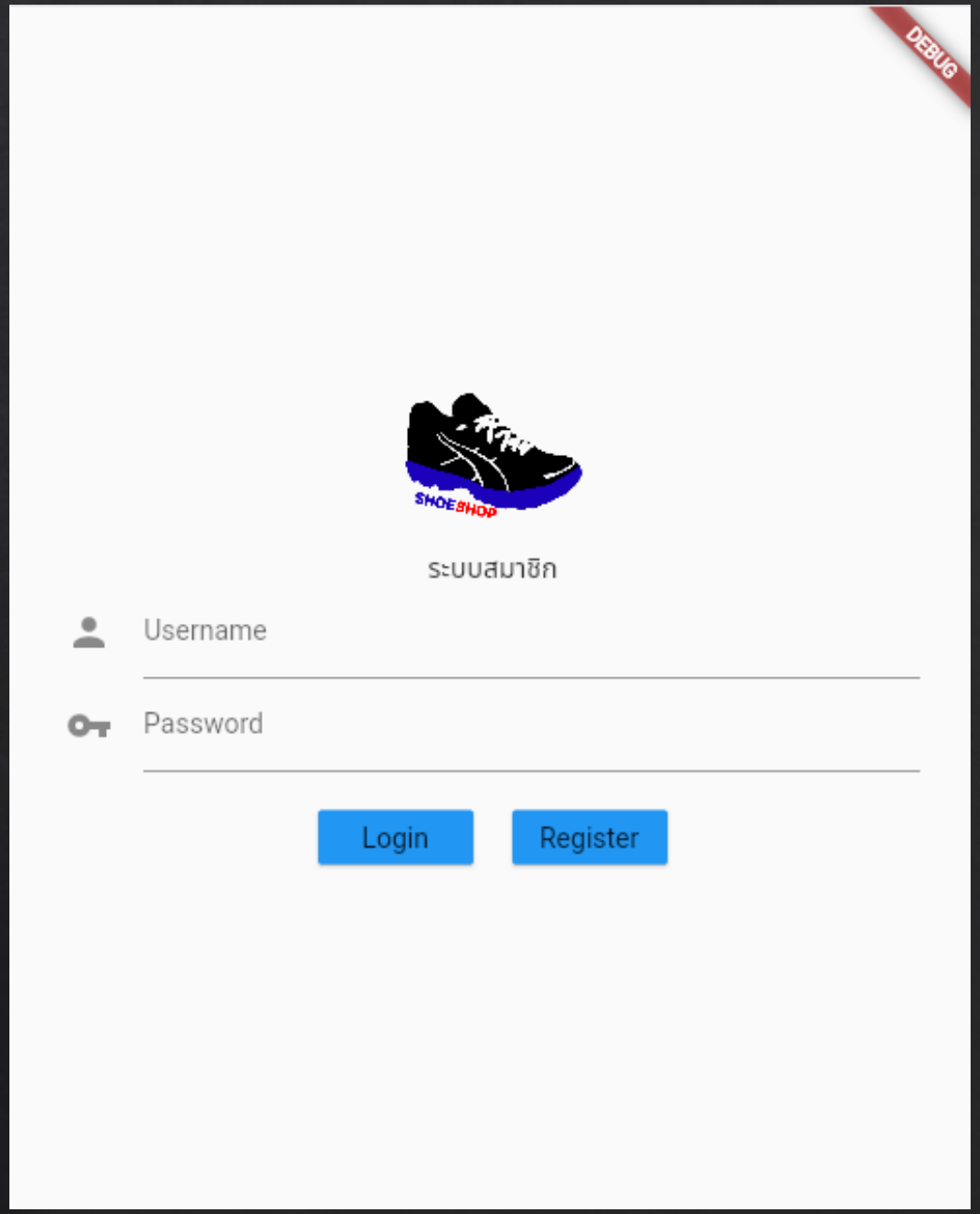
Chapter 4
การรับเข้าข้อมูล

By Aj.Lattagarn Kuikaew

กระบวนการรับข้อมูลเข้า

คือ ขั้นตอนการนำเข้าข้อมูลเพื่อให้แอพพลิเคชั่นสามารถมีข้อมูล**เพื่อใช้ในการประมวลผล** สั่งการ และปฏิบัติตามสิ่งที่นักพัฒนาแอพพลิเคชั่นได้ทำการออกแบบไว้



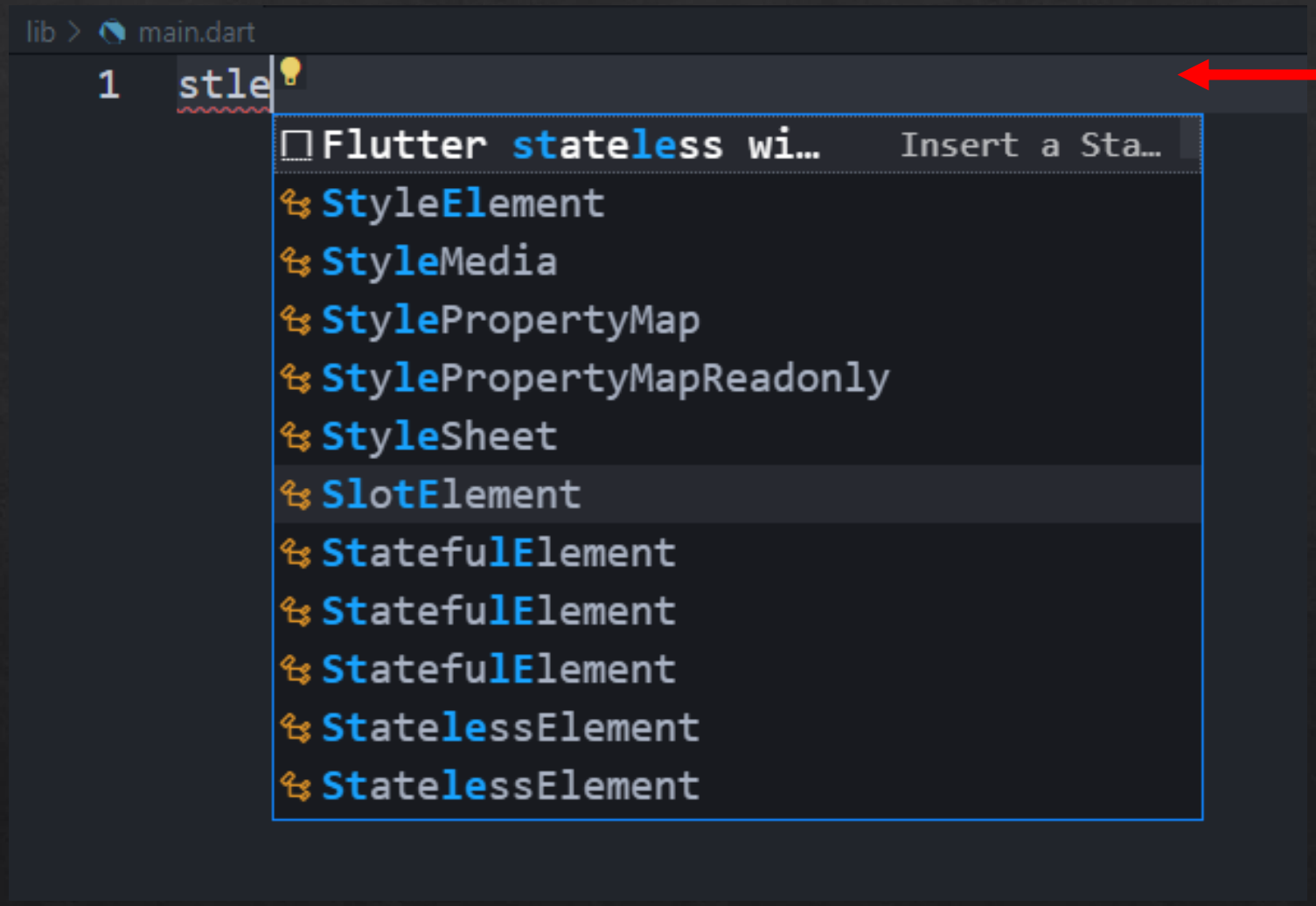
การออกแบบและใช้งาน Widget

- Stateless Widget
- Stateful Widget

Stateless Widget

ทำหน้าที่ในการแสดงผลของวิดเจ็ตต่างๆ ที่มีในแต่ละฉาก **เป็นวิดเจ็ตที่ไม่มีสถานะ** ดังนั้นจึงไม่สามารถโต้ตอบหรือเปลี่ยนแปลงสถานะต่างๆ ได้

ตัวช่วยในการพิมพ์คำสั่ง เรียกใช้งาน Stateless Widget



พิมพ์ stle
แล้วกด Enter

ใส่ชื่อคลาสที่จะสืบทอด
เป็น firstPage

```
7   class  extends StatelessWidget {
8     const ({ Key? key }) : super(key: key);
9
10    @override
11    Widget build(BuildContext context) {
12      return Container(
13
14      );
15    }
16  }
```

ปรับแก้การส่งคืนค่าของฟังก์ชัน
return เป็น Scaffold()

lib > firstPage.dart > ...

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

class firstPage extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return Scaffold();
  }
}
```

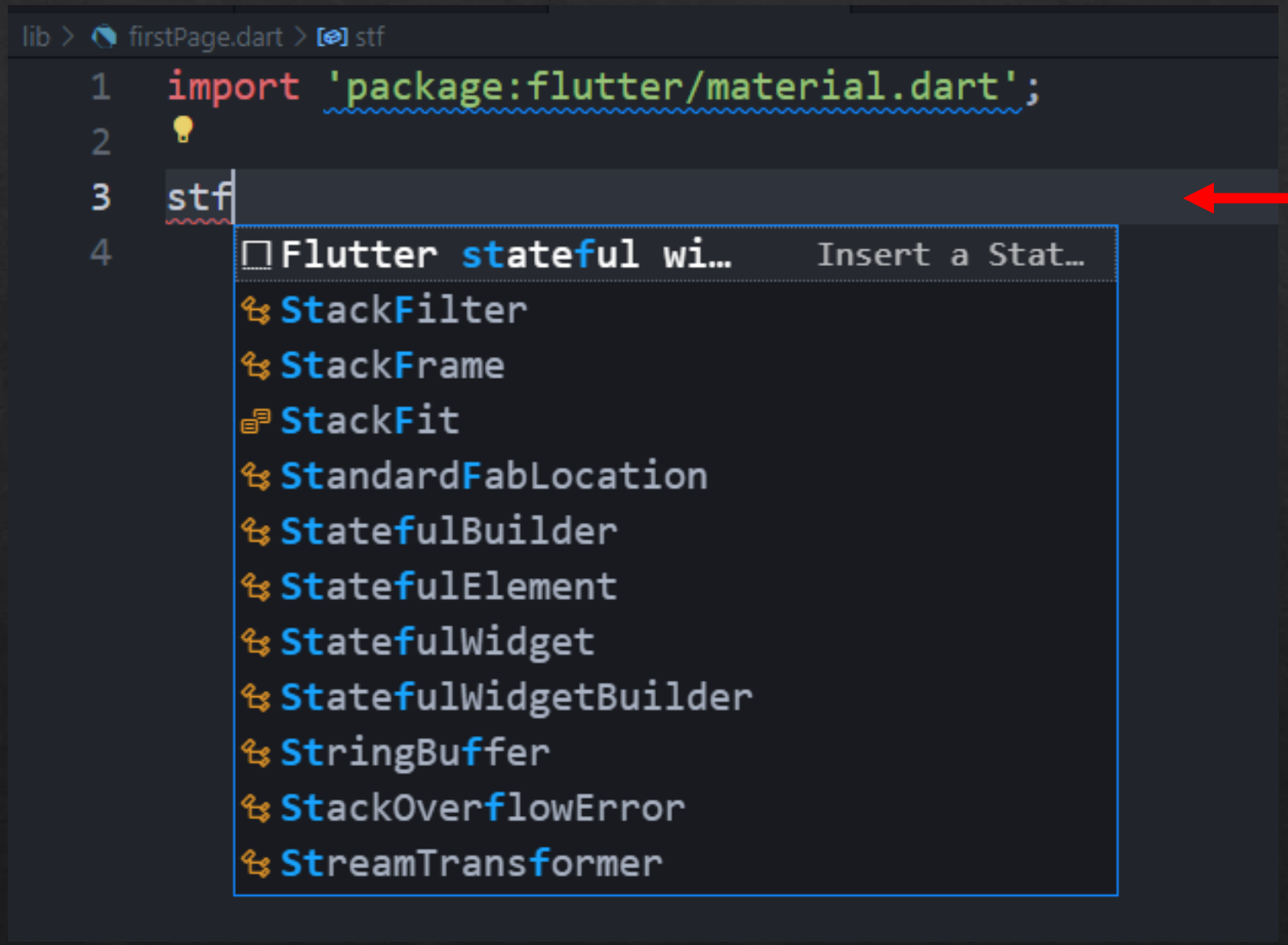
lib > main.dart > ...

```dart
import 'package:test1/firstPage.dart';
import 'package:flutter/material.dart';

Run | Debug | Profile
void main(List<String> args) {
  runApp(MyApp());
}

class MyApp extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: firstPage(),
    ); // MaterialApp
  }
}
```

Stateful Widget

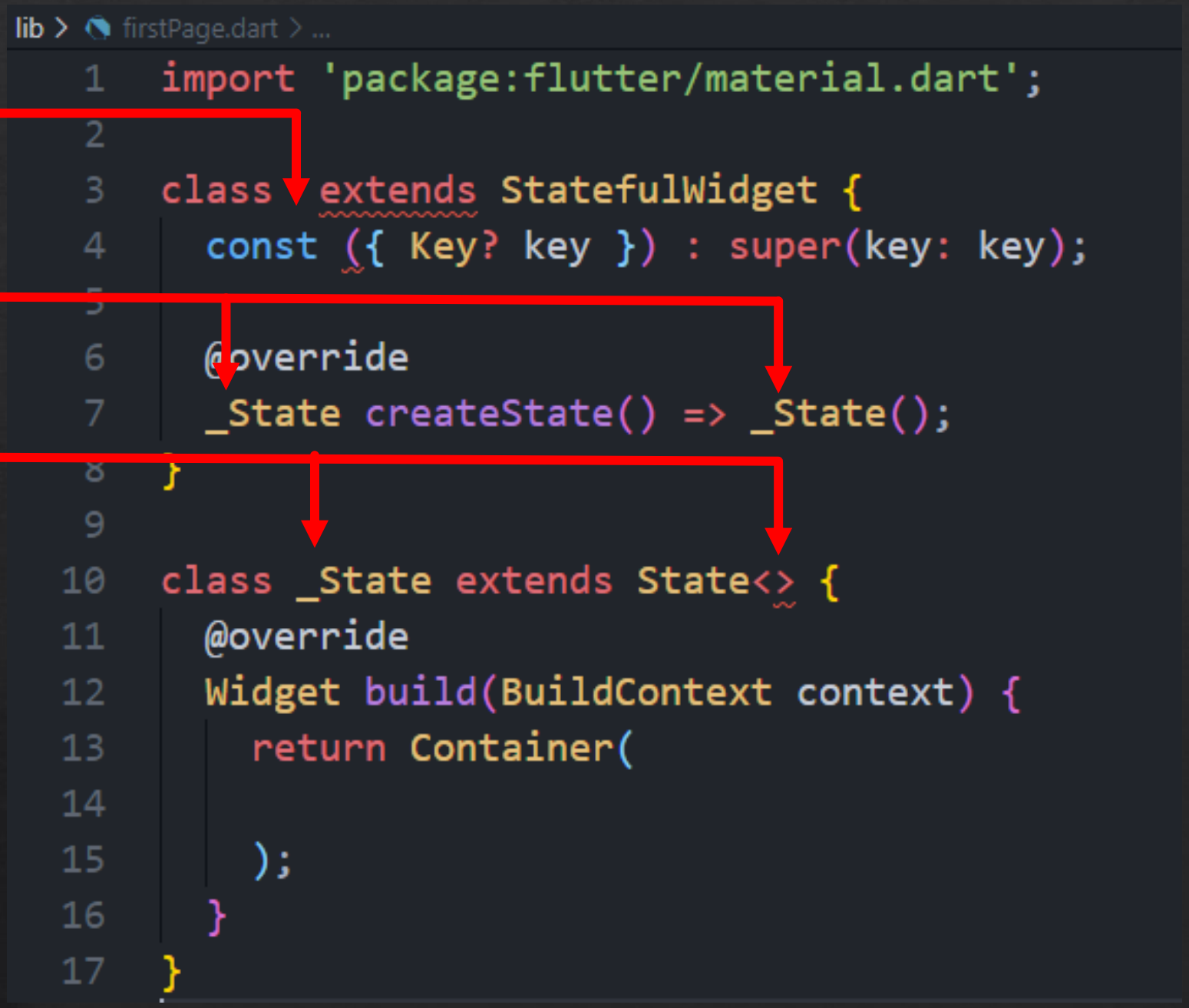
ทำหน้าที่ในการจัดการวิดเจ็ตต่างๆ คลาสกับ Stateless แต่แตกต่างกันที่ จะถูกนำมาใช้ในงานที่**สามารถตอบสนองกับผู้ใช้ได้** ทั้งรูปแบบที่ผู้ใช้สื่อสารมายังแอพพลิเคชั่น และการโต้ตอบหรือเปลี่ยนแปลงสถานะตามที่ผู้ใช้ต้องการ

ตัวช่วยในการพิมพ์คำสั่ง เรียกใช้งาน Stateful Widget



พิมพ์ stf
แล้วกด Enter

ใส่ชื่อคลาสที่จะสืบทอด
เป็น firstPage



lib > firstpage.dart > ...

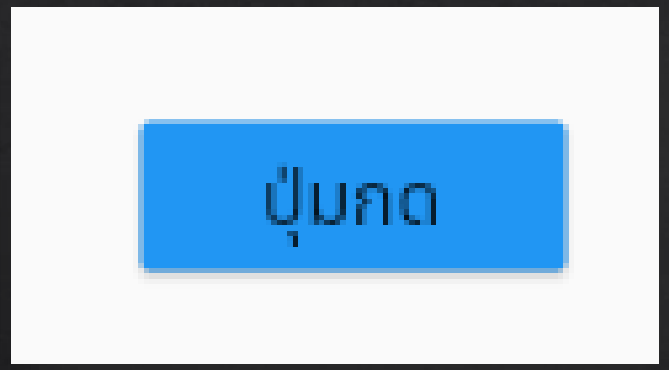
```dart
import 'package:flutter/material.dart';

class firstPage extends StatefulWidget {
  @override
  _firstPageState createState() => _firstPageState();
}

class _firstPageState extends State<firstPage> {
  @override
  Widget build(BuildContext context) {
    return Container(

    );
  }
}
```

สรุป

Stateless	Stateful
ไม่สามารถเปลี่ยนแปลงได้	สามารถเปลี่ยนแปลงได้ตลอด
การสร้างฟังก์ชัน หรือการวาดวิดเจ็ตทำครั้งเดียว	ใช้ setState() ในการทริกเพื่อสร้างฟังก์ชันหรือวาด state ใหม่

RaisedButton

เป็นวิดเจ็ตที่เป็นปุ่มกดในรูปแบบอย่างง่าย ปุ่มกดเป็นวิดเจ็ตที่ต้องรอรับการโต้ตอบกับผู้ใช้งาน

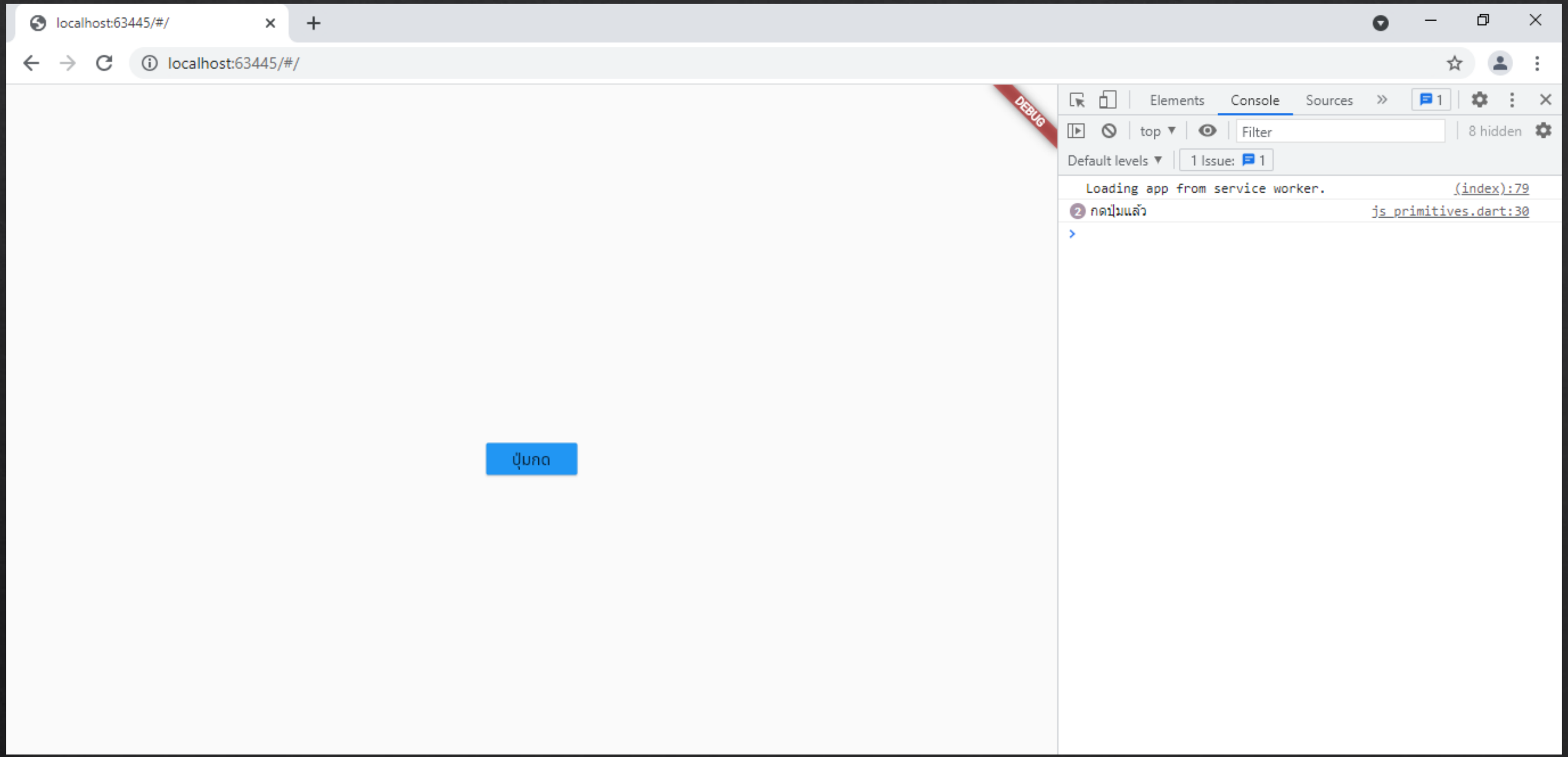


```dart
class _firstPageState extends State<firstPage> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Container(
          child: RaisedButton(
            onPressed: null,
            child: Text("ปุ่มกด"),
            color: Colors.blue,
          ),
        ),
      ),
    );
  }
}
```

สร้างฟังก์ชัน เพื่อใช้งานปุ่มกด

```
fnPrint() {
    return print("กดปุ่มแล้ว");
 }
```

```dart
class _firstPageState extends State<firstPage> {
  fnPrint() {
    return print("กดปุ่มแล้ว");
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      body: Center(
        child: Container(
          child: RaisedButton(
            onPressed: null,
            child: Text("ปุ่มกด"),
            color: Colors.blue,
          ),
        ),
      ),
    );
  }
}
```

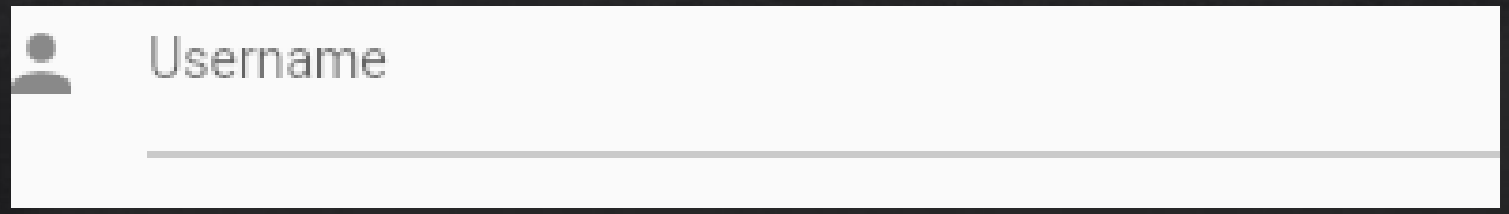

localhost:63445/#/
localhost:63445/#/
DEBUG
ปุ่มกด
Elements
Console
Sources
1
top
Filter
8 hidden
Default levels
1 Issue: 1
Loading app from service worker.
(index):79
กดปุ่มแล้ว
2
js_primitives.dart:30

ตกแต่งปุ่มด้วย Container

```
body: Center(
    child: Container(
      child: RaisedButton(
        onPressed: fnPrint,
        textColor: Colors.white,
        padding: const EdgeInsets.all(0.0),
        child: Container(
          width: 150,
          height: 40,
          decoration: BoxDecoration(
            gradient: LinearGradient(
              colors: <Color>[
                Color(0xFF0D47A1),
                Color(0xFF1976D2),
                Color(0xFF42A5F5),
              ],
            ),
          ),
          child: Center(
            child: Text('กดสี!!!'),
          ),
        ),
      ),
    ),
  ),
```

TextField

เป็นกล่องสำหรับรับข้อความที่ผู้ใช้งาน จะกรอกเพื่อใช้เป็นข้อมูลรับเข้าระบบ หรือบางครั้งสามารถใช้เพื่อการแสดงผลได้ด้วยเช่นกัน



Style

เป็นคุณสมบัติของการกำหนดค่ารูปแบบของข้อความภายใน TextField โดยค่าที่จะใช้ในการกำหนดต้องเป็นรูปแบบของ TextStyle ซึ่งอยู่ในคลาส Theme

textAlign

เป็นคุณสมบัติที่ใช้ในการจัดเรียงข้อความที่รอรับการกรอกข้อมูลจากผู้ใช้ ซึ่งมีรูปแบบและรายละเอียดของการจัดเรียงในรูปแบบของตัวอักษรตามการจัดทิศทางตามแนวนอน

ค่าคงที่ตัวเลข	รูปแบบตัวอักษร	ความหมาย
0	left	จัดแนวข้อความชิดขอบด้านซ้ายของ Container
1	right	จัดแนวข้อความชิดขอบด้านขวาของ Container
2	center	จัดแนวข้อความอยู่ที่กึ่งกลางของ Container
3	justify	กระจายข้อความเพื่อเติมให้เต็มตามความกว้างของ Container
4	start	จัดแนวข้อความจากขอบด้านหน้าของ Container
5	end	จัดแนวข้อความที่ขอบด้านท้ายของ Container

decoration

เป็นคุณสมบัติที่ช่วยตกแต่งรายละเอียด

- labelText คุณสมบัติของข้อความที่แสดงผลเพื่ออธิบายถึงข้อมูลที่ต้องกรอกใน TextField

- icon รูปภาพที่ใช้ช่วยอธิบายลักษณะของข้อมูลใน TextField

- labelStyle ใช้กำหนดรูปแบบลักษณะของ labelText

obscureText

เป็นคุณสมบัติของการกำหนดรูปแบบการแสดงผลสำหรับการกรอกข้อมูลที่ต้องการปกปิด เช่น รหัสผ่าน โดยการกำหนดค่าจะตั้งแบบบูลีน คือ ค่าจริง และเท็จ

autocorrect

เป็นคุณสมบัติในการช่วยจดจำข้อมูลที่เคยกรอกไว้ โดยการกำหนดค่าจะตั้งแบบบูลีน คือ ค่าจริง และเท็จ

autofocus

เป็นการกำหนดความพร้อมสำหรับการกรอกข้อมูลใน TextField เมื่อโปรแกรมเริ่มทำงานจะปรากฏเคอร์เซอร์เพื่อรอรับการพิมพ์ โดยการกำหนดค่าจะตั้งแบบบูลีน คือ ค่าจริง และเท็จ

enabled

เป็นการกำหนดสถานการณ์ใช้งาน TextField โดยการกำหนดค่าจะตั้งแบบบูลีน คือ ค่าจริง และ

readOnly

เป็นการกำหนดการใช้งานของ TextField คล้ายกับ enabled แต่ TextField ยังตอบสนองและรองรับการใช้งาน

keyboardType

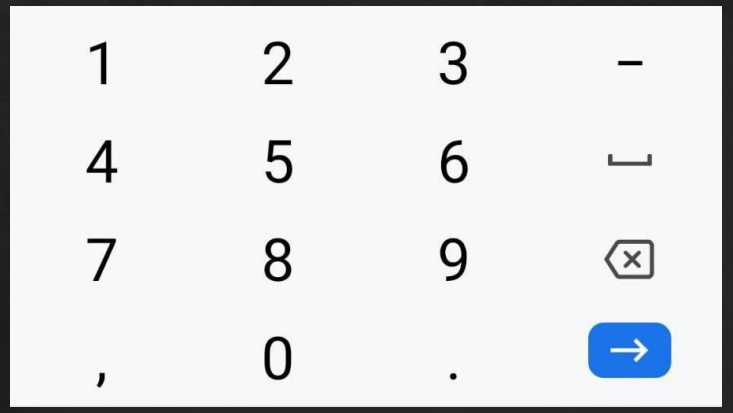
เป็นการกำหนดรูปแบบของแป้นพิมพ์ที่จะแสดงผล ที่จะแสดงผลเที่ออยู่ในสถานะที่จะกรอกข้อมูลของ TextField

รูปแบบข้อมูลแป้นพิมพ์	ความหมาย
TextInputType.datetime	แสดงตัวเลขและเครื่องหมายคั่นวันที่และเวลา
TextInputType.emailAddress	แสดงเครื่องหมาย ตัวเลข และตัวอักษรที่ใช้กับอีเมล์ได้
TextInputType.number	แสดงตัวเลข จุด เครื่องหมายทางคณิตศาสตร์
TextInputType.phone	แสดงตัวเลข สัญลักษณ์สำหรับหมายเลขโทรศัพท์
TextInputType.text	แสดงเครื่องหมาย ตัวเลข และตัวอักษรทั่วไป
TextInputType.url	แสดงเครื่องหมาย ตัวเลข และตัวอักษรที่ใช้กับเว็บ

แป้นพิมพ์อีเมล์

แป้นพิมพ์โทรศัพท์

แป้นพิมพ์ทั่วไป



แป้นพิมพ์ตัวเลข

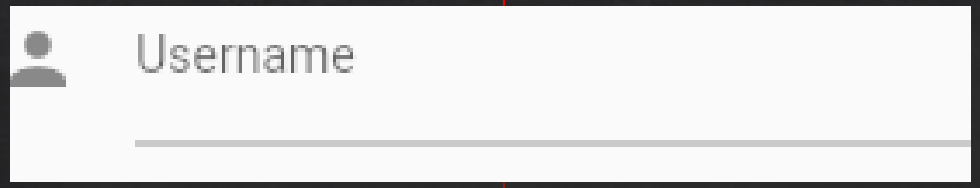
maxLenght

เป็นการกำหนดจำนวนตัวอักษรที่กรอกได้ใน TextField กำหนดการรับค่าเป็นแบบเลขจำนวนเต็ม ช่วยจำกัดจำนวนของข้อมูลที่จะรับจากผู้ใช้

controller

เป็นการกำหนดคุณสมบัติเพื่อควบคุมข้อความหรือข้อมูลที่อยู่ใน TextField

```
body: Center(
        child: Container(
          child: TextField(
            style: Theme.of(context).textTheme.bodyText1,
            textAlign: TextAlign.center,
            decoration: InputDecoration(
              labelText: 'Username',
              icon: Icon(Icons.person),
            ),
            keyboardType: TextInputType.emailAddress,
          ),
        ),
      ),
```



ระบบสมาชิก

Username

Password

Login Register

Page Navigator

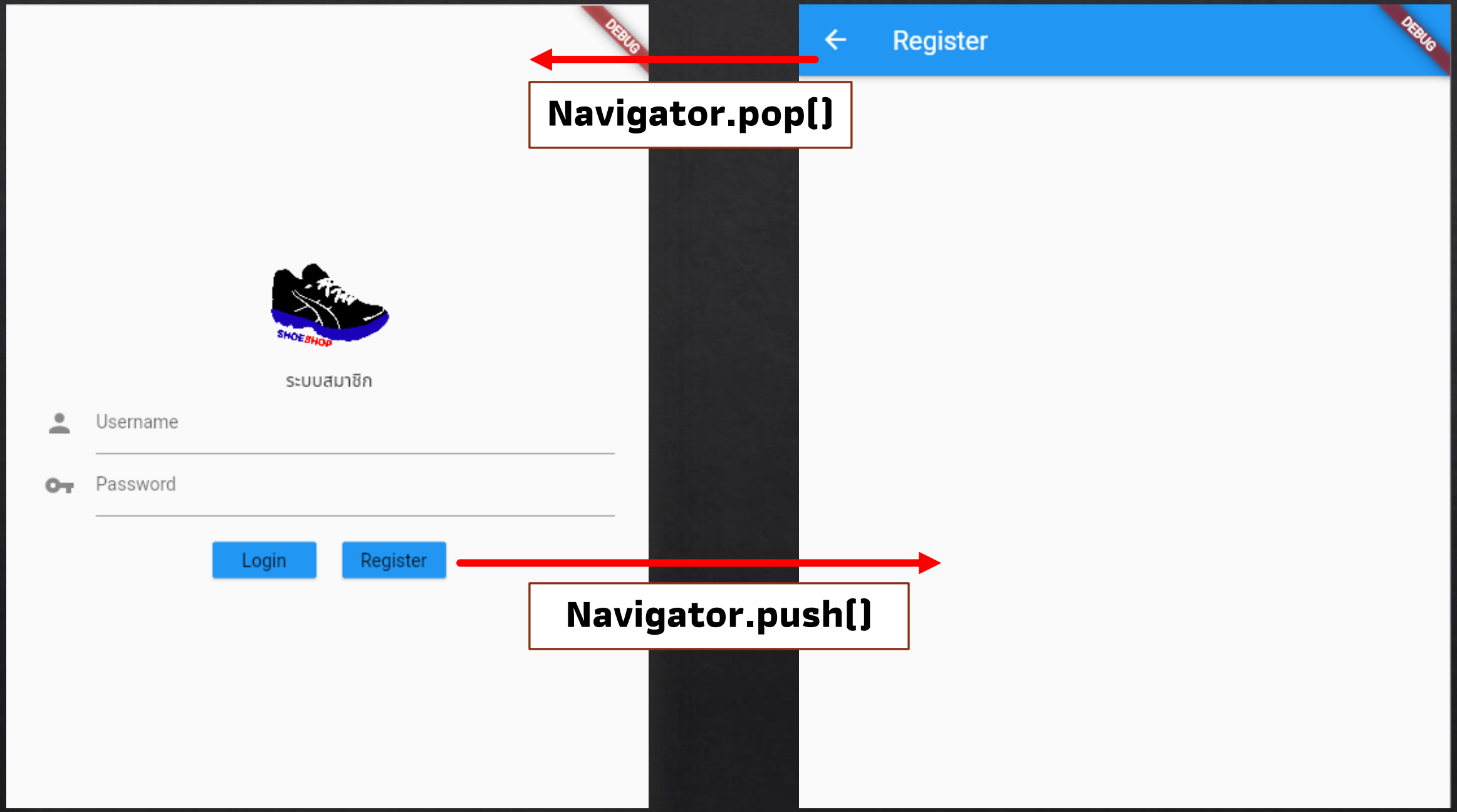
หรือ MaterialPageRoute คือ คำสั่งที่ใช้เชื่อมต่อไปยังหน้าแอพอื่นๆ

- Navigator.push
- Navigator.pop

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

class register extends StatefulWidget {
  @override
  _registerState createState() => _registerState();
}

class _registerState extends State<register> {
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('Register'),
      ), // AppBar
    ); // Scaffold
  }
}
```

เรียกใช้งานเมธอด เพื่อนำทางไปยังหน้าจออื่น

```
RaisedButton(
                    onPressed: () {
                      Navigator.push(
                          context,
                          MaterialPageRoute(
                              builder: (context) => register()));
                    },
                    child: Text('Register'),
                    color: Colors.blue,
                  ),
```


DEBUG
SHOESHOP
ระบบสมาชิก
Username
Password
Login
Register
Navigator.pop()
Register
DEBUG
Navigator.push()

Reference

- KongRuksiam Official. https://www.youtube.com/c/KongRuksiamOfficial
- เอกรินทร์ วทัญญูเลิศสกุล. (2563). พัฒนา Mobile App ด้วย Flutter & Dart. โปรวิชั่น, บจก.
- จีราวุธ วารินทร์. (2564). ต่อยอดพัฒนาโมไบล์แอพด้วย Flutter + Firebase. ซิมพลิฟาย, สนพ.