

Camembert

La classe `Camembert` en PyQt5 génère un graphique circulaire représentant la répartition des tailles de fichiers. Elle prend en entrée une liste de fichiers avec leurs tailles et une liste de couleurs. La méthode `dessine_camembert` crée un camembert en calculant le pourcentage de chaque fichier par rapport au total, en attribuant une couleur et une étiquette affichant la taille en Mo. Les petites tranches ont leurs étiquettes masquées pour plus de lisibilité. Le graphique est ensuite encapsulé dans un `QChartView` pour être affiché sous forme de widget PyQt5.

Analyse Fichier

Ce script Python analyse un répertoire donné, liste tous les fichiers qu'il contient, trie ces fichiers par taille, filtre les plus volumineux et sauvegarde les résultats dans un fichier JSON. Il utilise `pathlib` pour parcourir le répertoire, récupère la taille de chaque fichier, les trie du plus grand au plus petit, puis sélectionne ceux dépassant une certaine taille (1 Mo par défaut) en limitant leur nombre (100 maximum). Enfin, il enregistre ces fichiers dans `Fichier_Lourd.json` et affiche le nombre de fichiers sauvegardés. Ce script est utile pour identifier rapidement les fichiers volumineux dans un dossier.

Sélection Répertoire

Ce script utilise `PyQt5` pour afficher une boîte de dialogue permettant à l'utilisateur de sélectionner un dossier. Il initialise une application `QApplication`, configure `QFileDialog` en mode sélection de répertoire et attend l'interaction de l'utilisateur. Si un dossier est choisi, son chemin est récupéré et affiché. Sinon, `None` est retourné. Ce script est utile pour demander un dossier via une interface graphique.

Légendes

La classe `Légendes` en PyQt5 génère une liste de légendes associées à des fichiers. Chaque légende comprend une case à cocher, un carré coloré représentant le fichier et un texte affichant son chemin et sa taille. L'affichage est limité à 25 éléments à la fois. L'interface est construite avec des layouts verticaux et horizontaux pour organiser ces éléments. Enfin, une méthode permet de récupérer l'état des cases cochées pour savoir quels fichiers ont été sélectionnés. Cette classe est idéale pour afficher et gérer une liste interactive de fichiers dans une application PyQt5.

Pycache

Les fichiers **.pyc** sont des fichiers compilés de Python contenant le bytecode d'un script. Ils permettent un chargement plus rapide en évitant la recompilation à chaque exécution. Stockés dans le répertoire **__pycache__**, ils contribuent à une meilleure organisation du code. De plus, ils incluent des informations sur la version de Python utilisée, garantissant la compatibilité avec l'interpréteur.