TP1 - Développement d'un web crawler

Durée : 3 heures

Objectif

Créer un crawler en python qui explore les pages d'un site web en priorisant certaines pages.

Contexte

Vous développerez un crawler qui

- Extrait le titre, le premier paragraphe et les liens internes
- Suit tous les liens dans le body en priorisant les liens de produit
- Stocke les données dans un fichier json
- S'arrête après avoir visité 50 pages

Il crawler un site web contenant de pages produits que les utilisateurs voudront ensuite chercher en passant par un moteur de recherche dédié.

Étapes guidées

1. Configuration initiale (30 min)
   - Installer les bibliothèques requises:
     - Pour requêter les urls et lire les robots.txt: https://docs.python.org/fr/3/library/urllib.html
     - Pour lire les fichiers html: https://www.crummy.com/software/BeautifulSoup/bs4/doc/
   - Créer la structure du projet
   - Implémenter les fonctions de base pour les requêtes HTTP
   - Ajouter la notion de politesse

2. Extraction du contenu (45 min)
   - Développer une fonction qui s'assure que le crawler a le droit de parser une page
   - Développer une fonction pour parser le HTML
   - Extraire titre, premier paragraphe et liens et l'information d'où viennent les liens

3. Logique de crawling (45 min)
   - Implémenter une file d'attente des URLs à visiter
   - Implémenter l'arrêt après 50 pages
   - Implémenter un système de priorité pour prioriser les urls contenant le token 'product'

4. Stockage des urls crawlées (10 min)
   - Sortir les résultats dans un fichier json

5. Tests et optimisation (30 min)
   - Tester sur différentes pages de départ
   - Gérer les erreurs courantes

6. Documenter (10 min)
   - N'oubliez pas que plus vos noms de variables et noms de fonctions sont explicites, moins il y a besoin de documenter

Rappels de programmation:

- Une fonction ne fait qu'une action, si vous avec envie de nommer votre fonction do_something_and_do_something_else -> alors il vous faut deux fonctions
- Le nom d'une fonction commence toujours pas un verbe d'action

Livrable

Un script Python qui prend en entrée :

- L'URL de départ https://web-scraping.dev/products
- Le nombre de documents maximum à visiter

Et produit en sortie un fichier JSON contenant pour chaque page :

- Titre
- URL
- Premier paragraphe
- Liste des liens pertinents

Critères d’évaluation

- Respect des consignes
- Propreté et lisibilité du code
- Résultat équivalent au fichier json donné