WebScraping Project using Python

Source : DM Automobiles' website(The website hosts listings for car re-sales.)

Resources Used: Jupyter Notebooks(for Python), Beautiful Soup Library(for HTML parsing)

Objective & Approach : The approch behind executing this project has been to use python to scrape data of the source url and to provide the user with a csv containing the crucial data hosted by the website, viz The Make, Model, and Prices of the cars. Care has been taken to ensure the usability of the csv generated so that, the CSV is ready for analysis of the elements inside it. This includes removing irrelevant whitespaces and representing the data is in a access friendly format.

CSV Reommendations : To ensure best results, The recommended delimiter is " , ". The Decimal and Thousandths place separator must be set to standard EU currency format. (example 41,900 €)

Entrée [8]:
```
import requests # sedning requests to the URL
from bs4 import BeautifulSoup as bs # #Python parsing library for HTML
import os #for OS operations
import re # for removing whitespace from before the prices
import csv #for csv operations
```

Entrée [2]:
```
source = requests.get('https://pros.lacentrale.fr/C018357/?pro_only=0?pro_only=0&fromLC=true&fromLCHeader=true&max=100')
soup = bs(source, 'html.parser')
```

Entrée [9]:
```
csv_file = open('dataExport.csv', 'w', newline='')
csv_writer = csv.writer(csv_file)
csv_writer.writerow(['Make', 'Model', 'Prices'])
csv_file.close()
```

Entrée [10]:
```
#gives us the make and model of all cars
'''The general approach has been to visually see where these elements are located using browser's web inspecting tools
and then to grab the tag of the required element and call teh .text method on it to get all the text.
After grabbing the text I sliced text accordingly to separate the make and modele of the car and then append all this
data into their dedicted lists'''
import string
make_Names = []
model_Names= []
Prices_Cars = []
for var1 in soup.find_all('h3', class_ = 'brandModelTitle'):
    car_Names = var1.text
    test_Split = car_Names.split("\n")
    full_Names = test_Split[1:3]
    make = test_Split[1:2]
    model = test_Split[2:3]
    make_Names.append(make)
    model_Names.append(model)
```

Entrée [11]:
```
# Gives the prices of all cars
'''Utilizes similar approch used to grab the names, here we target a different class where the price lies.
I also removed white spaces that ensures usefullness of the generated Prices, subsequently the CSV. '''
for Prices in soup.find_all('span', class_ = 'f20 bold fieldPrice'):
    Prices = Prices.span.text
    Prices = re.sub("^\s+|\s+$", "",Prices, flags=re.UNICODE) # removing whitespace before the prices
    Prices_Cars.append(Prices)
```

Entrée [12]:

```
csv_file = open('dataExport.csv', 'a', newline='')
# newline='' used to avoid generation of blank rows in the generated csv
csv_writer = csv.writer(csv_file)
i = 0
while i < len(Prices_Cars):
    csv_writer.writerow([*make_Names[i], *model_Names[i], Prices_Cars[i]])
    # " * " used to unpack make_names, model_Names, remove the nested list
    i = i + 1

csv_file.close()
```

Entrée [13]:

```
# pwd
'''Uncomment _pwd_ to find the directory of the generated CSV'''
```

Out[13]: 'Uncomment _pwd_ to find the directory of the generated CSV'