

Material Design 3

ใช้ Material Design Widget ใน Flutter อย่างเข้าใจ
และประยุกต์ใช้งานได้

อนุชิต ชโลธ

สำนักพิมพ์ท้อปวง

Material Design 3

ใช้ Material Design Widget ใน Flutter อย่างเข้าใจและ
ประยุกต์ใช้งานได้

พิมพ์ครั้งที่ 2

อนุชิต ชโลธร

สำนักพิมพ์ก้อปวาง

บทนำ

หนังสือเล่มนี้เป็นคู่มือการใช้งานวิดเจ็ต (Widget) ของ Material Design 3 โดยเน้นภาพประกอบตัวอย่างโค้ด โดยแบ่ง Widget ออกเป็น 6 ประเภท ดังนี้

1. Widget ประเภทแอคชัน เช่น ปุ่มในรูปแบบต่างๆ
2. Widget ที่ใช้สื่อสารกับผู้ใช้ เช่น Badge, Snackbar เป็นต้น
3. Widget ครอบวิดเจ็ตอื่น เช่น Bottom Sheet, Card, Dialog เป็นต้น
4. Widget การนำทาง เช่น Navigation Drawer, NavigationBar, Tabs เป็นต้น
5. Widget ตัวเลือก เช่น Options, Slider, Switch เป็นต้น
6. Widget กล่องข้อความ เช่น Text Field เป็นต้น

คุณสามารถดูภาพตัวอย่างและตัวอย่างโค้ดเพื่อทำความเข้าใจการใช้งาน Widget ต่างๆ ได้

สารบัญ

บทนำ	1
แนะนำ Material Design	5
Material Design	5
Flutter	5
Material 3	6
Typography	12
Color Scheme	16
Elevation	24
วิดเจ็ตประเภทแอคชั่น	29
Buttons	29
Buttons with icon	31
Disabled button	33
Floating Action Button	35
Icon button	36
Segmented button	41
วิดเจ็ตใช้สื่อสารกับผู้ใช้	44
Badges	44
Progress indicator	48
Snackbar	50
วิดเจ็ตครอบวิดเจ็ตอื่น	52
Modal Bottom Sheet	52
Bottom Sheet	57
Cards	61
Dialog	64
Full-screen dialog	66
Divider	68
วิดเจ็ตการนำทาง	70
Bottom App Bar	70

Navigation Bar	73
Navigation Drawer	75
Navigation Rail	79
Tabs	83
Secondary Tab Bar	86
Search Bar	89
Small App Bar	94
Center App Bar	96
Medium App Bar	98
Large App Bar	100
Checkboxes	102
Chips	105
Menus	108
วิดเจ็ตตัวเลือก	114
Options	114
Sliders	117
Switches	120
วิดเจ็ตเลือกวันที่และเวลา	123
Date Picker	123
Time Picker	124
วิดเจ็ตกล่องข้อความ	126
Text fields	126
ดาวนโหลดซอร์สโค้ด	132
แนะนำหนังสือชุด “ก๊อปปวาง”	134
สูตรลัด Flutter	134
สูตรลัด GetX	135
สูตรลัด FlutterFire	136
สูตรลัด Hasura	137
คู่มือประกอบการเรียน Flutter Crash Course	138
Material Design 3	139
สูตรลัด Dart 3	140

ສູຕຣລັດ Dart Frog	142
ສູຕຣລັດ Appwrite	144
ສູຕຣລັດ Serverpod	146
Prisma Client Dart	148

แนะนำ Material Design

Material Design

Material Design เป็นระบบออกแบบที่ถูกสร้างและรองรับโดยนักออกแบบและนักพัฒนาของ Google ซึ่งประกอบด้วยคำแนะนำ User Experience ลึกซึ้งและการนำไปใช้งาน User Interface สำหรับ Android, Flutter และเว็บไซต์

เวอร์ชันล่าสุดคือ Material 3 ที่มีการปรับปรุงให้สามารถปรับเปลี่ยน Personal User Experience ตามความต้องการของผู้ใช้ได้ และเป็นการออกแบบสื่อสารอย่างชัดเจน ตั้งแต่การใช้สีแบบไดนามิก การเข้าถึงที่ง่ายขึ้นไปจนถึงการสร้างพื้นฐานสำหรับการออกแบบหน้าจอขนาดใหญ่

Flutter

Flutter เป็นเฟรมเวิร์กที่ช่วยให้นักพัฒนาสามารถสร้างแอปพลิเคชันที่สวยงามและถูกคอมไพล์เป็นแอปพลิเคชัน Native ได้จากโค้ดเดียวกัน Flutter สนับสนุน Material 3 ที่เป็นระบบออกแบบล่าสุดของ Google ซึ่งทำให้ผู้ใช้สามารถมีประสบการณ์ส่วนบุคคลที่ปรับได้และมีความคล่องตัวในการใช้งาน

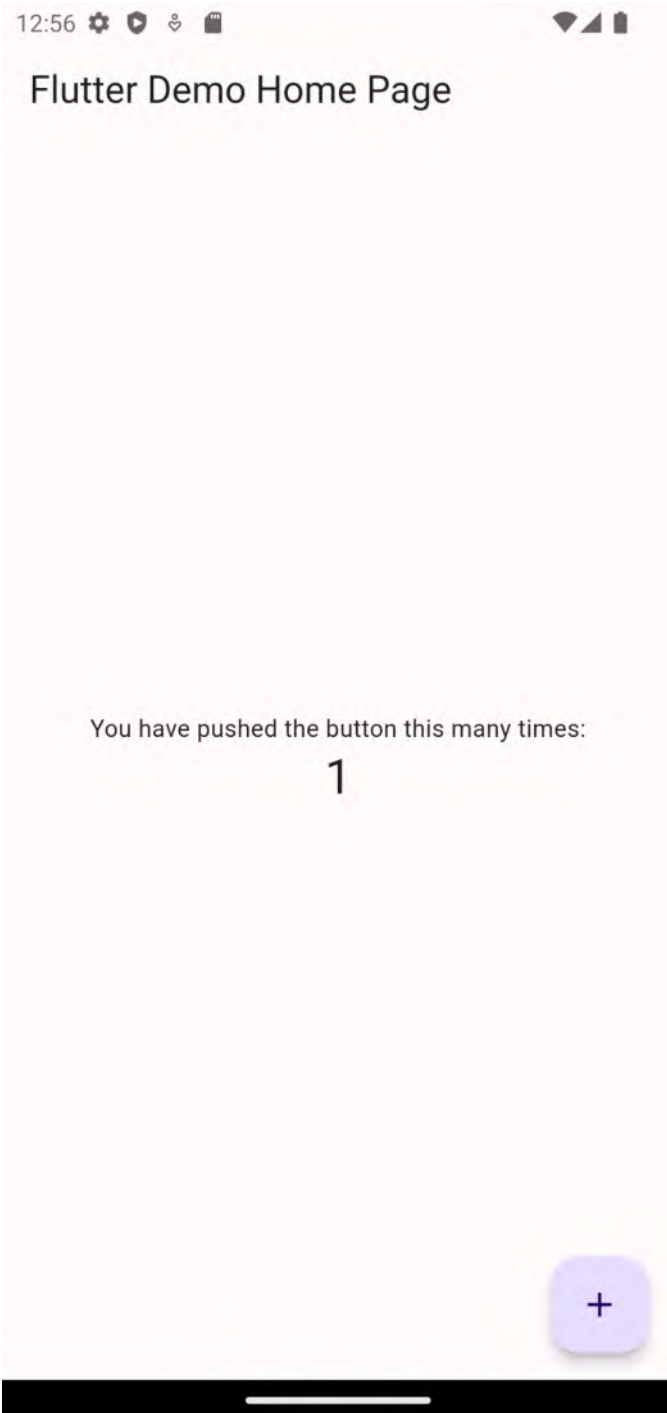
หนังสือเล่มนี้จัดเรียง Widget ตามกลุ่ม ดังนี้ แอคชั่น, สื่อสารกับผู้ใช้, ครอบคลุม, การนำทาง, ตัวเลือก และ กล่องข้อความ เพื่อช่วยให้นักพัฒนาสามารถนำเอา Widget ต่างๆ ของ Material Design มาประยุกต์ใช้ใน

Flutter ได้อย่างมีประสิทธิภาพ โดยคุณสามารถเรียนรู้การใช้งาน Widget ด้วยการอ่านโค้ดและดูภาพตัวอย่าง

นอกจากนี้ คุณยังสามารถสร้าง User Interface ที่สวยงาม และเป็นมาตรฐานด้วยการนำ Material Design มาประยุกต์ใช้ในแอปพลิเคชันของคุณได้ โดยการอ้างอิงในหนังสือเล่มนี้จะช่วยให้คุณสามารถใช้ Material Design ในการสร้าง User Experience ที่น่าสนใจในแอปพลิเคชันของคุณได้โดยง่ายและสะดวกยิ่งขึ้น

Material 3

Material Design จะถูกเรียกใช้งานผ่าน Widget `MaterialApp()` คุณสามารถกำหนดค่า `useMaterial3` เป็น `true` ใน `ThemeData()` เพื่อเรียกใช้ Material 3 เมื่อคุณกำหนดค่าให้เรียกใช้ Material 3 แล้ว รูปแบบ ระบบสี จะถูกเรียกใช้โดยอัตโนมัติ

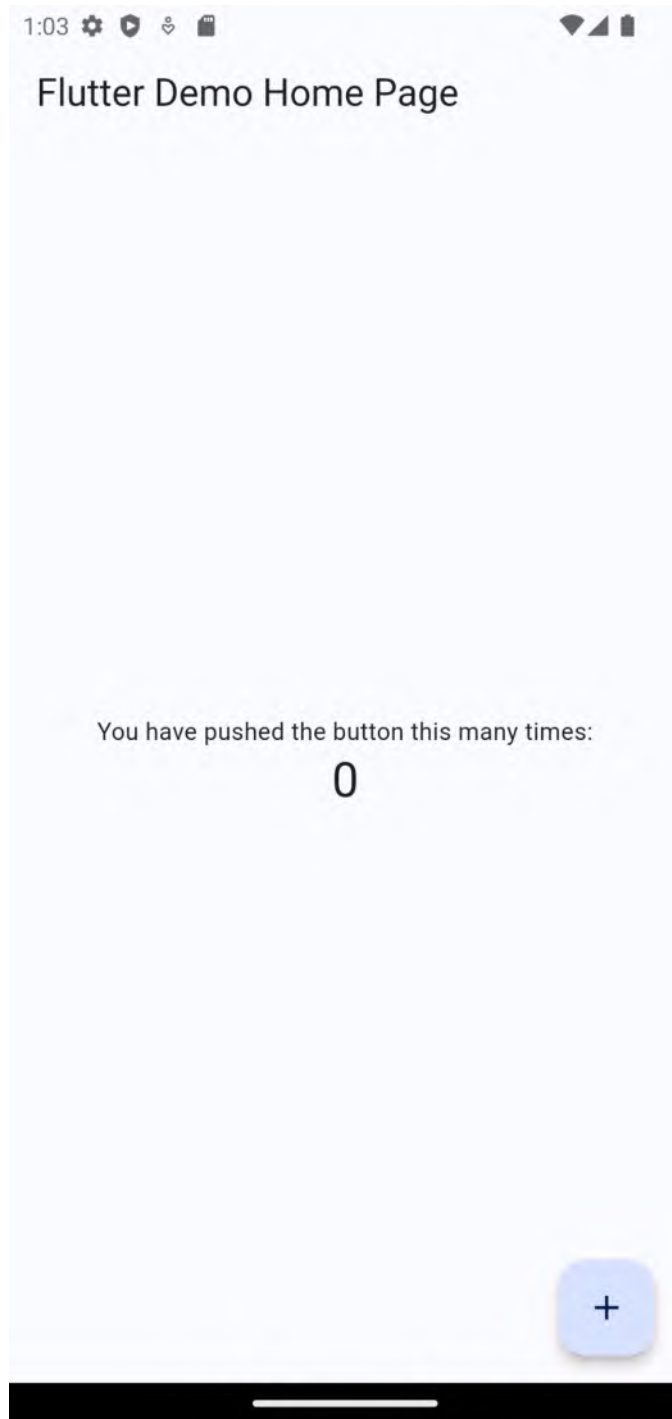


Material 3

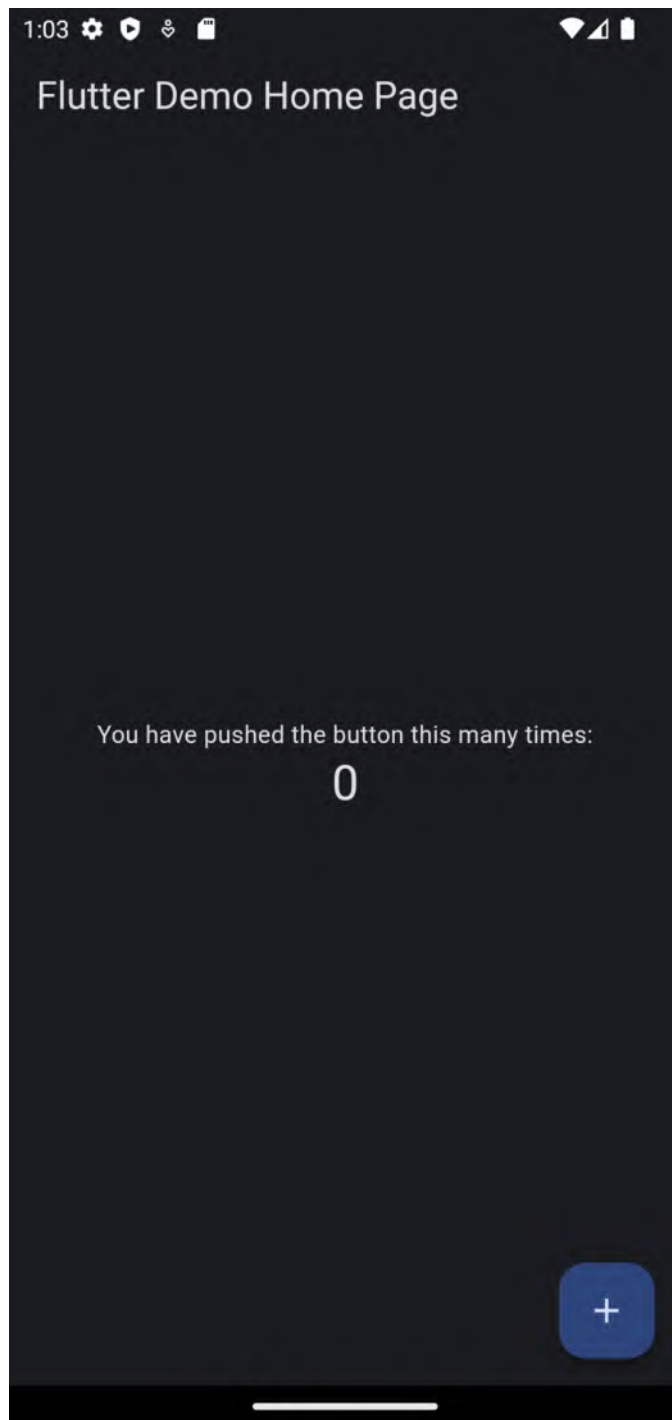
ตัวอย่างโค้ด

```
class MyApp extends StatelessWidget {  
  const MyApp({super.key});  
  
  @override  
  Widget build(BuildContext context) {  
    return MaterialApp(  
      debugShowCheckedModeBanner: false,  
      title: 'Flutter Demo',  
      theme: ThemeData(  
        useMaterial3: true,  
      ),  
      home: const MyHomePage(title: 'Flutter Demo Home Page'),  
    );  
  }  
}
```

เพื่อให้รองรับ Material You คุณสามารถใช้ปลั๊กอิน `dynamic_color` เพื่อเปลี่ยนสี Theme ของ App ตามสีของ Wallpaper ดังนี้



Light Theme



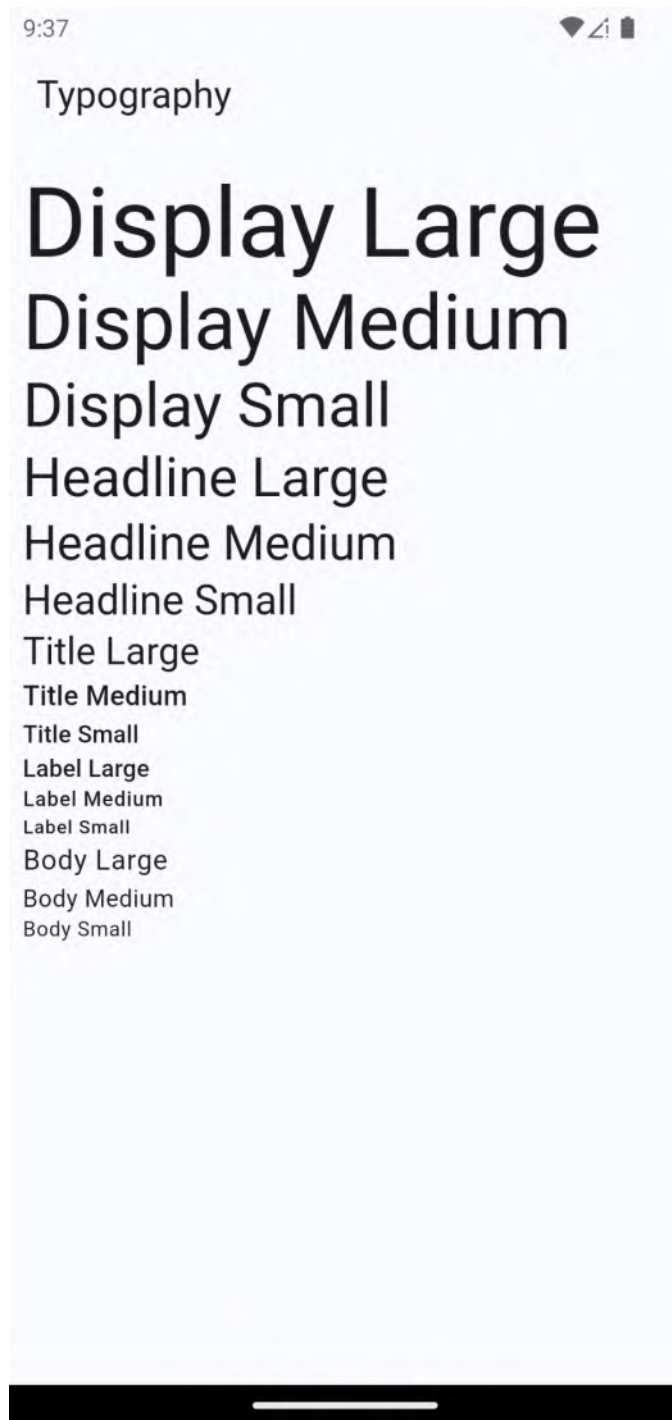
Dark Theme

ตัวอย่างโค้ด

```
class MyApp extends StatelessWidget {
  const MyApp({super.key});

  @override
  Widget build(BuildContext context) {
    return DynamicColorBuilder(builder: (lightDynamic, darkDynamic)
{
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'Flutter Demo',
      theme: ThemeData(
        useMaterial3: true,
        colorScheme: lightDynamic,
      ),
      darkTheme: ThemeData(
        useMaterial3: true,
        colorScheme: darkDynamic,
      ),
      home: const MyHomePage(title: 'Flutter Demo Home Page'),
    );
  });
}
```

Typography



ตัวอย่างโค้ด

```
class TypographyPage extends StatefulWidget {  
  const TypographyPage({super.key});  
  
  @override  
  State<TypographyPage> createState() => _TypographyPageState();  
}
```

```
class _TypographyPageState extends State<TypographyPage> {  
  @override  
  Widget build(BuildContext context) {  
    final textTheme = Theme.of(context).textTheme.apply(  
      displayColor: Theme.of(context).colorScheme.onSurface,  
    );  
  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Typography'),  
      ),  
      body: ListView(  
        padding: const EdgeInsets.all(8.0),  
        children: <Widget>[  
          const SizedBox(height: 7),  
          Text(  
            'Display Large',  
            style: textTheme.displayLarge!,  
          ),  
          Text(  
            'Display Medium',  
            style: textTheme.displayMedium!,  
          ),  
          Text(  
            'Display Small',  
            style: textTheme.displaySmall!,  
          ),  
          Text(  
            'Headline Large',  
            style: textTheme.headlineLarge!,  
          ),  
          Text(  

```