

Custom Widget

1. ui แบบดั้งเดิม

```
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)),
      ),
    ],
  ),
)
```

2. สร้าง UI Custom widget

```
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)),
    );
  }
}
```

วิธีใช้งาน

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

3.สร้าง UI Custom widget แบบ StatefulWidget

```
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'),
      ),
    ]));
  }
}
```

วิธีใช้งาน

```
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,  
      ),  
    ],  
  ),  
)
```

3.1 เวอร์ชันอัปเดต

```
import 'package:flutter/material.dart';  
  
class CustomCounterWidget extends StatefulWidget {  
  final String title;  
  final Color backgroundColor;  
  final Function onPlayerWin;  
  
  const CustomCounterWidget(  
    {super.key,  
    required this.title,  
    required this.backgroundColor,  
    required this.onPlayerWin});  
  
  @override  
  _CustomCounterWidgetState createState() => _CustomCounterWidgetState();  
}  
  
class _CustomCounterWidgetState extends State<CustomCounterWidget> {  
  int _counter = 0;  
  void _incrementCounter() {  
    setState(() {  
      _counter++;  
    });  
    if (_counter == 5) {  
      widget.onPlayerWin(widget.backgroundColor);  
    }  
  }  
}
```

```

}

@override
Widget build(BuildContext context) {
  return Container(
    padding: const EdgeInsets.all(16),
    height: 150,
    width: 150,
    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'),
      ),
    ])),
}
}

```

วิธีใช้งาน

```

//... state fulwidget

Color winPlayerColor = Colors.white;

void setWinPlayerColor(Color winColor) {
  setState(() {
    winPlayerColor = winColor;
  });
}

@override
Widget build(BuildContext context) {
  return Scaffold(
    backgroundColor: winPlayerColor,
    appBar: AppBar(title: const Text('Custom Widget')),
    body: Center(
      child: Column(mainAxisAlignment: MainAxisAlignment.center, children: [
        CustomCounterWidget(
          title: 'Team A',
          backgroundColor: Colors.red,
          onPlayerWin: setWinPlayerColor,

```

```

    ),
    const SizedBox(height: 10),
    CustomCounterWidget(
      title: 'Team B',
      backgroundColor: Colors.blue,
      onPlayerWin: setWinPlayerColor,
    ),
  )),
);
}

```

Theme

1.การประกาศ Theme

```

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();
)

```

2.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!"),
        ),
      ],
    ),
)

```

3.การใช้งาน Dark Mode or Light Mode

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

4.ปรับ Custom Theme Dark/Light

```
MaterialApp(  
  title: 'ThemeMode Demo',  
  theme: ThemeData.light().copyWith(  
    scaffoldBackgroundColor: Colors.white, // สีพื้นหลัง Light Mode  
    textTheme: const TextTheme(  
      bodyMedium: TextStyle(color: Colors.black)), // สีข้อความ  
    appBarTheme: const AppBarTheme(background-color: Colors.deepPurple),  
  ),  
  darkTheme: ThemeData.dark().copyWith(  
    scaffoldBackgroundColor: Colors.blueGrey[900], // สีพื้นหลัง Dark Mode  
    textTheme: const TextTheme(  
      bodyMedium: TextStyle(color: Colors.white)), // สีข้อความ  
    appBarTheme: AppBarTheme(background-color: Colors.grey[800]),  
  ),  
  themeMode: ThemeMode.light, // เปลี่ยนชีมตามค่าปัจจุบัน  
  home: const HomeScreen(),  
);
```

Animated

1. AnimatedContainer

```
Column(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [  
    AnimatedContainer(  
      duration: const Duration(seconds: 1), // ระยะเวลาของ Animation  
      curve: Curves.easeInOut, // รูปแบบการเคลื่อนไหว  
      width: _size, // เปลี่ยนขนาดความกว้าง
```

```

        height: _size, // เปลี่ยนขนาดสูง
        color: _color, // เปลี่ยนสี
      ),
      const SizedBox(height: 20),
      ElevatedButton(
        onPressed: _changeBox,
        child: const Text('Animate Box'),
      ),
    ],
  );

```

ทดสอบกำหนดขนาด และเรียกใช้ _changeBox() ผ่านปุ่มกด

```

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

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

```

2. AnimatedOpacity

```

Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [
    AnimatedOpacity(
      duration: const Duration(seconds: 1),
      opacity: _opacity,
      child: Container(
        width: 100,
        height: 100,
        color: Colors.blue,
      ),
    ),
    const SizedBox(height: 20),
    ElevatedButton(
      onPressed: _toggleOpacity,
      child: const Text('Toggle Opacity'),
    ),
  ],
)

```

```
);
```

ทดสอบกำหนดขนาด และเรียกใช้ `_toggleOpacity()` ผ่านปุ่มกด

```
double _opacity = 1.0;

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

3. 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'),
      ),
    ),
  ],
);
```

ทดสอบกำหนดขนาด และเรียกใช้ `_togglePosition()` ผ่านปุ่มกด

```
bool _isLeft = true;

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


4. AnimatedPadding

```
Column(  
  mainAxisAlignment: MainAxisAlignment.center,  
  children: [  
  
    AnimatedPadding(  
      duration: const Duration(seconds: 1),  
      padding: EdgeInsets.all(_padding),  
      child: Container(width: 100, height: 100, color: Colors.green),  
    ),  
  
    const SizedBox(height: 20),  
    ElevatedButton(  
      onPressed: _increasePadding,  
      child: const Text('Change Padding'),  
    ),  
  ],  
);
```

ทดสอบกำหนดขนาด และเรียกใช้ _increasePadding () ผ่านปุ่มกด

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

5. 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'),
      ),
    ),
  ],
);

```

ทดสอบกำหนดขนาด และเรียกใช้ _moveBox () ผ่านปุ่มกด

```

bool _isLeft = true;

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

```

6. AnimatedSwitcher

```

Column(
  mainAxisAlignment: MainAxisAlignment.center,
  children: [

    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),

    ),
    const SizedBox(height: 20),
    ElevatedButton(
        onPressed: _switchWidget,
        child: const Text('Switch Widget'),
    ),
],
);

```

ทดสอบกำหนดขนาด และเรียกใช้ _switchWidget () ผ่านปุ่มกด

```

bool _isFirst = true;

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

```