1. Pour extraire les citations des 5 premières pages j’ai utilisé la méthode suivante :

* Mon script utilise requests pour faire des requêtes HTTP à chaque page du site "https://quotes.toscrape.com/" et BeautifulSoup pour analyser le HTML retourné et extraire les informations sur les citations.
* Il collecte les citations, les auteurs et les tags de chaque citation trouvée sur les cinq premières pages, puis les enregistre dans un fichier CSV appelé results.csv.

import requests

from bs4 import BeautifulSoup

import csv

# Fonction pour extraire les citations d'une seule page

def scrape\_page(page\_url):

response = requests.get(page\_url)

soup = BeautifulSoup(response.text, 'html.parser')

quotes = soup.find\_all('div', class\_='quote')

return quotes

# URL de base

base\_url = "https://quotes.toscrape.com"

# Préparation pour l'écriture dans un fichier CSV

results = []

for i in range(1, 6): # Pour les cinq premières pages

page\_quotes = scrape\_page(f"{base\_url}/page/{i}/")

for quote in page\_quotes:

text = quote.find('span', class\_='text').text

author = quote.find('small', class\_='author').text

tags = [tag.text for tag in quote.find\_all('a', class\_='tag')]

results.append([text, author, ', '.join(tags)])

# Écriture des résultats dans un fichier CSV

with open('results.csv', 'w', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow(['Text', 'Author', 'Tags'])

writer.writerows(results)

print("Le fichier CSV a été créé avec succès.")

1. Après filtration des tags

* Extraction des citations et de leurs tags pour chaque page, en utilisant BeautifulSoup pour parcourir le contenu HTML.
* Filtrage des citations : seulement celles dont les tags incluent au moins un des quatre tags spécifiés ('love', 'inspirational', 'life', 'humor') sont conservées.
* Sauvegarde dans un fichier CSV : les citations filtrées sont enregistrées dans un fichier results.csv avec les champs Texte, Auteur, et Tags.

import requests

from bs4 import BeautifulSoup

import csv

# Fonction pour extraire les citations d'une seule page

def scrape\_page(page\_url):

response = requests.get(page\_url)

soup = BeautifulSoup(response.text, 'html.parser')

quotes = soup.find\_all('div', class\_='quote')

return quotes

# URL de base et tags désirés

base\_url = "https://quotes.toscrape.com"

desired\_tags = {'love', 'inspirational', 'life', 'humor'}

# Préparation pour l'écriture dans un fichier CSV

results = []

for i in range(1, 6): # Pour les cinq premières pages

page\_quotes = scrape\_page(f"{base\_url}/page/{i}/")

for quote in page\_quotes:

text = quote.find('span', class\_='text').text

author = quote.find('small', class\_='author').text

tags = [tag.text for tag in quote.find\_all('a', class\_='tag')]

# Filtrer les citations contenant au moins un des tags désirés

if any(tag in desired\_tags for tag in tags):

results.append([text, author, ', '.join(tags)])

# Écriture des résultats dans un fichier CSV

with open('results.csv', 'w', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow(['Text', 'Author', 'Tags'])

writer.writerows(results)

print("Le fichier CSV a été créé avec succès.")

1. Le fichier results.csv se crée automatiquement avec le code.
2. Pour trouver le token, on va dans inspecter la page du site a scrapper, on se place dans network, on se login, ensuite on va dans login, payload et le token sera affiché.
3. Pour ajouter les deux premières pages de citations ayant le tag books tout en evitant les doublons :

J’ai parcouru les cinq premières pages : J'ai collecté les citations contenant les tags 'love', 'inspirational', 'life', 'humor', ou 'books', et je les ai ajoutées à une liste de résultats tout en vérifiant que leurs tags n'étaient pas déjà présents dans un ensemble de tags vus (seen\_tags).

Ensuite j’ai parcouru les deux premières pages pour le tag 'books' : J'ai ensuite collecté spécifiquement les citations contenant le tag 'books' des deux premières pages. Avant d'ajouter chaque citation à la liste des résultats, j'ai vérifié si l'un de ses tags était déjà présent dans seen\_tags pour éviter les doublons, en ajoutant les tags de chaque citation ajoutée à cet ensemble.