Forward School

Program Code: J620-002-4:2020

Program Name: FRONT-END SOFTWARE DEVELOPMENT

Title : Exercise using BeautifulSoup and Selenium

Name: Phua Yan Han

IC Number: 050824070059

Date : 4/7/23

Introduction : learning how to use beautiful soup and selenium together

Conclusion : learned how to use both beautiful soup and selenium together

Exe09 - Exercise Using BeautifulSoup and Selenium on News Web Portal

Extract daily COVID-19 statistics from theStar

Location: https://www.thestar.com.my/news/nation/2020/03/23/covid-19-current-situation-in-malaysia-updated-daily (https://www.thestar.com.my/news/nation/2020/03/23/covid-19-current-situation-in-malaysia-updated-daily)

In [18]:
```
import requests
from bs4 import BeautifulSoup

url='https://www.thestar.com.my/news/nation/2020/03/23/covid-19-current-situati

# get the webpage
data = requests.get(url)

# load webpage into bs4
bs = BeautifulSoup(data.content, 'html.parser')
# get data simply by looking for all <a> links
bs.find_all('a')
```

Out[18]:
```
[<a class="navbar-brand brand-prime" data-content-id="https://www.thestar.c
om.my" data-content-title="The Star Online" data-content-type="Navigation"
data-list-type="Header" href="/">
 <svg aria-label="the star online" class="icon" height="55" role="img" widt
h="164">
 <image border="0" height="55" src="https://cdn.thestar.com.my/Themes/img/l
ogo-tsol-logov3.png" width="164" xlink:href="https://cdn.thestar.com.my/The
mes/img/logo-tsol-fullv3.svg"/>
 </svg>
 </a>,
 <a class="btn--subscribe" data-content-id="https://www.thestar.com.my/subs
cription" data-content-title="Subscription" data-content-type="Navigation"
data-list-type="Header" href="/subscription">Subscriptions</a>,
 <a class="login" data-content-id="https://sso.thestar.com.my/?lng=en&amp;c
hannel=1&amp;ru=HNQ8Auw31qgZZU47ZjHUhHKJStkK3H51/pPcFdJ1gQ9cFgPiSalasDvF6De
umuZwrPFzdYjofJj9eX1n44olyqGHD3HJYujVJKnBGSMMB/zfChfXgzd4SeyxRdNXN6ZWbrt8Vq
9CGyeRv3tJQMZkgrPs0PgxqXZTlEZW/jQG2aZ+b1eksd4EfiZDBUcWQcFYvs1m3Fkd04fguPM90
q6guFbCG4ZqfYK1HTduYl2eQNi53cvg+bra/Y0o0cgRGLoa7eTLY69YN/+roj7uviwmtQ==" da
ta-content-title="Log In" data-content-type="Outbound Referral" data-list-t
```

In [16]:
```
import requests
from bs4 import BeautifulSoup

url='https://www.thestar.com.my/news/nation/2020/03/23/covid-19-current-situati

# get the webpage
data = requests.get(url)

# load webpage into bs4
soup = BeautifulSoup(data.text,'html.parser')

data = []
# get data simply by looking for all <a> links
for a in soup.find_all('a'):
    data.append(a.text)
data
```

Out[16]:
```
['\n\n\n\n',
 'Subscriptions',
 '\n                        Log In\n                    ',
 '',
 'Manage Profile\n                                ',
 'Change Password\n                               ',
 'Manage Logins\n                              ',
 'Manage Subscription\n                                ',
 'Transaction History\n                                ',
 'Manage Billing Info\n                                ',
 'Manage For You\n                             ',
 'Manage Bookmarks\n                               ',
 'Package & Pricing\n                              ',
 'FAQs\n                                  ',
 'Log Out\n                                  ',
 '\n\n\n\n',
 '\n\n',
 '\n                    StarPlus\n                  ',
 '\n                    News\n                  ',
```

Check HTML code of the Web page again

Notice that there is an iFrame Tag highlighted above?

The actual location of the source web page is embeded within the iframe of theStar

Change the URL to the actual source.

In [4]:
```
import requests
from bs4 import BeautifulSoup

url='https://public.flourish.studio/visualisation/1641110/embed?auto=1'

# get the data
data = requests.get(url)

# load data into bs4
soup = BeautifulSoup(data.text,'html.parser')


data = []
# get data simply by looking for each a links
for tr in soup.find_all('div',attrs={'class': 'tr body-row'}):
    data.append(tr.text)
data
```

Out[4]:
```
[]
```

In [13]:
```
# soup.find_all('div')
soup.prettify
```

Out[13]:
```
<bound method Tag.prettify of <!DOCTYPE html>
<html><head>
<meta charset="utf-8"/>
<meta content="width=device-width, initial-scale=1" name="viewport"/>
<base target="_blank"/>
<link href="https://flo.uri.sh/template/1065/v24/static/style.css" rel="sty
lesheet" type="text/css"/>
<link href="https://fonts.googleapis.com/css?family=Source+Sans+Pro:400,70
0" rel="stylesheet" type="text/css"/>
<title>COVID-19 MALAYSIA TABLE</title></head>
<body>
<style id="cell-styling"></style>
<script>window.Flourish = {"static_prefix":"https://flo.uri.sh/template/106
5/v24/static","environment":"live"};</script><script>var template=function
(t){"use strict";var s={},f={table_min_width:300,table_border_color:"#aaaaa
a",table_border_width:0,sorting:{enabled:!0,order:"ascending",column_index:
null},reloader:{},color:{custom_palette:"Clinton:#1d6996\nTrump:#cc503e"},p
opup:{font_size:"1rem"},bar_columns:{enabled:!0,type:"bars",bar_1_column
s:"Clinton\nTrump",bar_1_column_name:"Vote share",bar_1_max_enabled:!1,bar_
```

Cannot Use BeautifulSoup

Check the Javascript found above.

The data for the table is within the Javascript coding.

2 options.

Option 1. Try to Scrape the Javascript. Not that possible, unless fully understand how the Javascript program going to output the HTML to the Web Page.

Option 2. Use Selenium Webdriver to run the Javascript within the webdriver and then scrape the HTML output.

In [6]:
```
# Use Selenium
from selenium import webdriver
from bs4 import BeautifulSoup

driver = webdriver.Chrome('chromedriver')

url='https://public.flourish.studio/visualisation/1641110/embed?auto=1'

# get the data
driver.get(url)

# load data into bs4
soup = BeautifulSoup(driver.page_source,'html.parser')

data = []
# get data simply by looking for each a links
for tr in soup.find_all('div',attrs={'class': 'tr body-row'}):
    data.append(tr.text)
driver.close()
data
```

Out[6]:
```
['22-Apr-21\n384688\n2875\n1407\n361267\n',
 '21-Apr-21\n381813\n2340\n1400\n358726\n',
 '20-Apr-21\n379473\n2341\n1389\n356816\n',
 '19-Apr-21\n377132\n2078\n1386\n355224\n',
 '18-Apr-21\n375054\n2195\n1378\n353822\n',
 '17-Apr-21\n372859\n2331\n1370\n352395\n',
 '16-Apr-21\n370528\n2551\n1365\n350563\n']
```

In [11]:
```
from selenium import webdriver
from bs4 import BeautifulSoup

driver = webdriver.Chrome('chromedriver')

url='https://public.flourish.studio/visualisation/1641110/embed?auto=1'

# get the data
driver.get(url)

# load data into bs4
soup = BeautifulSoup(driver.page_source,'html.parser')

data = []
# get data simply by looking for each a links
for tr in soup.find_all('div',attrs={'class': 'tr body-row'}):
    for td in soup.find_all('div',attrs={'class':'td'}):
        data.append(td.text)

driver.close()
data
```

Out[11]:
```
['Date',
 'Total cases',
 'New cases',
 'Total deaths',
 'Total recovered',
 '22-Apr-21\n',
 '384688\n',
 '2875\n',
 '1407\n',
 '361267\n',
 '21-Apr-21\n',
 '381813\n',
 '2340\n',
 '1400\n',
 '358726\n',
 '20-Apr-21\n',
 '379473\n',
 '2341\n',
 '1389\n',
```

In [13]:
```
from selenium import webdriver
from bs4 import BeautifulSoup

driver = webdriver.Chrome('chromedriver')

url='https://public.flourish.studio/visualisation/1641110/embed?auto=1'

# get the data
driver.get(url)

# load data into bs4
soup = BeautifulSoup(driver.page_source,"html.parser")

data = []
# get data simply by looking for each a links
for tr in soup.find_all('div',attrs={'class': 'tr body-row'}):
    for td in soup.find_all('div',attrs={'class':'td'}):
        data.append(td.text.rstrip())

data
```

Out[13]:
```
['Date',
 'Total cases',
 'New cases',
 'Total deaths',
 'Total recovered',
 '22-Apr-21',
 '384688',
 '2875',
 '1407',
 '361267',
 '21-Apr-21',
 '381813',
 '2340',
 '1400',
 '358726',
 '20-Apr-21',
 '379473',
 '2341',
 '1389',
```

In [14]:
```
# Next Page

driver.find_element_by_xpath('/html/body/main/section[4]/div[1]/div/div[4]/butt

soup = BeautifulSoup(driver.page_source,'html.parser')

data=[]
# get data simply by looking for each a links
for tr in soup.find_all('div',attrs={'class':'tr body-row'}):
    for td in soup.find_all('div',attrs={'class':'td'}):
        data.append(td.text)
data

# depends
# if first time scrape, must scrape all previous pages. then paginate and get t
# if only need to get the latest everyday, then no need to grab the same data a

# look at this class="pagination-total"
```

Out[14]:
```
['Date',
 'Total cases',
 'New cases',
 'Total deaths',
 'Total recovered',
 '15-Apr-21\n',
 '367977\n',
 '2148\n',
 '1363\n',
 '349039\n',
 '14-Apr-21\n',
 '365829\n',
 '1889\n',
 '1353\n',
 '347780\n',
 '13-Apr-21\n',
 '363940\n',
 '1767\n',
 '1345\n',
```

Footnote:

HTML iframe tag

Specification:

https://www.w3.org/html/wg/spec/the-iframe-element.html (https://www.w3.org/html/wg/spec/the-iframe-element.html)

EXERCISE:

-Scrape table on this URL: ""
-Use Selenium to scrape data
-Scrape data from 1st Jan 2021 until 20th Mar 2021
-Use drive.click() to navigate pagination
-Feel free to drop me questions/Google/refer notes during this exercise.

In [18]:
```
# 457

from selenium import webdriver
from bs4 import BeautifulSoup
import time
import pandas as pd
from datetime import datetime

driver = webdriver.Chrome('C:\\Users\\Asus\\Documents\\ChromeDriver\\chromedriv

url = 'https://public.flourish.studio/visualisation/1641110/embed?auto=1'

driver.get(url)
keepRunning = True
data = []

while keepRunning:
    soup = BeautifulSoup(driver.page_source, "html.parser")
    for tr in soup.find_all('div', attrs={'class': 'tr body-row'}):
        for td in tr.find_all('div', attrs={'class':'td'}):
            data.append(td.text.rstrip())

    button = driver.find_element_by_class_name('pagination-btn.next')
    if "disable" in button.get_attribute("class"):
        break
    else:
        button.click()
data
num_columns = 5  # Replace this with the actual number of columns in your data
num_rows = len(data) // num_columns

# Reshape the data into a 2D array
data_2d = [data[i*num_columns : (i+1)*num_columns] for i in range(num_rows)]

df = pd.DataFrame(data_2d, columns=['Date', 'Total Cases', 'New Cases', 'Total
print(df)


start_date = datetime.strptime('01-Jan-21', '%d-%b-%y')
end_date = datetime.strptime('20-Mar-21', '%d-%b-%y')
df['Date'] = pd.to_datetime(df['Date'], format='%d-%b-%y')
df[(df['Date'] >= start_date) & (df['Date'] <= end_date)]
```

```
          Date Total Cases New Cases Total Deaths Total Recovered
0    22-Apr-21      384688       2875         1407          361267
1    21-Apr-21      381813       2340         1400          358726
2    20-Apr-21      379473       2341         1389          356816
3    19-Apr-21      377132       2078         1386          355224
4    18-Apr-21      375054       2195         1378          353822
..         ...         ...        ...          ...             ...
452  26-Jan-20           4          0            0               0
453  25-Jan-20           4          4            0               0
454  24-Jan-20           0          0            0               0
455  23-Jan-20           0          0            0               0
456  22-Jan-20           0          0            0               0

[457 rows x 5 columns]
```

Out[18]:

	Date	Total Cases	New Cases	Total Deaths	Total Recovered
33	2021-03-20	331713	1671	1229	316042
34	2021-03-19	330042	1576	1225	314457
35	2021-03-18	328466	1213	1223	312461
36	2021-03-17	327253	1219	1220	310958
37	2021-03-16	326034	1063	1218	309612
...	...	...	...	...	...
107	2021-01-05	122845	2027	509	99449
108	2021-01-04	120818	1741	501	98228
109	2021-01-03	119077	1704	494	97218
110	2021-01-02	117373	2295	483	94492
111	2021-01-01	115078	2068	474	91171

79 rows × 5 columns