

Chapter 3 พื้นฐาน Widget เบื้องต้น

ผลลัพธ์วันนี้

The image shows a development environment with an IDE on the left and a browser preview on the right. The IDE displays the following Dart code in `main.dart`:

```
66 // called again, and so nothing would appear to hap
67   _counter++;
68 });
69 }
70
71 @override
72 Widget build(BuildContext context) {
73   // This method is rerun every time setState is called
74   // by the _incrementCounter method above.
75   //
76   // The Flutter framework has been optimized to make r
77   // fast, so that you can just rebuild anything that n
78   // than having to individually change instances of wi
79   return Scaffold(
80     body: Center(child: Text("test center", style: Te
81       fontWeight: FontWeight.bold, color: Colors.ambe
82     floatingActionButton: FloatingActionButton(
83       onPressed: _incrementCounter,
84       tooltip: 'Increment',
85       child: const Icon(Icons.add),
86     ),floatingActionButtonLocation: FloatingActionButton
87   ); // Scaffold
88 }
89
90
```

The browser preview, titled "Flutter Demo", shows a white screen with the text "test center" in a large, bold, orange font. A purple floating action button with a white plus sign is visible in the bottom right corner. A red "DEBUG" banner is present in the top right corner of the browser window.

Widget คืออะไร

แนวคิดในการแบ่ง User Interface ออกเป็นชิ้นส่วนต่างๆ แล้วนำมาแสดงผลบนหน้าจอ เราจะเรียกชิ้นส่วนนี้ว่า Widget เช่น ปุ่ม ข้อความ เป็นต้น

วิดเจ็ต (Widget) เป็นส่วนประกอบของส่วนติดต่อผู้ใช้งานผ่านการแสดงผลแบบกราฟิก

Widget แต่ละตัวจะมีส่วนที่เรียกว่า Properties สำหรับกำหนดคุณสมบัติให้กับWidget นั้น ๆ ซึ่ง Widget แต่ละตัวก็มี Properties ที่หลากหลายให้เราใช้งานแตกต่างกันออกไป เช่น กำหนดสี เส้นขอบ ส่วนโค้ง เป็นต้น

ประเภทของ Widget

1. Widget ที่ Flutter เตรียมไว้ให้ (Standard Widget)
2. Widget สร้างขึ้นเอง (Custom Widget) เป็นการนำเอา Widget ที่ Flutter เตรียมไว้มาจับกลุ่มแก้ไข หรือสร้างใหม่ให้มีรูปแบบที่ต้องการ
3. Widget จากนักพัฒนาคนอื่น (Third Party Widget) หากมีผู้สร้าง Widget ซึ่งตรงกับความต้องการอยู่แล้ว ก็ไม่ต้องเสียเวลาสร้างขึ้นมาใหม่ทั้งหมด เพราะเพียงแค่ติดตั้งไลบรารี และอิมพอร์ต Widget เหล่านั้นมาใช้ในโปรเจกต์ ก็จะได้ Widget ในแบบที่ต้องการทันที

ตัวอย่าง Widget Widget ที่ทาง Flutter เตรียมไว้ให้ (Standard Widget)

1. Text Widget สำหรับการแสดงข้อความใน Flutter
2. RaisedButton เป็นวิดเจ็ตที่เป็นปุ่มกดในรูปแบบอย่างง่ายที่ต้องรอรับการโต้ตอบกับผู้ใช้งาน
3. Row, Column สำหรับสร้าง Layout (เค้าโครงหน้าแอปพลิเคชัน) เป็นการจัดเรียง Widget แบบซ้ายไปขวา ส่วน Column เป็นการจัดเรียงแบบบนลงล่าง
4. Stack สำหรับการจัดวางวิดเจ็ตซ้อนทับกันเป็นชั้น ๆ
5. Container เป็นวิดเจ็ตสำหรับการบรรจุหรือจัดวางวิดเจ็ตย่อย ๆ เพื่อกำหนดโครงร่างตามต้องการ

ตัวอย่าง Widget สร้างขึ้นเอง (Custom Widget)

1. **Stateless Widget** เป็น Widget ใช้เพื่อแสดงผลเพียงอย่างเดียว และไม่มีการเปลี่ยนแปลงค่าในระหว่างใช้งาน **เช่น** Widget ที่ใช้แสดงรูปโลโก้ของแอปพลิเคชันหน้าจอหลัก Widget ที่ใช้แสดงรูปสินค้า เป็นต้น
2. **Stateful Widget** เป็น Widget มีลักษณะของการตอบสนองแบบไดนามิก ทำหน้าที่ในการจัดการ Widget ต่าง ๆ คล้ายกับ Stateless Widget แต่แตกต่างที่ Stateful Widget จะถูกนำมาใช้ในงานที่สามารถตอบสนองกับผู้ใช้งานได้ทั้งในรูปแบบที่ผู้ใช้สื่อสารมายังแอปพลิเคชันและการโต้ตอบหรือการเปลี่ยนแปลงสถานะตามสิ่งเร้าที่ผู้ใช้งานต้องการ Stateful Widget สามารถเก็บข้อมูลและติดตามสถานะของข้อมูล (State) เมื่อข้อมูลภายใน Widget เปลี่ยน **เช่น** Checkbox , Textfield เป็นต้น

ตัวอย่าง Stateless Widget

```
1 class HelloWorld extends StatelessWidget {  
2   @override  
3   Widget build(BuildContext context) {  
4     return Container();  
5   }  
6 }
```

ตัวอย่าง StatefulWidget

```
1 class HelloWorldWidget extends StatefulWidget {
2   @override
3   _HelloWidgetState createState() => _HelloWidgetState();
4 }
5
6 class _HelloWidgetState extends State<HelloWidget> {
7
8   @override
9   void initState() {
10    // init something.
11    super.initState();
12  }
13
14  @override
15  Widget build(BuildContext context) {
16    return Container();
17  }
18 }
```


สรุป

1.Stateless Widget เป็น Widget ที่ไม่สามารถเปลี่ยนแปลงค่าได้ใช้สำหรับสร้าง Widget แบบคงที่ เช่น ข้อความ ไอคอน เป็นต้น

2.Stateful Widget เป็น Widget ที่สามารถเปลี่ยนแปลงค่าได้หรือทำงานได้หลายสถานะ เช่น Checkbox , Textfield เป็นต้น

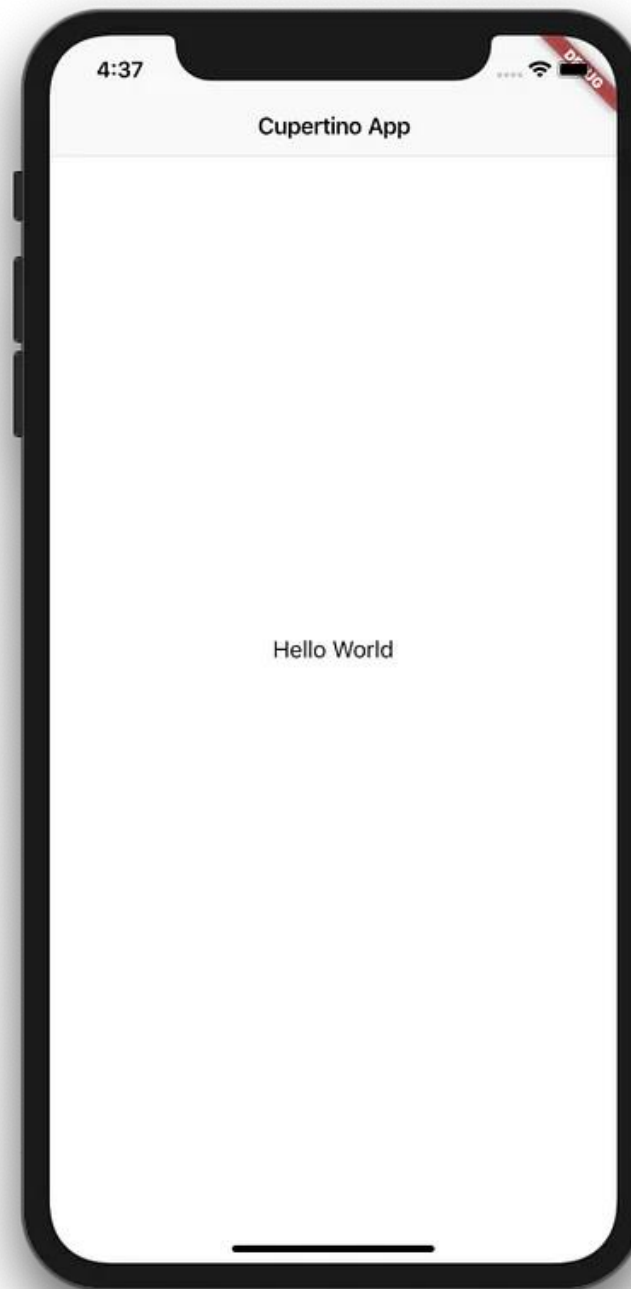
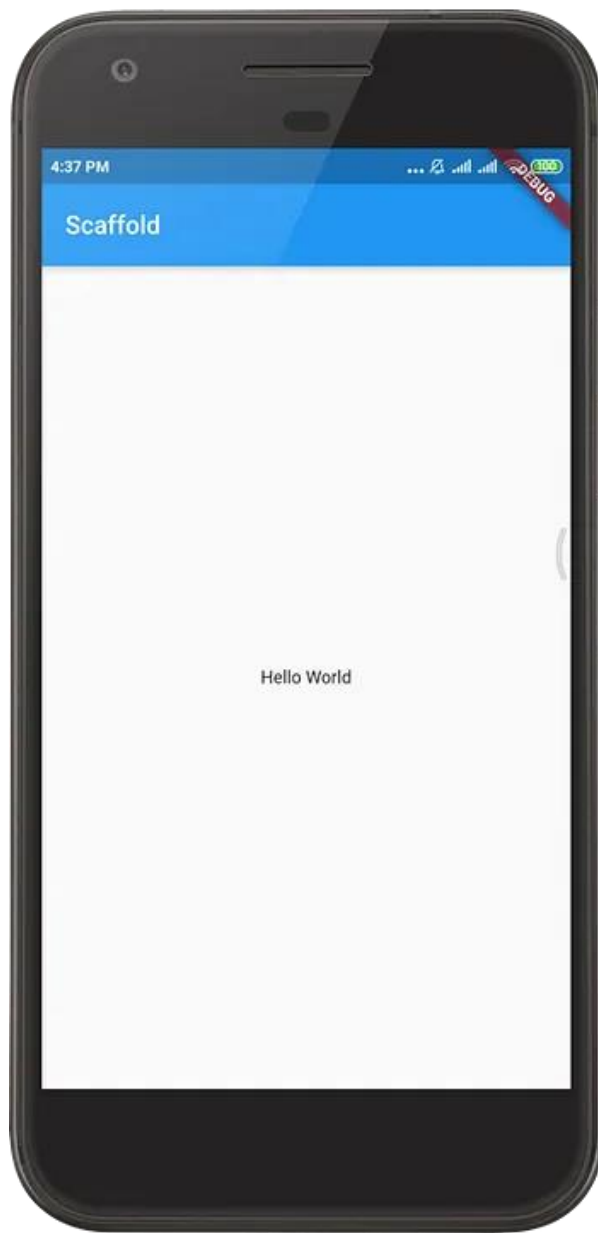
รูปแบบการแสดงผลแอปพลิเคชันที่สร้างจาก Flutter

สามารถแบ่งออกได้เป็น 2 แนวทาง ดังนี้

1. แบบที่ 1 Material Design เป็นการออกแบบหน้าต่างในสไตล์ของ Google
2. แบบที่ 2 Cupertino Design เป็นการออกแบบ หน้าตาแอปพลิเคชันในสไตล์ของ iOS

ตัวอย่างหน้าตาแบบ

การแสดงผลแอปพลิเคชัน



iPhone XR — 12.2

1.cmd folder ที่เราสร้าง

C:\Windows\System32\cmd.exe

Microsoft Windows [Version 10.0.19045.3693]
(c) Microsoft Corporation. All rights reserved.

C:\napaphatFlutter>

2. flutter doctor -v เป็นการตรวจสอบการใช้งาน และเวอร์ชัน ของโปรแกรมต่างๆ ในการสร้างแอป

```
C:\Windows\System32\cmd.exe - flutter doc v - flutter doc -v - flutter doctor -v
```

```
C:\napaphatFlutter>flutter doctor -v
```

```
[✓] Flutter (Channel stable, 3.13.9, on Microsoft Windows [Version 10.0.19045.3693], locale th-TH)
```

- Flutter version 3.13.9 on channel stable at C:\flutter
- Upstream repository <https://github.com/flutter/flutter.git>
- Framework revision d211f42860 (6 weeks ago), 2023-10-25 13:42:25 -0700
- Engine revision 0545f8705d
- Dart version 3.1.5
- DevTools version 2.25.0

```
[✓] Windows Version (Installed version of Windows is version 10 or higher)
```

```
[!] Android toolchain - develop for Android devices (Android SDK version 31.0.0)
```

- Android SDK at C:\Users\U21H1\AppData\Local\Android\Sdk
- Platform android-31, build-tools 31.0.0
- ANDROID_HOME = C:\Users\U21H1\AppData\Local\Android\Sdk
- 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
```

- Chrome at C:\Program Files\Google\Chrome\Application\chrome.exe

3. flutter create ชื่อ project แอปที่ต้องการ หมายเหตุ* ชื่อ project ต้องชื่อติดกัน ไม่เว้นเคาะ

```
C:\napaphatFlutter>flutter create secondapp
Creating project secondapp...
Resolving dependencies in secondapp... (2.4s)
Got dependencies in secondapp.
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 secondapp
$ flutter run

Your application code is in secondapp\lib\main.dart.
```

PC > Local Disk (C:) > napaphatFlutter >

Name	Date modified	Type	Size
firstapp	15/11/2566 20:27	File folder	
secondapp	9/12/2566 9:54	File folder	

Date created: 9/12/2566 9:54
 Size: 513 KB
 Folders: .dart_tool, .idea, android, ios, lib, linux, macos, ...
 Files: .gitignore, .metadata, analysis_options.yaml, ...

4. คู่มือการสร้าง

C > Local Disk (C:) > napaphatFlutter > secondapp >

Name	Date modified	Type	Size
.dart_tool	9/12/2566 9:54	File folder	
.idea	9/12/2566 9:54	File folder	
android	9/12/2566 9:54	File folder	
ios	9/12/2566 9:54	File folder	
lib	9/12/2566 9:54	File folder	
linux	9/12/2566 9:54	File folder	
macos	9/12/2566 9:54	File folder	
test	9/12/2566 9:54	File folder	
web	9/12/2566 9:54	File folder	
windows	9/12/2566 9:54	File folder	
.gitignore	9/12/2566 9:54	Git Ignore Source ...	1 KB
.metadata	9/12/2566 9:54	METADATA File	2 KB
analysis_options.yaml	9/12/2566 9:54	Yaml Source File	2 KB
pubspec.lock	9/12/2566 9:54	LOCK File	6 KB
pubspec.yaml	9/12/2566 9:54	Yaml Source File	4 KB
README.md	9/12/2566 9:54	Markdown Source ...	1 KB
secondapp.iml	9/12/2566 9:54	IML File	1 KB

56
256
ch

5.cd ชื่อ project แอปที่เพิ่งสร้าง

```
C:\napaphatFlutter>cd secondapp
```

```
C:\napaphatFlutter\secondapp>
```


6. ใช้คำสั่ง flutter run เพื่อรันแอป

```
C:\napaphatFlutter\secondapp>flutter run
Connected devices:
Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.19045.3693]
Chrome (web)      • chrome • web-javascript • Google Chrome 119.0.6045.200
Edge (web)        • edge   • web-javascript • Microsoft Edge 119.0.2151.97
[1]: Windows (windows)
[2]: Chrome (chrome)
[3]: Edge (edge)
Please choose one (or "q" to quit):
```

7. เลือก devices

```
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... 42.8s  
This app is linked to the debug service: ws://127.0.0.1:52717/BqB2vH7f1AY=/ws  
Debug service listening on ws://127.0.0.1:52717/BqB2vH7f1AY=/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://127.0.0.1:52717/BqB2vH7f1AY=  
The Flutter DevTools debugger and profiler on Chrome is available at:  
http://127.0.0.1:9100?uri=http://127.0.0.1:52717/BqB2vH7f1AY=
```

8. ได้ผลลัพธ์ดังภาพ

```
C:\Windows\System32\cmd.exe - flutter doc v - flutter doc -v - flutter doctor -v - flutter create secondapp - flutter run
```

```
$ cd secondapp  
$ flutter run
```

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

```
C:\napaphatFlutter>cd secondapp
```

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

Connected devices:

Windows (desktop) • windows • windows-x64 • Microsoft Windows [Version 10.0.190

Chrome (web) • chrome • web-javascript • Google Chrome 119.0.6045.200

Edge (web) • edge • web-javascript • Microsoft Edge 119.0.2151.97

[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... 42.8s

This app is linked to the debug service: ws://127.0.0.1:52717/BqB2vH7f1AY=/ws

Debug service listening on ws://127.0.0.1:52717/BqB2vH7f1AY=/ws

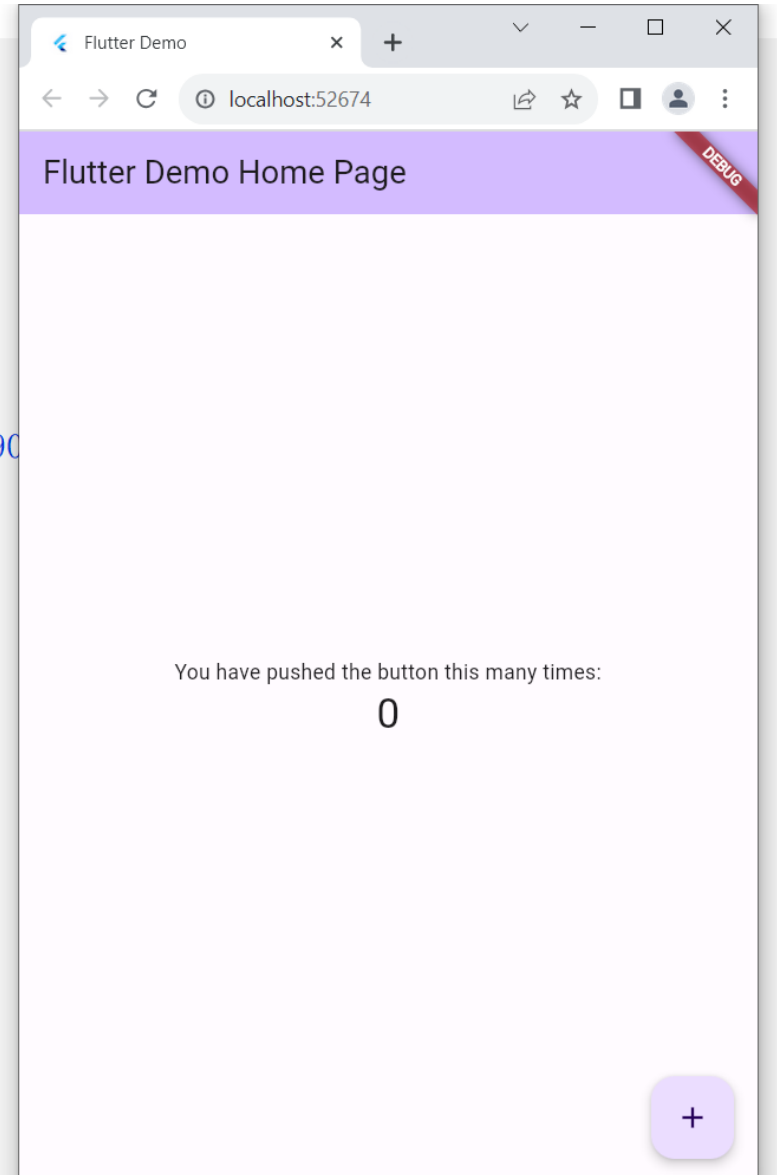
To hot restart changes while running, press "r" or "R".

For a more detailed help message, press "h". To quit, press "q".

A Dart VM Service on Chrome is available at: http://127.0.0.1:52717/BqB2vH7f1AY=

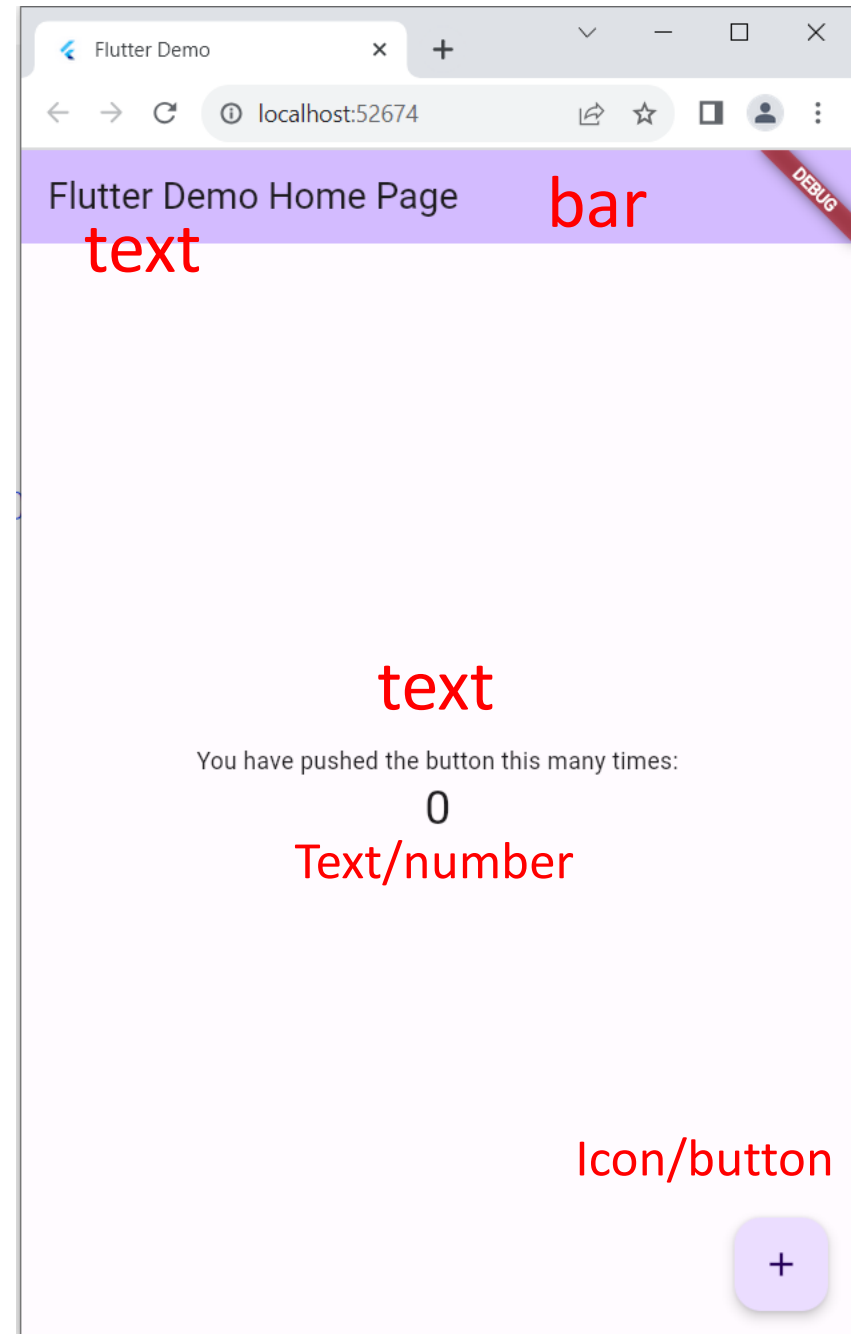
The Flutter DevTools debugger and profiler on Chrome is available at:

http://127.0.0.1:9100?uri=http://127.0.0.1:52717/BqB2vH7f1AY=

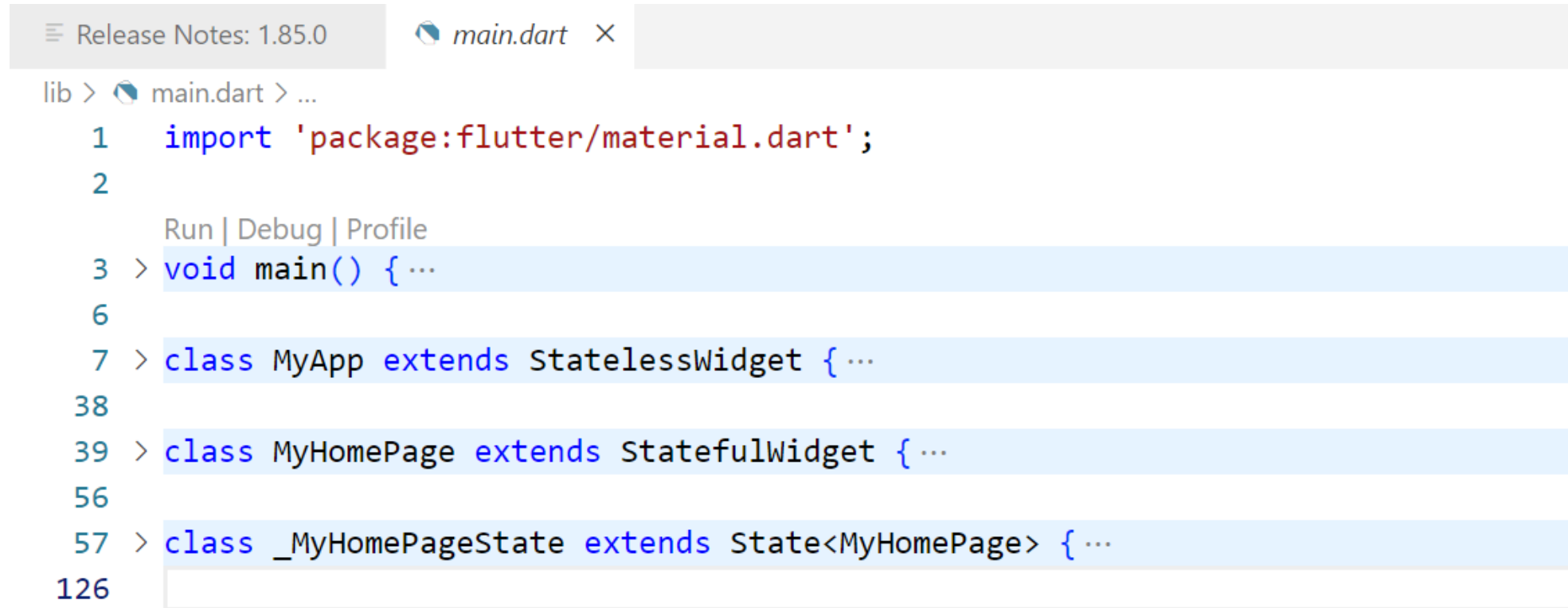


9. ลองดูแต่ละส่วนประกอบบนหน้า แอป

หมายเหตุ* ขยาย font กด ctrl+
หรือ - เพื่อเพิ่มลดขนาดตัวอักษร



10. เปิดโค้ด ใน Visual Studio Code ทำความเข้าใจโค้ด



```
lib > main.dart > ...  
1 import 'package:flutter/material.dart';  
2  
Run | Debug | Profile  
3 > void main() { ...  
6  
7 > class MyApp extends StatelessWidget { ...  
38  
39 > class MyHomePage extends StatefulWidget { ...  
56  
57 > class _MyHomePageState extends State<MyHomePage> { ...  
126
```

หมายเหตุ* ขยาย font กด **ctrl+** หรือ - เพื่อเพิ่มลดขนาดตัวอักษร

จัด Layout หน้าแอปด้วย Scaffold Widget

- Scaffold Widget เป็นวิดเจ็ตที่ใช้กำหนดโครงสร้างพื้นฐานสำหรับแอปพลิเคชัน เมื่อกำหนด Scaffold วิดเจ็ตจะได้เค้าโครงการออกแบบตามมาตรฐานของแอปพลิเคชันบนมือถือทั่วไป เช่น สามารถกำหนดแถบด้านบน (เรียกว่า AppBar), สามารถ กำหนดเนื้อหาไว้ในส่วนของ body, สามารถกำหนดปุ่มที่เรียกว่า FloatingActionButton เป็นต้น

Properties : Child VS Children

`child` takes a single widget

```
child: Text('foo')
```

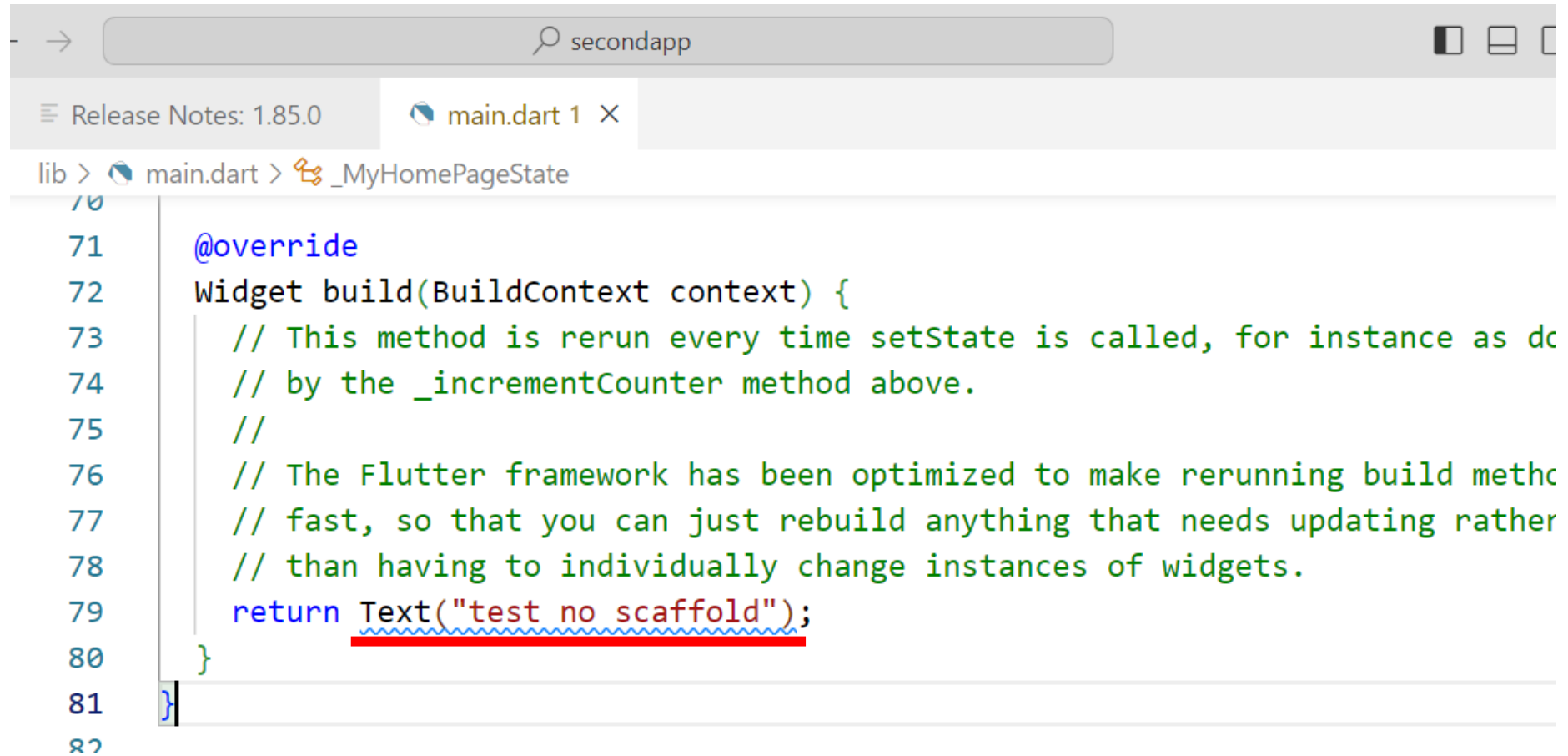
`children` takes a list of widgets

```
children: <Widget>[Text('foo'), Text('bar')]
```

11. ลองเอา Scaffold ออก

```
79 return Scaffold(  
80   appBar: AppBar(  
81     // TRY THIS: Try changing the color here to a specific color (to  
82     // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar  
83     // change color while the other colors stay the same.  
84     backgroundColor: Theme.of(context).colorScheme.inversePrimary,  
85     // Here we take the value from the MyHomePage object that was created by  
86     // the App.build method, and use it to set our appBar title.  
87     title: Text(widget.title),  
88   ), // AppBar  
89   body: Center(  
90     // Center is a layout widget. It takes a single child and positions it  
91     // in the middle of the parent.  
92     child: Column(  
93       // Column is also a layout widget. It takes a list of children and  
94       // arranges them vertically. By default, it sizes itself to fit its  
95       // children horizontally, and tries to be as tall as its parent.  
96       //  
97       // Column has various properties to control how it sizes itself and  
98       // how it positions its children. Here we use mainAxisAlignment to  
99       // center the children vertically; the main axis here is the vertical  
100      // axis because Columns are vertical (the cross axis would be  
101      // horizontal).  
102      //  
103      // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
```


12. แก้เป็นดังภาพ



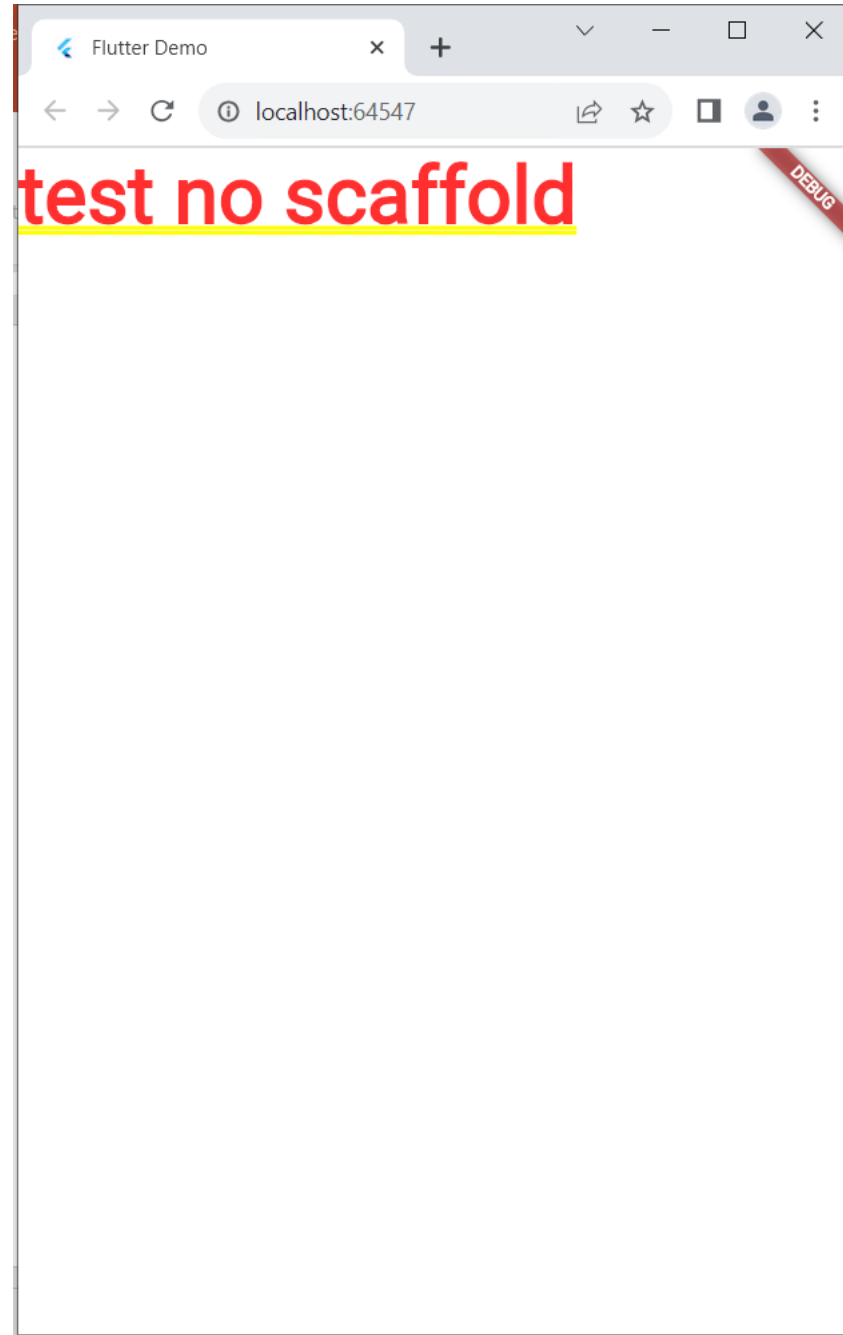
```
lib > main.dart > _MyHomePageState
70
71 @override
72 Widget build(BuildContext context) {
73     // This method is rerun every time setState is called, for instance as done
74     // by the _incrementCounter method above.
75     //
76     // The Flutter framework has been optimized to make rerunning build methods
77     // fast, so that you can just rebuild anything that needs updating rather
78     // than having to individually change instances of widgets.
79     return Text("test no scaffold");
80 }
81
82
```

13. ลองรันได้ผลดังภาพ

หมายเหตุ* กด r/shift r เพื่อ restart

r แก้ไขเล็กๆ บางส่วน ใช้เวลาน้อย

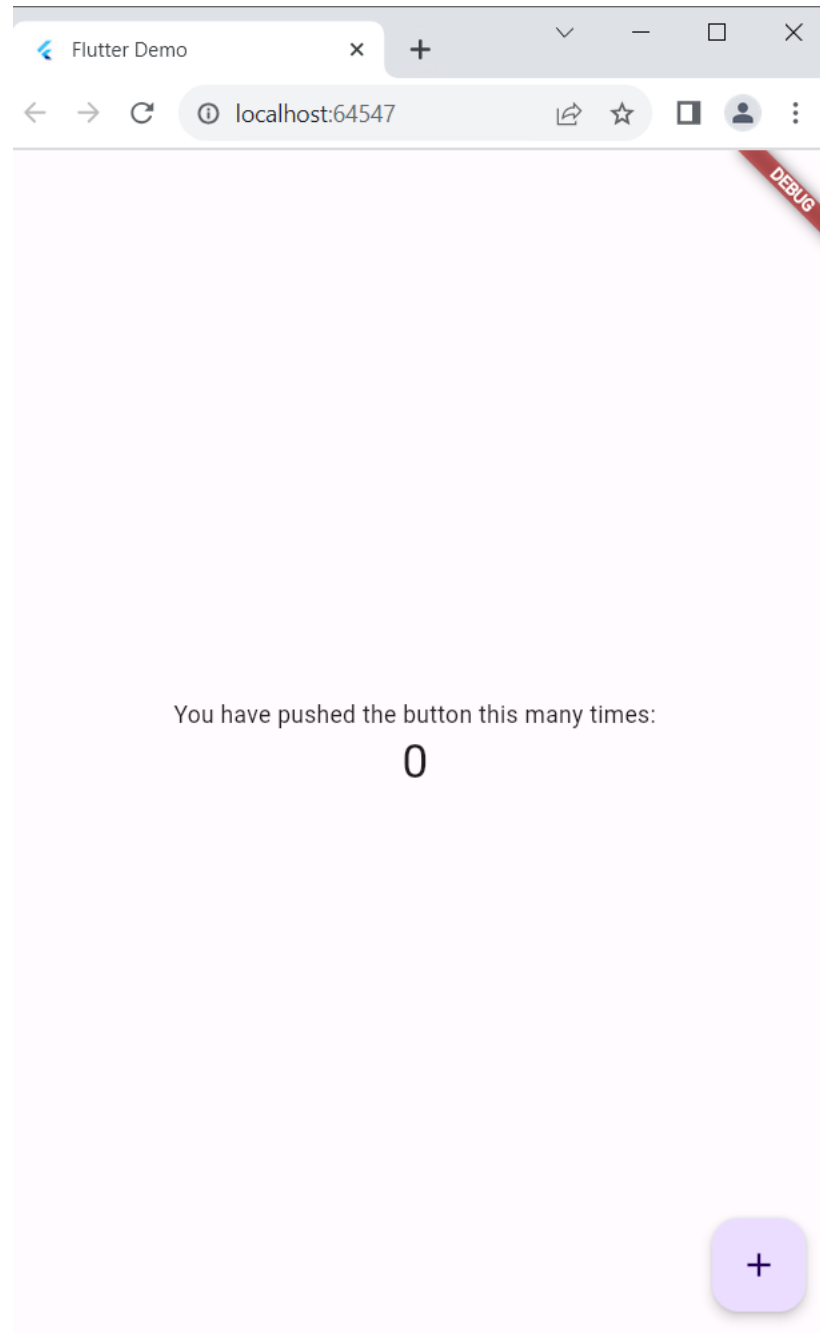
shift r ต้องการ run ทั้งหมด ใช้เวลามากกว่า r



14. ctrl+z กลับโค้ดมาเหมือนเดิม จากนั้น ลองเอา appBar ออก

```
main.dart •
lib > main.dart > _MyHomePageState > build
73 // This method is rerun every time setState is called, for instance as done
74 // by the _incrementCounter method above.
75 //
76 // The Flutter framework has been optimized to make rerunning build methods
77 // fast, so that you can just rebuild anything that needs updating rather
78 // than having to individually change instances of widgets.
79 return Scaffold(
80   appBar: AppBar(
81     // TRY THIS: Try changing the color here to a specific color (to
82     // Colors.amber, perhaps?) and trigger a hot reload to see the AppBar
83     // change color while the other colors stay the same.
84     backgroundColor: Theme.of(context).colorScheme.inversePrimary,
85     // Here we take the value from the MyHomePage object that was created by
86     // the App.build method, and use it to set our appBar title.
87     title: Text(widget.title),
88   ), // AppBar
89   body: Center(
90     // Center is a layout widget. It takes a single child and positions it
91     // in the middle of the parent.
```

15. ลองรันได้ผลดังภาพ



16. จากนั้น ลองเอา center หลัง body ออก

```
lib > main.dart > _MyHomePageState > build
80
81 body: Center(
82     // Center is a layout widget. It takes a single child and positions it
83     // in the middle of the parent.
84     child: Column(
85         // Column is also a layout widget. It takes a list of children and
86         // arranges them vertically. By default, it sizes itself to fit its
87         // children horizontally, and tries to be as tall as its parent.
88         //
89         // Column has various properties to control how it sizes itself and
90         // how it positions its children. Here we use mainAxisAlignment to
91         // center the children vertically; the main axis here is the vertical
92         // axis because Columns are vertical (the cross axis would be
93         // horizontal).
94         //
95         // TRY THIS: Invoke "debug painting" (choose the "Toggle Debug Paint"
96         // action in the IDE, or press "p" in the console), to see the
97         // wireframe for each widget.
98         mainAxisAlignment: MainAxisAlignment.center,
99         children: <Widget>[
100             const Text(
101                 'You have pushed the button this many times:',
102             ), // Text
103             Text(
104                 '$_counter',
105             style: Theme.of(context).textTheme.headlineMedium
```

17. พิมพ์โค้ดดังภาพ

```
lib > main.dart > _MyHomePageState > build
69     }
70
71     @override
72     Widget build(BuildContext context) {
73         // This method is rerun every time setState is called, for instance as done
74         // by the _incrementCounter method above.
75         //
76         // The Flutter framework has been optimized to make rerunning build methods
77         // fast, so that you can just rebuild anything that needs updating rather
78         // than having to individually change instances of widgets.
79         return Scaffold(
80             body: Text("test center"),
81             floatingActionButton: FloatingActionButton(
82                 onPressed: _incrementCounter,
83                 tooltip: 'Increment',
84                 child: const Icon(Icons.add),
85             ), // This trailing comma makes auto-formatting nicer for build methods. // Flo
86         ); // Scaffold
87     }
88 }
```

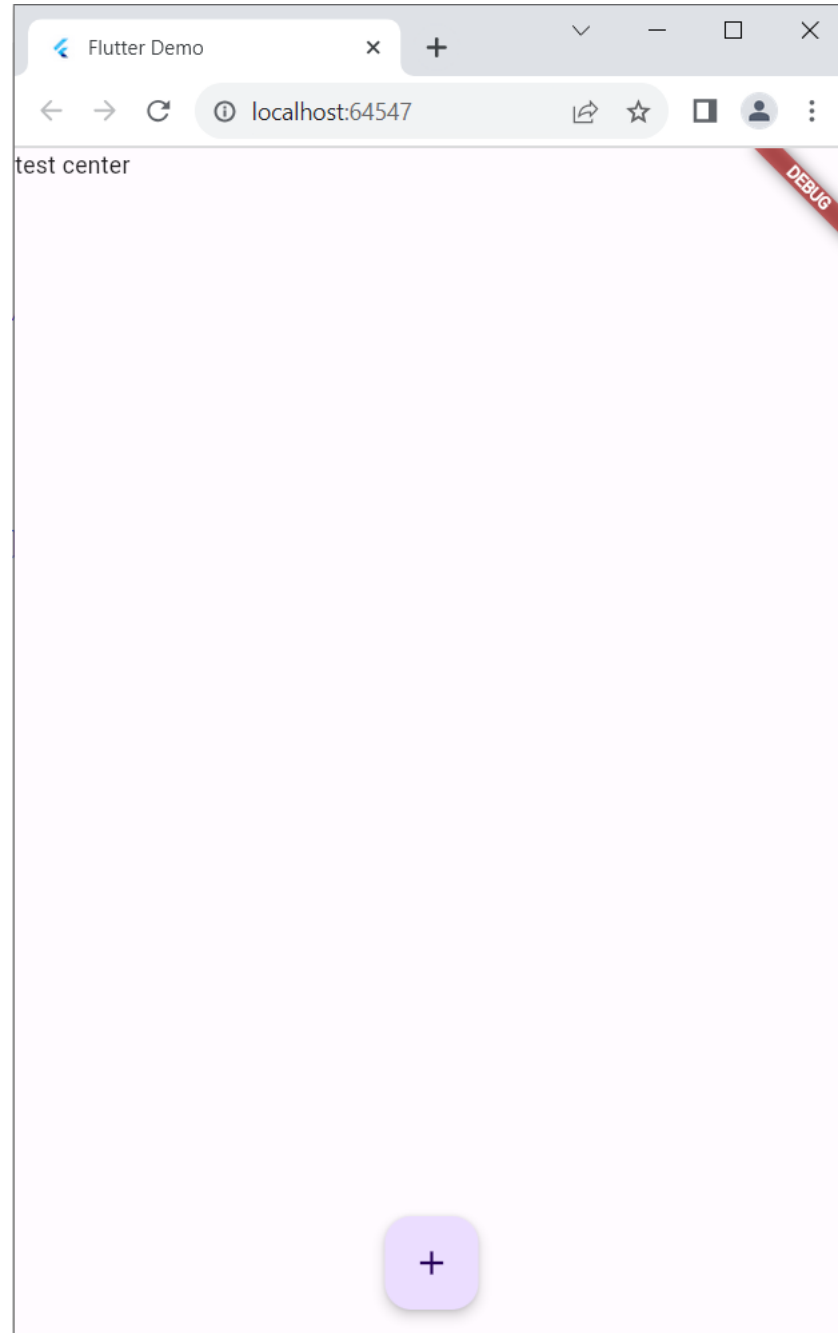
18. ได้ผลการรันดังรูป



19. ลองแก้ไข floatingActionButton เป็นดั่งภาพ

```
81     body: Text("test center"),
82     floatingActionButton: FloatingActionButton(
83       onPressed: _incrementCounter,
84       tooltip: 'Increment',
+ 85       child: const Icon(Icons.add),
86     ), // FloatingActionButton
87     floatingActionButtonLocation: FloatingActionButtonLocation.centerFloat,// This trailing comma
88   ); // Scaffold
89 }
90 }
91
```


20. ได้ผลการรันดั่งภาพ



21. ลองค้นข้อมูลจาก stackoverflow เพื่อหาวิธีการใส่ตำแหน่งของ FloatingActionButton ใน

How can I move floating action button to the left side of the Screen in Flutter?

Asked 3 years, 4 months ago Modified 1 year, 8 months ago Viewed 19k times

9

flutter flutter-layout floating-action-button

Share Improve this question Follow

edited Jul 25, 2019 at 18:40

asked Jul 25, 2019 at 17:38

James Z 12.1k 10 27 44

Jay Tillu 1,138 5 18 37

Add a comment

The Overflow Blog

- Here's what it's like to develop (Ep. 508)
- Why writing by hand is still t to retain information

Featured on Meta

- The Windows Phone SE site archived
- Stack Gives Back to Open So

2022 Community Moderator

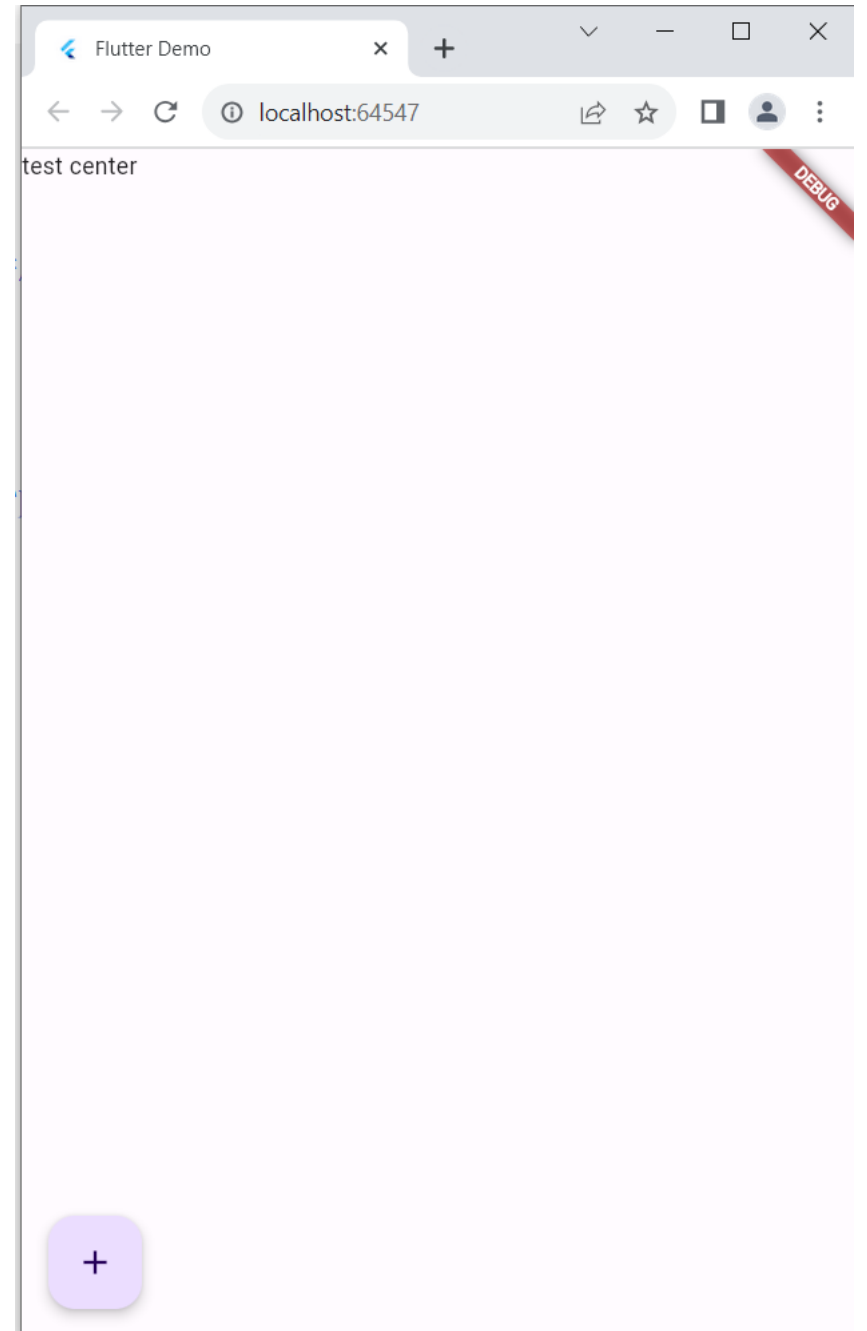
cr. <https://stackoverflow.com/questions/57207469/how-can-i-move-floating-action-button-to-the-left-side-of-the-screen-in-flutter>

22. นำโค้ดที่ได้มาแก้ไขโค้ดเดิม

```
body: Text("test center"),  
floatingActionButton: FloatingActionButton(  
  onPressed: _incrementCounter,  
  tooltip: 'Increment',  
  child: const Icon(Icons.add),  
), // FloatingActionButton  
floatingActionButtonLocation: FloatingActionButtonLocation.startFloat, // This trailing comma  
); // Scaffold  
}
```

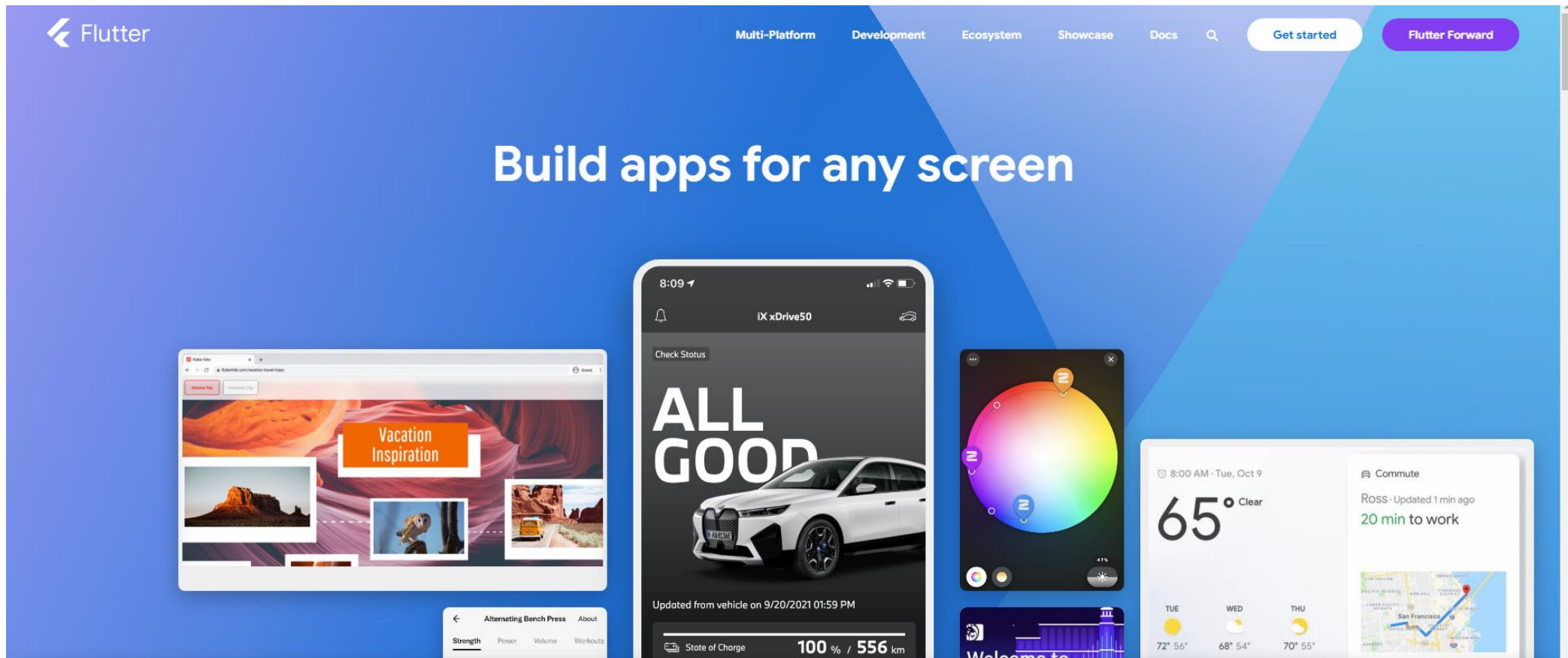
```
}
```

23. ได้ผลการรันดั่งภาพ



24. ลองอ่าน documents เพื่อเปลี่ยนรูปแบบ style font

24.1 ไปที่เว็บ flutter.dev > ไปที่เมนู Docs



Google uses cookies to deliver its services, to personalize ads, and to analyze traffic. You can adjust your privacy controls anytime in your [Google settings](#). [Learn more](#).

Okay

24. ลองอ่าน documents เพื่อเปลี่ยนรูปแบบ style font

24.2 ไป Widget catalog

Flutter

Multi-Platform ▾ Development ▾ Ecosystem ▾ Showcase Docs ▾ 🔍 ✕

SuperDash is the new Flutter + Flame game and example.
Learn how SuperDash was built in 6 days and play the game on iOS, Android, or web!

Get started ▾
Stay up to date ▾
Samples & codelabs ▾
App solutions ▾

User interface
Introduction
Widget catalog
Layout ▾
Design & theming ▾
Interactivity ▾
Assets & media ▾
Navigation & routing ▾
Animations & transitions ▾
Accessibility & ▾

Widget catalog

UI > Widgets



Create beautiful apps faster with Flutter’s collection of visual, structural, platform, and interactive widgets. In addition to browsing widgets by category, you can also see all the widgets in the [widget index](#).

- Accessibility**
Make your app accessible.
[Visit](#)
- Animation and Motion**
Bring animations to your app.
[Visit](#)
- Assets, Images, and Icons**
Manage assets, display images, and show icons.
[Visit](#)
- Async**
Async patterns to your Flutter application.
- Basics**
Widgets you absolutely need to know before building your first Flutter app.
- Cupertino (iOS-style widgets)**
Beautiful and high-fidelity widgets for current iOS design language.

24. ลองอ่าน documents เพื่อเปลี่ยนรูปแบบ style font

24.3 คลิกที่ Text




Text widgets  

UI > Widgets > Text


Display and style text.

See more widgets in the [widget catalog](#).




DefaultTextStyle

The text style to apply to descendant Text widgets without



RichText

The RichText widget displays text that uses multiple different styles.



Text

A run of text with a single style.

24. ลองอ่าน documents เพื่อเปลี่ยนรูปแบบ style font

24.4 คลิกที่ TextStyle จะได้น้ำเว็บดังภาพ โดยสามารถตกแต่ง font ตามรูปแบบในเว็บ

Constructors

```
Text(String data, {Key? key, TextStyle? style, StrutStyle? strutStyle, TextAlign? textAlign, TextDirection? textDirection, Locale? locale, bool? softWrap, TextOverflow? overflow, double? textScaleFactor, int? maxLines, String? semanticsLabel, TextWidthBasis? textWidthBasis, TextHeightBehavior? textHeightBehavior, Color? backgroundColor})
```

Creates a text widget.

const



24. ลองอ่าน documents เพื่อเปลี่ยนรูปแบบ style font

24.5 จะได้น้ำเว็บดังภาพ โดยสามารถตกแต่ง font ตามรูปแบบในเว็บ

TextStyle class



An immutable style describing how to format and paint text.



Bold

Here, a single line of text in a `Text` widget is given a specific style override. The style is mixed with the ambient `DefaultTextStyle` by the `Text` widget.

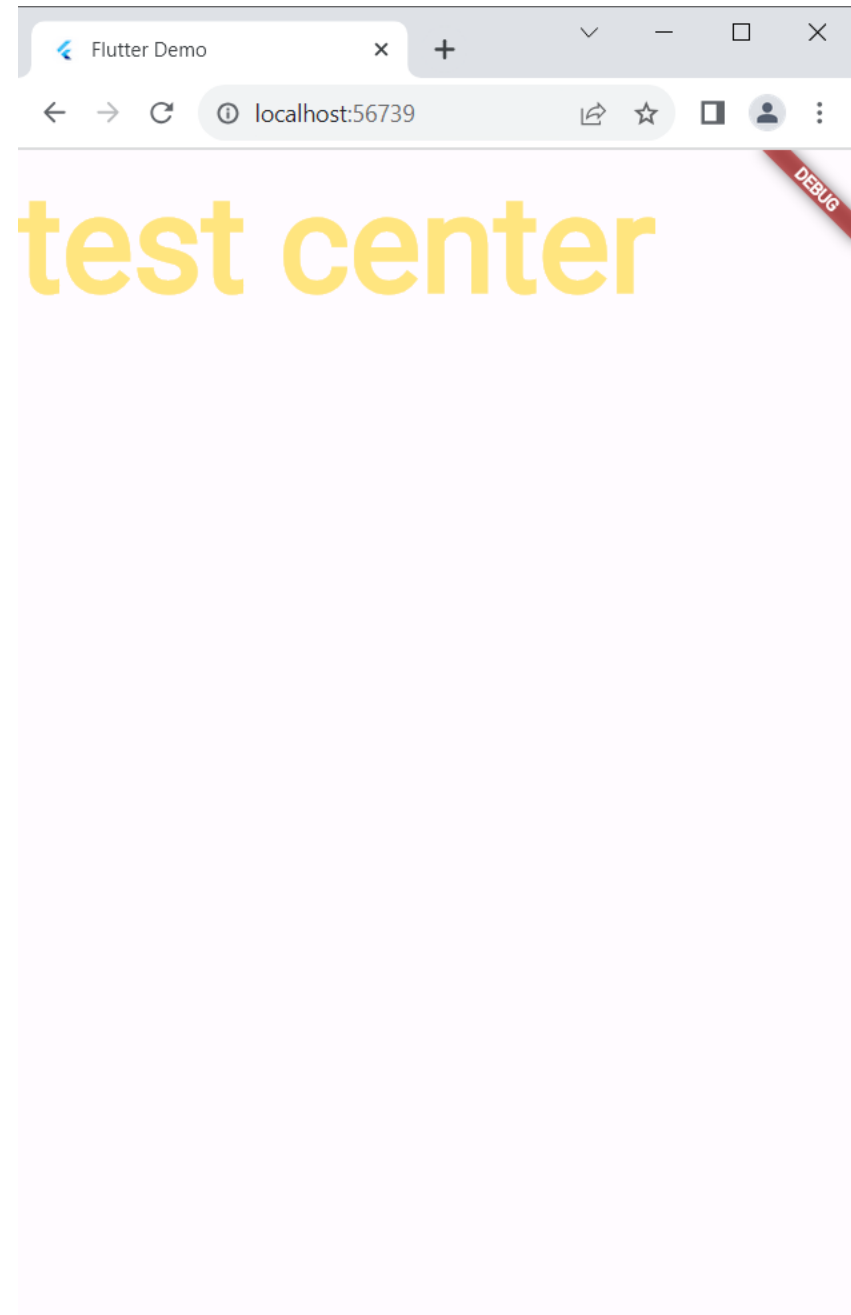


25. กำหนด style ให้กับ text

```
3  
1   body: Text("test center", style: TextStyle(fontSize: 80,  
2     fontWeight: FontWeight.bold, color: Colors.amberAccent[100])), // TextStyle // Text  
3     floatingActionButtonLocation: FloatingActionButtonLocation.startFloat, // This trailing comma  
4   ); // Scaffold  
5 }  
5 }
```

หมายเหตุ* กด **ctrl+spacebar** เพื่อแสดงคำสั่งอัตโนมัติ

26. ได้ผลการรันดั่งภาพ



27. กำหนด text ให้อยู่ตรงกลางด้วย center()

นำ **copy** โค้ด **text style** ก่อนหน้าไว้ แล้วนำมาใส่หลังจาก ใส่ **center** แต่ก่อนวางให้เพิ่มโค้ด **child**

```
80  
81   body: Center(child: Text("test center", style: TextStyle(fontSize: 80,  
82 fontWeight: FontWeight.bold, color: Colors.amberAccent[100]))), // TextStyle // Text // Cent  
83 floatingActionButtonLocation: FloatingActionButtonLocation.startFloat,// This trailing comma  
84 ); // Scaffold  
85 }  
86 }
```

28. ได้ผลการรันดังภาพ

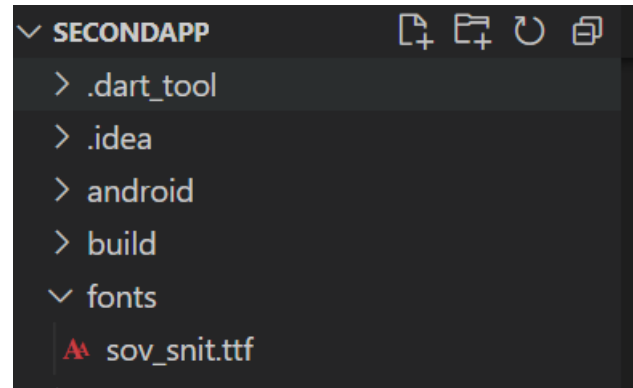


29. กำหนดรูปแบบ font ให้กับ text

download font



สร้าง folder



ไปที่ File ชื่อ pubspec.yaml

```
78 fonts:  
79 - family: weee  
80 fonts:  
81 - asset: fonts/Weee.ttf  
82 # - asset: fonts/Schlyer-Italic.ttf  
83 # style: italic
```

ใส่ fontFamily ที่สร้างใน main.dart

```
74 body: Center(child: Text("สวัสดิ์", style: TextStyle(fontSize: 100, fontWeight:  
75 FontWeight.bold,color: Colors.blue[300], fontFamily: 'weee')),), // TextStyle // Text // Center  
76 floatingActionButton: FloatingActionButton(
```

29. กำหนดรูปแบบ font ให้กับ text

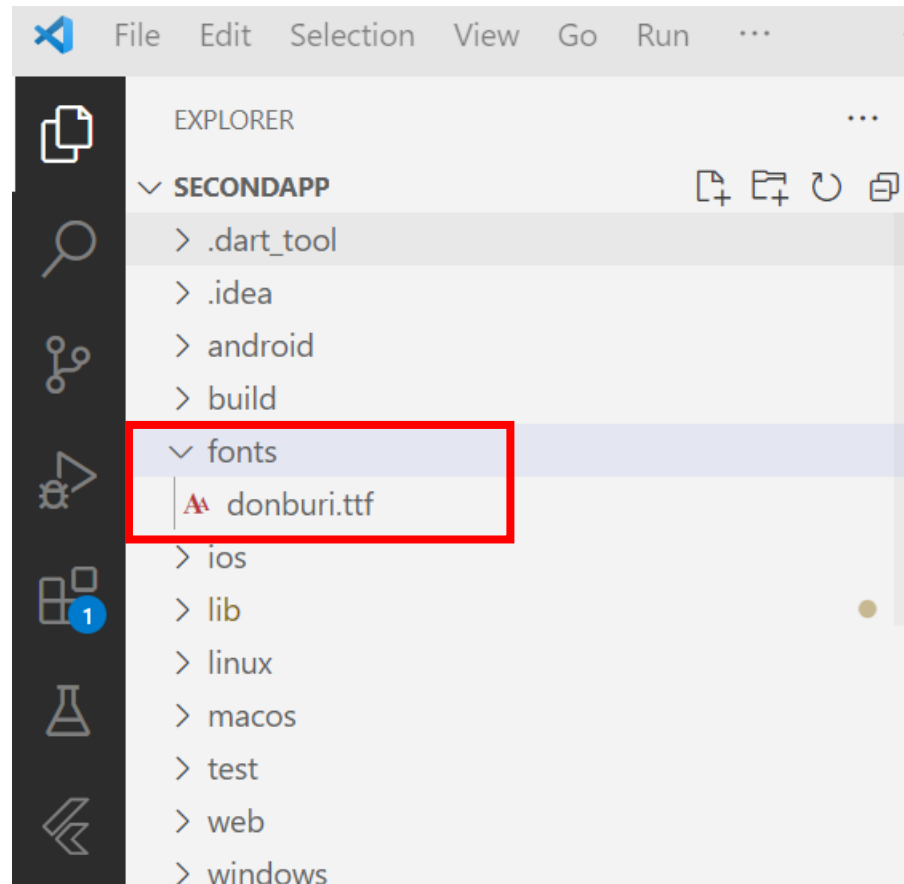
29.1 ไปที่เว็บ <https://www.f0nt.com/>



The screenshot shows the f0nt.com website interface. At the top, there is a dark red header with the 'f0nt' logo and the tagline 'เต็มวิญญูฯ ใช้งานอักษร'. Below the header, there are navigation links: 'ดาวน์โหลดฟอนต์', 'ฟอรัม', 'เกี่ยวกับเรา', and 'สัญญาอนุญาต'. A search bar with the text 'เพิ่มประสิทธิภาพโดย Google' and a 'ค้นหา' button is also present. The main content area features a large advertisement for a font set. The advertisement includes a red tag that says 'แจกฟรี!' (Free!) and '(สำหรับใช้ส่วนตัว)' (For personal use only). The font name 'กาหลง' (KaLong) is prominently displayed in a large, stylized font. Below the font name, it says 'ชุดพิเศษ (MN KaLong)' and 'โดย มานี มีฟอนต์ • 08/12/2566'. The advertisement also mentions 'แจกฟรี!! สำหรับใช้ส่วนตัวกับงานที่ไม่เกิดรายได้เท่านั้น ถูกใจ! ต้องการใช้เชิงพาณิชย์ ราคาเริ่มต้น 290.-'.

29. กำหนดรูปแบบ font ให้กับ text

29.2 สร้าง folder fonts นำ font ที่โหลดมาใส่



29. กำหนดรูปแบบ font ให้กับ text

29.3 ไปที่ File ชื่อ pubspec.yaml เพื่อกำหนด font

```
77 fonts:
78 |   - family: donburi
79 |     fonts:
80 |       - asset: fonts/donburi.ttf
81 |   #       - asset: fonts/Schyler-Italic.ttf
82 |     #       style: italic
83 |   # - family: Trajan Pro
84 |     # fonts:
85 |     #   - asset: fonts/TrajanPro.ttf
86 |     #   - asset: fonts/TrajanPro_Bold.ttf
87 |     #   weight: 700
88 |   #
```

29. กำหนดรูปแบบ font ให้กับ text

29.4 ไปที่ File main.dart เพื่อกำหนดรูปแบบ

```
79   return Scaffold(  
80     body: Center(child: Text("test center", style: TextStyle(fontSize: 80,  
81       fontWeight: FontWeight.bold, color: Colors.amber[100], fontFamily: "donburi")),), // TextSty  
82     floatingActionButton: FloatingActionButton(  
83       onPressed: _incrementCounter,  
84       tooltip: 'Increment',  
85       child: const Icon(Icons.add),  
86     ),floatingActionButtonLocation: FloatingActionButtonLocation.startFloat, // This trailing comma n  
87   ); // Scaffold  
88 }
```

29. กำหนดรูปแบบ font ให้กับ text

29.5 ดูผลการรัน



Reference

ลุงวิศวกร สอนคำนวณ. Flutter 2 Mobile App Bootcamp 2021 - Uncle Engineer
KongRuksia Official. <https://www.youtube.com/c/KongRuksiamOfficial>
Toupawa Studio.
พื้นฐานการออกแบบ UI ใน Flutter. [https://toupawa.com/introduction-create-
widget-ui-in-flutter/](https://toupawa.com/introduction-create-widget-ui-in-flutter/)

