

Structure du Code

1. Importations :
 - Le code commence par importer des modules nécessaires pour l'interface graphique (tkinter), la gestion des fichiers (comme filedialog pour ouvrir et enregistrer les images), ainsi que des outils de traitement d'images avec Pillow (bibliothèque PIL) et Numpy pour les opérations sur les pixels d'image.
2. La Classe ApplicationFiltresImage :
 - Cette classe est le cœur de notre application. Elle gère toute l'interface utilisateur et les interactions avec l'image.
 - Constructeur `__init__` :
 - Nous définissons la fenêtre principale de l'application (root), configurons le titre, la taille et le fond de la fenêtre.
 - Nous initialisons les variables pour stocker l'image originale et filtrée (self.image_originale et self.image_filtrée).
 - Nous créons des éléments d'interface graphique : boutons pour ouvrir une image, appliquer un filtre, réinitialiser l'image et enregistrer l'image. Il y a aussi un menu déroulant pour choisir un filtre.
3. Fonctions de l'interface :
 - `ouvrir_image()` : Nous permettons à l'utilisateur d'ouvrir un fichier image depuis le système de fichiers avec un sélecteur de fichiers.
 - `appliquer_filtre()` : Nous appliquons un filtre à l'image selon le choix de l'utilisateur dans le menu déroulant. Chaque filtre est défini dans une fonction spécifique.
 - `enregistrer_image()` : Nous permettons à l'utilisateur d'enregistrer l'image filtrée.
 - `reinitialiser_image()` : Nous réinitialisons l'image filtrée à l'originale.
4. Fonctions de traitement d'images :
 - Ce sont des méthodes que nous avons définies pour appliquer différents filtres ou effets visuels à l'image. Par exemple :
 - Grayscale (niveau de gris)
 - Brightness (luminosité)
 - Contrast (contraste)
 - Sepia (effet sépia)
 - Pointillisme, Cubisme, Impressionnisme, etc.
 - Chaque méthode correspond à un effet ou transformation particulière appliquée à l'image, et elles utilisent la bibliothèque Pillow pour effectuer ces manipulations.
5. Exécution de l'application :
 - À la fin du fichier, si le fichier est exécuté en tant que programme principal, nous créons une instance de la classe ApplicationFiltresImage et lançons l'interface graphique.

Un seul fichier

Ce code est un fichier unique pour l'ensemble de l'application. Toutes les fonctionnalités nécessaires pour l'ouverture de l'image, l'application des filtres, la réinitialisation, l'enregistrement et l'affichage de l'image modifiée sont contenues dans ce fichier. Ce fichier contient donc :

- L'interface graphique (Tkinter).
- Les traitements d'image (avec Pillow et Numpy).
- Les interactions utilisateur (via des boutons et menus déroulants).

Conclusion

En résumé, ce programme est entièrement géré dans un seul fichier Python. Il nous permet de créer une interface graphique permettant de charger une image, d'appliquer des filtres visuels, et de sauvegarder l'image modifiée. Tout cela est réalisé avec Tkinter pour l'interface et Pillow pour le traitement des images, le tout dans un seul fichier Python bien structuré.