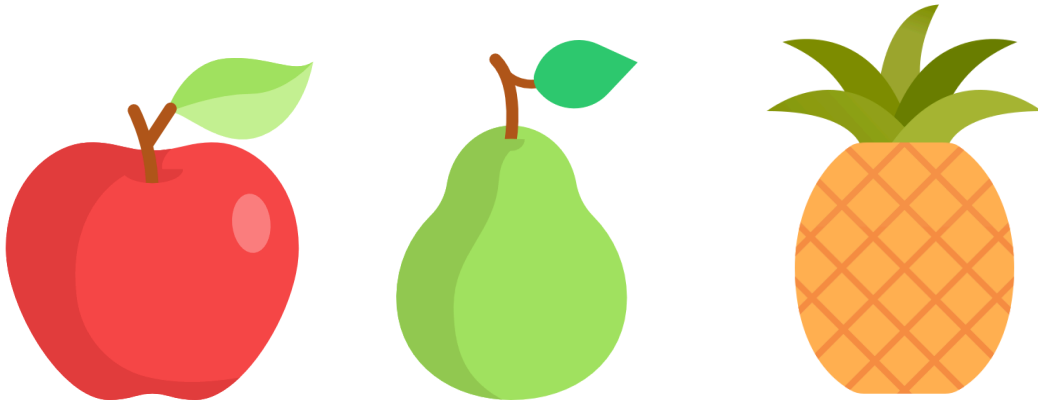


# TP - Pomme, Poire et Ananas

Flutter



## Objectifs

- Prise en main de Flutter,
- Découverte de la mise en page à l'aide de widgets,
- Gestion des interactions de l'utilisateur,
- Gestion de l'état applicatif (Stateless / Stateful),
- Emploi d'assets,
- Construction de liste dynamique,
- Affichage conditionnel.

Aidez-vous de la documentation officielle de Flutter pour réaliser ce TP (cf. <https://docs.flutter.dev/cookbook>).

Observez les captures d'écran à la fin du document.

# Etape 1 : Initialisation

1. Pour réaliser ce projet, vous adapterez le projet "*Counter*" généré par défaut avec le CLI de Flutter via la commande

```
$ flutter create pomme_poire_ananas
```

# Etape 2 : Liste dynamique

1. Utilisez un widget ***ListView*** permettant de construire dynamiquement une ***liste verticale d'items scrollable*** dans le ***body*** du widget ***Scaffold***.
2. Faites en sorte qu'à chaque clic sur le ***FloatingActionButton*** du ***Scaffold***, un nouvel item correspondant à la valeur courante de ***\_counter*** soit ajouté à une liste typée ne pouvant contenir que des nombres entiers. Nommez cette liste ***\_fruits***.
3. Associez la liste ***\_fruits*** au widget ***ListView*** afin de "*l'alimenter*" en items.
4. Employez un widget ***ListTile*** pour afficher chaque item au sein de la ***ListView***.

# Etape 3 : Assets

1. Créez un dossier ***images*** à la racine du projet et placez les fichiers ***poire.png***, ***pomme.png*** et ***ananas.png*** (cf. fichiers fournis).
2. Adaptez le fichier ***./pubspec.yaml*** afin d'intégrer les images au build de l'application (attention à l'indentation du fichier yaml).  
Un redémarrage de l'application est nécessaire pour que les adaptations dans le fichier ***./pubspec.yaml*** soient effectives.

# Etape 4 : Affichage dynamique et conditionnel

1. Affichez les items dont la valeur
  - a. est un nombre **pair** avec un arrière-plan de la couleur de votre choix (ex: Indigo),
  - b. est un nombre impair avec un arrière-plan d'une autre couleur de votre choix (ex: Cyan).

Dans tous les cas, affichez le nombre en **blanc**.

2. Appliquez le même code couleur pour le **FloatingActionButton**. Par défaut, lorsque le compteur est égal à 0, employez une autre couleur (ex: Orange).
3. A côté du nombre :
  - affichez l'image **ananas**, si la valeur est un **nombre premier** ([https://fr.wikipedia.org/wiki/Nombre\\_premier](https://fr.wikipedia.org/wiki/Nombre_premier)).

Pour ce faire, employez une fonction pure, de type "prédicat", qui indiquera si la valeur passée en paramètre est un nombre premier ou non.

Sinon :

- si la valeur est un nombre **pair**, affichez l'image **poire**,
- si la valeur est un nombre **impair**, affichez l'image **pomme**.

4. Dans le widget **AppBar**, affichez la valeur courante de **\_counter** suivie de son type en tant que titre :
  - **nombre premier**,
  - **pair**,
  - **impair**.

Ex : "1 : Nombre impair"

## Etape 5 : Interaction

- Au clic sur un **ListTile** contenu dans la **ListView**, affichez un widget **AlertDialog** contenant :
  - un titre adapté dynamiquement à la valeur du nombre cliqué, affichant la mention :
    - **nombre premier**,
    - **pair**,
    - **impair**.
  - une couleur d'arrière-plan correspondant au code couleur associé à la valeur du nombre cliqué,
  - une image de fruit correspondant à la règle associée à la valeur du nombre cliqué.
- Dans le widget **AlertDialog**, ajoutez un bouton permettant de
  - **supprimer le nombre cliqué** de la liste **\_fruits**
  - fermer l'**AlertDialog**.

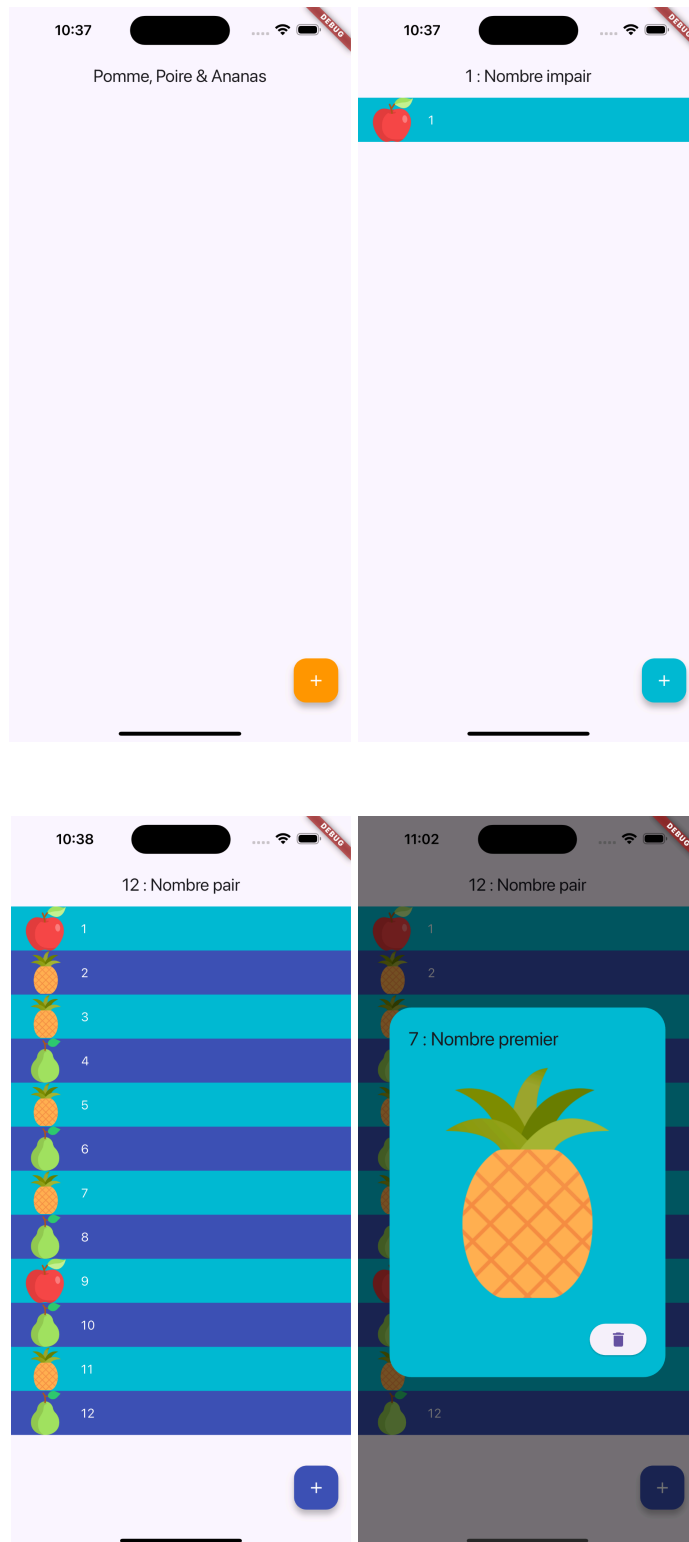
## Etape 6 : Refactoring

- Organisez au mieux votre code afin d'éviter les duplications,
- Employez des **méthodes pures** au sein de la classe de **State** afin de retourner dynamiquement les couleurs et images, selon les règles de gestion et la valeur communiquée en entrée.

## Crédits

- Images : <https://www.flaticon.com/>

# Aperçus



Flutter

Alexandre Leroux (alex@shrp.dev) - 2024