IT2381 Web Scraping Using Python Extra Exercise

Exercise 1:

BeautifulSoup

```python
# import necessary libarary
import requests
import pandas as pd

from bs4 import BeautifulSoup

# URL for the webpage
url = "https://www.nytimes.com/books/best-sellers/combined-print-and-e-book-fiction/"

# Sending a request to the speciifed URL
resp = requests.get(url)

# Converting the response to Beautiful Soup Object
content = BeautifulSoup(resp.content, 'html')

books=content.find_all('li',{"class":"css-sggj6j"})

book_list = []
for book in books:

    # retrieve book title
    title=book.find('h3',{"class":"css-2jegzb"}).get_text(strip=True)

    # retrieve book price
    author=book.find('p',{"class":"css-1aaqvca"}).get_text(strip=True)

    data={
        "Title":title,
        "Author":author,
    }

    book_list.append(data)

book_list = pd.DataFrame(book_list)
print(book_list)

book_list.to_csv('NYTimes_book_BS.csv', index=False)
```

Selenium

```python
# import necessary libary
import requests
import pandas as pd

from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
from selenium import webdriver

from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException

url = "https://www.nytimes.com/books/best-sellers/combined-print-and-e-book-fiction/"

driver=webdriver.Chrome(service=Service(ChromeDriverManager().install()))
driver.get(url)

# Parse and Extract HTML content
books=driver.find_elements(By.CSS_SELECTOR, "li[class='css-sggj6j']" )
print(len(books))

book_list = []
for book in books:

    title = book.find_element(By.CSS_SELECTOR, "h3[class='css-2jegzb']").text

    author=book.find_element(By.CSS_SELECTOR,"p[class='css-1aaqvca']").text

    data={
        "Title":title,
        "Author":author,
    }

    book_list.append(data)

book_list = pd.DataFrame(book_list)
print(book_list)

driver.close()
book_list.to_csv('NewYork_Times_Selenium.csv', index=False)
```

Exercise 2:

BeautifulSoup

```
# import necessary libarary
import requests
import pandas as pd

from bs4 import BeautifulSoup

# URL for the webpage
url = "https://www.courts.com.sg/furniture/furniture/study-desks"

# Sending a request to the speciifed URL
resp = requests.get(url)

# Converting the response to Beautiful Soup Object
content = BeautifulSoup(resp.content, 'html')

products=content.find_all('li',{"class":"item product product-item"})

product_list = []
for product in products:

    # retrieve product namke
    prodName=product.find('h3',{"class":"product name product-item-name"}).get_text(strip=True)

    # retrieve current product price

    curPrice_content=product.find('span',{"class":"special-price"})
    if curPrice_content is not None:
        curPrice=curPrice_content.find('span',{"class":"price"}).get_text(strip=True)
    else:
        curPrice=product.find('span',{"class":"price"}).get_text(strip=True)

     # retrieve original product price
    oldPrice_content=product.find('span',{"class":"old-price"})
    if oldPrice_content is None:
        oldPrice=""
    else:
        oldPrice=oldPrice_content.find('span',{"class":"price"}).get_text(strip=True)

    data={
        "Name":prodName,
        "Original Price": oldPrice,
        "Current Price": curPrice,
    }

    product_list.append(data)

df_product = pd.DataFrame(product_list)
print(df_product)

df_product.to_csv('Courts.csv', index=False)
```

Selenium

```
# import necessary libary
import requests
import pandas as pd

from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.chrome.service import Service
from selenium import webdriver

from selenium.webdriver.common.by import By
from selenium.common.exceptions import NoSuchElementException

url = "https://www.courts.com.sg/furniture/furniture/study-desks"

driver=webdriver.Chrome(service=Service(ChromeDriverManager().install()))
driver.get(url)

# Parse and Extract HTML content

products=driver.find_elements(By.CSS_SELECTOR, "li[class='item product product-item']" )

product_list = []
for product in products:

    prodName = product.find_element(By.CSS_SELECTOR, "h3[class='product name product-item-name']").text

    try:
        curPrice_content=product.find_element(By.CSS_SELECTOR,"span[class='special-price']")
        curPrice=curPrice_content.find_element(By.CSS_SELECTOR,"span[class='price']").text
    except NoSuchElementException:
        curPrice=product.find_element(By.CSS_SELECTOR,"span[class='price']").text

    # retrieve original product price
    try:
        oldPrice_content=product.find_element(By.CSS_SELECTOR,"span[class='old-price']")
        oldPrice=oldPrice_content.find_element(By.CSS_SELECTOR,"span[class='price']").text
    except NoSuchElementException:
        oldPrice=""



    data={
        "Name":prodName,
        "Original Price": oldPrice,
        "Current Price": curPrice,
    }

    product_list.append(data)

df_product = pd.DataFrame(product_list)
print(df_product)

driver.close()
df_product.to_csv('Courts.csv', index=False)
```