FLUTTER WEEK6

PATIPAN WATJANAPRON

ADVANCED UI AND ANIMATION

CUSTOM WIDGET

CUSTOM WIDGET คืออะไร

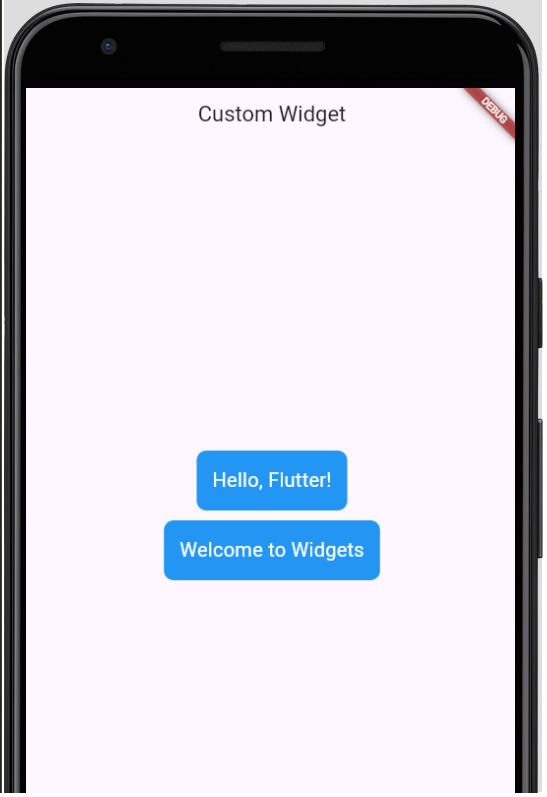
- Custom Widgets คือ Widget ที่เราสร้างขึ้นเอง เพื่อให้สามารถนำกลับมาใช้ซ้ำได้ง่ายขึ้น แทนที่จะต้องเขียนโค้ดซ้ำ ๆ ทุกครั้งที่ต้องการ UI ที่คล้ายกัน

ทำไมต้องใช้ CUSTOM WIDGETS?

- **ลดการซ้ำซ้อนของโค้ด** แยกส่วน UI ออกเป็น Component ที่นำมาใช้ซ้ำได้
- **ทำให้โค้ดอ่านง่ายขึ้น** แยก UI ออกจาก Business Logic ชัดเจน
- **แก้ไขและบำรุงรักษาง่าย** ถ้าต้องแก้ไข UI ก็แก้เพียงจุดเดียว

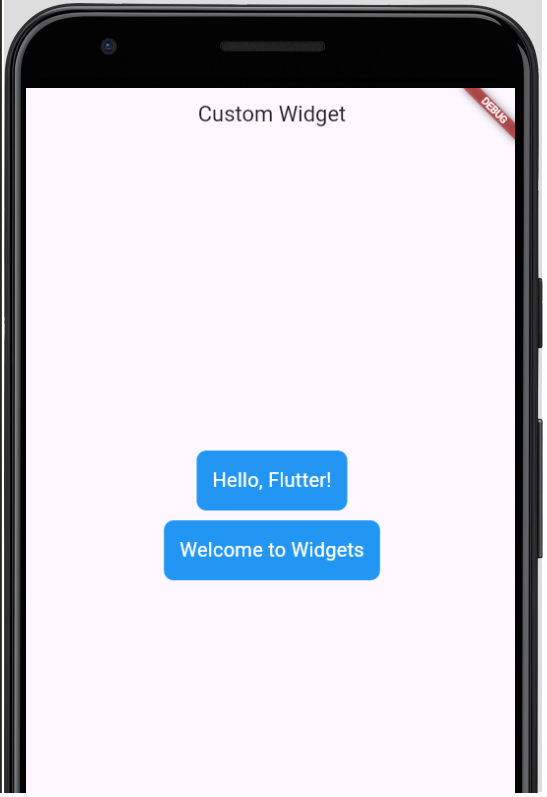
ตัวอย่างการสร้าง UI แบบปกติ

```
Center(
            child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            Container(
              padding: const EdgeInsets.all(16),
              decoration: BoxDecoration(
                  color: Colors.blue, borderRadius: BorderRadius.circular(10)),
              child: const Text("Hello, Flutter!",
                  style: TextStyle(fontSize: 20, color: Colors.white)),
            ),
            const SizedBox(height: 10),
            Container(
              padding: const EdgeInsets.all(16),
              decoration: BoxDecoration(
                  color: Colors.blue, borderRadius: BorderRadius.circular(10)),
              child: const Text("Welcome to Widgets",
                  style: TextStyle(fontSize: 20, color: Colors.white)),
            ),
          ],
        ))
```



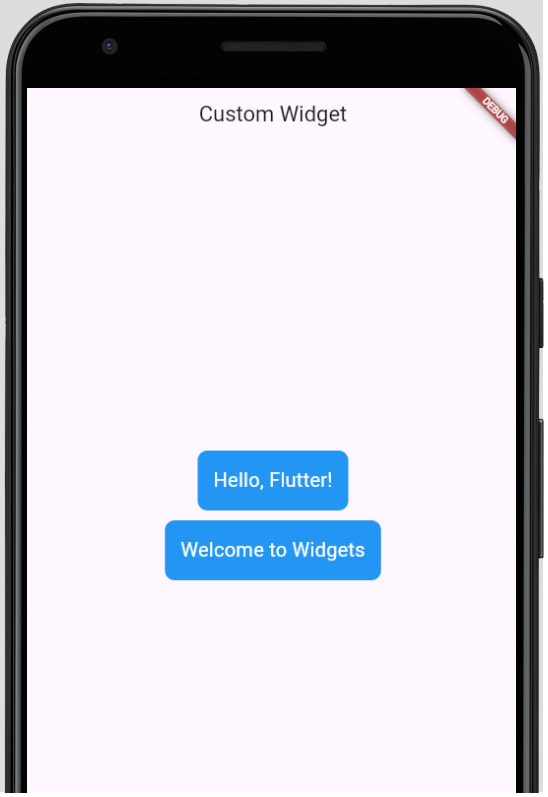
สังเกตเห็น โค้ดที่ทำงานซ้ำแบบเดิมไหม ?

```
Center(
          child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          Container(
            padding: const EdgeInsets.all(16),
            decoration: BoxDecoration(
                color: Colors.blue, borderRadius: BorderRadius.circular(10)),
            child: const Text("Hello, Flutter!",
                style: TextStyle(fontSize: 20, color: Colors.white)),
          ),
          const SizedBox(height: 10),
          Container(
            padding: const EdgeInsets.all(16),
            decoration: BoxDecoration(
                color: Colors.blue, borderRadius: BorderRadius.circular(10)),
            child: const Text("Welcome to Widgets",
                style: TextStyle(fontSize: 20, color: Colors.white)),
          ),
        ],
      ))
```



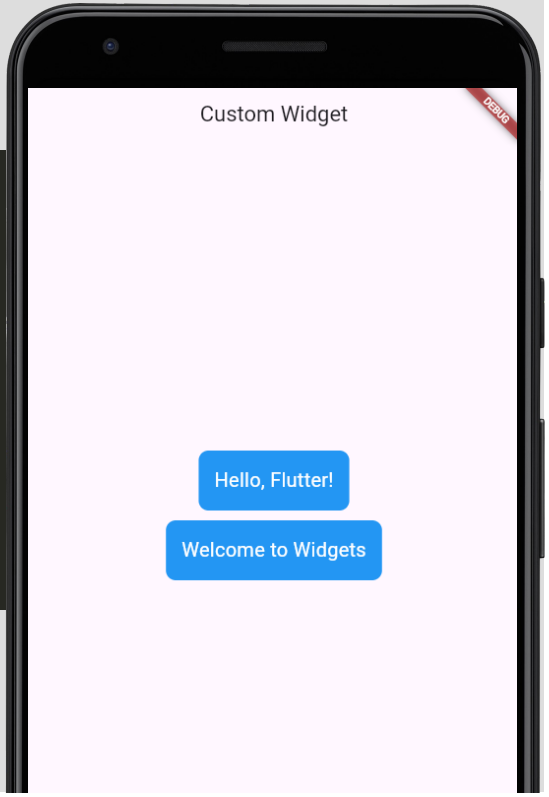
ทำ CUSTOM WIDGET เพื่อใช้งานดีกว่า

```dart
class CustomCard extends StatelessWidget {
  final String text;

  CustomCard({required this.text});

  @override
  Widget build(BuildContext context) {
    return Container(
      padding: const EdgeInsets.all(16),
      decoration: BoxDecoration(
          color: Colors.blue, borderRadius: BorderRadius.circular(10)),
      child:
          Text(text, style: const TextStyle(fontSize: 20, color: Colors.white)),
    );
  }
}
```



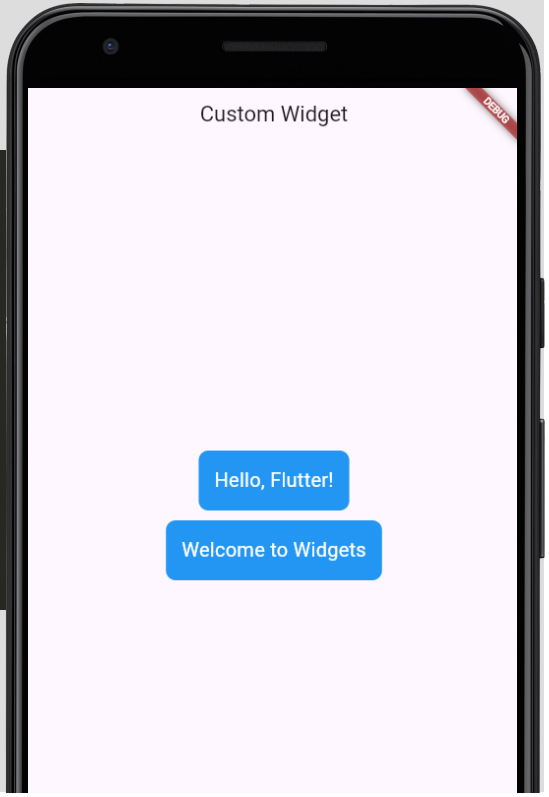
เรียกใช้งาน CUSTOM WIDGET

```
Center(
          child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          CustomCard(text: "Hello, Flutter!"),
          const SizedBox(height: 10),
          CustomCard(text: "Welcome to Widgets"),
        ],
      ))
```



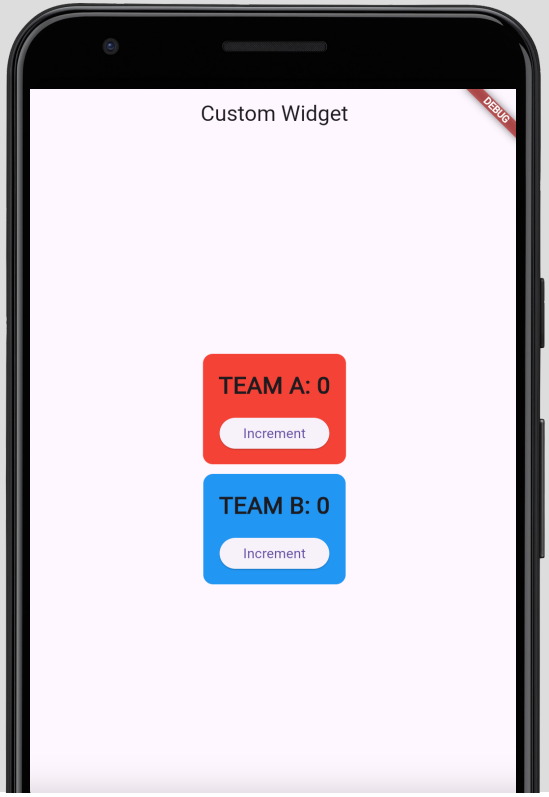
เรียกใช้งาน CUSTOM WIDGET

```
Center(
          child: Column(
        mainAxisAlignment: MainAxisAlignment.center,
        children: [
          CustomCard(text: "Hello, Flutter!"),
          const SizedBox(height: 10),
          CustomCard(text: "Welcome to Widgets"),
        ],
      ))
```



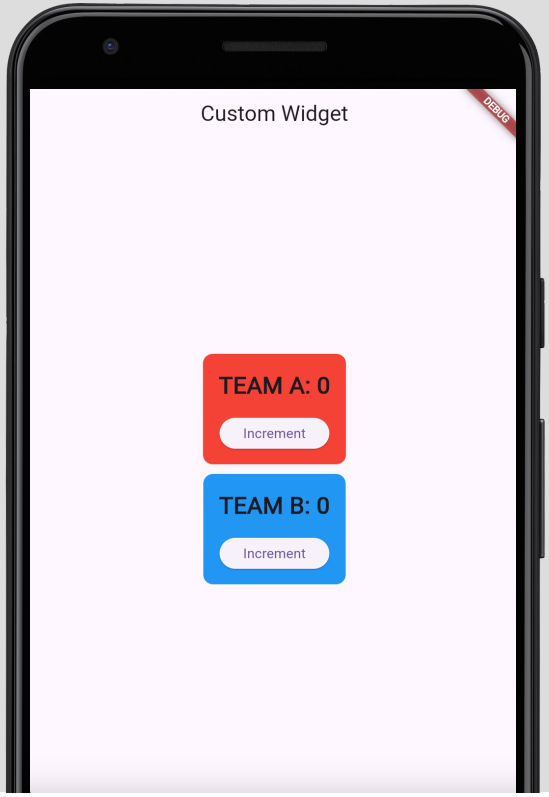
```dart
class CustomCounterWidget extends StatefulWidget {
  final String title;
  final Color backgroundColor;
  const CustomCounterWidget(
      {super.key, required this.title, required this.backgroundColor});
  @override
  _CustomCounterWidgetState createState() => _CustomCounterWidgetState();
}
class _CustomCounterWidgetState extends State<CustomCounterWidget> {
  int _counter = 0;
  void _incrementCounter() {
    setState(() {
      _counter++;
    });
  }
  @override
  Widget build(BuildContext context) {
    return Container(
        padding: const EdgeInsets.all(16),
        decoration: BoxDecoration(
            color: widget.backgroundColor,
            borderRadius: BorderRadius.circular(10)),
        child: Column(mainAxisAlignment: MainAxisAlignment.center, children: [
          Text(
            '${widget.title}: $_counter',
            style: const TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
          ),
          const SizedBox(height: 16),
          ElevatedButton(
            onPressed: _incrementCounter,
            child: const Text('Increment'),
          ),
        ]));
  }
}
```

ทำ CUSTOM WIDGET แบบ STATEFUL



เรียกใช้งาน CUSTOM WIDGET (STATFUL)

```
const Center(
            child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            CustomCounterWidget(
              title: 'TEAM A',
              backgroundColor: Colors.red,
            ),
            SizedBox(
              height: 10,
            ),
            CustomCounterWidget(
              title: 'TEAM B',
              backgroundColor: Colors.blue,
            ),
          ],
        )
      )
```

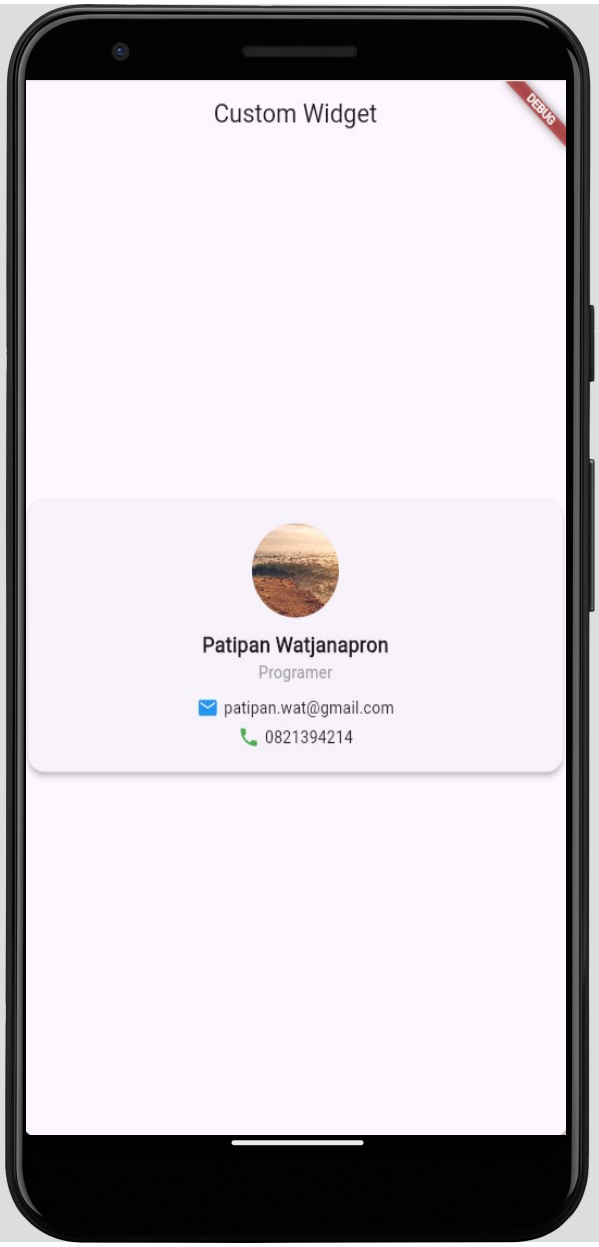


สร้าง CUSTOM WIDGET ง่ายๆ

โจทย์: สร้าง Custom widget Profile Card (Stateless)

- ดูโครงสร้างจากการเรียกใช้งาน Custom widget ด้านล่าง และตัวอย่างหน้าจอ (ไม่ต้องเหมือนเป๊ะ)

```
ProfileCard(
  name: 'Patipan Watjanapron',
  position: 'Programer',
  email: 'patipan.wat@gmail.com',
  phoneNumber: '0821394214',
  imageUrl:
      'https://fastly.picsum.photos/id/27/3264/1836.jpg?hmac=p3BVIgKKQpHhfGRRCbsi2MCAzw8mWBCayBsKxxtWO8g',
) // ProfileCard
```



THEMES

THEMES คืออะไร

- Custom Widgets คือ Widget ที่เราสร้างขึ้นเอง เพื่อให้สามารถนำกลับมาใช้ซ้ำได้ง่ายขึ้น แทนที่จะต้องเขียนโค้ดซ้ำ ๆ ทุกครั้งที่ต้องการ UI ที่คล้ายกัน

ประโยชน์ของ THEMES

- **ช่วยให้ UI เป็นระบบ** กำหนดสี ฟอนต์ และสไตล์เดียวกันทั้งแอป
- **เปลี่ยนธีมง่ายขึ้น** แค่เปลี่ยนค่าที่เดียว UI ทั้งแอปจะเปลี่ยนตาม
- **รองรับ Dark Mode** Flutter รองรับ Light/Dark Theme ได้ในตัว

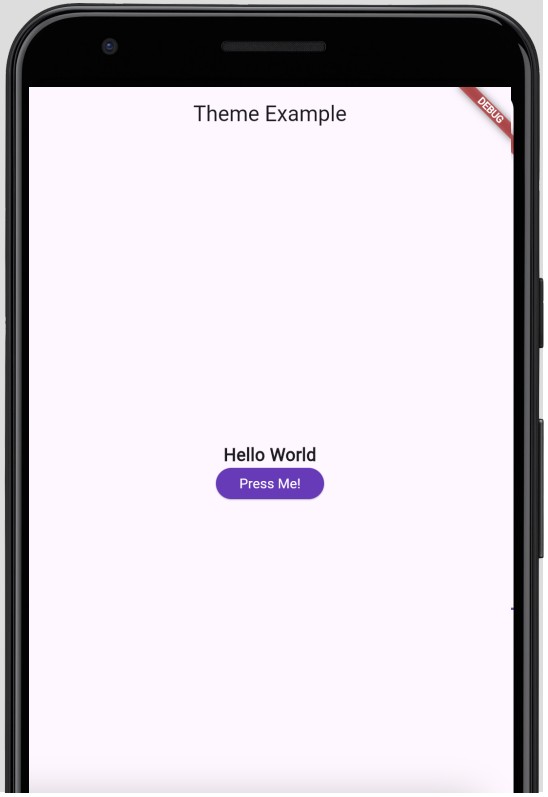
กำหนด THEME ใน MATERIALAPP

- Flutter มี ThemeData สำหรับกำหนดธีมหลักของแอป

```
MaterialApp(
    theme: ThemeData(
        primarySwatch: Colors.red, // กำหนดสีหลัก
        textTheme: const TextTheme(
          bodyMedium: TextStyle(fontSize: 18, fontWeight: FontWeight.bold),
        ),
        elevatedButtonTheme: ElevatedButtonThemeData(
          style: ElevatedButton.styleFrom(
            backgroundColor: Colors.deepPurple, // สีปุ่มหลัก
            foregroundColor: Colors.white, // สีข้อความปุ่ม
          ),
        )),
    home: const HomeScreen());
```

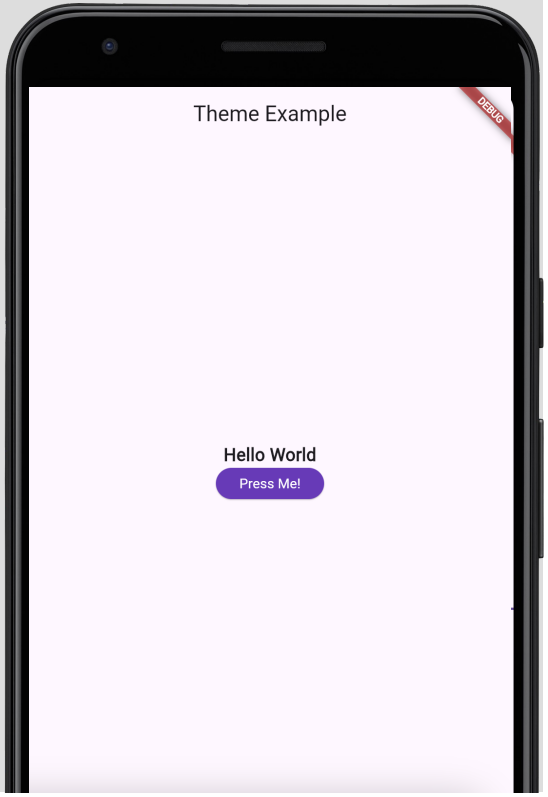
การดึงค่าจาก THEME ใน WIDGET

```
Center(
          child:
          Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text('Hello World',
                style: Theme.of(context).textTheme.bodyMedium),
              ElevatedButton(
                onPressed: () {},
                child: const Text("Press Me!"),
        )
      ])
      ),
```

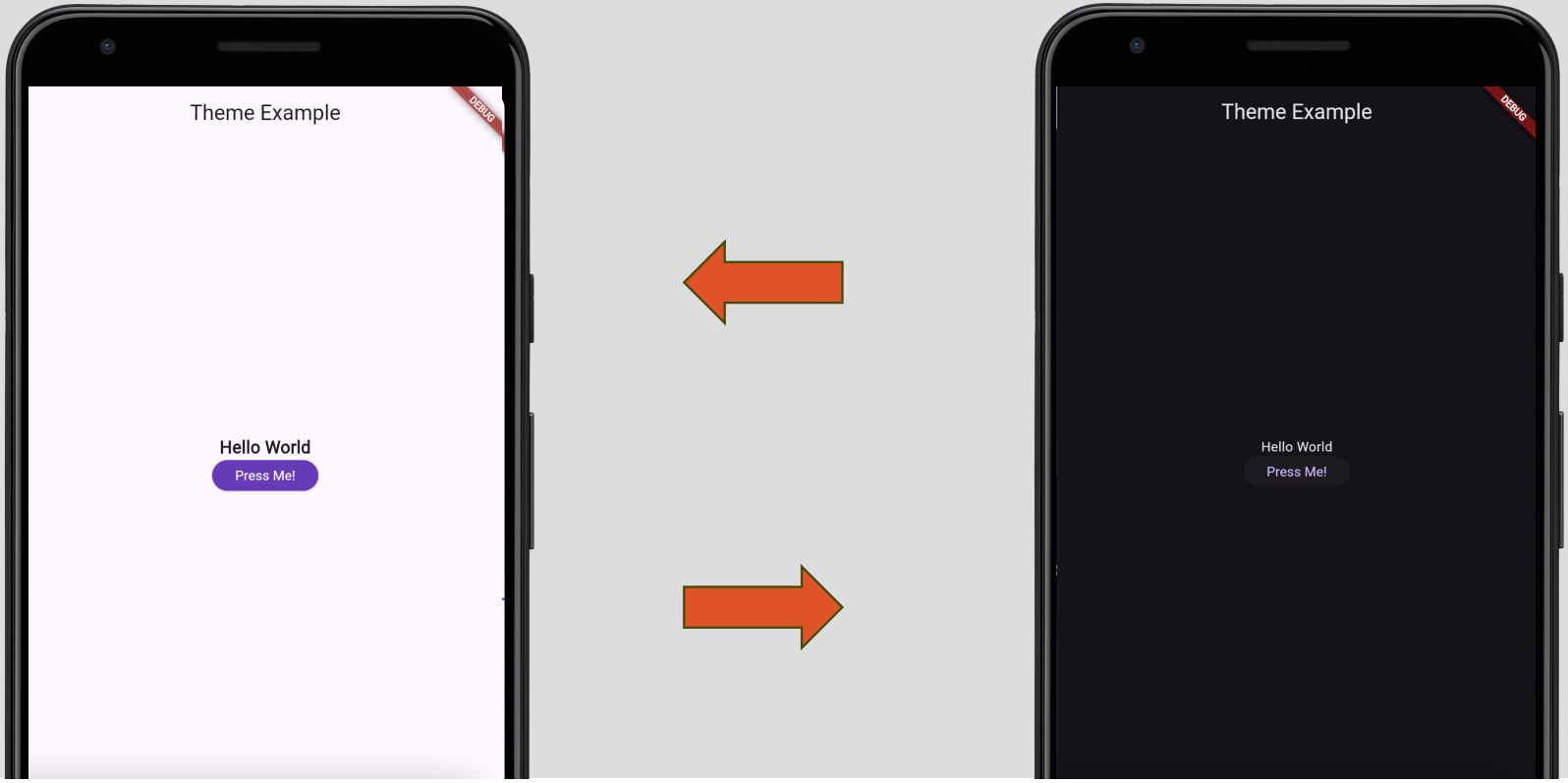


การดึงค่าจาก THEME ใน WIDGET

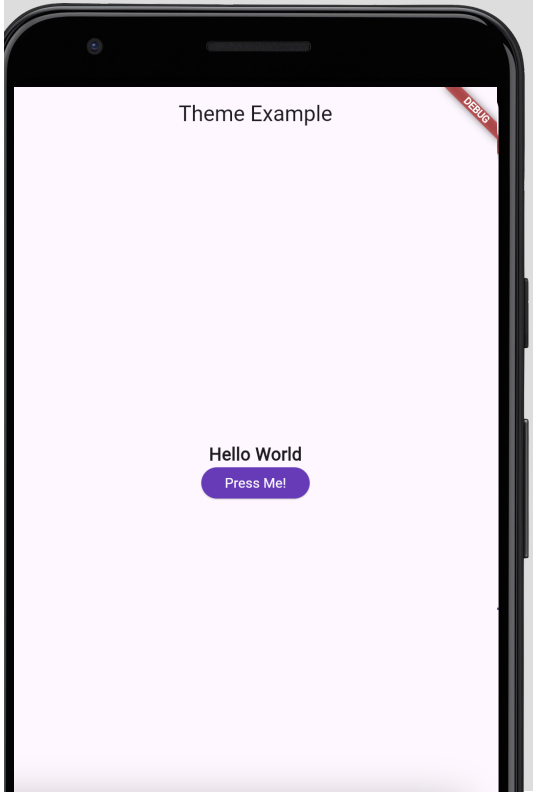
```
Center(
          child:
          Column(
            mainAxisAlignment: MainAxisAlignment.center,
            children: [
              Text('Hello World',
                style: Theme.of(context).textTheme.bodyMedium),
              ElevatedButton(
                onPressed: () {},
                child: const Text("Press Me!"),
        )
      ])
      ),
```



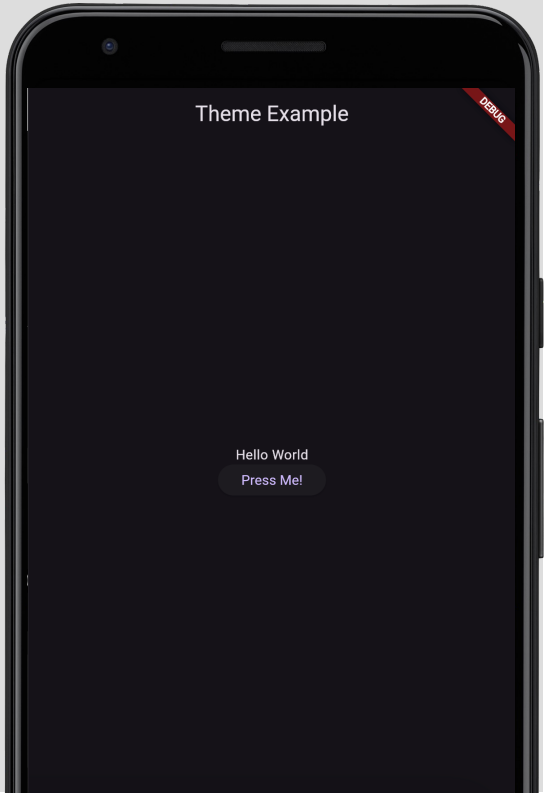
LIGHT/DARK


Theme Example
DEBUG
Hello World
Press Me!
Theme Example
DEBUG
Hello World
Press Me!

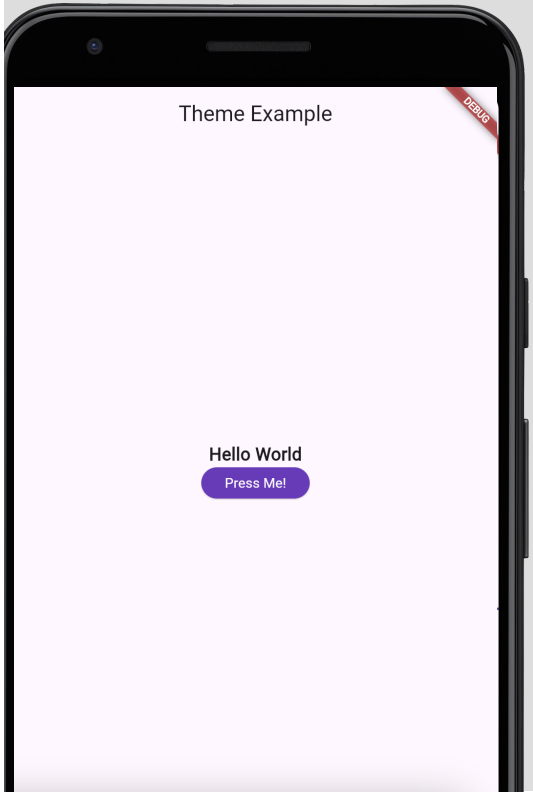
การใช้งาน LIGHT/DARK



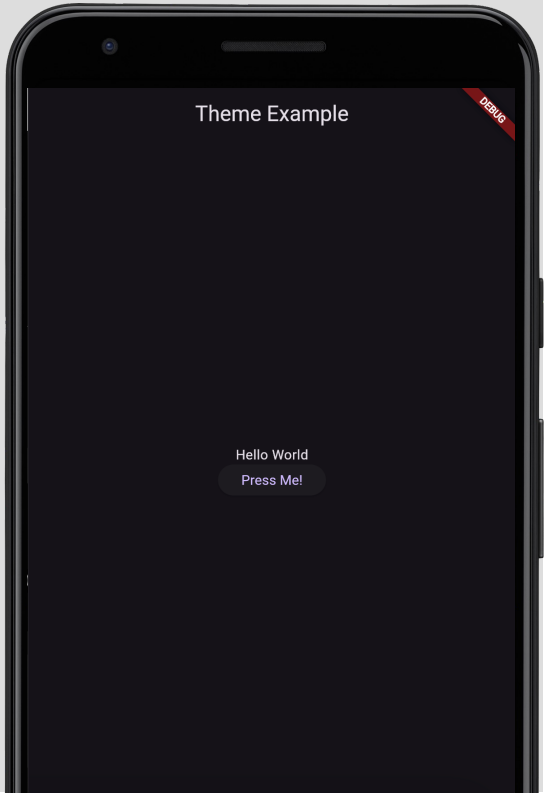

```dart
MaterialApp(
        darkTheme: ThemeData.dark(), // ธีมมืด
        theme: ThemeData.light(), // ธีมสว่าง
        themeMode: ThemeMode.system, // เปลี่ยนตามระบบ (Auto)
        home: const HomeScreen()
);
```



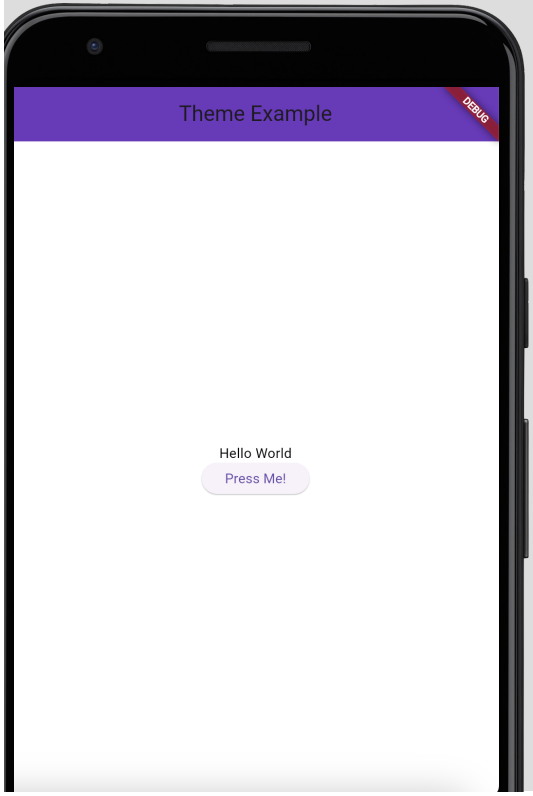
การใช้งาน LIGHT/DARK



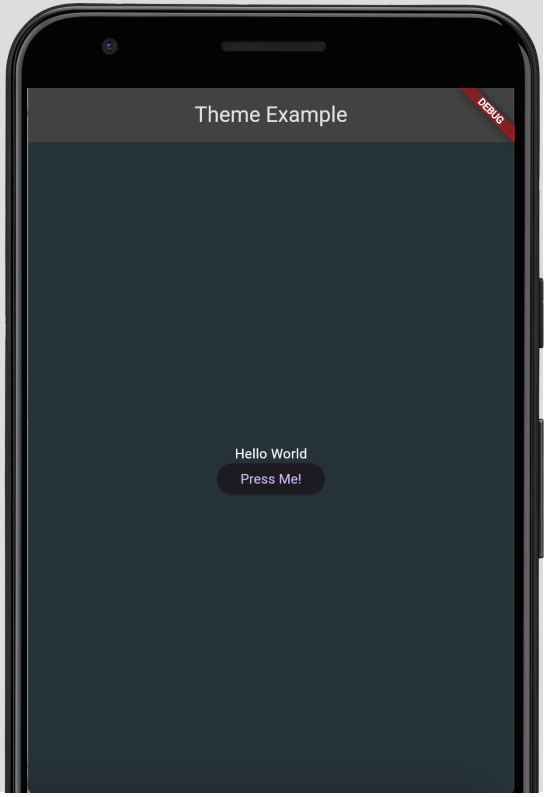

```dart
MaterialApp(
        darkTheme: ThemeData.dark(), // ธีมมืด
        theme: ThemeData.light(), // ธีมสว่าง
        themeMode: ThemeMode.system, // เปลี่ยนตามระบบ (Auto)
        home: const HomeScreen()
);
```



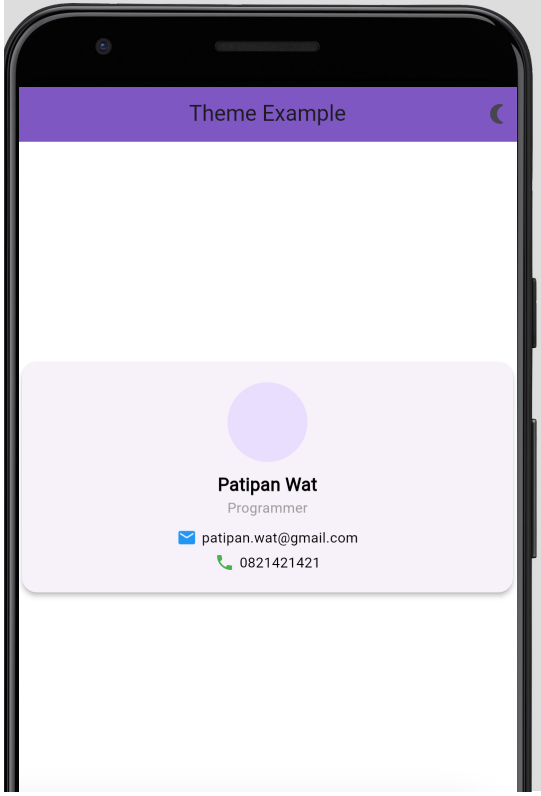
ตั้งค่า LIGHT/DARK (CUSTOM)




```dart
MaterialApp(
      title: 'ThemeMode Demo',
      theme: ThemeData.light().copyWith(
        scaffoldBackgroundColor: Colors.white, // สีพื้นหลังLight Mode
        textTheme: const TextTheme(
            bodyMedium: TextStyle(color: Colors.black)), // สีข้อความ
        appBarTheme: const AppBarTheme(backgroundColor: Colors.blue),
      ),
      darkTheme: ThemeData.dark().copyWith(
        scaffoldBackgroundColor: Colors.blueGrey[900], // สีพื้นหลังDark Mode
        textTheme: const TextTheme(
            bodyMedium: TextStyle(color: Colors.white)), // สีข้อความ
        appBarTheme: AppBarTheme(backgroundColor: Colors.grey[800]),
      ),
      themeMode: ThemeMode.dark, // เปลี่ยนธีมตามค่าปัจจุบัน
      home: const HomeScreen(),
    );
```

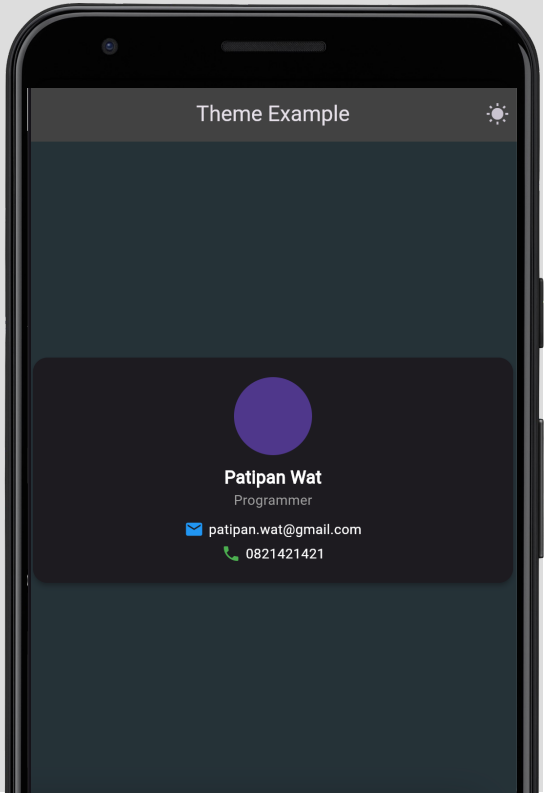


อัปเกรด CUSTOM WIDGET



โจทย์: อัปเกรด Custom widget Profile Card ให้มี Mode dark/light

- เพิ่มปุ่มขวามือ (action) บน AppBar โดยเพิ่มปุ่มเปลี่ยนโหมด สลับไปมาระหว่าง dark/light
- Hint : แนะนำให้ส่งฟังก์ชัน เปลี่ยนโหมด ThemeMode.light หรือ ThemeMode.dark จาก หน้า MateriaApp พร้อมสถานะ ThemeMode เพื่อเช็คสถานะสำหรับเปลี่ยน icon ปุ่ม



ANIMATION

ANIMATION ใน FLUTTER

- การใช้งาน Animation
  - ทำให้ UI มีชีวิตชีวา เหมือนมีชีวิต และดูเป็นธรรมชาติ
  - ใช้สำหรับ เปลี่ยนสถานะของ UI เช่น การเปลี่ยนสี, ขนาด, การเคลื่อนที่
- แบ่งเป็น Implicit Animation และ Explicit Animation (Custom ได้มากกว่า แต่ซับซ้อนมากกว่า)

IMPLICIT ANIMATIONS คืออะไร?

- **Implicit Animations** คือรูปแบบของ **Animation** ใน Flutter ที่ทำให้ UI เคลื่อนไหวได้อย่างราบรื่น **โดยไม่ต้องจัดการกับ Animation Controller เอง**

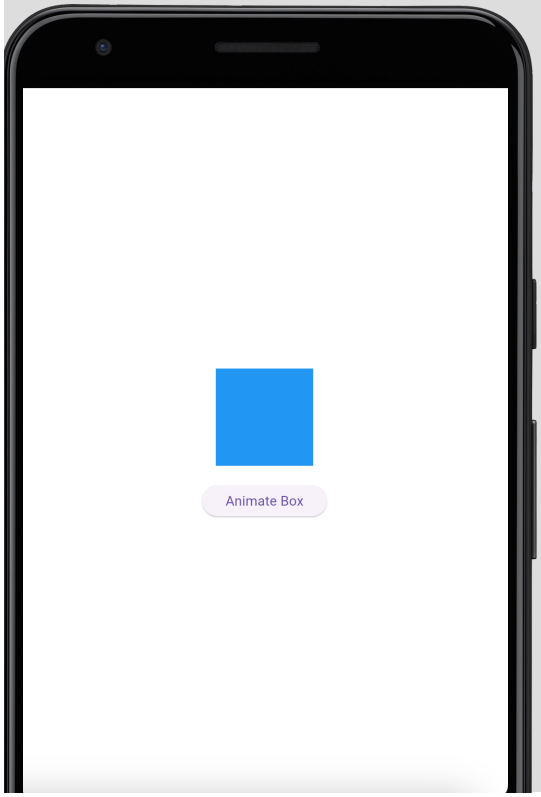
จุดเด่น IMPLICIT ANIMATIONS

- ใช้งานง่าย ไม่ต้องเขียนโค้ด Animation ซับซ้อน
- Flutter จะจัดการให้เองเมื่อค่าของ Widget เปลี่ยนแปลง
- ใช้ setState() เพื่ออัปเดตค่า แล้ว UI จะค่อย ๆ เปลี่ยนไปเอง

WIDGET ที่แนะนำ IMPLICIT ANIMATIONS

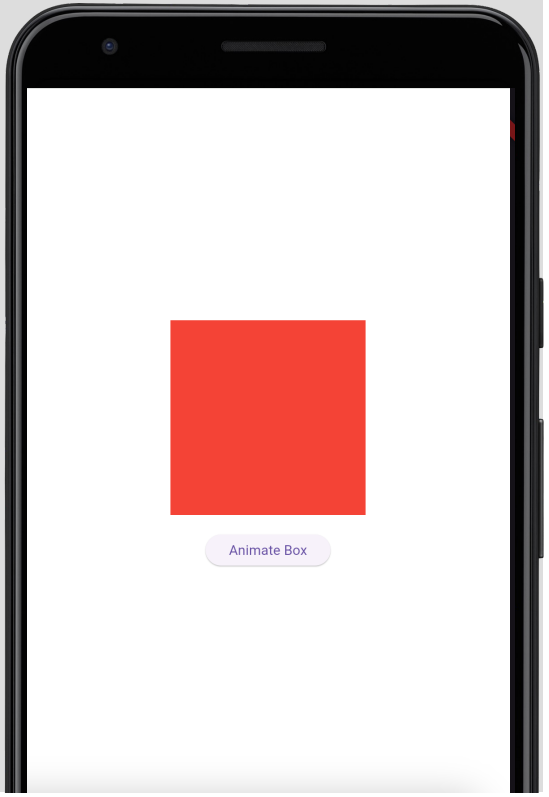
Widget	ใช้ทำอะไร
AnimatedContainer	เปลี่ยนขนาด, สี, padding และอื่น ๆ
AnimatedOpacity	เปลี่ยนความโปร่งแสงของ Widget
AnimatedAlign	เปลี่ยนตำแหน่งของ Widget ใน Align
AnimatedPadding	เปลี่ยนค่า padding ของ Widget
AnimatedPositioned	ใช้กับ Stack เพื่อเปลี่ยนตำแหน่ง Widget
AnimatedSwitcher	สลับ Widget แบบมี Animation

AnimatedContainer

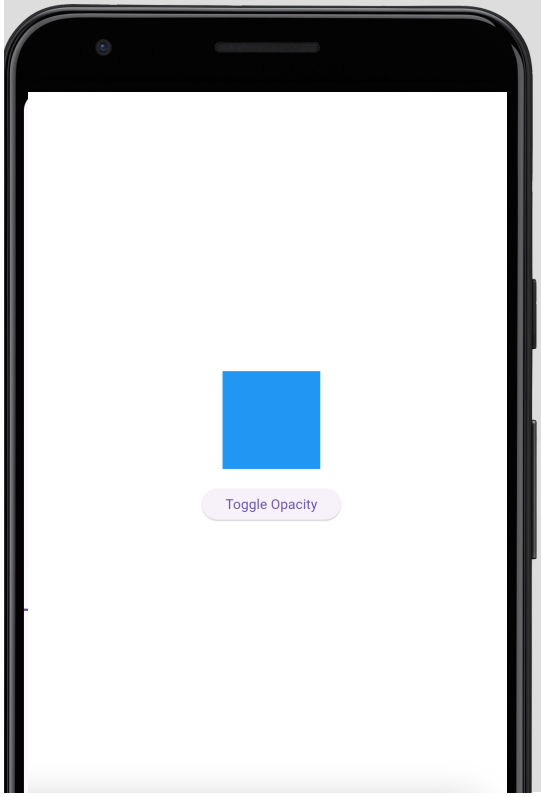



```
AnimatedContainer(
          duration: const Duration(seconds: 1), // ระยะเวลาของAnimation
          curve: Curves.easeInOut, // รูปแบบการเคลื่อนไหว
          width: _size, //เปลี่ยนขนาดความกว้าง
          height: _size,//เปลี่ยนขนาดสูง
          color: _color,//เปลี่ยนสี
        )
```

```
double _size = 100;
Color _color = Colors.blue;

 void _changeBox() {
   setState(() {
     _size = _size == 100 ? 200 : 100;
     _color = _color == Colors.blue ? Colors.red : Colors.blue;
   });
 }
```

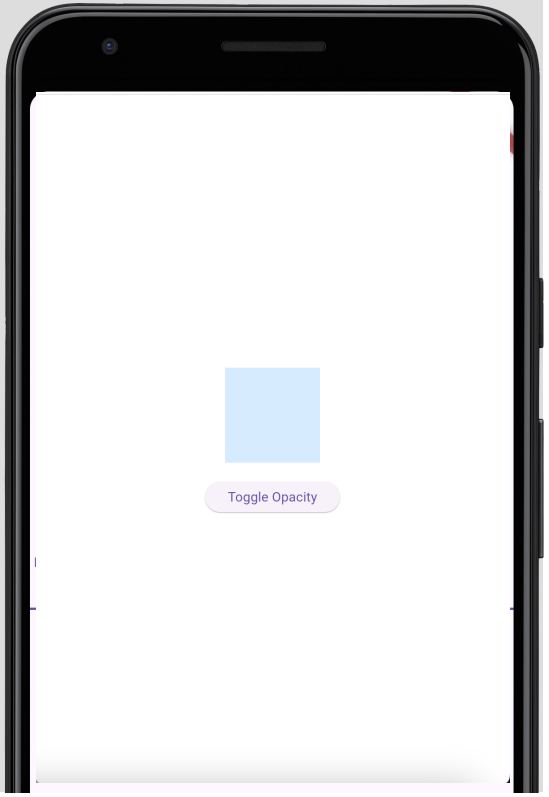


AnimatedOpacity

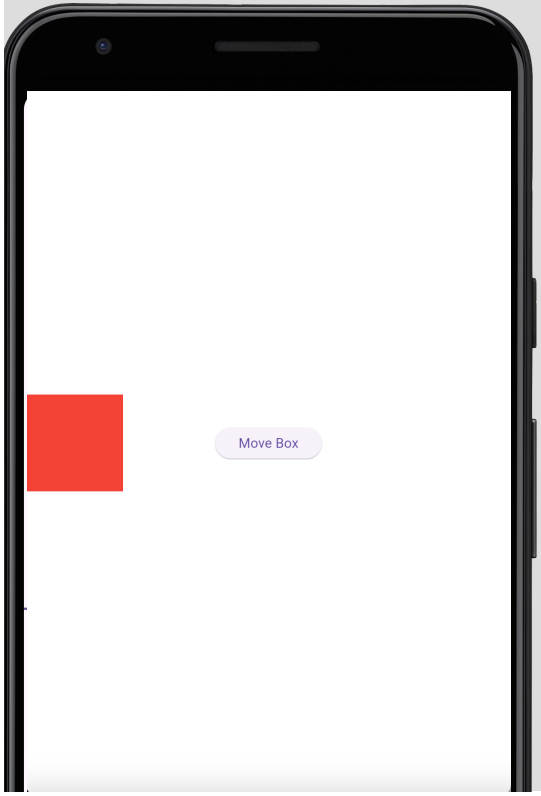



```
AnimatedOpacity(
          duration: const Duration(seconds: 1),
          opacity: _opacity,
          child: Container(
            width: 100,
            height: 100,
            color: Colors.blue,
          ),
        ),
```

```
double _opacity = 1.0;

 void _toggleOpacity() {
   setState(() {
     _opacity = _opacity == 1.0 ? 0.2 : 1.0;
   });
 }
```

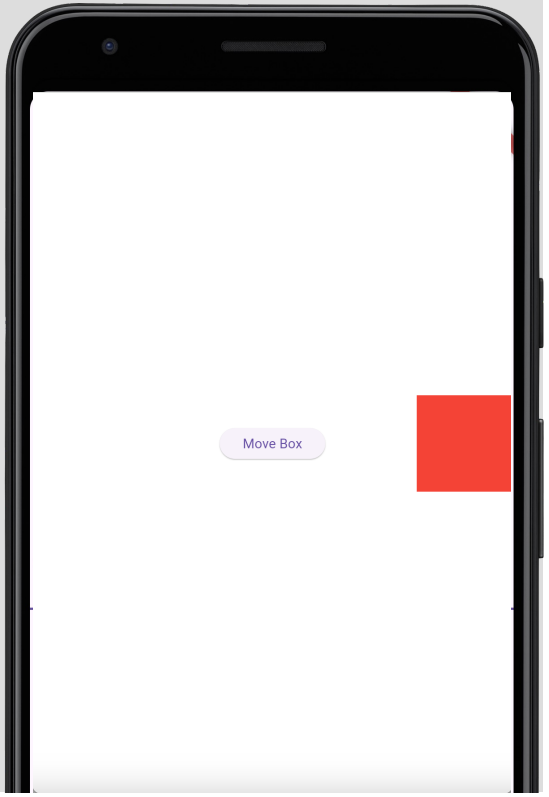


AnimatedAlign

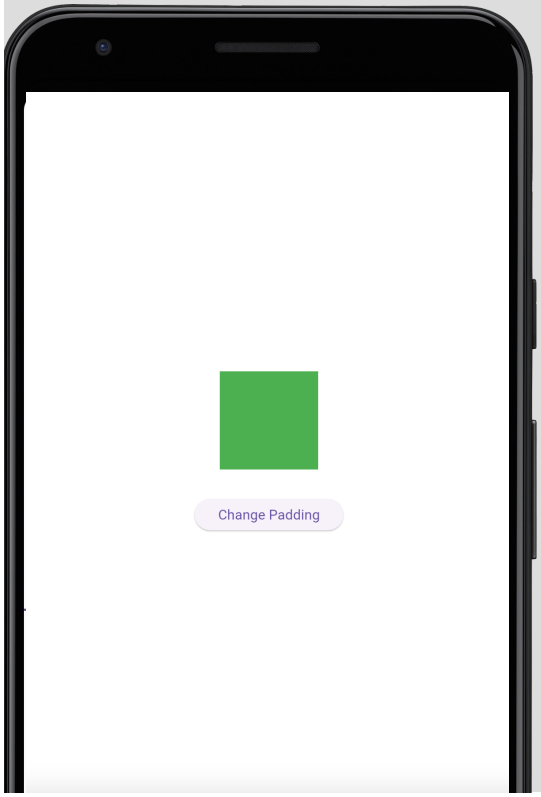



```
Stack(
      children: [
        AnimatedAlign(
          duration: const Duration(seconds: 1),
          alignment: _isLeft ? Alignment.centerLeft : Alignment.centerRight,
          child: Container(width: 100, height: 100, color: Colors.red),
        ),
        Center(
          child: ElevatedButton(
            onPressed: _togglePosition,
            child: const Text('Move Box'),
          ),
        ),
      ],
    );
```

```
bool _isLeft = true;

 void _togglePosition() {
   setState(() {
     _isLeft = !_isLeft;
   });
 }
```

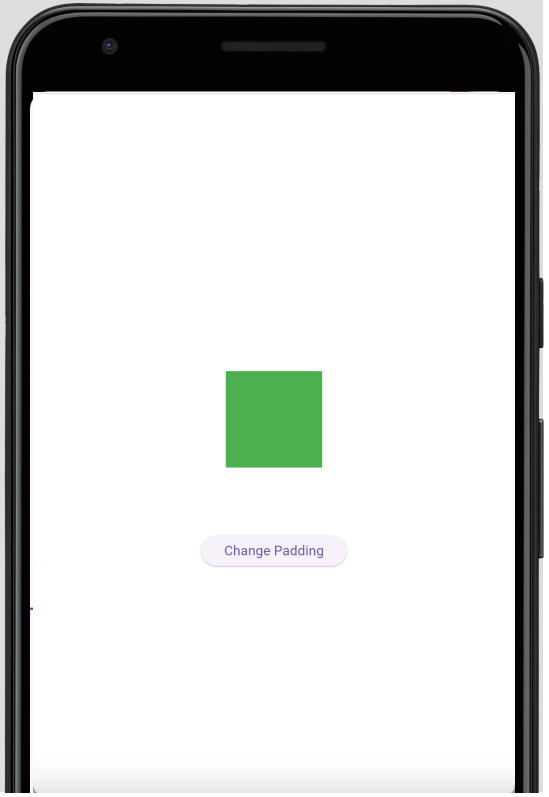


AnimatedPadding

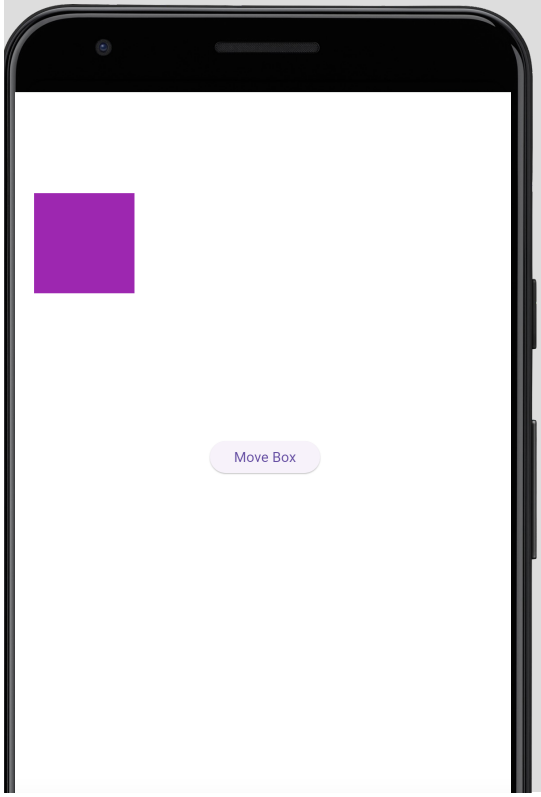



```dart
AnimatedPadding(
  duration: const Duration(seconds: 1),
  padding: EdgeInsets.all(_padding),
  child: Container(width: 100, height: 100, color: Colors.green),
),
```

```dart
double _padding = 10;

void _increasePadding() {
  setState(() {
    _padding = _padding == 10 ? 50 : 10;
  });
}
```

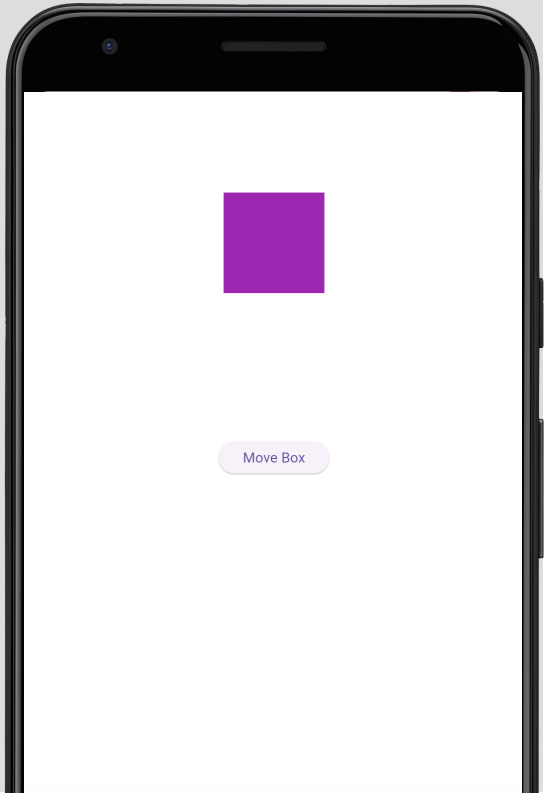


AnimatedPositioned

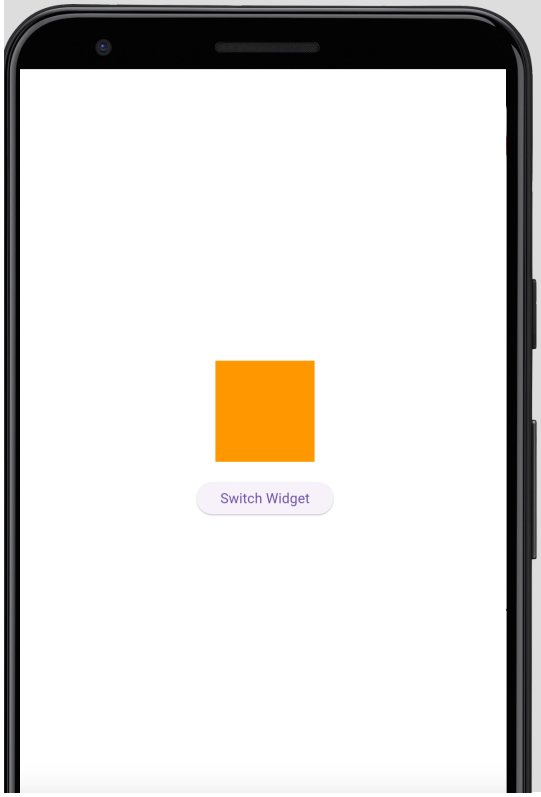



```
Stack(
      children: [
        AnimatedPositioned(
          duration: const Duration(seconds: 1),
          left: _isLeft ? 20 : 200,
          top: 100,
          child: Container(width: 100, height: 100, color: Colors.purple),
        ),
        Center(
          child: ElevatedButton(
            onPressed: _moveBox,
            child: const Text('Move Box'),
          ),
        ),
      ],
    );
```

```
bool _isLeft = true;

  void _moveBox() {
    setState(() {
      _isLeft = !_isLeft;
    });
  }
```

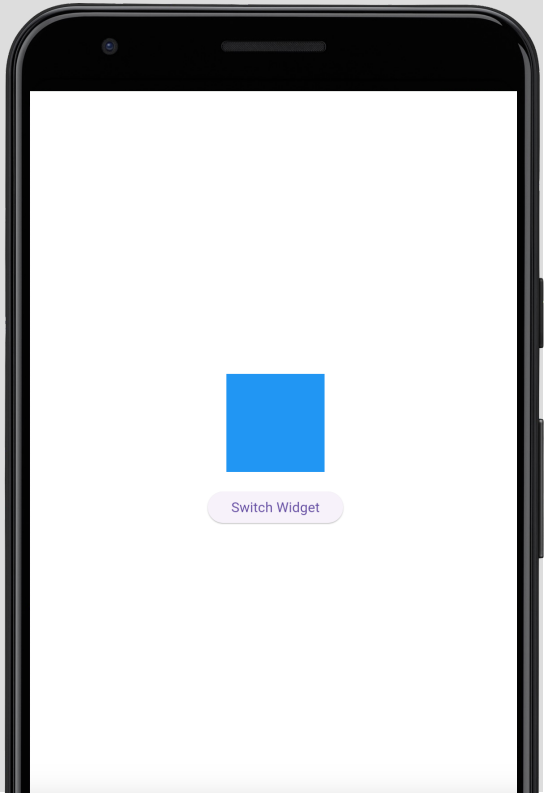


AnimatedSwicher




```
      AnimatedSwitcher(
        duration: const Duration(seconds: 1),
        child: _isFirst
            ? Container(
                key: const ValueKey(1),
                width: 100,
                height: 100,
                color: Colors.orange)
            : Container(
                key: const ValueKey(2),
                width: 100,
                height: 100,
                color: Colors.blue),
```

```
bool _isFirst = true;

  void _switchWidget() {
    setState(() {
      _isFirst = !_isFirst;
    });
  }
```



แบบฝึกหัด

โจทย์: ทำแอปให้พี่จ่ากดไฟจราจร

- สร้าง สัญญาณไฟจราจร ที่มีไฟ แดง, เหลือง, เขียว
- มีปุ่ม "เปลี่ยนไฟ" เพื่อเปลี่ยนสถานะของไฟแบบวนลูป
- เงื่อนไขการทำงานของไฟจราจร:
  - เริ่มต้น ไฟแดงสว่าง (opacity = 1.0), ไฟอื่นจาง (opacity = 0.3)
  - กดปุ่ม "เปลี่ยนไฟ" ⟶ ไฟแดงจางลง ไฟเหลืองสว่าง (opacity = 1.0)
  - กดอีกครั้ง ⟶ ไฟเหลืองจางลง ไฟเขียวสว่าง
  - กดอีกครั้ง ⟶ ไฟเขียวจางลง กลับไปเริ่มต้นที่ไฟแดง
- **Hint** : AnimatedOpacity เพื่อทำให้ไฟค่อย ๆ เปลี่ยนแบบนุ่มนวล และใช้ StatefulWidget เพื่อจัดการสถานะของไฟ