HAI103I
Indexation arborescente de mots
-
Pierre Pompidor

Le but du TP est d'indexer les mots extraits de différents fichiers sous une forme arborescente (l'arbre étant implémenté sous la forme d'une imbrication de dictionnaires).

Imaginons que nous avons les trois mots suivants à indexer : chat, canard et cane, l'index créé aurait cet structure (sans références aux fichiers contenant ces mots) :

```
{'c':
    {'h':
        {'a':
            {'t': {}}
        },
    'a':
        {'n':
            {'a':
                {'r':
                    {'d': {}}
                },
            'e': {}
            }
        }
    }
}
```

Deux étapes :

Vous devez faire le TP en deux étapes :

- *indexeur.py* : prend en paramètres une suite de mots et crée l'index
- *indexeur_ 2.py* : prend en paramètre une suite de noms de fichiers et utilise les fonctions légèrement modifiées d'*indexeur.py* (qui pourra être renommé en *indexeur_ 1.py*) pour référencer les fichiers qui contiennent ces mots dans l'index, puis demande à l'utilisateur des mots à rechercher.

Pour pouvoir utiliser des fonctions d'un script python dans un autre script :

```
import indexeur_1

print(indexeur_1.index)  # exemple
```

Des fonctions récursives à programmer :

Tant que les premiers caractères du mot à indexer sont dans l'index, descente dans celui-ci → fonction récursive.
Les caractères suivants ne sont pas l'index : création des dictionnaires imbriqués → fonction récursive.

Pour aller plus loin :

Pour aller plus loin, trois extensions (qui peuvent se combiner) vous sont proposées :

- indexer non pas des mots, mais n'importe quelle chaîne de caractères prise parmi les mots
- indexer les fichiers qui se trouvent dans une arborescence de dossiers (couplage avec le TP d'exploration récursive)
- sauvegarde de l'index dans un fichier (sérialisation, désérialisation)