Le Widget Switch dans Flutter

[sommaire]

Le Widget Switch dans Flutter

1. **Objectif**

- Apprendre à utiliser le Widget Switch dans Flutter.

- **Qu'est-ce qu'un widget `Switch` ?**

  - Le widget "`Switch`" de Flutter est un composant d'interface utilisateur qui permet à l'utilisateur de basculer entre deux états. Il affiche un bouton à bascule qui peut être activé ou désactivé.
  - Un bouton de commutation est un widget Flutter avec seulement deux états, **vrai/faux** ou **activé/désactivé**. En règle générale, un "`Switch`" est un bouton avec un curseur de pouce pour que l'utilisateur le fasse glisser de gauche à droite et vice versa pour basculer entre les états. Il ne maintient pas son état tout seul. Vous devez appeler "**onChanged**" pour maintenir le bouton activé ou désactivé.

  - **Propretés:**

    - ***onChanged()***:
      - Appelle chaque fois que l'utilisateur appuie ou clique sur Switch.
    - ***value***:
      - contrôle si le switch sera activé ou désactivé en utilisant la valeur booléenne **true - false**.
    - ***activeColor***:
      - **activeColor** est une propriété de widget `Switch` dans Flutter qui définit la couleur du pouce du commutateur lorsque le commutateur est dans un état actif (c'est-à-dire lorsqu'il est activé). contrôle la couleur de la boule ronde **Switch** on **Switch on**.
    - **Exemple:**

apcpedagogie.com

```
Switch(
  value: true,
  activeColor: Colors.orangeAccent,
  onChanged: (value) {},
)
```

- **Interprétation de l'exemple:**
  - Dans cet exemple, le widget **Switch** est défini sur la valeur **true**, ce qui signifie qu'il est activé. La propriété **activeColor** est définie sur **Colors.orangeAccent**, ce qui signifie que la couleur de la piste sera **orangeAccent** lorsque l'interrupteur est activé.



- ***activeTrackColor***:
  - **activeTrackColor** est une propriété du widget Switch dans Flutter définit la couleur de la piste du commutateur lorsque le commutateur est dans un état actif (c'est-à-dire lorsqu'il est activé).
  - Vous pouvez utiliser cette propriété pour personnaliser l'apparence du commutateur en définissant la couleur de la piste pour qu'elle corresponde au jeu de couleurs de votre application.
  - **Exemple:**

```
Switch(
  value: true,
  activeTrackColor: Colors.blue,
  onChanged: (value) {},
)
```

  - **Interprétation de l'exemple:**
    - Dans cet exemple, le widget **Switch** est défini sur la valeur **true**, ce qui signifie qu'il est activé. La propriété **activeTrackColor** est définie sur **Colors.blue**, ce qui signifie que la couleur de la piste sera bleue lorsque l'interrupteur est activé.



- ***inactiveThumbColor***:
  - **inactiveThumbColor** est une propriété du widget dans Flutter qui définit la couleur du pouce du Switch lorsque celui-ci est dans un état inactif (c'est-à-dire lorsqu'il est

désactivé).Switch.

- **Exemple:**

```
Switch(
  value: true,
  inactiveThumbColor: Colors.grey,
  onChanged: (value) {},
)
```

- **Interprétation de l'exemple:**
  - Dans cet exemple, le widget **Switch** est défini sur la valeur **true**, ce qui signifie qu'il est activé. La propriété **inactiveThumbColor** est définie sur **Colors.green**, ce qui signifie que la couleur du pouce sera verte lorsque l'interrupteur est désactivé.



- ***inactiveTrackColor***:
  - contrôle l'événement Changer la couleur de la barre de piste lors de l'arrêt.
  - **inactiveTrackColor** est une propriété de widget `Switch` dans Flutter qui définit la couleur de la piste du commutateur lorsque le commutateur est dans un état inactif (c'est-à-dire lorsqu'il est éteint).
    - **Exemple:**

```
Switch(
  value: false,
  inactiveTrackColor: Colors.red,
  onChanged: (value) {},
)
```

    - **Interprétation de l'exemple:**
      - Dans cet exemple, le widget **Switch** est défini sur la valeur **false**, ce qui signifie qu'il est désactivé. La propriété **inactiveTrackColor** est définie sur **Colors.red**, ce qui signifie que la couleur de la piste sera rouge lorsque l'interrupteur est désactivé.



apcpedagogie.com

- **Exemple:**

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

void main() => runApp(const MyApp());

/// main application widget
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);

  static const String _title = 'Flutter Application';

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      home: Scaffold(
        appBar: AppBar(title: const Text(_title)),
        body: const MyStatefulWidget(),
      ),
    );
  }
}

/// stateful widget that the main application instantiates
class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({Key? key}) : super(key: key);

  @override
  State createState() => _MyStatefulWidgetState();
}

/// private State class that goes with MyStatefulWidget
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  bool isSwitched = false;
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding:
          const EdgeInsets.all(40.0), // set your desired padding value here
      child: Center(
        child: Column(
```

apcpedagogie.com

```
          children: [
            Switch(
activeColor: Colors.orangeAccent,
              activeTrackColor: Colors.blue,
              inactiveThumbColor: Colors.green,
              inactiveTrackColor: Colors.red,
              value: isSwitched,
              onChanged: (value) {
                setState(() {
                  isSwitched = value;
                });
              },
            ),
          ],
        ),
      ),
    );
  }
}
```

- ## Qu'est-ce qu'un widget `ToggleButton` ?

  - De même, un widget **`ToggleButton`** (bascule) n'a que deux états : **vrai/faux** ou a **ctivé/désactivé**. Mais un widget **`ToggleButton`** crée plusieurs boutons disposés en rangées, permettant aux utilisateurs de basculer entre eux.
  - En Flutter, un widget **`ToggleButton`** est un bouton qui permet à l'utilisateur de sélectionner une ou plusieurs options dans une liste de choix. Les boutons de bascule peuvent être configurés pour avoir un aspect différent selon leur état actuel, qu'il soit sélectionné ou désélectionné.

- ## Applications

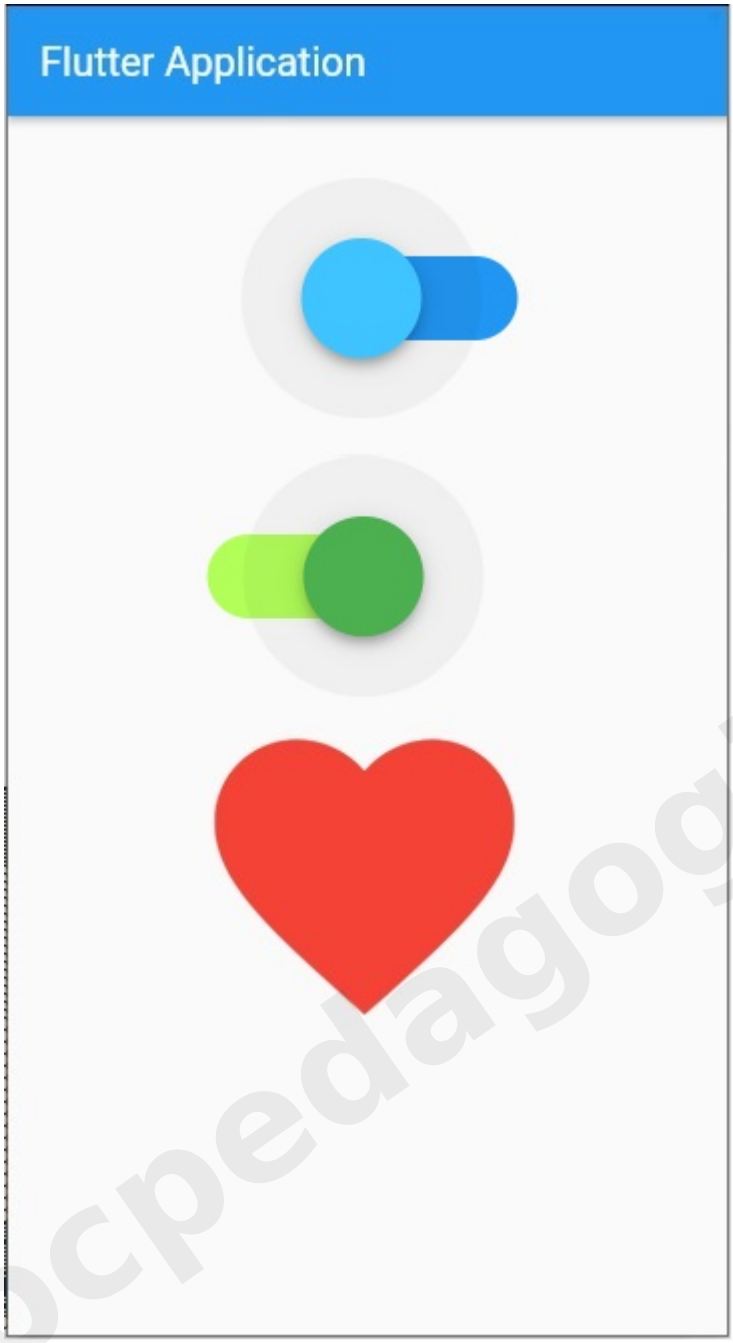
  - **App01**

    - Réalisez une application Flutter contenant un bouton **bascule Switch** bien décoré. Lorsque l'utilisateur active le bouton, un icône de cœur doit s'afficher à l'écran. Lorsque le bouton est désactivé, l'icône de cœur doit être masqué. Assurez-vous que le bouton bascule et l'icône de cœur sont esthétiquement agréables et fonctionnent correctement
    - Vous devez procéder comme suit
      - **1-** Commencez par créer un nouveau projet Flutter en utilisant la commande « flutter create » ou en utilisant l'option « New Flutter Project » dans votre

environnement de développement intégré.

- **2-**Dans votre fichier « main.dart », créez un nouveau StatefulWidget nommé « MyApp » et initialisez sa méthode « createState » pour renvoyer une nouvelle instance de « MyAppState ».
- **3-**Dans la classe "MyAppState", créez une variable booléenne nommée "**isSwitched**" qui représentera l'état du widget switch.
- **4-**Ajoutez un Scaffold à votre arbre de widgets et créez un AppBar avec un titre "**Switch Example**".
- **5-**Dans le corps de votre Scaffold, créez un Row contenant un Text "Show Heart Icon" et un Switch qui prendra la valeur de la variable "**isSwitched**". Placez-les dans un Container pour pouvoir les espacer.
- **6-**Ajoutez un GestureDetector autour d'un IconButton contenant l'icône cœur. Lorsque vous appuyez sur l'icône, vous pouvez utiliser la méthode setState pour inverser la valeur de "**isSwitched**".
- **7-**Définissez un thème pour votre application et personnalisez la couleur du widget switch en fonction de son état.
- **8-**Ajoutez des décorations pour rendre le widget switch plus attrayant, comme des bordures, des ombres ou des icônes personnalisées.
- **9-**Exécutez votre application Flutter et testez le widget switch en activant et désactivant l'icône cœur.
- **10-**Vous avez terminé l'exercice avec succès ! Vous pouvez maintenant explorer davantage Flutter en ajoutant d'autres fonctionnalités à votre application.

apcpedagogie.com



Solution

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

void main() => runApp(const MyApp());

/// main application widget
class MyApp extends StatelessWidget {
  const MyApp({Key? key}) : super(key: key);
```

apcpedagogie.com

```
  static const String _title = 'Flutter Application';

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      title: _title,
      home: Scaffold(
        appBar: AppBar(title: const Text(_title)),
        body: const MyStatefulWidget(),
      ),
    );
  }
}

/// stateful widget that the main application instantiates
class MyStatefulWidget extends StatefulWidget {
  const MyStatefulWidget({Key? key}) : super(key: key);

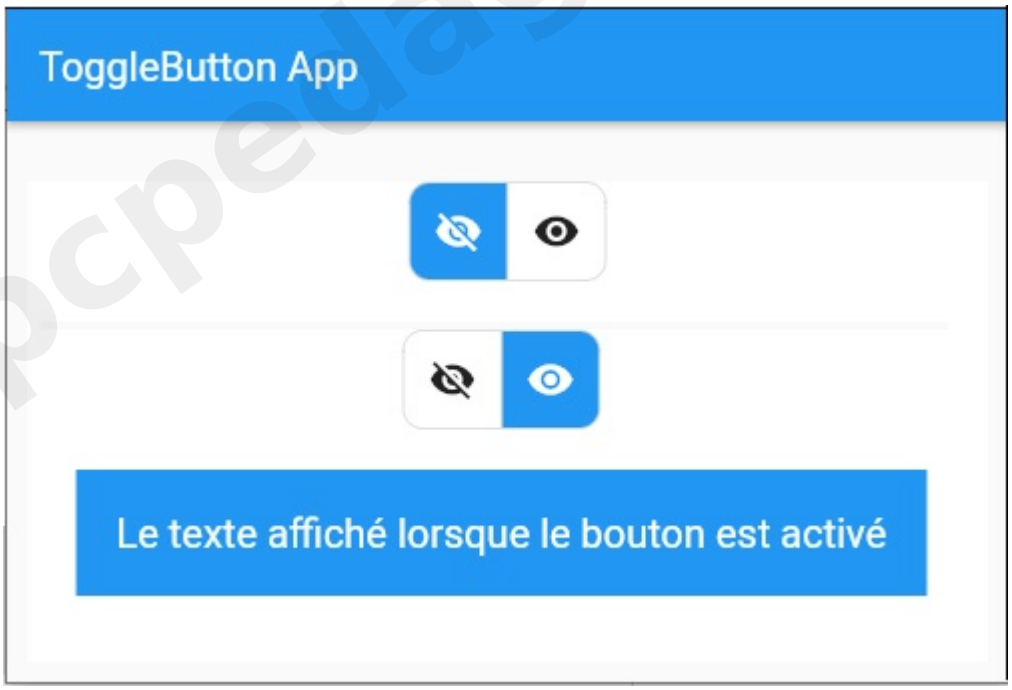
  @override
  State createState() => _MyStatefulWidgetState();
}

/// private State class that goes with MyStatefulWidget
class _MyStatefulWidgetState extends State<MyStatefulWidget> {
  bool isSwitched = false;
  @override
  Widget build(BuildContext context) {
    return Padding(
      padding:
        const EdgeInsets.all(40.0), // set your desired padding value here
      child: Center(
        child: Column(
          children: <Widget>[
            Switch(
             activeTrackColor: Colors.lightGreenAccent,
                activeColor: Colors.green,
                inactiveThumbColor: Colors.lightBlueAccent,
                inactiveTrackColor: Colors.blue,
              value: isSwitched,
              onChanged: (value) {
                setState(() {
                  isSwitched = value;
                });
              },
```

apcpedagogie.com

```
            ),
            AnimatedOpacity(
              opacity: isSwitched ? 1.0 : 0.0,
              duration: const Duration(milliseconds: 500),
              child: const Icon(
                Icons.favorite,
                color: Colors.red,
                size: 180.0,
              ),
            ),
          ],
        ),
      ),
    );
  }
}
```

- **App02**
  - Créer une application Flutter qui utilise le widget Toggle pour activer ou désactiver l'affichage d'un widget Text, avec une couleur d’arrière-plan pour le widget Text ?
  - Le schéma ci-dessous résume la tâche à accomplir.



Solution

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

void main() {
  runApp(MyApp());
}

class MyApp extends StatefulWidget {
  @override
  _MyAppState createState() => _MyAppState();
}

class _MyAppState extends State<MyApp> {
  bool _isVisible = false;

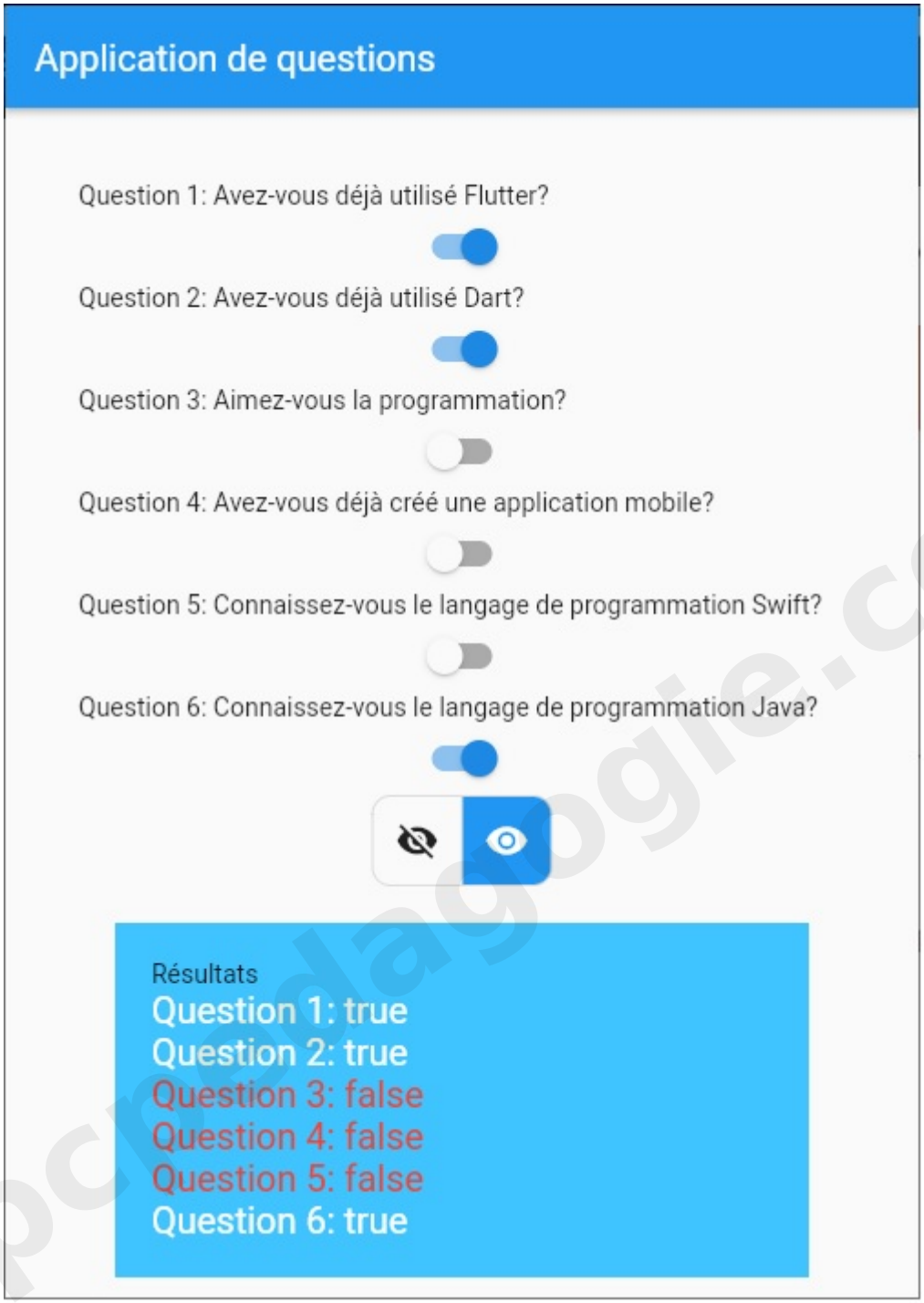
  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      home: Scaffold(
        appBar: AppBar(
          title: Text('ToggleButton App'),
        ),
        body: Container(
          margin: EdgeInsets.fromLTRB(10, 30, 10, 10),
          color: Colors.white,
          child: Column(
            mainAxisAlignment: MainAxisAlignment.start,
            children: <Widget>[
              Center(
                child: ToggleButtons(
                  children: <Widget>[
                    Icon(Icons.visibility_off),
                    Icon(Icons.visibility),
                  ],
                  isSelected: <bool>[
                    !_isVisible,
                    _isVisible,
                  ],
                  onPressed: (index) {
                    setState(() {
                      _isVisible = !_isVisible;
                    });
                  },
                  selectedColor: Colors.white,
```

apcpedagogie.com

```
                fillColor: Colors.blue,
                borderRadius: BorderRadius.circular(10),
              ),
            ),
            Visibility(
              visible: _isVisible,
              child: Container(
                margin: EdgeInsets.all(20),
                padding: EdgeInsets.all(20),
                color: Colors.blue,
                child: Text(
                  'Le texte affiché lorsque le bouton est activé',
                  style: TextStyle(color: Colors.white, fontSize: 20),
                ),
              ),
            ),
          ],
        ),
      ),
    ),
  );
 }
}
```

- **App03**
  - Créer une application Flutter qui propose six questions à l'utilisateur sous forme de widgets Switch, lui permettant de répondre à chaque question. Les résultats de ses réponses seront affichés dans un widget Text avec une couleur d’arrière-plan bleu ciel. Les réponses doivent être affichées en blanc si l’utilisateur à choisit une réponse vraie, et en rouge si l’utilisateur à choisit une réponse fausse Le schéma ci-dessous résume la tâche à accomplir.

apcpedagogie.com



Solution

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

void main() {
  runApp(MyWidget());
}
```

apcpedagogie.com

```
class MyWidget extends StatelessWidget {
  const MyWidget({super.key});

  @override
  Widget build(BuildContext context) {
    return MaterialApp(
      debugShowCheckedModeBanner: false,
      title: 'App02',
      home: QuestionsPage(),
    );
  }
}

class QuestionsPage extends StatefulWidget {
  @override
  _QuestionsPageState createState() => _QuestionsPageState();
}

class _QuestionsPageState extends State {
  // variables pour stocker les réponses de l'utilisateur
  bool reponse1 = false;
  bool reponse2 = false;
  bool reponse3 = false;
  bool reponse4 = false;
  bool reponse5 = false;
  bool reponse6 = false;
  bool _isVisible = false;
  @override
  Widget build(BuildContext context) {
    return Scaffold(
      appBar: AppBar(
        title: const Text('Application de questions'),
      ),
      body: Container(
        margin: EdgeInsets.all(20),
        padding: const EdgeInsets.all(20.0),
        child: Column(
            crossAxisAlignment: CrossAxisAlignment.stretch,
            children: <Widget>[
              // Question 1
              const Text('Question 1: Avez-vous déjà utilisé Flutter?'),
              Switch(
                value: reponse1,
                onChanged: (bool value) {
```

apcpedagogie.com

```
      setState(() {
        reponse1 = value;
      });
    },
  ),
  // Question 2
  const Text('Question 2: Avez-vous déjà utilisé Dart?'),
  Switch(
    value: reponse2,
    onChanged: (bool value) {
      setState(() {
        reponse2 = value;
      });
    },
  ),
  // Question 3
  const Text('Question 3: Aimez-vous la programmation?'),
  Switch(
    value: reponse3,
    onChanged: (bool value) {
      setState(() {
        reponse3 = value;
      });
    },
  ),
  // Question 4
  const Text(
      'Question 4: Avez-vous déjà créé une application mobile?'),
  Switch(
    value: reponse4,
    onChanged: (bool value) {
      setState(() {
        reponse4 = value;
      });
    },
  ),
  // Question 5
  const Text(
      'Question 5: Connaissez-vous le langage de programmation Swift?'),
  Switch(
    value: reponse5,
    onChanged: (bool value) {
      setState(() {
        reponse5 = value;
```

apcpedagogie.com

```
      });
    },
  ),
  // Question 6

  const Text(
      'Question 6: Connaissez-vous le langage de programmation Java?'),
  Switch(
    value: reponse6,
    onChanged: (bool value) {
      setState(() {
        reponse6 = value;
      });
    },
  ),
  // Bouton pour afficher les résultats

  Center(
    child: ToggleButtons(
      children: <Widget>[
        Icon(Icons.visibility_off),
        Icon(Icons.visibility),
      ],
      isSelected: <bool>[
        !_isVisible,
        _isVisible,
      ],
      onPressed: (index) {
        setState(() {
          _isVisible = !_isVisible;
        });
      },
      selectedColor: Colors.white,
      fillColor: Colors.blue,
      borderRadius: BorderRadius.circular(10),
    ),
  ),
  Visibility(
    visible: _isVisible,
    child: Container(
      margin: EdgeInsets.all(20),
      padding: EdgeInsets.all(20),
      color: Colors.lightBlueAccent,
      child: ListBody(
```

apcpedagogie.com

```
                    children: <Widget>[
                      const Text('Résultats'),
                      Text(
                        'Question 1: $reponse1',
                        style: TextStyle(
                            color: reponse1 ? Colors.white : Colors.red,
                            fontSize: 20),
                      ),
                      Text(
                        'Question 2: $reponse2',
                        style: TextStyle(
                            color: reponse2 ? Colors.white : Colors.red,
                            fontSize: 20),
                      ),
                      Text(
                        'Question 3: $reponse3',
                        style: TextStyle(
                            color: reponse3 ? Colors.white : Colors.red,
                            fontSize: 20),
                      ),
                      Text(
                        'Question 4: $reponse4',
                        style: TextStyle(
                            color: reponse4 ? Colors.white : Colors.red,
                            fontSize: 20),
                      ),
                      Text(
                        'Question 5: $reponse5',
                        style: TextStyle(
                            color: reponse5 ? Colors.white : Colors.red,
                            fontSize: 20),
                      ),
                      Text(
                        'Question 6: $reponse6',
                        style: TextStyle(
                            color: reponse6 ? Colors.white : Colors.red,
                            fontSize: 20),
                      ),
                    ],
                  ),
                ),
              ),
            ]),
      ),
```

apcpedagogie.com

```
    );
  }
}
```

Cours Flutter

Source : https://apcpedagogie.com/le-widget-switch-dans-flutter · téléchargé le 04/11/2026 · © APC Pédagogie · apcpedagogie.com

apcpedagogie.com