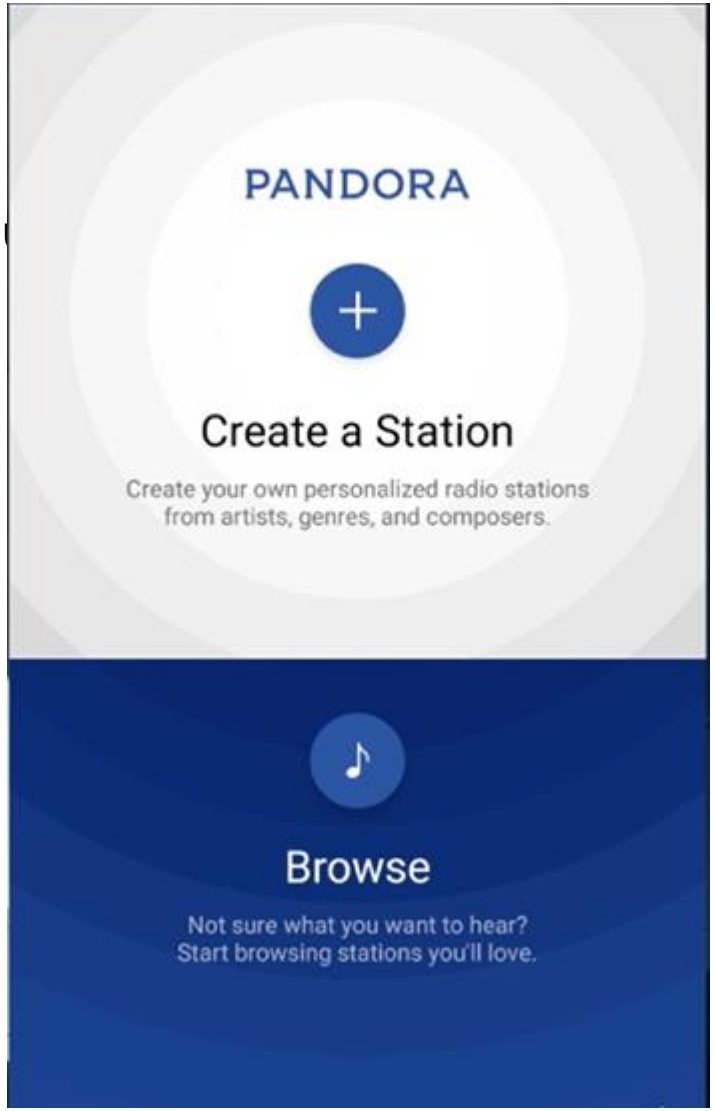
Qui est ce qui fait une bonne interface utilisateur

1. Beaucoup de clarté
   - tout est mis en page de manière simple et claire pour que l'utilisateu ambiguïté
   - prédire ce qui se passera quand il l'utilisera
   - la clarté inspire confiance qui conduit à une utilisation ultérieure
2. Permet l'interaction
3. Garde l'utilisateur concentré sur une seule chose
   - une chose = un écran
4. Garde le contrôle de l'utilisateur
5. Fournit une manière naturelle d'aller à la prochaine étape
6. Intuitive : l'apparence suit le comportement



Material Design : Qui est ce que c'est

1. Material est un système de conception créé par Google pour aider les équipes à créer des expériences numériques de haute qualité pour Android, iOS, Flutter et le Web.
2. Material Design utilise davantage de mises en page basées sur une grille, des animations et des transitions réactives, un remplissage et des effets de profondeur tels que l'éclairage et les ombres.
3. Material Design s'inspire du monde physique et de ses textures, y compris la façon dont ils reflètent la lumière et projettent des ombres. Les surfaces matérielles réinventent les médiums du papier et de l'encre.

Material Design : composants

1. Les composants matériels sont des blocs de construction interactifs permettant de créer une interface utilisateur et incluent un système d'états intégré pour communiquer les états de focus, de sélection, d'activation, d'erreur, de survol, d'appui, de glissement et de désactivation. Les bibliothèques de composants sont disponibles pour Android, iOS, Flutter et le Web.
2. Les composants couvrent une gamme de besoins d'interface, notamment :
   - **Affichage** : placer et organiser le contenu à l'aide de composants tels que des cartes, des listes et des feuilles.
   - **Navigation** : Permet aux utilisateurs de se déplacer dans le produit à l'aide de composants tels que des « tabs » et des onglets de navigation.
   - **Actions** : autoriser les utilisateurs à effectuer des tâches à l'aide de composants tels que le bouton d'action flottant.
   - **Saisie** : permet aux utilisateurs de saisir des informations ou d'effectuer des sélections à l'aide de composants tels que des champs de texte, des puces et des contrôles de sélection.
   - **Communication** : Alerter les utilisateurs sur les informations et les messages clés à l'aide de composants tels que des snack-bars, des bannières et des boîtes de dialogue.

Material Design : thème visuel

1. Material Theming facilite la personnalisation de Material Design pour qu'elle corresponde à l'apparence et à la convivialité de votre marque, avec une prise en charge et des conseils intégrés pour la personnalisation des couleurs, des styles de typographie et des formes d'angle.

Flutter – thème visuel

1. Thème : une façon de partager les couleurs, la typographie, et le style général à travers toute l'application
2. Pour quoi on en a besoin?
   - Imaginez que le client change d'avis sur l'ensemble de l'apparence de l'application, il veut:
     - changer toutes les polices d'en-tête en une plus grande taille
     - changer la couleur d'arrière-plan de l'appBar, la couleur de l'icône, etc.
     - toutes ce demande ont été faites après le cycle de développement

3. Solution : utilisation d'un thème
   1. une forme plus globale de stylisation de votre application
   2. facile à changer/modifier l'apparence de l'ensemble de votre application
   3. réduit la répétition de style
   4. Cohérence de l'interface utilisateur

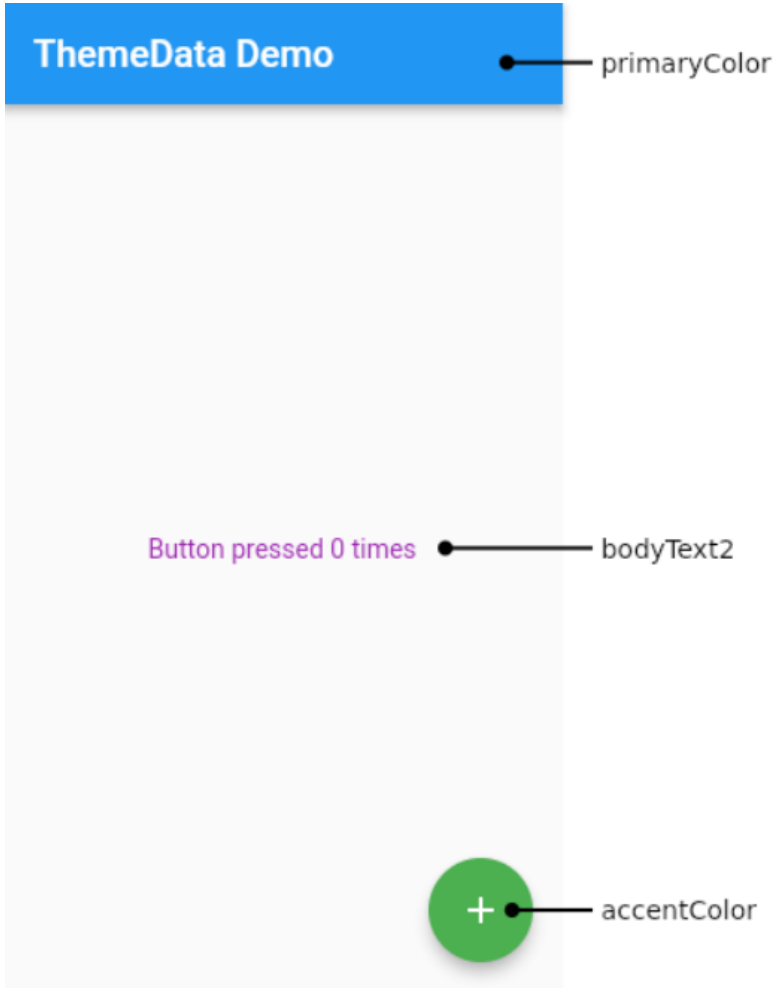
Flutter – utilisation de la classe ThemeData

1. Définit la configuration du thème visuel global pour une MaterialApp ou une sous-arborescence de widgets dans l'application.
2. La propriété thème de MaterialApp peut être utilisée pour configurer l'apparence de l'ensemble de l'application
3. Les sous-arborescences de widgets dans une application peuvent remplacer le thème de l'application en incluant un widget de thème en haut de la sous-arborescence
4. Les widgets dont l'apparence doit s'aligner sur le thème général peuvent obtenir la configuration du thème actuel avec Theme.of
   - Theme.of(context).textTheme.headline6,
   - Theme.of(context).colorScheme.primary

5. https://api.flutter.dev/flutter/material/ThemeData-class.html

Flutter – utilisation de la classe ThemeData

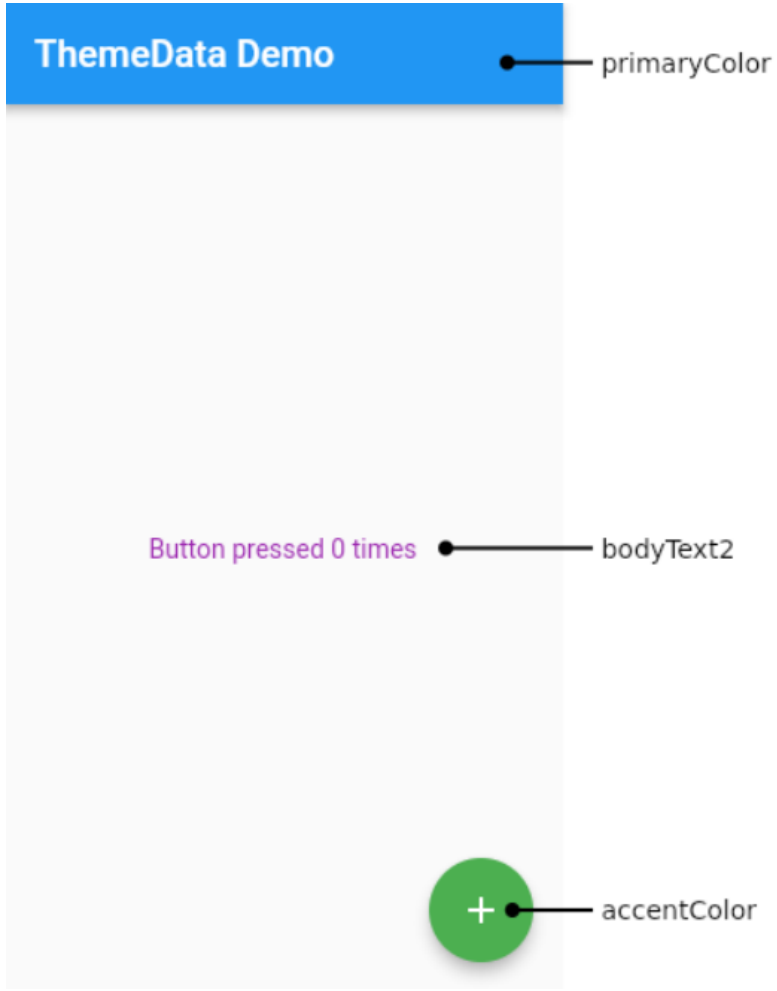
```
MaterialApp(
  theme: ThemeData(
    colorScheme: ColorScheme.fromSwatch(
      primarySwatch: Colors.blue,
    ).copyWith(
      secondary: Colors.green,
    ),
    textTheme: const TextTheme(bodyText2: TextStyle(color: Colors.purple)),
  ),
  home: Scaffold(
    appBar: AppBar(
      title: const Text('ThemeData Demo'),
    ),
    floatingActionButton: FloatingActionButton(
      child: const Icon(Icons.add),
      onPressed: () {},
    ),
    body: const Center(
      child: Text('Button pressed 0 times'),
    ),
  ),
)
```



Flutter – utilisation de la classe ThemeData

ColorSchema: Un ensemble de douze couleurs basé sur les spécification Material qui peut être utilisé pour configurer les propriétés de couleur de la plupart des composants.
https://material.io/design/color/the-color-system.html

```
@immutable
class ColorScheme with Diagnosticable {
  /// Create a ColorScheme instance.
  const ColorScheme({
    required this.primary,
    required this.primaryVariant,
    required this.secondary,
    required this.secondaryVariant,
    required this.surface,
    required this.background,
    required this.error,
    required this.onPrimary,
    required this.onSecondary,
    required this.onSurface,
    required this.onBackground,
    required this.onError,
    required this.brightness,
```



Flutter – utilisation de la classe ThemeData