**IV- Extraire toutes les produits de toutes les catégories**

Écrit un script qui peut réussir à :

○ Extraire les données de tous les livres de toutes les catégories.

Écrivez un script qui visite la page d'accueil de Books to Scrape et extrait les liens vers toutes les catégories de livres disponibles.

Combinez ce script avec ce que vous avez effectué lors de l'étape précédente. Le code doit visiter la page de chaque catégorie et extraire les données produit pour tous les livres de chaque catégorie.

○ Générer un fichier au format CSV pour chaque catégorie.

Inscrivez les données de chaque catégorie de livre dans un fichier CSV distinct

CODE

import requests

from bs4 import BeautifulSoup

import csv

import pandas as pd

import os

URL = 'https://books.toscrape.com/'

response = requests.get(URL)

page\_contents = response.text

doc = BeautifulSoup(page\_contents,'html.parser')

''' I- nous allons collecté des informations sur la page unique (ou) la première page de notre site Web.

Scrape\_Single\_Page collecte les informations de la première page.'''

#-----Saisir des titres de livre-----------

def get\_book\_titles(doc):

    Book\_title\_tags = doc.find\_all('h3')

    Book\_titles = []

    for tags in Book\_title\_tags:

        Book\_titles.append(tags.text)

    return Book\_titles

get\_book\_titles(doc)

# ----------Saisir les prix de chaque livre------------

def get\_book\_price(doc):

    Book\_price\_tags = doc.find\_all('p', class\_ = 'price\_color')

    Book\_price = []

    for tags in Book\_price\_tags:

        Book\_price.append(tags.text.replace('Â',''))

    return Book\_price

get\_book\_price(doc)

#-----Saisir la disponibilité des stocks------------

def get\_stock\_availability(doc):

    Book\_stock\_tags = doc.find\_all('p', class\_ = 'instock availability')

    Book\_stock = []

    for tags in Book\_stock\_tags:

        Book\_stock.append(tags.text.strip())

    return Book\_stock

get\_stock\_availability(doc)

# --------Saisir des liens pour chaque livre-------------

def get\_book\_url(Book\_title\_tags):

    Book\_url = []

    for article in Book\_title\_tags:

        for link in article.find\_all('a', href = True):

            url = link['href']

            links = 'https://books.toscrape.com/' + url

            if links not in Book\_url:

                Book\_url.append(links)

    return Book\_url

#--Collecte de données à partir de plusieurs pages et stockez dans Pandas DataFrame---

'''II- nous allons créer une fonction Scrape\_Multiple\_Pages pour

collecter les informations à partir de plusieurs pages.'''

def get\_doc(url):

    response = requests.get(url)

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

    if response.status\_code != 200:

        raise Exception('Failed to load page {}'.format(response))

    return doc

def scrape\_multiple\_pages(n):

    URL = 'https://books.toscrape.com/catalogue/page-'

    titles,prices,stocks\_availability,urls = [],[],[],[]

    for page in range(1,n+1):

        doc = get\_doc(URL + str(page)+ '.html')

        titles.extend(get\_book\_titles(doc))

        prices.extend(get\_book\_price(doc))

        stocks\_availability.extend(get\_stock\_availability(doc))

        urls.extend(get\_book\_url(doc.find\_all('h3')))

    book\_dict1 = {

                'TITLE':titles,

                'PRICE':prices,

                'STOCK AVAILABILTY':stocks\_availability,

                'URL':urls}

    return pd.DataFrame(book\_dict1)

#----Creating a CSV file------

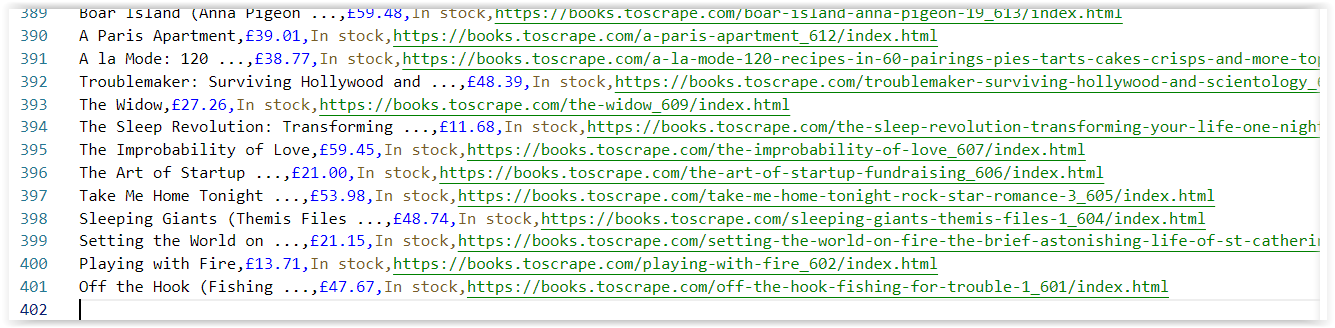
scrape\_multiple\_pages(20).to\_csv('books111.csv',index = None)

print("Tous les produits de toutes les catégories ont été extraits et enregistrés dans books111.csv")

Réponse



…



Ou

Scrap15.py

import requests

from bs4 import BeautifulSoup

import csv

import pandas as pd

import os

URL = 'https://books.toscrape.com/catalogue/page-'

books=[]

Price=[]

Stock\_availability=[]

Books\_url=[]

# Pour que la boucle rampe et obtienne des informations à partir de plusieurs pages.

for page in range(1,6):

    Source = requests.get(URL + str(page) + '.html')

    Scrape = BeautifulSoup(Source.text,'html.parser')

    #print(Scrape.prettify())

    # Saisir des titres de livre

    for article in Scrape.find\_all('article'):

        books.append(article.h3.a.text)

        #print(books)

    # Prix du livre

        Price.append(article.find('div', class\_ = 'product\_price').p.text.replace('Â',''))

        #print(Price)

    # Saisir la disponibilité des stocks de chaque livre

        Stock = Scrape.find('p', class\_ = 'instock availability').text.strip()

        Stock\_availability.append(Stock)

        #print(Stock\_availability)

    # Saisir le lien de chaque livre

        for link in article.find\_all('a',href = True):

            url = link['href']

        Books\_url.append('https://books.toscrape.com/catalogue/' + url)

         #print(Books\_url)

Scraped\_Data = {'TITRE':books,'PRIX':Price,'DISPONIBILITÉ STOCK':Stock\_availability,'URL':Books\_url}

Scraped\_Books = pd.DataFrame(Scraped\_Data)

Scraped\_Books[:4]

Scraped\_Books.to\_csv('scrap15.csv',index = None)

print("Données extraites avec succès et enregistrées dans scrap15.csv")

print(Scraped\_Books)

reponse scrap15.csv



…..

