

Cours 7. Popup et Navigator dans Flutter

[O.R. Merad Boudia](#)

Université d'Oran 1, Ahmed Ben Bella

M1 GBM : 2023/2024

ElevatedButton

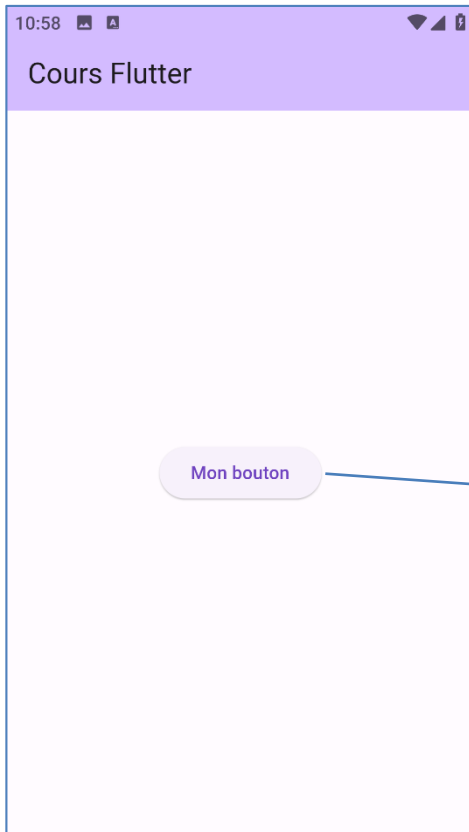
- ✓ Créer un nouveau projet.
- ✓ Enlever tous les commentaires ainsi que le body par défaut.
- ✓ Ajouter un nouveau Body.
- ✓ Créer une nouvelle classe « Body »
- ✓ Ajouter un **ElevatedButton**
- ✓ Une méthode pour la gestion du clic.
- ✓ Un texte sur le bouton

```
class _MyHomePageState extends State<MyHomePage> {  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        backgroundColor: Theme.of(context).colorScheme.inversePrimary,  
        title: Text(widget.title),  
      ), // AppBar  
      body: new Body(),  
    ); // Scaffold  
  }  
}
```

```
import 'package:flutter/material.dart';  
  
class Body extends StatefulWidget {  
  @override  
  _BodyState createState() => new _BodyState();  
}  
  
class _BodyState extends State<Body> {  
  
  @override  
  Widget build(BuildContext context) {  
    return Center(  
      child: ElevatedButton(  
        onPressed: pressed,  
        child: Text("Mon bouton"),  
      ), // ElevatedButton  
    ); // Center  
  }  
  
  void pressed() {  
    setState() {  
      print("Tu as appuyé sur le bouton");  
    }  
  }  
}
```

ElevatedButton

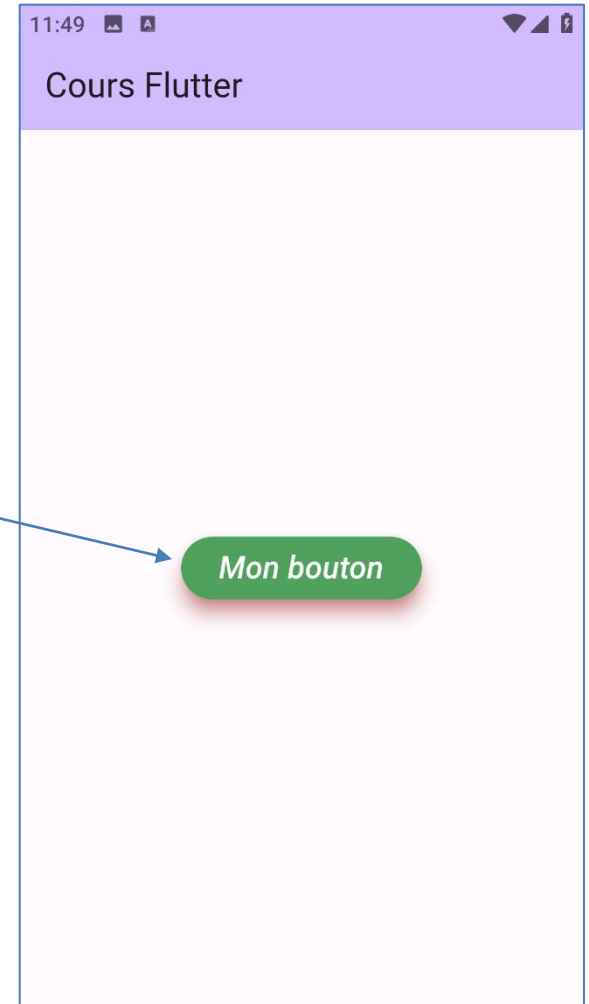
✓ Résultat:



```
main.dart (SM S901N) x
Console
↑ Installing build\app\outputs\flutter-apk\app-debug.apk...
↓ Debug service listening on ws://127.0.0.1:49850/wjJeELBTXus=/ws
Syncing files to device SM S901N...
D/EGL_adreno(10133): eglMakeCurrent: 0x7fff6c073760: ver 3 1 (tinfo 0x7fff54564cc0)
I/flutter (10133): Tu as appuyé sur le bouton
```

ElevatedButton

```
child: ElevatedButton(  
  style: ElevatedButton.styleFrom(  
    foregroundColor: Colors.white,  
    backgroundColor: Colors.teal,  
    shadowColor: Colors.red,  
    elevation: 10.0,  
  ),  
  onPressed: pressed,  
  child: const Text(  
    "Mon bouton",  
    style: TextStyle(fontStyle: FontStyle.italic, fontSize: 20.0),  
  ), // Text  
), // ElevatedButton
```



SnackBar

```
Widget build(BuildContext context) {
  return Center(
    child: ElevatedButton(
      style: ElevatedButton.styleFrom(
        foregroundColor: Colors.white,
        backgroundColor: Colors.green,
        shadowColor: Colors.red,
        elevation: 10.0,
      ),
      onPressed: snack,
      child: const Text(
        "Mon bouton",
        style: TextStyle(fontStyle: FontStyle.italic, fontSize: 20.0),
      ), // Text
    ), // ElevatedButton
  ); // Center
}

void snack() {
  SnackBar snackBar = const SnackBar(
    content: Text("Je suis une SnackBar"),
    duration: Duration(seconds: 5),
  ); // SnackBar
  ScaffoldMessenger.of(context).showSnackBar(snackBar);
}
```



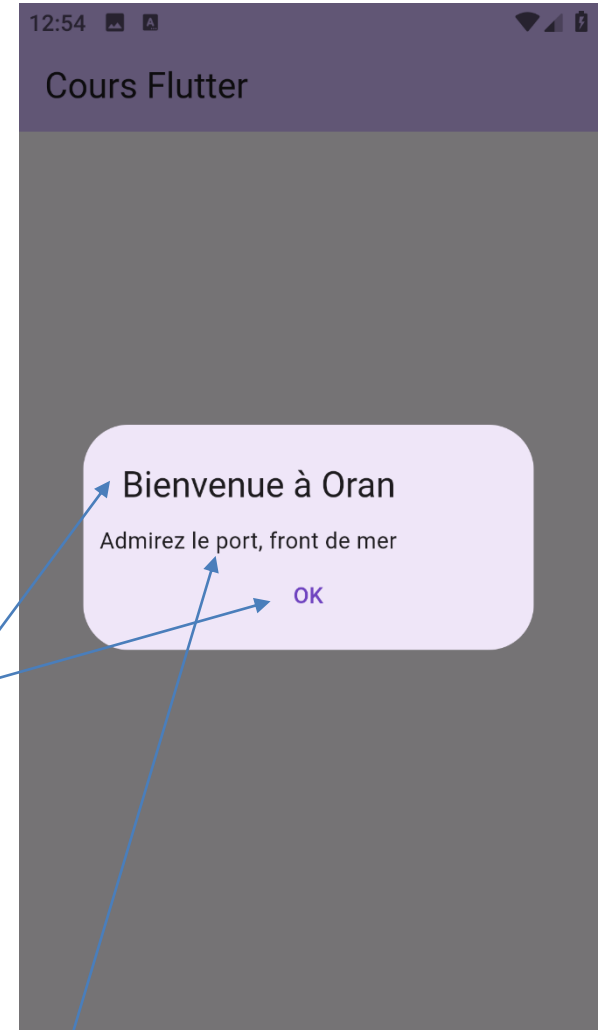
AlertDialog

```
Future<Null> alerte() async {
  return showDialog(
    context: context,
    barrierDismissible: false,
    builder: (BuildContext context) {
      return AlertDialog(
        title: const Text("Je suis une alerte"),
        content: const Text("Bonjour", textAlign: TextAlign.center),
        contentPadding: const EdgeInsets.all(5.0),
        actions: <Widget>[
          TextButton(
            onPressed: () {
              print("Abort");
              Navigator.pop(context);
            },
            child: const Text(
              "Annuler",
              style: TextStyle(color: Colors.red),
            ), // Text, TextButton
          TextButton(
            onPressed: () {
              print("Proceed");
              Navigator.pop(context);
            },
            child: const Text(
              "Continuer",
              style: TextStyle(color: Colors.blue),
            ), // Text, TextButton
          ], // <Widget>[]
        ); // AlertDialog
      });
    }
  }
}
```



SimpleDialog

```
Future<Null> dialog(String title, String desc) async {  
  return showDialog(  
    context: context,  
    barrierDismissible: true,  
    builder: (BuildContext context) {  
      return SimpleDialog(  
        title: Text(  
          title,  
          textScaleFactor: 1,  
        ), // Text  
        contentPadding: EdgeInsets.all(10.0),  
        children: [  
          Text(desc),  
          TextButton(  
            onPressed: () {  
              print("appuiyé");  
              Navigator.pop(context);  
            },  
            child: const Text("OK")) // TextButton  
        ],  
      ); // SimpleDialog  
    });  
}
```



```
onPressed: (() => dialog ("Bienvenue à Oran", "Admirez le port, front de mer")),
```

Allez vers une nouvelle page

- ✓ Créer un nouveau fichier **nouvelle_page.dart**:

```
import 'package:flutter/material.dart';

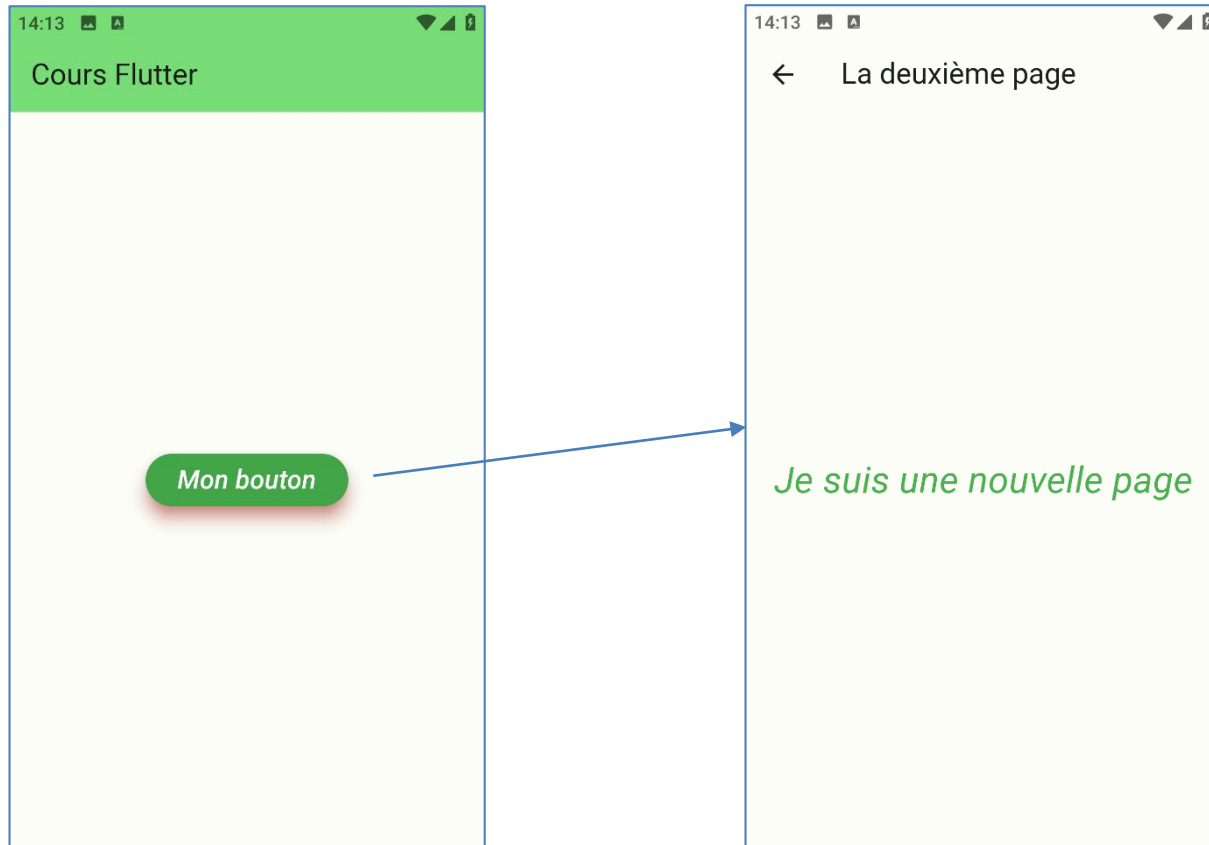
class NouvellePage extends StatelessWidget {
  String title = "";
  NouvellePage(String title) {
    this.title = title;
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: Text(title),
      ), // AppBar
      body: const Center(
        child: Text(
          "Je suis une nouvelle page",
          textScaleFactor: 2.0,
          textAlign: TextAlign.center,
          style: TextStyle(color: Colors.green, fontStyle: FontStyle.italic),
        ), // Text
      ), // Center
    ); // Scaffold
  }
}
```

- ✓ Retourner à **body.dart**:

```
void versNouvellePage() {
  Navigator.push(context, MaterialPageRoute(builder: (BuildContext context) {
    return NouvellePage("La deuxième page");
  })); // MaterialPageRoute
}
```


Allez vers une nouvelle page

✓ Résultat:



Les widgets interactifs

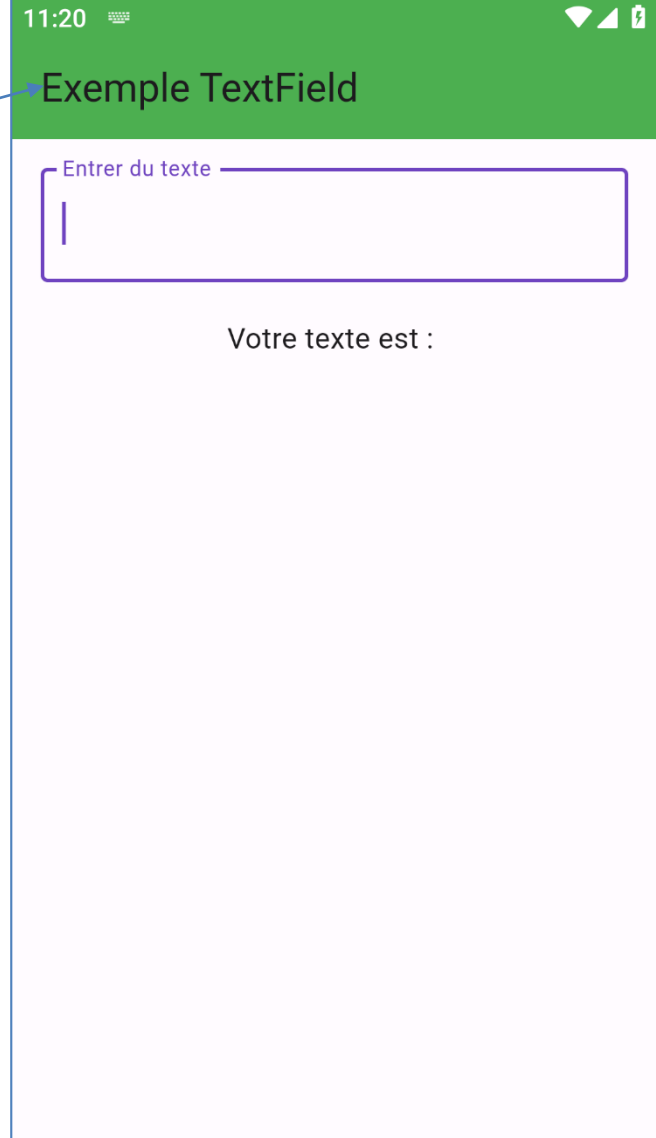
Le TextField

Pour gérer le contenu du **TextField**

```
class _MyHomePageState extends State<MyHomePage> {  
  final TextEditingController _controller = TextEditingController();  
  String _displayText = '';  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Exemple TextField'),  
        backgroundColor: Colors.green,  
      ), // AppBar  
      body: Padding(  
        padding: const EdgeInsets.all(32.0),  
        child: Column(  
          children: [  
            TextField(  
              controller: _controller,  
              onChanged: (text) {  
                setState() {  
                  _displayText = text;  
                }  
              });  
            decoration: const InputDecoration(  
              labelText: 'Entrez du texte',  
              border: OutlineInputBorder(),  
            ), // InputDecoration, TextField  
            const SizedBox(height: 20),  
            Text(  
              'Votre texte est : $_displayText',  
              style: const TextStyle(fontSize: 16),  
            ), ], ), ), ); } // Text, Column, Padding, Scaffold
```

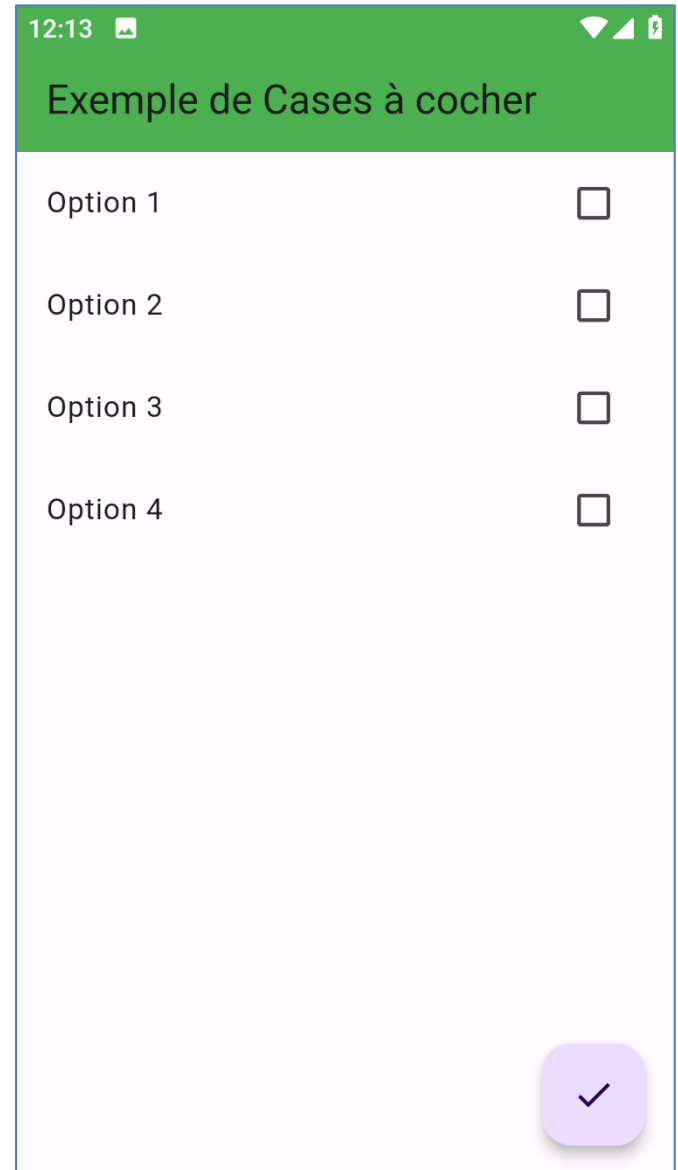
Pour stocker le texte saisi

Lorsque le texte change, la fonction **onChanged** est appelée, mettant à jour **_displayText** et provoquant une mise à jour de l'interface.



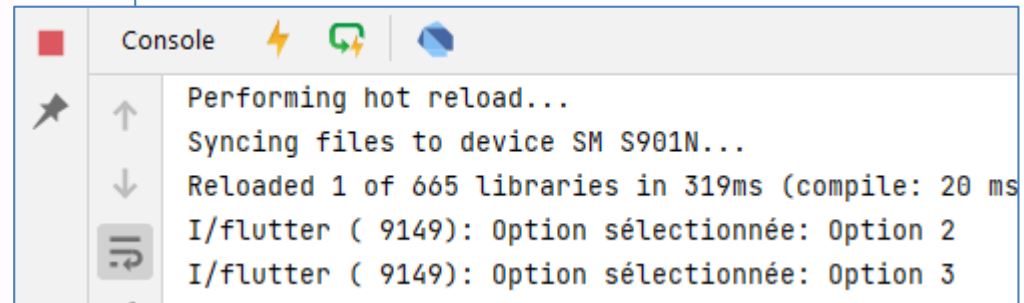
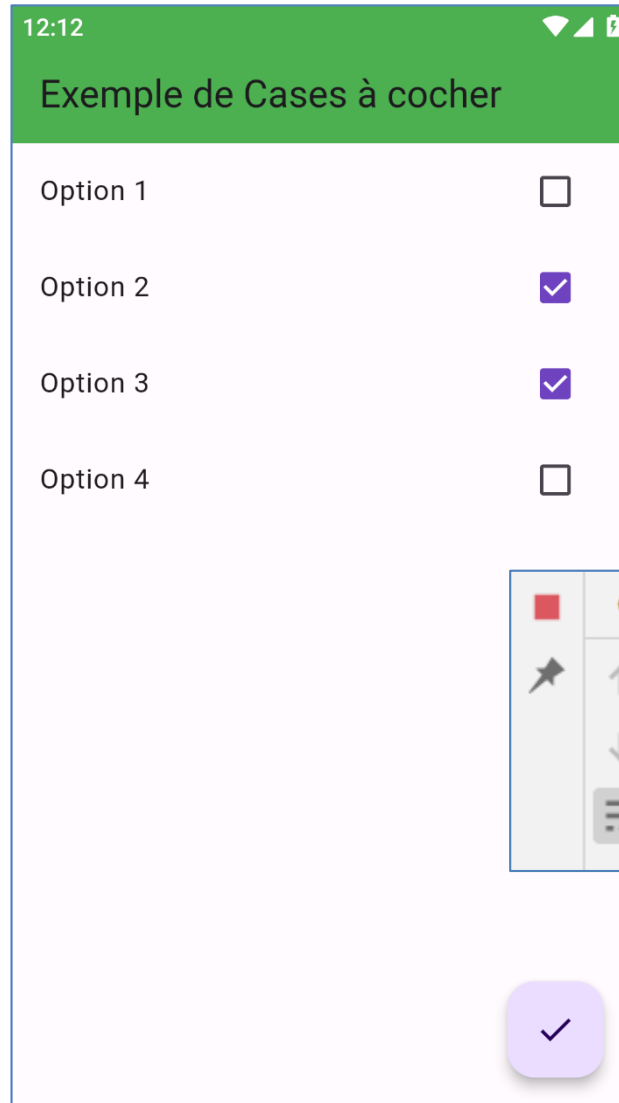
CheckBox

```
class _MyHomePageState extends State<MyHomePage> {  
  // Liste de noms pour nos cases à cocher  
  final List<String> _options = ['Option 1', 'Option 2', 'Option 3', 'Option 4'];  
  // Liste pour stocker l'état des cases à cocher  
  final List<bool> _selectedOptions = [false, false, false, false];  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Exemple de Cases à cocher'),  
        backgroundColor: Colors.green,  
      ), // AppBar  
      body: ListView.builder(  
        itemCount: _options.length,  
        itemBuilder: (BuildContext context, int index) {  
          return CheckboxListTile(  
            title: Text(_options[index]),  
            value: _selectedOptions[index],  
            onChanged: (bool? value) {  
              setState(() {  
                _selectedOptions[index] = value!;  
              }); }, ); }, ), // CheckboxListTile, ListView.builder  
      floatingActionButton: FloatingActionButton(  
        onPressed: () {  
          // Affiche les options sélectionnées dans la console  
          for (int i = 0; i < _options.length; i++) {  
            if (_selectedOptions[i]) {  
              print('Option sélectionnée: ${_options[i]}');  
            }  
          }  
        },  
        child: const Icon(Icons.check),  
      ), ); } // FloatingActionButton, Scaffold  
    }  
  }  
}
```



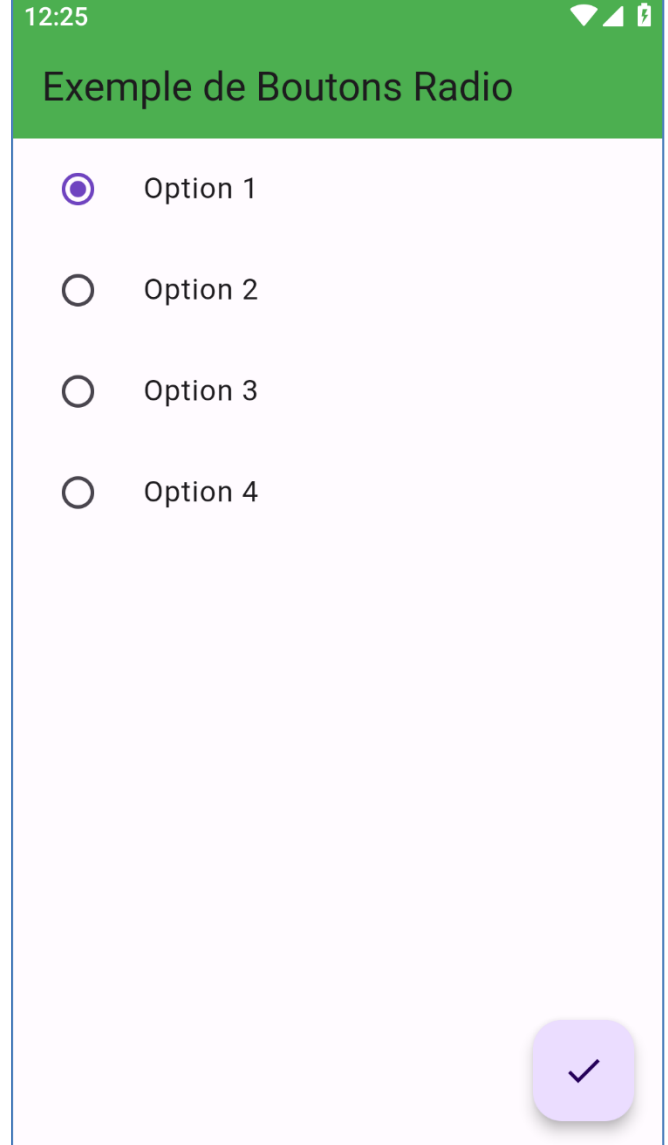
CheckBox

☐ Résultat:



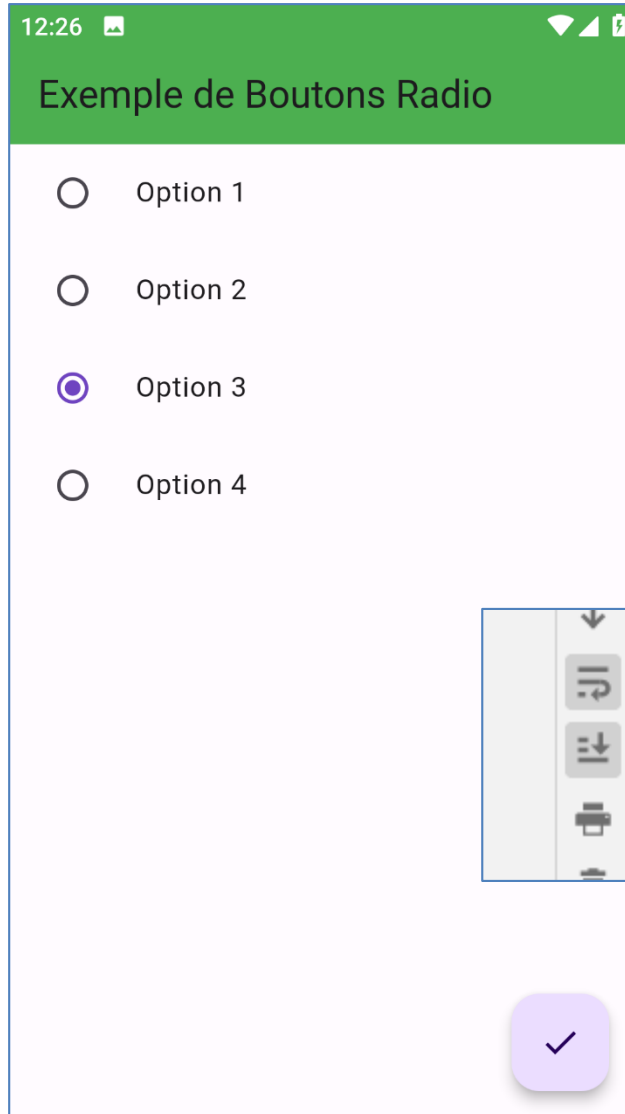
Boutons Radio

```
class _MyHomePageState extends State<MyHomePage> {  
  // Liste de noms pour nos cases à cocher  
  final List<String> _options = ['Option 1', 'Option 2', 'Option 3', 'Option 4'];  
  // Index de l'option sélectionnée  
  int _selectedOption = 0;  
  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Exemple de Boutons Radio'),  
        backgroundColor: Colors.green,  
      ), // AppBar  
      body: ListView.builder(  
        itemCount: _options.length,  
        itemBuilder: (BuildContext context, int index) {  
          return RadioListTile(  
            title: Text(_options[index]),  
            value: index,  
            groupValue: _selectedOption,  
            onChanged: (int? value) {  
              setState(() {  
                _selectedOption = value!;  
              }); }, ), // RadioListTile, ListView.builder  
        },  
        floatingActionButton: FloatingActionButton(  
          onPressed: () {  
            print('Option sélectionnée: ${_options[_selectedOption]}');  
          },  
          child: const Icon(Icons.check),  
        ), // FloatingActionButton  
      ); // Scaffold  
    }  
  }  
}
```



Introduction à Dart

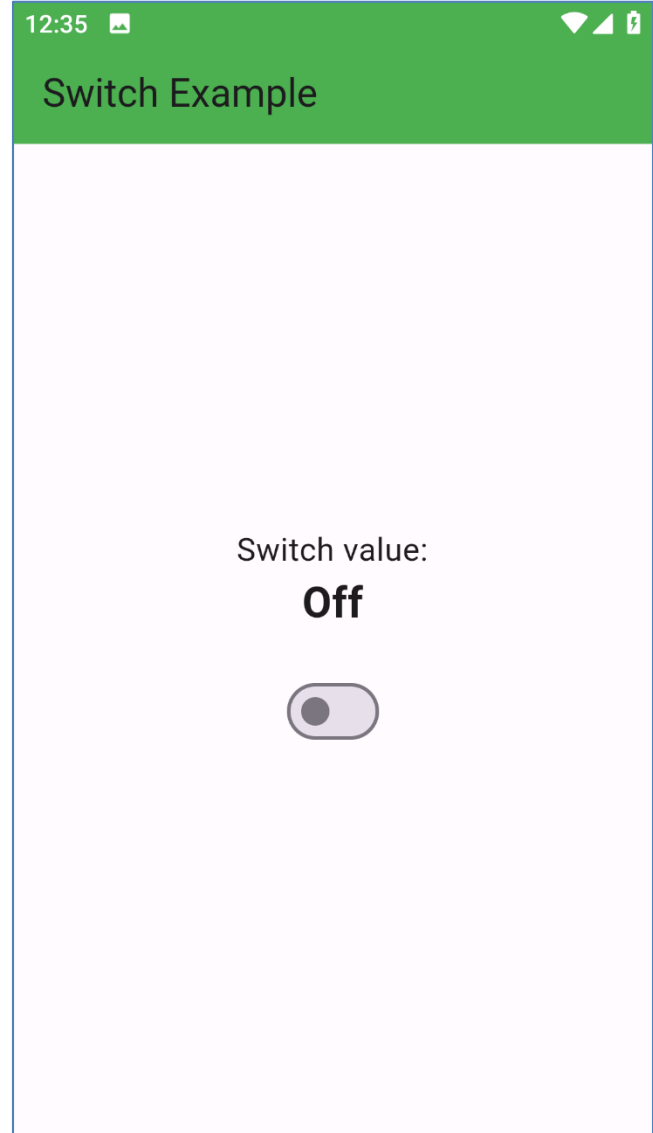
☐ Résultat:



```
✓ Built build\app\outputs\flutter-apk\app-debug.apk.  
Installing build\app\outputs\flutter-apk\app-debug.apk..  
Debug service listening on ws://127.0.0.1:50482/6-6onbPp..  
Syncing files to device SM S901N..  
D/EGL_adreno(10755): eglMakeCurrent: 0x7fff6c05a4e0: ver..  
I/flutter (10755): Option sélectionnée: Option 3
```

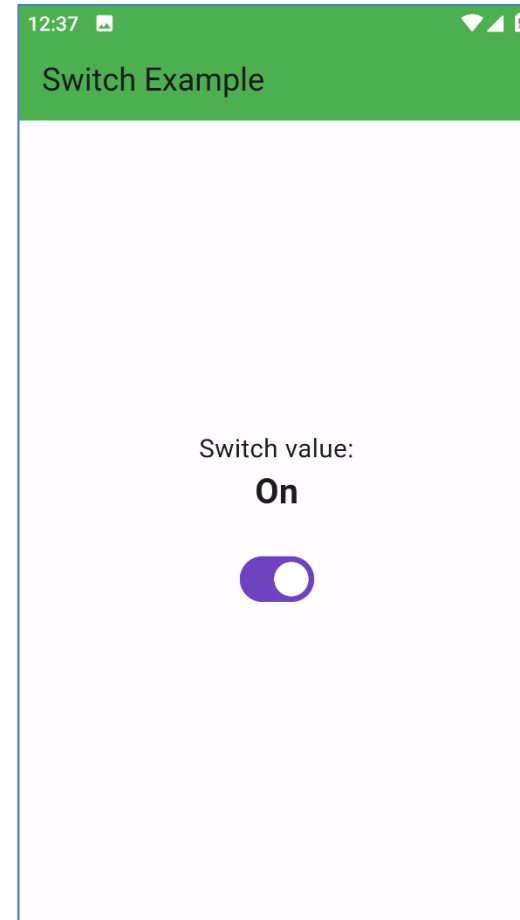
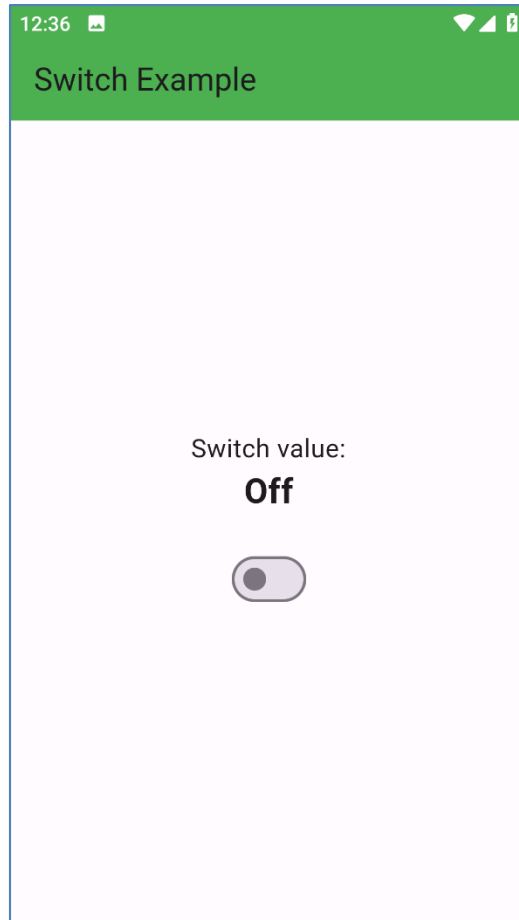
Le widget Switch

```
class _MyHomePageState extends State<MyHomePage> {
  bool _switchValue = false;
  void _handleSwitchChanged(bool newValue) {
    setState(() {
      _switchValue = newValue;
    });
  }
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Switch Example'),
        backgroundColor: Colors.green,
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            const Text(
              'Switch value:',
              style: TextStyle(fontSize: 18),
            ), // Text
            Text(
              _switchValue ? 'On' : 'Off',
              style: const TextStyle(fontSize: 24, fontWeight: FontWeight.bold),
            ), // Text
            const SizedBox(height: 20),
            Switch(
              value: _switchValue,
              onChanged: _handleSwitchChanged,
            ), // Switch
          ], // <Widget>[]
        ), // Column
      ), // Center
    ); // Scaffold
  }
}
```



Le widget Switch

☐ Résultat:

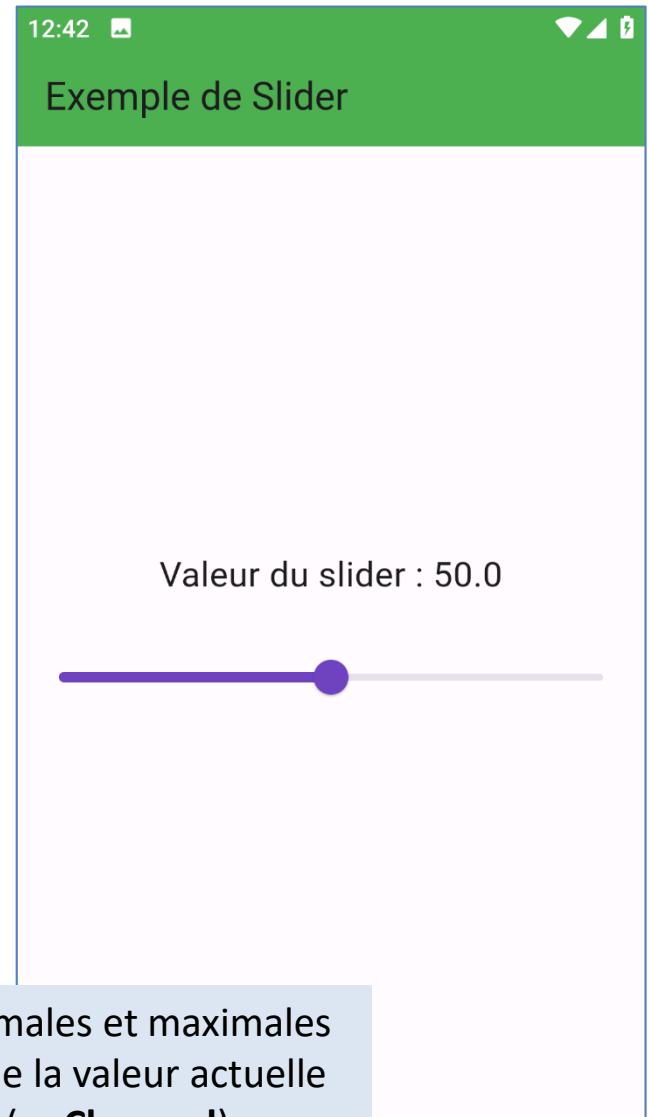


Le widget Slider

```
class _MyHomePageState extends State<MyHomePage> {
  double _sliderValue = 50.0;

  void _updateSliderValue(double newValue) {
    setState(() {
      _sliderValue = newValue;
    });
  }

  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Exemple de Slider'),
        backgroundColor: Colors.green,
      ), // AppBar
      body: Center(
        child: Column(
          mainAxisAlignment: MainAxisAlignment.center,
          children: <Widget>[
            Text(
              'Valeur du slider : ${_sliderValue.toStringAsFixed(1)}',
              style: const TextStyle(fontSize: 20.0),
            ), // Text
            const SizedBox(height: 20.0),
            Slider(
              value: _sliderValue,
              min: 0,
              max: 100,
              onChanged: _updateSliderValue,
            ), // Slider
          ], // <Widget>[]
        ), // Column
      ), // Center
    ); // Scaffold
  }
}
```

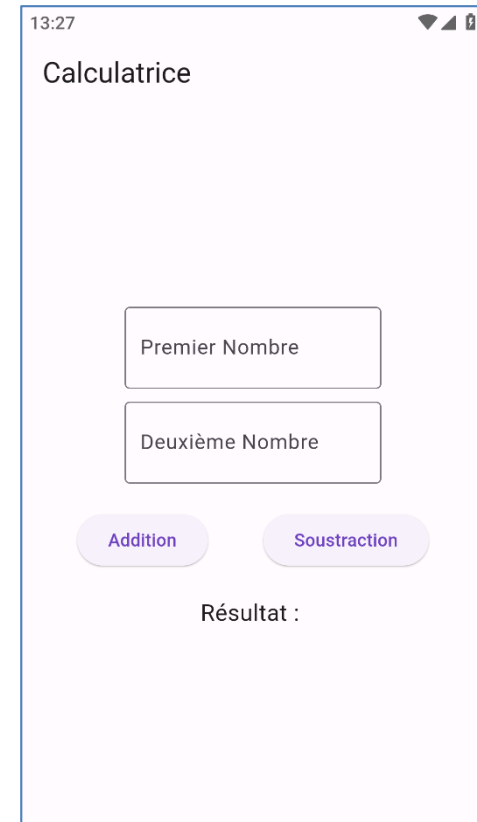


Vous définissez les valeurs minimales et maximales possibles (**min** et **max**), ainsi que la valeur actuelle (**value**) et la fonction de rappel (**onChanged**) pour gérer les modifications.

Calculatrice (1)

```
class _MyHomePageState extends State<MyHomePage> {  
  final TextEditingController controlleurPremierNombre = TextEditingController();  
  final TextEditingController controlleurSecondNombre = TextEditingController();  
  String result = '';  
  @override  
  Widget build(BuildContext context) {  
    return Scaffold(  
      appBar: AppBar(  
        title: const Text('Calculatrice'),  
      ), // AppBar  
      body: Center(  

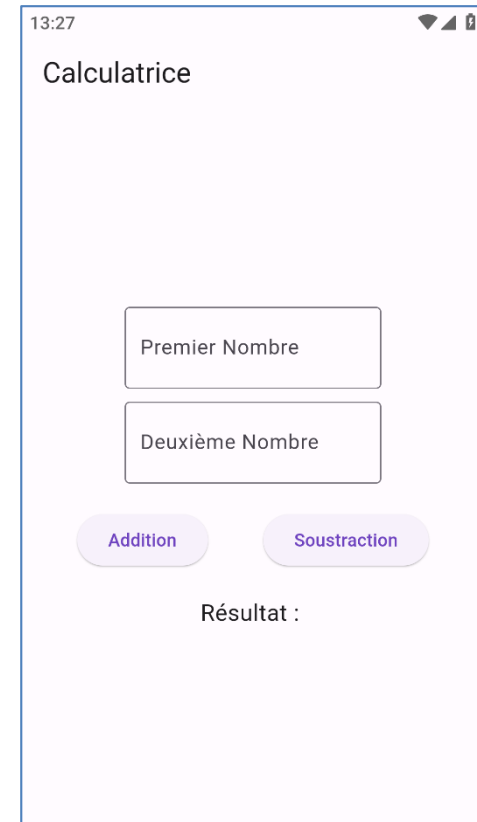
```



Calculatrice (2)

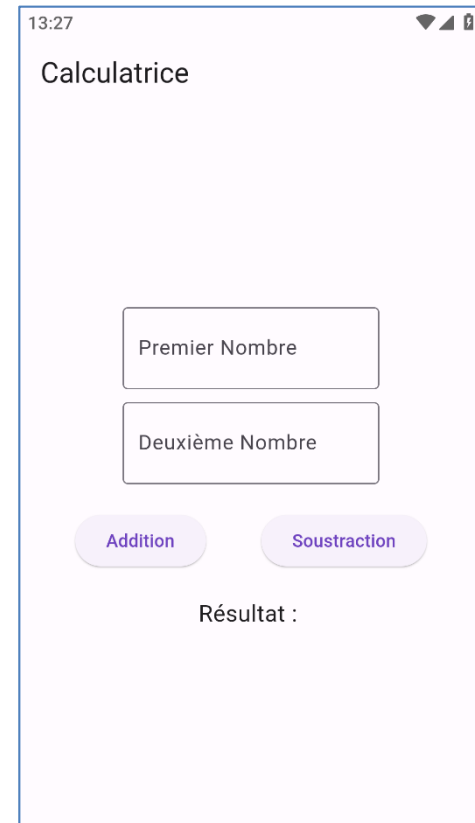
```
body: Center(  
  child: Column(  
    mainAxisAlignment: MainAxisAlignment.center,  
    children: <Widget>[  
      SizedBox(  
        width: 200,  
        child: TextField(  
          controller: controleurPremierNombre,  
          keyboardType: TextInputType.number,  
          decoration: const InputDecoration(  
            labelText: 'Premier Nombre',  
            border: OutlineInputBorder(),  
          ), // InputDecoration  
        ), // TextField  
      ), // SizedBox  
      const SizedBox(height: 10),  
      SizedBox(  
        width: 200,  
        child: TextField(  
          controller: controleurSecondNombre,  
          keyboardType: TextInputType.number,  
          decoration: const InputDecoration(  
            labelText: 'Deuxième Nombre',  
            border: OutlineInputBorder(),  
          ), // InputDecoration  
        ), // TextField  
      ), // SizedBox  
      const SizedBox(height: 20),  
      Row(  

```



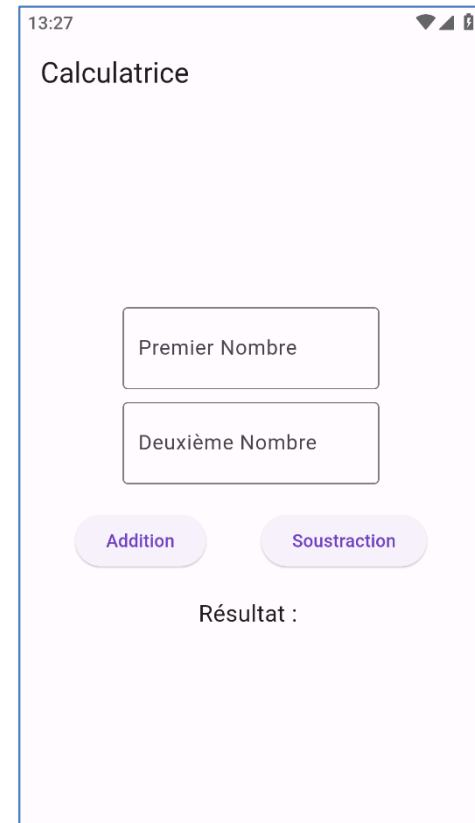
Calculatrice (3)

```
Row(  
  mainAxisAlignment: MainAxisAlignment.spaceEvenly,  
  children: [  
    ElevatedButton(  
      onPressed: () {  
        calculateResult('+');  
      },  
      child: const Text('Addition'),  
    ), // ElevatedButton  
    ElevatedButton(  
      onPressed: () {  
        calculateResult('-');  
      },  
      child: const Text('Soustraction'),  
    ), // ElevatedButton  
  ],  
), // Row  
const SizedBox(height: 20),  
Text('Résultat : $result', style: TextStyle(fontSize: 18)),  
] // <Widget>[]  
) // Column  
) // Center  
); // Scaffold  
}
```

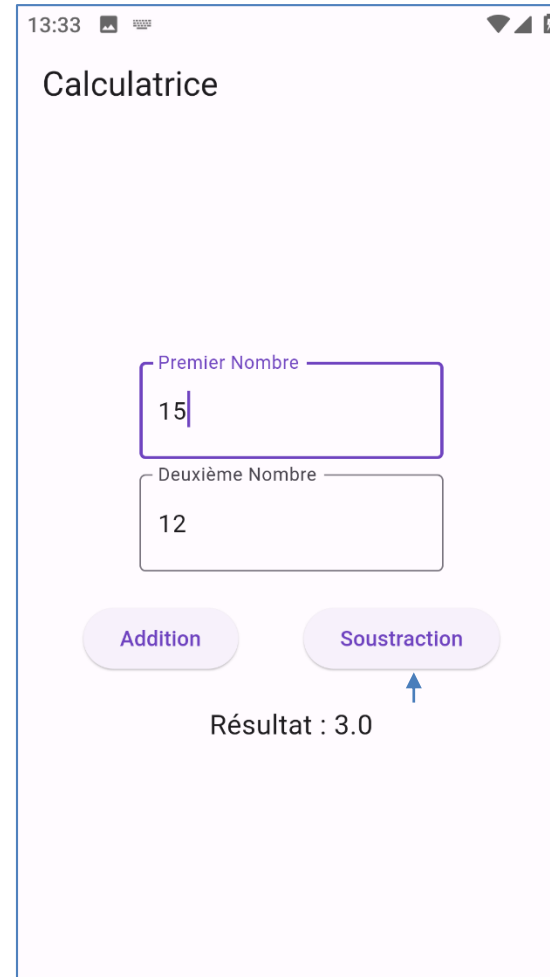
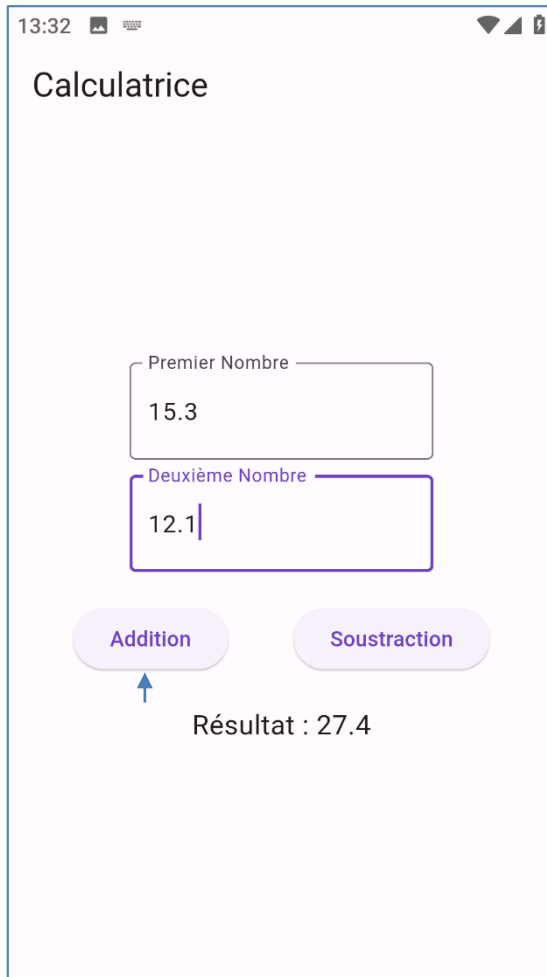


Calculatrice (4)

```
void calculateResult(String operation) {  
    double num1 = double.TryParse(controlleurPremierNombre.text) ?? 0;  
    double num2 = double.TryParse(controlleurSecondNombre.text) ?? 0;  
  
    setState(() {  
        if (operation == '+') {  
            result = (num1 + num2).toString();  
        } else if (operation == '-') {  
            result = (num1 - num2).toString();  
        }  
    });  
}
```

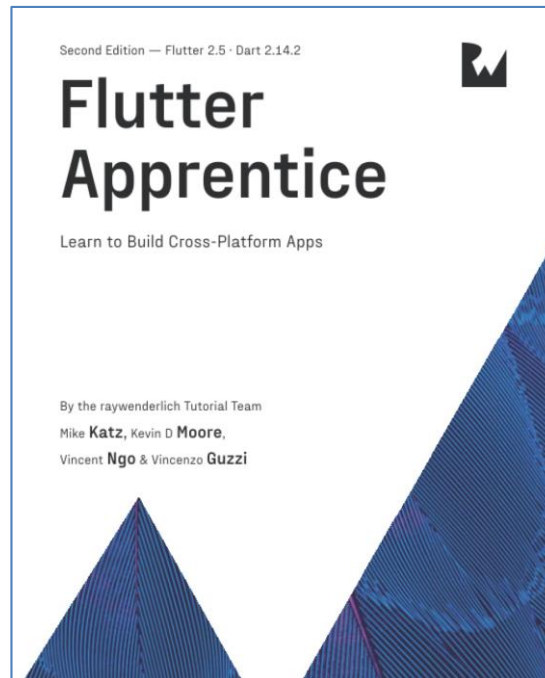
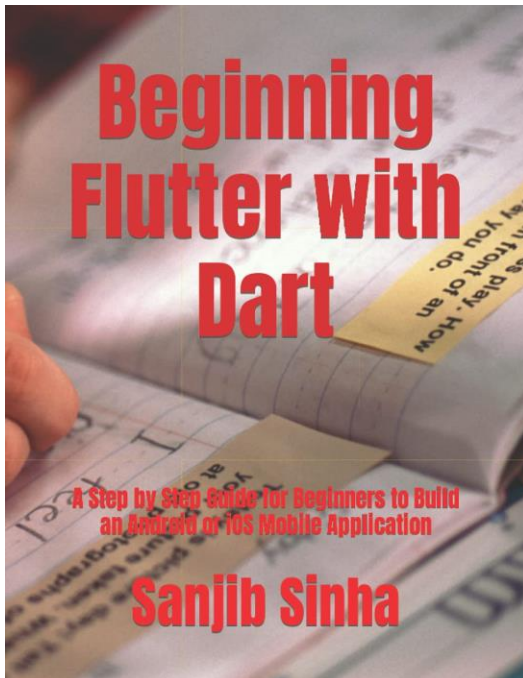


Calculatrice (5)



Références

- ✓ **Sanjib Sinha**. Beginning Flutter with Dart: A Step by Step Guide for Beginners to Build an Android or iOS Mobile Application (Flutter, Dart and Algorithm), 2021.
- ✓ **Mike Katz et al**. Flutter Apprentice Learn to Build Cross-Platform Apps, 2nd Edition, 2021.
- ✓ **Dieter Meiller**. Modern App Development with Dart and Flutter 2. 2021



Applications mobiles

