

## ใบงานการทดลองที่ 5

### เรื่อง Flutter และ Widgets

#### 1. จุดประสงค์

- 1 สามารถใช้ Text Widget ได้
- 2 สามารถใช้ Image Widget ได้
- 3 สามารถใช้ Row Widget ได้
- 4 สามารถใช้ Column Widget ได้

#### 2. ทฤษฎี

#### บทที่ 3.1 Widgets พื้นฐาน

Widgets คือองค์ประกอบหลักที่ใช้ในการสร้าง UI ใน Flutter. มันเป็นสิ่งที่อยู่ในภายในต้นฉบับของ Flutter ทุกอย่าง, ทำให้มันมีความสามารถที่หลากหลายและยืดหยุ่น

Flutter มี Widgets พื้นฐานมากมาย Widgets พื้นฐาน โดยหลัก ๆ แล้วจะมี 4 อันดังนี้

##### 3.1.1 Text Widget

Text Widget เป็น Widget พื้นฐานใน Flutter ใช้ในการแสดงข้อความบนหน้าจอ Widget นี้ไม่มีสถานะของตัวเอง สถานะของ Text Widget จะถูกกำหนดโดยข้อมูลภายนอก เช่น ข้อมูลในตัวแปรหรือข้อมูลในฐานะข้อมูล

Text Widget สามารถสร้างข้อความได้หลากหลายรูปแบบ ตัวอย่างเช่น สามารถสร้างข้อความแบบธรรมดา ข้อความแบบอักษรพิเศษ ข้อความแบบย่อหน้า หรือข้อความแบบโค้ด

ตัวอย่างการใช้ Text Widget

```
Text('Hello, World!');
```

โค้ดข้างต้นจะแสดงข้อความ "Hello, Flutter!" บนหน้าจอ

คุณสมบัติของ Text Widget

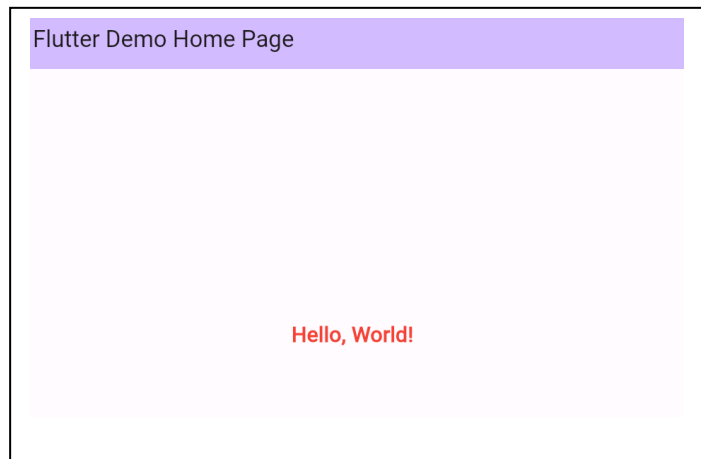
Text Widget มีคุณสมบัติหลายอย่างที่สามารถกำหนดค่าได้ คุณสมบัติเหล่านี้ได้แก่

- text: ข้อความที่จะแสดง
- style: สไตล์ของข้อความ เช่น สี ขนาด รูปแบบตัวอักษร
- textAlign: การจัดตำแหน่งของข้อความ

ตัวอย่างการกำหนดค่าคุณสมบัติของ Text Widget

```
Text(
  'Hello, World!',
  style: TextStyle(
    color: Colors.red,
    fontSize: 20,
    fontWeight: FontWeight.bold,
  ),
  textAlign: TextAlign.center,
);
```

ผลลัพธ์



โค้ดข้างต้นจะแสดงข้อความ "Hello,World!" บนหน้าจอด้วยสีสีแดง ขนาด 20 พิกเซล ตัวหนา การจัดตำแหน่งแบบกึ่งกลาง

### 3.1.2 Image Widget

Image Widget เป็น Widget พื้นฐานใน Flutter ใช้ในการแสดงรูปภาพบนหน้าจอ Widget นี้ไม่มีสถานะของตัวเอง สถานะของ Image Widget จะถูกกำหนดโดยข้อมูลภายนอก เช่น ข้อมูลในตัวแปรหรือข้อมูลพื้นฐานข้อมูล

Image Widget สามารถแสดงรูปภาพได้หลากหลายรูปแบบ ตัวอย่างเช่น สามารถใช้รูปภาพจากไฟล์ รูปภาพจากอินเทอร์เน็ต หรือรูปภาพแบบเคลื่อนไหว

ตัวอย่างการใช้ Image Widget

#### 1. อัปโหลดไฟล์จากในเครื่อง

```
Image.asset('assets/images/my_image.jpg');
```

โค้ดข้างต้นจะแสดงรูปภาพจากไฟล์ my\_image.jpg ที่เก็บอยู่ในโฟลเดอร์ assets ของโปรเจกต์

#### 2. อัปโหลดไฟล์จากอินเทอร์เน็ต

```
const Image(
  image: NetworkImage('https://flutter.github.io/assets-for-api-docs/assets/widgets/owl.jpg'),
)
```

หรือ

```
Image.network('https://flutter.github.io/assets-for-api-docs/assets/widgets/owl.jpg')
```

### คุณสมบัติของ Image Widget

Image Widget มีคุณสมบัติหลายอย่างที่สามารถกำหนดค่าได้ คุณสมบัติเหล่านี้ได้แก่

- image: รูปภาพที่จะใช้
- width: ความกว้างของรูปภาพ
- height: ความสูงของรูปภาพ
- fit: วิธีการปรับขนาดรูปภาพ
- alignment: การจัดตำแหน่งของรูปภาพ

ตัวอย่างการกำหนดค่าคุณสมบัติของ Image Widget

```
Image.asset(
  'assets/images/my_image.jpg',
  width: 100,
  height: 100,
  fit: BoxFit.cover,
  alignment: Alignment.center,
);
```

โค้ดข้างต้นจะแสดงรูปภาพจากไฟล์ my\_image.jpg ที่เก็บอยู่ในโฟลเดอร์ assets ของโปรเจกต์ รูปภาพจะมีขนาด 100x100 พิกเซล รูปภาพจะปรับขนาดให้พอดีกับพื้นที่ที่กำหนด รูปภาพจะจัดตำแหน่งแบบกึ่งกลาง

### 3.1.3 Row Widget

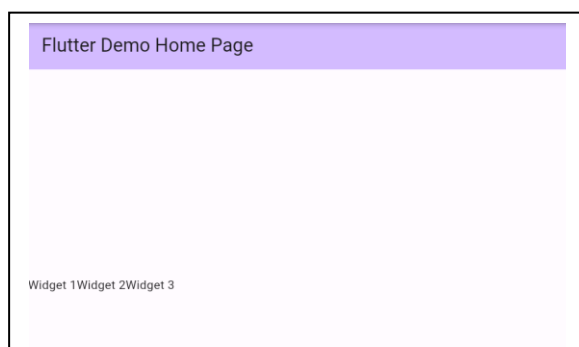
Row Widget เป็น Widget พื้นฐานใน Flutter ใช้ในการจัดเรียง Widgets แนวนอน Widget นี้ไม่มีสถานะของตัวเอง สถานะของ Row Widget จะถูกกำหนดโดยข้อมูลภายนอก เช่น ข้อมูลในตัวแปรหรือข้อมูลในฐานะข้อมูล

Row Widget สามารถจัดเรียง Widgets ได้อย่างหลากหลายรูปแบบ ตัวอย่างเช่น สามารถจัดเรียง Widgets แบบเรียงจากซ้ายไปขวา แบบเรียงจากขวาไปซ้าย แบบเรียงจากบนลงล่าง หรือแบบเรียงจากล่างขึ้นบน

ตัวอย่างการใช้ Row Widget

```
Row(
  children: [
    Text('Widget 1'),
    Text('Widget 2'),
    Text('Widget 3'),
  ],
);
```

ผลลัพธ์



โค้ดข้างต้นจะจัดเรียง Widgets สามตัวแบบเรียงจากซ้ายไปขวา

คุณสมบัติของ Row Widget

Row Widget มีคุณสมบัติหลายอย่างที่สามารถกำหนดค่าได้ คุณสมบัติเหล่านี้ได้แก่

- children: Widgets ที่จะจัดเรียง
- mainAxisAlignment: การจัดตำแหน่งหลักของ Widgets ในแนวนอน
- crossAxisAlignment: การจัดตำแหน่งรองของ Widgets ในแนวตั้ง

จากรูปที่ 3.1.3.1 จะแสดงให้เห็นว่าการเรียงข้อมูล Row ในแบบต่างแตกต่างกันอย่างไร



รูปที่ 3.1.3.1 การจัดเรียงข้อมูลใน Row

ตัวอย่างการกำหนดค่าคุณสมบัติของ Row Widget

```
Row(
  children: [
    Text("Widget 1"),
    Text("Widget 2"),
    Text("Widget 3"),
  ],
  mainAxisAlignment: MainAxisAlignment.center,
  crossAxisAlignment: CrossAxisAlignment.start,
);
```

ผลลัพธ์




โค้ดข้างต้นจะจัดเรียง Widgets สามตัวแบบเรียงจากซ้ายไปขวา Widgets แต่ละตัวจะจัดตำแหน่งแบบกึ่งกลางในแนวนอนและแนวตั้ง Widgets จะมีขนาดเท่ากับเนื้อหาที่แสดง

### 3.1.4 Column Widget

Column Widget เป็น Widget พื้นฐานใน Flutter ใช้ในการจัดเรียง Widgets แนวตั้ง Widget นี้ไม่มีสถานะของตัวเอง

Column Widget จะถูกกำหนดโดยข้อมูลภายนอก เช่น ข้อมูลในตัวแปรหรือข้อมูลในฐานะข้อมูล Column Widget สามารถจัดเรียง Widgets ได้อย่างหลากหลายรูปแบบ ตัวอย่างเช่น สามารถจัดเรียง Widgets แบบเรียงจากบนลงล่าง แบบเรียงจากล่างขึ้นบน แบบเรียงจากซ้ายไปขวา หรือแบบเรียงจากขวาไปซ้าย

ตัวอย่างการใช้ Column Widget

Column( children: [ Text('Widget 1'), Text('Widget 2'), Text('Widget 3'), ], );	ผลลัพธ์
	

โค้ดข้างต้นจะจัดเรียง Widgets สามตัวแบบเรียงจากบนลงล่าง

คุณสมบัติของ Column Widget

Column Widget มีคุณสมบัติหลายอย่างที่สามารถกำหนดค่าได้ คุณสมบัติเหล่านี้ได้แก่

- children: Widgets ที่จะจัดเรียง
- mainAxisAlignment: การจัดตำแหน่งหลักของ Widgets ในแนวตั้ง
- crossAxisAlignment: การจัดตำแหน่งรองของ Widgets ในแนวนอน

จากรูปที่ 3.2.2.2 จะแสดงให้เห็นว่าการเรียงข้อมูล Column ในแบบต่างแตกต่างกันอย่างไร

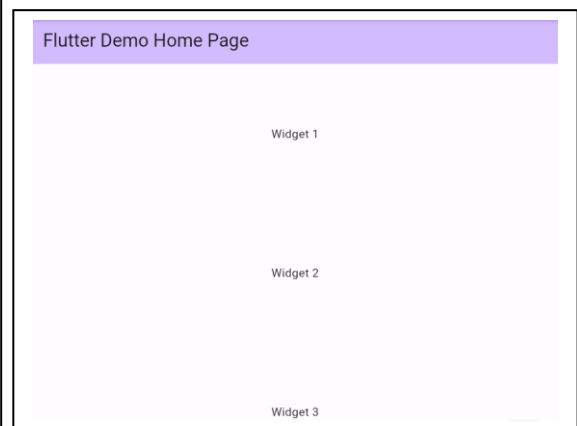


รูปที่ 3.1.4.1 การจัดเรียงข้อมูลใน Column

ตัวอย่างการกำหนดค่าคุณสมบัติของ Column Widget

```
Column(
  children: [
    Text('Widget 1'),
    Text('Widget 2'),
    Text('Widget 3'),
  ],
  mainAxisAlignment: MainAxisAlignment.spaceAround,
  crossAxisAlignment: CrossAxisAlignment.center,
),
```

ผลลัพธ์



### 3. เครื่องมือและอุปกรณ์การทดลอง

- 3.1 คอมพิวเตอร์ 1 เครื่อง
- 3.2 ใบงานที่ 1 เรื่อง คำสั่งเลือกเงื่อนไข

### 4. ลำดับขั้นการทดลอง

- 4.1 นักเรียนศึกษาเนื้อหา เรื่อง คำสั่งเลือกเงื่อนไข
- 4.2 ให้นักเรียนตอบคำถาม ลงในใบงานที่ 1
- 4.3 ส่งงานครูหลังจากเสร็จเรียบร้อยแล้ว

คำสั่ง ให้นักศึกษาเขียนคำตอบตามที่โจทย์กำหนดให้ถูกต้อง (สามารถแนบรูปโค้ดและผลลัพธ์คำตอบของโปรแกรมได้)

1. จัดทำ Resume โดยใช้ Widgets ดังต่อไปนี้ และใช้ คุณสมบัติของ Widgets นั้น อย่างน้อย 1 รายการ พร้อมผลลัพธ์ที่แสดงผลบนหน้าจอมือถือที่เชื่อม VSCode ไว้

Widgets ที่ต้องใช้ 1. text Widget 2. Image Widget 3. Row Widget 4. Column Widget

```

1  import 'package:flutter/material.dart';
2
3  Run | Debug | Profile
4  void main() {
5    runApp(const MyApp());
6  }
7
8  class MyApp extends StatelessWidget {
9    const MyApp({super.key});
10
11    @override
12    Widget build(BuildContext context) {
13      return MaterialApp(
14        title: 'Resume',
15        theme: ThemeData(
16          colorScheme: ColorScheme.fromSeed(seedColor: Colors.green),
17          useMaterial3: true,
18        ), // ThemeData
19        home: const MyHomePage(title: 'Resume'),
20      ); // MaterialApp
21    }
22  }
23
24  class MyHomePage extends StatefulWidget {
25    const MyHomePage({super.key, required this.title});
26    final String title;
27
28    @override
29    State<MyHomePage> createState() => _MyHomePageState();
30  }
31
32  class _MyHomePageState extends State<MyHomePage> {
33    @override
34    Widget build(BuildContext context) {
35      return Scaffold(
36        appBar: AppBar(
37          backgroundColor: Theme.of(context).colorScheme.inversePrimary,
38          title: Text(widget.title),
39        ), // AppBar
40        body: Center(
41          child: Row(
42            mainAxisAlignment: MainAxisAlignment.center,
43            crossAxisAlignment: CrossAxisAlignment.start,
44            children: <Widget>[

```



```

44         Image.asset(
45             'images/pic.png',
46             width: 150,
47             height: 200,
48             fit: BoxFit.cover,
49         ), // Image.asset
50         const Textname(
51             Textn: 'Athit Suntalodom',
52             Textl: 'ลลิต์ สุนทโรดม',
53             Texttr: 'อายุ : 22',
54             Textb: 'ชื่อเล่น : ต๋ม',
55             Textbday: 'วันเดือนปีเกิด: 17 พฤษภาคม 2544',
56             Textfb: 'FB : athit_suntalodom',
57             Textgm: 'GM : tooy0303@gmail.com',
58             Texthm: 'HM : athit_su65@live.rmutil.ac.th',
59         ), // Textname
60     ], // <Widget>[]
61 ), // Row
62 ), // Center
63 ); // Scaffold
64 }
65 }
66
67 class Textname extends StatelessWidget {
68     const Textname({
69         super.key,
70         required this.Textn,
71         required this.Textl,
72         required this.Texttr,
73         required this.Textb,
74         required this.Textbday,
75         required this.Textfb,
76         required this.Textgm,
77         required this.Texthm});
78     final String Textn;
79     final String Textl;
80     final String Texttr;
81     final String Textb;
82     final String Textbday;
83     final String Textfb;
84     final String Textgm;
85     final String Texthm;
86
87     @override
88     Widget build(BuildContext context) {
89         return Padding(
90             padding: const EdgeInsets.only(left: 10, top: 10),
91             child: Column(crossAxisAlignment: CrossAxisAlignment.start, children: [
92                 Text(
93                     Textn,
94                     style: const TextStyle(fontSize: 25, fontWeight: FontWeight.bold),
95                 ), // Text
96                 Text(
97                     Textl,
98                     style: const TextStyle(
99                         fontSize: 15,
100                         height: 0.001,
101                         color: Color.fromARGB(255, 158, 158, 158)), // TextStyle
102                 ), // Text
103                 Text(
104                     Texttr,
105                     style: const TextStyle(
106                         height: 4,
107                         fontSize: 13,
108                     ), // TextStyle
109                 ), // Text
110                 Text(
111                     Textb,
112                     style: const TextStyle(
113                         height: -1,
114                         fontSize: 13,
115                     ), // TextStyle
116                 ), // Text
117                 Text(
118                     Textbday,
119                     style: const TextStyle(
120                         height: 2,
121                         fontSize: 13,
122                     ), // TextStyle
123                 ), // Text
124                 Text(
125                     Textfb,
126                     style: const TextStyle(
127                         height: 3,

```

```

128         fontSize: 12,
129       ), // TextStyle
130     ), // Text
131     Text(
132       Textgm,
133       style: const TextStyle(
134         height: 0,
135         fontSize: 12,
136       ), // TextStyle
137     ), // Text
138     Text(
139       Texthm,
140       style: const TextStyle(
141         height: 3,
142         fontSize: 12,
143       ), // TextStyle
144     ), // Text
145   ]), // Column
146 ); // Padding
147 }
148 }
149

```



## 5. สรุปผลการทดลอง

จากการทดลองพบว่า Flutter สามารถสร้างแอปพลิเคชันที่มีความสวยงามและตอบสนองได้ดีบนอุปกรณ์มือถือและเดสก์ท็อป โดยใช้โค้ดเพียงชุดเดียว สิ่งนี้ทำให้ Flutter เป็นตัวเลือกที่น่าสนใจสำหรับนักพัฒนาที่ต้องการสร้างแอปพลิเคชันที่ทำงานได้อย่างราบรื่นบนอุปกรณ์หลายประเภท นอกจากนี้ การทดลองยังพบว่า Widgets ของ Flutter เป็นเครื่องมือที่มีประสิทธิภาพสำหรับการสร้างอินเทอร์เฟซผู้ใช้ที่ตอบสนองและปรับแต่งได้ Widgets ของ Flutter ช่วยให้นักพัฒนาสามารถสร้างอินเทอร์เฟซผู้ใช้ที่ซับซ้อนได้อย่างรวดเร็วและง่ายดายโดยสรุป การทดลองของ Flutter และ Widgets แสดงให้เห็นว่า Flutter เป็นแพลตฟอร์มที่มีศักยภาพสำหรับการพัฒนาแอปพลิเคชันข้ามแพลตฟอร์มที่มีประสิทธิภาพและน่าประทับใจ