**V- Extraire et enregistrer les fichiers images**

Ajouté les fonctionnalités suivantes au script :

○ Télécharger les images associées de tous les livres de toutes les catégories.

Pour un livre image21.py (OK)

#Télecharger et enregistrer le fichier image de chaque page Produit qu’on consulte

import requests

from bs4 import BeautifulSoup as bs

import urllib.request

import os

url="https://books.toscrape.com/catalogue/soumission\_998/index.html"

response = requests.get(url)

if response.ok:

    soup = bs(response.text,'lxml')

    image = soup.find('div', class\_="item active").find('img')

    print(image)

    #<img alt="A Light in the Attic" src="../../media/cache/fe/72/fe72f0532301ec28892ae79a629a293c.jpg"/>

def recup\_chemin\_image(soup):

    target\_link\_image = soup.find('div', class\_="item active").find('img')['src']

    print( target\_link\_image)

    #../../media/cache/fe/72/fe72f0532301ec28892ae79a629a293c.jpg

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

    complete\_link = base\_url + target\_link\_image

    return complete\_link.replace("../../",'')

    print(recup\_chemin\_image(soup))

 #https://books.toscrape.com/media/cache/fe/72/fe72f0532301ec28892ae79a629a293c.jpg

image\_url = recup\_chemin\_image(soup)

def sauv\_image(image\_url,fichier):

    my\_path=os.path.dirname(\_\_file\_\_)+ '\img2'

    if not os.path.exists(my\_path):

        os.makedirs(my\_path)

    fullfilename = os.path.join(my\_path, fichier)

    urllib.request.urlretrieve(image\_url,fullfilename)

sauv\_image(image\_url,'sauvefichier2.jpeg')

print("création du dossier 'img2' ou est enregistré le fichier 'sauvefichier2.jpeg'")

sauvegarde images de tous les livres et de toutes les categories (à parametrer)

# Télécharger les images associées de tous les livres de toutes les catégories

import requests

from bs4 import BeautifulSoup

import csv

import os

# Fonction pour extraire les liens vers toutes les catégories de livres

def extract\_category\_links():

    base\_url = 'http://books.toscrape.com/index.html'

    response = requests.get(base\_url)

    if response.status\_code == 200:

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

        category\_links = soup.select('ul.nav-list li ul li a')

        category\_urls = []

        for link in category\_links:

            category\_name = link.text.strip()

            category\_url = 'http://books.toscrape.com/' + link['href']

            category\_urls.append((category\_name, category\_url))

        return category\_urls

    else:

        print(f'Erreur {response.status\_code} lors de la requête.')

        return []

# Fonction pour télécharger une image

def download\_image(image\_url, save\_dir, category):

    response = requests.get(image\_url)

    if response.status\_code == 200:

        image\_name = os.path.basename(image\_url)

        category\_dir = os.path.join(save\_dir, category)

        save\_path = os.path.join(category\_dir, image\_name)

        if not os.path.exists(category\_dir):

            os.makedirs(category\_dir)

        with open(save\_path, 'wb') as f:

            f.write(response.content)

        print(f'Image téléchargée : {image\_name}')

    else:

        print(f'Échec du téléchargement de l\'image depuis {image\_url}')

# Fonction pour extraire les informations d'un livre

def extract\_book\_info(book\_url, category, image\_dir):

    response = requests.get(book\_url)

    if response.status\_code == 200:

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

        # Extraire les informations du livre en utilisant BeautifulSoup

        # Remplacez les sélecteurs CSS par ceux adaptés au site web en question

        title = soup.h1.text.strip()

        price\_including\_tax = soup.select('p.price\_color')[0].text

        price\_excluding\_tax = soup.select('p.price\_color')[1].text

        availability = soup.select('p.availability')[0].text.strip()

        product\_description = soup.select('meta[name="description"]')[0]['content']

        upc = soup.select('table tr')[0].find\_all('td')[0].text

        review\_rating = soup.select('p.star-rating')['class'][1]

        image\_url = 'http://books.toscrape.com' + soup.find('img')['src']

        # Télécharger l'image dans le répertoire correspondant à la catégorie

        download\_image(image\_url, image\_dir, category)

        book\_info = {

            'Title': title,

            'Price Including Tax': price\_including\_tax,

            'Price Excluding Tax': price\_excluding\_tax,

            'Availability': availability,

            'Product Description': product\_description,

            'UPC': upc,

            'Review Rating': review\_rating,

            'Image URL': image\_url,

        }

        return book\_info

    else:

        print(f'Erreur {response.status\_code} lors de la requête.')

        return None

# Fonction pour extraire les informations de tous les livres d'une catégorie et les écrire dans un fichier CSV

def extract\_and\_write\_category\_books(category\_name, category\_url, image\_dir):

    response = requests.get(category\_url)

    if response.status\_code == 200:

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

        book\_links = soup.select('h3 a')

        category\_info = []

        for book\_link in book\_links:

            book\_url = 'http://books.toscrape.com/catalogue' + book\_link['href'][8:]

            book\_info = extract\_book\_info(book\_url, category\_name, image\_dir)

            if book\_info:

                category\_info.append(book\_info)

        # Créer un répertoire pour les fichiers CSV

        csv\_dir = 'csv\_files'

        if not os.path.exists(csv\_dir):

            os.makedirs(csv\_dir)

        # Créer un fichier CSV pour la catégorie

        csv\_filename = os.path.join(csv\_dir, f'{category\_name}.csv')

        with open(csv\_filename, 'w', newline='', encoding='utf-8') as csv\_file:

            csv\_writer = csv.DictWriter(csv\_file, fieldnames=[

                'Title', 'Price Including Tax', 'Price Excluding Tax', 'Availability',

                'Product Description', 'UPC', 'Review Rating', 'Image URL'

            ])

            csv\_writer.writeheader()

            csv\_writer.writerows(category\_info)

        print(f'Données de la catégorie {category\_name} extraites et enregistrées dans {csv\_filename}')

# Extraire les liens vers toutes les catégories

category\_links = extract\_category\_links()

# Pour chaque catégorie, extraire les données et écrire dans un fichier CSV distinct

for category\_name, category\_url in category\_links:

    extract\_and\_write\_category\_books(category\_name, category\_url, 'images')

○ Sauvegarder les fichiers images localement.

N.B : Vous devez étendre la partie du script qui extrait les détails d'un livre spécifique pour télécharger et enregistrer le fichier image du livre.

● Vous devrez choisir un module Python qui vous permet de télécharger les fichiers images.

● Portez une attention particulière aux chemins relatifs et à la façon dont vous devez les convertir en chemins absolus pour que le script puisse les télécharger