**III- : Extraire toutes les données des produits d’une catégorie**

Écrit un script qui peut réussir à :

○ Extraire les données de tous les livres de la catégorie sélectionnée.

N'importe quelle catégorie de livres (listée dans la colonne de gauche de la page d'accueil) sur Books to Scrape et écrire un script Python qui visite la page de cette catégorie et extrait l'URL de la page Produit pour chaque livre de la catégorie.

○ Stocker ces données dans un fichier local au format CSV.

Combinez ce script avec ce que vous avez effectué lors de l'étape précédente I pour extraire les données produit de chaque livre de la catégorie choisie et pour inscrire les données dans un seul fichier CSV

Donnéescategorycsv.py (à

#Extraire les données de tous les livres de la catégorie sélectionnée + CSV repertoriés + sauv image répértorié

import requests

from bs4 import BeautifulSoup

import csv

import os

# 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')

        product\_page\_url = book\_url

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

        title = soup.h1.text.strip()

        price\_including\_tax = soup.select('table tr')[3].find\_all('td')[2].text

        price\_excluding\_tax = soup.select('table tr')[2].find\_all('td')[2].text

        number\_available = soup.select('table tr')[5].find\_all('td')[2].text

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

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

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

        book\_info = {

            'product\_page\_url': product\_page\_url,

            'universal\_product\_code (upc)': upc,

            'title': title,

            'price\_including\_tax': price\_including\_tax,

            'price\_excluding\_tax': price\_excluding\_tax,

            'number\_available': number\_available,

            'product\_description': product\_description,

            'category': category,

            'review\_rating': review\_rating,

            'image\_url': image\_url

        }

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

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

        return book\_info

    else:

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

        return None

# 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 de tous les livres d'une catégorie

def extract\_books\_in\_category(category\_url, category\_name, csv\_writer, image\_dir):

    response = requests.get(category\_url)

    if response.status\_code == 200:

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

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

        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:

                csv\_writer.writerow(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 répertoire pour les images

image\_dir = 'images'

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

    os.makedirs(image\_dir)

# URL de la catégorie sélectionnée (remplacez par l'URL de la catégorie de votre choix)

category\_url = 'http://books.toscrape.com/catalogue/category/books/romance\_8/index.html'

category\_name = 'Romance'  # Nom de la catégorie

# Créer un fichier CSV pour la catégorie sélectionnée

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

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

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

        'product\_page\_url', 'universal\_product\_code (upc)', 'title',

        'price\_including\_tax', 'price\_excluding\_tax', 'number\_available',

        'product\_description', 'category', 'review\_rating', 'image\_url'

    ])

    csv\_writer.writeheader()

    extract\_books\_in\_category(category\_url, category\_name, csv\_writer, image\_dir)

print(f'Données extraites et enregistrées dans {csv\_filename}')