นไปได้ง่ายขึ้นไม่ว่าจะเป็น Build-In ที่ช่วยในการออกแบบ UI ให้มีความสวยงามยิ่งขึ้นอย่าง Material Design และ Cupertino (iOS-flavor), มี Framework ที่ช่วยให้การทำ animation ต่าง ๆ หรือ gesture ของ UI เป็นเรื่องง่ายยิ่งขึ้น และยังสามารถใช้งานร่วมกับ IDE ที่กำลังเป็นที่นิยมอยู่ในปัจจุบันอย่าง VS Code และ Android Studio ได้อีกด้วย



Anatomy of Flutter



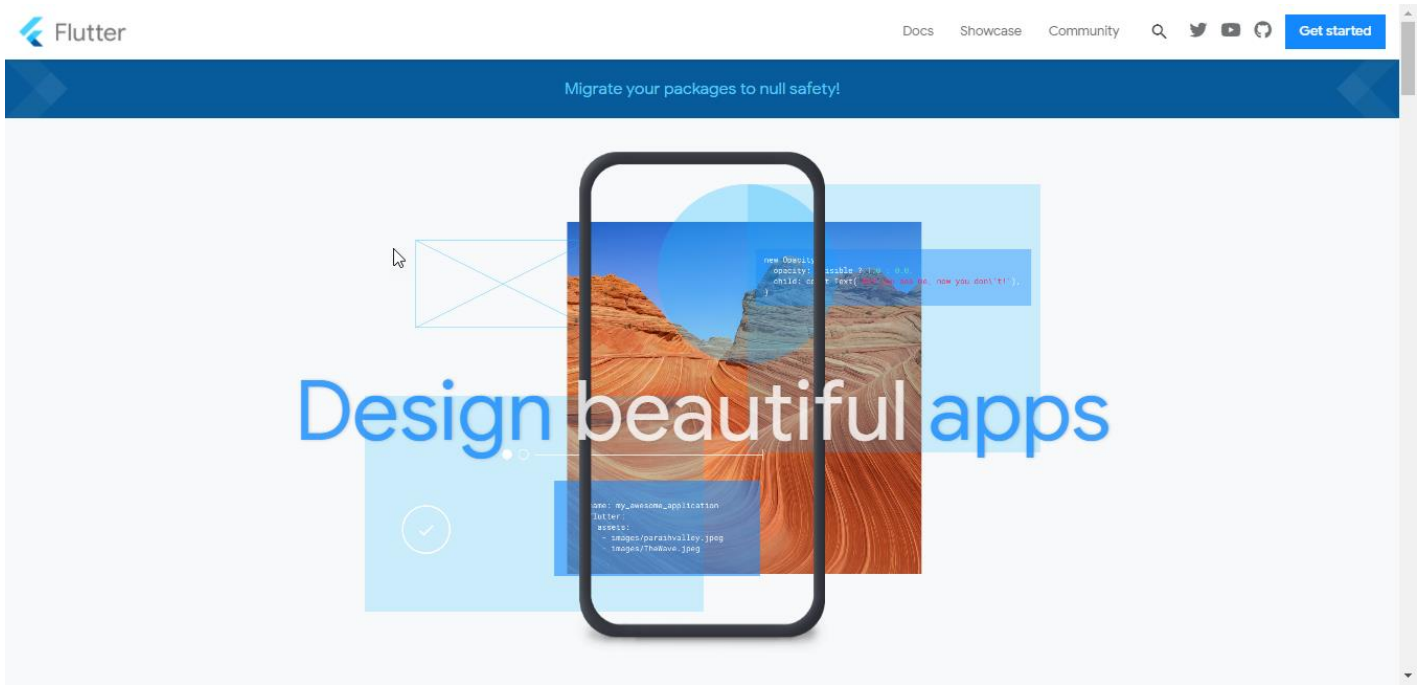
Anatomy of Flutter

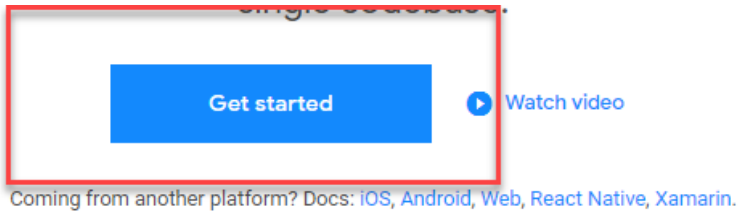



```
Scaffold(
  appBar: AppBar(),
  body: Container(
    child: Column(
      children: [
        Row(
          Text(),
          Icon(),
        ),
        Text(),
      ]
    )
  ),
)
```

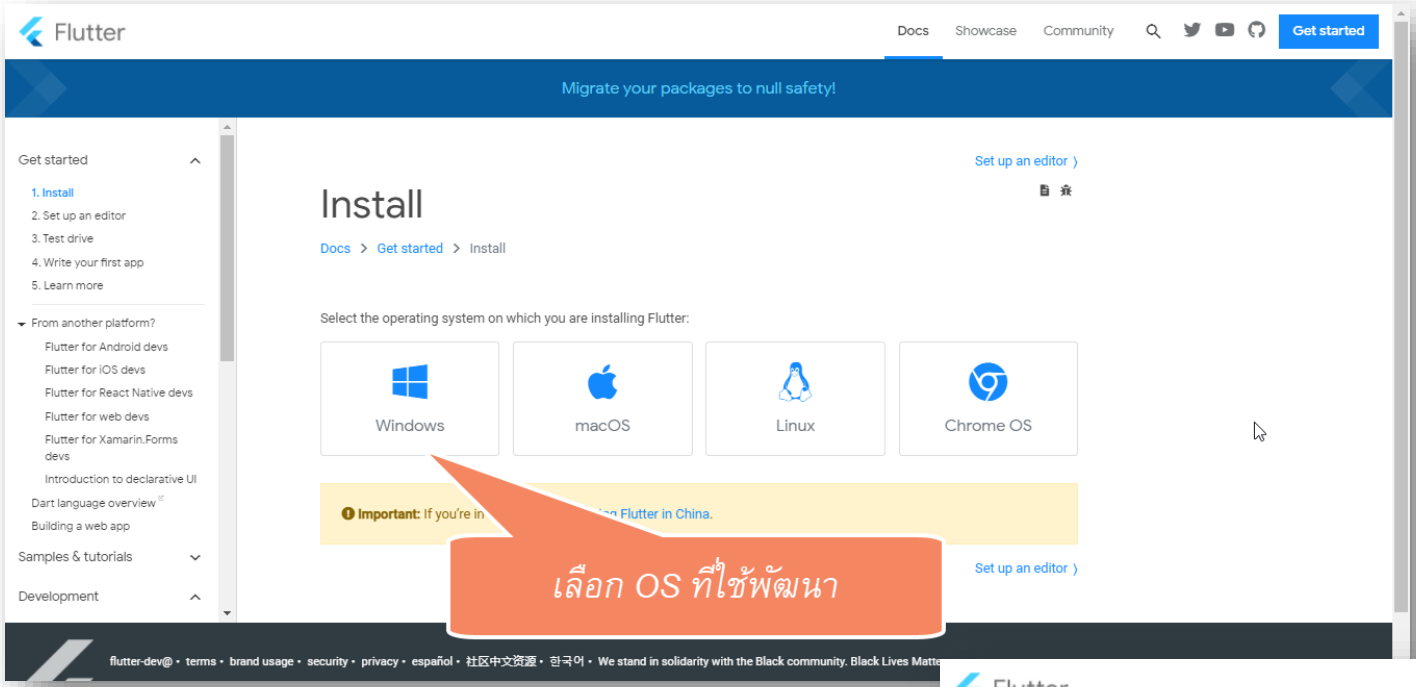
1. Install Flutter

1.1 เข้า Link Download https://flutter.dev/

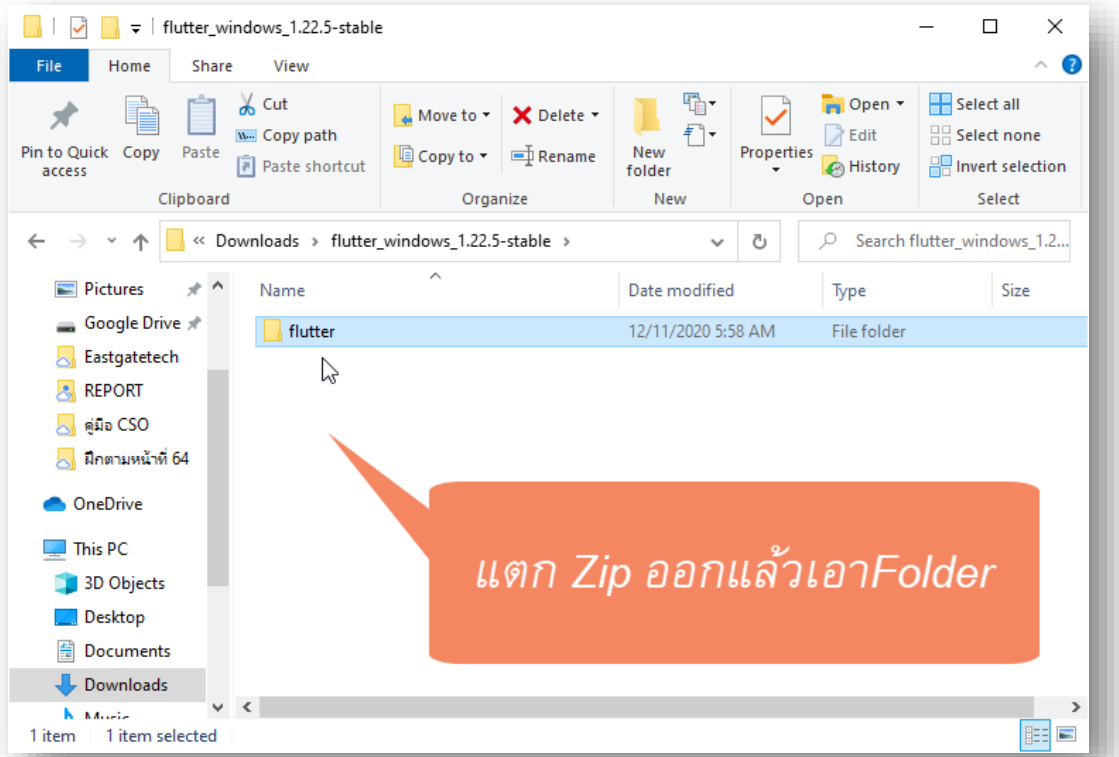


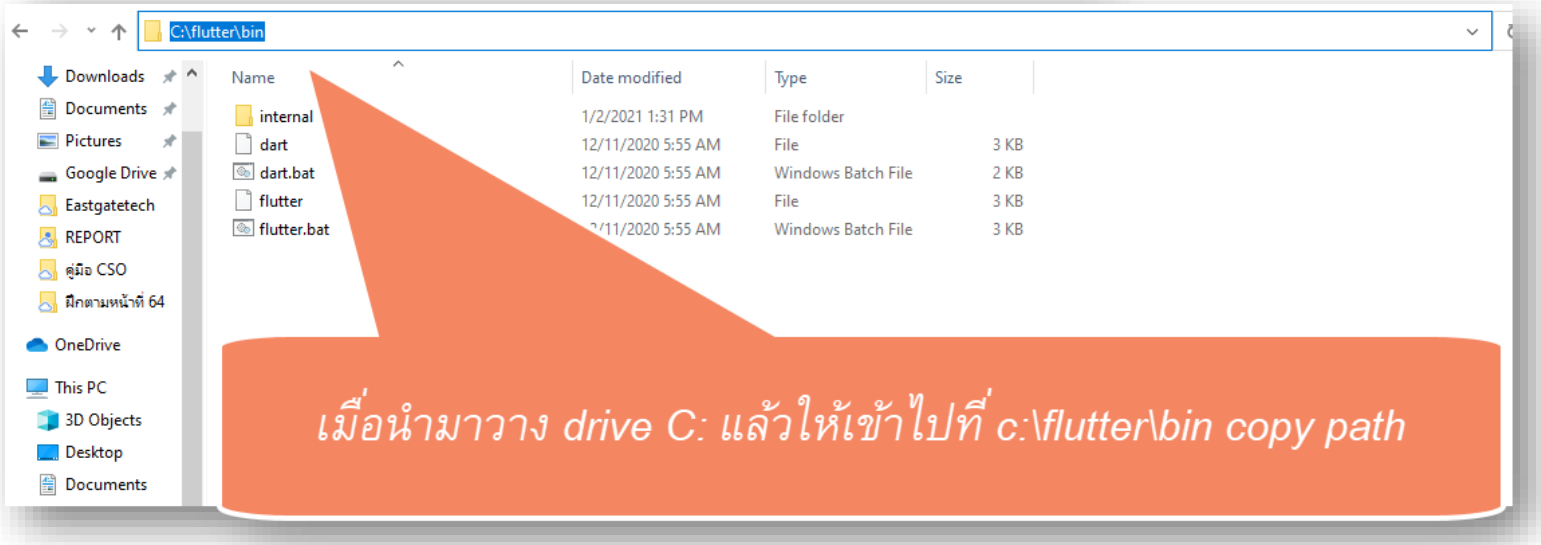


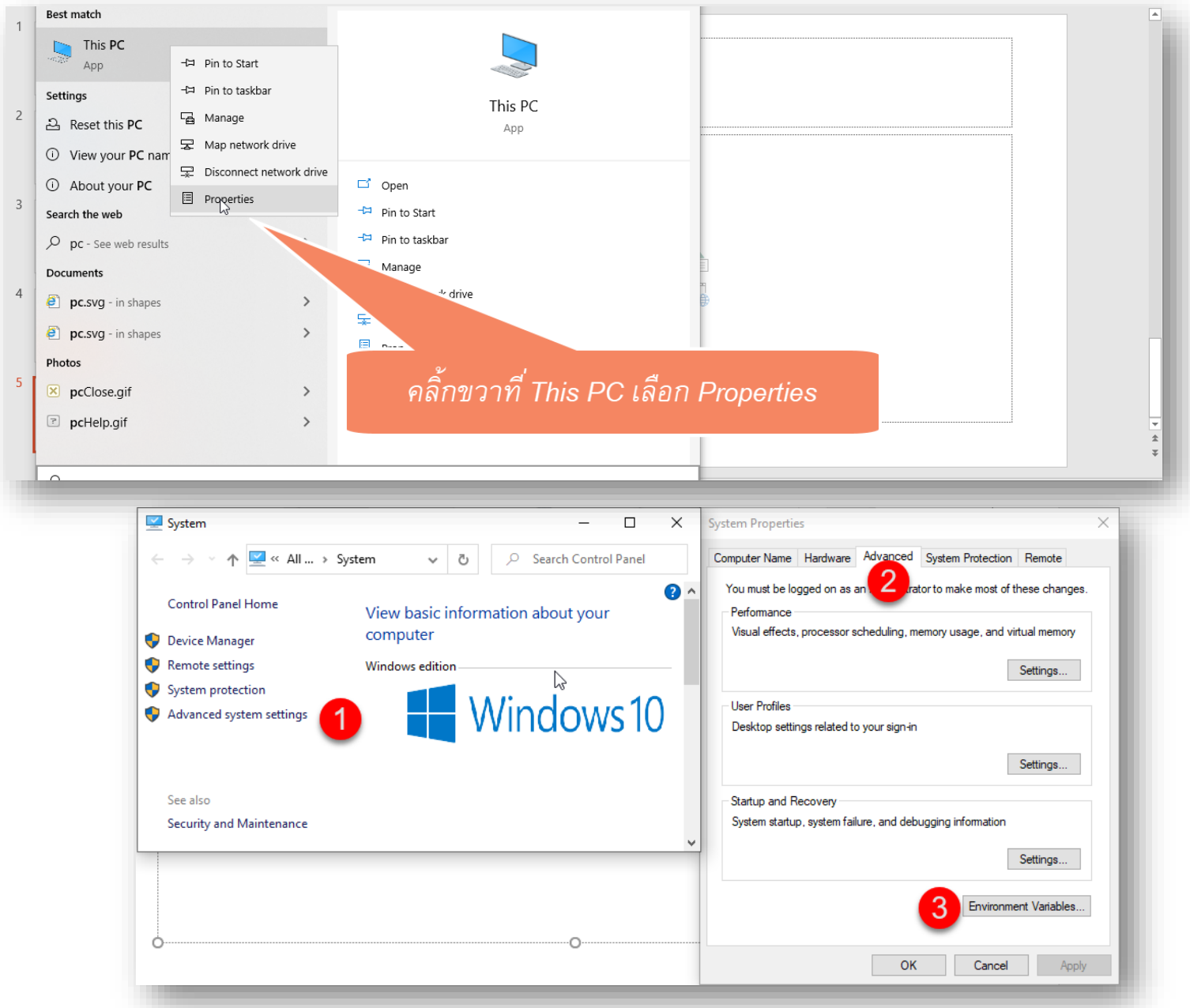
แล้วกด Get started


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
Set up an editor)
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
Flutter in China.
Set up an editor)
เลือก OS ที่ใช้พัฒนา
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 Matte

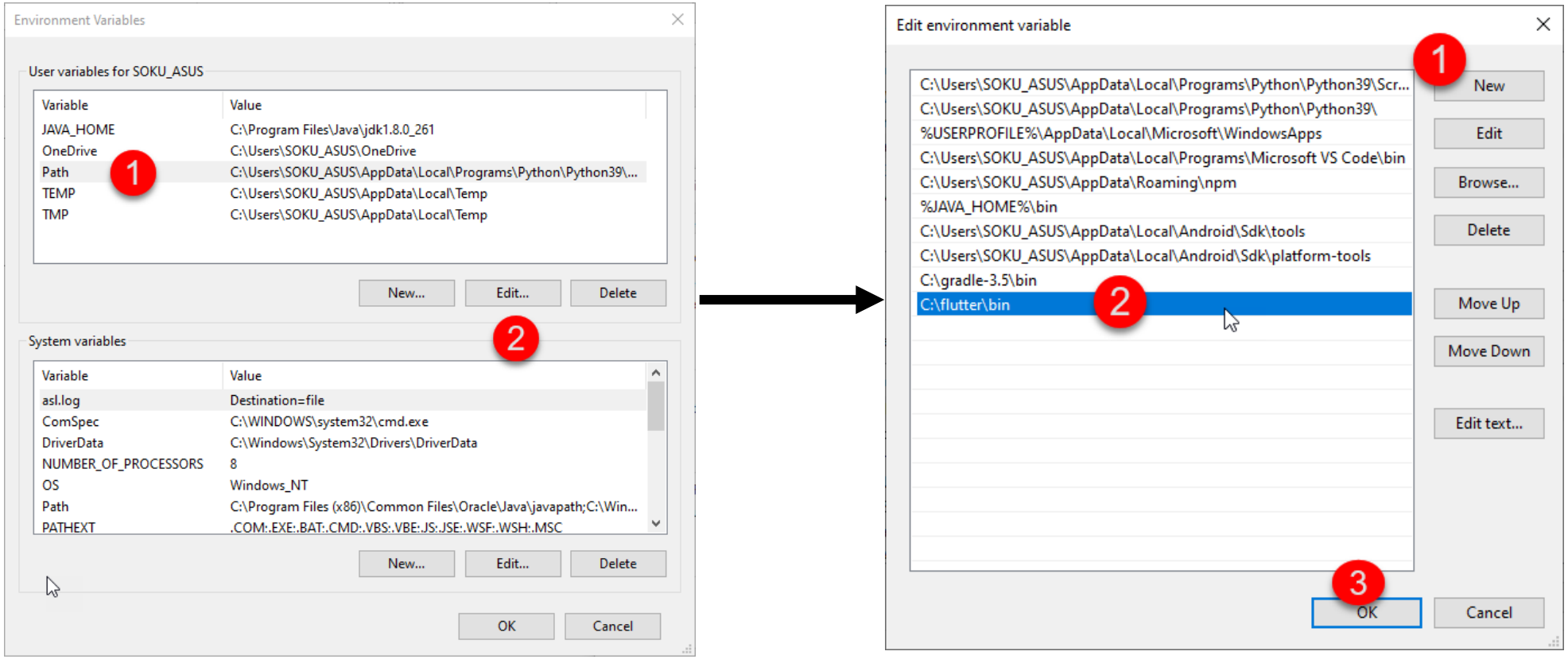

Flutter
Docs
Showcase
Community
Get started
Get started
1. Install
2. Set up an editor
3. Test drive
4. Write your first app
5. Learn more
From another platform?
Flutter for Android devs
Flutter for iOS devs
Flutter for React Native devs
Flutter for web devs
Flutter for Xamarin.Forms devs
Introduction to declarative UI
Dart language overview
Building a web app
Samples & tutorials
Development
User interface
Data & backend
Accessibility & internationalization
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.
If Git for Windows is already installed, make sure you can run git commands from the command prompt or PowerShell.
Get the Flutter SDK
1. Download the following installation bundle to get the latest stable release of the Flutter SDK:
flutter_windows_1.22.5-stable.zip
For other release channels, and older builds, see the
2. Extract the zip file and place the contained flutt
C:\src\flutter).
กด Download ที่นี่
Warning: Do not install Flutter in a directory like C:\Program Files\ that requires elevated privileges.
If you don't want to install a fixed version of the installation bundle, you can skip steps 1 and 2. Instead, get the source code from the Flutter repo on GitHub, and change branches or tags as needed. For example:
C:\src>git clone https://github.com/flutter/flutter.git -b stable
in the Flutter Console.
Contents
System requirements
Get the Flutter SDK
Update your path
Run flutter doctor
Android setup
Install Android Studio
Set up your Android device
Set up the Android emulator
Web setup
Next step
https://storage.googleapis.com/flutter_infra/releases/stable/windows/flutter_windows_1.22.5-stable.zip


flutter_windows_1.22.5-stable
File
Home
Share
View
Pin to Quick access
Copy
Paste
Cut
Copy path
Paste shortcut
Clipboard
Move to
Delete
Copy to
Rename
Organize
New folder
New
Properties
Open
Edit
History
Open
Select all
Select none
Invert selection
Select
Downloads › flutter_windows_1.22.5-stable ›
Search flutter_windows_1.2...
Pictures
Google Drive
Eastgatetech
REPORT
คู่มือ CSO
ฝึกตามหน้าที่ 64
OneDrive
This PC
3D Objects
Desktop
Documents
Downloads
Name
Date modified
Type
Size
flutter
12/11/2020 5:58 AM
File folder
แตก Zip ออกแล้วเอาFolder
1 item
1 item selected


C:\flutter\bin
Downloads
Documents
Pictures
Google Drive
Eastgatetech
REPORT
คู่มือ CSO
ฝึกตามหน้าที่ 64
OneDrive
This PC
3D Objects
Desktop
Documents
Name
Date modified
Type
Size
internal
1/2/2021 1:31 PM
File folder
dart
12/11/2020 5:55 AM
File
3 KB
dart.bat
12/11/2020 5:55 AM
Windows Batch File
2 KB
flutter
12/11/2020 5:55 AM
File
3 KB
flutter.bat
12/11/2020 5:55 AM
Windows Batch File
3 KB
เมื่อนำมาวาง drive C: แล้วให้เข้าไปที่ c:\flutter\bin copy path

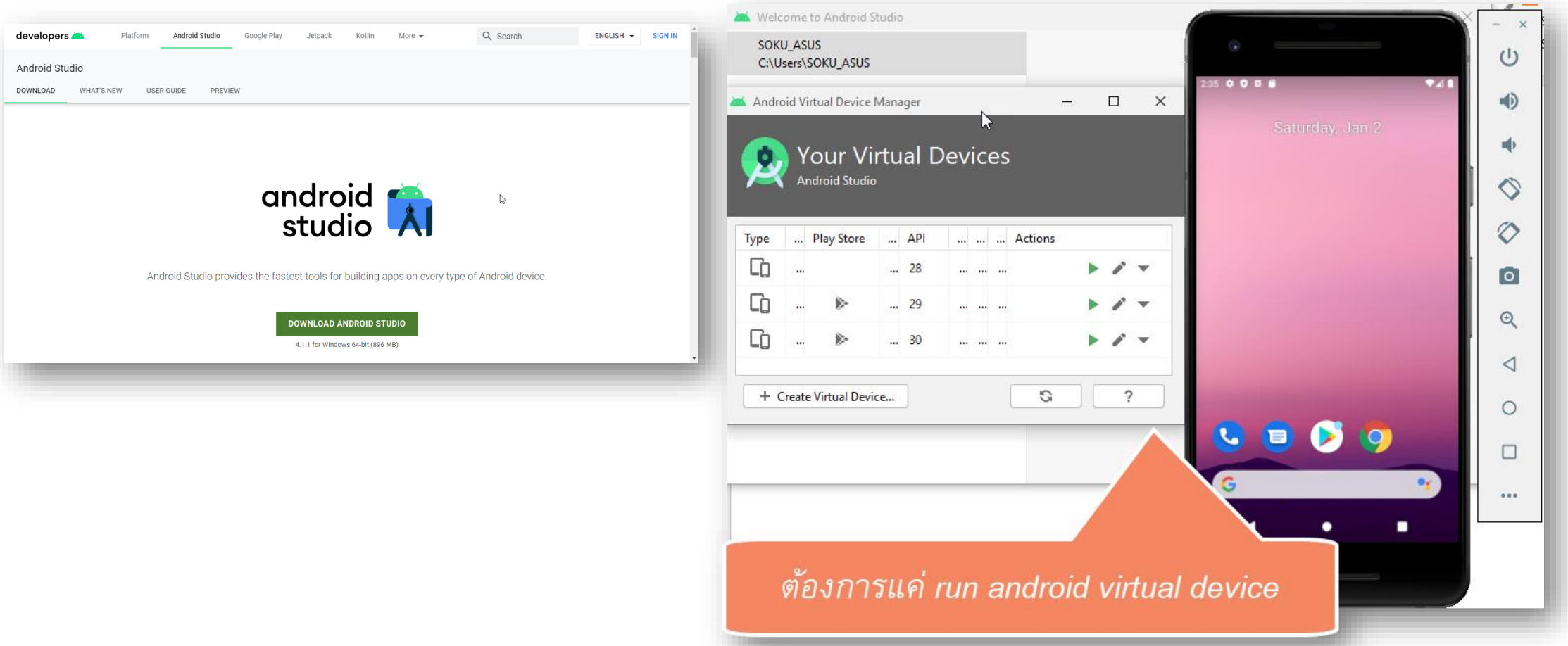


Advance System setting->Advance->Environment Variable


Environment Variables
User variables for SOKU_ASUS
Variable
Value
JAVA_HOME
C:\Program Files\Java\jdk1.8.0_261
OneDrive
C:\Users\SOKU_ASUS\OneDrive
Path
C:\Users\SOKU_ASUS\AppData\Local\Programs\Python\Python39\...
TEMP
C:\Users\SOKU_ASUS\AppData\Local\Temp
TMP
C:\Users\SOKU_ASUS\AppData\Local\Temp
1
New...
Edit...
Delete
2
System variables
Variable
Value
asl.log
Destination=file
ComSpec
C:\WINDOWS\system32\cmd.exe
DriverData
C:\Windows\System32\Drivers\DriverData
NUMBER_OF_PROCESSORS
8
OS
Windows_NT
Path
C:\Program Files (x86)\Common Files\Oracle\Java\javapath;C:\Win...
PATHEXT
.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH;.MSC
New...
Edit...
Delete
OK
Cancel
Edit environment variable
C:\Users\SOKU_ASUS\AppData\Local\Programs\Python\Python39\Scr...
C:\Users\SOKU_ASUS\AppData\Local\Programs\Python\Python39\
%USERPROFILE%\AppData\Local\Microsoft\WindowsApps
C:\Users\SOKU_ASUS\AppData\Local\Programs\Microsoft VS Code\bin
C:\Users\SOKU_ASUS\AppData\Roaming\npm
%JAVA_HOME%\bin
C:\Users\SOKU_ASUS\AppData\Local\Android\Sdk\tools
C:\Users\SOKU_ASUS\AppData\Local\Android\Sdk\platform-tools
C:\gradle-3.5\bin
C:\flutter\bin
1
2
3
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel

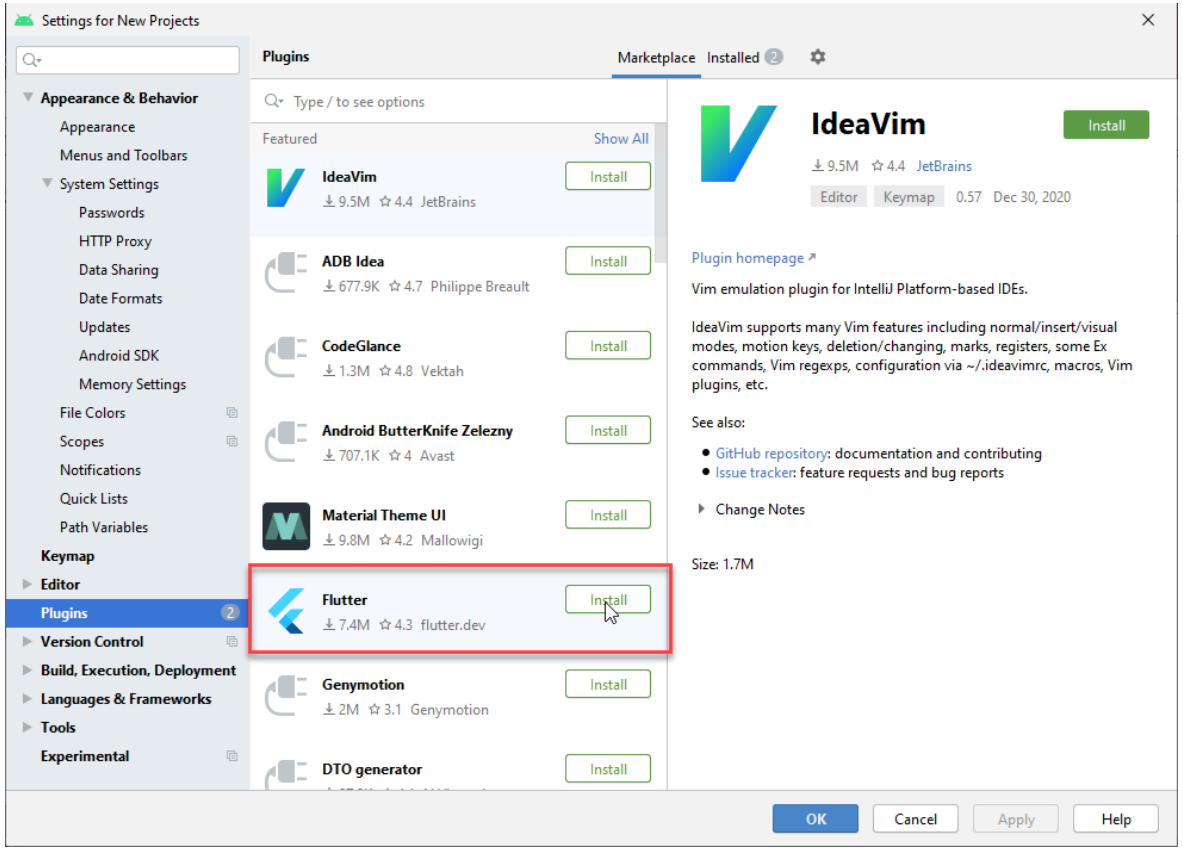
2 Install Android Studio

- https://developer.android.com/studio ดาวน์โหลดแล้ว install

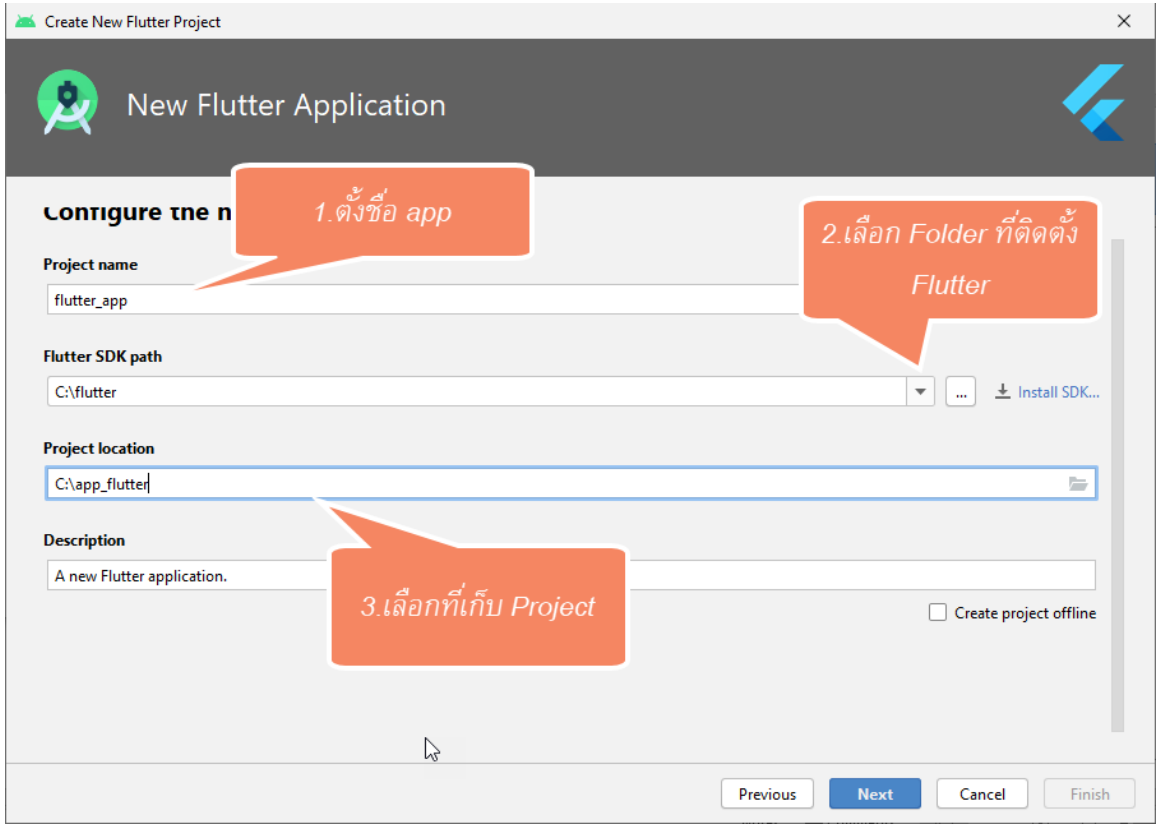


การสร้าง Project บน Android Studio

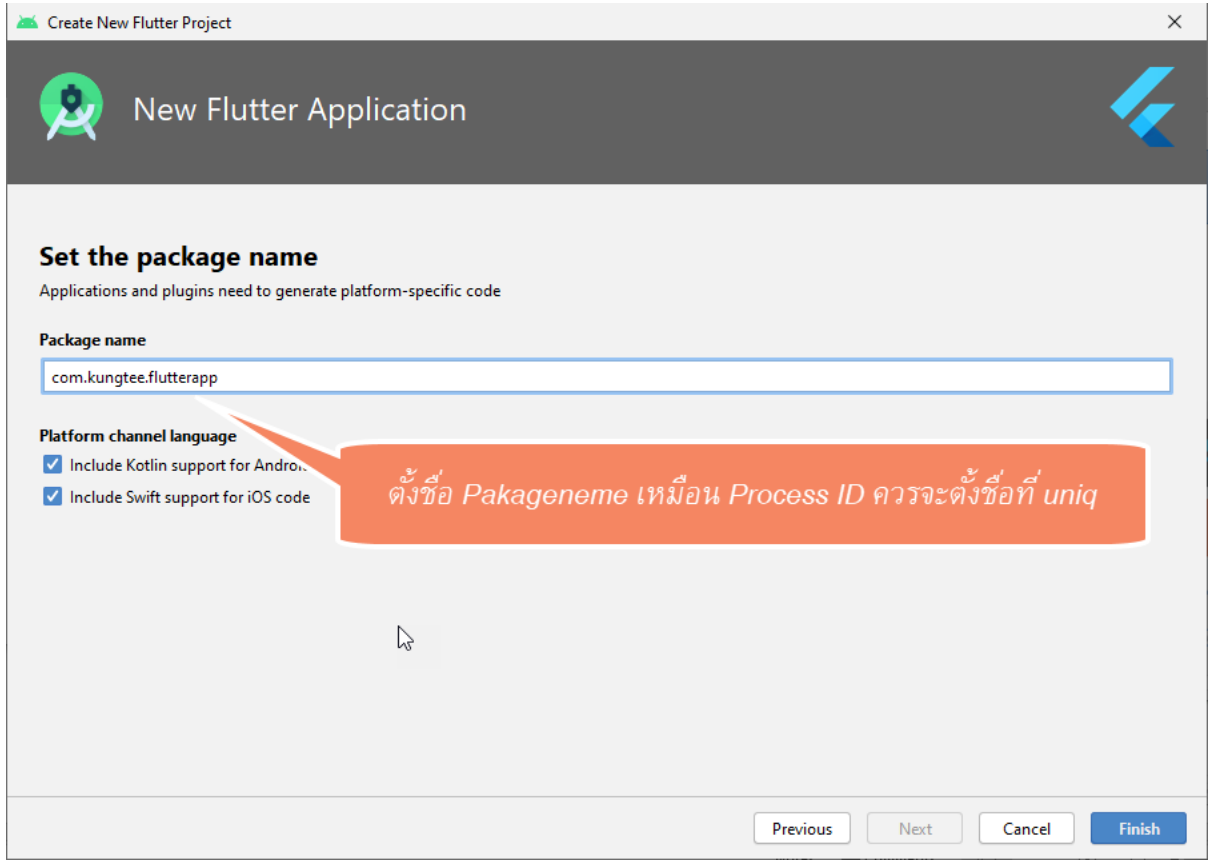
- เปิด Android Studio ขึ้นมาแล้วไป Config->Setting->plugin ทำการ Install Flutter

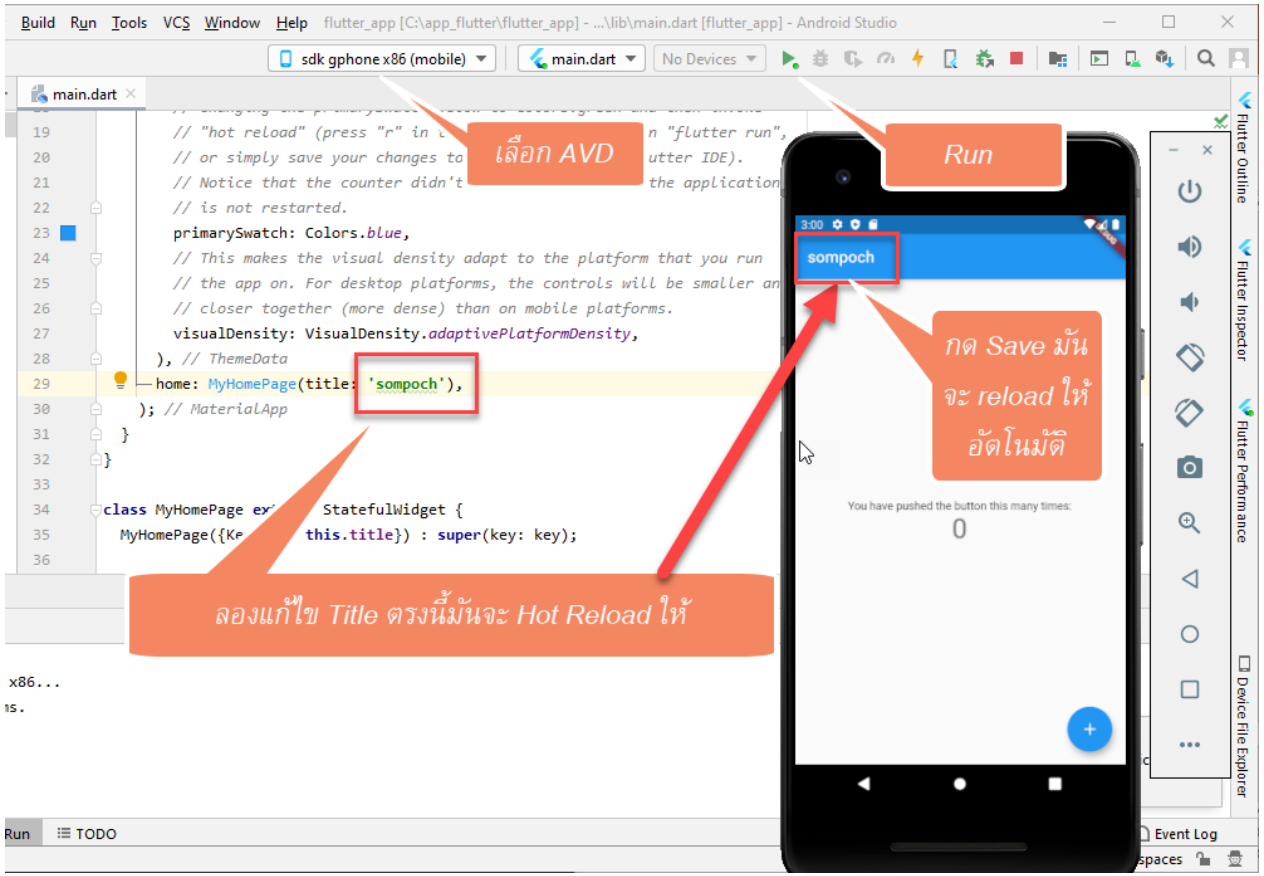


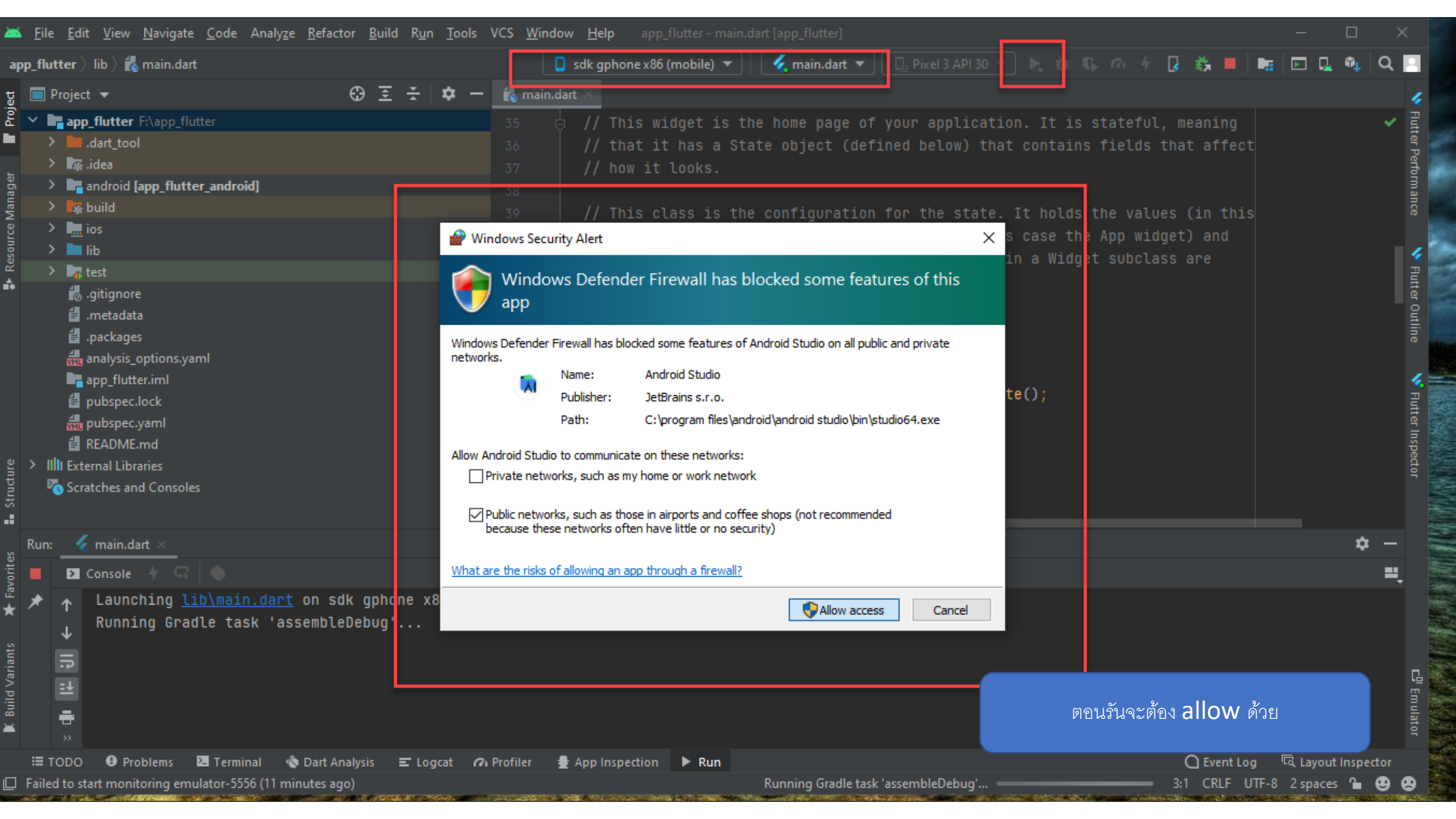
หลังจากนั้นให้ New Project เลือก New Flutter Application แล้วตั้งค่าตามภาพด้านล่าง



การสร้าง Project บน Android Studio

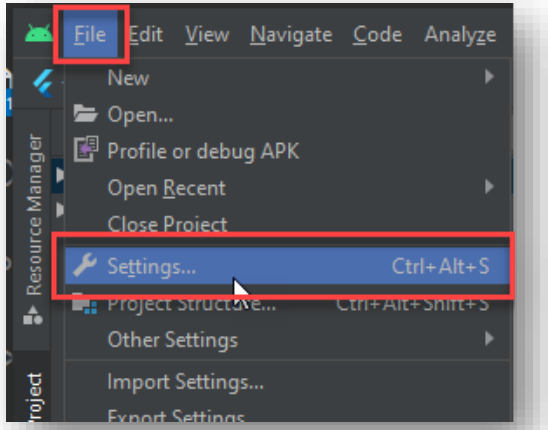


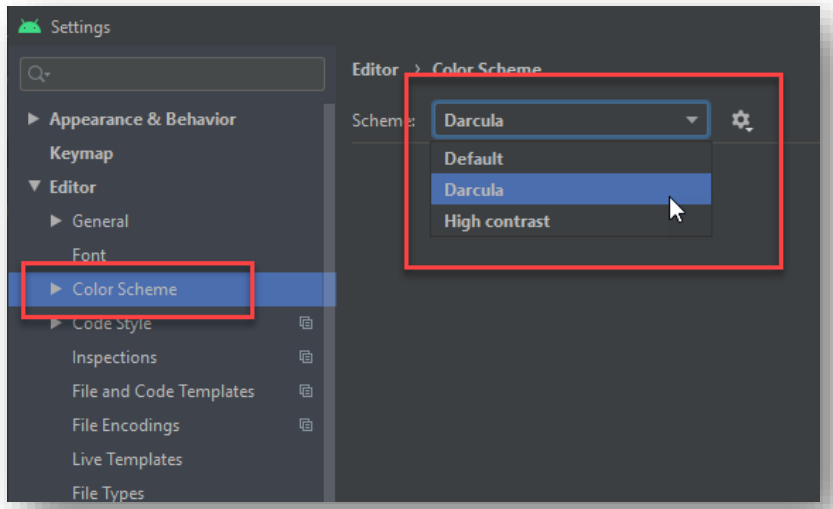


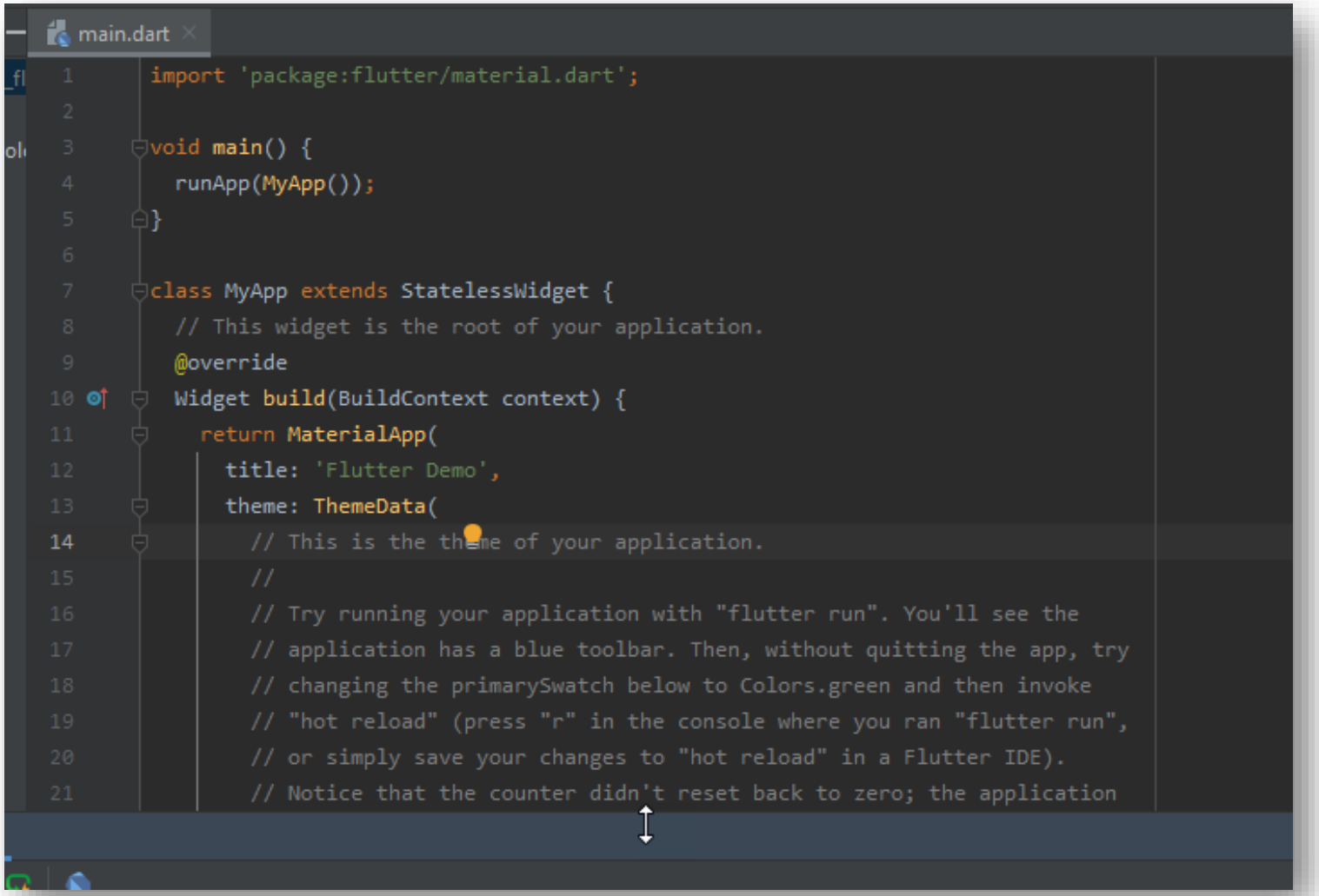

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help
app_flutter - main.dart [app_flutter]
app_flutter lib main.dart
sdk gphone x86 (mobile)
main.dart
Pixel 3 API 30
Project
app_flutter F:\app_flutter
.dart_tool
.idea
android [app_flutter_android]
build
ios
lib
test
.gitignore
.metadata
.packages
analysis_options.yaml
app_flutter.iml
pubspec.lock
pubspec.yaml
README.md
External Libraries
Scratches and Consoles
// This widget is the home page of your application. It is stateful, meaning
// that it has a State object (defined below) that contains fields that affect
// how it looks.
// This class is the configuration for the state. It holds the values (in this
Windows Security Alert
Windows Defender Firewall has blocked some features of this app
Windows Defender Firewall has blocked some features of Android Studio on all public and private networks.
Name: Android Studio
Publisher: JetBrains s.r.o.
Path: C:\program files\android\android studio\bin\studio64.exe
Allow Android Studio to communicate on these networks:
Private networks, such as my home or work network
Public networks, such as those in airports and coffee shops (not recommended because these networks often have little or no security)
What are the risks of allowing an app through a firewall?
Allow access
Cancel
Run: main.dart
Console
Launching lib\main.dart on sdk gphone x8
Running Gradle task 'assembleDebug'...
ตอนรันจะต้อง allow ด้วย
TODO Problems Terminal Dart Analysis Logcat Profiler App Inspection Run
Event Log Layout Inspector
Failed to start monitoring emulator-5556 (11 minutes ago)
Running Gradle task 'assembleDebug'...
3:1 CRLF UTF-8 2 spaces

Android Studio เปลี่ยนสี theme ให้เป็นสีดำเพื่อจะเขียนโค้ดสบายตา

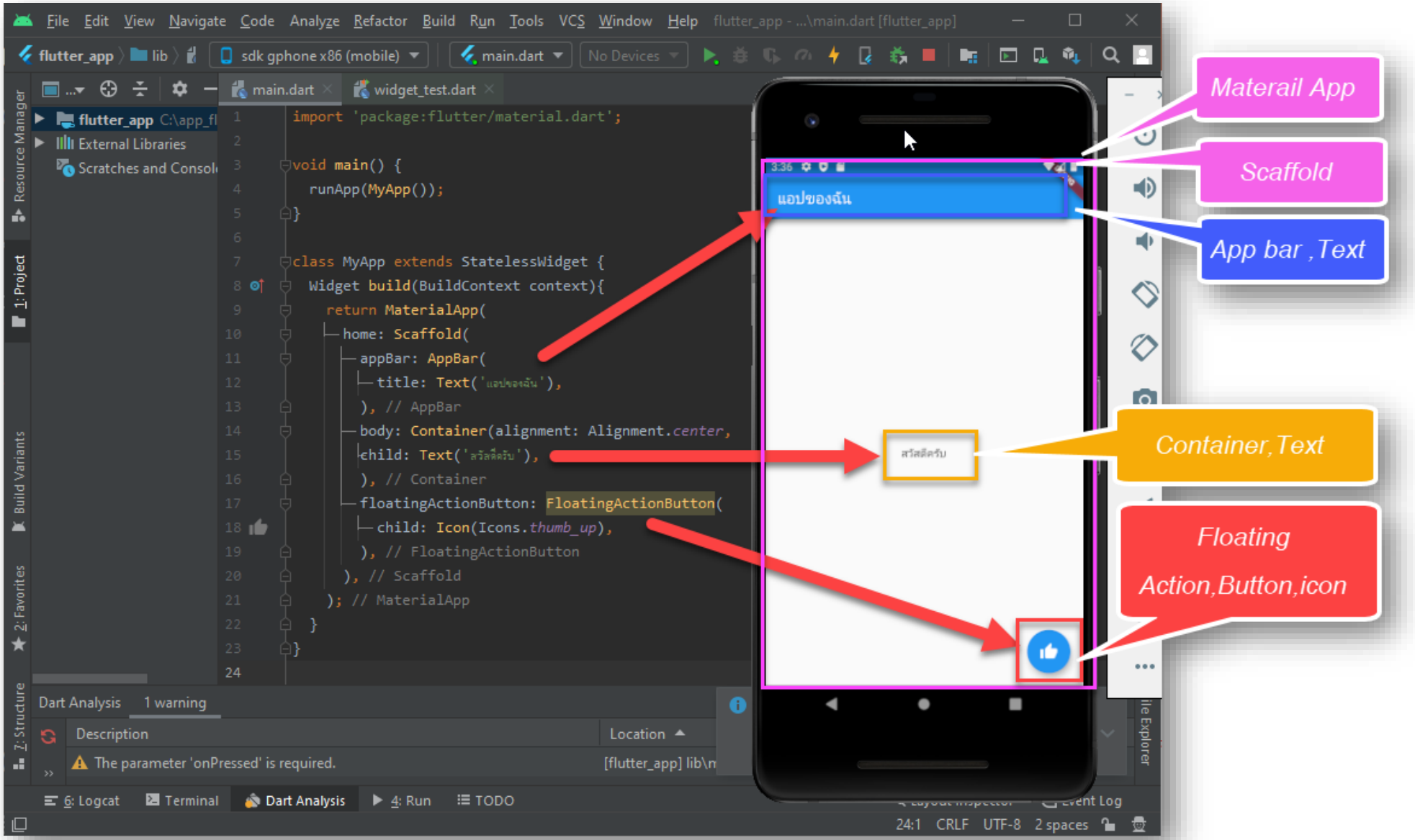
- ไปที่เมนู File->Settings->Color Schema->Darcula (บางเวอรชั่นจะอยู่ในเมนู Appearance->Theme)



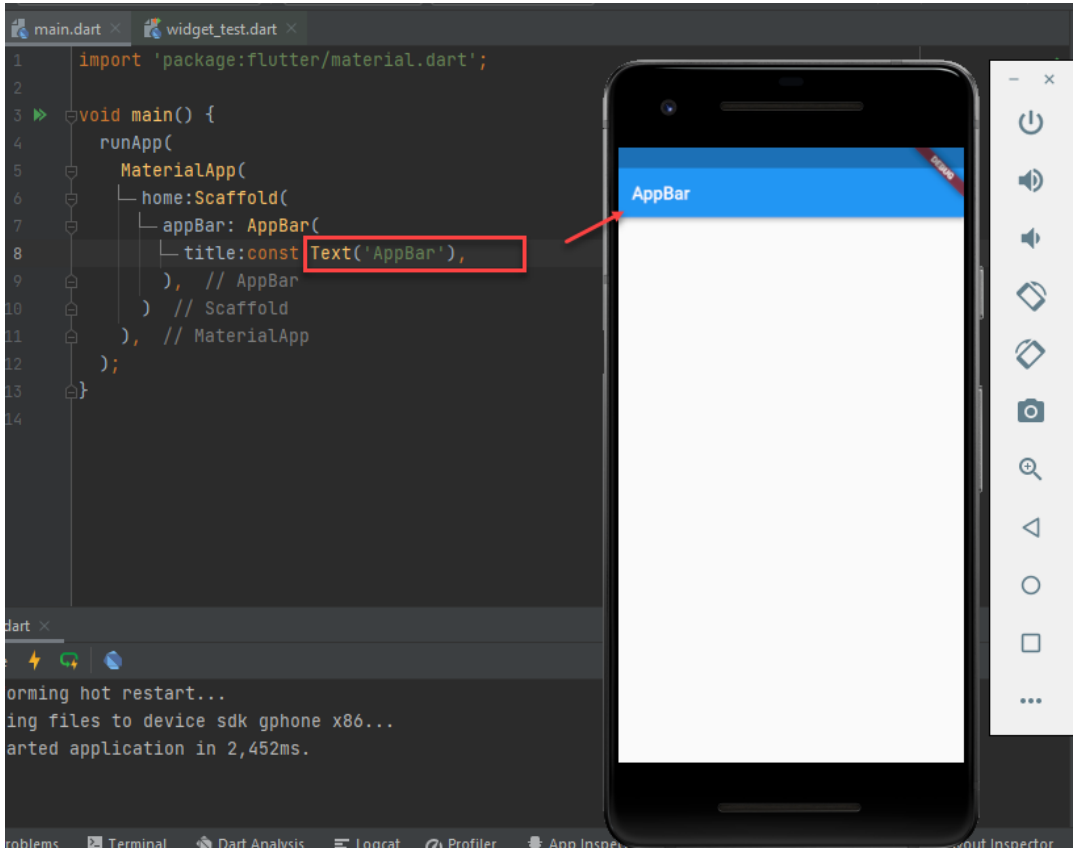




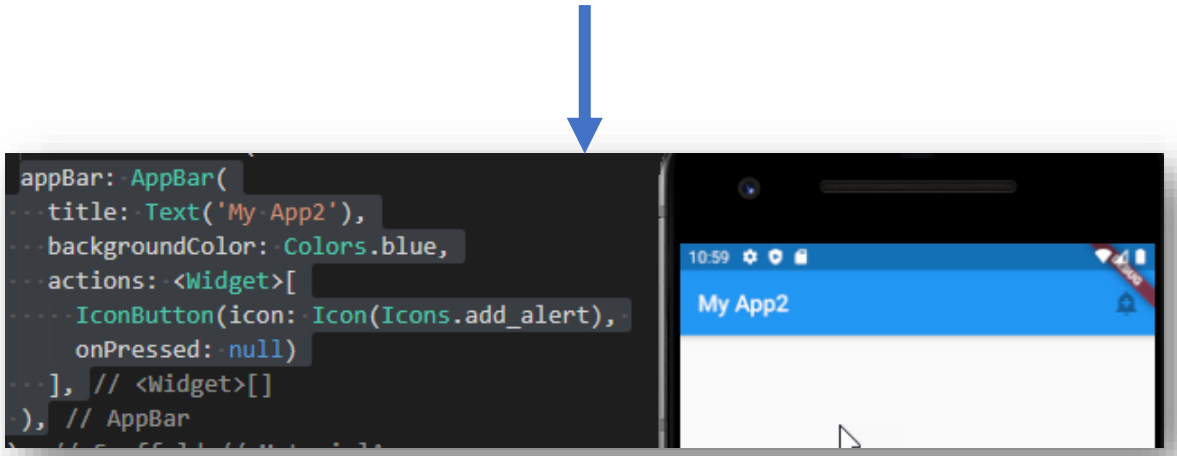
ตัวอย่างแรกสร้าง Appbar สร้าง Content สร้างปุ่มกด Like



Scaffold() สร้าง AppBar



ใน Appbar จะมี Properties ให้แก้ไขได้เยอะมาก เช่น ใส่ข้อความ เปลี่ยนสี BackGround ใส่ icon

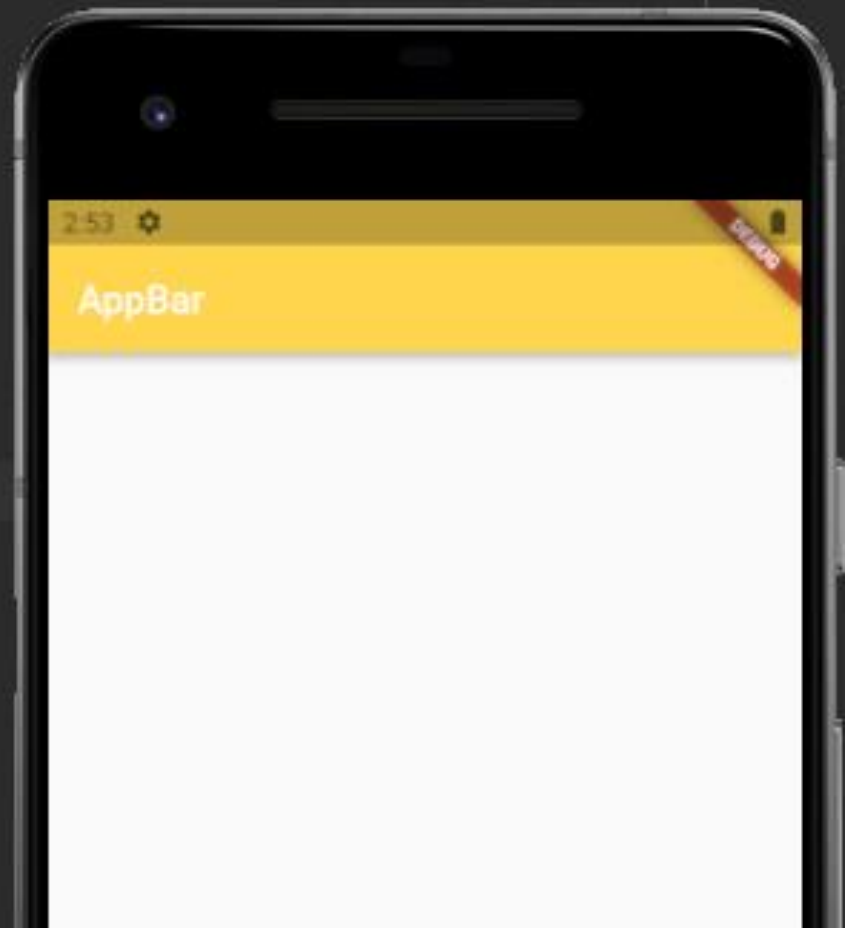


- https://api.flutter.dev/flutter/material/AppBar-class.html
- สามารถดูสีได้ที่ https://material.io/design/color/the-color-system.html#tools-for-picking-colors

Body Color

main.dart | widget_test.dart

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

void main() {
  runApp(
    MaterialApp(
      home:Scaffold(
        appBar: AppBar(
          title:const Text('AppBar'),
          backgroundColor: Colors.amber[300],
        ),  // AppBar
      )  // Scaffold
    ),  // MaterialApp
  );
}
```

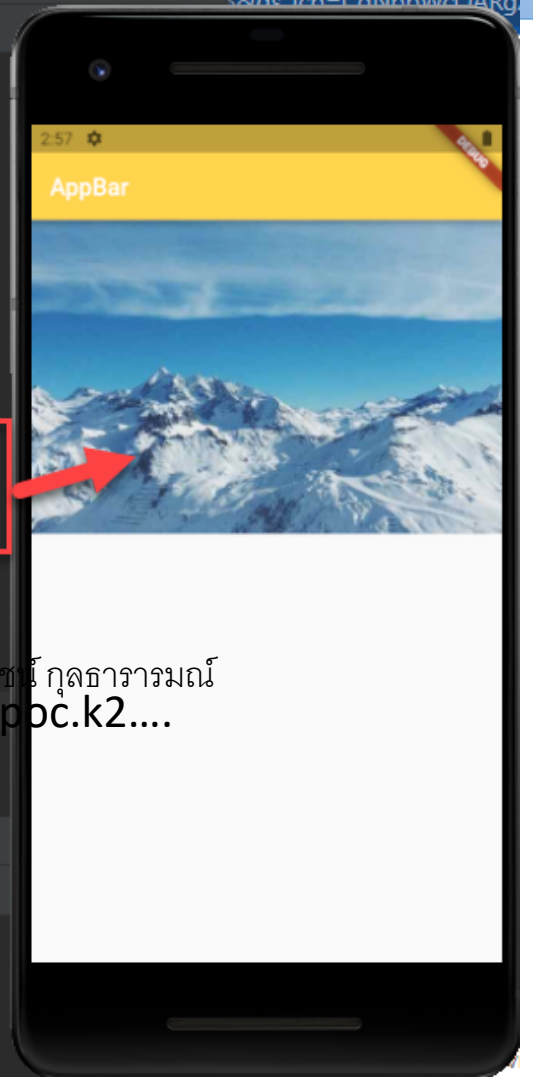


Body การแทรก Image

main.dart widget_test.dart

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

void main() {
  runApp(
    MaterialApp(
      home:Scaffold(
        appBar: AppBar(
          title:const Text('AppBar'),
          backgroundColor: Colors.amber[300],
        ),  // AppBar
        body:const Image(
          image:NetworkImage('https://www.w3schools.com/w3css/img_snowtops.jpg')
        )  // Image
      )  // Scaffold
    ),  // MaterialApp
  );
}
```

```
forming hot restart...
cing files to device sdk gphone x86...
tarted application in 2,979ms.
```



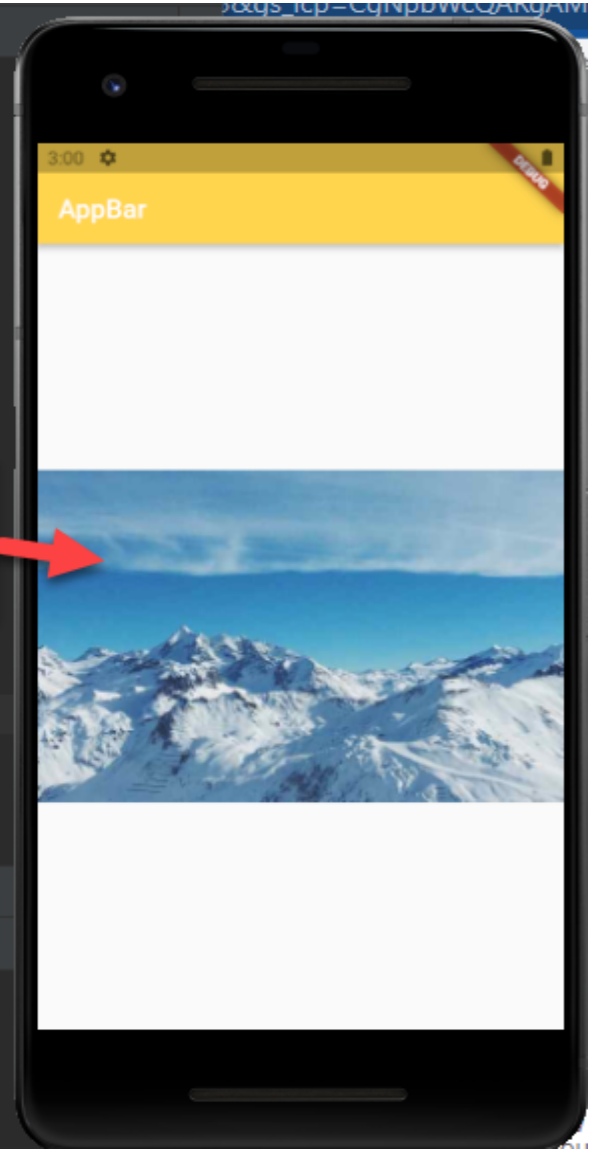
Name : สมโภชน์ กุลธารามณ์
email : sompoc.k2....

Body Image ถ้าต้องการให้อยู่กึงกลางจะต้องใช้ Center Widget แล้วสร้าง Child

main.dart | widget_test.dart

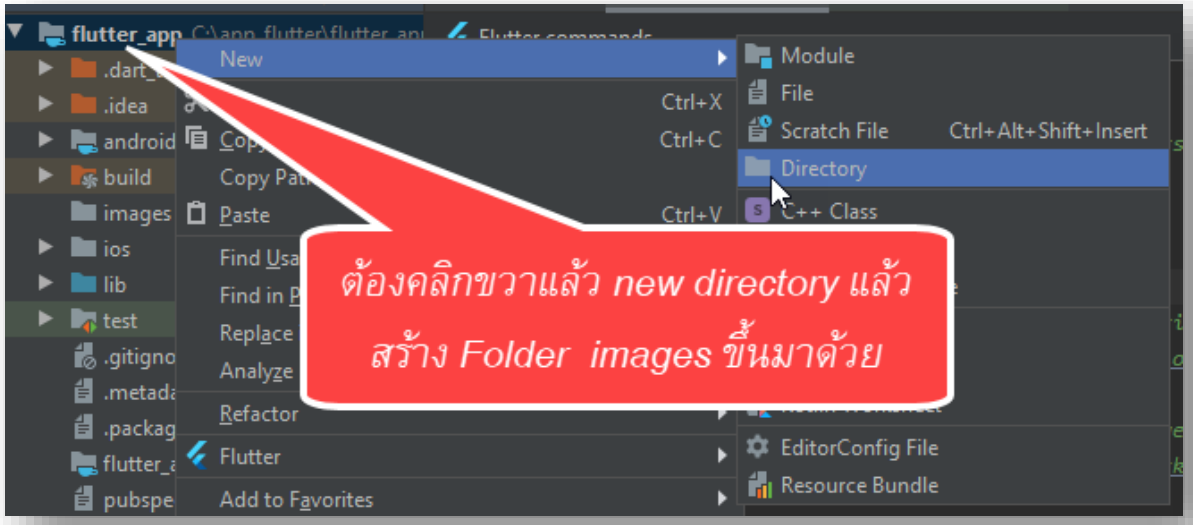
```dart
import 'package:flutter/material.dart';

void main() {
  runApp(
    MaterialApp(
      home:Scaffold(
        appBar: AppBar(
          title:const Text('AppBar'),
          backgroundColor: Colors.amber[300],
        ),  // AppBar
        body:const Center(child: Image(
            image:NetworkImage('https://www.w3schools.com/w3css/img_snowtops.jpg')
            )  // Image
        )  // Center
      )  // Scaffold
    ),  // MaterialApp
  );
}
```

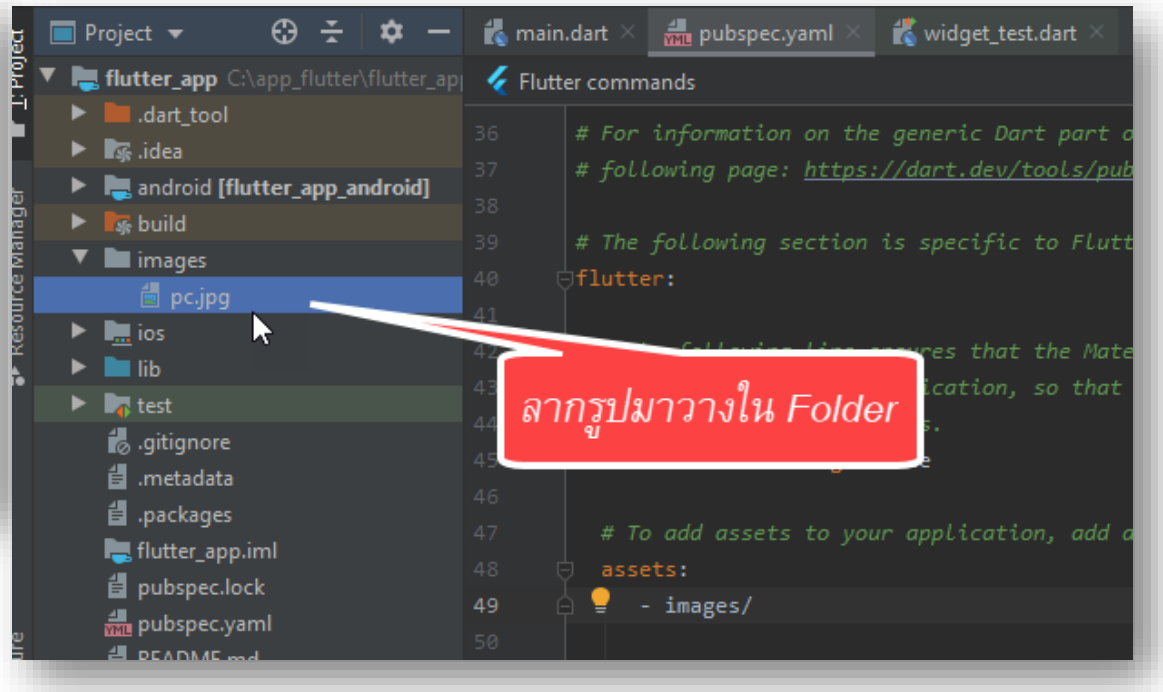
```
forming hot restart...
cing files to device sdk gphone x86...
tarted application in 2,679ms.
```



กรณีที่ต้องการเอารูปภาพมาใส่ไว้ใน app เลย

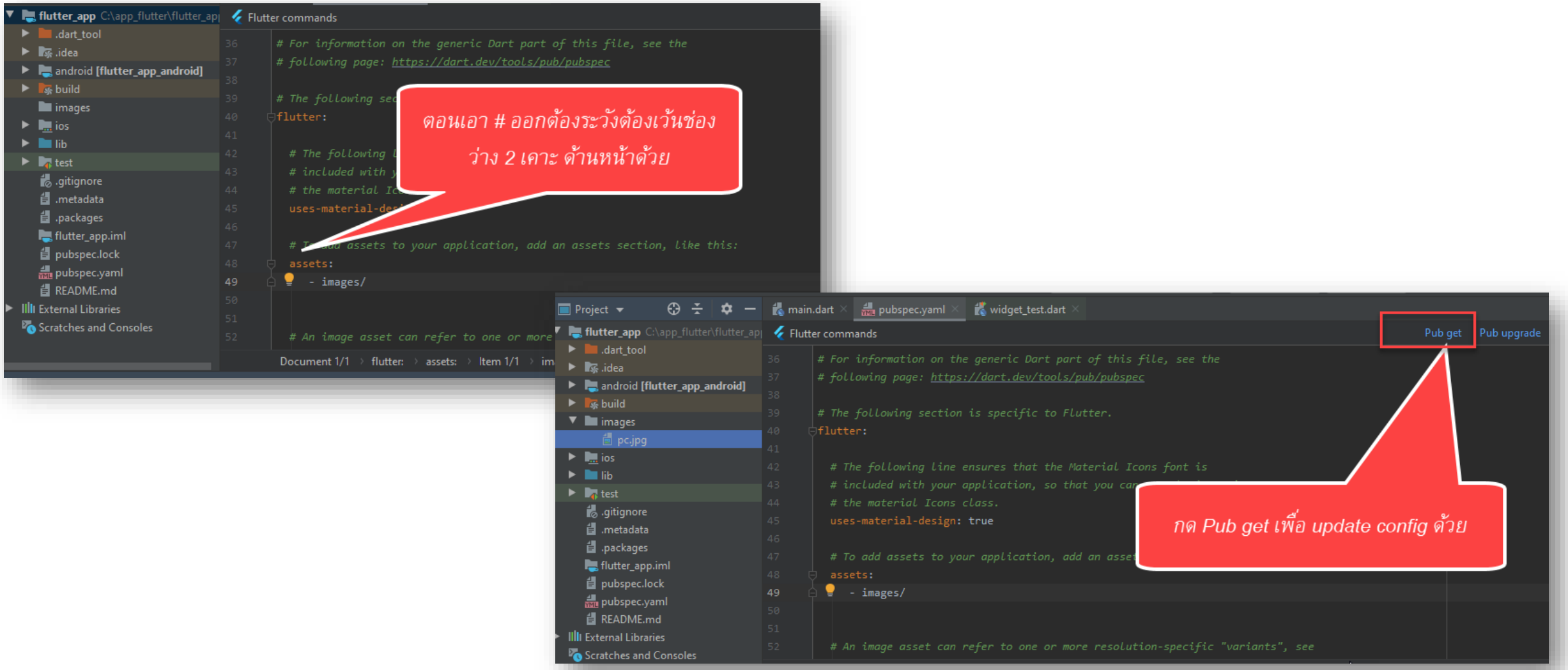
- ในกรณีนี้จะต้องไปเปิดใช้งานในไฟล์ pubspec.yaml (YAML เป็น markup language คล้ายๆ ภาษา html ที่สำหรับใช้ config เป็นภาษาที่มนุษย์เข้าใจง่าย แต่เครื่องคอมพิวเตอร์จะต้องแปลงก่อน) ไปเปิด comment ในส่วนของ asset เพื่อให้โปรแกรมรู้จัก และกด pub get เพื่อให้ update ค่า config





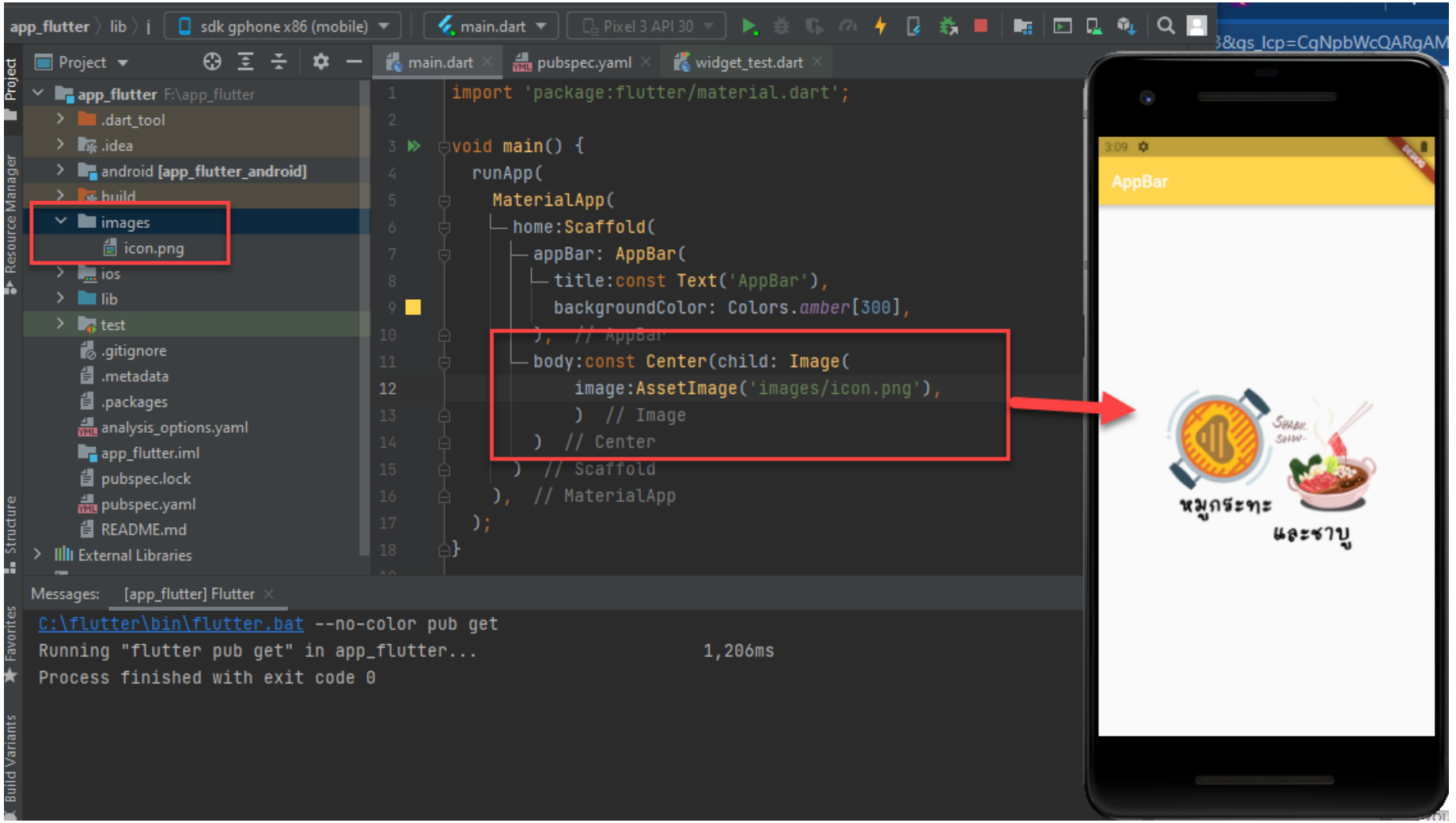
กรณีที่ต้องการเอารูปภาพมาใส่ไว้ใน app เลย (ต่อ)

- เอา # ออก เพื่อเปิดใช้งาน



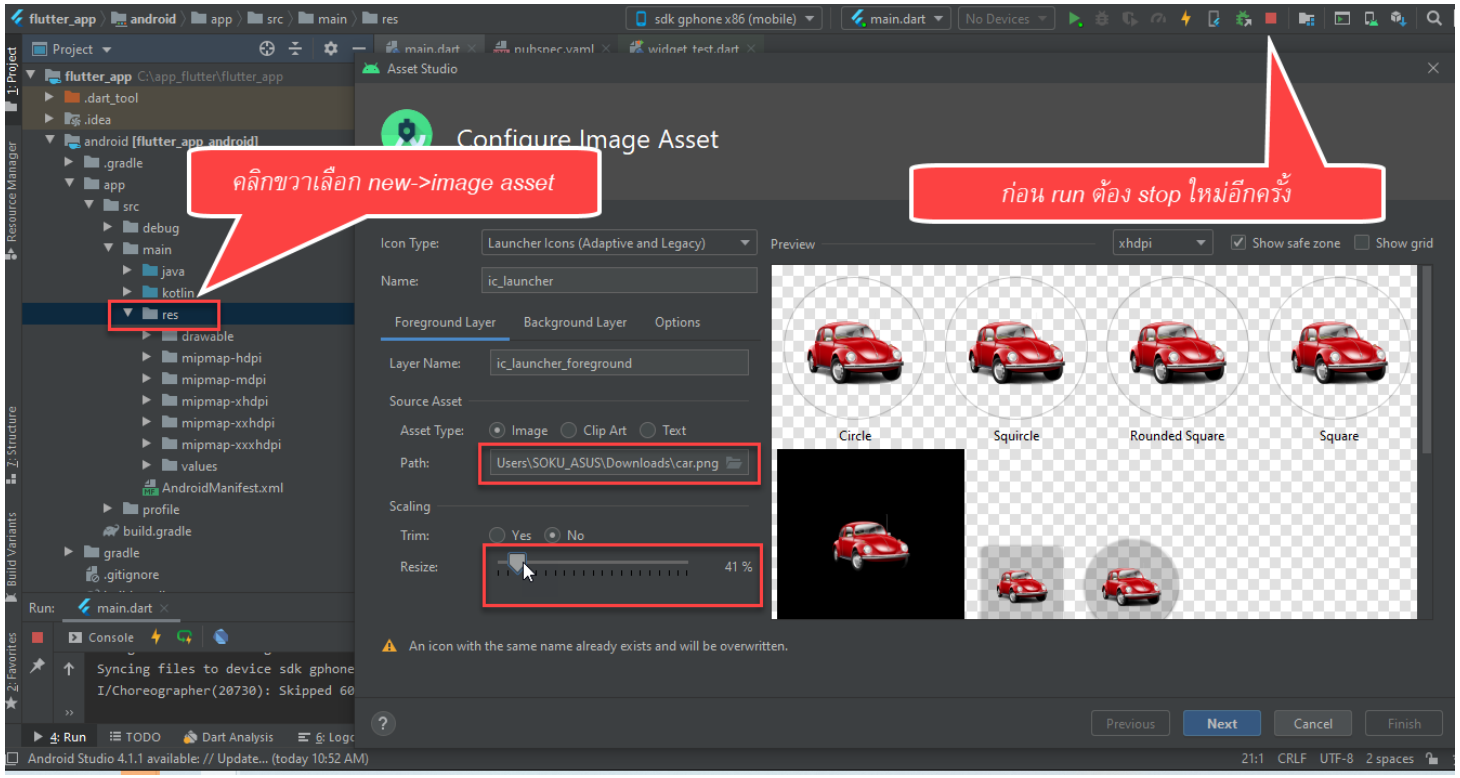
กรณีที่ต้องการเอารูปภาพมาใส่ไว้ใน app เลย (ต่อ)

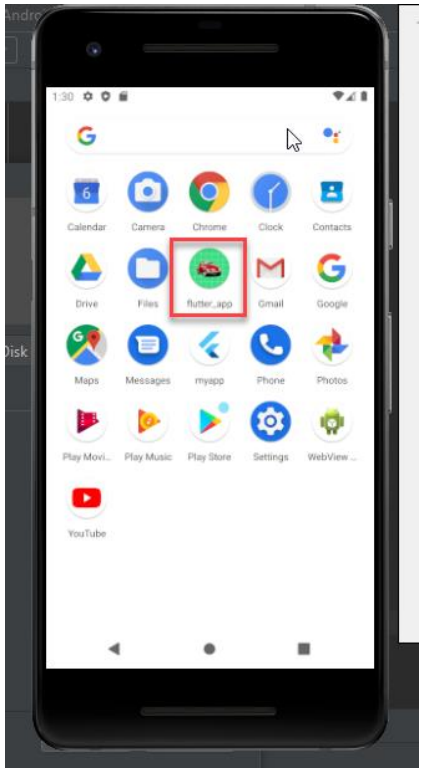
- กลับมาแก้ไขโค้ดในไฟล์ lib/main.dart



การสร้าง icon บน ios และ android

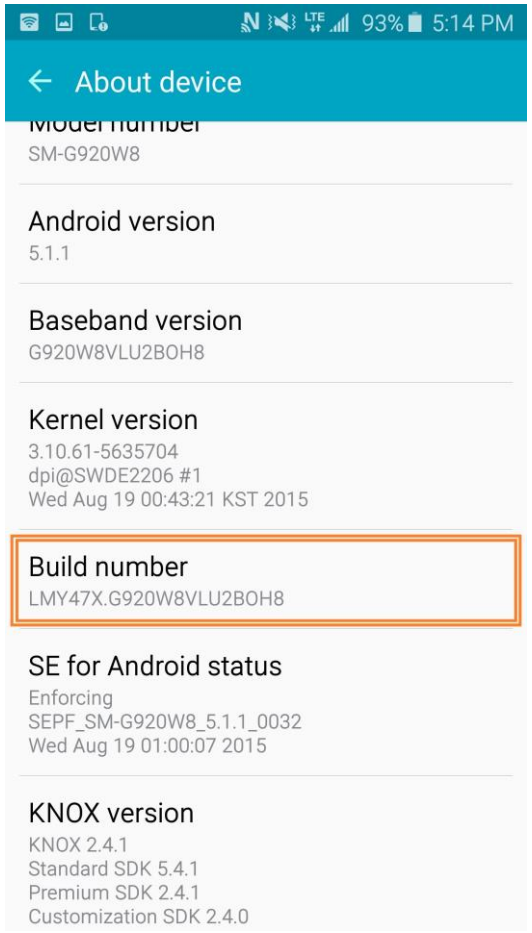
- หารูป icon ที่ต้องการหรือไปดาวน์โหลดมาจากเว็บ iconarchive นามสกุล png
- ใน android studio เลือก android->app->src->main->res แล้วคลิ้กขวาที่ res เลือก new เลือก image assets
- ให้เลือกไฟล์ icon ที่จะสร้าง แล้วปรับขนาดดูภาพจากช่องด้านขวา แล้วก็กด next -> finish แล้วรอง run ดู
- ถ้าหา image asset ไม่เจอจะต้อง file-open เลือกโปรเจ็คแล้วเลือกไปด้านในให้ถึง app->android แทน

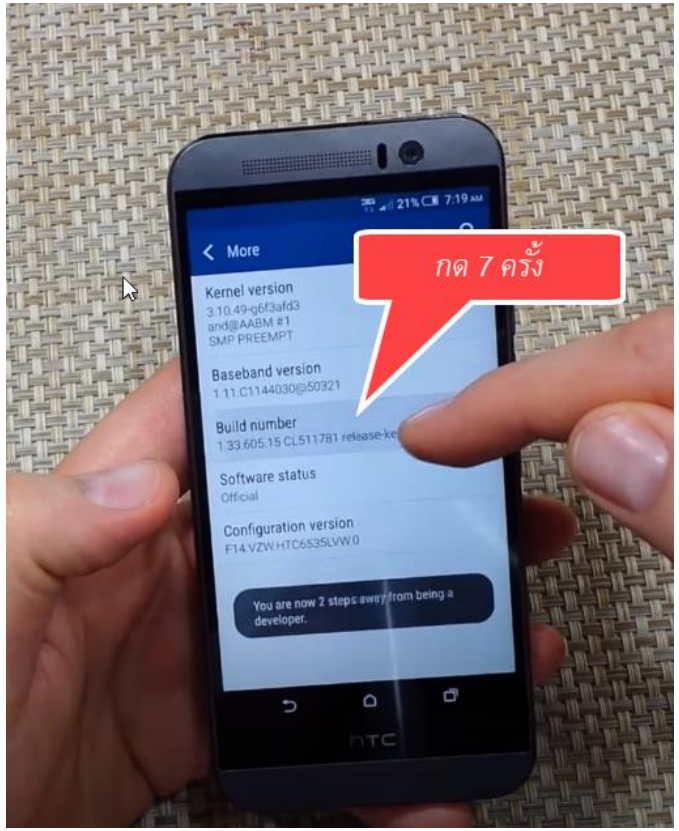


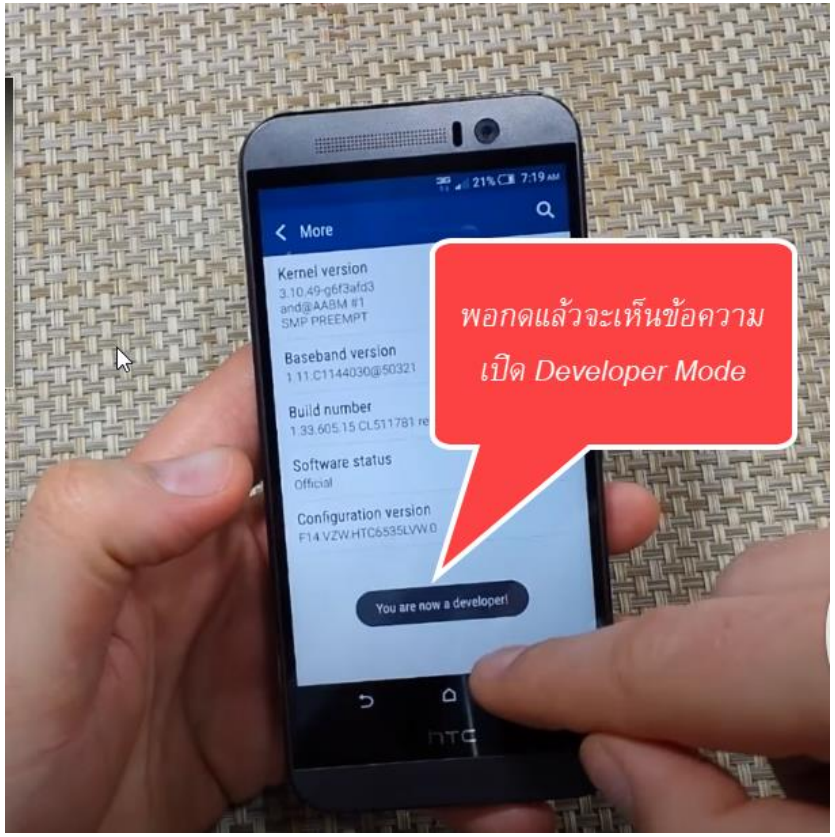


การ Deploy app สู่ physical android device

- ต้องนำเครื่องมือถือ android เข้าไปใน setting ไปหา build number กด 7 ครั้งเพื่อเปิด Developer Mode เพื่อให้อุปกรณ์ android สามารถเสียบสาย usb แล้ว debug app ได้ เพื่อจะได้ลงแอพที่ทำจากโปรแกรมเราได้



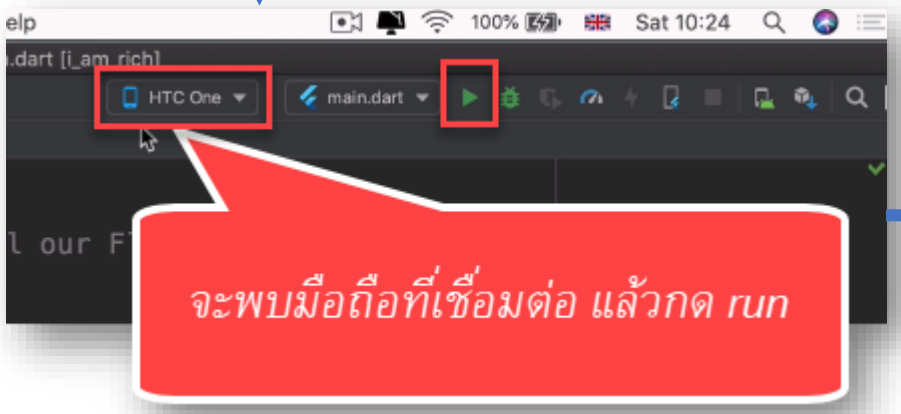


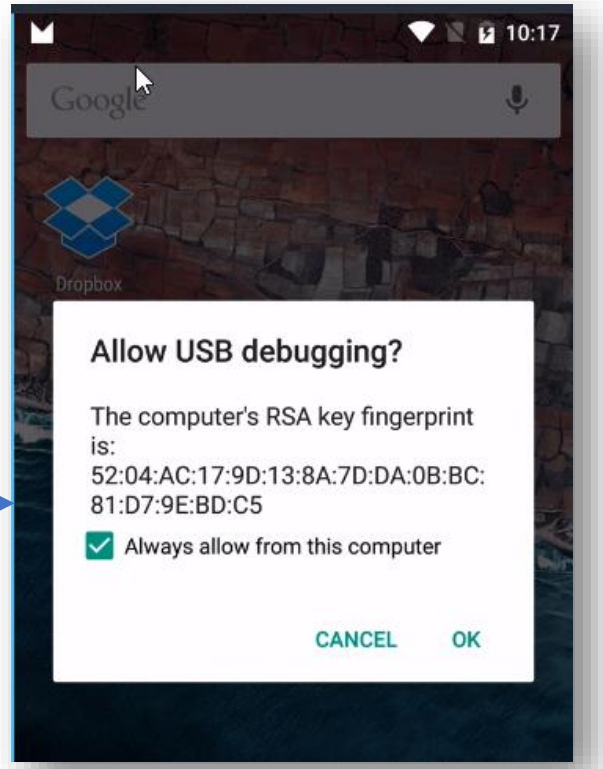


การ Deploy app สู่ physical android device (ต่อ)

Steps

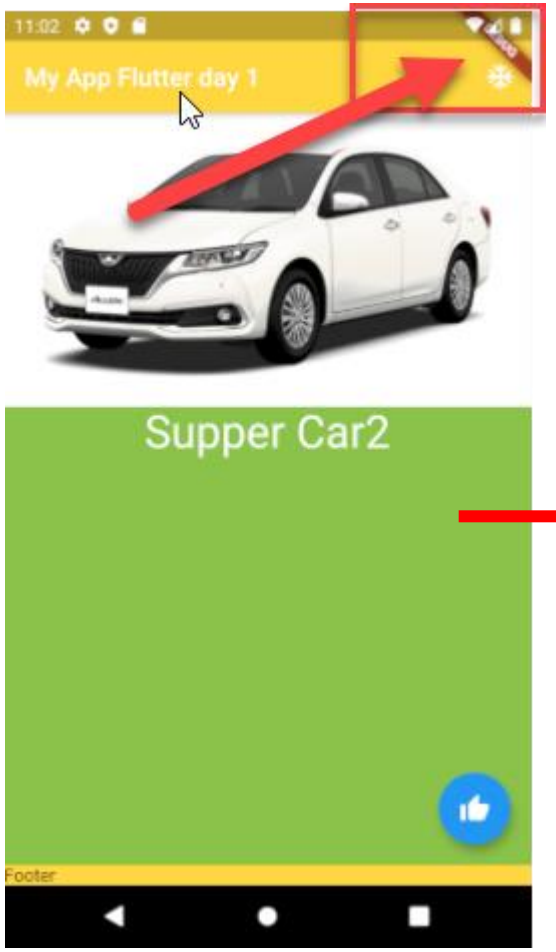
1. Enable Developer Mode (Phone)
2. Enable USB Debugging (Phone)
3. Connect Your Phone with USB
4. Trust Your Computer if Prompted (Phone)





หลังจากนั้น กด run ใน android studio ที่เชื่อมกับมือถือแล้ว ในหน้าจอมือถือจะแสดงข้อความให้กด allow usb debugging

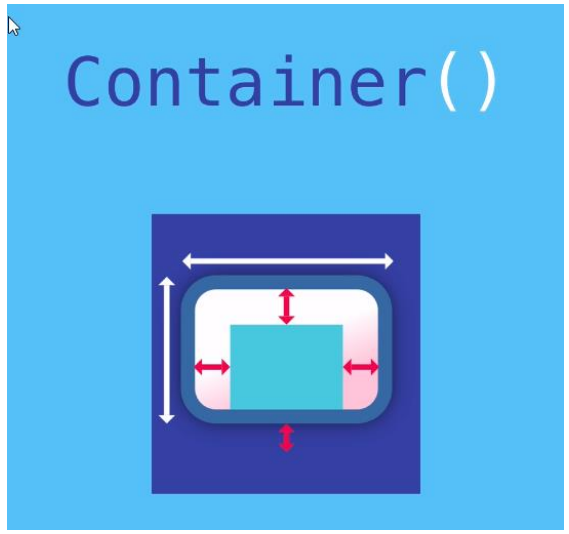
การเอา Icon Debug Banner ออกจากโค้ด Flutter

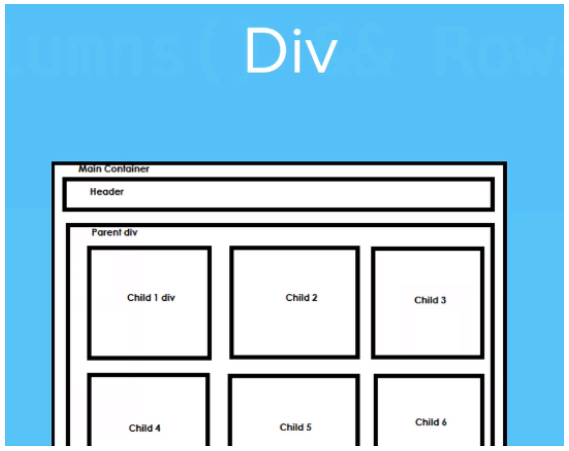


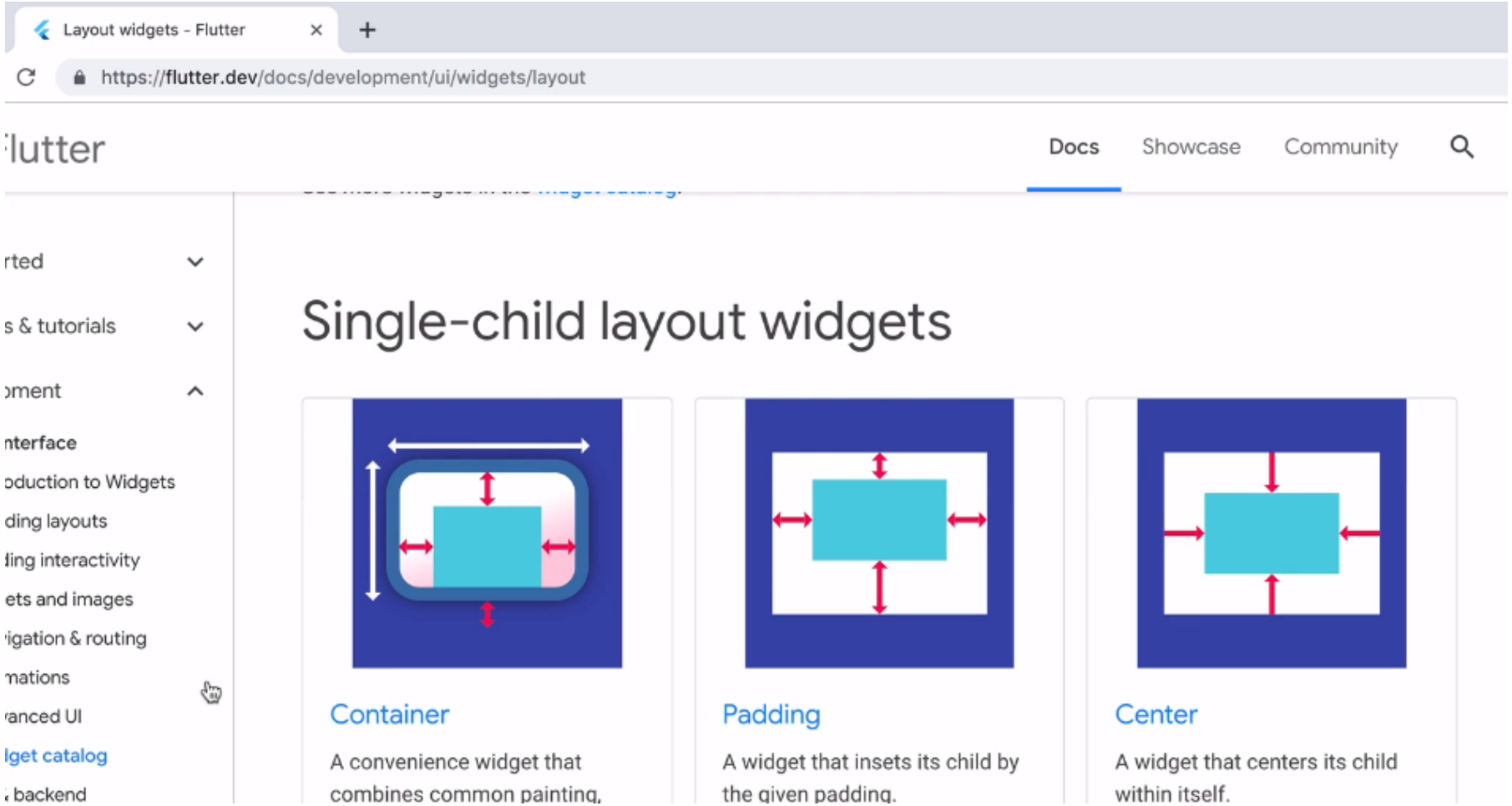
ให้ใส่ debugShowCheckedModeBanner: false, ก่อน home

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

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

class MyApp extends StatelessWidget{
  Widget build(BuildContext context){
    var hello='สมโภชน์';
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      home:Scaffold(
        appBar: AppBar(
          backgroundColor: Colors.amberAccent,
          title:Text('My App Flutter day 1'),
          actions: <Widget>[
            IconButton(icon: Icon(Icons.ac_unit), onPressed: (){

              print('click appbar');

            }) // IconButton
```

การวาง Container Layout

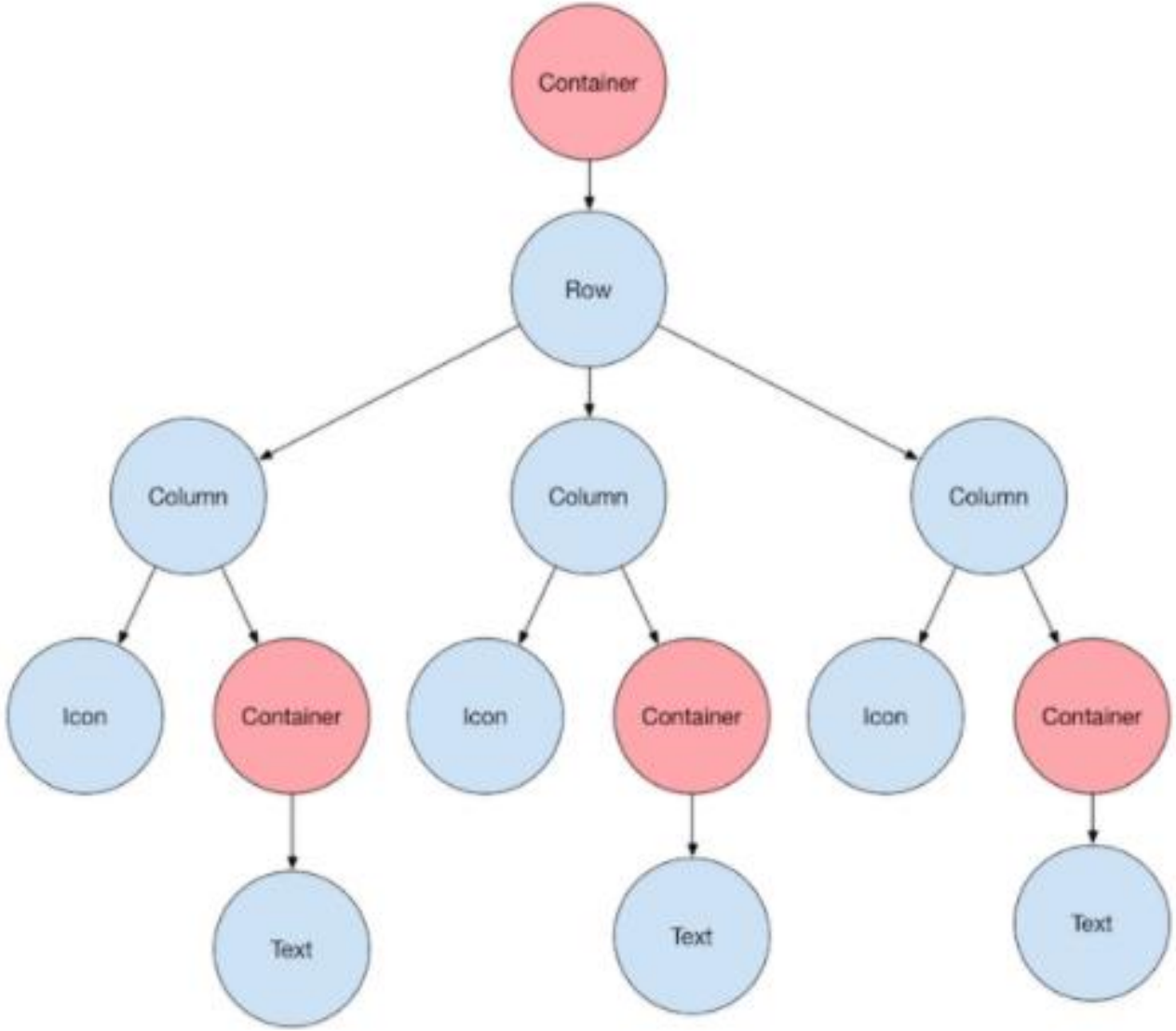




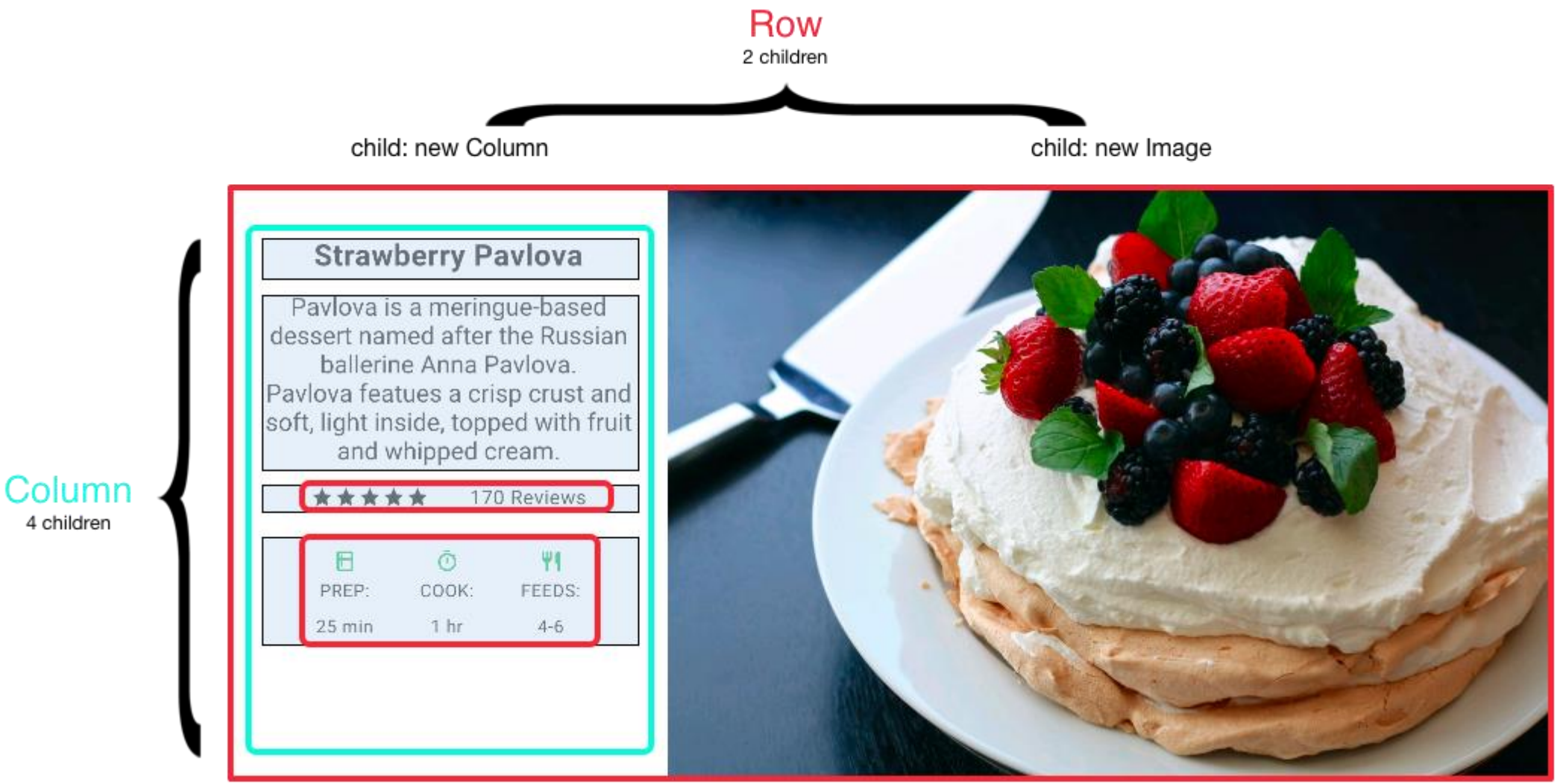


https://flutter.dev/docs/development/ui/layout

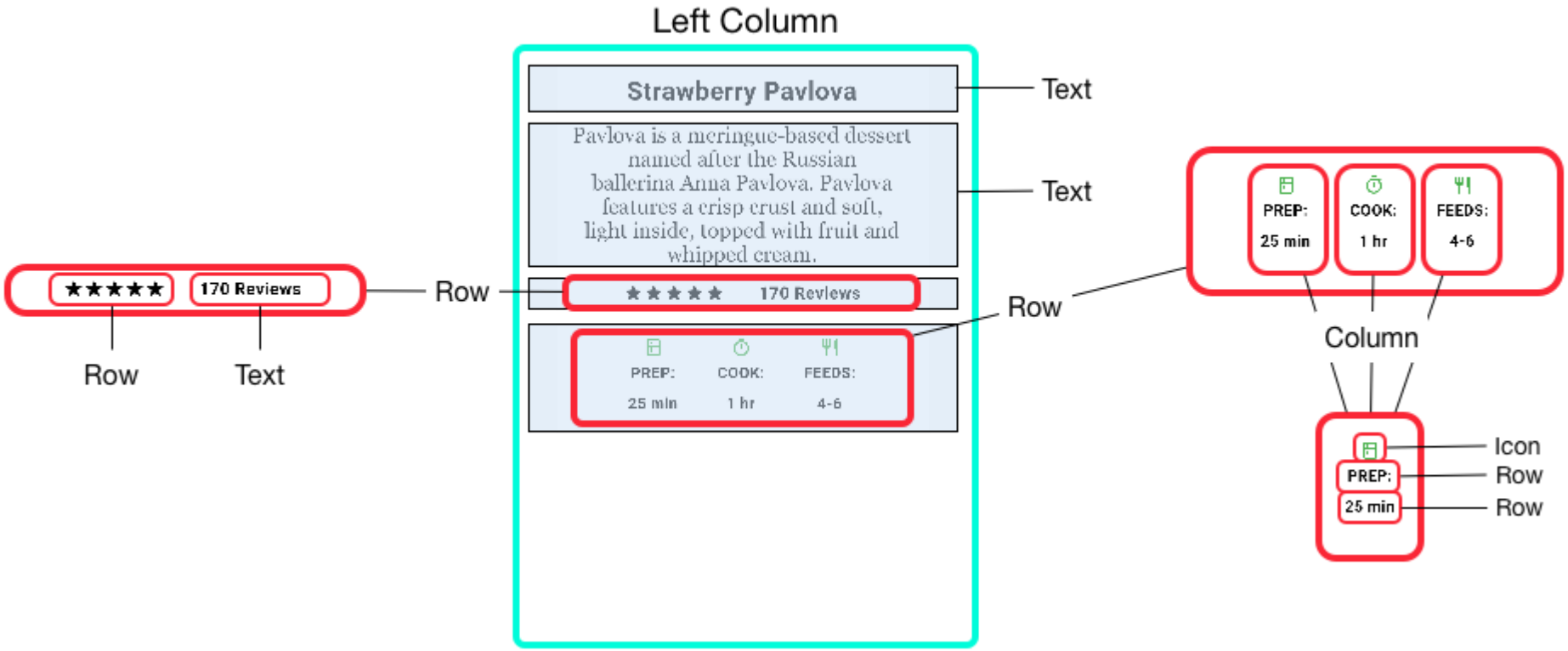
Layouts in Flutter



Layouts in Flutter **Row & Column**

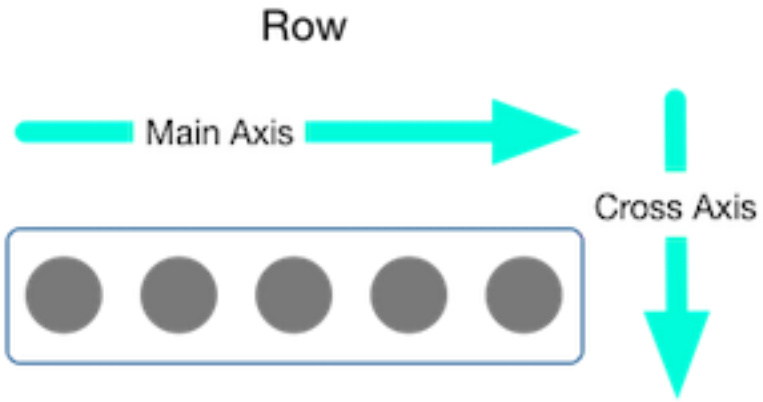


Layouts in Flutter **Row & Column**



- จากภาพจะสังเกตว่า column และ row อาจจะซ่อนอยู่ข้างในของกันและกันได้ ให้จำง่ายๆ ว่าคอลัมน์เรียงบนลงล่าง ส่วน row เรียงจากซ้ายไปขวา

การจัด Aligning Widget




```
Row(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    Image.asset('images/pic1.jpg'),
    Image.asset('images/pic2.jpg'),
    Image.asset('images/pic3.jpg'),
  ],
);
```

App source: row_column

```
Column(
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,
  children: [
    Image.asset('images/pic1.jpg'),
    Image.asset('images/pic2.jpg'),
    Image.asset('images/pic3.jpg'),
  ],
);
```

App source: row_column

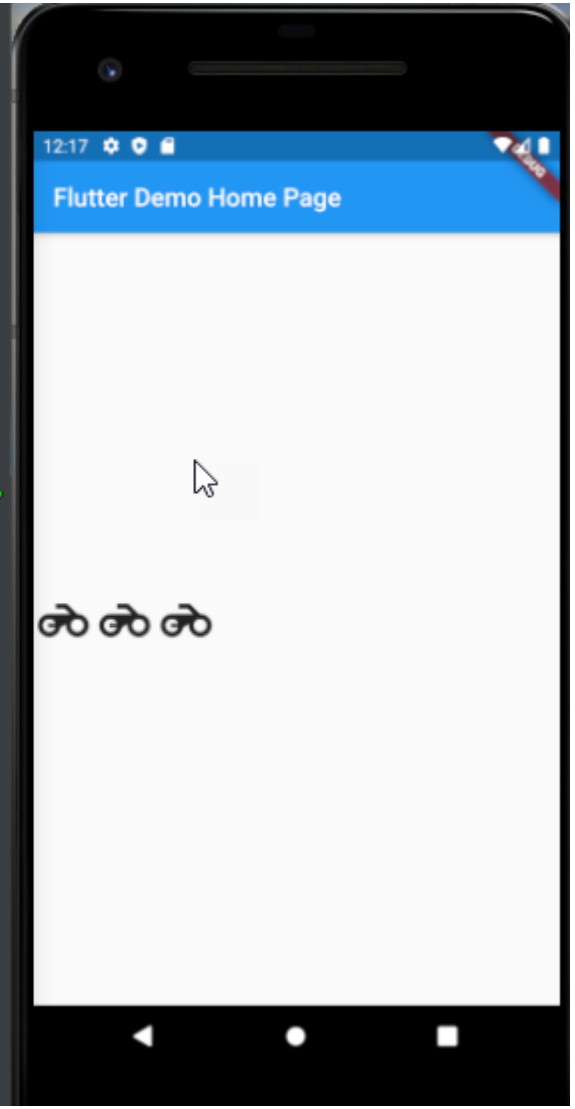
Layout

MainAxisAlignment
การจัดในแกนหลัก เช่น ถ้าใช้ Row คือ การจัดในแนวนอน, ถ้าใช้ Column คือ การจัดในแนวตั้ง

นอกจาก Main แล้วจะมี CrossAxisAlignment คือแนวนอนตัด แกน X

Example Row & MainAxisAlignment.start

```dart
}

@override
Widget build(BuildContext context) {

  return Scaffold(
    appBar: AppBar(

      title: Text(widget.title),
    ), // AppBar

    body: Center(

      child: Row(
        mainAxisAlignment: MainAxisAlignment.start,
        children: <Widget>[
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
        ], // <Widget>[]
      ), // Row


    ), // Center


  ); // Scaffold
```

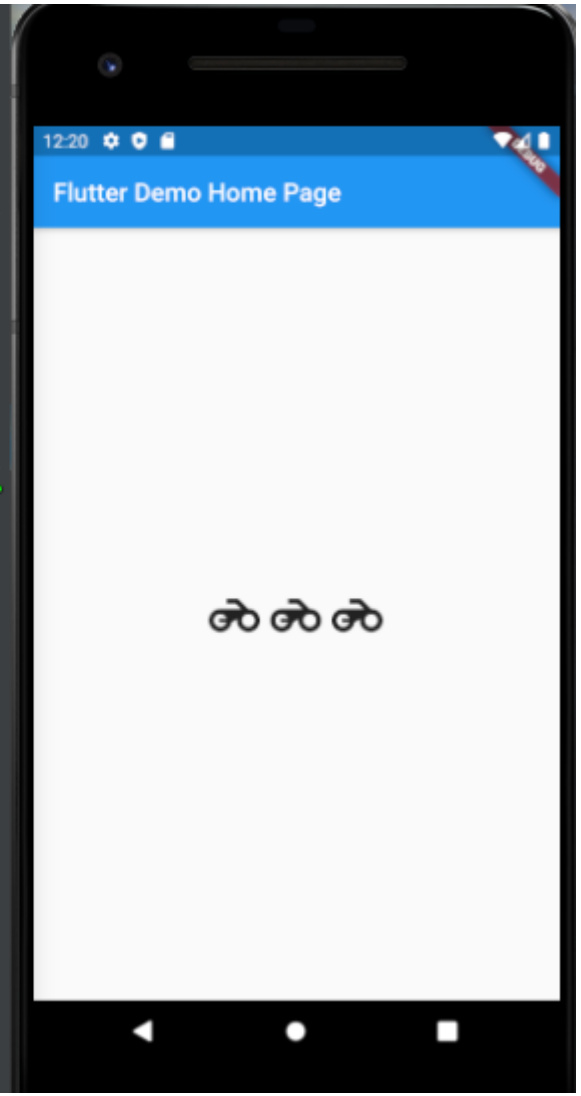


Example Column & MainAxisAlignment.start

```dart
}

@override
Widget build(BuildContext context) {

  return Scaffold(
    appBar: AppBar(

      title: Text(widget.title),
    ), // AppBar

    body: Center(

      child: Column(
        mainAxisAlignment: MainAxisAlignment.start,
        children: <Widget>[
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
        ], // <Widget>[]
      ), // Column


    ), // Center


  ); // Scaffold
```

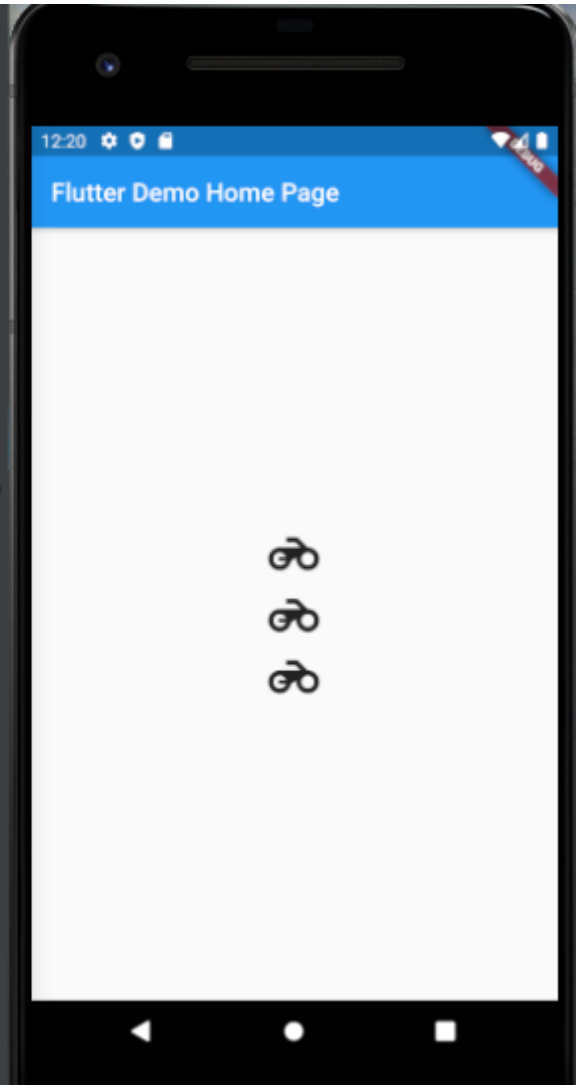
Example Row & MainAxisAlignment.center

```dart
}

@override
Widget build(BuildContext context) {

  return Scaffold(
    appBar: AppBar(

      title: Text(widget.title),
    ), // AppBar

    body: Center(

      child: Row(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
        ], // <Widget>[]
      ), // Row


    ), // Center


  ); // Scaffold
```



Example Column & MainAxisAlignment.center

```
}

@override
Widget build(BuildContext context) {

  return Scaffold(
    appBar: AppBar(

      title: Text(widget.title),
    ), // AppBar

    body: Center(

      child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: <Widget>[
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
        ], // <Widget>[]
      ), // Column


    ), // Center


  ); // Scaffold
```



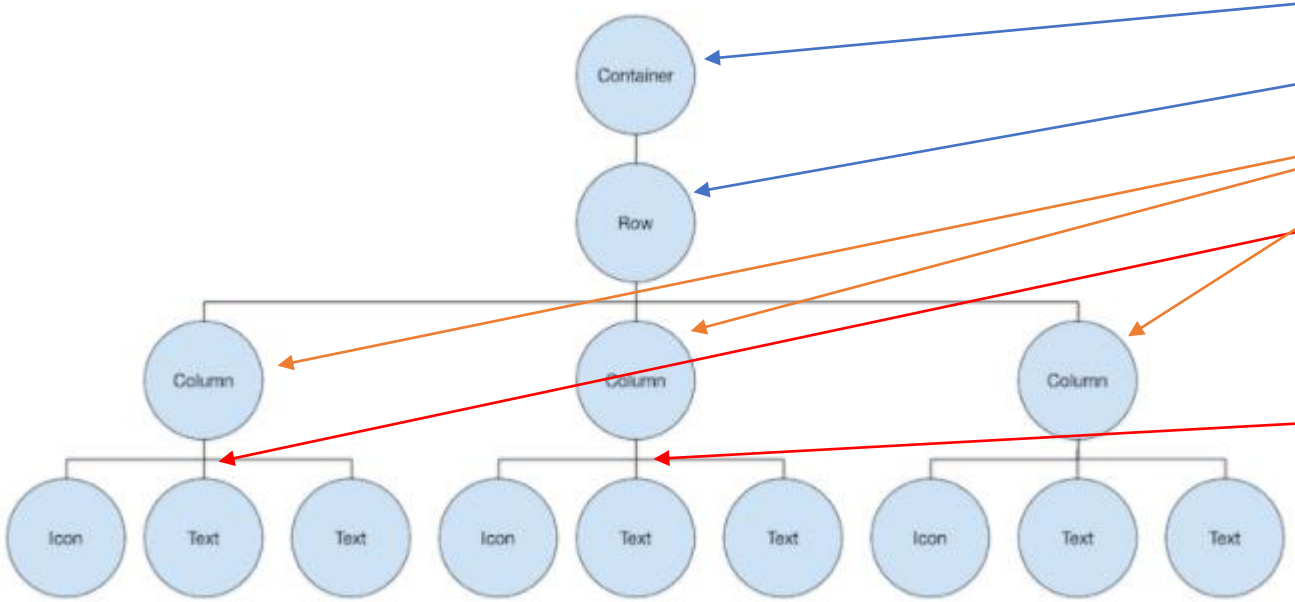
Example Row & MainAxisAlignment.end

```dart
}

@override
Widget build(BuildContext context) {

  return Scaffold(
    appBar: AppBar(

      title: Text(widget.title),
    ), // AppBar

    body: Center(

      child: Row(
        mainAxisAlignment: MainAxisAlignment.end,
        children: <Widget>[
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
        ], // <Widget>[]
      ), // Row


    ), // Center


  ); // Scaffold
```

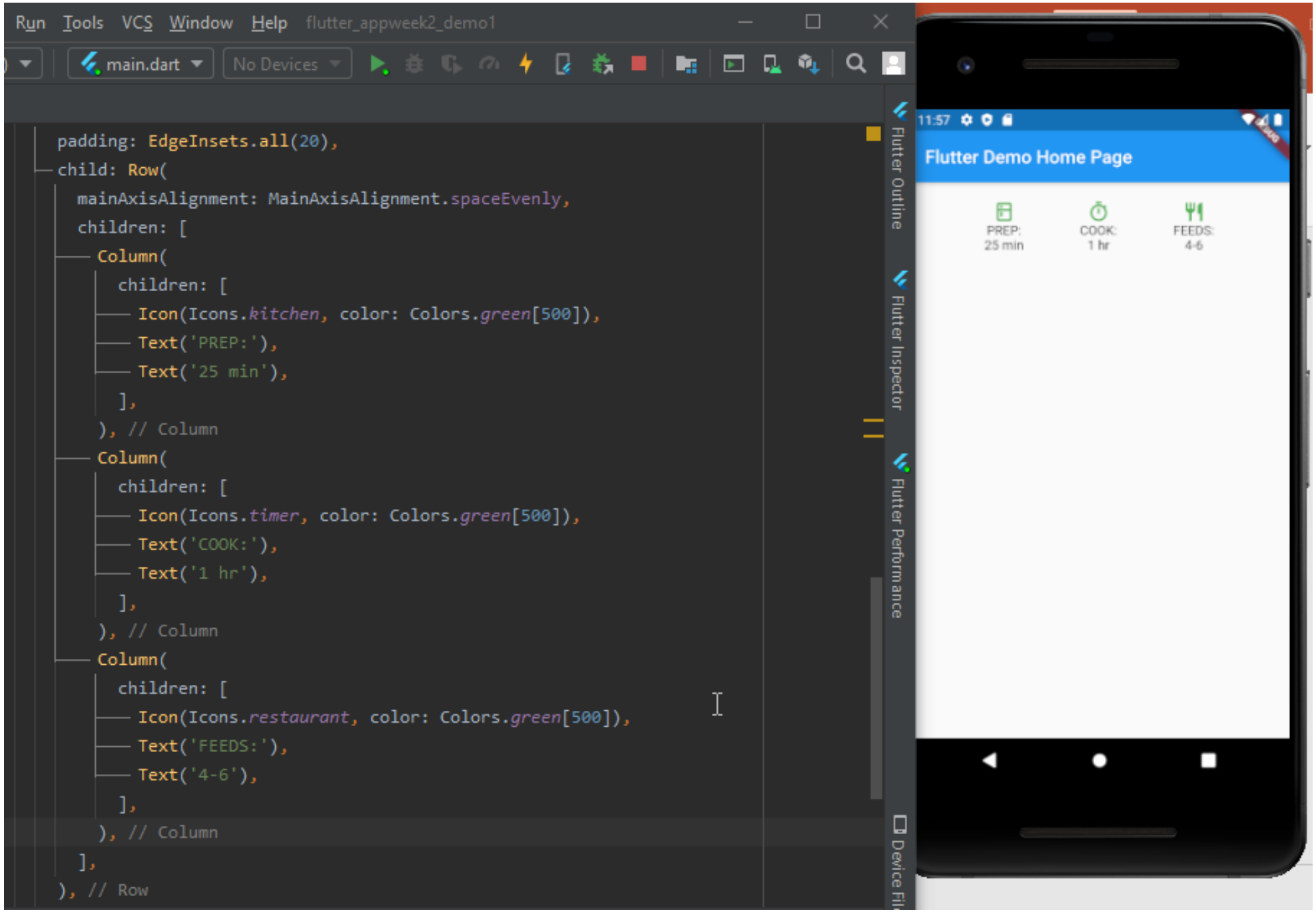
Example Column & MainAxisAlignment.end

```
}

@override
Widget build(BuildContext context) {

  return Scaffold(
    appBar: AppBar(

      title: Text(widget.title),
    ), // AppBar

    body: Center(

      child: Column(
        mainAxisAlignment: MainAxisAlignment.end,
        children: <Widget>[
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
          Icon(Icons.motorcycle, size: 48),
        ], // <Widget>[]
      ), // Column


    ), // Center


  ); // Scaffold
```

Child & Children

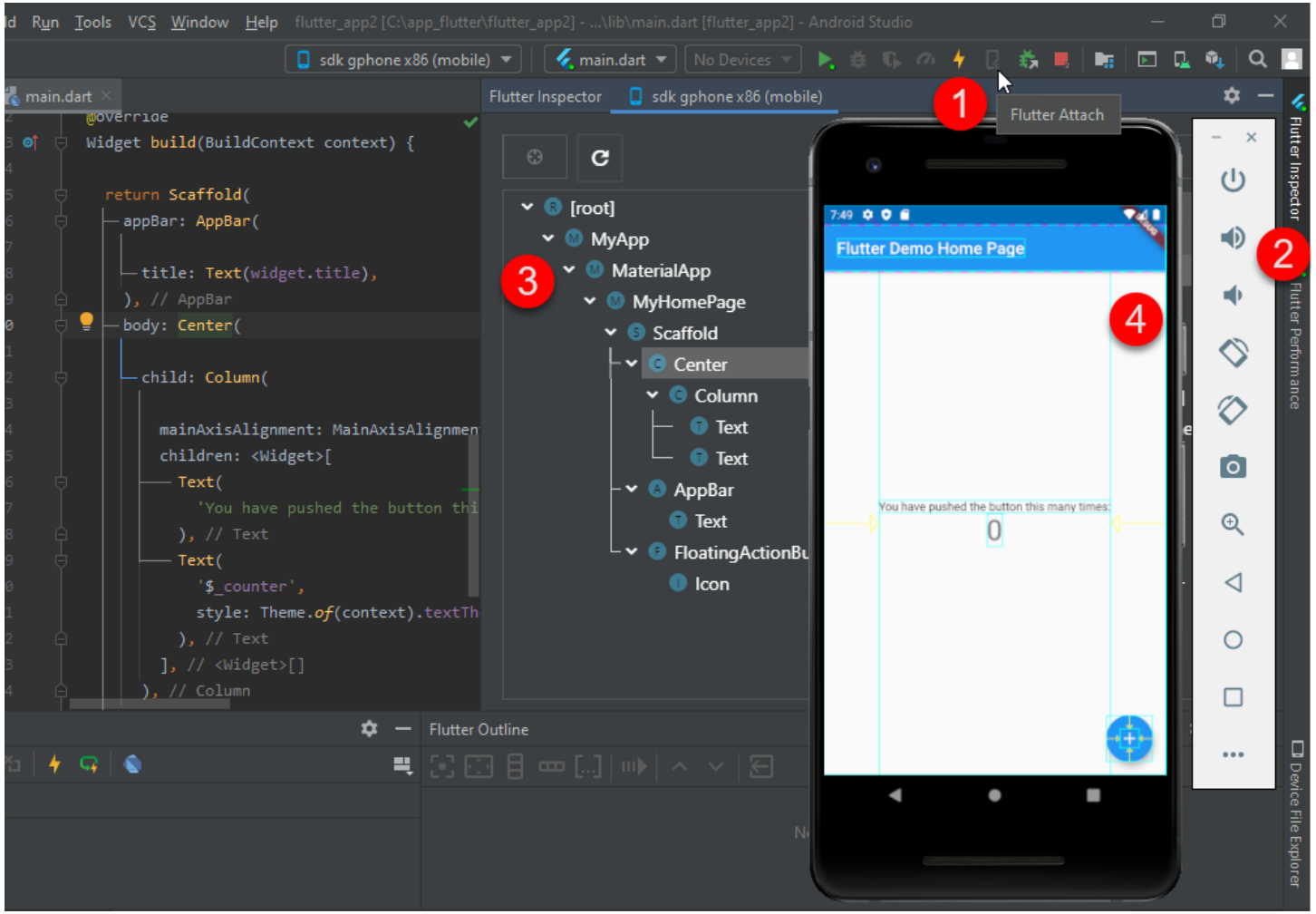



```dart
// DefaultTextStyle.merge() allows you to create a default text
// style that is inherited by its child and all subsequent children.
final iconList = DefaultTextStyle.merge(
  style: descTextStyle,
  child: Container(
    padding: EdgeInsets.all(20),
    child: Row(
      mainAxisAlignment: MainAxisAlignment.spaceEvenly,
      children: [
        Column(
          children: [
            Icon(Icons.kitchen, color: Colors.green[500]),
            Text('PREP:'),
            Text('25 min'),
          ],
        ),
        Column(
          children: [
            Icon(Icons.timer, color: Colors.green[500]),
            Text('COOK:'),
            Text('1 hr'),
          ],
        ),
        Column(
          children: [
            Icon(Icons.restaurant, color: Colors.green[500]),
            Text('FEEDS:'),
            Text('4-6'),
          ],
        ),
      ],
    ),
  ),
);
```


Run Tools VCS Window Help flutter_appweek2_demo1
main.dart
No Devices
padding: EdgeInsets.all(20),
child: Row(
mainAxisAlignment: MainAxisAlignment.spaceEvenly,
children: [
Column(
children: [
Icon(Icons.kitchen, color: Colors.green[500]),
Text('PREP:'),
Text('25 min'),
], // Column
Column(
children: [
Icon(Icons.timer, color: Colors.green[500]),
Text('COOK:'),
Text('1 hr'),
], // Column
Column(
children: [
Icon(Icons.restaurant, color: Colors.green[500]),
Text('FEEDS:'),
Text('4-6'),
], // Column
], // Row
Flutter Outline
Flutter Inspector
Flutter Performance
Device File
11:57
Flutter Demo Home Page
PREP:
25 min
COOK:
1 hr
FEEDS:
4-6

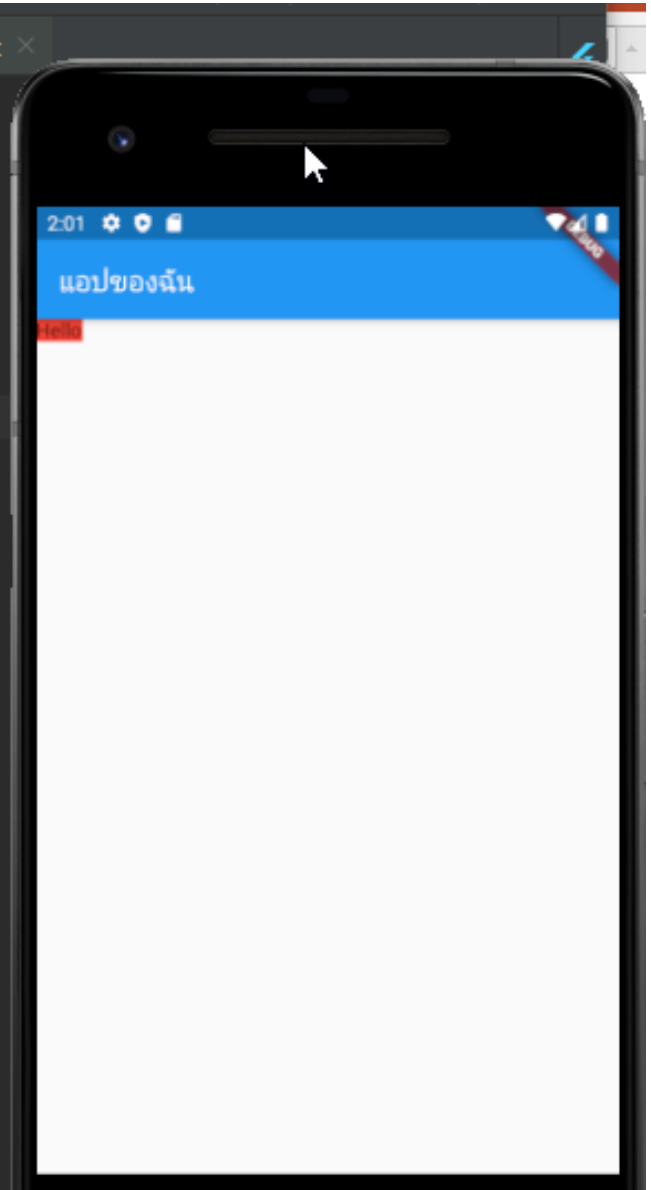
Flutter บน Android Studio จะมี Tool

- กด Flutter Attach ด้านบนตอน run แล้วมากด Flutter Inspector แล้วมาเลื่อนดู Layout ได้ดังภาพ

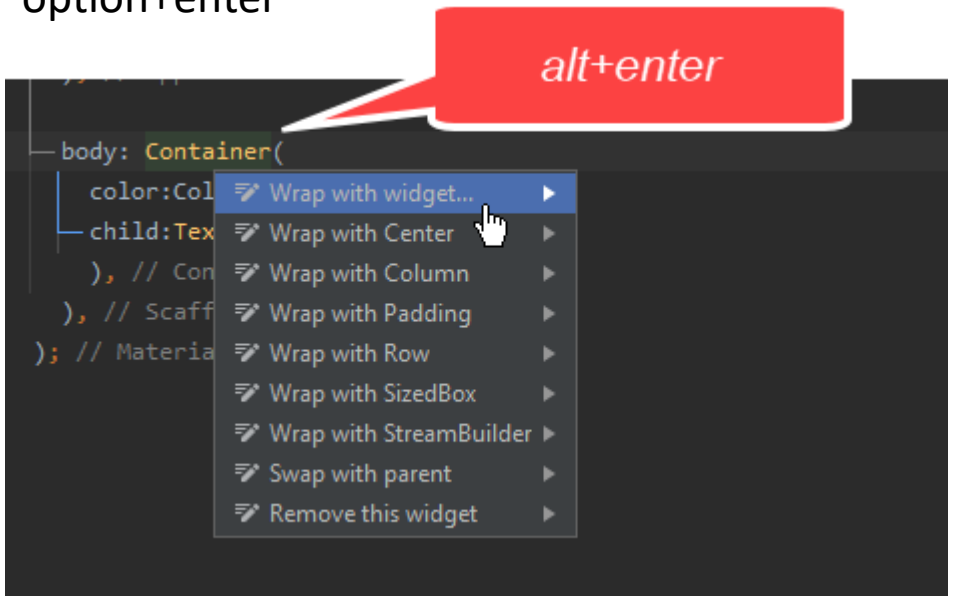


Home Layout

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

class MyApp extends StatelessWidget {
  Widget build(BuildContext context){
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('แอปของฉัน'),
        ), // AppBar

        body: Container(
          color:Colors.red,
          child:Text('Hello'),
          ), // Container
        ), // Scaffold
      ); // MaterialApp

  }
}
```

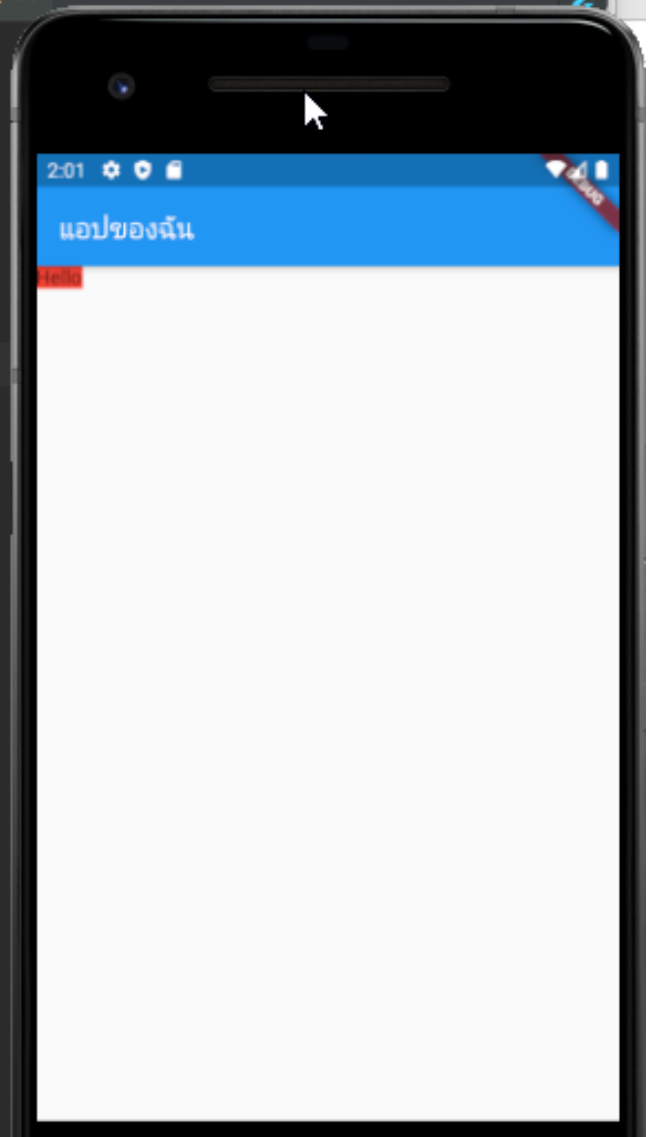


ถ้าใส่โค้ดตามภาพจะสังเกตุว่า ข้อความ Hello จะชิดกับ appbar เราสามารถกำหนด Layout ได้ โดยในส่วนของ body:Containner ให้กด alt+enter หรือ option+enter

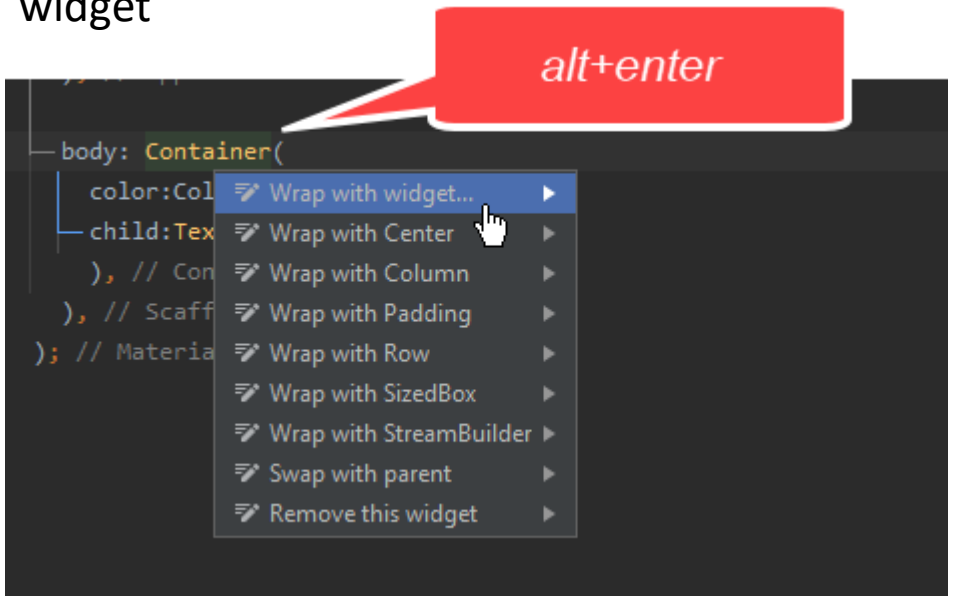


Home Layout

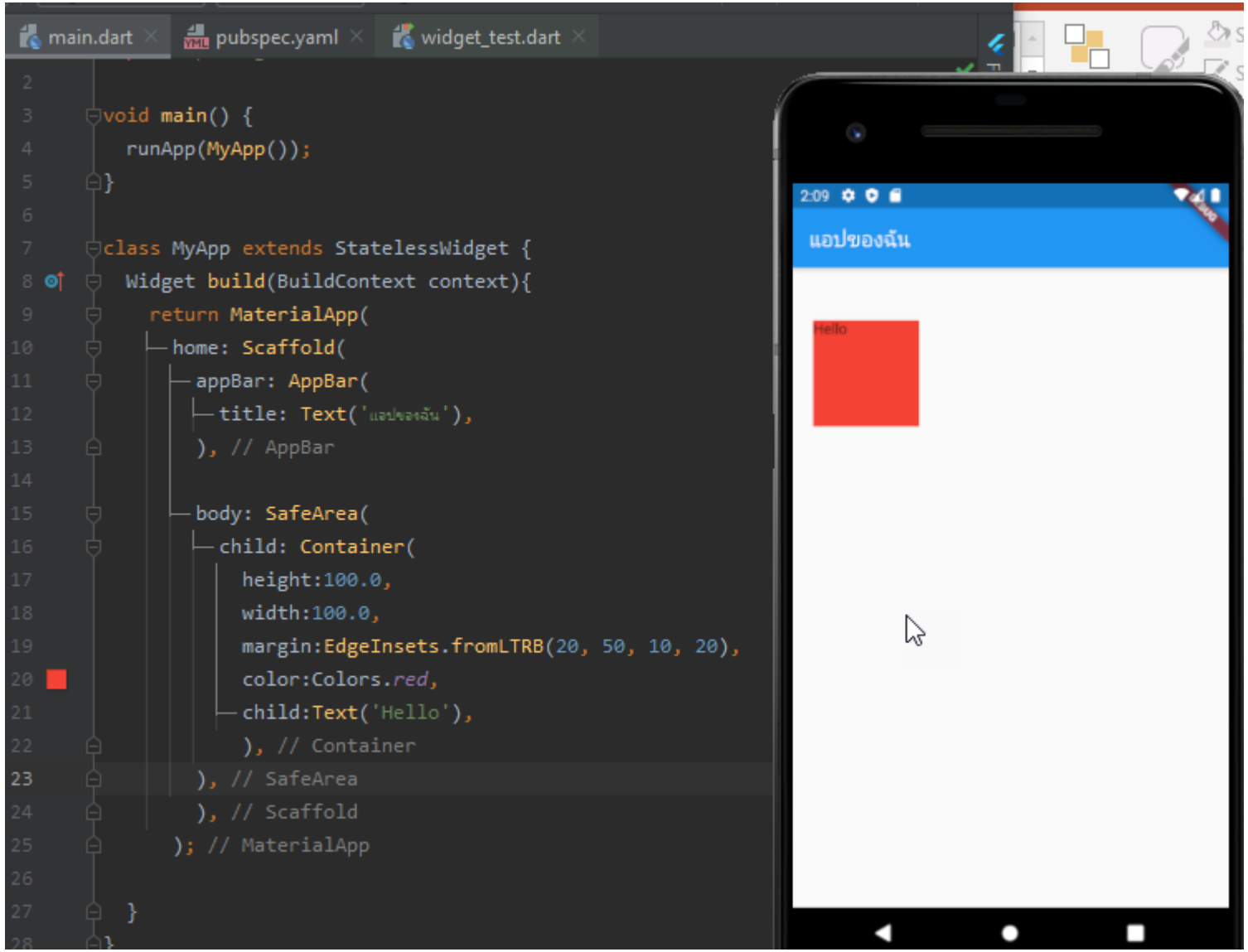
```dart
void main() {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  Widget build(BuildContext context){
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('แอปของฉัน'),
        ), // AppBar

        body: Container(
          color:Colors.red,
          child:Text('Hello'),
          ), // Container
        ), // Scaffold
      ); // MaterialApp

  }
}
```



ถ้าใส่โค้ดตามภาพจะสังเกตุว่า ข้อความ Hello จะชิดกับ appbar เราสามารถกำหนด Layout ได้ โดยในส่วนของ body:Containner ให้กด alt+enter หรือ option+enter เลือกอันแรก wrap width widget



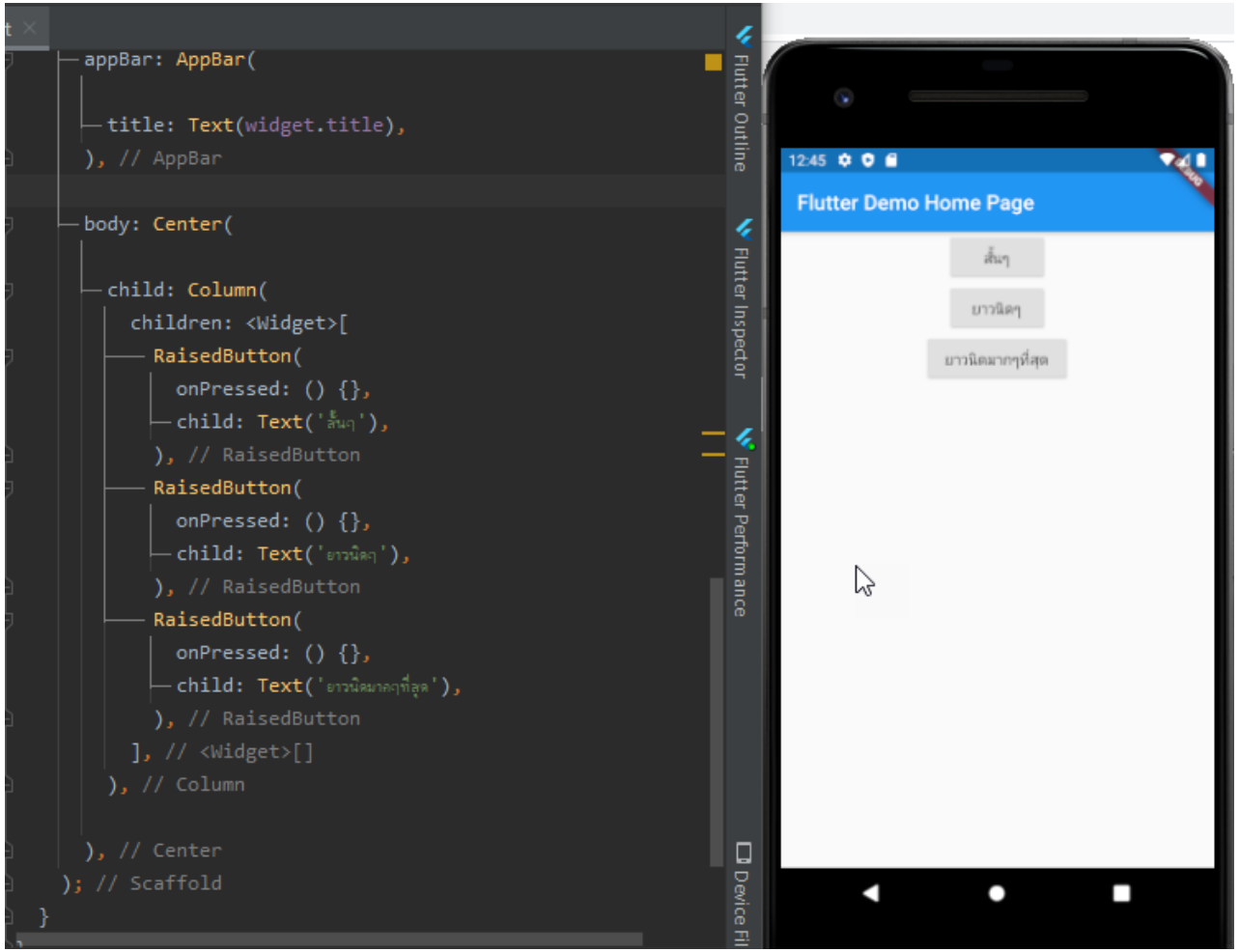
Home Layout



หลังจากนั้นให้แก้ widget เป็น SafeArea เพื่อให้มัน Fix Layout แล้วกำหนดขนาด ความกว้าง ความสูง และ margin ของ element ได้ตามโค้ด

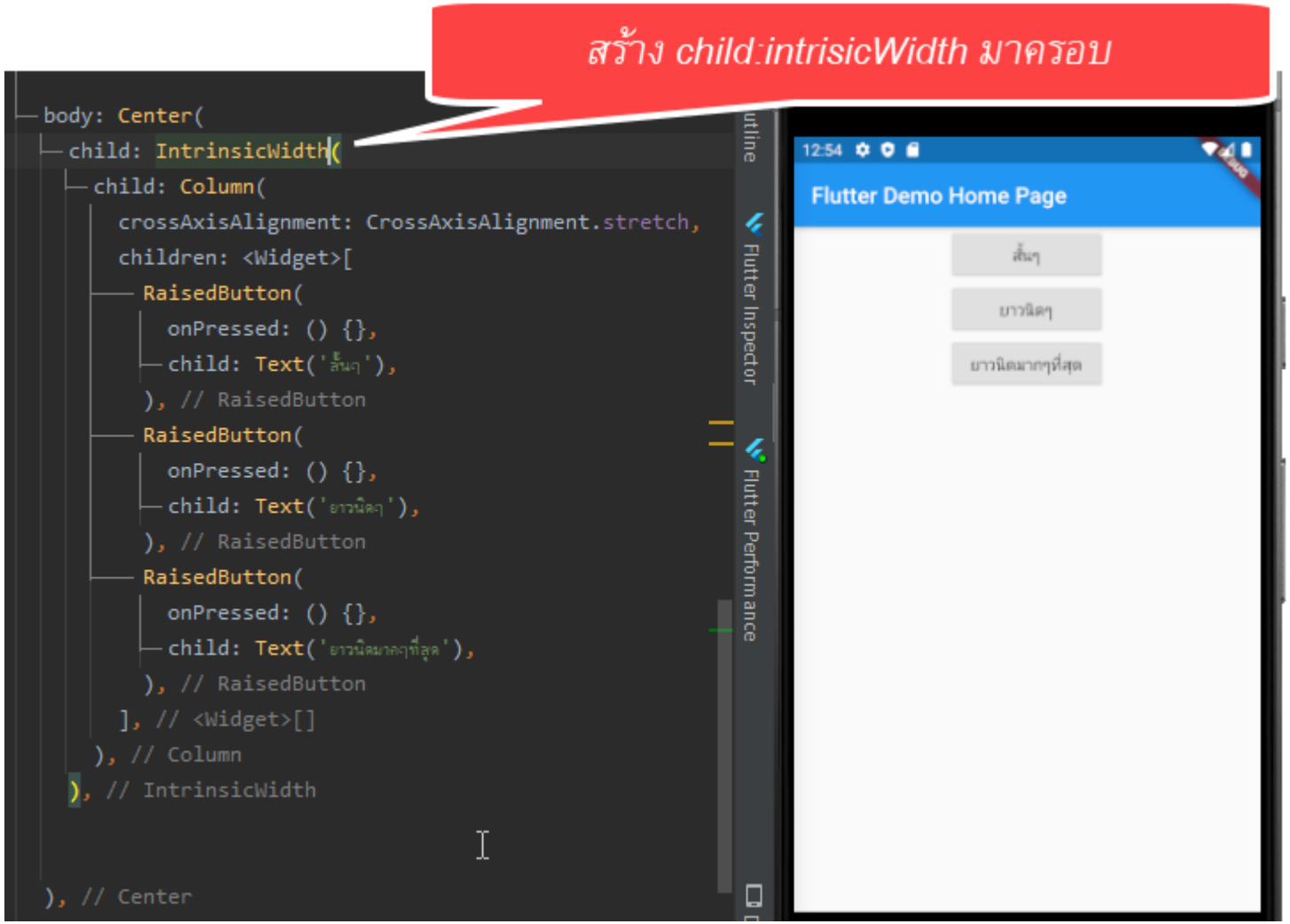
IntrinsicWidth and IntrinsicHeight

- Intrinsic แปลว่า แท้จริง, ที่อยู่ภายใน ในกรณีที่ต้องการให้ความกว้าง หรือความสูง นั้น กว้างเท่ากับ widget ที่กว้างที่สุดใน Column หรือสูงเท่ากับ widget ที่สูงที่สุดใน Row เราไม่ต้องยุ่งยากไปใช้วิธีอื่นๆ โดยทั่วไปถ้าเราไม่ใช้ Intrinsic จะเป็นดังภาพด้านล่าง



IntrinsicWidth and IntrinsicHeight

- พอใช้ IntrinsicWidth จะได้แบบภาพด้านล่าง

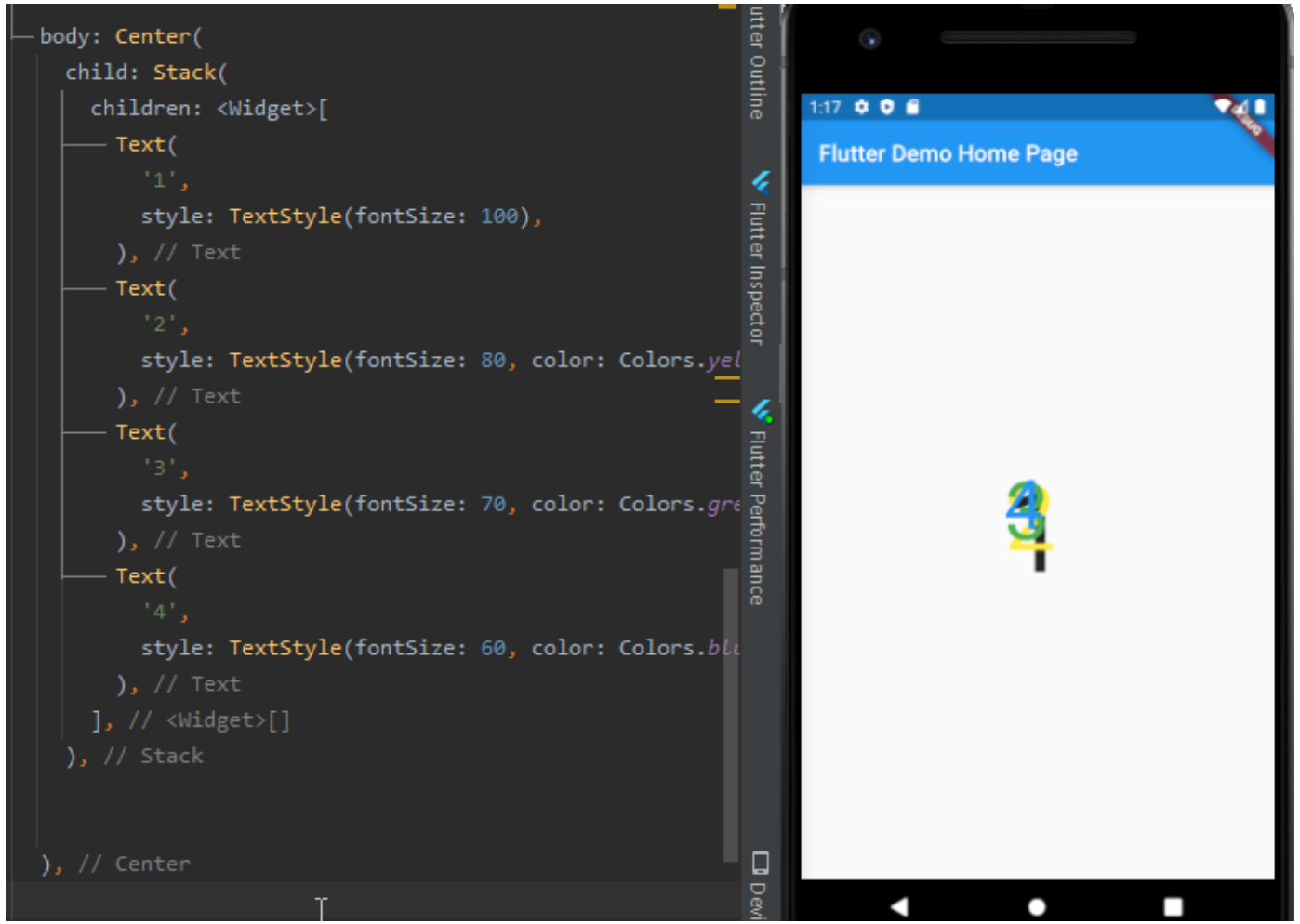


IntrinsicHeight

```
body: Center(
  child: IntrinsicHeight(
    child: Row(
      crossAxisAlignment: CrossAxisAlignment.stretch,
      children: <Widget>[
        RaisedButton(
          onPressed: () {},
          child: Text('สั้นๆ'),
        ), // RaisedButton
        RaisedButton(
          onPressed: () {},
          child: Text('ยาวนิดๆ'),
        ), // RaisedButton
        RaisedButton(
          onPressed: () {},
          child: Text('ยาวนิดมากๆที่สุด'),
        ), // RaisedButton
      ], // <Widget>[]
    ), // Row
  ), // IntrinsicHeight

), // Center
```

Stack

- ใช้สำหรับการวาง layout แบบวาง widget ซ้อนทับกันนั่นเอง โดยใช้ List<Widget> widget ที่อยู่อันดับแรกสุดคืออยู่ล่างสุด widget ที่อยู่อันดับสุดท้ายคือตัวที่อยู่บนสุด



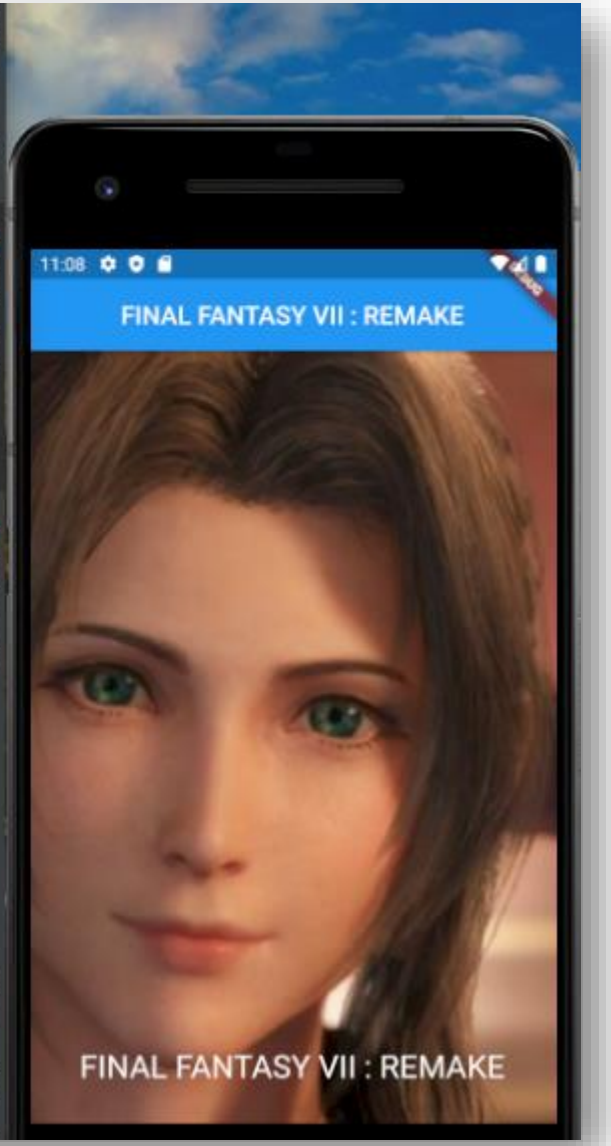
Stack

- สำหรับกรณีที่ต้องการจัดตำแหน่งสามารถใช้ Positioned ในการวางตำแหน่งตามที่ต้องการได้เลย โดยพื้นที่วางนั้นขึ้นอยู่กับขนาดของ widget แม่

```
body: Center(
  child: Stack(
    fit: StackFit.expand,
    children: <Widget>[
      Material(color: Colors.blueGrey),
      Positioned(
        top: 0,
        left: 0,
        child: Text(
          '1',
          style: TextStyle(fontSize: 100, color: Colors.
        ), // Text
      ), // Positioned
      Positioned(
        bottom: 0,
        right: 0,
        child: Text(
          '2',
          style: TextStyle(fontSize: 80, color: Colors.
        ), // Text
      ), // Positioned

    ], // <Widget>[]
  ), // Stack

), // Center
```

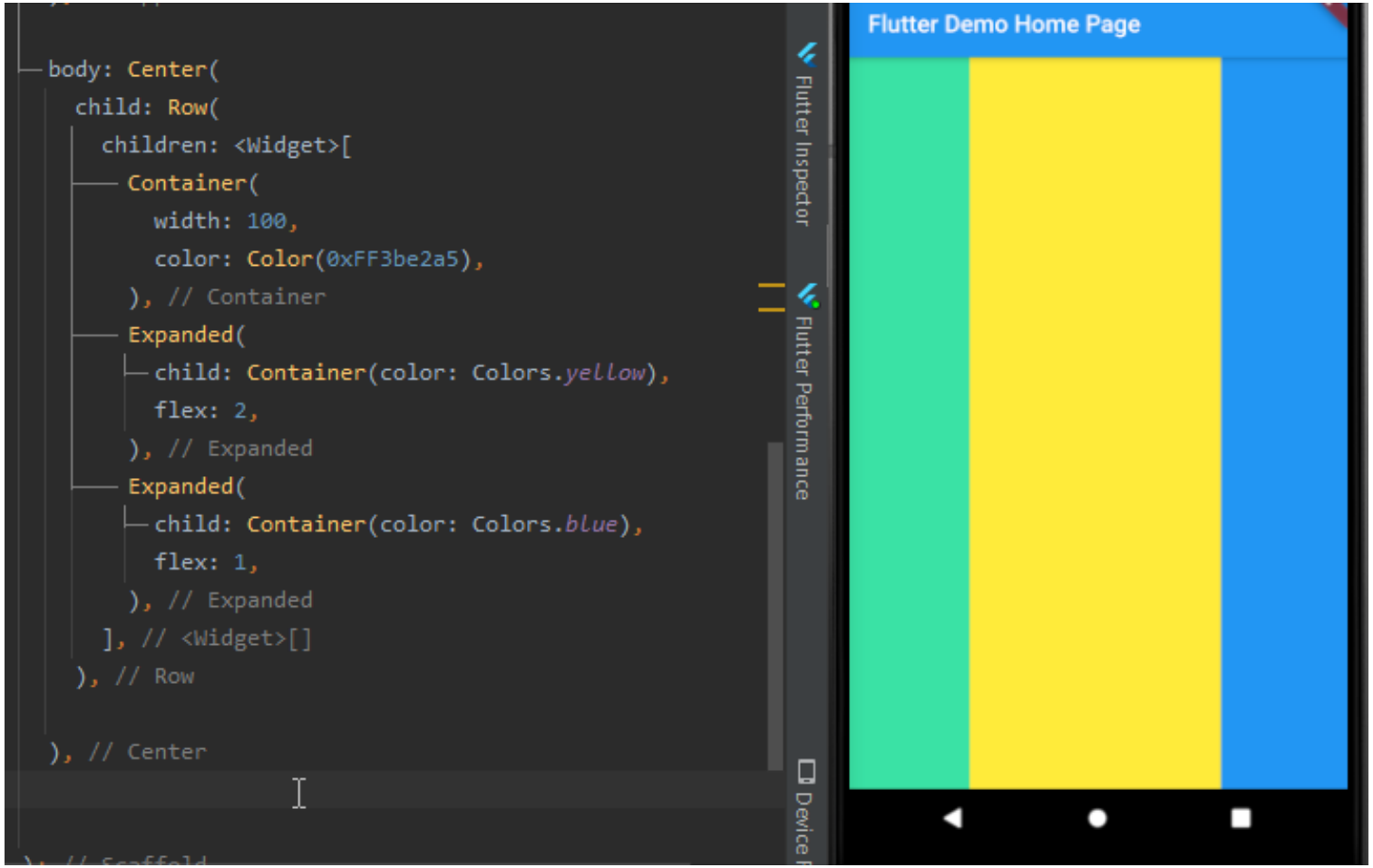
Stack เอารูปภาพ ซ่อน Image ได้

```
body: SafeArea(
  child: Stack(
    children: <Widget>[
      // Background Image
    Container(
        decoration: BoxDecoration(
          image: DecorationImage(
            image: NetworkImage('https://static3.cbrimages.com/wordpress/wp-content/uploa
            fit: BoxFit.cover,
          ), // DecorationImage
        ), // BoxDecoration
        child: null), // Container
    Center(
      child: Column(
        mainAxisAlignment: MainAxisAlignment.end,
        children: <Widget>[
        Padding(
          padding: const EdgeInsets.only(bottom: 30),
          child: Text(
            'FINAL FANTASY VII : REMAKE',
            style: TextStyle(
              color: Colors.white,
              fontSize: 25,
            ), // TextStyle
          ), // Text
        ), // Padding
        ], // <Widget>[]
      ), // Column
```



Expanded

- Expanded จะใช้สำหรับกรณีที่ต้องการที่จะใช้พื้นที่ที่เหลือทั้งเหลือทั้งหมดของ widget แม่ หาก widget แม่นั้น มี Expanded มากกว่า 1 ตัวจะต้องทำการใส่สัดส่วนให้มันผ่านค่า flex เพิ่มเติมเข้าไปด้วย



Widget คืออะไร

ใน Flutter จะมองทุกอย่างเกือบทั้งหมดเป็น widget

Widget คือ ส่วนที่ถูกใช้สร้างเป็นหน้าตาของ App หรือที่เรียกวา user interface (UI) โดยนำมาประกอบเรียงกันเป็นลำดับขั้นขึ้นเป็นโครงสร้าง แต่ละ widget จะถูกวางซ้อนอยู่ภายใน Parent widget และได้รับการส่งต่อสืบทอดคุณสมบัติต่างๆ จาก Parent อีกที แม้กระทั้ง application object ก็ถือเป็น widget ซึ่งเราเรียกว่า root widget MaterialApp คือ root widget

เราอาจจำแนก Widget ตามการใช้งาน ได้เป็น ดังนี้

- ใช้กำหนดโครงสร้าง (Structural Element) เช่น ปุ่ม button หรือ menu
- ใช้กำหนดลักษณะ หรือรูปแบบ (Stylistic Element) เช่น font หรือ color
- ใช้จัดวาง และกำหนดมุมมองเลเอาท์ (Aspect of Layout) เช่น padding หรือ alignment

StatelessWidget และ StatefulWidget คืออะไร

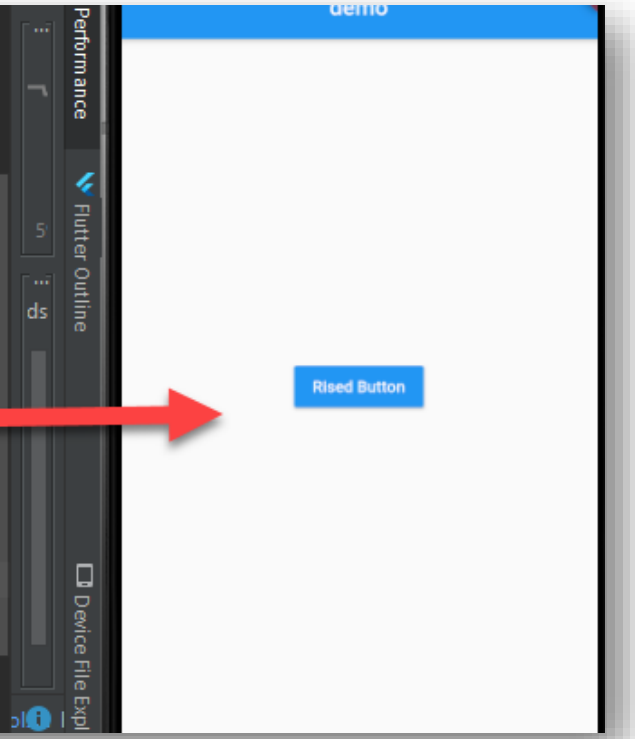
ใน App ของเราจะมี widget อยู่ 2 ประเภทหลัก ที่ใช้งานคือ stateless และ stateful widget โดย state ก็คือสภาวะของสิ่งนั้นๆ stateless จึงหมายถึง widget ที่ไม่มี state หรือไม่มีสภาวะการเปลี่ยนแปลง หรือไม่จำเป็นต้องใช้งานการเปลี่ยนแปลง จึงใช้งาน widget นี้ ส่วน stateful หมายถึง widget ที่มี state หรือมีสภาวะการเปลี่ยนแปลง ไปตามข้อมูลที่ได้รับหรือจากการกำหนดจากผู้ใช้ข้อแตกต่างที่สำคัญของทั้งสองส่วนนี้คือ stateful widget จะมี State object ที่ใช้ในการเก็บข้อมูล state และ ทำการส่งต่อสำหรับใช้งานในกระบวนการสร้าง widget ใหม่เมื่อมีการเปลี่ยนแปลง ทำให้ค่า state ไม่ได้หายไปไหน

การใช้งาน StatelessWidget

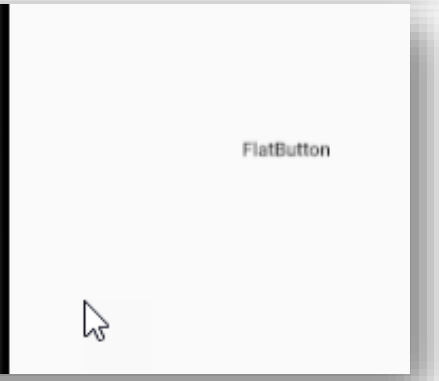
Stateless widget ใน Flutter เป็น widget ที่ไม่จำเป็นที่ต้องมีการเปลี่ยนแปลง state เกิดขึ้น โดยเราจะใช้ stateless widget สำหรับสร้าง widget แบบคงที่ เหมาะสำหรับใช้ในการสร้าง และกำหนดส่วนของ UI ซึ่งจะปรับแต่งเฉพาะค่าข้อมูลของ ตัว widget เท่านั้นเช่น Text widget ก็ถือเป็น stateless widget ที่เป็น subclass ของ StatelessWidget

Button

```dart
class _MyHomePageState extends State<MyHomePage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Center(child: Text('demo')),
      ), // AppBar
      body: Center(
        child: RaisedButton(
          onPressed: (){},
          color:Colors.blue,
          child:Text(
            'Rised Button',
            style: TextStyle(color: Colors.white),
          ) // Text
        ), // RaisedButton
      )); // Center, Scaffold
  }
}
```

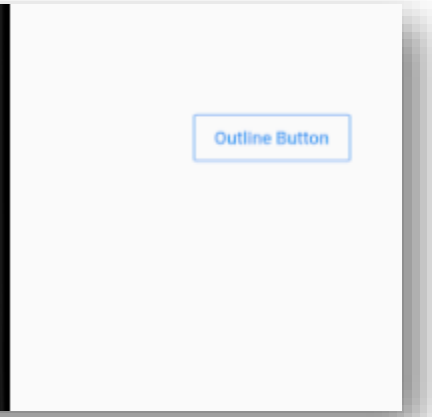



```dart
body: Center(
  child: FlatButton(
    onPressed: (){},
    child:Text(
      'FlatButton',
    ) // Text
  ), // FlatButton
)); // Center, Scaffold
```

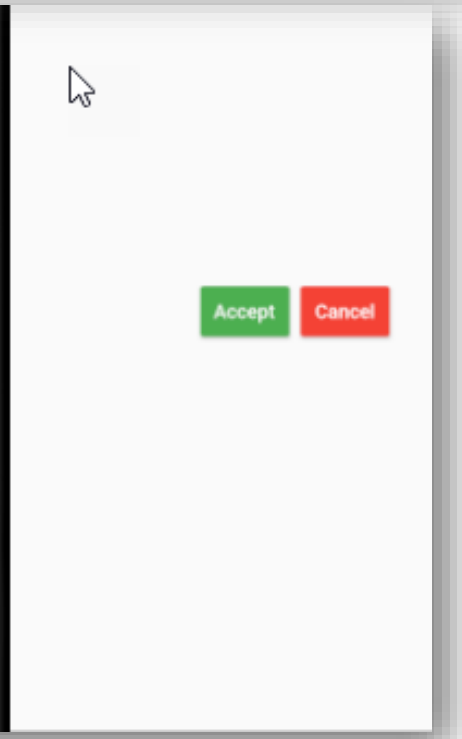


Button

```
body: Center(
  child: OutlineButton(
    onPressed: () => {},
    textColor: Colors.blue,
    borderSide: BorderSide(color: Colors.blue, width: 1.0, style: BorderStyle.solid),
    child: Text(
      'Outline Button',
    ), // Text
  ), // OutlineButton
)); // Center, Scaffold
```



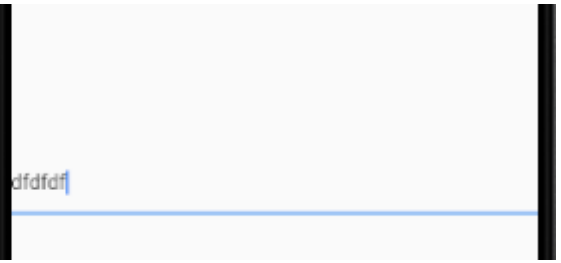

```
body: Center(
  child: ButtonBar(
    alignment: MainAxisAlignment.center,
    children: <Widget>[
      RaisedButton(
        onPressed: () => {},
        color: Colors.green,
        child: Text('Accept', style: TextStyle(color: Colors.white),),
      ), // RaisedButton
      RaisedButton(
        onPressed: () => {},
        color: Colors.red,
        child: Text('Cancel', style: TextStyle(color: Colors.white),),
      ), // RaisedButton
    ], // <Widget>[]
  ) // ButtonBar
)); // Center, Scaffold
```



TextField

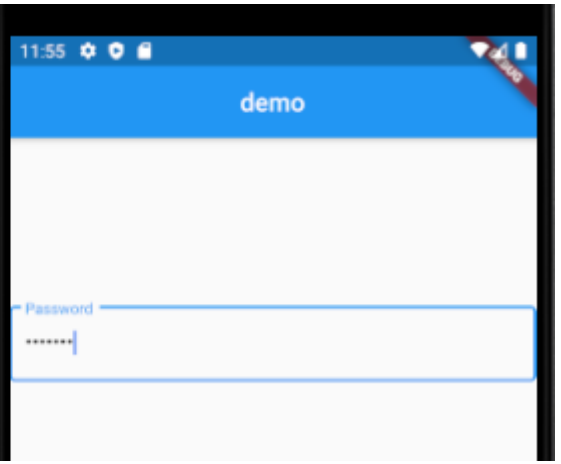
TextField คือ Widget Input ที่จะรับค่าจากการกด key ตัวอักษรลงไปบนระบบ

```
body: Center(

  child:TextField(
    ) // TextField

)); // Center, Scaffold
```



สามารถกำหนดรูปแบบเหมือนกรอก password ได้

```
body: Center(

  child:TextField(
     obscureText: true,
     decoration: InputDecoration(
       border: OutlineInputBorder(),
       labelText: 'Password',
     ), // InputDecoration
    ) // TextField

)); // Center, Scaffold
```

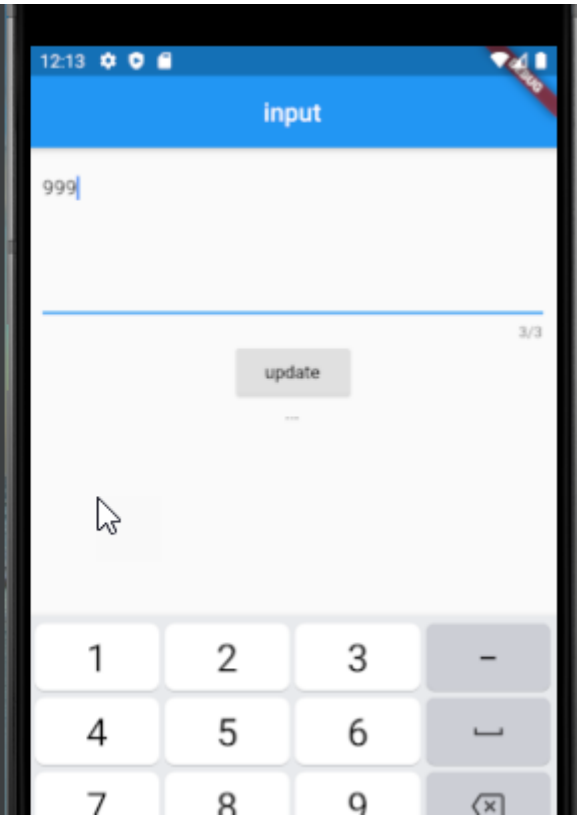


ตัวอย่างการกรอกข้อมูลใน TextFiled แล้วกดปุ่มให้แสดงค่าที่กดได้

```dart
class _MyHomePageState extends State<MyHomePage> {
  String _input="...";
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Center(child: Text('input')),
      ), // AppBar
      body: Container(
            padding: EdgeInsets.all(10),
            child:Column(
              children: <Widget>[
              TextField(
                keyboardType: TextInputType.number,
              ), // TextField
              RaisedButton(child:Text('update'),onPressed: (){
                setState(() {

                });
              }), // RaisedButton
              Text('$_input'),
              ], // <Widget>[]
            ) // Column


      )); // Container, Scaffold
  }
}
```

TextField มี Properties ที่น่าสนใจดังนี้

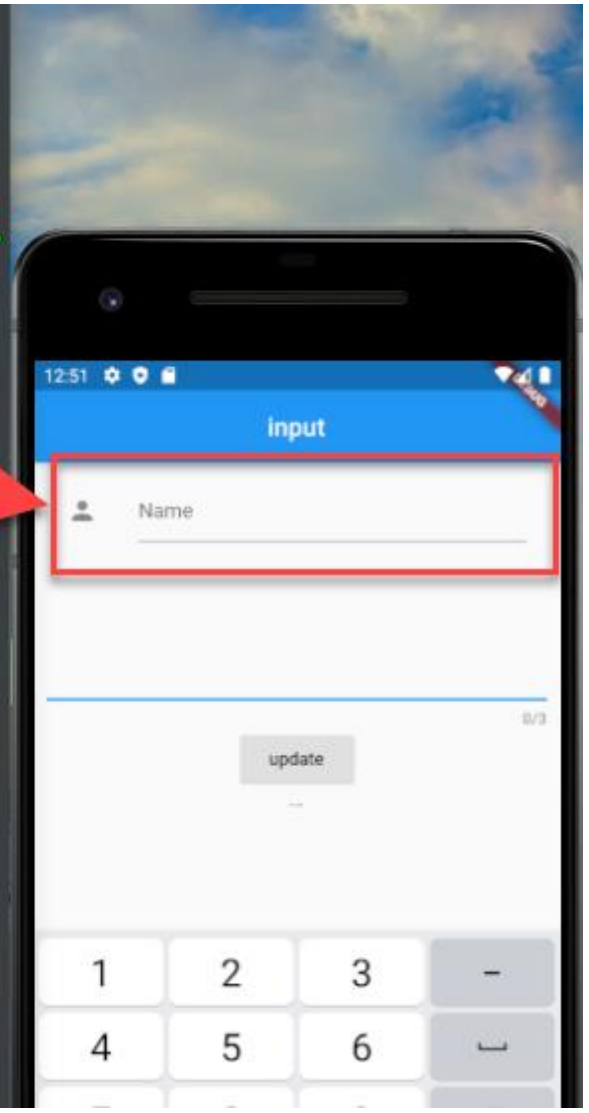
```
body: Container(
      padding: EdgeInsets.all(10),
      child:Column(
        children: <Widget>[
        TextField(
          keyboardType: TextInputType.number,
          maxLength: 3,
          maxLines: 5,
        ), // TextField
        RaisedButton(child:Text('update'),onPressed: (){
          setState(() {

          });
        }), // RaisedButton
        Text('$_input'),
        ], // <Widget>[]
      ) // Column


)); // Container, Scaffold
```



- keyboardType กำหนดรูปแบบการกรอกข้อมูล เช่นกรอกแต่ตัวเลข กรอกอีเมล
- maxLength กำหนดขนาดความยาวของตัวอักษร
- maxLines กำหนดจำนวนไลน์ที่สามารถกรอกได้ (จำนวนบรรทัดในการกรอก)
- TextInputAction สามารถกำหนดให้มีรูปแหวนขยายมาใน keyboard ได้ (TextInputAction=TextInputAction.search)

ListTile widget ที่สามารถใส่ leading และ textfield ได้

```
body: Container(
      padding: EdgeInsets.all(10),
      child:Column(

        children: <Widget>[
        ListTile(
          leading: Icon(Icons.person),
          title:TextField(
            decoration: InputDecoration(
              hintText:"Name",
            ), // InputDecoration
          ), // TextField
        ), // ListTile
        TextField(
          keyboardType: TextInputType.number,
          maxLength: 3,
          maxLines: 5,
        ), // TextField
        RaisedButton(child:Text('update'),onPressed: (){
          setState(() {

          });
        }), // RaisedButton
        Text('$_input'),
      ], // <Widget>[]

    ) // Column
```

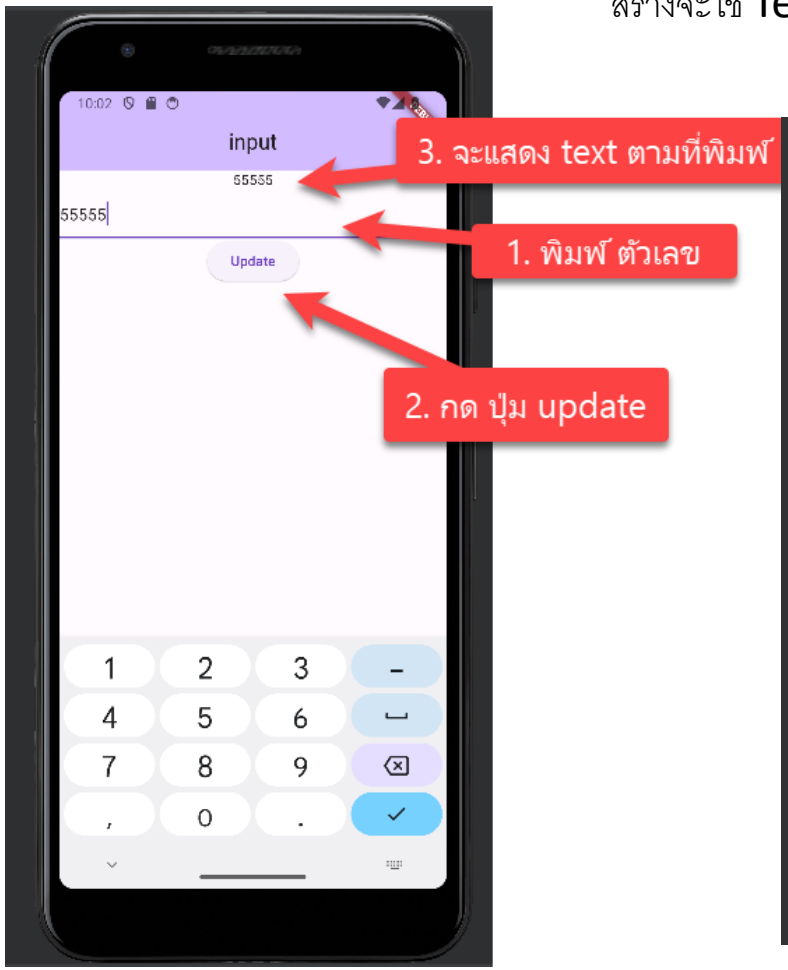


Input Text

- การตั้งค่า input ของ text

สร้างจะใช้ TextFormField และใช้ Controller เป็น id

และจะสร้างตัวแปล controller มารับค่า input ที่สร้าง




```dart
class MyHomePage extends StatefulWidget {
  const MyHomePage({super.key, required this.title});


  final String title;

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

class _MyHomePageState extends State<MyHomePage> {

  TextEditingController _inputText=TextEditingController();

  void _incrementCounter() {
    setState(() {


    });
  }

  @override
  Widget build(BuildContext context) {

    return Scaffold(
      appBar: AppBar(

        backgroundColor: Theme.of(context).colorScheme.inversePrimary,
```

1.ประกาศตัวแปร Controller

```dart
@override
Widget build(BuildContext context) {

  return Scaffold(
    appBar: AppBar(

      backgroundColor: Theme.of(context).colorScheme.inversePrimary,

      title: Center(child:Text('input')),
    ),  // AppBar
    body: Center(

      child: Column(
        children:<Widget> [
          Text(_inputText.text),
          TextField(
            keyboardType: TextInputType.number,
            controller: _inputText,
          ),  // TextField
          ElevatedButton(onPressed: (){
            setState((){

            });
          }, child: Text('Update'))  // ElevatedButton
        ],  // <Widget>[]
      ),  // Column
    ),  // Center

  );  // Scaffold
}
}
```

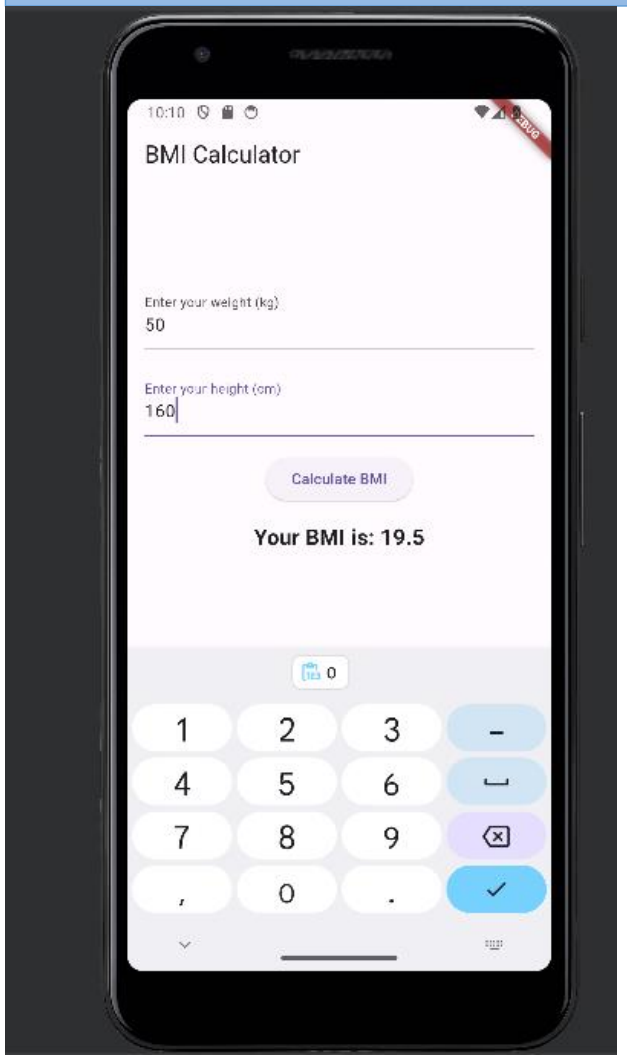
2. สร้างปุ่มเรียกใช้งาน

Text แสดงผล

input text รับค่า

ปุ่มกด

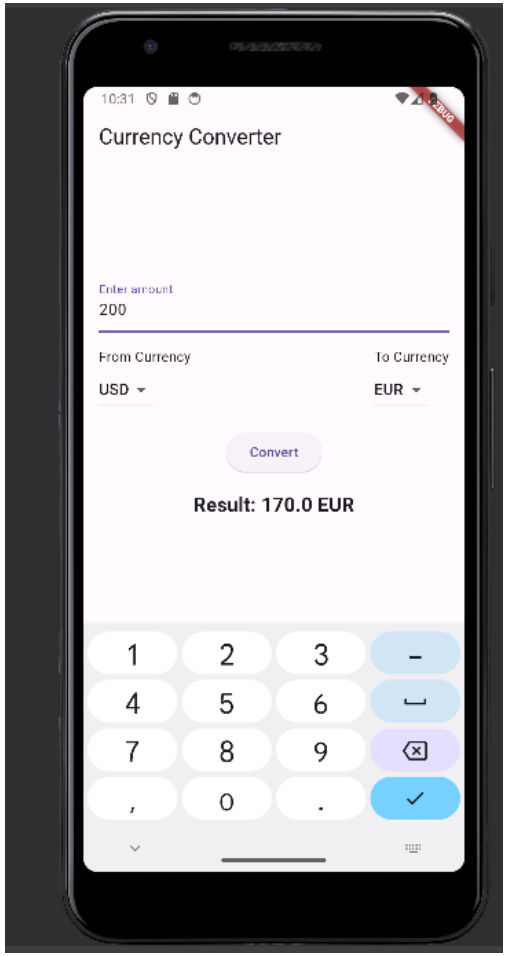
App BMI



- ใช้ ChatGPT ทำการระบุไปว่า อยากได้โค้ดสร้าง **application** คำนวน **BMI** โดยให้กรอก น้ำหนักและส่วนสูง ขอเป็น **flutter**

App Convert Currency

- ถาม Chat GPT อยากได้ **application convert** สกุลเงิน โดยมี **texboxt** กรอกสกุลเงินเข้าไปแล้วแปลงสกุลเงินออกมา ขอเป็น **flutter**



บางที ChatGPT ก็ให้โค้ดมาต้องแก้ เช่น App Convert Currency

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

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

class CurrencyConverterApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: 'Currency Converter',
      theme: ThemeData(
        primarySwatch: Colors.blue,
      ),
      home: CurrencyConverterScreen(),
    );
  }
}

class CurrencyConverterScreen extends StatefulWidget {
  @override
  _CurrencyConverterScreenState createState() =>
      _CurrencyConverterScreenState();
}

class _CurrencyConverterScreenState extends State<CurrencyConverterScre
  final TextEditingController amountController = TextEditingController(
  String fromCurrency = 'USD';
  String toCurrency = 'EUR';
  double result = 0.0;
```

เรียก CurrencyConverterApp

```dart
Widget _buildCurrencyDropdown(
    String label, String selectedCurrency, Function(String) onChanged
  return Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      Text('$label Currency'),
      DropdownButton<String>(
        value: selectedCurrency,
        onChanged: onChanged,
        items: currencies.map((currency) {
          return DropdownMenuItem<String>(
            value: currency,
            child: Text(currency),
          );
        }).toList(),
      ),
    ],
  );
}

void convertCurrency() {
  double amount = double.tryParse(amountController.text) ?? 0.0;

  if (amount > 0) {
    double exchangeRate = getExchangeRate(fromCurrency, toCurrency);
    setState(() {
      result = amount * exchangeRate;
    });
  }
}
```

ตัดออก

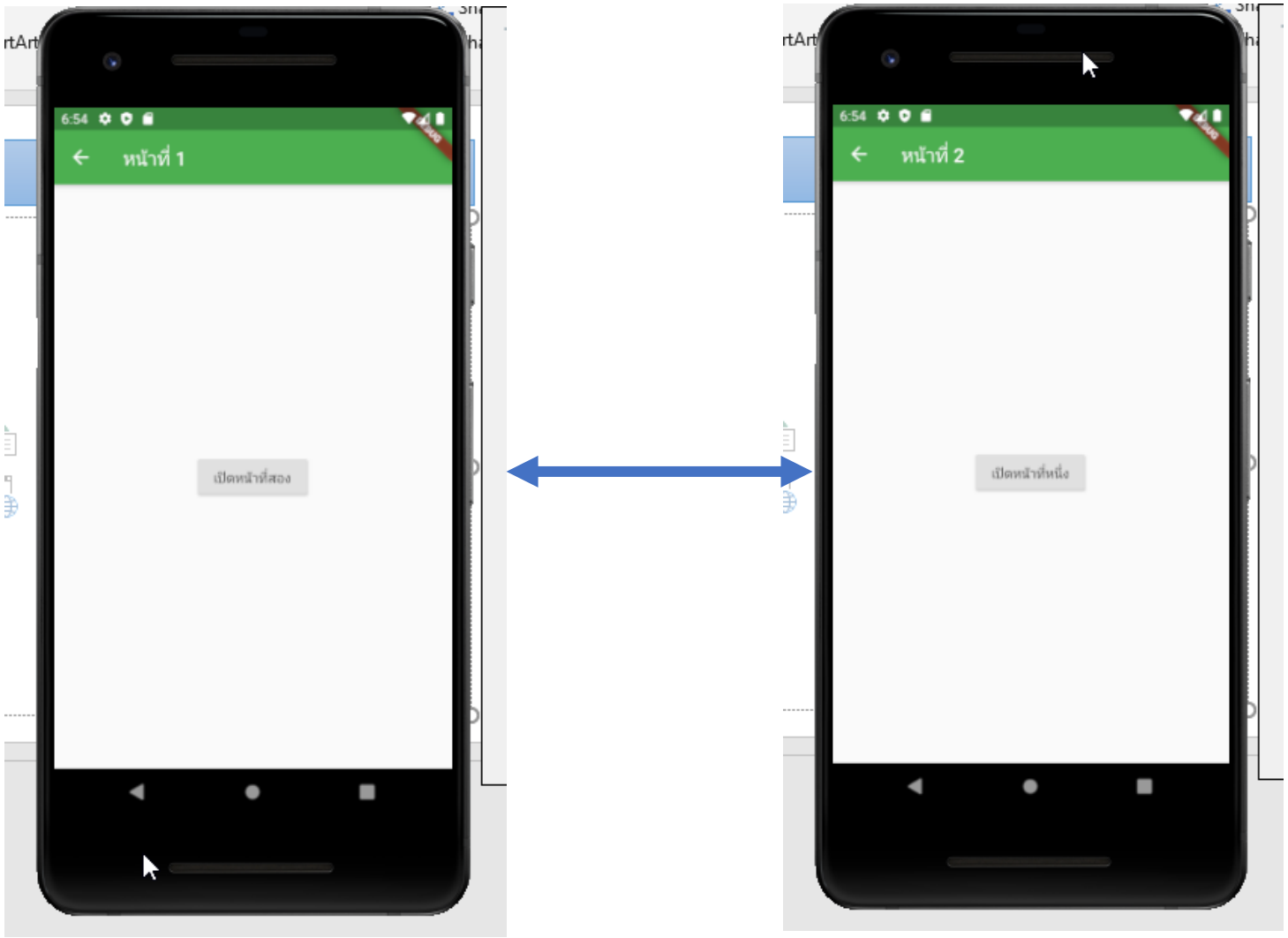
```dart
    double exchangeRate = getExchangeRate(fromCurrency, toCurrency);
    setState(() {
      result = amount * exchangeRate;
    });
  }
}

double getExchangeRate(String fromCurrency, String toCurrency) {
  // สามารถเพิ่มตารางอัตราแลกเปลี่ยนจริงได้
  Map<String, double> exchangeRates = {
    'USD': 1.0,
    'EUR': 0.85,
    'GBP': 0.75,
    'JPY': 110.0,
  };

  if (exchangeRates.containsKey(fromCurrency) &&
      exchangeRates.containsKey(toCurrency)) {
    return exchangeRates[toCurrency] / exchangeRates[fromCurrency];
  } else {
    return 0.0;
  }
}
```

ใส่เครื่องหมาย ! เพื่อเช็ค null ไว้ด้านหลัง]

StatelessWidget ตามโค้ดด้านล่าง

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

void main(){runApp(
  MyStatelessWidget(text: 'StatelessWidget Example to show immutable data')
 );
}

class MyStatelessWidget extends StatelessWidget {
 final String text;
 // constuctor
 MyStatelessWidget({Key key, this.text}) : super(key: key);

 @override
 Widget build(BuildContext context) {
  return Center(
  child: Text(
  text,
  textDirection: TextDirection.ltr,
  ), // Text
  ); // Center
 }
}
```

การใช้ StatefulWidget การทำ route เพื่อเปิดอีกหน้า



เมื่อกดปุ่มแล้วจะเปลี่ยนไปอีกหน้า

การทำ route เพื่อเปิดอีกหน้า

- Main.dart

main.dart | Page2.dart | Page1.dart | register.dart

```dart
import 'package:flutter/material.dart';
import 'Page1.dart';
import 'Page2.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: 'ทดลองการใช้ Route',
    theme: ThemeData(
      primarySwatch: Colors.green,
    ), // ThemeData
    initialRoute: '/',
    routes: {
      // เมื่อ ใช้ "/" route ให้สร้างหน้าแรก
      '/': (context) => Page1(title: "หน้าที่ 1",),
      // เมื่อ ใช้ "/page2" route ให้สร้างหน้าที่สอง
      '/page2': (context) => Page2(title: "หน้าที่ 2",),
    },
  ); // MaterialApp
}
}
```

การทำ route เพื่อเปิดอีกหน้า (ต่อ)

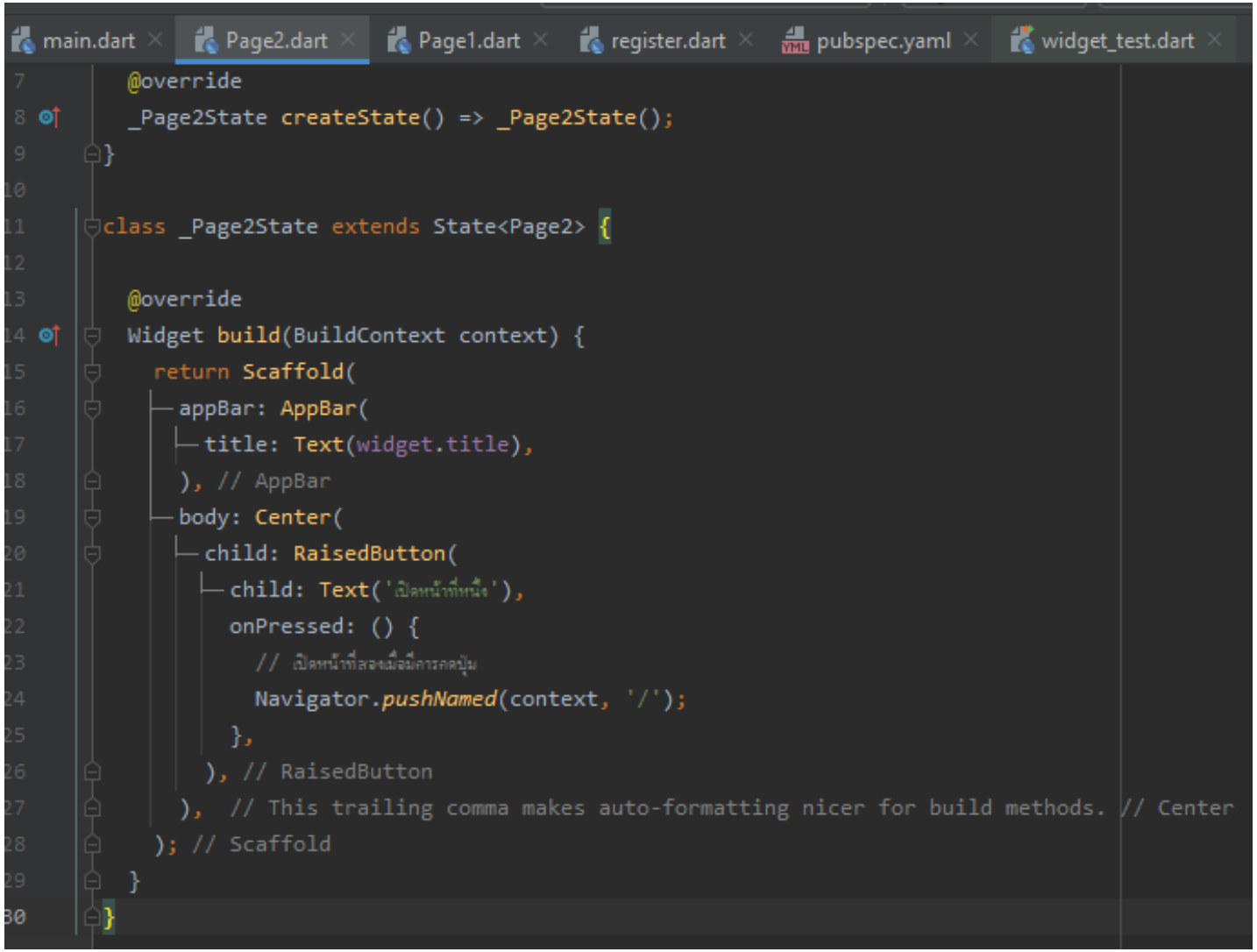
- Page1.dart

main.dart | Page2.dart | Page1.dart | register.dart | pubspec.yaml | widget_test.dart

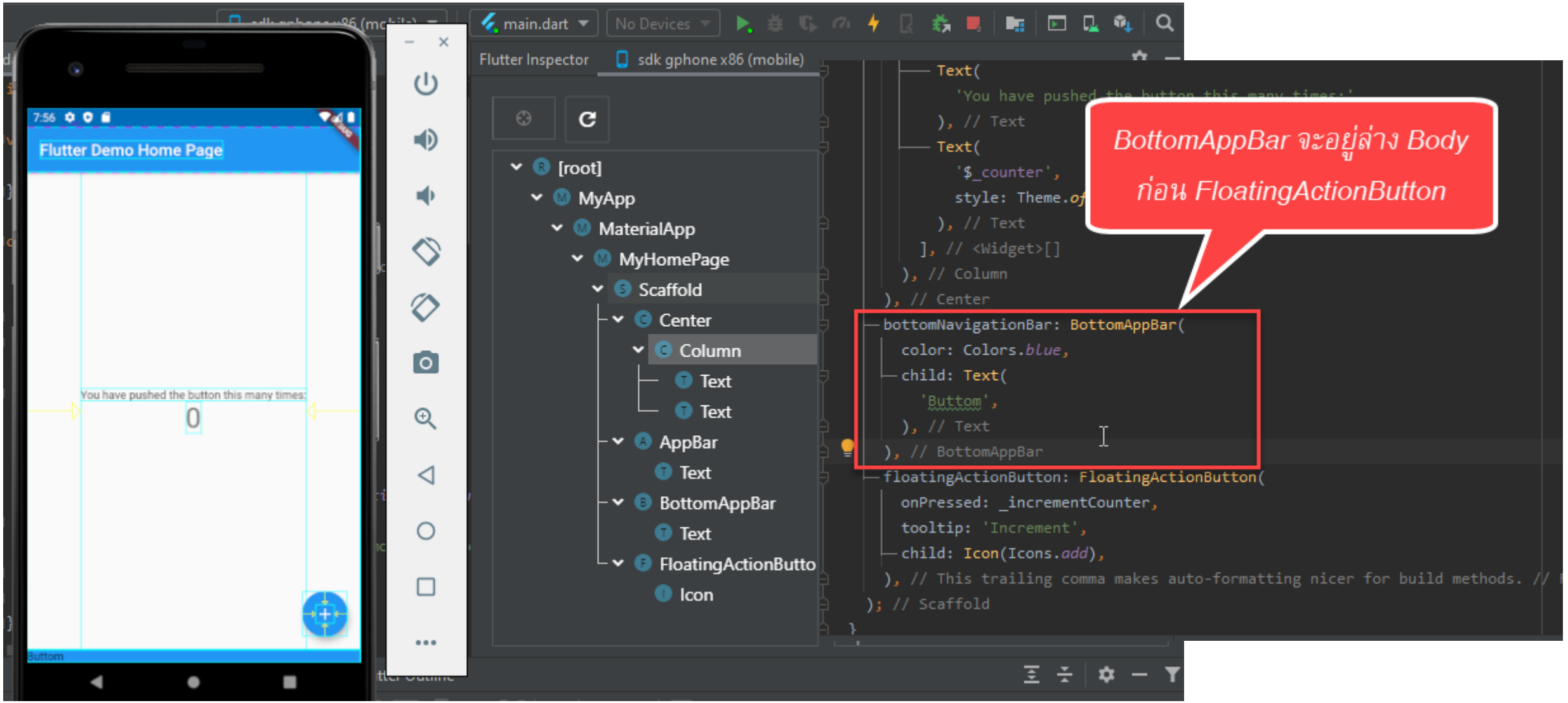
```dart
import 'package:flutter/material.dart';
class Page1 extends StatefulWidget {
  Page1({Key key, this.title}) : super(key: key);

  final String title;

  @override
  _Page1State createState() => _Page1State();
}

class _Page1State extends State<Page1> {

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(widget.title),
      ), // AppBar
      body: Center(
        child: RaisedButton(
          child: Text('เปิดหน้าที่สอง'),
          onPressed: () {
            // เปิดหน้าที่สองเมื่อมีการกดปุ่ม
            Navigator.pushNamed(context, '/page2');
          },
        ), // RaisedButton
      ),  // This trailing comma makes auto-formatting nicer for build methods. // Center
    ); // Scaffold
  }
```

การทำ route เพื่อเปิดอีกหน้า (ต่อ)

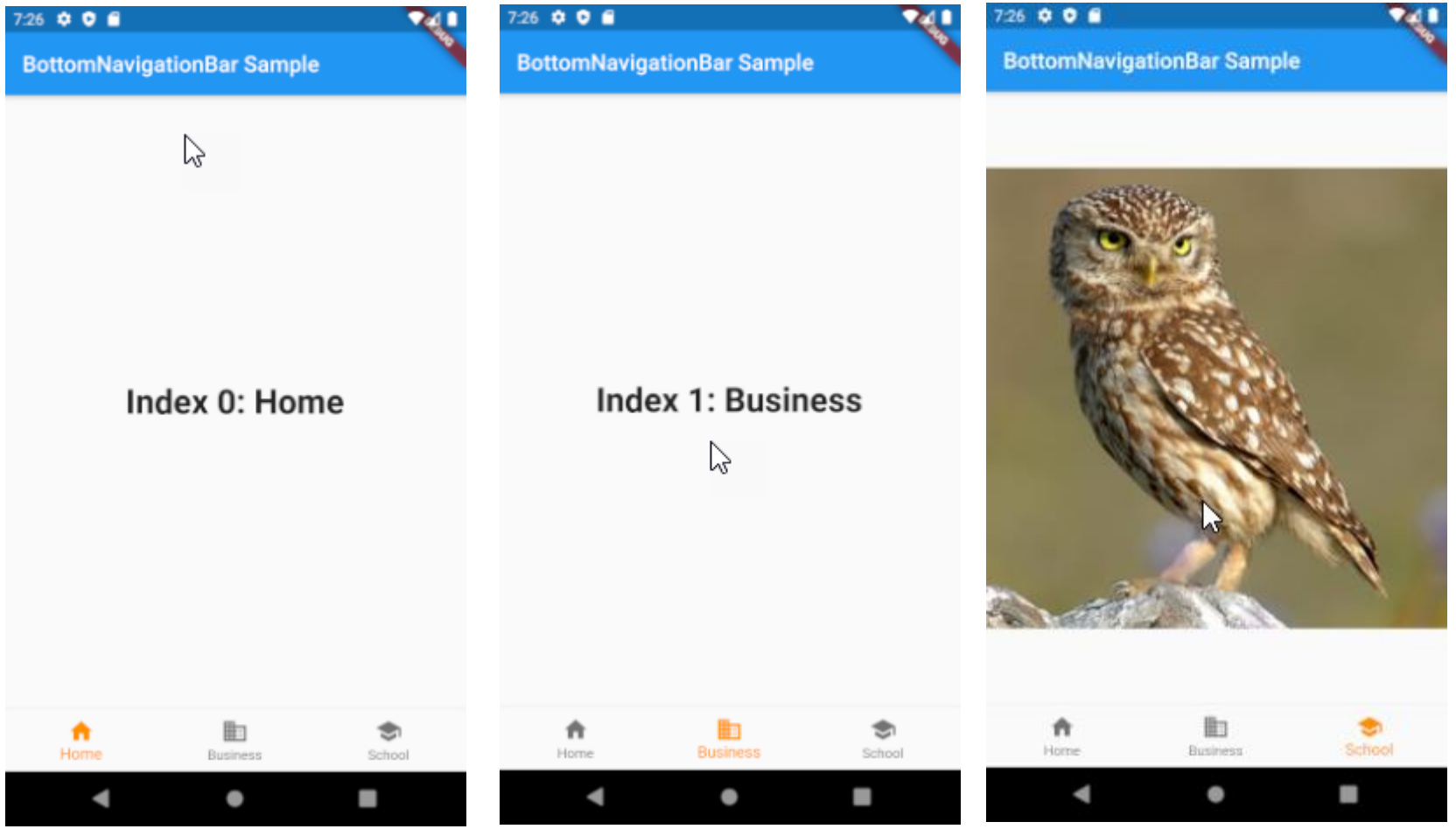
- Page2.dart



BottomAppBar



BottomNavigationBar



- https://api.flutter.dev/flutter/widgets/Icon-class.html หาเปลี่ยนไอคอนได้จากที่นี่
- https://api.flutter.dev/flutter/material/Icons-class.html