

Voici une liste des choix structurels présentes dans Le code, ainsi qu'une brève justification :

#### 1. Listes :

- utilisez des listes pour stocker les chemins des images (``chemins_images``), les noms uniques d'essences d'arbres (``essences_uniques``), et pour afficher les résultats dans l'interface utilisateur (``essences_uniques``).

- Justification : Les listes sont adaptées lorsque a une collection ordonnée d'éléments. Elles offrent une flexibilité pour ajouter, supprimer ou accéder aux éléments par index.

#### 2. Dictionnaires :

- utilisez un dictionnaire pour spécifier les entêtes dans le classeur Excel (``{"Nom de l'image", "Proportion de blanc (%)", "Surface en km2""}``).

- Justification : Les dictionnaires sont utiles pour stocker des paires clé-valeur. Dans ce cas, ils fournissent une manière lisible d'associer des entêtes aux colonnes du classeur Excel.

#### 3. DataFrame Pandas :

- utilisez des DataFrames Pandas pour manipuler et analyser des données tabulaires provenant des fichiers Excel (``arbres_df``, ``surfaces_df``, ``result_df``).

- Justification : Les DataFrames Pandas sont bien adaptés pour travailler avec des données tabulaires. Ils offrent des fonctionnalités puissantes pour filtrer, fusionner, et analyser les données, ce qui facilite le traitement et la manipulation des données.

#### 4. Numpy Array :

- utilisez des tableaux Numpy pour effectuer des opérations mathématiques sur des images, par exemple, dans la fonction ``compter_pixels_et_calculer_superficie``.

- Justification : Les tableaux Numpy sont optimisés pour les opérations vectorielles et matricielles, ce qui les rend adaptés pour le traitement d'images et le calcul numérique.

#### 5. Classe Workbook d'Openpyxl :

- utilisez la classe `Workbook` d'Openpyxl pour créer et manipuler des fichiers Excel.

- Justification : La classe Workbook d'Openpyxl est spécialement conçue pour travailler avec des fichiers Excel en Python. Elle offre des méthodes pour créer, lire et écrire des feuilles de calcul Excel. Et comme je vous le disais dans un cadre d'appréhension et en corrélation cela m'entraîne à manipuler ce type de fichier pour les application concrète demander par mes supérieurs.

#### 6. Tkinter Widgets :

- utilisez des widgets Tkinter tels que `Listbox` et `Text` pour créer une interface utilisateur.

- Justification : Tkinter fournit une bibliothèque simple pour créer des interfaces utilisateur graphiques. Les widgets Tkinter sont adaptés pour créer des fenêtres, boutons, listes, etc.

#### 7. Types de données simples (int, str, list) :

- utilisez des types de données simples tels que `int` et `str` pour représenter des entiers et des chaînes de caractères dans vos calculs.

- Justification : L'utilisation de types de données simples est appropriée lorsque l'on travaille avec des valeurs individuelles et que la complexité d'autres structures de données n'est pas nécessaire.

Dans ce code nous utilisons 2 interface graphique différentes une tkinter pour coller avec le cadre du projet python puis une sur html pour avoir un rendu plus agréable mais également pour manipuler des structures de code différentes et ainsi s'exercer sur d'autre chose en lien avec de la pluri discipline.