

## Import Package

```
// คำสั่งนำเข้า package
// package:flutter/material.dart = package ที่มี Widget ต่างๆ สำหรับสร้าง UI ตามแบบ Material Design
import 'package:flutter/material.dart';
```

## ฟังก์ชัน main

```
// ฟังก์ชันหลักที่เป็นจุดเริ่มต้นของโปรแกรม Dart ทุกโปรแกรม
void main() {
  // runApp() = ฟังก์ชันของ Flutter ที่ใช้เริ่มต้นแอปพลิเคชัน
  runApp(const MyApp());
}
```

## คลาส MenuItem

```
// สร้างคลาสชื่อ MenuItem สำหรับเก็บข้อมูลอาหาร
class MenuItem {
  // final = ไม่สามารถเปลี่ยนแปลงได้หลังจากกำหนดค่า
  final String nameTH; // ตัวแปรเก็บชื่ออาหารภาษาไทย
  final String nameEN; // ตัวแปรเก็บชื่ออาหารภาษาอังกฤษ
  final double price; // ตัวแปรเก็บราคาอาหาร (double = ตัวเลขทศนิยม)
  final String imageUrl; // ตัวแปรเก็บ URL ของรูปภาพ

  // Constructor แบบ named parameters
  // required = บังคับว่าต้องใส่ค่า parameter นี้
  const MenuItem({
    required this.nameTH,
    required this.nameEN,
    required this.price,
    required this.imageUrl,
  });
}
```

## คลาส MyApp

```
// สร้างคลาส MyApp ที่สืบทอดจาก StatelessWidget (Widget ที่ไม่มี state เปลี่ยนแปลง)
class MyApp extends StatelessWidget {
  // Constructor ที่รับ key จาก parent class
  const MyApp({super.key});

  // @override = บอกว่าเมธอดนี้ override จาก parent class
  @override
  // เมธอดที่ Flutter เรียกเพื่อสร้าง UI
  Widget build(BuildContext context) {
```

```

// MaterialApp = Widget หลักของแอป Flutter ที่ใช้ Material Design
// title = ชื่อของแอปที่จะแสดงใน task switcher
// home = กำหนดหน้าแรกของแอปเป็น MenuPage
return MaterialApp(title: 'เมนูอาหาร', home: const MenuPage());
}
}

```

## คลาส MenuPage และข้อมูลอาหาร

```

// สร้างคลาส MenuPage สำหรับหน้าแสดงเมนู
class MenuPage extends StatelessWidget {
  // Constructor ของ MenuPage
  const MenuPage({super.key});

  @override
  Widget build(BuildContext context) {
    // สร้าง List เก็บข้อมูล MenuItem หลายรายการ
    final List<MenuItem> menuItems = [
      // สร้าง object MenuItem แต่ละรายการ
      const MenuItem(
        nameTH: 'ผัดไทย',
        nameEN: 'Pad Thai',
        price: 60,
        imageUrl: 'https://images.unsplash.com/photo-1559314809-0d155014e29e?w=400',
      ),
    ];
  }
}

```

## Scaffold และ AppBar

```

// Scaffold = โครงสร้างพื้นฐานของหน้าจอ มี AppBar, body, floatingActionButton ฯลฯ
return Scaffold(
  // AppBar = แถบด้านบนของหน้าจอ
  appBar: AppBar(
    // title = ข้อความ title ของ AppBar
    title: const Text(
      'เมนูอาหาร',
      // TextStyle = กำหนดให้ตัวหนา ขนาด 24
      style: TextStyle(fontWeight: FontWeight.bold, fontSize: 24),
    ),
    // centerTitle = จัดข้อความให้อยู่ตรงกลาง
    centerTitle: true,
  ),
)

```

## ListView.builder

```

// ListView.builder = สร้าง list แบบ lazy loading (โหลดเฉพาะที่เห็น)
body: ListView.builder(

```

```
// padding = เว้นระยะห่างรอบๆ 16 pixels
padding: const EdgeInsets.all(16),
// itemCount = จำนวน item = จำนวนรายการอาหาร (5 รายการ)
itemCount: menuItems.length,
// itemBuilder = ฟังก์ชันที่สร้าง Widget สำหรับแต่ละ item
itemBuilder: (context, index) {
  // ดึงข้อมูลรายการอาหารตาม index
  final item = menuItems[index];
```

---

## Card Widget

```
// Card = Widget ที่มีเงาและขอบมน
return Card(
  // margin = เว้นระยะด้านล่าง 16 pixels
  margin: const EdgeInsets.only(bottom: 16),
  // elevation = ระดับความสูงของเงา
  elevation: 4,
  // RoundedRectangleBorder = กำหนดรูปร่างให้เป็นสี่เหลี่ยมขอบมน
  shape: RoundedRectangleBorder(
    // BorderRadius.circular(16) = มุมโค้ง 16 pixels
    borderRadius: BorderRadius.circular(16),
  ),
  // ClipRRect = ตัดขอบ Widget ลูกให้โค้งมน
  child: ClipRRect(
    borderRadius: BorderRadius.circular(16),
```

---

## Column

```
// Column = เรียง Widget ลูกในแนวตั้ง
child: Column(
  // crossAxisAlignment = จัดให้ชิดซ้าย (แกนขวาง)
  crossAxisAlignment: CrossAxisAlignment.start,
  // children = รายการ Widget ลูกที่จะแสดง
  children: [
```

---

## Image.network

```
// Image.network = โหลดรูปภาพจาก URL
Image.network(
  // item.imageUrl = URL ของรูปภาพจากข้อมูล MenuItem
  item.imageUrl,
  // height = ความสูง 200 pixels
  height: 200,
  // width = กว้างเต็มที่
  width: double.infinity,
  // fit = ปรับขนาดรูปให้เต็มพื้นที่โดยครอบส่วนเกิน
  fit: BoxFit.cover,
```

```

// loadingBuilder = ฟังก์ชันที่ทำงานขณะโหลดรูป
loadingBuilder: (context, child, loadingProgress) {
  // ถ้าโหลดเสร็จแล้วให้แสดงรูป (child)
  if (loadingProgress == null) return child;
  // SizedBox = กล่องขนาด 200 pixels
  return SizedBox(
    height: 200,
    // CircularProgressIndicator = แสดงวงกลมหมุนขณะโหลด
    child: const Center(child: CircularProgressIndicator()),
  );
},
// errorBuilder = ฟังก์ชันที่ทำงานเมื่อโหลดรูปไม่สำเร็จ
errorBuilder: (context, error, stackTrace) {
  return SizedBox(
    height: 200,
    child: const Center(
      // Icons.restaurant = แสดงไอคอนร้านอาหารแทนรูปที่โหลดไม่ได้
      child: Icon(Icons.restaurant, size: 64),
    ),
  );
},
),

```

## ข้อมูลชื่ออาหาร

```

// Padding = Widget สำหรับเว้นระยะห่าง
Padding(
  // EdgeInsets.all(16) = เว้นระยะรอบด้าน 16 pixels
  padding: const EdgeInsets.all(16),
  child: Column(
    crossAxisAlignment: CrossAxisAlignment.start,
    children: [
      // Text = แสดงชื่ออาหารภาษาไทย
      Text(
        item.nameTH,
        style: const TextStyle(
          fontSize: 22, // ขนาดตัวอักษร 22
          fontWeight: FontWeight.bold, // ตัวหนา
        ),
      ),
      // SizedBox = ช่องว่างแนวดิ่ง 4 pixels
      const SizedBox(height: 4),
      // Text = แสดงชื่ออาหารภาษาอังกฤษ
      Text(
        item.nameEN,
        style: TextStyle(
          fontSize: 16,
          fontStyle: FontStyle.italic, // ตัวเอียง
        ),
      ),
    ],
  ),

```

```
// SizedBox = ช่องว่างแนวดิ่ง 8 pixels
const SizedBox(height: 8),
```

---

## แสดงราคา

```
// Container = กล่องสำหรับแสดงราคา
Container(
  // EdgeInsets.symmetric = padding แนวนอน 12, แนวดิ่ง 6
  padding: const EdgeInsets.symmetric(
    horizontal: 12,
    vertical: 6,
  ),
  // BoxDecoration = กำหนดรูปแบบของ Container
  decoration: BoxDecoration(
    // BorderRadius.circular(20) = มุมโค้ง 20 pixels (ทำให้เป็นแคปซูล)
    borderRadius: BorderRadius.circular(20),
  ),
  child: Text(
    // '฿${item.price.toStringAsFixed(0)}' = แสดงราคาแบบไม่มีทศนิยม เช่น ฿60
    // toStringAsFixed(0) = แปลงเป็น String โดยไม่มีทศนิยม
    '฿${item.price.toStringAsFixed(0)}',
    style: const TextStyle(
      fontSize: 18,
      fontWeight: FontWeight.bold,
    ),
  ),
),
```