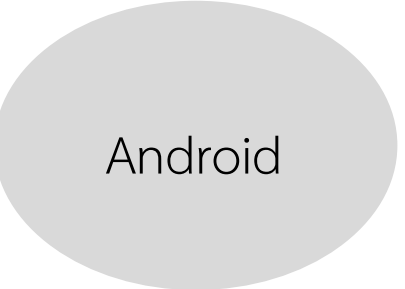
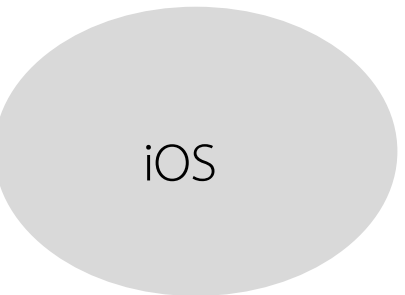
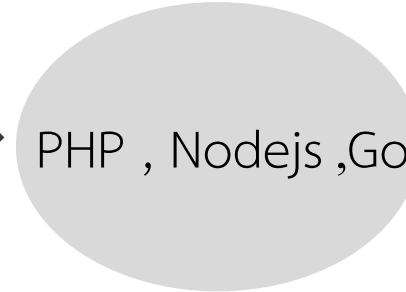
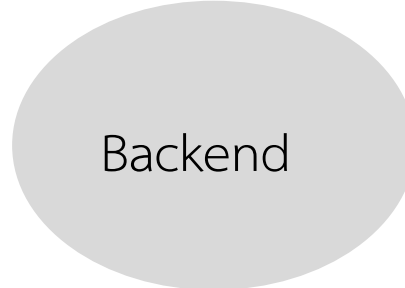
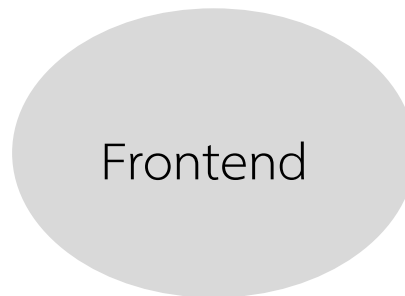


Chapter 1 Introduction to Mobile Application Development part1

ทำความรู้จักกับ

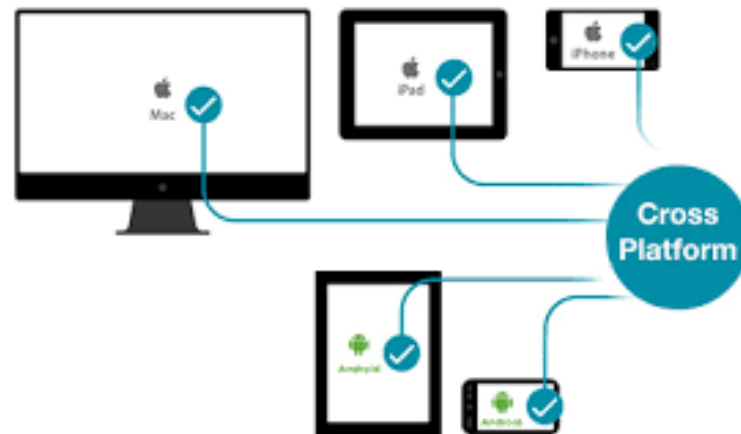
Flutter & Dart





การพัฒนาแอปพลิเคชันแบบ Cross Platform

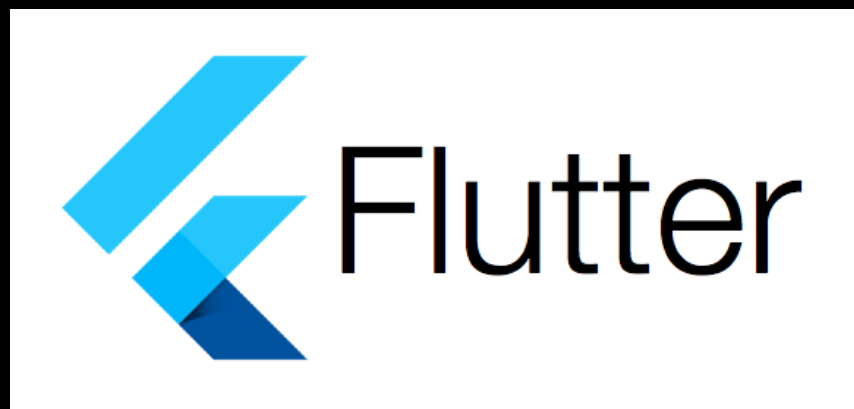
- การพัฒนาแอปพลิเคชันอุปกรณ์เคลื่อนที่แบบข้ามแพลตฟอร์ม (Cross Platform Mobile Application Development) เป็นรูปแบบในการพัฒนาแอปพลิเคชัน โดยการใช้การเขียนโค้ดเพียงครั้งเดียวแต่สามารถใช้งานได้ในทุกแพลตฟอร์ม จึงสามารถรองรับการใช้งานระบบปฏิบัติการต่าง ๆ เช่น Android, iOS และ Windows ได้
- การพัฒนาแอปพลิเคชันเคลื่อนที่แบบข้ามแพลตฟอร์มจะช่วยแก้ปัญหาเรื่องระยะเวลาที่ต้องใช้ในการพัฒนา





ทำความรู้จักกับ Flutter

- Flutter คือเครื่องมือที่ถูกพัฒนาขึ้นโดย Google สำหรับนำมาพัฒนา Mobile Application ที่ทำงานในอุปกรณ์พกพา เช่น smart phone , tablet
- โดย Flutter เป็นชุดโปรแกรมที่เน้นในส่วนของหน้าต่างของแอปพลิเคชัน หรือ User Interface (UI) ให้นักพัฒนาสามารถสร้างและปรับปรุงแก้ไขหน้าต่างของแอปพลิเคชัน ได้สะดวกตามที่ต้องการ





Dart

ทำความรู้จักกับ Flutter

- Flutter มีภาษาโปรแกรมก็คือภาษา Dart ซึ่งภาษา Dart จะถูกใช้ในการสร้าง User Interface หรือการเขียนโปรแกรมส่วนอื่น ๆ เช่น การติดต่อฐานข้อมูล การเข้าถึงและจัดการเกี่ยวกับอุปกรณ์พกพา เช่น กล้องถ่ายรูป และ ตัวสแกนบาร์โค้ด เป็นต้น

ภาษา Dart สามารถนำไปใช้ในการพัฒนา Web Application ได้ด้วยโดยการแปลงภาษา Dart เป็น JavaScript และรันใน Web Browser สามารถเข้าดูรายละเอียดเพิ่มเติมได้ที่ >> <https://flutter.dev/docs/get-started/web>



ทำความรู้จักกับ Flutter

- Flutter นั้นจะทำให้เราเขียนและพัฒนาแอปพลิเคชันได้ง่ายมากยิ่งขึ้น โดยการเขียนแอปพลิเคชันแค่ครั้งเดียวแต่สามารถใช้งานแอปพลิเคชันนั้นได้ในระบบหลัก ๆ ถึง 2 ระบบ คือ iOS และ Android หรือ เรียกว่า Cross Platform





สรุป Flutter

1. ชุดโปรแกรมด้าน User Interface ถูกสร้างโดย Google
2. จัดการส่วนต่าง ๆ ได้ง่ายและสะดวก
3. เป็น Open Source ใช้งานได้ฟรี
4. พัฒนา Mobile Application ให้สามารถใช้งานได้ทั้ง iOS และ Android

คู่มือ flutter จาก wikipedia

Flutter (software) 🌐 30 languages

Article Talk Read Edit View history Tools

From Wikipedia, the free encyclopedia

This article reads like a press release or a news article and may be **largely based on routine coverage**. Please **expand this article** with **properly sourced content** to meet Wikipedia's **quality standards**, **event notability guideline**, or **encyclopedic content policy**. *(January 2023)*

Flutter is an open-source UI software development kit created by Google. It is used to develop cross platform applications from a single codebase for any web browser,^[4] Fuchsia, Android, iOS, Linux, macOS, and Windows.^[5] First described in 2015,^{[6][7]} Flutter was released in May 2017.^[1]

History [edit]


The first version of Flutter was known as "Sky"^[8] and ran on the Android operating system. It was unveiled at the 2015 Dart developer summit^[9] with the stated intent of being able to render consistently at 120 frames per second.^[10] During the keynote of Google Developer Days in Shanghai in September 2018, Google announced Flutter Release Preview 2, the last major release before Flutter 1.0. On December 4, 2018, Flutter 1.0 was released at the Flutter Live event, denoting the first stable version of the framework. On December 11, 2019, Flutter 1.12 was released at the Flutter Interactive event.^[11]

On May 6, 2020, the Dart software development kit (SDK) version 2.8 and Flutter 1.17.0 were released, adding support for the Metal API which improves performance on iOS devices by approximately 50%, as well as new Material widgets and network tracking development tools.

On March 3, 2021, Google released Flutter 2 during an online Flutter Engage event. This major update brought official support for web-based applications with a new Canvas Kit renderer and web specific widgets, early-access desktop application support for Windows, macOS, and Linux and improved Add-to-App APIs.^[12] This release also utilized Dart 2.0 that featured null-safety, which caused many breaking changes and issues with many external packages; however, the Flutter team included instructions and tools to mitigate these issues.^[13]

On September 8, 2021, Dart 2.14 and Flutter 2.5 were released by Google. The update brought improvements to the Android full-screen mode and the latest version of Google's Material Design called Material You. Dart received two new updates, standardizing lint conditions and marking support for Apple Silicon as stable.^{[14][15]}

Flutter



Original author(s) Google

Developer(s) Google and community

Initial release Alpha (v0.0.6) / May 12, 2017; 6 years ago^[1]

Stable release 3.13.0^[2] / 16 August 2023; 2 months ago

Repository github.com/flutter/flutter

Written in C, C++, Dart^[3]

Platform Android, iOS, Google Fuchsia, Web platform, Linux, macOS and Windows

Type Application framework

License New BSD License

Website flutter.dev

[https://en.wikipedia.org/wiki/Flutter_\(software\)](https://en.wikipedia.org/wiki/Flutter_(software))

คู่มือ dart จาก wikipedia

Dart (programming language)

🌐 34 languages ▾

Article Talk

Read Edit View history Tools ▾

From Wikipedia, the free encyclopedia

This article is about a programming language. For the advertising application formerly named Google Dart, see [DoubleClick for Publishers by Google](#). For other uses, see [Dart](#).

Dart is a programming language designed by Lars Bak and Kasper Lund and developed by Google.^[8] It can be used to develop web and mobile apps as well as server and desktop applications.

Dart is an object-oriented, class-based, garbage-collected language with C-style syntax.^[9] It can compile to machine code, JavaScript, or WebAssembly. It supports interfaces, mixins, abstract classes, reified generics and type inference.^[4]

History [edit]

Dart was unveiled at the GOTO conference in Aarhus, Denmark, October 10–12, 2011.^[10] Lars Bak and Kasper Lund founded the project.^[11] Dart 1.0 was released on November 14, 2013.^[12]

Dart had a mixed reception at first. Some criticized the Dart initiative for fragmenting the web because of plans to include a Dart VM in Chrome. Those plans were dropped in 2015 with the Dart 1.9 release. Focus changed to compiling Dart code to JavaScript.^[13]

Dart 2.0 was released in August 2018 with language changes including a type system.^[14]

Dart 2.6 introduced a new extension, `dart2native`. This extended native compilation to the Linux, macOS, and Windows desktop platforms.^[15] Earlier developers could create new tools using only Android or iOS devices. With this extension, developers could deploy a program into self-contained executables. The Dart SDK doesn't need to be installed to run these self-contained executables.^[16] The Flutter toolkit integrates Dart, so it can compile on small services like backend support.^{[17][18]}

Dart 3.0 changed the type system to require sound null safety. This release included new features like records, patterns,^[19] and class modifiers.^[20] Dart 3 also previewed support for Web Assembly.^[21]

Specification [edit]

Dart released the 5th edition of its language specification on April 9, 2021.^[22] This covers all syntax through Dart 2.10. A draft of the 6th edition includes all syntax through 2.13.^[23] [Accepted proposals](#) for the specification and [drafts of potential features](#) can be found in the Dart language repository on GitHub.^[24]

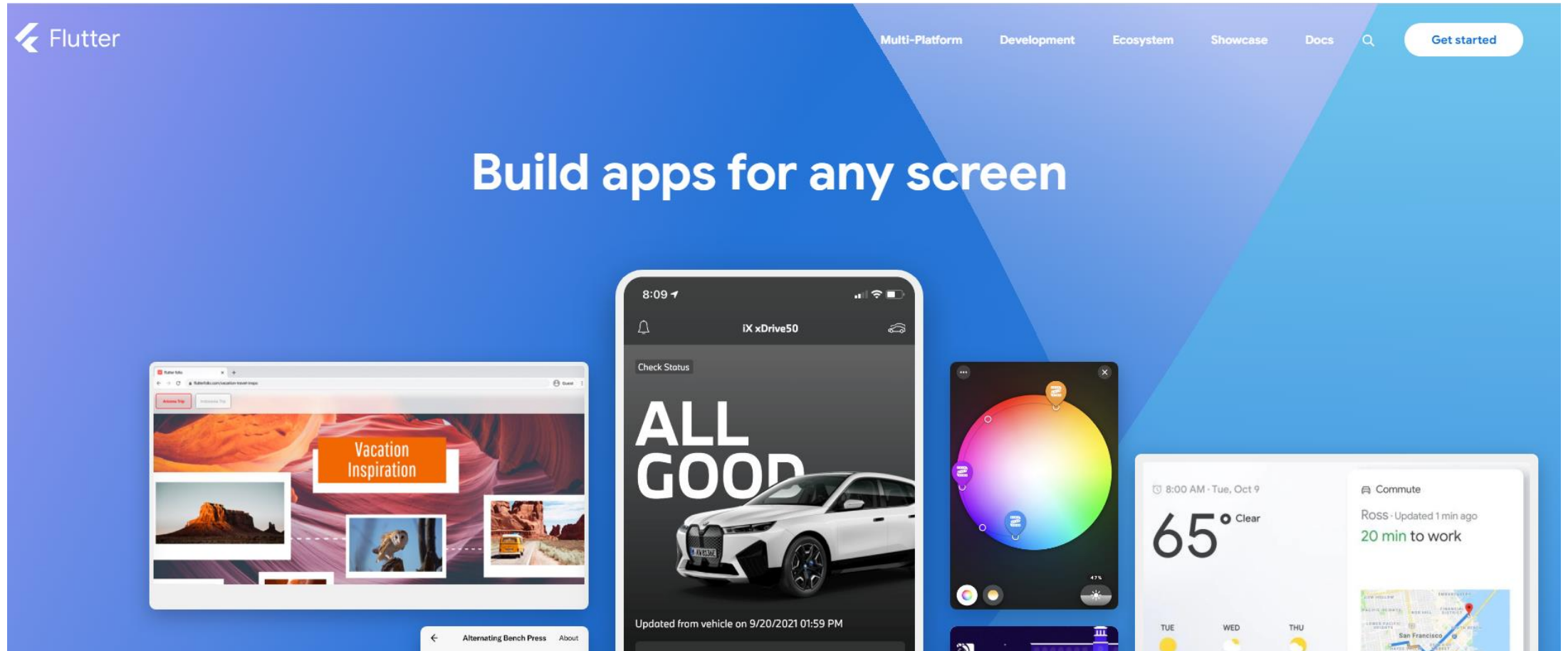
Dart



Paradigm	Multi-paradigm: functional, imperative, object-oriented, reflective ^[1]
Designed by	Lars Bak, Kasper Lund
Developer	Google
First appeared	October 10, 2011; 12 years ago ^[2]
Stable release	3.1.5 ^[3] / 25 October 2023; 19 days ago
Typing discipline	1.x: Optional 2.x: Inferred ^[4] (static, strong)
Platform	Cross-platform
OS	Cross-platform
License	BSD
Filename extensions	.dart
Website	dart.dev
Major implementations	Dart VM, dart2native, dart2js, DDC, Flutter
Influenced by	C, C++, C#, Erlang, Java, JavaScript, Ruby, Smalltalk, Strongtalk, ^[5] TypeScript ^[6]

[https://en.wikipedia.org/wiki/Dart_\(programming_language\)](https://en.wikipedia.org/wiki/Dart_(programming_language))

เข้าเว็บ <https://flutter.dev/>



🔗 System requirement

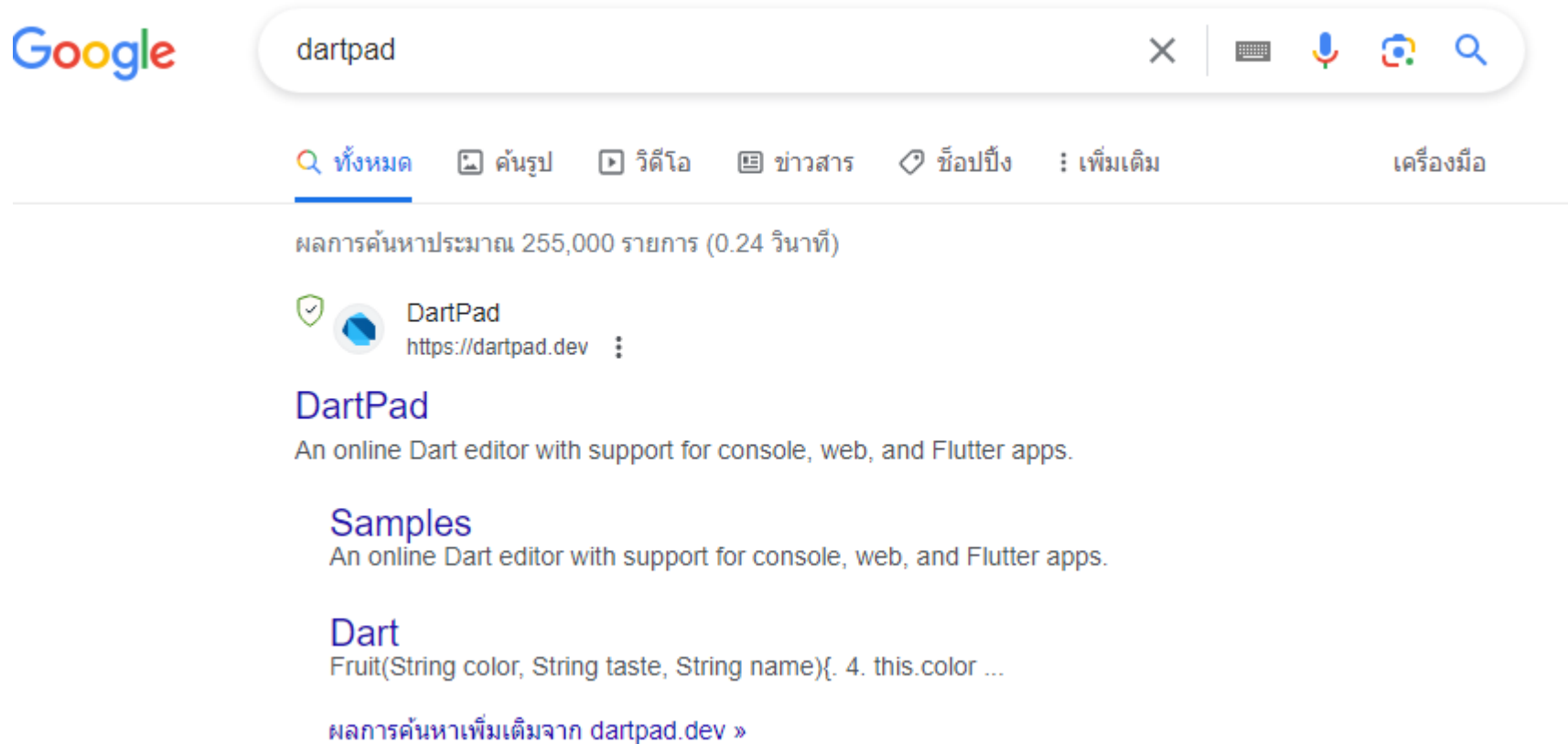
The screenshot shows the Flutter website's 'Install' page. At the top left is the Flutter logo. On the right, there are navigation links: 'Multi-Platform', 'Development', 'Ecosystem', and 'Showcase'. A dark blue banner at the top right contains the text 'Flutter 3.13 scrolls into view! Check out the announcement and the what's new page'. On the left side, there is a navigation menu with the following items: 'Get started' (with a sub-menu: '1. Install', '2. Set up an editor', '3. Test drive', '4. Write your first app', '5. Learn more'), 'From another platform?' (with a sub-link: 'Dart language overview'), 'Stay up to date', 'Samples & codelabs', 'App solutions', 'User interface' (with sub-links: 'Introduction', 'Widget catalog', 'Layout', 'Design & theming', 'Interactivity'). The main content area is titled 'Install' and includes a breadcrumb 'Get started > Install'. Below the title, it says 'Select the operating system on which you are installing Flutter:' and displays four buttons for 'Windows', 'macOS', 'Linux', and 'ChromeOS'. A yellow warning box contains the text: 'Important: If you develop apps in China, check out using Flutter in China.' At the bottom right of the main content area, there is a link 'Set up an editor'.

ไปที่ Get started > Install

ไปที่ Introduction เพื่อทดลองรันโค้ด

The screenshot shows the Flutter website's introduction page. At the top left is the Flutter logo. On the right, there are navigation links: Multi-Platform, Development, Ecosystem, and Showcase. A dark blue banner at the top right contains the text "Flutter 3.13 scrolls into view! Check out the announcement and the what's new page". On the left side, there is a vertical navigation menu with categories like "Get started", "Stay up to date", "Samples & codelabs", "App solutions", "User interface" (with "Introduction" highlighted), "Widget catalog", "Layout", "Design & theming", "Interactivity", "Assets & media", "Navigation & routing", "Animations & transitions", "Accessibility & internationalization", and "Beyond UI". The main content area is titled "Building user interfaces with Flutter" and includes a sub-section "Hello world" with a code editor snippet showing the first line of a Dart file: `import 'package:flutter/material.dart';`. A note box highlights that Flutter widgets are built using a modern framework inspired by React, and a note suggests checking out basic layout codelabs and adding interactivity to a Flutter app.

ลองใช้ DartPad > ค้น google ด้วยคำว่า dartpad




The image shows a Google search interface. The search bar contains the text "dartpad". Below the search bar, there are navigation icons for "ทั้งหมด" (All), "ค้นรูป" (Images), "วิดีโอ" (Videos), "ข่าวสาร" (News), "ข้อป้ิ่ง" (Maps), "เพิ่มเติม" (More), and "เครื่องมือ" (Tools). The search results show approximately 255,000 results in 0.24 seconds. The top result is "DartPad" with the URL "https://dartpad.dev". Below the title, there are three sub-sections: "DartPad" (An online Dart editor with support for console, web, and Flutter apps.), "Samples" (An online Dart editor with support for console, web, and Flutter apps.), and "Dart" (Fruit(String color, String taste, String name){. 4. this.color ...). At the bottom, there is a link to "ผลการค้นหาเพิ่มเติมจาก dartpad.dev »".

Google

dartpad

ทั้งหมด ค้นรูป วิดีโอ ข่าวสาร ข้อป้ิ่ง เพิ่มเติม เครื่องมือ

ผลการค้นหาประมาณ 255,000 รายการ (0.24 วินาที)

 DartPad
https://dartpad.dev

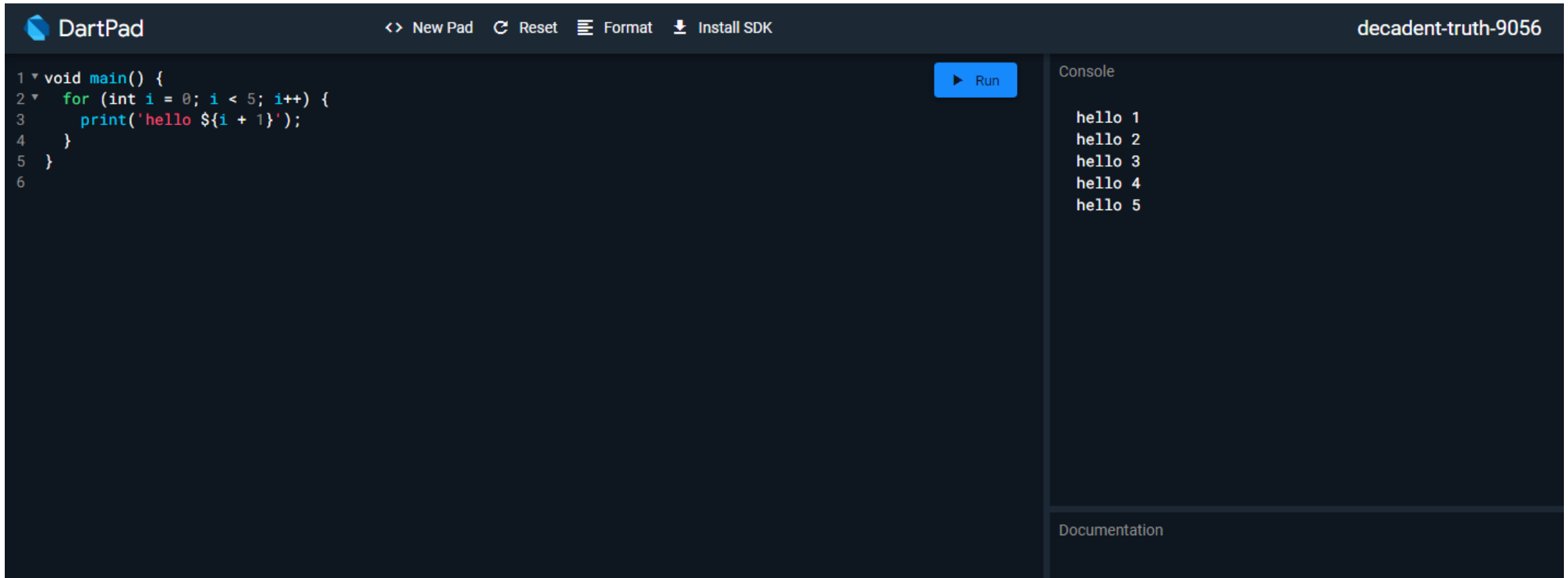
DartPad
An online Dart editor with support for console, web, and Flutter apps.

Samples
An online Dart editor with support for console, web, and Flutter apps.

Dart
Fruit(String color, String taste, String name){. 4. this.color ...

ผลการค้นหาเพิ่มเติมจาก dartpad.dev »

เข้าเว็บ <https://dartpad.dev/>



The image shows a screenshot of the DartPad web IDE. The interface is dark-themed and includes a top navigation bar with the DartPad logo, a username 'decadent-truth-9056', and menu items: '<> New Pad', 'Reset', 'Format', and 'Install SDK'. The main editor area contains the following Dart code:

```
1 void main() {  
2   for (int i = 0; i < 5; i++) {  
3     print('hello ${i + 1}');  
4   }  
5 }  
6
```

To the right of the code editor is a 'Run' button. Below the code editor is a 'Console' panel displaying the output of the program:

```
hello 1  
hello 2  
hello 3  
hello 4  
hello 5
```

At the bottom of the interface is a 'Documentation' panel.

ลอง DartPad

The screenshot shows the DartPad web editor interface. At the top, there is a navigation bar with the DartPad logo and menu items: '<> New Pad', 'Reset', 'Format', and 'Install SDK'. The current file is named 'Hello World'. On the right side of the top bar, there is a 'Samples' dropdown menu and a three-dot menu icon. Below the navigation bar, the main editor area is split into two panels. The left panel contains the Dart source code for a 'Hello World' program. The right panel is divided into two sections: 'Console' and 'Documentation'. The 'Console' section shows the output of the program, which is 'hello 0', 'hello 1', 'hello 2', and 'hello 3' on separate lines. The 'Documentation' section is currently empty.

```
1 // Copyright 2015 the Dart project authors. All rights reserved.  
2 // Use of this source code is governed by a BSD-style license  
3 // that can be found in the LICENSE file.  
4  
5 void main() {  
6   for (var i = 0; i < 4; i++) {  
7     print('hello $i');  
8   }  
9 }  
10
```

Run

Console

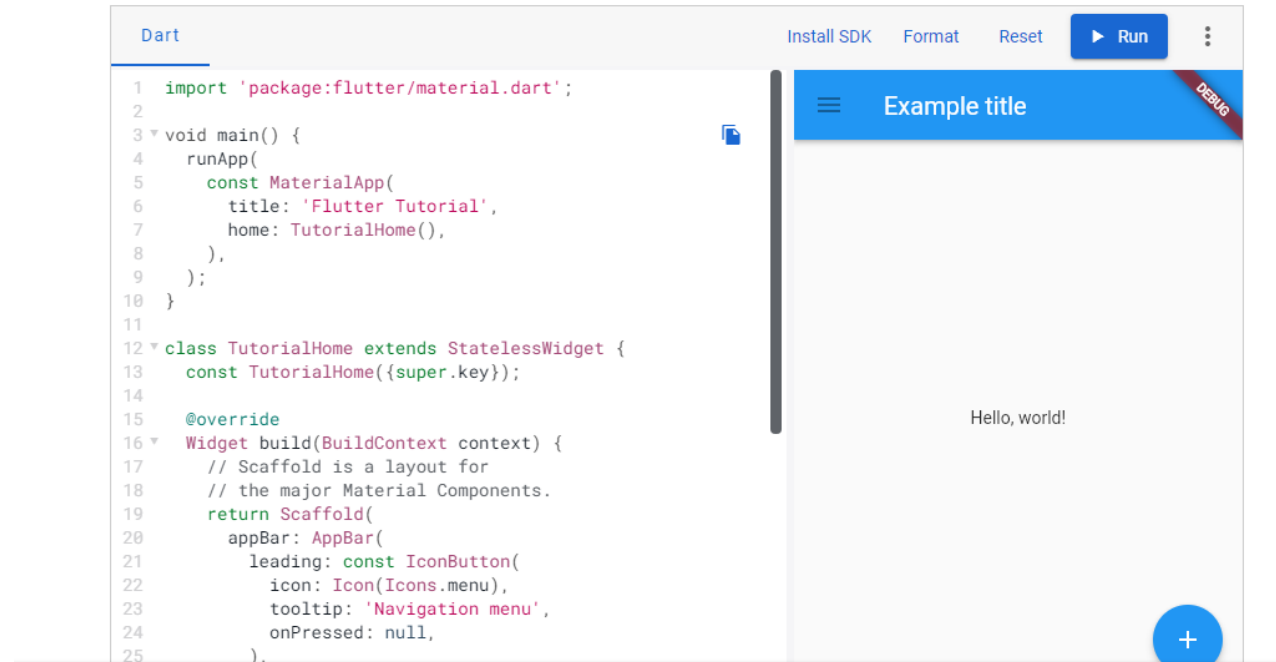
```
hello 0  
hello 1  
hello 2  
hello 3
```

Documentation

Copy โค้ดจาก Introduction to widgets

Using Material Components

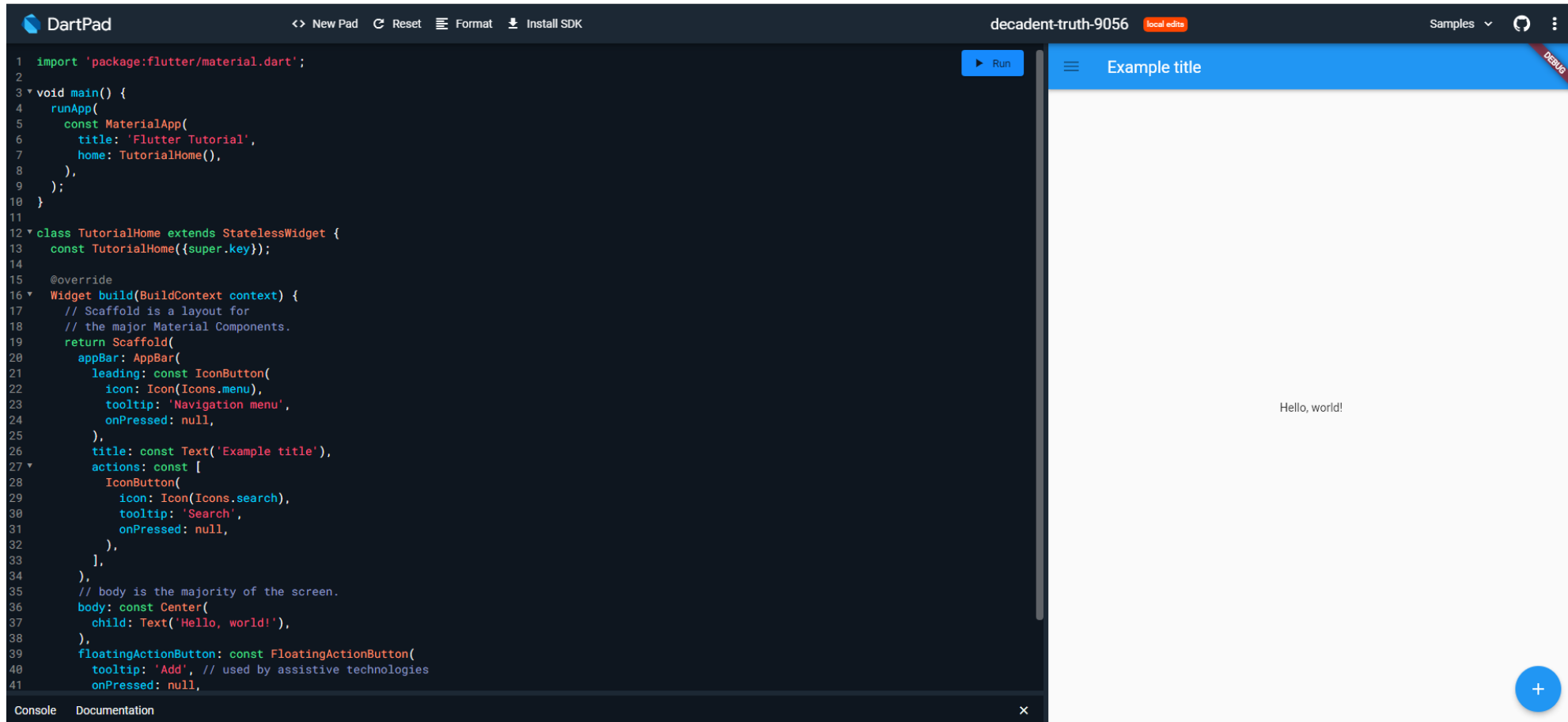
Flutter provides a number of widgets that help you build apps that follow Material Design. A Material app starts with the `MaterialApp` widget, which builds a number of useful widgets at the root of your app, including a `Navigator`, which manages a stack of widgets identified by strings, also known as “routes”. The `Navigator` lets you transition smoothly between screens of your application. Using the `MaterialApp` widget is entirely optional but a good practice.



The screenshot shows an IDE window with a Dart editor on the left and a Flutter app preview on the right. The Dart code defines a `main` function that runs a `MaterialApp` with a `TutorialHome` widget. The `TutorialHome` widget extends `StatelessWidget` and overrides the `build` method to return a `Scaffold` with an `AppBar` containing a `IconButton` with a menu icon. The app preview shows a blue `AppBar` with the title "Example title" and a hamburger menu icon. The main content area is white and contains the text "Hello, world!". A blue circular button with a white plus sign is visible in the bottom right corner of the preview. The IDE interface includes a "Dart" tab, "Install SDK", "Format", "Reset", and "Run" buttons, and a "DEBUG" indicator in the top right corner of the preview area.

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const MaterialApp(
6       title: 'Flutter Tutorial',
7       home: TutorialHome(),
8     ),
9   );
10 }
11
12 class TutorialHome extends StatelessWidget {
13   const TutorialHome({super.key});
14
15   @override
16   Widget build(BuildContext context) {
17     // Scaffold is a layout for
18     // the major Material Components.
19     return Scaffold(
20       appBar: AppBar(
21         leading: const IconButton(
22           icon: Icon(Icons.menu),
23           tooltip: 'Navigation menu',
24           onPressed: null,
25         ),
```

ใส่ใน dartpad จากนั้นกดปุ่ม Run



The image shows the DartPad web interface. On the left is a code editor with the following Dart code:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const MaterialApp(
6       title: 'Flutter Tutorial',
7       home: TutorialHome(),
8     ),
9   );
10 }
11
12 class TutorialHome extends StatelessWidget {
13   const TutorialHome({super.key});
14
15   @override
16   Widget build(BuildContext context) {
17     // Scaffold is a layout for
18     // the major Material Components.
19     return Scaffold(
20       appBar: AppBar(
21         leading: const IconButton(
22           icon: Icon(Icons.menu),
23           tooltip: 'Navigation menu',
24           onPressed: null,
25         ),
26         title: const Text('Example title'),
27         actions: const [
28           IconButton(
29             icon: Icon(Icons.search),
30             tooltip: 'Search',
31             onPressed: null,
32           ),
33         ],
34       ),
35       // body is the majority of the screen.
36       body: const Center(
37         child: Text('Hello, world!'),
38       ),
39       floatingActionButton: const FloatingActionButton(
40         tooltip: 'Add', // used by assistive technologies
41         onPressed: null,
```

On the right is a preview window titled "Example title" showing the rendered application. The app has a blue header with the title "Example title" and a hamburger menu icon. The main content area is white and contains the text "Hello, world!". There is a red "debug" label in the top right corner of the preview area and a blue "+" button in the bottom right corner.

ทำความเข้าใจโครงสร้างของแอปคร่าว ๆ

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(MaterialApp(title: 'Flutter Tutorial', home: TutorialHome()));
5 }
6
7 class TutorialHome extends StatelessWidget {
8   @override
9   Widget build(BuildContext context) {
10    // Scaffold is a layout for
11    // the major Material Components.
12    return Scaffold(
13      appBar: AppBar(
14        leading: IconButton(
15          icon: Icon(Icons.menu),
16          tooltip: 'Navigation menu',
17          onPressed: null,
18        ),
19        title: Text('Example title'),
20        actions: <Widget>[
```

import.

Function หลัก.

main()
↳ runApp
↳ MaterialApp...
↳ title.
↳ home

ทดลองแก้ไขข้อความ Hello, world!

The image shows the DartPad web editor interface. The top navigation bar includes the DartPad logo, menu items like 'New Pad', 'Reset', 'Format', and 'Install SDK', the user ID 'decadent-truth-9056', a 'local edits' indicator, and a 'Samples' dropdown. The main editor area on the left contains Dart code for a Flutter widget. The code defines a Scaffold with an AppBar containing a menu icon and a title 'Example title'. The main body of the Scaffold contains a Center widget with a Text widget displaying 'Sawaddee'. A 'Run' button is located above the code. The right side of the interface shows a preview of the app running in a mobile-like environment, with a blue header bar containing the title 'Example title' and a 'DEBUG' indicator. The main content area of the preview shows the text 'Sawaddee' centered on a white background. A blue circular button with a white plus sign is visible in the bottom right corner of the preview area. At the bottom of the editor, there are tabs for 'Console' and 'Documentation'.

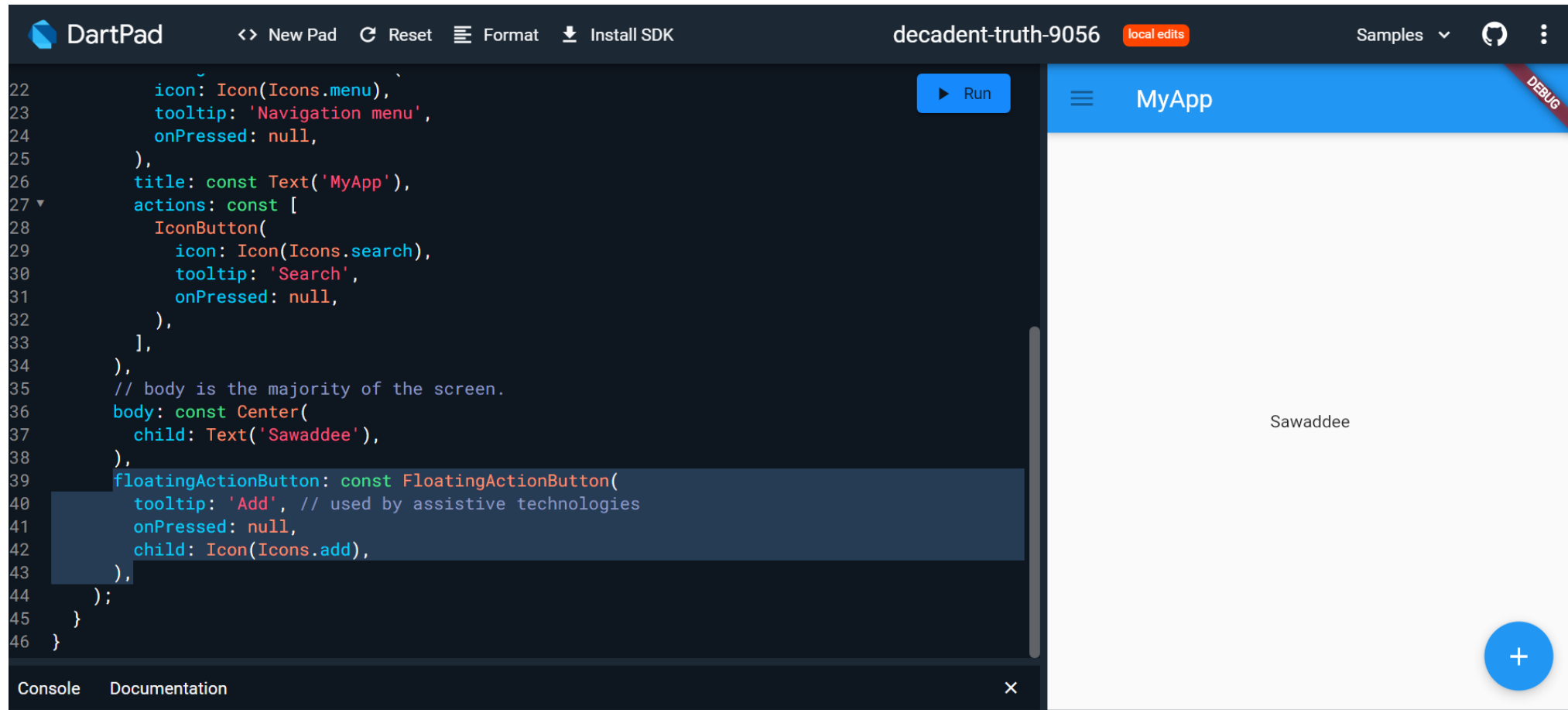
```
14  
15 @override  
16 Widget build(BuildContext context) {  
17   // Scaffold is a layout for  
18   // the major Material Components.  
19   return Scaffold(  
20     appBar: AppBar(  
21       leading: const IconButton(  
22         icon: Icon(Icons.menu),  
23         tooltip: 'Navigation menu',  
24         onPressed: null,  
25       )),  
26     title: const Text('Example title'),  
27     actions: const [  
28       IconButton(  
29         icon: Icon(Icons.search),  
30         tooltip: 'Search',  
31         onPressed: null,  
32       )),  
33     ],  
34   ),  
35   // body is the majority of the screen.  
36   body: const Center(  
37     child: Text('Sawaddee'),  
38   )),  
39   floatingActionButton: const FloatingActionButton(  
40     child: Icon(Icons.add),  
41     onPressed: null,  
42   )),  
43 
```


ทดลองแก้ไขข้อความ appBar Title

The image shows the DartPad web IDE interface. The top bar includes the DartPad logo, navigation options like 'New Pad', 'Reset', 'Format', and 'Install SDK', the user name 'decadent-truth-9056', a 'local edits' indicator, and a 'Samples' dropdown. The main editor area displays Dart code for a Flutter widget. The code defines a blue AppBar with a menu icon and a title 'MyApp'. The main body of the app is white and contains the text 'Sawaddee'. A 'Run' button is visible next to the code. The bottom of the interface has a 'Console' and 'Documentation' section.

```
4
5 @override
6 Widget build(BuildContext context) {
7   // Scaffold is a layout for
8   // the major Material Components.
9   return Scaffold(
10    appBar: AppBar(
11      leading: const IconButton(
12        icon: Icon(Icons.menu),
13        tooltip: 'Navigation menu',
14        onPressed: null,
15      ),
16      title: const Text('MyApp'),
17      actions: const [
18        IconButton(
19          icon: Icon(Icons.search),
20          tooltip: 'Search',
21          onPressed: null,
22        ),
23      ],
24    ),
25    // body is the majority of the screen.
26    body: const Center(
27      child: Text('Sawaddee'),
28    ),
29    floatingActionButton: const FloatingActionButton(
```

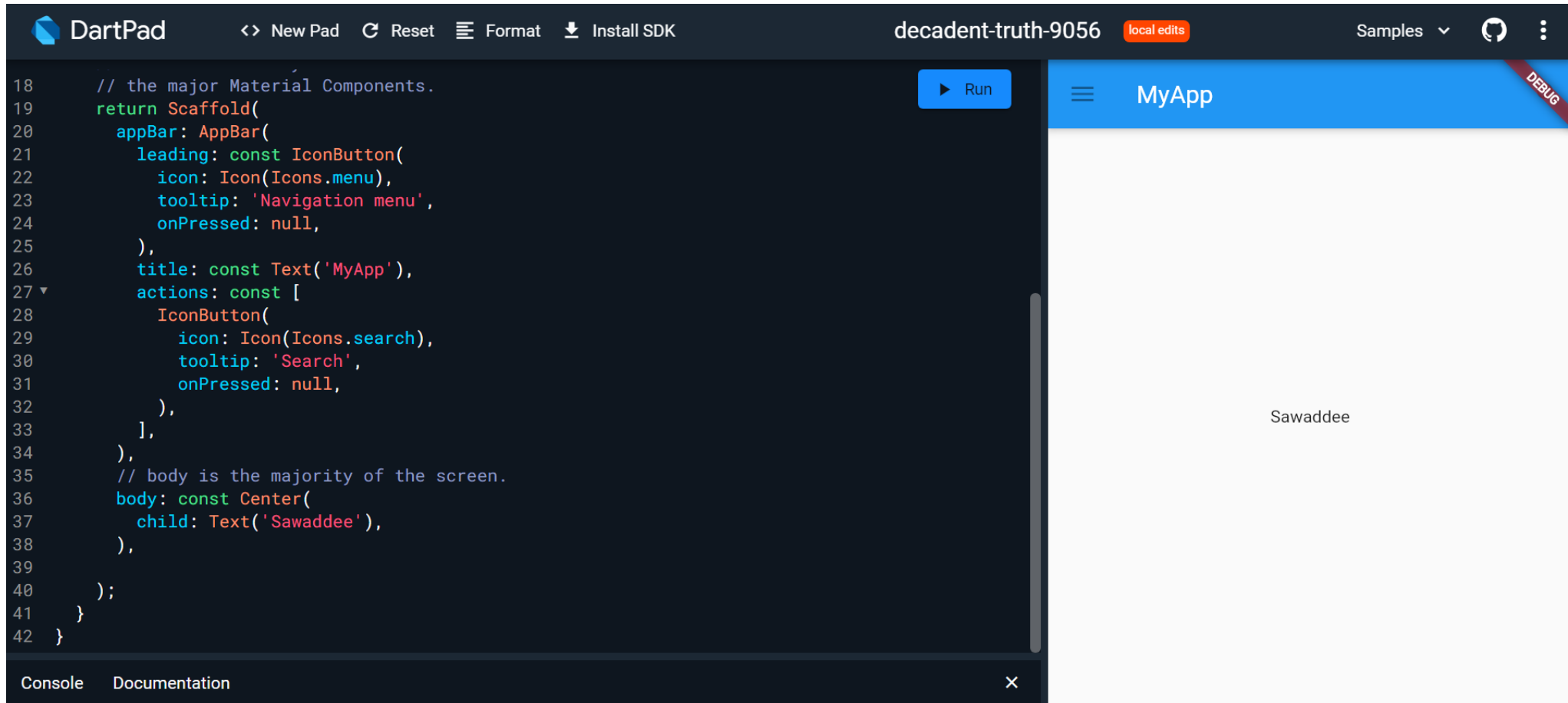
ลองเอา FloatingActionButton ออก



The image shows the DartPad IDE interface. The left pane contains Dart code for a Flutter application. The right pane shows the rendered app with a blue header, a white body containing the text "Sawaddee", and a blue floating action button with a white plus sign in the bottom right corner. A "DEBUG" banner is visible in the top right of the app view.

```
22     icon: Icon(Icons.menu),
23     tooltip: 'Navigation menu',
24     onPressed: null,
25   ),
26   title: const Text('MyApp'),
27   actions: const [
28     IconButton(
29       icon: Icon(Icons.search),
30       tooltip: 'Search',
31       onPressed: null,
32     ),
33   ],
34 ),
35 // body is the majority of the screen.
36 body: const Center(
37   child: Text('Sawaddee'),
38 ),
39 floatingActionButton: const FloatingActionButton(
40   tooltip: 'Add', // used by assistive technologies
41   onPressed: null,
42   child: Icon(Icons.add),
43 ),
44 );
45 }
46 }
```

จะได้ผลการ run ดังภาพ



```
18 // the major Material Components.
19 return Scaffold(
20   appBar: AppBar(
21     leading: const IconButton(
22       icon: Icon(Icons.menu),
23       tooltip: 'Navigation menu',
24       onPressed: null,
25     ),
26     title: const Text('MyApp'),
27     actions: const [
28       IconButton(
29         icon: Icon(Icons.search),
30         tooltip: 'Search',
31         onPressed: null,
32       ),
33     ],
34   ),
35   // body is the majority of the screen.
36   body: const Center(
37     child: Text('Sawaddee'),
38   ),
39 );
40 }
41 }
42 }
```

MyApp

Sawaddee

Console Documentation

ลองเอาส่วน Scaffold ออก

The image shows the DartPad IDE interface. On the left, a code editor displays a Dart function `@override Widget build(BuildContext context) {` that returns a `Scaffold` widget. The `Scaffold` is configured with an `AppBar` containing a menu icon and a title 'Example title', a list of actions including a search icon, and a `floatingActionButton` with an 'Add' icon. The `body` of the `Scaffold` is a `Center` widget containing a `Text` widget with the text 'Hello, world!'. A 'Run' button is visible next to the code. On the right, the rendered application is shown, featuring a blue `AppBar` with a menu icon, the title 'Example title', and a 'DEBUG' indicator. The main content area is white and contains the text 'Hello, world!' centered. A blue circular `floatingActionButton` with a white plus sign is located in the bottom right corner of the application view.

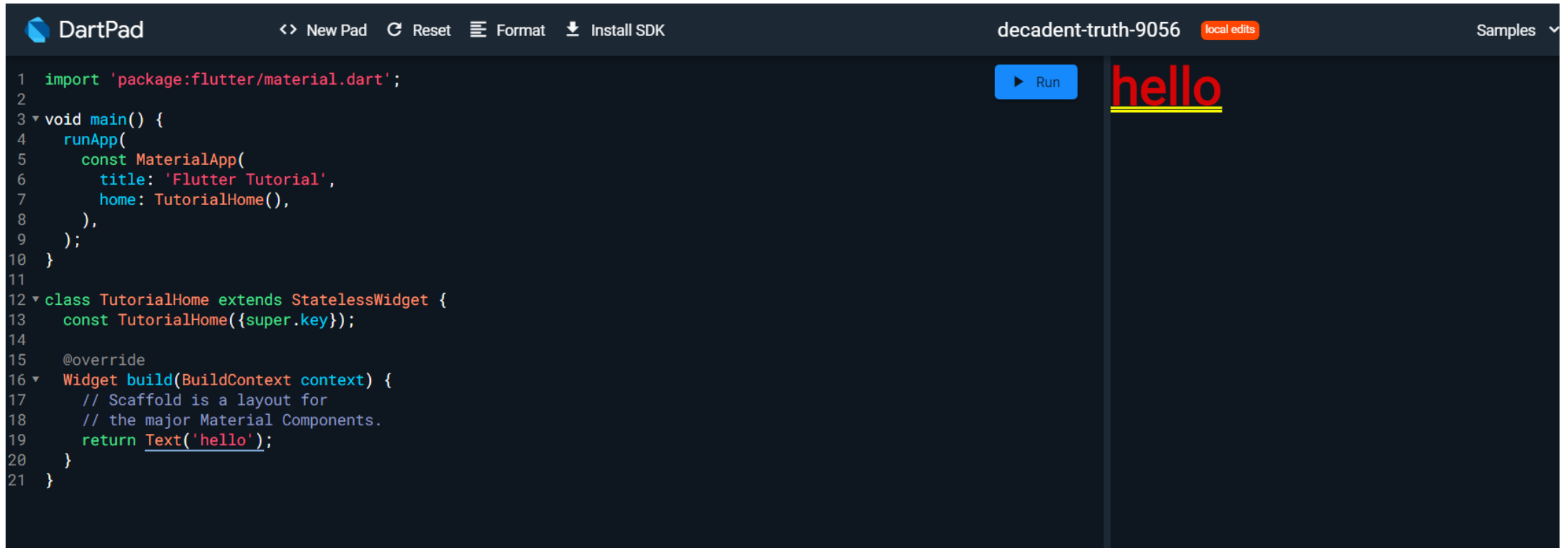
```
15 @override
16 Widget build(BuildContext context) {
17   // Scaffold is a layout for
18   // the major Material Components.
19   return Scaffold(
20     appBar: AppBar(
21       leading: const IconButton(
22         icon: Icon(Icons.menu),
23         tooltip: 'Navigation menu',
24         onPressed: null,
25       ),
26       title: const Text('Example title'),
27       actions: const [
28         IconButton(
29           icon: Icon(Icons.search),
30           tooltip: 'Search',
31           onPressed: null,
32         ),
33       ],
34     ),
35     // body is the majority of the screen.
36     body: const Center(
37       child: Text('Hello, world!'),
38     ),
39     floatingActionButton: const FloatingActionButton(
40       tooltip: 'Add', // used by assistive technologies
41       onPressed: null,
42       child: Icon(Icons.add),
43     ),
44   );
45 }
46 }
```

และเพิ่มโค้ดค้างภาพ

```
DartPad <> New Pad ⌂ Reset ☰ Format ⬇ Install SDK

1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const MaterialApp(
6       title: 'Flutter Tutorial',
7       home: TutorialHome(),
8     ),
9   );
10 }
11
12 class TutorialHome extends StatelessWidget {
13   const TutorialHome({super.key});
14
15   @override
16   Widget build(BuildContext context) {
17     // Scaffold is a layout for
18     // the major Material Components.
19     return Text('hello');
20   }
21 }
```

ทดลอง run จะได้ผลการ run ดังภาพ



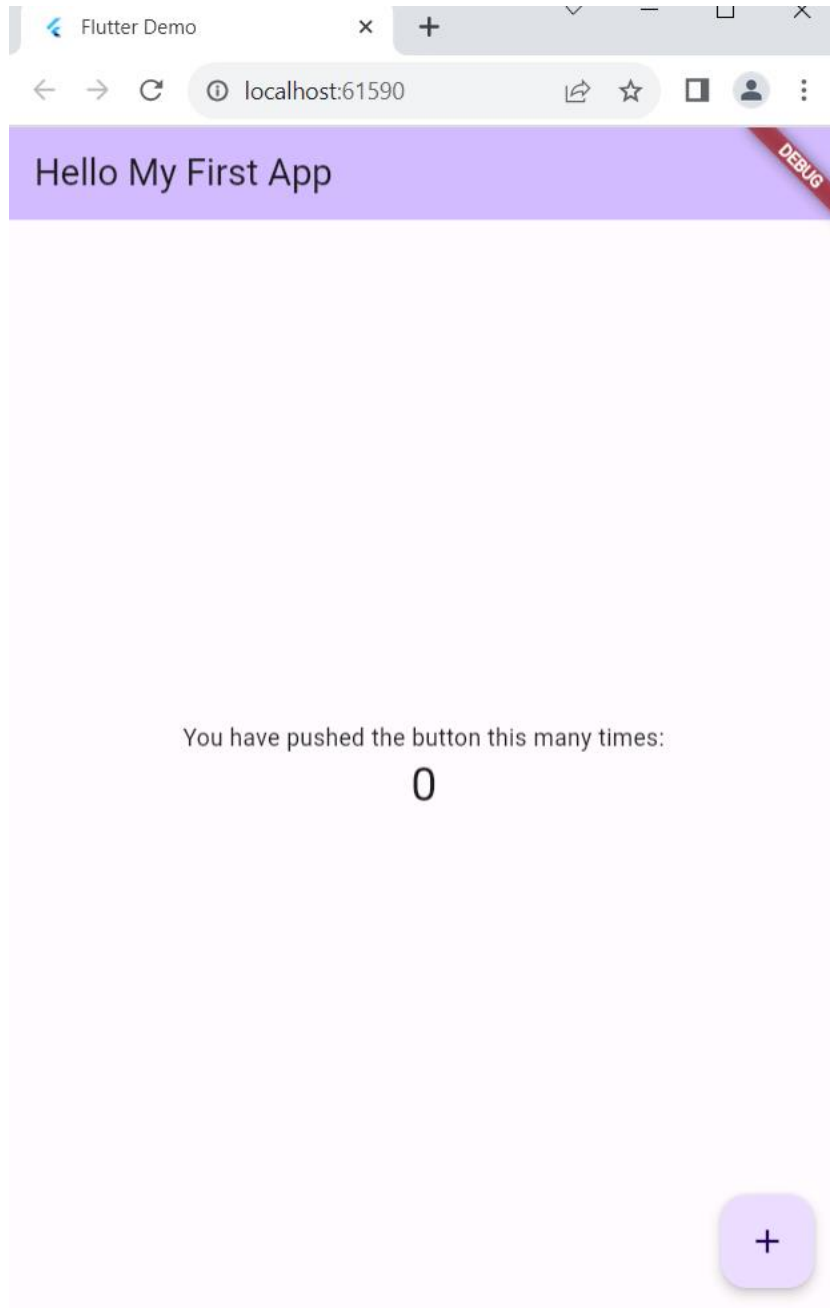
The image shows a screenshot of the DartPad web IDE. The interface includes a top navigation bar with options like 'New Pad', 'Reset', 'Format', and 'Install SDK'. The user's name 'decadent-truth-9056' and a 'local edits' indicator are visible. The main area is split into two panels: a code editor on the left and a preview window on the right. The code editor contains the following Dart code:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(
5     const MaterialApp(
6       title: 'Flutter Tutorial',
7       home: TutorialHome(),
8     ),
9   );
10 }
11
12 class TutorialHome extends StatelessWidget {
13   const TutorialHome({super.key});
14
15   @override
16   Widget build(BuildContext context) {
17     // Scaffold is a layout for
18     // the major Material Components.
19     return Text('hello');
20   }
21 }
```

The preview window on the right shows the output of the code, which is the word "hello" in a red, serif font, underlined with a yellow line. A blue 'Run' button is located between the code editor and the preview window.

Introduction to Mobile Development part 2

ผลลัพธ์ที่จะได้





เตรียมพร้อมสำหรับการ
สร้างแอป



1. ลงโปรแกรมที่เกี่ยวข้อง



Flutter SDK

Windows:

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

คลิกปุ่มสีฟ้าเป็นไฟล์ .zip ตามภาพ

Get the Flutter SDK

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

`flutter_windows_3.3.7-stable.zip`

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

2. Extract the zip file and place the contained `flutter` in the desired installation location for the Flutter SDK (for example, `C:\src\flutter`).



Flutter SDK

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

1-download flutter .zip

2-แตกไฟล์

3-ก็อปไปวางไว้ในโฟลเดอร์ใดก็ได้ C:\




VS Code

Editor ที่ใช้ในการเขียน Flutter ได้อย่างง่ายดาย

<https://code.visualstudio.com/download>

Download Visual Studio Code

Free and built on open source. Integrated Git, debugging and extensions.



Platform	Format	Architecture
Windows	User Installer	64 bit
		32 bit
	System Installer	64 bit
		32 bit
.zip	64 bit	
Linux	.deb	64 bit
		ARM
	.rpm	64 bit
		ARM
	.tar.gz	64 bit
		ARM
Snap Store		
Mac	.zip	Universal
		Intel Chip
		Apple Silicon

หลังจากติดตั้ง VS Code ไปที่เมนูด้านขวาของ VS Code เพื่อค้นหา และติดตั้ง **VS Code Extension** ดังนี้



Flutter 3.22.0

Flutter support and debugger for Visual Studio Code.
Dart Code

2.9M ★ 5



Bracket Pair Colorizer 2 0.2.1

A customizable extension for colorizing matching brackets
CoenraadS

2.8M ★ 4.5

Install



Material Icon Theme 4.8.0

Material Design Icons for Visual Studio Code
Philipp Kief

8.2M ★ 5

Reload Required



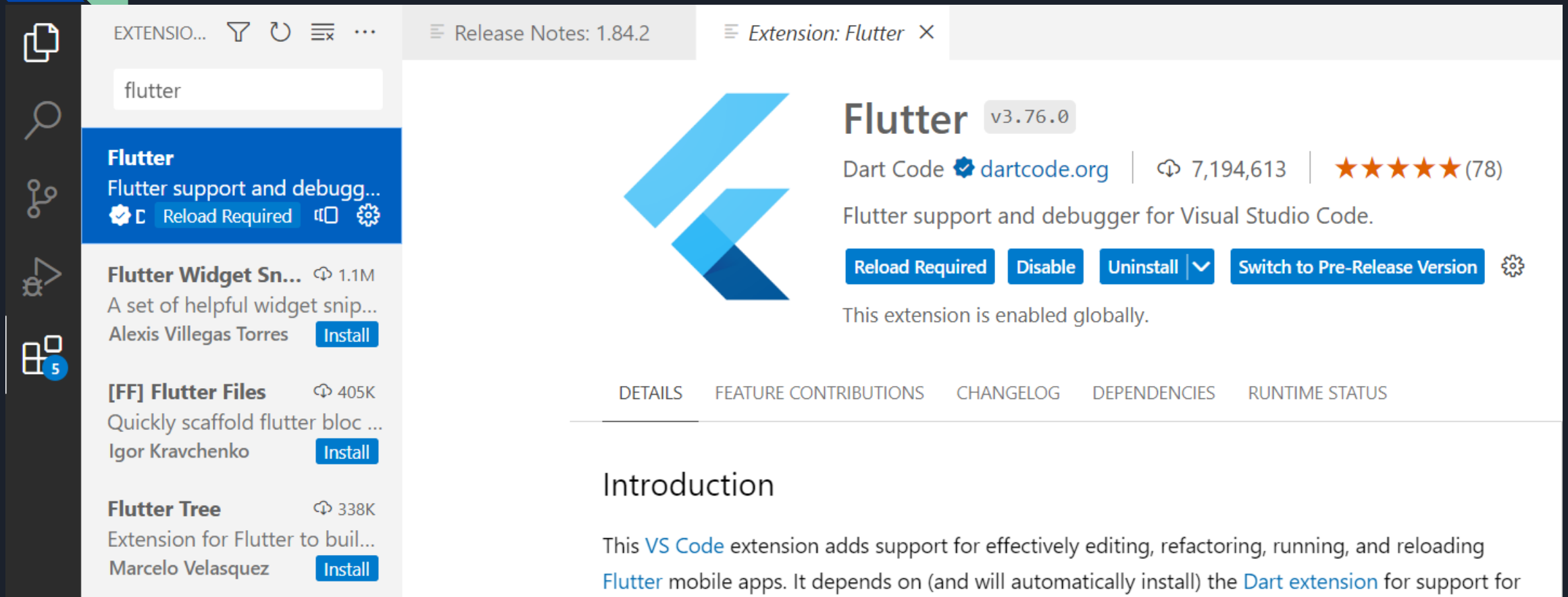
Dart 3.22.0

Dart language support and debugger for Visual Studio Code.
Dart Code

3.1M ★ 5



VS Code Extension > flutter



The screenshot shows the Visual Studio Code interface with the extension marketplace open. The search bar contains "flutter". The left sidebar shows a list of search results for Flutter-related extensions. The main panel displays the details for the "Flutter" extension by Dart Code, including its logo, version (v3.76.0), download count (7,194,613), and a 5-star rating (78 reviews). The extension is currently installed and enabled globally. Below the main panel, there are tabs for "DETAILS", "FEATURE CONTRIBUTIONS", "CHANGELOG", "DEPENDENCIES", and "RUNTIME STATUS". The "Introduction" section is visible, starting with "This VS Code extension adds support for effectively editing, refactoring, running, and reloading Flutter mobile apps. It depends on (and will automatically install) the Dart extension for support for".

EXTENSIO... | Release Notes: 1.84.2 | Extension: Flutter X

flutter

Flutter
Flutter support and debugg...
Reload Required

Flutter Widget Sn... 1.1M
A set of helpful widget snip...
Alexis Villegas Torres **Install**

[FF] Flutter Files 405K
Quickly scaffold flutter bloc ...
Igor Kravchenko **Install**

Flutter Tree 338K
Extension for Flutter to buil...
Marcelo Velasquez **Install**

Flutter v3.76.0
Dart Code dartcode.org | 7,194,613 | ★★★★★ (78)
Flutter support and debugger for Visual Studio Code.

Reload Required | Disable | Uninstall | Switch to Pre-Release Version

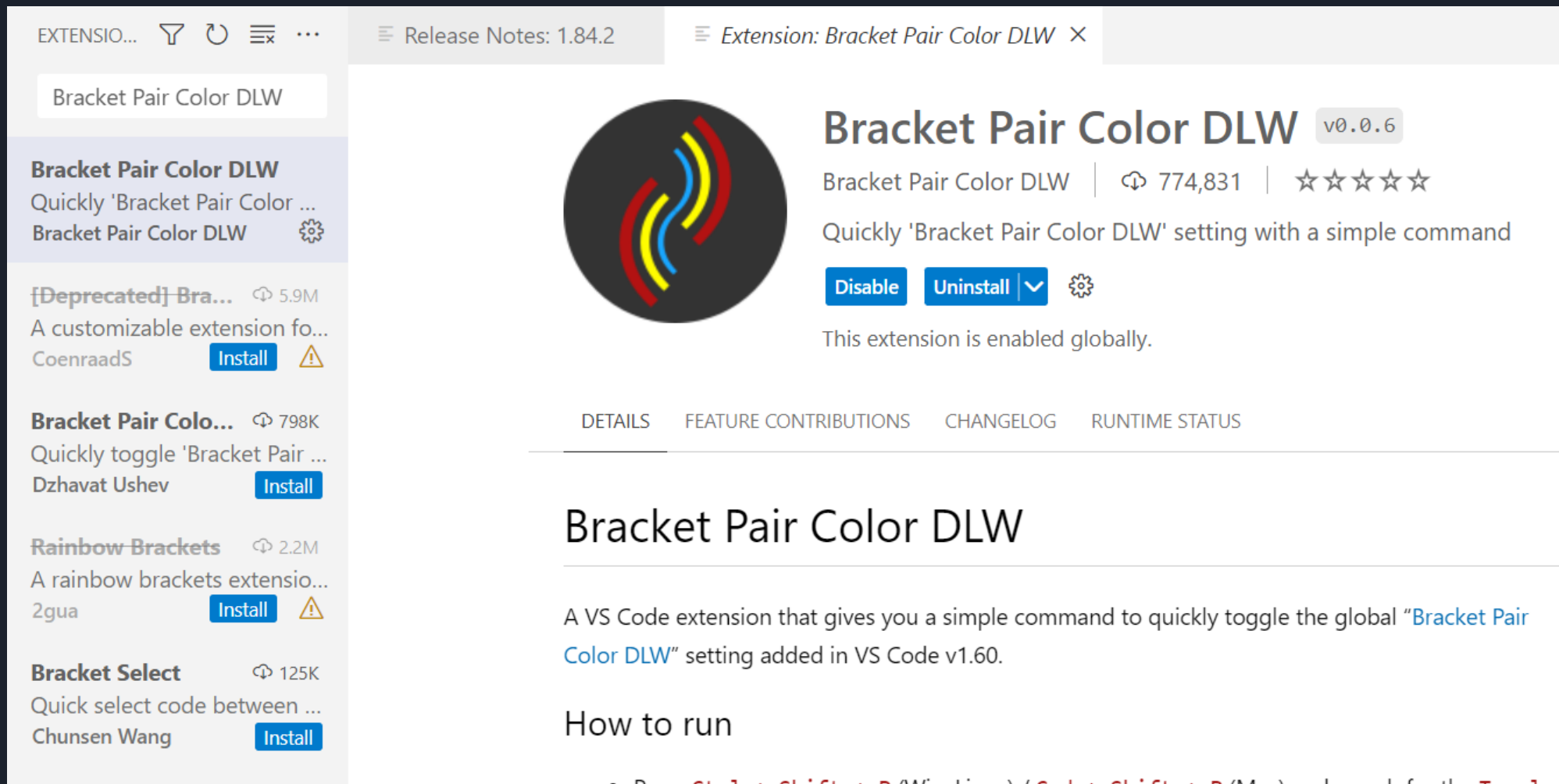
This extension is enabled globally.

DETAILS | FEATURE CONTRIBUTIONS | CHANGELOG | DEPENDENCIES | RUNTIME STATUS

Introduction

This [VS Code](#) extension adds support for effectively editing, refactoring, running, and reloading [Flutter](#) mobile apps. It depends on (and will automatically install) the [Dart extension](#) for support for

VS Code Extension > Bracket Pair Color DLW : สำหรับแสดง สีคู่ของวงเล็บเช่น () [] { } เพื่อให้อ่าน code ง่าย



The screenshot displays the VS Code Extension Marketplace interface. On the left, a search bar contains 'Bracket Pair Color DLW'. Below it, a list of search results is shown, with the top result being the 'Bracket Pair Color DLW' extension by CoenraadS, which has 5.9M downloads and an 'Install' button. Other results include '[Deprecated] Bra...' by CoenraadS, 'Bracket Pair Colo...' by Dzhavat Ushev, 'Rainbow Brackets' by 2gua, and 'Bracket Select' by Chunsen Wang.

The main content area shows the details for the 'Bracket Pair Color DLW' extension (version v0.0.6). It features a circular logo with three curved lines in red, yellow, and blue. The extension has 774,831 downloads and a 5-star rating. The description states: 'Quickly toggle "Bracket Pair Color DLW" setting with a simple command'. Below the description are buttons for 'Disable', 'Uninstall', and a settings gear icon. A status message indicates 'This extension is enabled globally.' Navigation tabs for 'DETAILS', 'FEATURE CONTRIBUTIONS', 'CHANGELOG', and 'RUNTIME STATUS' are visible. The 'How to run' section is partially visible at the bottom, starting with 'Press Ctrl + Shift + P (Win, Linux) / Cmd + Shift + P (Mac) and search for the Toggle'.

VS Code Extension > Material Icon Theme

สำหรับเพิ่มสีสัน และ ไอคอนต่าง ๆ ให้กับไฟล์ และ โฟลเดอร์

The screenshot displays the VS Code extension marketplace interface. On the left, a search bar contains 'Material Icon Theme'. Below it, a list of search results is shown, with the top result being the 'Material Icon Theme' by Philipp Kief, which is highlighted in blue. The main panel shows the details for this extension, including its icon (a blue circle with a white folder icon), the version number 'v4.31.0', the author 'Philipp Kief' with a verified badge and website 'pkief.com', and a download count of 20,599,229. The extension has a 5-star rating from 316 reviews. Below the rating, the description reads 'Material Design Icons for Visual Studio Code'. There are three buttons: 'Set File Icon Theme' (highlighted in blue), 'Disable', and 'Uninstall'. A gear icon for settings is also present. A message states 'This extension is enabled globally.' At the bottom, there are tabs for 'DETAILS', 'FEATURE CONTRIBUTIONS', 'CHANGELOG', and 'RUNTIME STATUS'. The 'DETAILS' tab is selected. Below the tabs, there is a large blue circle with a white folder icon, which is a visual representation of the extension's theme.

EXTENSIO...

Material Icon Theme

Material Icon The... 327ms
Material Design Icons for Vi...
Philipp Kief

Material Theme I... 3.3M
Material Theme Icons, the ...
Equinusocio

Gruvbox Material ... 39K
Gruvbox Material Icons
JonathanHarty

Unofficial Material ... 3K
Unofficial Material Design Ic...
artalatarta

Material Icon The... 686
Material Design Icons for Vi...
simonhe

White Material Ico... 1K
Material Design Icons for Vi...
viphakorn

Release Notes: 1.84.2 Extension: Material Icon Theme ×

Material Icon Theme v4.31.0

Philipp Kief pkief.com | 20,599,229 | ★★★★★ (316)

Material Design Icons for Visual Studio Code

This extension is enabled globally.

DETAILS FEATURE CONTRIBUTIONS CHANGELOG RUNTIME STATUS

VS Code Extension > dart

The screenshot displays the Visual Studio Code extension marketplace interface. At the top, there are navigation icons and tabs for 'EXTENSIO...', 'Release Notes: 1.84.2', and 'Extension: Dart'. A search bar contains the text 'dart'. The left sidebar lists search results, with the top result being the 'Dart' extension by Dart Code, which is highlighted in blue and marked as 'Reload Required'. Below it are 'dart-import' by Luan and 'Dart (Syntax Hig...' by oscarcs. The main content area shows the details for the 'Dart' extension, including its logo, version 'v3.76.1', publisher 'Dart Code', website 'dartcode.org', download count '7,883,403', and a 5-star rating from 77 reviews. The description states it is for 'Dart language support and debugger for Visual Studio Code.' Action buttons include 'Reload Required', 'Disable', 'Uninstall', and 'Switch to Pre-Release Version'. A status message indicates the extension is enabled globally. At the bottom, there are tabs for 'DETAILS', 'FEATURE CONTRIBUTIONS', 'CHANGELOG', and 'RUNTIME STATUS', along with social media links for chat, discord, twitter, dartcode, help, and contribute. The 'Introduction' section is partially visible at the bottom.

EXTENSIO...

Release Notes: 1.84.2 Extension: Dart

dart

Dart v3.76.1
Dart Code dartcode.org | 7,883,403 | ★★★★★ (77)
Dart language support and debugger for Visual Studio Code.

Reload Required **Disable** **Uninstall** **Switch to Pre-Release Version**

This extension is enabled globally.

DETAILS FEATURE CONTRIBUTIONS CHANGELOG RUNTIME STATUS

[chat](#) [discord](#) [twitter](#) [dartcode](#) [help](#) [contribute](#)

Introduction

Dart SDK

Version 2 ขึ้นไปจะมาพร้อมกับ Flutter อยู่แล้วในโฟลเดอร์ bin ไม่ต้องโหลดก็ได้ แต่หากต้องการดาวโหลดให้โหลดผ่านลิงค์นี้แทน

<https://dart.dev/tools/sdk/archive>

Stable channel

Stable channel builds are tested and approved for production use.

Version: 3.1.5 < OS: Windows <

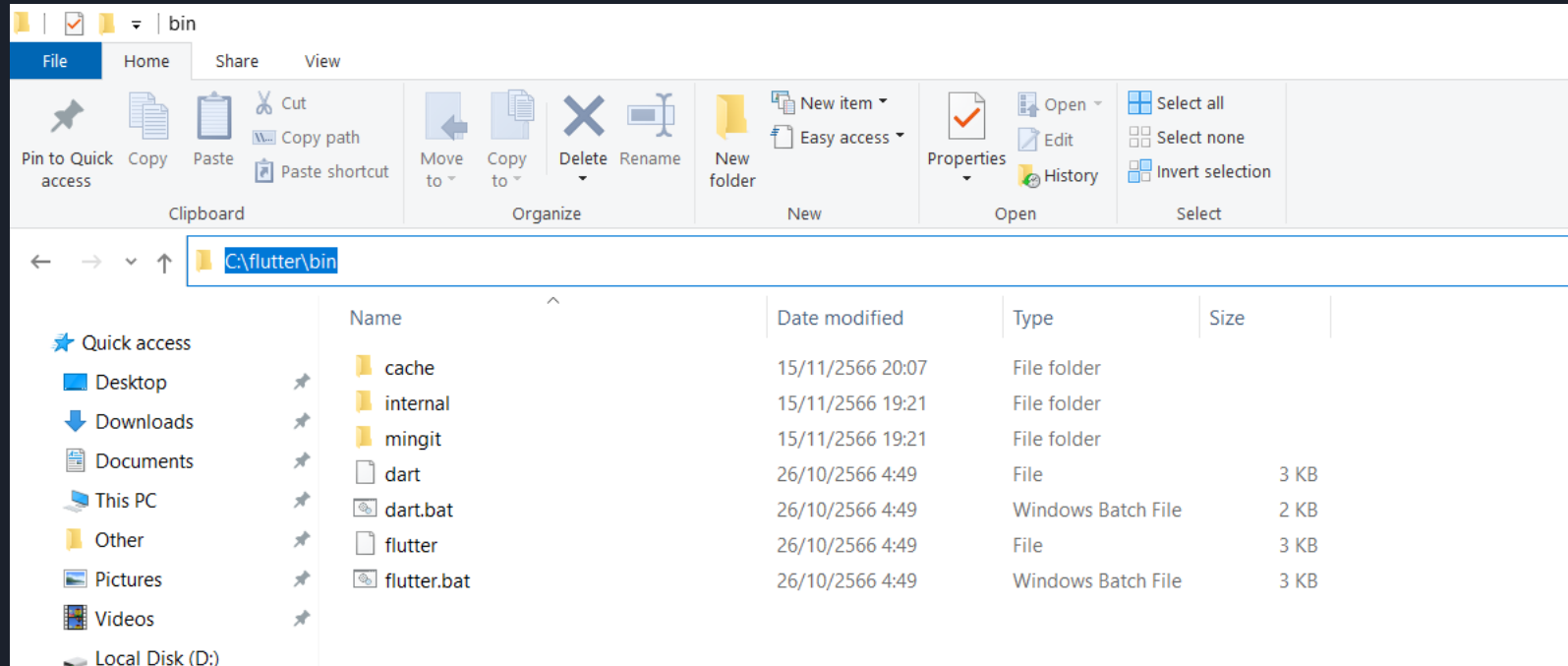
Version	OS	Architecture	Release date	Downloads
3.1.5 (ref 5e42305)	Windows	x64	Oct 25, 2023	Dart SDK (SHA-256)
3.1.5 (ref 5e42305)	Windows	IA32	Oct 25, 2023	Dart SDK (SHA-256)
3.1.5 (ref 5e42305)	---	---	Oct 25, 2023	API docs



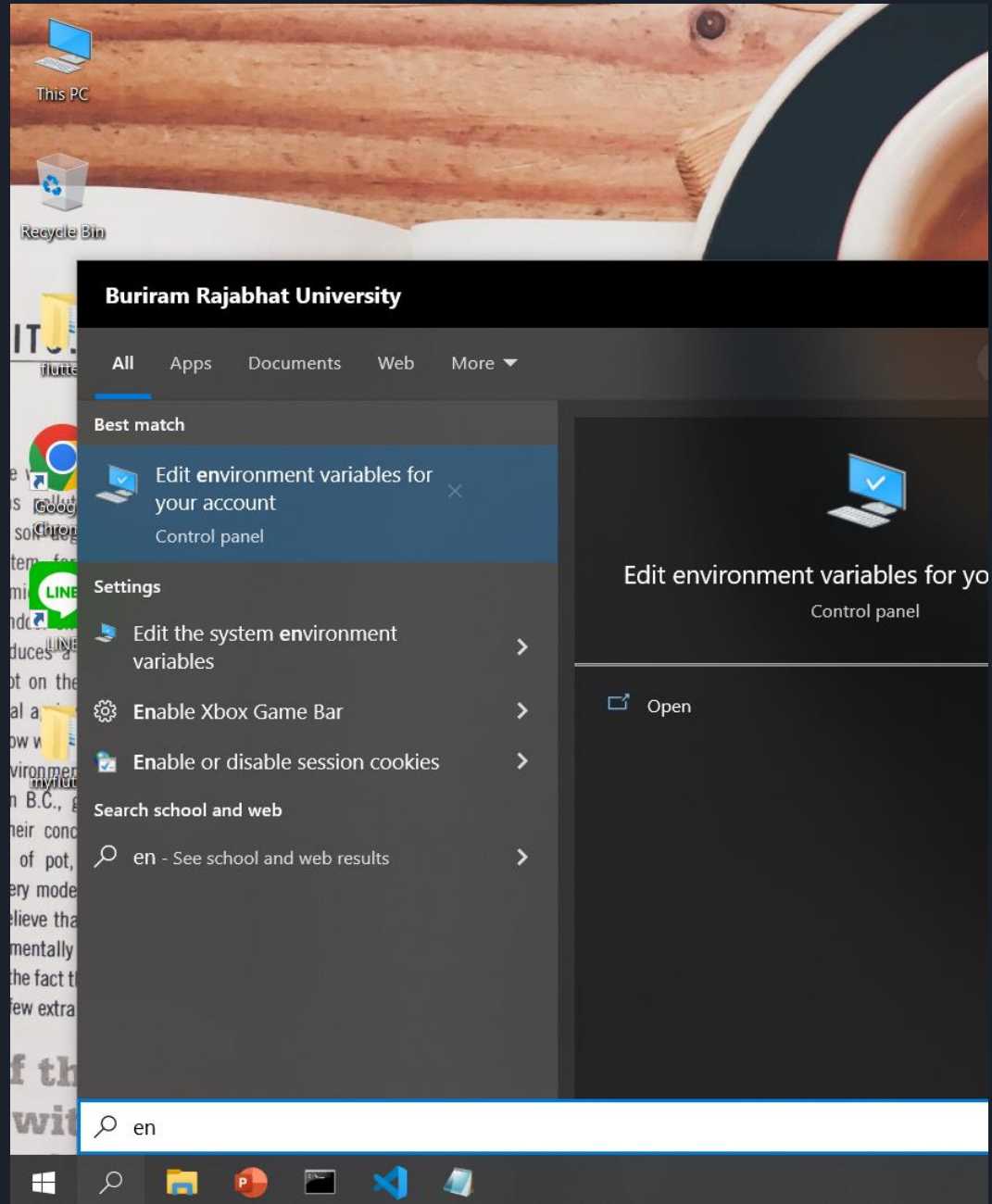
2. Set path

2.1 copy path นี้

C:\flutter\bin



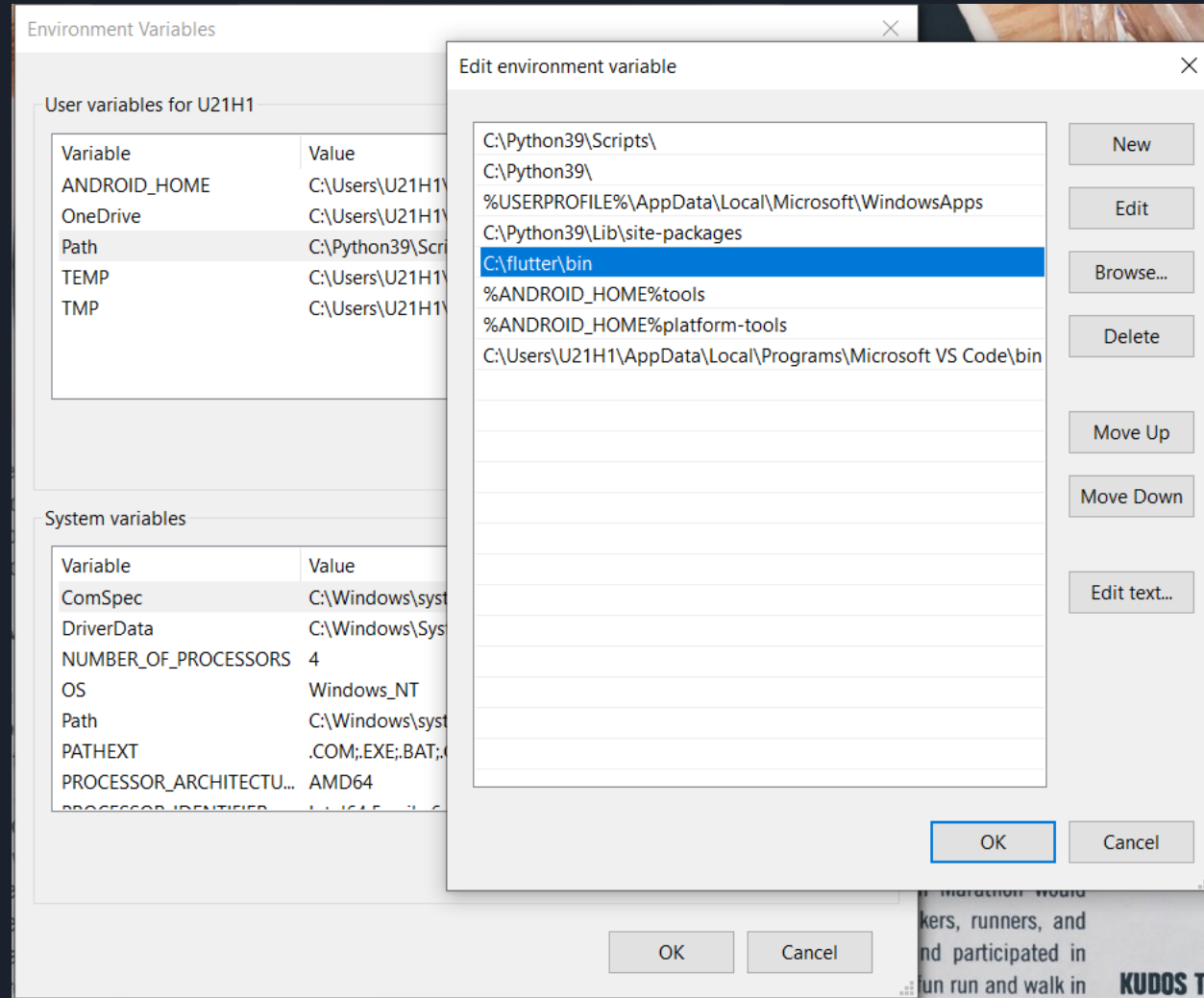
2.2 ไปที่ edit environment variable



Set path

2. 3 ไปที่ path สร้าง path ใหม่ ดังนี้

C:\flutter\bin





3. ตรวจสอบการใช้งานด้วย command prompt

3.1 ตรวจสอบการใช้งานด้วย command prompt ด้วยคำสั่ง flutter

```
Command Prompt - flutter
Microsoft Windows [Version 10.0.19045.3570]
(c) Microsoft Corporation. All rights reserved.

C:\Users\U21H1>flutter
Manage your Flutter app development.

Common commands:

flutter create <output directory>
    Create a new Flutter project in the specified directory.

flutter run [options]
    Run your Flutter application on an attached device or in an emulator.

Usage: flutter <command> [arguments]

Global options:
-h, --help                Print this usage information.
-v, --verbose              Noisy logging, including all shell commands executed.
                           If used with "--help", shows hidden options. If used with "flutter doctor",
                           shows additional diagnostic information. (Use "-vv" to force verbose
                           logging in those cases.)
-d, --device-id           Target device id or name (prefixes allowed).
--version                 Reports the version of this tool.
--suppress-analytics      Suppress analytics reporting for the current CLI invocation.
--disable-telemetry       Disable telemetry reporting each time a flutter or dart command runs, until
                           it is re-enabled.
--enable-telemetry        Enable telemetry reporting each time a flutter or dart command runs.
```

3.2 ตรวจสอบการใช้งานด้วย command prompt ด้วยคำสั่ง flutter doctor

กรณี “ผ่าน”

```
C:\Users\U21H1>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.3.8, on Microsoft Windows [Version 10.0.19045.2130], locale th-TH)
Checking Android licenses is taking an unexpectedly long time... [✓] Android toolchain - develop for Android devices (Android SDK version 31.0.0)
[✓] Chrome - develop for the web
[✓] Visual Studio - develop for Windows (Visual Studio Community 2022 17.4.0)
[✓] Android Studio (version 2021.3)
[✓] VS Code (version 1.73.1)
[✓] Connected device (3 available)
[✓] HTTP Host Availability

• No issues found!

C:\Users\U21H1>
```

แต่เนื่องจากใน lab นี้จะไม่รันผ่าน Android และไม่รันผ่าน windows จึงไม่ต้องแก้ error เหล่านี้

```
C:\Users\U21H1>flutter doctor
Doctor summary (to see all details, run flutter doctor -v):
[✓] Flutter (Channel stable, 3.13.9, on Microsoft Windows [Version 10.0.19045.3570], locale th-TH)
[✓] Windows Version (Installed version of Windows is version 10 or higher)
[!] Android toolchain - develop for Android devices (Android SDK version 31.0.0)
    X No Java Development Kit (JDK) found; You must have the environment variable JAVA_HOME set and the
      java binary in your PATH. You can download the JDK from
      https://www.oracle.com/technetwork/java/javase/downloads/.
[✓] Chrome - develop for the web
[X] Visual Studio - develop Windows apps
    X Visual Studio not installed; this is necessary to develop Windows apps.
      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)
[✓] VS Code (version 1.84.2)
[✓] Connected device (3 available)
[✓] Network resources

! Doctor found issues in 3 categories.
```



ตรวจสอบการใช้งานด้วย command prompt กรณี “ไม่ผ่าน” แก้ปัญหา ได้ดังนี้

ถ้าเจอคำว่า **Android Studio not Installed**

`flutter config --android-studio-dir="C:\Program Files\Android\Android Studio"`
Path ใน "" ให้ใส่ path ที่เก็บจริงของเครื่องเรา

ถ้าเจอคำว่า **Android Accept License**

`flutter doctor --android-licenses`

ถ้าผ่าน **Android Accept License** จะขึ้นให้กด y enter เพื่อ Accept ตลอด



ตรวจสอบการใช้งานด้วย command prompt กรณี “ไม่ผ่าน” แก้ปัญหา ได้ดังนี้

ถ้าเจอคำว่า `Android toolchain - develop for Android devices` X `ANDROID_HOME = C:\Users\User\AppData\Local\Android\Sdk` but Android SDK not found at this location.

แก้โดยพิมพ์

`flutter config --android-sdk` ใส่ path SDK ของเราตรงนี้

cr. <https://stackoverflow.com/questions/49175231/flutter-does-not-find-android-sdk>



ตรวจสอบการใช้งานด้วย command prompt กรณี “ไม่ผ่าน” แก้ปัญหา ได้ดังนี้

ถ้าเจอคำว่า **No devices available**

วิธีแก้ปัญหา no device รันคำสั่งต่อไปนี้ ทีละคำสั่ง

`flutter channel beta`

`flutter upgrade`

`flutter config --enable-web`

ตรวจสอบการใช้งานด้วย command prompt กรณี “ไม่ผ่าน” แก้ปัญหา ได้ดังนี้

ถ้าเจอคำว่า

```
[X] Visual Studio - develop Windows apps
X Visual Studio not installed; this is necessary to develop Windows apps.
  Download at https://visualstudio.microsoft.com/downloads/.
  Please install the "Desktop development with C++" workload, including all of its default
  components
```

แก้โดย :

As it states, you need to download Visual Studio (which is different from Visual Studio Code).

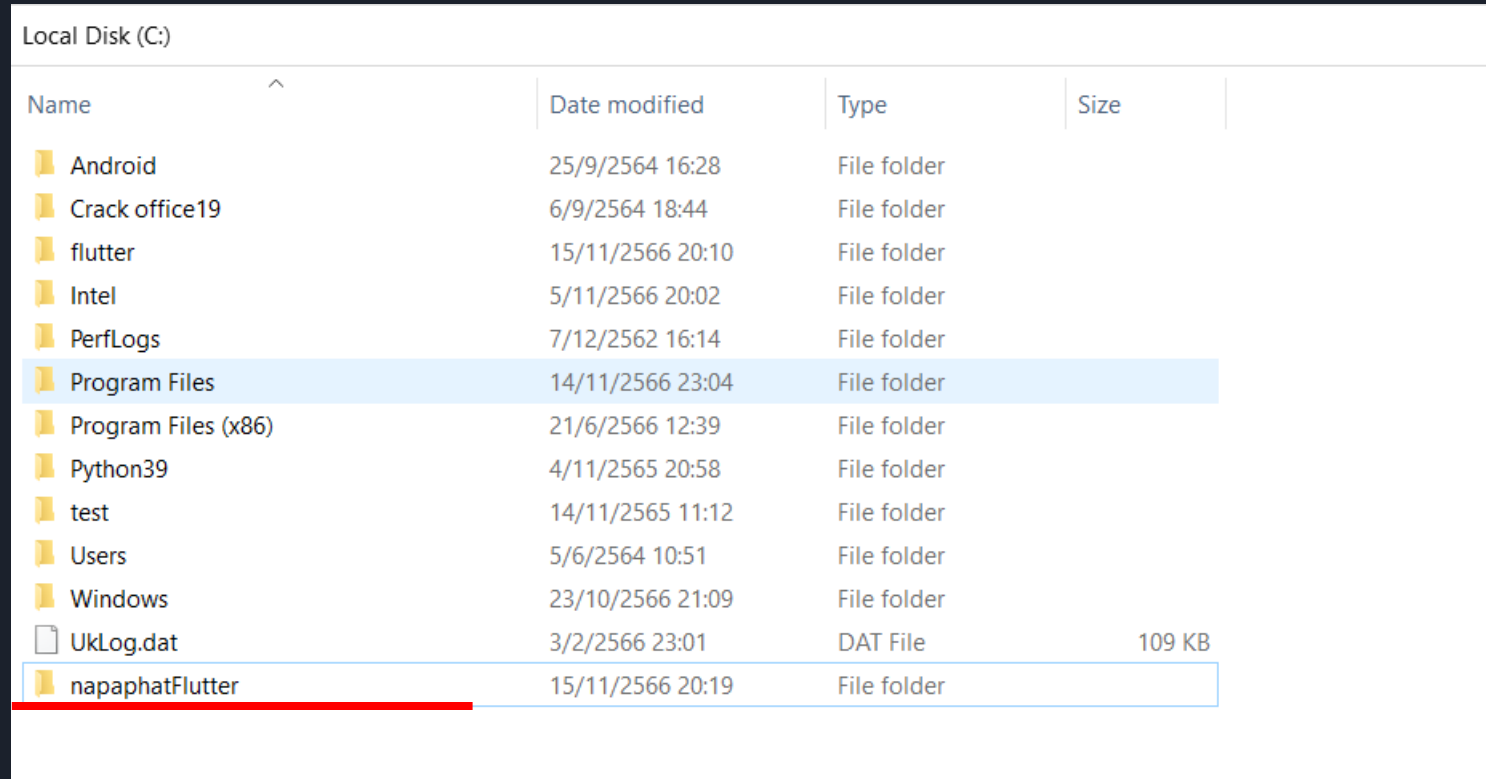
When installing it, remember to select the required package Desktop development with C++:

Cr. <https://stackoverflow.com/questions/71080518/visual-studio-not-installed-this-is-necessary-for-windows-development>



4. มาเริ่มสร้าง project กัน

4.1 สร้าง folder เก็บไฟล์



Local Disk (C:)

Name	Date modified	Type	Size
Android	25/9/2564 16:28	File folder	
Crack office19	6/9/2564 18:44	File folder	
flutter	15/11/2566 20:10	File folder	
Intel	5/11/2566 20:02	File folder	
PerfLogs	7/12/2562 16:14	File folder	
Program Files	14/11/2566 23:04	File folder	
Program Files (x86)	21/6/2566 12:39	File folder	
Python39	4/11/2565 20:58	File folder	
test	14/11/2565 11:12	File folder	
Users	5/6/2564 10:51	File folder	
Windows	23/10/2566 21:09	File folder	
UkLog.dat	3/2/2566 23:01	DAT File	109 KB
napaphatFlutter	15/11/2566 20:19	File folder	



4.2 cd เข้า folder ที่สร้าง

ใช้คำสั่ง `cd "path folder ที่สร้าง"`

```
C:\Users\U21H1>cd "C:\napaphatFlutter"
```

```
C:\napaphatFlutter>
```

4.3 สร้าง project ด้วยคำสั่ง `flutter create` ชื่อที่ต้องการตั้ง ชื่อต้องเป็นตัวเล็กทั้งหมด

```
C:\napaphatFlutter>flutter create firstapp
Creating project firstapp...
Resolving dependencies in firstapp... (2.5s)
Got dependencies in firstapp.
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 firstapp
$ flutter run

Your application code is in firstapp\lib\main.dart.

C:\napaphatFlutter>
```

4.4 ใช้คำสั่ง cd ชื่อที่ตั้ง เพื่อเข้าสู่ folder project ที่สร้าง

```
C:\napaphatFlutter>cd firstapp  
C:\napaphatFlutter\firstapp>
```

4.5 ใช้คำสั่ง Flutter run เพื่อสั่ง run project

```
C:\napaphatFlutter\firstapp>flutter run
```

```
Connected devices:
```

```
Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.19045.3570]  
Chrome (web) • chrome • web-javascript • Google Chrome 119.0.6045.124  
Edge (web) • edge • web-javascript • Microsoft Edge 119.0.2151.58
```

```
[1]: Windows (windows)
```

```
[2]: Chrome (chrome)
```

```
[3]: Edge (edge)
```

```
Please choose one (or "q" to quit):
```

4.6 เลือก device

```
Please choose one (or "q" to quit): 2
Launching lib\main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome... 34.3s
This app is linked to the debug service: ws://127.0.0.1:61615/Rr5atL6j7DE=/ws
Debug service listening on ws://127.0.0.1:61615/Rr5atL6j7DE=/ws

  To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".

A Dart VM Service on Chrome is available at: http://127.0.0.1:61615/Rr5atL6j7DE=
The Flutter DevTools debugger and profiler on Chrome is available at:
http://127.0.0.1:9100?uri=http://127.0.0.1:61615/Rr5atL6j7DE=
```


4.7 จะได้ผลการรันดังภาพ

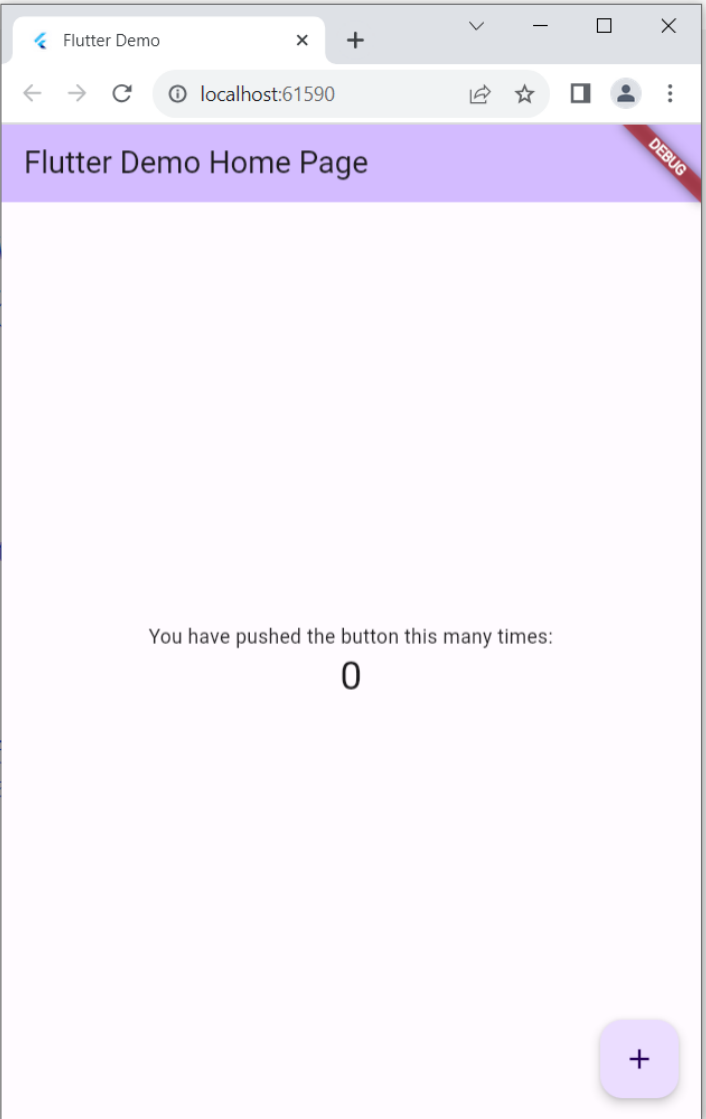
```
Command Prompt - flutter - flutter doctor - flutter - flutter doctor - flutter create firstapp - flutter run

C:\napaphatFlutter>cd firstapp

C:\napaphatFlutter\firstapp>flutter run
Connected devices:
Windows (desktop) • windows • windows-x64 • Microsoft Windows [V
Chrome (web) • chrome • web-javascript • Google Chrome 119.0.
Edge (web) • edge • web-javascript • Microsoft Edge 119.0
[1]: Windows (windows)
[2]: Chrome (chrome)
[3]: Edge (edge)
Please choose one (or "q" to quit): 2
Launching lib\main.dart on Chrome in debug mode...
Waiting for connection from debug service on Chrome...
This app is linked to the debug service: ws://127.0.0.1:61615/Rr5atL6j7DE=
Debug service listening on ws://127.0.0.1:61615/Rr5atL6j7DE=ws

To hot restart changes while running, press "r" or "R".
For a more detailed help message, press "h". To quit, press "q".

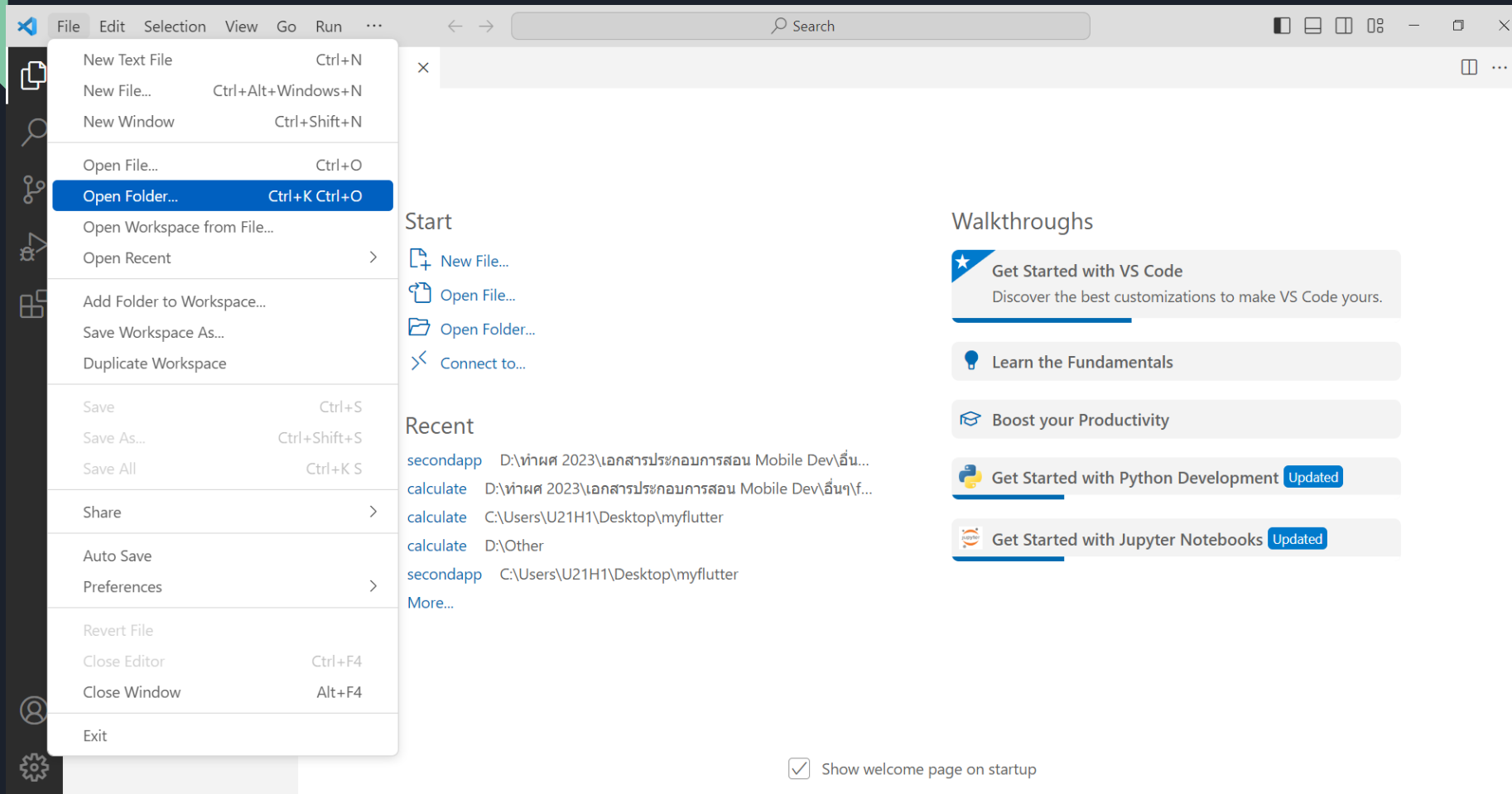
A Dart VM Service on Chrome is available at: http://127.0.0.1:61615
The Flutter DevTools debugger and profiler on Chrome is available at
http://127.0.0.1:9100?uri=http://127.0.0.1:61615/Rr5atL6j7DE=
```



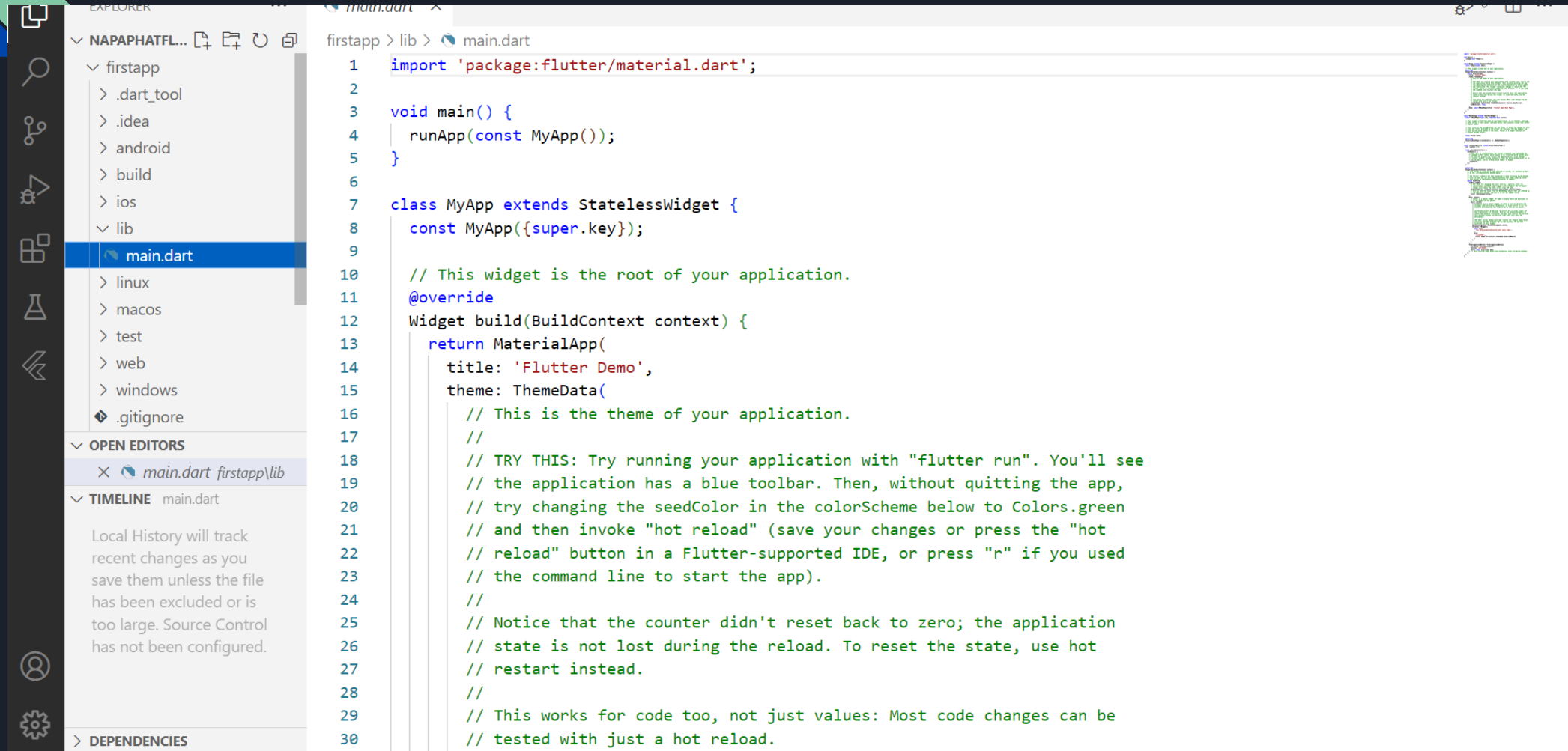


5. ลองเปิดไฟล์ใน vscode

5.1 เปิด vscode เลือก folder project



5.2 เปิด folder lib > เลือกไฟล์ main.dart




The screenshot shows an IDE interface with the following components:

- EXPLORER (Left Panel):** Displays the project structure. The 'lib' folder is expanded, and 'main.dart' is selected. Other folders include '.dart_tool', '.idea', 'android', 'build', 'ios', 'linux', 'macos', 'test', 'web', and 'windows'. Below the file explorer are sections for 'OPEN EDITORS' (showing 'main.dart firstapp\lib') and 'TIMELINE' (showing 'main.dart').
- Code Editor (Main Area):** Shows the content of 'main.dart' with line numbers 1 through 30. The code is as follows:

```
1 import 'package:flutter/material.dart';
2
3 void main() {
4   runApp(const MyApp());
5 }
6
7 class MyApp extends StatelessWidget {
8   const MyApp({super.key});
9
10  // This widget is the root of your application.
11  @override
12  Widget build(BuildContext context) {
13    return MaterialApp(
14      title: 'Flutter Demo',
15      theme: ThemeData(
16        // This is the theme of your application.
17        //
18        // TRY THIS: Try running your application with "flutter run". You'll see
19        // the application has a blue toolbar. Then, without quitting the app,
20        // try changing the seedColor in the colorScheme below to Colors.green
21        // and then invoke "hot reload" (save your changes or press the "hot
22        // reload" button in a Flutter-supported IDE, or press "r" if you used
23        // the command line to start the app).
24        //
25        // Notice that the counter didn't reset back to zero; the application
26        // state is not lost during the reload. To reset the state, use hot
27        // restart instead.
28        //
29        // This works for code too, not just values: Most code changes can be
30        // tested with just a hot reload.
```
- Debugger (Right Panel):** Shows a stack trace with multiple frames, indicating the application is running.

5.3 ลองแก้ไขไฟล์ main.dart

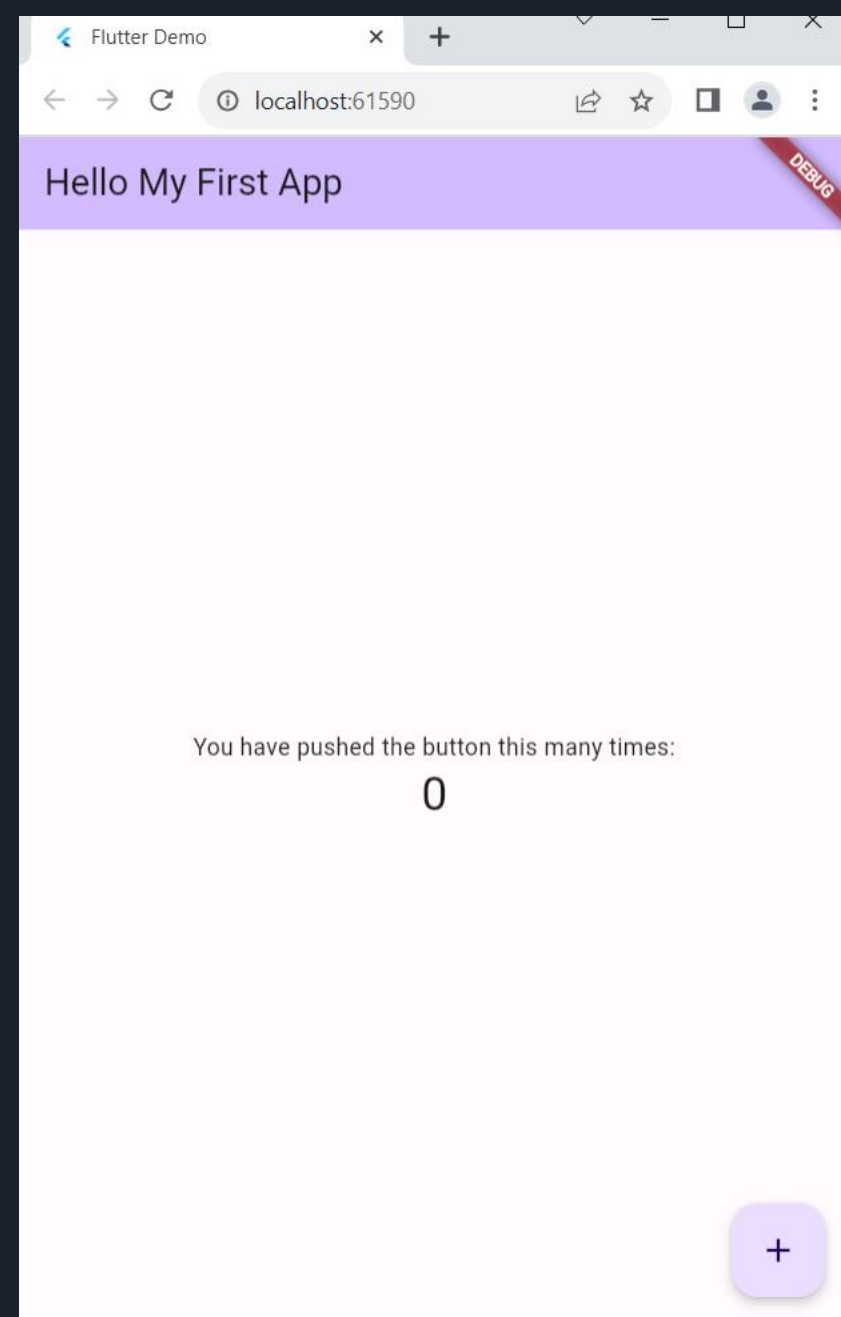
```
main.dart
firstapp > lib > main.dart > MyApp > build
19 // the application has a blue toolbar. Then, without quitting the app,
20 // try changing the seedColor in the colorScheme below to Colors.green
21 // and then invoke "hot reload" (save your changes or press the "hot
22 // reload" button in a Flutter-supported IDE, or press "r" if you used
23 // the command line to start the app).
24 //
25 // Notice that the counter didn't reset back to zero; the application
26 // state is not lost during the reload. To reset the state, use hot
27 // restart instead.
28 //
29 // This works for code too, not just values: Most code changes can be
30 // tested with just a hot reload.
31 colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
32 useMaterial3: true,
33 ), // ThemeData
34 home: const MyHomePage(title: 'Hello My First App'),
35 ); // MaterialApp
36 }
37 }
```




5.4 กด save > แล้วกด r/shift r เพื่อ restart
หมายเหตุ* r แก้ไขเล็กๆ บางส่วน ใช้เวลาน้อย
shift r ต้องการ run ทั้งหมด ใช้เวลามากกว่า r

```
To hot restart changes while running, press "r" or "R".  
For a more detailed help message, press "h". To quit, press "q".  
  
A Dart VM Service on Chrome is available at: http://127.0.0.1:61615/Rr5atL6j7DE=  
The Flutter DevTools debugger and profiler on Chrome is available at:  
http://127.0.0.1:9100?uri=http://127.0.0.1:61615/Rr5atL6j7DE=  
  
Performing hot restart... 741ms  
Restarted application in 746ms.
```

5.5 ผลการรัน หลังแก้ไขโค้ด จะได้ผลดังรูป





5.6 กด ctrl c กรณีค้าง ระบบจะขึ้นข้อความดังรูป ให้กด y
ระบบจะออกจากการรัน สามารถเข้ารันแอปใหม่ได้

```
Terminate batch job (Y/N)? y  
C:\napaphatFlutter\firstapp>
```


Reference

KongRuksia Official. <https://www.youtube.com/c/KongRuksiamOfficial>

ลุงวิศวกร สอนคำนวณ. Flutter 2 Mobile App Bootcamp 2021 - Uncle Engineer