

ใบงานการทดลองที่ 7

เรื่อง Widgets

1. จุดประสงค์

1. ใช้ RadioButton widget ได้
2. ใช้ Checkbox widget ได้
3. ใช้ Slider widget ได้
4. ใช้ Switch widget ได้
5. ใช้ AlertDialog widget ได้
6. ใช้ TextField widget ได้

2. ทฤษฎี

บทที่ 3.3 Input Widget

3.3.1 Radio button

Radio button คือ จะใช้ในการเลือกแค่ตัวเลือกนั้นตัวเลือกเดียว ไม่สามารถเลือกได้หลายตัวเลือก จะอยู่ในรูปของวงกลมเป็นพื้นสีขาว ๆ ถ้าเราต้องการเลือก สามารถการกดจุดเข้าไปในวงกลมนั้นได้ การใช้ตัวเลือกลักษณะนี้อาจทำได้หลายหลายรูปแบบเช่น การทำแบบทดสอบ การประเมินผล การทำแบบสอบถามความคิดเห็น ทำให้ง่ายต่อการเก็บข้อมูล Radio button ประกอบไปด้วย

คุณสมบัติ (Property) ที่สำคัญที่สำคัญของ Radio button

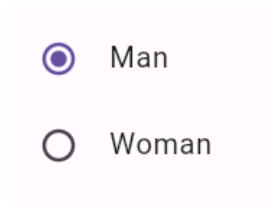
1. **value** ค่าปัจจุบันของ Radio button ใช้ในการกำหนดค่าเริ่มต้นของ Radio button
2. **groupValue** ค่าของ Radio button ที่ถูกเลือกในกลุ่มทุก Radio button ในกลุ่มควรมี groupValue เดียวกัน
3. **onChanged** ฟังก์ชัน callback ที่เรียกเมื่อมีการเปลี่ยนแปลงในสถานะของ Radio button

ตัวอย่างการใช้งาน Radio button

```
class _RadioExampleState extends State<RadioExample> {
  String sex = sexoptions[0];
  @override
  Widget build(BuildContext context) {
    return Column(
      children: <Widget>[
        ListTile(
          title: const Text('Man'),
          leading: Radio(
            value: sexoptions[0],
            groupValue: sex,
            onChanged: (value) {
              setState(() {
                sex = value.toString();
              });
            },
          ),
        ),
      ],
    ),
  ),
}
```

```
ListTile(
  title: const Text('Woman'),
  leading: Radio(
    value: sexoptions[1],
    groupValue: sex,
    onChanged: (value) {
      setState(() {
        sex = value.toString();
      });
    },
  ),
),
],
);
}
```

ผลลัพธ์



จากตัวอย่างเป็นการสร้าง Radio button ที่มีตัวเลือก Man, Woman ที่สามารถเลือกได้เลือก 1 ตัวเลือก

3.3.2 Checkbox

Checkbox คือ ส่วนหนึ่งของ Widget จะใช้เมื่อถ้าผู้สร้างเว็บไซต์ต้องการให้ผู้เยี่ยมชมเว็บไซต์ กรอกข้อมูลบางอย่าง และทำการเลือกข้อมูลโดยกดเครื่องหมายถูกที่ตัวใดตัวหนึ่ง หรือ ตัวเลือกมากกว่านั้น ทำให้เลือกได้หลายรายการ Checkbox จะมาในรูปแบบของกล่องสี่เหลี่ยม ด้านในเป็นพื้นที่ว่าง เพื่อให้ผู้ใช้ กดเครื่องหมายถูก เพื่อเลือกสิ่งที่เราต้องการหรือเลือกได้มากกว่านั้น โดยไม่จำเป็นว่าจะต้องเลือกแค่อย่างเดียว

คุณสมบัติ (Property) ที่สำคัญที่สำคัญของ Checkbox

1. **Value** ค่าปัจจุบันของ Checkbox ใช้ในการกำหนดค่าเริ่มต้นของ Checkbox
2. **onChanged** ฟังก์ชัน callback ที่จะถูกเรียกเมื่อมีการเปลี่ยนแปลงในสถานะของ Checkbox
3. **activeColor** สีที่ใช้เป็นพื้นหลังของ Checkbox เมื่อถูกเลือก
4. **checkColor** สีที่ใช้เป็นสีของเครื่องหมายถูกภายใน Checkbox เมื่อถูกเลือก

ตัวอย่างการใช้งาน Checkbox

```
class _CheckboxExampleState extends
State<CheckboxExample> {
  bool isChecked = false;
  bool isChecked1 = false;
  bool isChecked2 = false;
  bool isChecked3 = false;
  @override
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.start,
      children: [
        Row(
          children: [
            Checkbox(
              checkColor: Colors.red,
              value: isChecked,
              onChanged:(bool? value) {
                setState() {
                  isChecked = value!;
                });
              },),Text('ไทย'),
          ],
        ),
      ],
    ),
  ),
}
```

```

Row(
  children: [
    Checkbox(
      checkColor: Colors.red,
      value: isChecked1,
      onChanged: (bool? value) {
        setState(() {
          isChecked1 = value!;
        });
      },),Text('อังกฤษ'),
  ],),
Row(
  children: [
    Checkbox(
      checkColor: Colors.red,
      value: isChecked2,
      onChanged: (bool? value) {
        setState(() {
          isChecked2 = value!;
        });
      },),Text('จีน'),
  ],),

```

```

Row(
  children: [
    Checkbox(
      checkColor: Colors.red,
      value: isChecked1,
      onChanged: (bool? value) {
        setState(() {
          isChecked1 = value!;
        });
      },),Text('อังกฤษ'),
  ],),
Row(
  children: [
    Checkbox(
      checkColor: Colors.red,
      value: isChecked2,
      onChanged: (bool? value) {
        setState(() {
          isChecked2 = value!;
        });
      },),Text('จีน'),
  ],),

```

ผลลัพธ์

ภาษาที่พูดได้

- ☐ ไทย
- ☐ อังกฤษ
- ☐ จีน
- ☐ ญี่ปุ่น

จากตัวอย่างเป็นการสร้าง Checkbox ที่เครื่องหมายถูกเป็นสีขาว สีพื้นหลังเมื่อมีเครื่องหมายถูกเป็นสีเขียว เมื่อกดที่กล่องสี่เหลี่ยมจะเป็นการติ๊กเครื่องหมายถูกหากกดซ้ำจะเป็นการนำออก

3.3.3 Slider

Slider คือ Widget ที่ช่วยให้ผู้ใช้สามารถเลื่อนค่าในช่วงหนึ่ง (range) โดยการเลื่อนตัวบ่งบอกหรือลูกศรบนบาร์หรือสไลด์เดอร์ (slider) ตามขอบเขตที่กำหนด Slider ประกอบไปด้วย

คุณสมบัติ (Property) ที่สำคัญที่สำคัญของ Slider

1. **Value** ค่าปัจจุบันของ Slider ใช้ในการกำหนดค่าเริ่มต้นของ Slider

2. **Max** ค่าสูงสุดของ Slider

3. **min** ค่าน้อยสุดของ Slider

4. **divisions** แบ่งช่วงค่าของ Slider เป็นส่วนเท่า ๆ กัน

5. **onChange** ฟังก์ชัน callback ที่จะถูกเรียกเมื่อผู้ใช้ปล่อยการคลิกที่ Slider (หลังจากที่คลิกแล้วปล่อยมือ) ฟังก์ชันนี้จะรับค่าที่เป็น double

ตัวอย่างการใช้ Slider

```
class _SliderExampleState extends
State<SliderExample> {
  double _currentSliderValue = 1;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(title: const Text('Slider')),
```

```
body: Slider(
  value: _currentSliderValue,
  min: 0,
  max: 100,
  divisions: 5,
  onChanged: (double value) {
    setState(() {
      _currentSliderValue = value;
    });
  },
),
);
}
}
```

ผลลัพธ์



จากตัวอย่างเป็นการสร้าง Slider โดยที่ค่าสูงสุดของ Slider ในตัวอย่างคือ 100 และแบ่งช่วงค่าเป็น 5 ช่วง

3.3.4 Switch

Switch คือ Widget ที่ให้ผู้ใช้สามารถเปิด-ปิด หรือสลับสถานะระหว่างสองสถานะได้ (on/off, true/false) โดยมีการใช้ตัวเลื่อนที่สลับไปมาระหว่างสองสถานะ

คุณสมบัติ (Property) ที่สำคัญที่สำคัญของ Switch

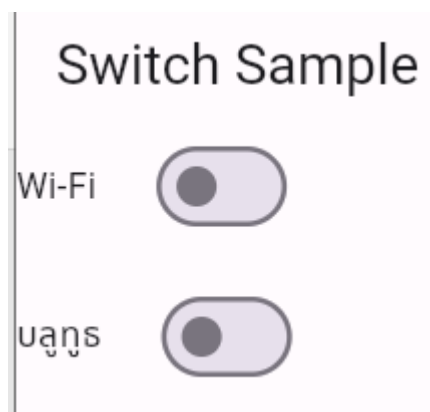
1. **Value** ค่าปัจจุบันของ Switch ใช้ในการกำหนดค่าเริ่มต้นของ Switch
2. **onChanged** ฟังก์ชัน callback ที่จะถูกเรียกเมื่อมีการเปลี่ยนแปลงค่าของ Switch
3. **activeColor** สีที่ใช้เป็นพื้นหลังของ Switch เมื่อถูกเลือก

ตัวอย่างการใช้ Switch

```
class _SwitchExampleState extends State<SwitchExample> {
  bool switch1 = false;
  bool switch2 = false;
  @override
  Widget build(BuildContext context) {
    return Column(
      mainAxisAlignment: MainAxisAlignment.start,
      children: [
        Row(
          children: [
            Text('Wi-Fi'),
            SizedBox(width: 20,),
            Switch(
              value: switch1,
              onChanged:(bool value) {
                setState(() {
                  switch1 = value;
                });
              },
            ),
          ],
        ),
      ],
    ),
  ),
}
```

```
SizedBox(height: 20),
Row(
  children: [
    Text('บลูทูธ'),
    SizedBox(width: 20),
    Switch(
      value: switch2,
      onChanged:(bool value) {
        setState(() {
          switch2 = value;
        });
      },
    ),
  ],
),
];
);
}
```

ผลลัพธ์



จากตัวอย่างเป็นการสร้าง Switch เมื่อกด on Switch จะมีลักษณะเป็นสีแดง

3.3.5 AlertDialog

AlertDialog คือ Widget ที่ให้แสดงหน้าต่างข้อความแจ้งเตือน (alert) ที่มีตัวอักษรสำหรับผู้ใช้เพื่อทำให้สามารถทราบข้อมูลหรือดำเนินการตามคำแนะนำที่ระบุได้ AlertDialog มักจะใช้เพื่อแจ้งเตือนข้อความที่สำคัญหรือในกรณีที่ผู้ใช้ต้องตอบกลับด้วยการกดปุ่ม

คุณสมบัติ (Property) ที่สำคัญที่สำคัญของ AlertDialog

1. **title** ใช้แสดงข้อความหัวเรื่องของ AlertDialog
2. **Content** ใช้แสดงเนื้อหาหรือข้อความใน AlertDialog
3. **Actions** ลิสต์ของ Widget ที่ใช้เพื่อกำหนดปุ่ม (action) ที่จะแสดงในด้านล่างของ AlertDialog สามารถใส่ปุ่มต่าง ๆ เพื่อให้ผู้ใช้ทำตามที่ต้องการ

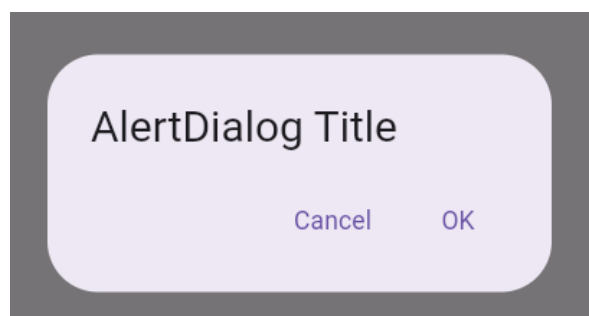
ตัวอย่างการใช้ AlertDialog

```
class DialogExample extends StatelessWidget {
  @override
  Widget build(BuildContext context) {
    return OutlinedButton(
      onPressed: () {
        showDialog(
          context: context,
          builder: (BuildContext context) {
            return AlertDialog(
              title: Text('AlertDialog Title'),
              actions: <Widget>[
                TextButton(
                  onPressed: () {
                    Navigator.pop(context, 'Cancel');
                  },

```

```
child: const Text('Cancel'),
                ),
                TextButton(
                  onPressed: () {
                    Navigator.pop(context, 'OK');
                  },
                  child: Text('OK'),
                ),
              ],
            );
          },
        );
      },
      child: const Text('Show Dialog'),
    );
  }
}
```

ผลลัพธ์



จากตัวอย่างสร้าง Text Button ชื่อ Show Dialog ขึ้นและใน Text Button ได้สร้าง AlertDialog เมื่อกดที่ Show Dialog จะมีหน้าต่างแจ้งเตือนขึ้นมาตามรูปในตัวอย่าง

3.3.6 TextField

TextField คือ Widget ที่ให้ผู้ใช้ป้อนข้อมูลผ่านทางอินพุตข้อความ ซึ่งสามารถใช้ในการรับข้อมูลจากผู้ใช้ได้หลากหลายรูปแบบ เช่น รับชื่อผู้ใช้, รับรหัสผ่าน, หรือข้อความอื่น ๆ ที่ต้องการจากผู้ใช้

คุณสมบัติ (Property) ที่สำคัญที่สำคัญของ TextField

1. Decoration InputDecoration ที่ใช้กำหนดรูปแบบและลักษณะของ TextField, เช่น ข้อความแนะนำ (hint text), พื้นหลัง, และขอบ
2. keyboardType ประเภทของแป้นพิมพ์ที่จะแสดง, เช่น text, number, email, และอื่น ๆ
3. onChanged ฟังก์ชัน callback ที่จะถูกเรียกเมื่อมีการเปลี่ยนแปลงในข้อความที่ป้อน
4. onSubmitted ฟังก์ชัน callback ที่จะถูกเรียกเมื่อผู้ใช้กดปุ่ม "Enter" หรือเสร็จสิ้นการป้อนข้อมูล
5. controller เป็นทางเลือกในการจัดการข้อความของ Text Field เพื่อการอ่านหรือการกำหนดค่าของ Text Field ค่าที่จะกำหนดให้แก่พารามิเตอร์ controller ต้องสร้างจาก class TextEditingController

ตัวอย่างการใช้ TextField

```
class _MyHomePageState extends State<MyHomePage> {
  TextEditingController _cmController = TextEditingController();
  String _inchesResult = "";
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text('test'),
      ),
      body: Padding(
        padding: const EdgeInsets.all(16.0),
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: [
            TextField(
              controller: _cmController,
              keyboardType: TextInputType.number,
              decoration: InputDecoration(
                labelText: 'ป้อนค่าเซนติเมตรหรือนิ้ว',
                border: OutlineInputBorder(),
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

```

    SizedBox(height: 16.0),
    ElevatedButton(
      onPressed: () {
        double cmValue = double.parse(_cmController.text);
        double inchesResult = cmValue / 2.54;
        setState(() {
          _inchesResult = '$cmValue เซนติเมตร = ${inchesResult.toStringAsFixed(2)} นิ้ว';
        });
      },
      child: Text('นิ้ว'),
    ),
    ElevatedButton(
      onPressed: () {
        double cmValue = double.parse(_cmController.text);
        double inchesResult = cmValue * 2.54;
        setState(() {
          _inchesResult = '$cmValue นิ้ว = ${inchesResult.toStringAsFixed(2)} เซนติเมตร';
        });
      },
      child: Text('เซนติเมตร'),
    ),
    SizedBox(height: 16.0),
    Text(_inchesResult),
  ],
),
);
}
}

```

ป้อนค่าเซนติเมตรหรือนิ้ว

12

นิ้ว

เซนติเมตร

12 นิ้ว = 30.48 เซนติเมตร

จากตัวอย่างสร้าง Text Field ที่ป้อนข้อมูลขึ้นและเมื่อมีการกด Enter จะมีหน้าต่างแจ้งเตือนดังขึ้นตามภาพ

3. เครื่องมือและอุปกรณ์การทดลอง

- 3.1 คอมพิวเตอร์ 1 เครื่อง
- 3.2 ใบงานที่ 7 เรื่อง Widgets

4. ลำดับขั้นตอนการทดลอง

- 4.1 นักเรียนศึกษาเนื้อหา เรื่อง คำสั่งเลือกเงื่อนไข
- 4.2 ให้นักเรียนตอบคำถาม ลงในใบงานที่ 7
- 4.3 ส่งงานครูหลังจากเสร็จเรียบร้อยแล้ว

คำสั่ง ให้นักศึกษาเขียนคำตอบตามที่โจทย์กำหนดให้ถูกต้อง (สามารถแนบรูปโค้ดและผลลัพธ์คำตอบของโปรแกรมได้)

1. ใช้ Radio Button, Checkbox, Slider, Switch, AlertDialog, TextField ให้เป็นตามหน้าจอตามตัวอย่าง โดยประกอบไปด้วยส่วนที่สำคัญดังนี้

คอลัมน์ที่ 1

- 1.1) กล่องใส่ชื่อ-นามสกุล
- 1.2) ตัวเลือกเพศชายหรือหญิง
- 1.3) ตัวเลือกSecที่เรียน
- 1.4) เลือกอายุ
- 1.5) เลือกความถนัดในภาษาคอมพิวเตอร์
- 1.6) ปุ่มกดบันทึกข้อมูล

คอลัมน์ที่ 2

แสดงข้อมูลที่เรากرอกทั้งหมดใน คอลัมน์ที่ 1

ข้อ 1

Enter Name
Ninja

ชาย ☒ หญิง

☒ Sec182 ☐ Sec307

อายุ
21

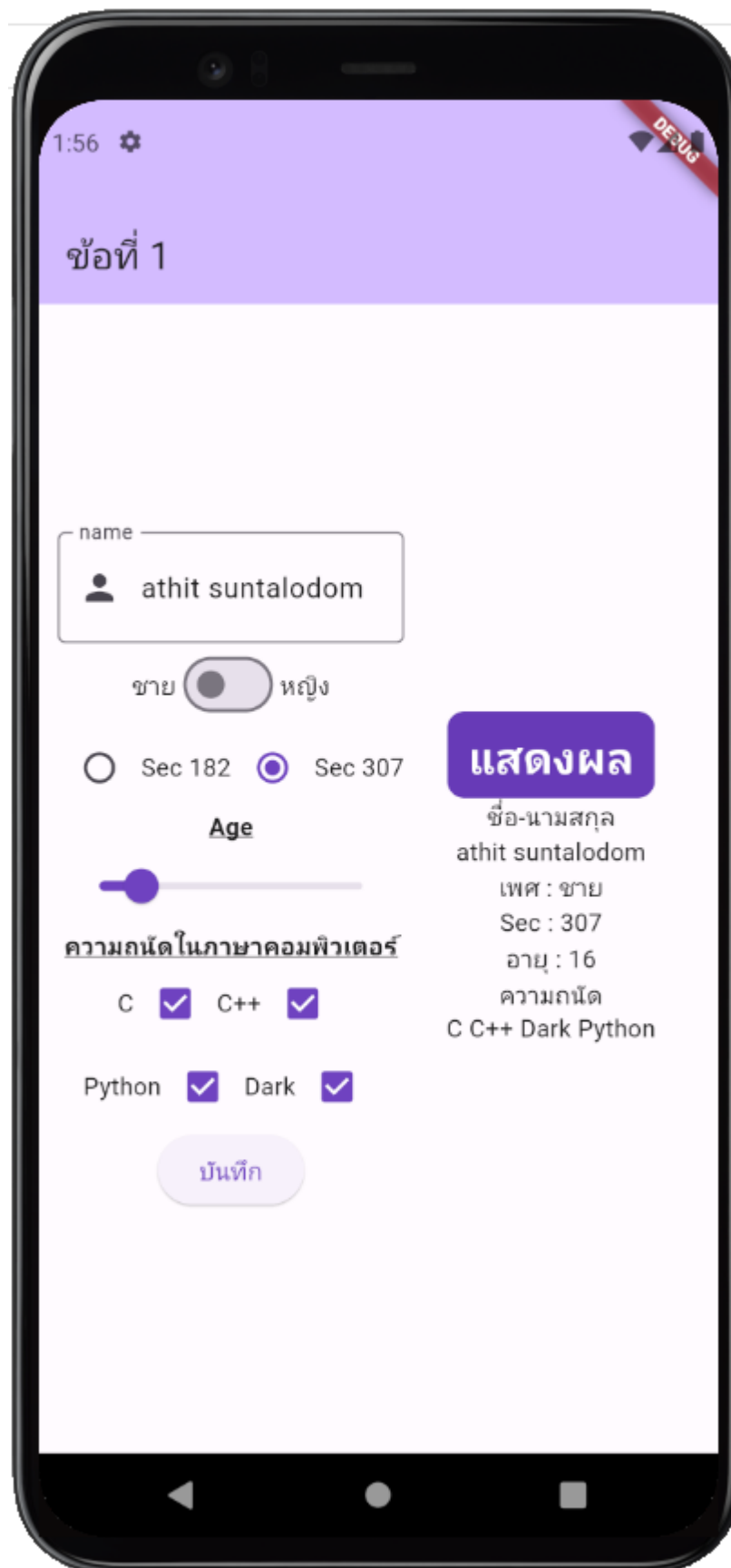
ความถนัดในภาษาคอมพิวเตอร์

☒ C ☒ C++
☒ Python ☐ Dart

บันทึก

แสดงผล

ชื่อ-นามสกุล : Ninja
เพศ : ชาย
Sec : Sec182
อายุ : 21
ความถนัด : C C++ Python



```

lib > main.dart > ...
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: 'Flutter Demo',
15        theme: ThemeData(
16          colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
17          useMaterial3: true,
18        ), // ThemeData
19        home: const MyHomePage(title: 'หน้า 1'),
20      ); // MaterialApp
21    }
22  }
23
24  class MyHomePage extends StatefulWidget {
25    const MyHomePage({super.key, required this.title});
26
27    final String title;
28
29    @override
30    State<MyHomePage> createState() => _MyHomePageState();
31  }
32
33  class _MyHomePageState extends State<MyHomePage> {
34    TextEditingController _cmController = TextEditingController();
35    String _showName = "name";
36    String sexUser = "";
37    bool sex = false;
38    List<String> sec = ['182', '307'];
39    String secUser = "";
40    double ageSlider = 1;
41    bool checkAptitudeC = false;
42    bool checkAptitudeCplus = false;
43    bool checkAptitudePy = false;
44    bool checkAptitudeDa = false;
45    String scheckAptitudeC = "";
46    String scheckAptitudeCplus = "";
47    String scheckAptitudePy = "";
48    String scheckAptitudeDa = "";
49
50    @override
51    Widget build(BuildContext context) {
52      return Scaffold(
53        appBar: AppBar(
54          backgroundColor: Theme.of(context).colorScheme.inversePrimary,
55          title: Text(widget.title),
56        ), // AppBar
57        body: Center(

```

```

57   child: Row(
58     crossAxisAlignment: CrossAxisAlignment.center,
59     mainAxisAlignment: MainAxisAlignment.center,
60     children: [
61       Container(
62         height: 400,
63         width: 200,
64         child: Column(
65           children: [
66             TextField(
67               controller: _cmController,
68               keyboardType: TextInputType.name,
69               decoration: const InputDecoration(
70                 prefixIcon: Icon(Icons.person),
71                 labelText: 'name',
72                 border: OutlineInputBorder(),
73               ), // InputDecoration
74             ), // TextField
75             Container(
76               child: Row(
77                 mainAxisAlignment: MainAxisAlignment.center,
78                 children: [
79                   const Text('ชาย'),
80                   Switch(
81                     value: sex,
82                     onChanged: (bool value) {
83                       setState(() {
84                         sex = value;
85                         sexUser = value == true ? "หญิง" : "ชาย";
86                       });
87                     },
88                   ), // Switch
89                   const Text('หญิง'),
90                 ],
91               ), // Row
92             ), // Container
93             Container(
94               child: Row(
95                 mainAxisAlignment: MainAxisAlignment.spaceBetween,
96                 children: [
97                   Radio(
98                     value: sec[0],
99                     groupValue: secUser,
100                    onChanged: (value) {
101                      setState(() {
102                        secUser = value.toString();
103                      });
104                    },
105                  ), // Radio
106                  const Text('Sec 182'),
107                  Radio(
108                    value: sec[1],
109                    groupValue: secUser,
110                    onChanged: (value) {
111                      setState(() {
112                        secUser = value.toString();

```

```

114         },
115     ), // Radio
116     const Text('Sec 307'),
117 ],
118 ), // Row
119 ), // Container
120 const Text(
121   'Age',
122   style: TextStyle(
123     fontWeight: FontWeight.bold,
124     decoration: TextDecoration.underline,
125   ), // TextStyle
126 ), // Text
127 Slider(
128   value: ageSlider,
129   min: 0,
130   max: 100,
131   divisions: 100,
132   onChanged: (double value) {
133     setState(() {
134       ageSlider = value;
135     });
136   },
137 ), // Slider
138 const Text(
139   'ความถนัดในภาษาคอมพิวเตอร์',
140   style: TextStyle(
141     fontWeight: FontWeight.bold,
142     decoration: TextDecoration.underline,
143   ), // TextStyle
144 ), // Text
145 Container(
146   child: Row(
147     mainAxisAlignment: MainAxisAlignment.center,
148     children: [
149       const Text(
150         'C',
151       ), // Text
152       Checkbox(
153         value: checkAptitudeC,
154         onChanged: (bool? value) {
155           setState(() {
156             checkAptitudeC = value!;
157             scheckAptitudeC = value == true ? "C" : "";
158           });
159         },
160       ), // Checkbox
161       const Text(
162         'C++',
163       ), // Text
164       Checkbox(
165         value: checkAptitudeCplus,
166         onChanged: (bool? value) {
167           setState(() {
168             checkAptitudeCplus = value!;
169             scheckAptitudeCplus = value == true ? "C++" : "";
170           });

```

```

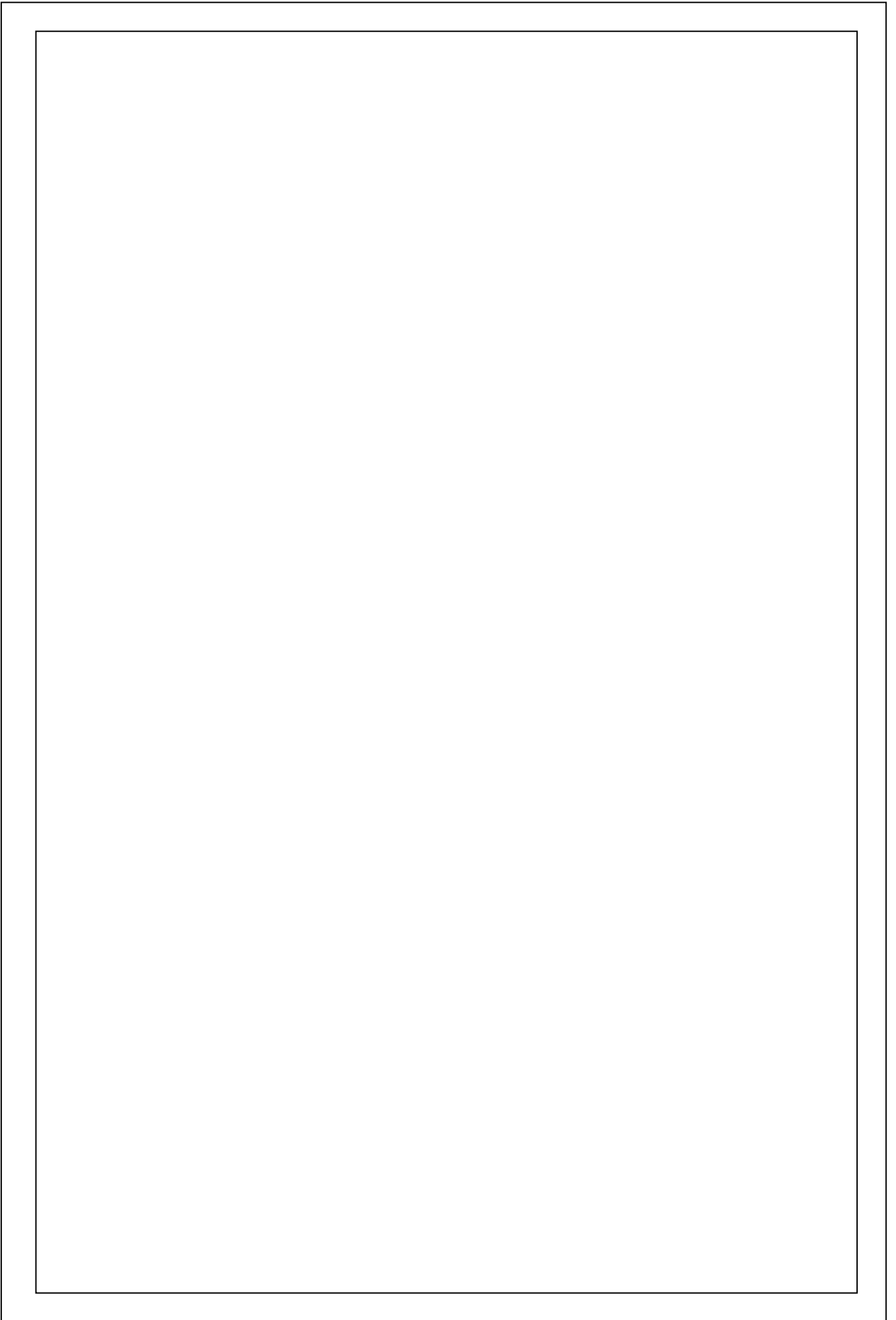
171         },
172       ), // Checkbox
173     ],
174   ), // Row
175 ), // Container
176 Container(
177   child: Row(
178     mainAxisAlignment: MainAxisAlignment.center,
179     children: [
180       const Text(
181         'Python',
182       ), // Text
183       Checkbox(
184         value: checkAptitudePy,
185         onChanged: (bool? value) {
186           setState(() {
187             checkAptitudePy = value!;
188             scheckAptitudePy = value == true ? "Python" : "";
189           });
190         },
191       ), // Checkbox
192       const Text(
193         'Dark',
194       ), // Text
195       Checkbox(
196         value: checkAptitudeDa,
197         onChanged: (bool? value) {
198           setState(() {
199             checkAptitudeDa = value!;
200             scheckAptitudeDa = value == true ? "Dark" : "";
201           });
202         },
203       ), // Checkbox
204     ],
205   ), // Row
206 ), // Container
207 ElevatedButton(
208   onPressed: () {
209     setState(() {
210       _showName = _cmController.text;
211     });
212   },
213   child: const Text('บันทึก'),
214 ), // ElevatedButton
215 ],
216 ), // Column
217 ), // Container
218 Container(
219   height: 400,
220   width: 170,
221   child: Column(
222     crossAxisAlignment: CrossAxisAlignment.center,
223     mainAxisAlignment: MainAxisAlignment.center,
224     children: [
225       Container(
226         height: 50,
227         width: 120,

```

```

228     decoration: BoxDecoration(
229       color: Colors.deepPurple,
230       borderRadius: BorderRadius.circular(10),
231     ), // BoxDecoration
232     child: const Center(
233       child: Text(
234         'แสดงผล',
235         style: TextStyle(
236           fontSize: 25,
237           fontWeight: FontWeight.bold,
238           color: Colors.white,
239           fontFamily: 'Arial',
240         ), // TextStyle
241       ), // Text
242     ), // Center
243   ), // Container
244   Container(
245     child: Row(
246       mainAxisAlignment: MainAxisAlignment.center,
247       crossAxisAlignment: CrossAxisAlignment.center,
248       children: [
249         Container(
250           child: Column(
251             children: [
252               const Text('ชื่อ-นามสกุล'),
253               Text(_showName),
254               Text('เพศ : $sexUser'),
255               Text('Sec : $secUser'),
256               Text('อายุ : ${ageSlider.toStringAsFixed(0)}'),
257               const Text('ความถนัด'),
258               Text(
259                 '$scheckAptitudeC $scheckAptitudeCplus $scheckAptitudeDa $scheckAptitudePy', // Text
260               ),
261             ], // Column
262           ), // Container
263         ], // Row
264       ), // Container
265     ), // Column
266   ), // Center
267 ); // Scaffold
272 }
273
274

```

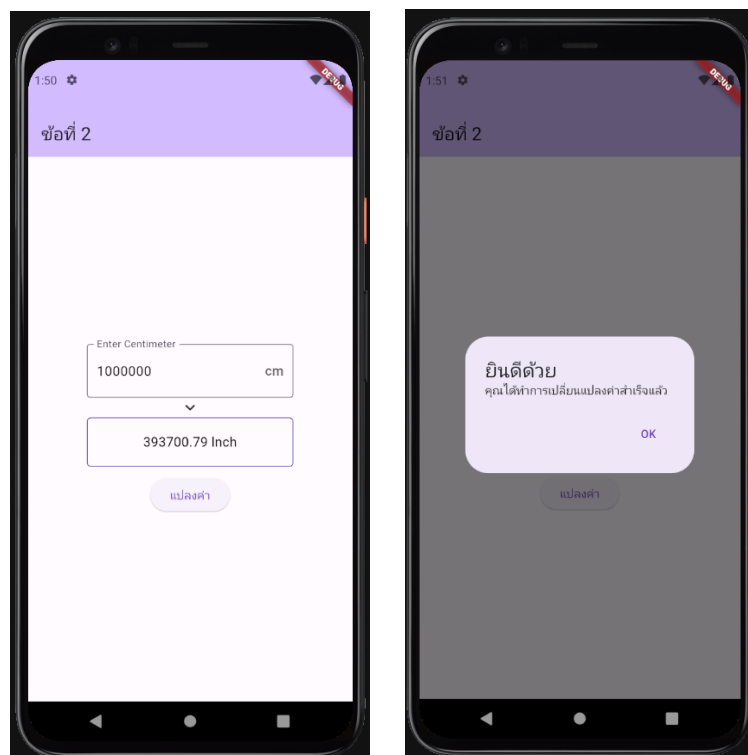
2. สร้างแอปพลิเคชันที่แปลงค่าจาก เซนติเมตร เป็น นิ้ว โดยจะต้อง กรอกค่า เซนติเมตรลงไป และกดแปลงเพื่อแสดงค่าที่แปลงจากเซนติเมตรเป็นนิ้วออกมา พร้อมทั้งแสดงหน้า AlertDialog ว่า “แปลงสำเร็จ”

ข้อ 2

The screenshot shows a mobile application interface with a light yellow background. At the top, there is a green header bar with the text "ข้อ 2". Below the header, there is a text input field with the placeholder text "Enter Centimeter". The input field contains the number "20". To the right of the input field, there is a unit selector dropdown menu showing "cm". Below the input field, there is a green button with a downward arrow. Below the button, there is a green rounded rectangle displaying the converted value "7.86 inch". At the bottom, there is a green button labeled "แปลงค่า".

ข้อ 2

The screenshot shows the same mobile application interface as the previous one, but with a dark gray background. A green alert dialog is displayed in the center. The dialog has a title "Alert" and a message "เปลี่ยนแปลงค่าสำเร็จ". Below the dialog, there is a green button labeled "แปลงค่า".



```

lib > main.dart > _MyHomePageState > build
1  import 'package:flutter/material.dart';
2
   Run | Debug | Profile
3  void main() {
4    runApp(const MyApp());
5  }
6
7  class MyApp extends StatelessWidget {
8    const MyApp({super.key});
9
10   @override
11   Widget build(BuildContext context) {
12     return MaterialApp(
13       title: 'Flutter Demo',
14       theme: ThemeData(
15         colorScheme: ColorScheme.fromSeed(seedColor: Colors.deepPurple),
16         useMaterial3: true,
17       ), // ThemeData
18       home: const MyHomePage(title: 'หน้า 2'),
19     ); // MaterialApp
20   }
21 }
22
23 class MyHomePage extends StatefulWidget {
24   const MyHomePage({super.key, required this.title});
25
26   final String title;
27
28   @override
29   State<MyHomePage> createState() => _MyHomePageState();
30 }
31
32 class _MyHomePageState extends State<MyHomePage> {
33   final numCm = TextEditingController();
34   String tochange = "0";
35   @override
36   Widget build(BuildContext context) {
37     return Scaffold(
38       appBar: AppBar(
39         backgroundColor: Theme.of(context).colorScheme.inversePrimary,
40         title: Text(widget.title),
41       ), // AppBar
42       body: Center(
43         child: Container(
44           width: 250,
45           child: Column(
46             mainAxisAlignment: MainAxisAlignment.center,
47             children: [
48               TextField(
49                 controller: numCm,
50                 keyboardType: TextInputType.number,
51                 decoration: InputDecoration(
52                   labelText: 'Enter Centimeter',
53                   suffixText: 'cm',
54                   border: OutlineInputBorder(),
55                 ), // InputDecoration
56               ), // TextField

```

```

57     const Icon(Icons.keyboard_arrow_down),
58     Container(
59       width: 250,
60       height: 60,
61       padding: EdgeInsets.all(10),
62       decoration: BoxDecoration(
63         borderRadius: BorderRadius.circular(5),
64         border: Border.all(
65           color: Colors.deepPurple, // สีของเส้นกรอบ
66           width: 1.0, // ความกว้างของเส้นกรอบ
67         ), // Border.all
68       ), // BoxDecoration
69       alignment: Alignment.center,
70       child: Text(
71         '$tochange Inch',
72         textAlign: TextAlign.center,
73         style: TextStyle(
74           fontSize: 16,
75           color: Colors.black,
76           fontFamily: 'Arial',
77         ), // TextStyle
78       ), // Text
79     ), // Container
80     SizedBox(
81       height: 10,
82     ), // SizedBox
83     ElevatedButton(
84       onPressed: () {
85         double cmVa = double.parse(numCm.text);
86         double inch = cmVa / 2.54;
87         setState(() {
88           tochange = '${inch.toStringAsFixed(2)}';
89         });
90         showDialog(
91           context: context,
92           builder: (BuildContext context) {
93             return AlertDialog(
94               title: Column(
95                 crossAxisAlignment: CrossAxisAlignment.start,
96                 children: [
97                   Text('ยินดีด้วย'),
98                   Text('คุณได้ทำการแปลงค่าสำเร็จแล้ว',
99                     style: TextStyle(fontSize: 14)), // Text
100                 ],
101               ), // Column
102               actions: <Widget>[]
103               TextButton(
104                 onPressed: () {
105                   Navigator.pop(context, 'OK');
106                 },
107                 child: const Text('OK'),
108               ), // TextButton
109             ], // <Widget>[]
110           ); // AlertDialog
111       },

```

```
112         );  
113     },  
114     child: const Text('เพลงคำ'),  
115     ), // ElevatedButton  
116 ],  
117 ), // Column  
118 ), // Container  
119 ); // Center // Scaffold  
120 }  
121 }  
122
```

5. สรุปผลการทดลอง

สรุปผลการทดลองได้ดังนี้ **RadioButton widget** ใช้สำหรับสร้างปุ่มตัวเลือกแบบวิทย์ ผู้ใช้สามารถเลือกตัวเลือกได้เพียงตัวเลือกเดียวจากตัวเลือกต่างๆ **Checkbox widget** ใช้สำหรับสร้างปุ่มตัวเลือกแบบเช็คบ็อกซ์ ผู้ใช้สามารถเลือกตัวเลือกได้หลายตัวเลือกจกตัวเลือกต่างๆ **Slider.widget** ใช้สำหรับสร้างแถบเลื่อน ผู้ใช้สามารถเลื่อนแถบเลื่อนเพื่อเลือกค่าต่างๆ ได้ **Switch widget** ใช้สำหรับสร้างสวิตช์ ผู้ใช้สามารถเปิดหรือปิดสวิตช์ได้ **AlertDialog widget** ใช้สำหรับสร้างกล่องโต้ตอบแบบแจ้งเตือน กล่องโต้ตอบนี้มักใช้เพื่อแจ้งเตือนผู้ใช้เกี่ยวกับบางสิ่งบางอย่างหรือขออนุญาตจากผู้ใช้ **TextField widget** ใช้สำหรับสร้างช่องข้อความ ผู้ใช้สามารถป้อนข้อความลงในช่องข้อความได้

สรุป ผลการทดลองการใช้ widget ต่างๆ ใน Flutter สามารถกำหนดค่าต่างๆ ให้กับ widget ได้ และสามารถจับเหตุการณ์ต่างๆ ของ widget ได้

6. คำถามหลังการทดลอง

6.1 จากการทดลอง Radio button กับ Checkbox widget widget ใดมีความยืดหยุ่นมากกว่ากัน

RadioButton widget นั้นมีความยืดหยุ่นน้อยกว่า **Checkbox widget** เนื่องจาก **RadioButton widget** อนุญาตให้ผู้ใช้เลือกตัวเลือก.....
ได้เพียงตัวเลือกเดียวเท่านั้น ในทางกลับกัน **Checkbox widget** อนุญาตให้ผู้ใช้เลือกตัวเลือกได้หลายตัวเลือกพร้อมกัน.....
ทำให้มีความยืดหยุ่นมากกว่า.....