



# Flutter

Week 3

4123006 การพัฒนาโปรแกรมประยุกต์บน  
อุปกรณ์เคลื่อนที่ 3 (2-2-5)



# ทบทวน Child กับ Children

```
body: Center(  
  child: Container(  
    padding: const EdgeInsets.all(20),  
    child: Column(  
      children: <Widget>[  
        Container(  
          padding: EdgeInsets.fromLTRB(5, 5, 5, 5),  
          child: TextField(  
            obscureText: true,  
            decoration: InputDecoration(  
              border: OutlineInputBorder(),  
              labelText: 'password',  
            ), // InputDecoration  
          ), // TextField  
        ), // Container  
        RaisedButton(  
          onPressed: () {},  
          color: Colors.blue,  
          child: Text('Submit', style: TextStyle(color: Colors.white),)  
        ), // RaisedButton  
      ], // <Widget>[]  
    ), // Column  
  ), // Container  
) // Center
```

ประกาศ child แบบ column เพื่อเรียงเป็นบันทัดบนลงล่าง

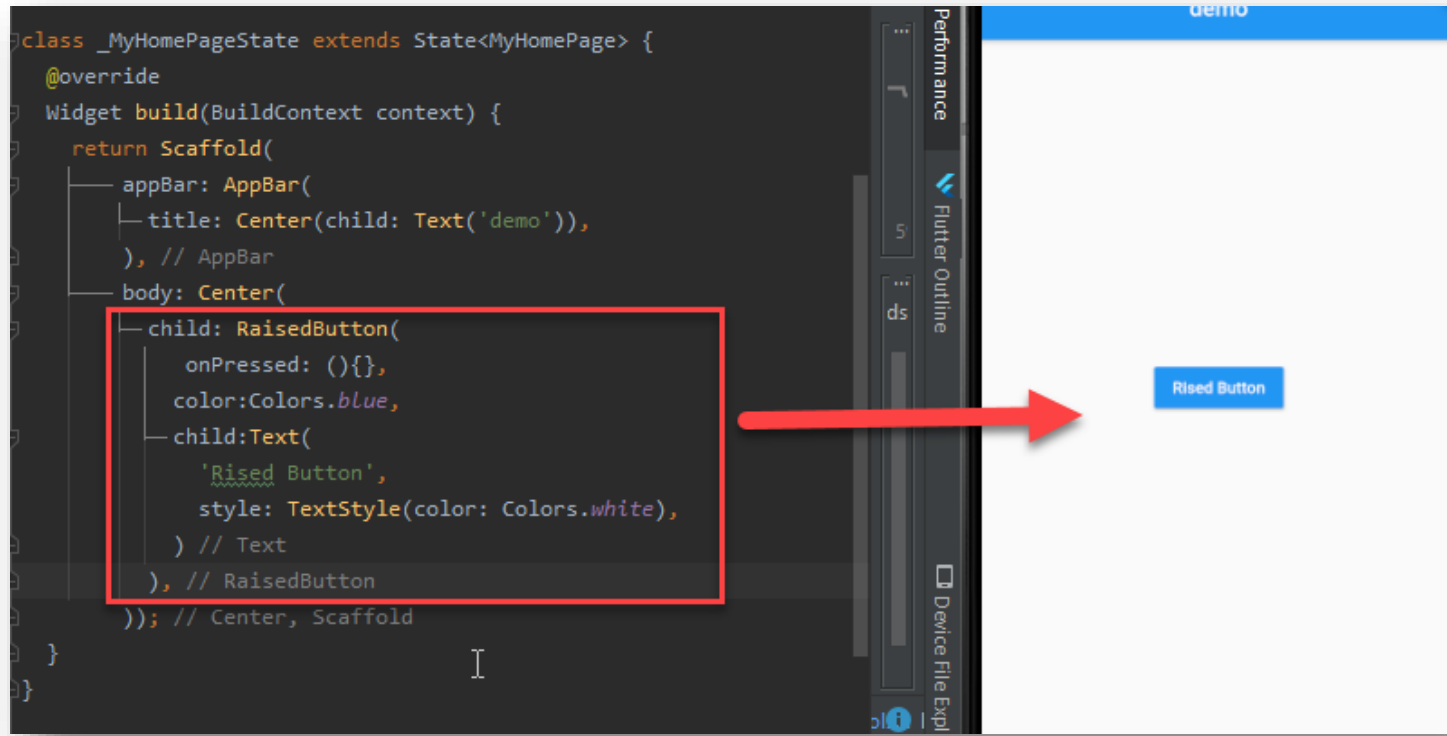
ถ้ามี item มากกว่า 1 จะใช้ children ครอบ

เอา Container ครอบเพราะต้องการจัด padding

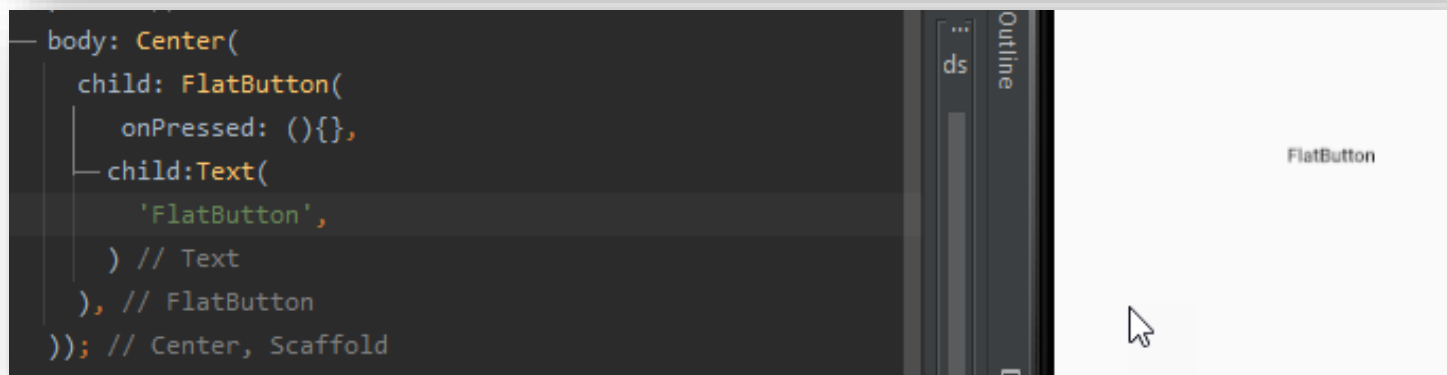
ภายใน children จะไม่ประกาศ child เลย จะต้องประกาศปกติก่อนเพราะติด Widget เช่นเอา Container ขึ้นเลย ภายในค่อยประกาศ child อีกที

# Button

```
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  
  }  
}
```



```
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
```



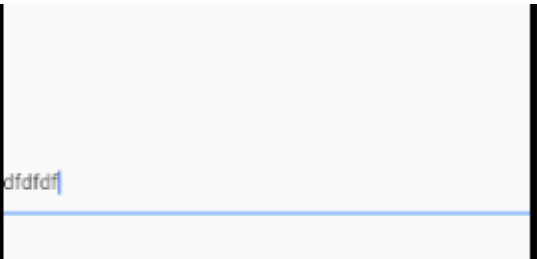
```
body: Center(  
  child: ButtonBar(  
    alignment: MainAxisAlignment.center,  
    children: <Widget>[  
      RaisedButton(  
        onPressed: () => {},  
        color: Colors.green,  
        child: Text('Accept', style: TextStyle(color: Colors.white)),  
      ), // RaisedButton  
      RaisedButton(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



# TextField

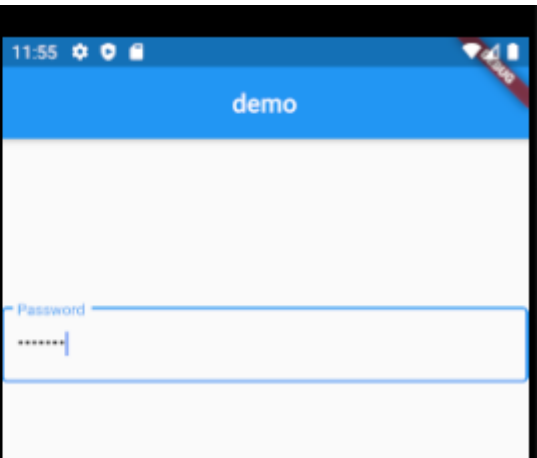
TextField คือ Widget Input ที่จะรับค่าจากการกด key ตัวอักษรลงไปบนระบบ

```
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
```



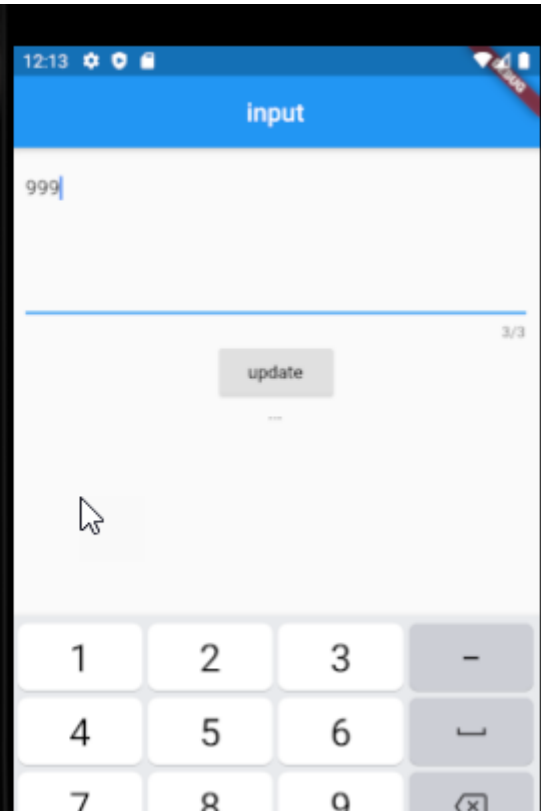
# ตัวอย่างการกรอกข้อมูลใน TextField แล้วกดปุ่มให้แสดงค่าที่กดได้

```
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(() {  
                _input = '123';  
              });  
            } // RaisedButton  
            Text('${_input}'),  
          ], // <Widget>[]  
        ) // Column  
      )); // Container, Scaffold  
    }  
  }  
}
```



# TextField มี Properties ที่น่าสนใจดังนี้

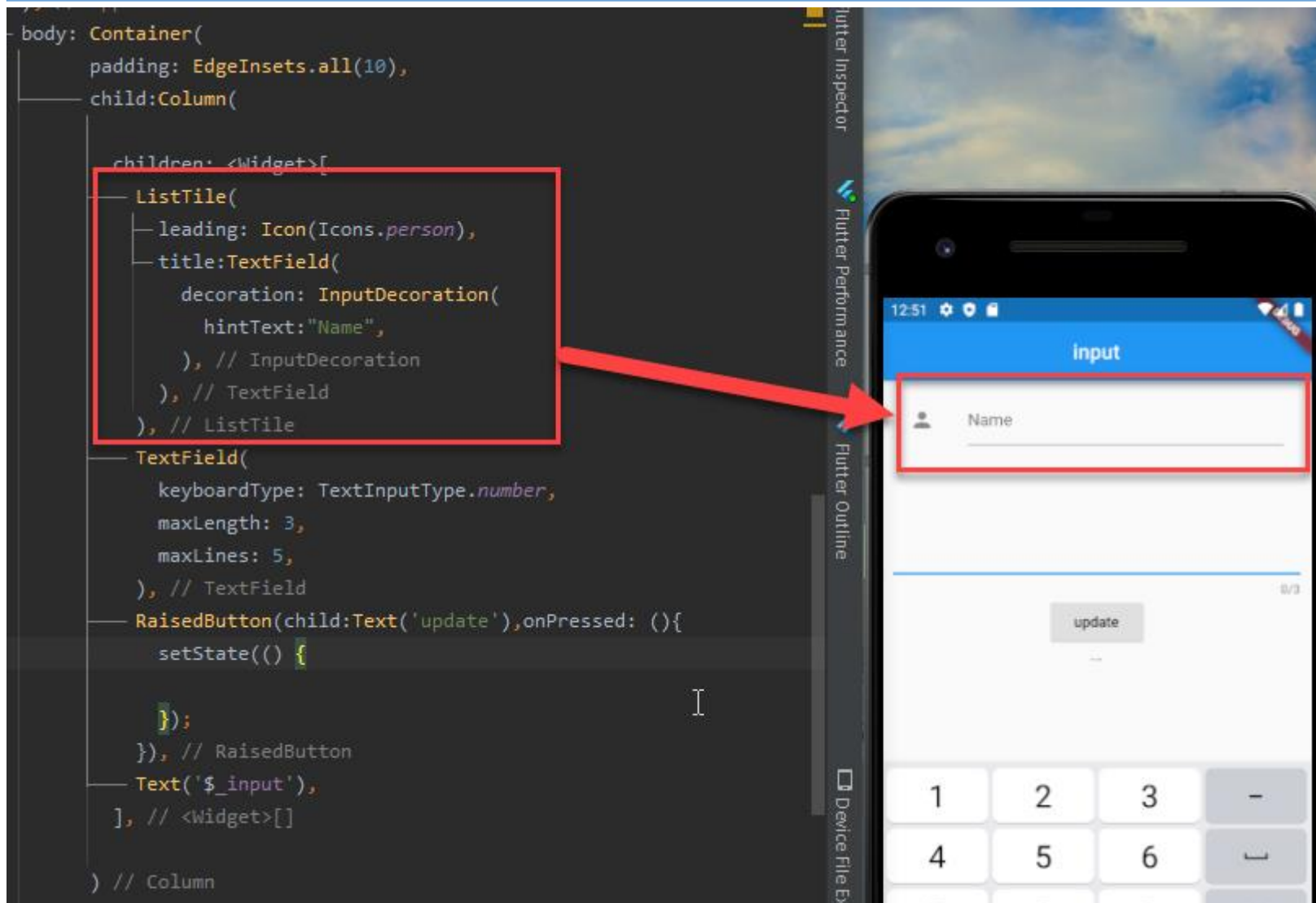
```
body: Container(  
  padding: EdgeInsets.all(10),  
  child: Column(  
    children: <Widget>[  
      TextField(  
        keyboardType: TextInputType.number,  
        maxLength: 3,  
        maxLines: 5,  
      ), // TextField  
      RaisedButton(child:Text('update'),onPressed: (){  
      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



The image displays a Flutter development environment. On the left, the Dart code for a ListTile widget is shown, enclosed in a red box. The code defines a ListTile with a leading icon (person) and a text field with a hint text of "Name". A red arrow points from this code to the right, where a mobile app preview is shown. The app preview, titled "input", displays the rendered ListTile widget on a smartphone screen. The widget consists of a person icon on the left and a text input field with the placeholder text "Name". Below the ListTile, there is a text field with a numeric keyboard type, a button labeled "update", and a text field with the value "\$\_input". The mobile app preview also shows a numeric keypad at the bottom.



# การสร้างปุ่ม button และให้กด click

The image shows an IDE window with the following components:

- Code Editor:** Displays the `main.dart` file with the following code:

```
14  
15  
16 body: SafeArea(  
17   child: Container(  
18     height:100.0,  
19     width:100.0,  
20     margin:EdgeInsets.fromLTRB(20, 50, 10, 20),  
21     color:Colors.red,  
22     child:showButton(),  
23   ), // Container  
24 ), // SafeArea  
25 ), // Scaffold  
26 ); // MaterialApp  
27  
28 Widget showButton(){  
29   return OutlineButton(  
30     child: Text('Submit'),  
31     onPressed: (){  
32       print('clicked.');33     },  
34   ); // OutlineButton  
35 }
```
- Project Explorer:** Shows the project structure for `flutter_app` and `flutter_app_android`.
- Run Console:** Shows the output of the application:

```
I/flutter ( 4476): clicked.  
I/flutter ( 4476): clicked.  
I/chatty ( 4476): uid=10153(com.kungtee.flutter_app) 1.ui identical 1 line  
I/flutter ( 4476): clicked.
```
- Mobile Emulator:** Displays the rendered UI on a smartphone. The screen has a blue header with the text "แอปของฉัน" and a red button with the text "Submit".

Red circles with numbers 1, 2, 3, and 4 are overlaid on the image to highlight specific elements:

- 1:** Points to the `showButton()` call in the `Container` widget.
- 2:** Points to the `showButton()` function definition.
- 3:** Points to the "Submit" button in the emulator.
- 4:** Points to the "clicked." output in the console.

# Widget คืออะไร

ใน Flutter จะมองทุกอย่างเกือบทั้งหมดเป็น widget

Widget คือ ส่วนที่ถูกใช้สร้างเป็นหน้าตาของ App หรือที่เรียกว่า user interface (UI) โดยนำมาประกอบเรียงกันเป็นลำดับขั้นขึ้นเป็นโครงสร้าง แต่ละ widget จะถูกวางซ้อนอยู่ภายใน Parent widget และได้รับการส่งต่อสืบทอดคุณสมบัติต่างๆ จาก Parent อีกที แม้กระทั่ง 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 ก็คือสถานะของสิ่งนั้นๆ stateless จึงหมายถึง widget ที่ไม่มี state หรือไม่มีสถานะการเปลี่ยนแปลง หรือไม่จำเป็นต้องใช้งานการเปลี่ยนแปลง จึงใช้งาน widget นี้ ส่วน stateful หมายถึง widget ที่มี state หรือมีสถานะการเปลี่ยนแปลง ไปตามข้อมูลที่ได้รับหรือจากการกำหนดจากผู้ใช้อื่นๆที่แตกต่างกันที่สำคัญของทั้งสองส่วนนี้คือ stateful widget จะมี State object ที่ใช้ในการเก็บข้อมูล state และ ทำการส่งต่อสำหรับใช้งานในกระบวนการสร้าง widget ใหม่เมื่อมีการเปลี่ยนแปลง ทำให้ค่า state ไม่ได้หายไปไหน

## การใช้งาน StatelessWidget

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

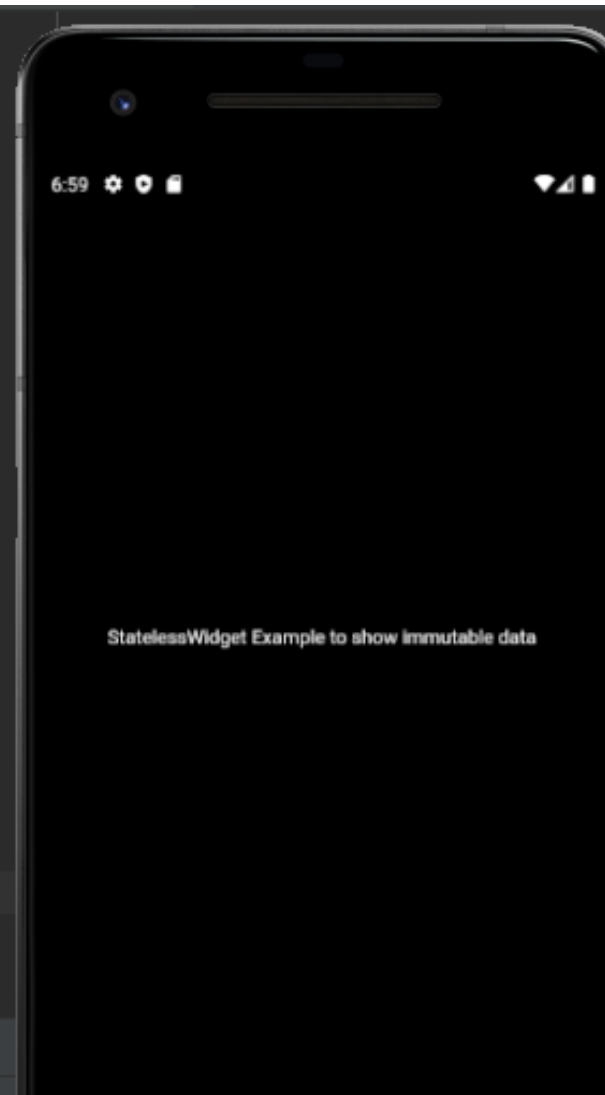
# StatelessWidget ตามโค้ดด้านล่าง

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

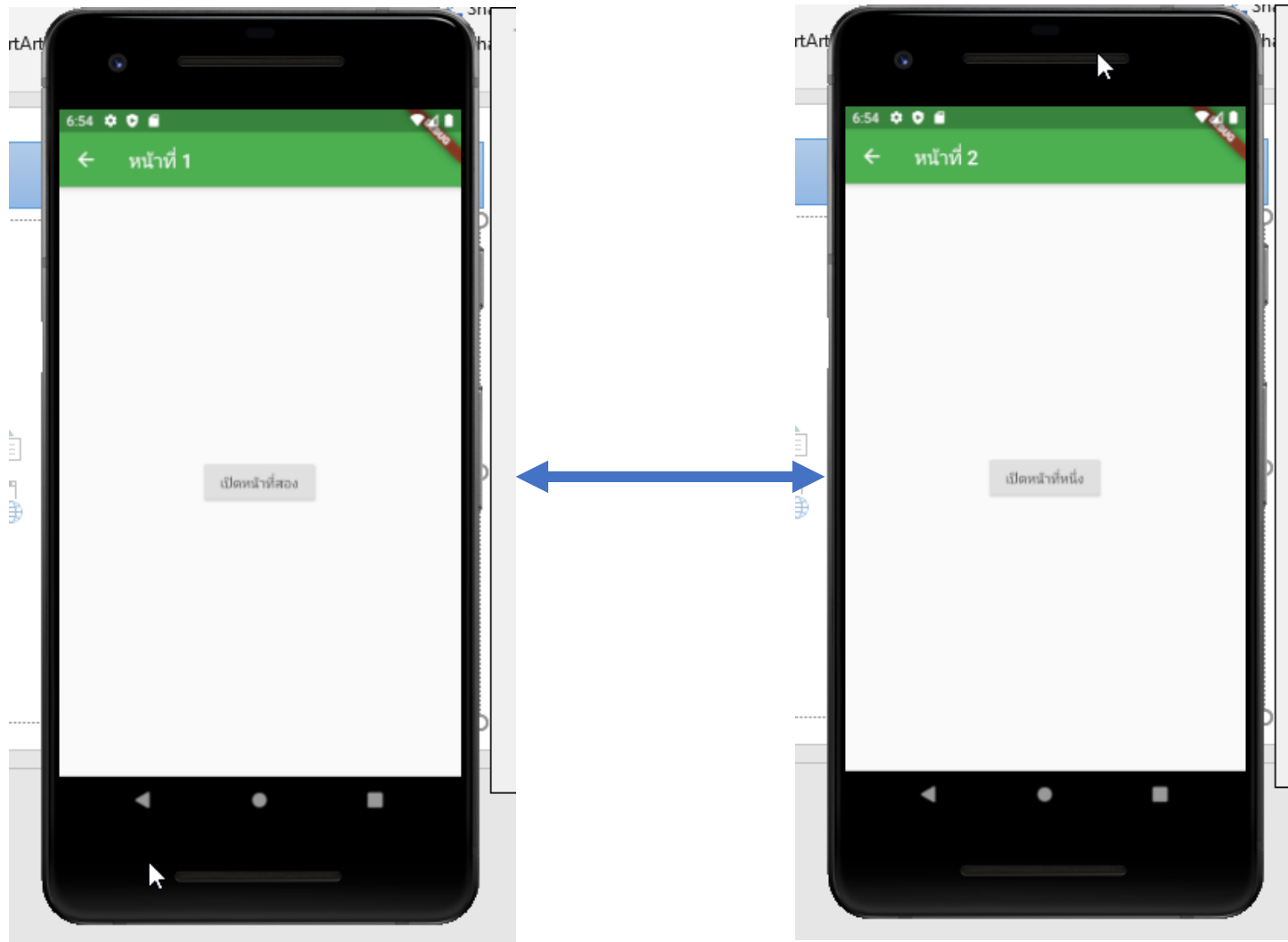
void main(){runApp(
  MyStatelessWidget(text: 'StatelessWidget Example to show immutable data')
);
}

class MyStatelessWidget extends StatelessWidget {
  final String text;
  // constructor
  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 x Page2.dart x Page1.dart x register.dart x
1 import 'package:flutter/material.dart';
2   import 'Page1.dart';
3   import 'Page2.dart';
4
5
6 void main() {
7   runApp(MyApp());
8 }
9
10 class MyApp extends StatelessWidget {
11   // This widget is the root of your application.
12   @override
13   Widget build(BuildContext context) {
14     return MaterialApp(
15       title: 'ทดลองการใช้ Route',
16       theme: ThemeData(
17         primarySwatch: Colors.green,
18       ), // ThemeData
19       initialRoute: '/',
20       routes: {
21         // เมื่อ ให้ "/" route ให้ล้้าหน้าแรก
22         '/': (context) => Page1(title: "หน้าที่ 1",),
23         // เมื่อ ให้ "/page2" route ให้ล้้าหน้าที่สอง
24         '/page2': (context) => Page2(title: "หน้าที่ 2",),
25       },
26     ); // MaterialApp
27   }
28 }
```

# การทำ route เพื่อเปิดอีกหน้า (ต่อ)

- Page1.dart

```
main.dart x Page2.dart x Page1.dart x register.dart x pubspec.yaml x widget_test.dart x
1 import 'package:flutter/material.dart';
2 class Page1 extends StatefulWidget {
3   Page1({Key key, this.title}) : super(key: key);
4
5   final String title;
6
7   @override
8   _Page1State createState() => _Page1State();
9 }
10
11 class _Page1State extends State<Page1> {
12
13   @override
14   Widget build(BuildContext context) {
15     return Scaffold(
16       appBar: AppBar(
17         title: Text(widget.title),
18       ), // AppBar
19       body: Center(
20         child: RaisedButton(
21           child: Text('เปิดหน้าที่สอง'),
22           onPressed: () {
23             // เปิดหน้าที่สองเมื่อมีการกดปุ่ม
24             Navigator.pushNamed(context, '/page2');
25           },
26         ), // RaisedButton
27       ), // This trailing comma makes auto-formatting nicer for build methods. // Center
28     ); // Scaffold
29 }
```

# การทำ route เพื่อเปิดอีกหน้า (ต่อ)

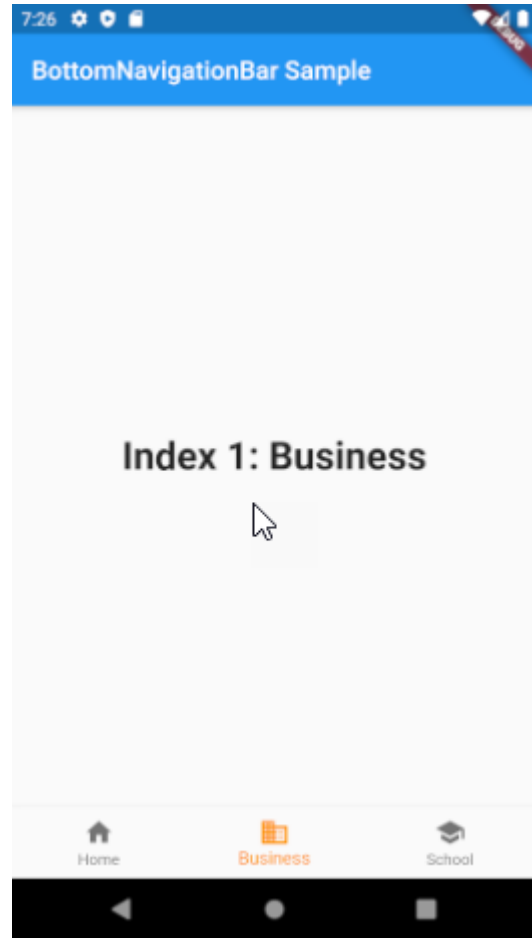
- Page2.dart

```
main.dart x Page2.dart x Page1.dart x register.dart x pubspec.yaml x widget_test.dart x
7   @override
8   _Page2State createState() => _Page2State();
9 }
10
11 class _Page2State extends State<Page2> {
12
13   @override
14   Widget build(BuildContext context) {
15     return Scaffold(
16       appBar: AppBar(
17         title: Text(widget.title),
18       ), // AppBar
19       body: Center(
20         child: RaisedButton(
21           child: Text('เปิดหน้าที่หนึ่ง'),
22           onPressed: () {
23             // เปิดหน้าที่สองเมื่อมีการกดปุ่ม
24             Navigator.pushNamed(context, '/');
25           },
26         ), // RaisedButton
27       ), // This trailing comma makes auto-formatting nicer for build methods. // Center
28     ); // Scaffold
29   }
30 }
```





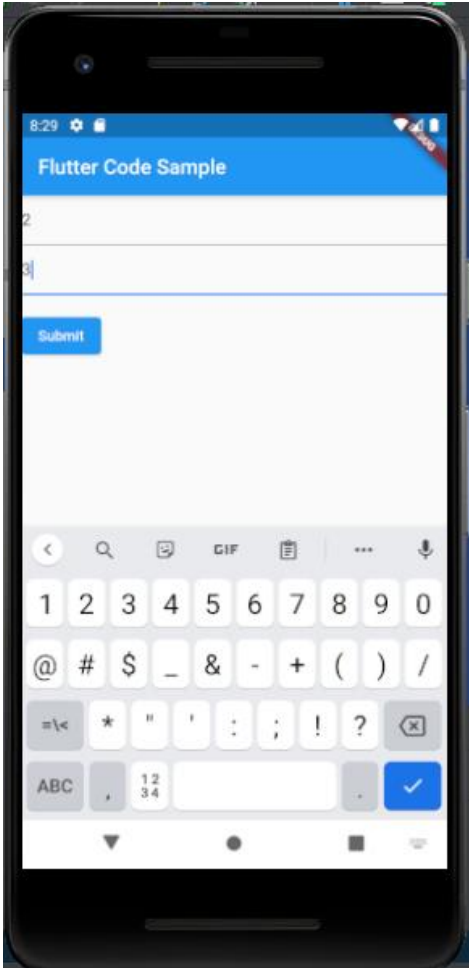
# BottomNavigationBar



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

# Input Text

- การตั้งค่า input ของ text



สร้างจะใช้ TextFormField และใช้ Controller เป็น id

```
crossAxisAlignment: CrossAxisAlignment.start,
children: <Widget>[
  TextFormField(
    controller: _ageController,
    decoration: const InputDecoration(
      hintText: 'Enter your age',
    ), // InputDecoration
    validator: (value) {
      if (value.isEmpty) {
        return 'Please enter some text';
      }
      return null;
    },
  ), // TextFormField
  TextFormField(
    controller: _heightController,
    decoration: const InputDecoration(
      hintText: 'Enter your height',
    ), // InputDecoration
```

และจะสร้างตัวแปล controller มารับค่า input ที่สร้าง

```
/// This is the private State class that goes with MyStatefulWidget.
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  final _formKey = GlobalKey<FormState>();
  final TextEditingController _ageController = TextEditingController();
  final TextEditingController _heightController = TextEditingController();
  @override
  Widget build(BuildContext context) {
    return Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
```

# Week 3 App 1 แอปทำนายตัวเลข



## ขั้นตอนการทำ

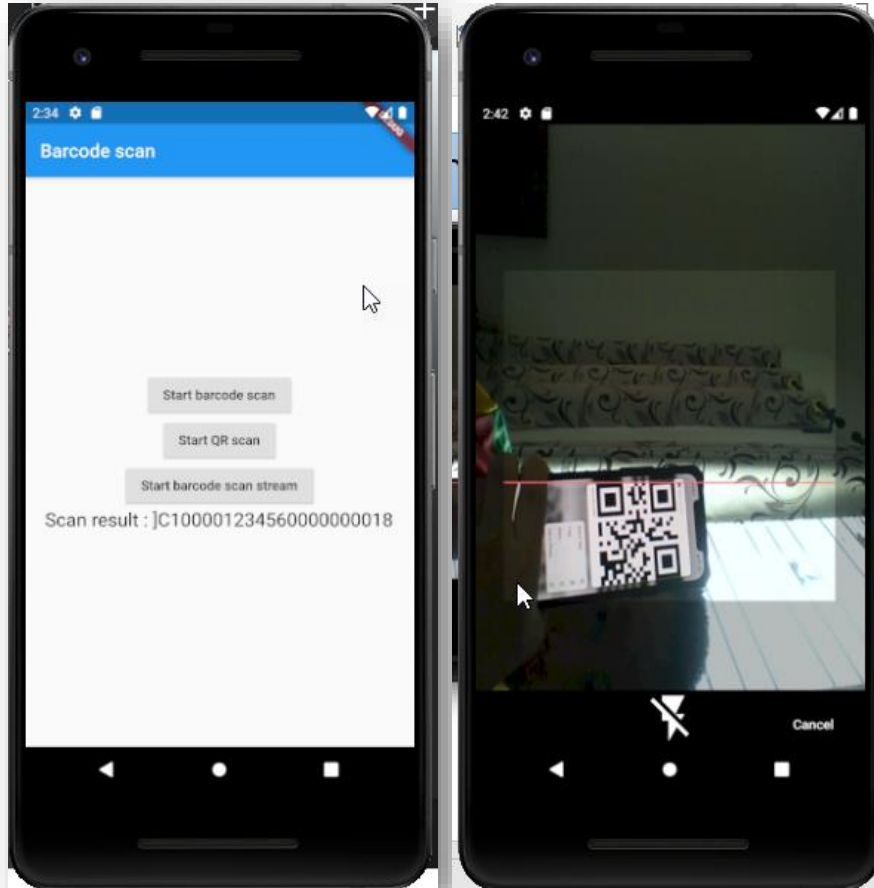
- สร้างแอป Flutter ชื่อ week3\_random\_number โดยมี Widget ต่าง ๆ ดังนี้
  - สร้าง TextField ให้กรอกได้เฉพาะตัวเลข
  - สร้าง Text เพื่อแสดงข้อความผลลัพธ์ที่ได้
  - สร้างปุ่ม ส่งคำตอบสีแดง และ ปุ่มเริ่มใหม่สีน้ำเงิน
- วิธีเล่นคือกรอกตัวเลขที่คิดว่าจะเป็นคำตอบลงไป ใน TextField แล้วกดปุ่มส่งคำตอบ ถ้าส่งผิด จะขึ้นข้อความว่า ผิดคำตอบที่ถูกคือ (ตัวเลขที่แอปเรนดอมขึ้นมา) ถ้าส่งถูกจะขึ้นว่าสุดยอด
- กดปุ่มเล่นใหม่ เพื่อเริ่มเดาตัวเลขใหม่อีกครั้ง

## สิ่งที่ได้

1. สามารถวาง Layout ได้
2. เรียนรู้เรื่อง State และการสร้างตัวแปร ตลอดจนการกำหนด Controller ทำงานกับ TextField และปุ่มกด
3. การเรียก Import class math ในภาษา dart มาเพื่อใช้ random number

ขั้นตอนการทำจะอธิบายได้ในเว็บ [kungtee.com](http://kungtee.com) และวิดีโอประกอบการสอน Week 3 ครับ

# Week 3 App 2 Scan Bar Code



## ขั้นตอนการทำ

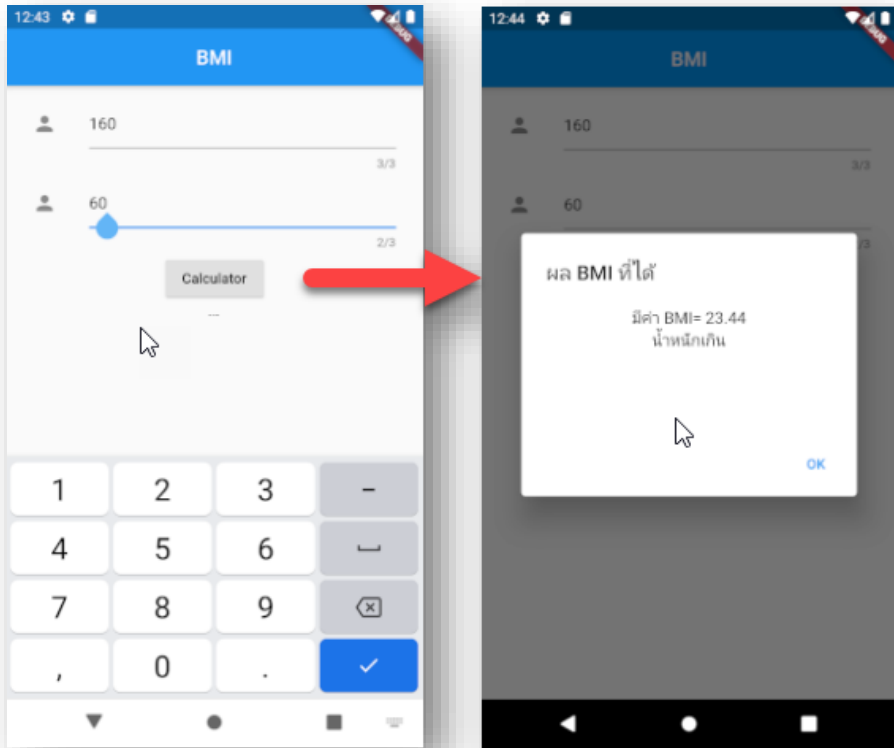
- สร้างแอป Flutter ชื่อ week3\_scanbarcode โดยมี Widget ต่าง ๆ ดังนี้
  - สร้าง ปุ่ม Scan Barcode
  - สร้าง ปุ่ม Scan QR Code
  - สร้าง ปุ่ม Scan Barcode แบบ StreamMode
  - สร้าง Text แสดงผลการ Scan
- การใช้งานกดปุ่มที่ต้องการสแกน กล้องเปิดแล้วนำไฟล์ภาพ Barcode หรือ QrCode มาสแกน

## สิ่งที่ได้

1. สามารถวาง Layout ได้
2. เรียนรู้เรื่อง State เรียนรู้การใช้ Future และ Asynchronous การใช้ plugin flutter\_barcode\_scanner

ขั้นตอนการทำจะอธิบายได้ในเว็บ [kungtee.com](http://kungtee.com) และวิดีโอประกอบการสอน Week 3 BarCode ครับ

# Week 3 App 2 การคำนวณ BMI



## ขั้นตอนการทำ

- สร้างแอป Flutter ชื่อ week3\_scanbarcode โดยมี Widget ต่าง ๆ ดังนี้
  - สร้าง TextField เก็บค่า Width และ Height
  - กำหนดให้กรอกได้แต่ตัวเลข 3 ตัวอักษร
  - สร้าง ปุ่ม กด Calculator
- การใช้งานกรอกน้ำหนักส่วนสูง แล้วกดปุ่ม Calculator คำนวณค่า BMI แล้วแสดง Popup ขึ้นมา

## สิ่งที่ได้

1. สามารถวาง Layout ได้
2. เรียนรู้เรื่อง State เรียนรู้การใช้ showDialog และใช้ Widget AlertDialog ที่สามารถปรับแต่งความสวยงามได้ และเรียนรู้การส่งค่าตัวแปร และการตัดจุดทศนิยม รวมไปถึง การใช้ Navigator.pop เพื่อยกเลิก Widget AlertDialog

ขั้นตอนการทำจะอธิบายได้ในเว็บ [kungtee.com](http://kungtee.com) และวิดีโอประกอบการสอน Week 3 BMI ครับ

# แบ่งกลุ่มทำ Miniproject

- ให้แบ่งกลุ่ม กลุ่มละไม่เกิน 3 คน
- โดยให้ทำ MiniProject พร้อมเล่มรายงาน
- โดยต้องใช้สิ่งที่เรียนรู้จากการเรียนมาใช้ หรือหาความรู้เพิ่มเติมจากอินเทอร์เน็ตแต่ห้าม copy แต่ดัดแปลงเพิ่มฟังก์ชันได้
  1. จะต้องมี Widget TextField, Image, Button, Text หรือประกอบด้วย Widget อื่นๆ
  2. ต้องมีการใช้ Stateful
  3. ต้องมีการใช้ตัวแปรเก็บค่า
  4. ต้องมีการแสดงผลผ่าน Widget
  5. ต้องทำ icon app ด้วย
- ตัวแอป 20 คะแนน เล่มรายงาน 10 คะแนน
- ส่งชื่อกลุ่มและแอปที่จะทำ (กลุ่มไหนส่งก่อนมีสิทธิได้ทำแอปนั้น ห้ามทำซ้ำกันเพื่อจะได้ไม่เกิดการเปรียบเทียบ)
- กำหนดส่งและ present สัปดาห์ที่ 6