Buổi 3. Thu thập thông tin với Beautiful Soup

1 Thông tin chung

Mục tiêu buổi học

- Giới thiệu thư viện Beautiful Soup.
- Hướng dẫn cài đặt và sử dụng thư viện Beautiful Soup để thu thập thông tin Web.

Kiến thức và kỹ năng đạt được

- Nắm vững và sử dụng được các đối tượng trong thư viện Beautiful Soup.
- Áp dụng cài đặt được các bài tập thực hành.

Công cụ thực hành

- Ngôn ngữ lập trình: Python
- Công cụ thực hành: Anaconda, colab

Thời gian thực hành: 3 tiết

2 Nội dung lý thuyết

Beautiful Soup

- Là bộ thư viện thu thập dữ liệu từ các trang HTML, XML.
- Hỗ trợ bộ phân tích cú pháp HTML (html.parser), XML (lxml.parser).
- Đơn giản, dễ sử dụng.

Cài đặt thư viện BeautifulSoup

```
pip install BeautifulSoup4
```

3 Nội dung thực hành

3.1 Lấy thông tin từ mã nguồn HTML tĩnh

[78]:
```
from bs4 import BeautifulSoup
html_doc = """<!DOCTYPE html><html><body><p><a id="link1"
href="www.3schools.com">www.3schools.com</a><a id="link2"
href="https://developer.mozilla.org">
https://developer.mozilla.org</a></p><p>This is a paragraph.</p><p>
This is another paragraph</p></body></html>"""
```

```
#
soup = BeautifulSoup(html_doc, "html.parser")
```

[79]:
```
print(soup.prettify())
```

```
<!DOCTYPE html>
<html>
 <body>
  <p>
   <a href="www.3schools.com" id="link1">
    www.3schools.com
   </a>
   <a href="https://developer.mozilla.org" id="link2">
    https://developer.mozilla.org
   </a>
  </p>
  <p>
   This is a paragraph.
  </p>
  <p>
   This is another paragraph
  </p>
 </body>
</html>
```

[80]:
```
print(soup.find(id="link1"))
```

```
<a href="www.3schools.com" id="link1">www.3schools.com</a>
```

[81]:
```
print(soup.find_all(name="a"))
```

```
[<a href="www.3schools.com" id="link1">www.3schools.com</a>, <a
href="https://developer.mozilla.org" id="link2">
https://developer.mozilla.org</a>]
```

[82]:
```
p_tag = soup.p
print(p_tag)
```

```
<p><a href="www.3schools.com" id="link1">www.3schools.com</a><a
href="https://developer.mozilla.org" id="link2">
https://developer.mozilla.org</a></p>
```

[83]:
```
a_tag = p_tag.a
print(a_tag)
```

```
<a href="www.3schools.com" id="link1">www.3schools.com</a>
```

[84]:
```
print(a_tag.name, a_tag.attrs, a_tag.string)
```

```
a {'id': 'link1', 'href': 'www.3schools.com'} www.3schools.com
```

[85]:
```
name_parents_a_tag = [tag.name for tag in a_tag.parents]
print(name_parents_a_tag)
```

```
['p', 'body', 'html', '[document]']
```

[86]:
```
siblings_p_tag = [tag for tag in p_tag.next_siblings]
# siblings_p_tag = list(p_tag.next_siblings)
print(siblings_p_tag)
```

```
[<p>This is a paragraph.</p>, <p>
This is another paragraph</p>]
```

[87]:
```
list_id = [tag.attrs["id"] for tag in p_tag.children]
print(list_id)
```

```
['link1', 'link2']
```

3.2 Lấy thông tin trang Web trực tuyến

Ví dụ: Lấy về GDP của tất cả các quốc gia trên thế giới.

- Lấy về mã nguồn trang Web.

[88]:
```
from bs4 import BeautifulSoup
from urllib.request import urlopen

url = 'https://en.wikipedia.org/wiki/List_of_countries_by_GDP_(nominal)'
html_doc = urlopen(url).read()
soup = BeautifulSoup(html_doc, 'html.parser')
```

[89]:
```
# Lấy tiêu đề trang web
soup.title
```

[89]:
```
<title>List of countries by GDP (nominal) - Wikipedia</title>
```

- Cần đọc mã nguồn trang Web và phân tích cấu trúc của nó.

[90]:
```
table_tag = soup.find("table", attrs = {"class": "wikitable"})
rows = table_tag.find("tbody").find_all("tr")

# rows[1]
```

[91]:
```
# Tại dòng 0, lấy về tiêu đề 3 bảng GDP
table_headers = []
for td in rows[0].find_all("td"):
    table_headers.append(td.b.text.replace('\n', ' ').strip())
```

```
table_headers
```

[91]:
```
['Per the International Monetary Fund (2019 estimates)',
 'Per the World Bank (2019)',
 'Per the United Nations (2018)']
```

[92]:
```
# Tại dòng 1, lấy về tiêu đề 3 cột dữ liệu
columns = []
for th in rows[1].find("table").find_all("th"):
    columns.append(th.text.replace('\n', ' ').strip())

columns
```

[92]:
```
['Rank', 'Country/Territory', 'GDP(US$million)']
```

[93]:
```
# Duyệt qua từng dòng lấy về thông tin tương ứng với cột Ranks, Countrys, GDP
data = {}
for table, header in zip(rows[1].find_all("table"), table_headers):
    table_data = []
    for tr in table.tbody.find_all("tr"):
        row = {}
        for td, th in zip(tr.find_all("td"), columns):
            row[th] = td.text.replace('\n', '').strip()
        table_data.append(row)
    data[header] = table_data

# data
# data['Per the International Monetary Fund (2019 estimates)']
# data.items()
```

[94]:
```
import pandas as pd
```

[95]:
```
df_international = pd.DataFrame(data[table_headers[0]])
df_international
```

[95]:

	Rank	Country/Territory	GDP(US$million)
0	NaN	NaN	NaN
1		World[19]	87,265,226
2	1	United States	21,439,453
3	-	European Union[23][n 1]	18,705,132
4	2	China[n 2]	14,140,163
..	...	...	...
190	182	Palau	291
191	183	Marshall Islands	220
192	184	Kiribati	184
193	185	Nauru	108
194	186	Tuvalu	42

```
[195 rows x 3 columns]
```

[96]:
```
df_work_bank = pd.DataFrame(data[table_headers[1]])
df_work_bank
```

[96]:
```
    Rank Country/Territory GDP(US$million)
0    NaN               NaN             NaN
1                    World      87,751,541
2      1    United States      21,427,700
3      2       China[n 5]      14,342,903
4      3            Japan       5,081,770
..   ...              ...             ...
187  180            Palau      284 (2018)
188  181  Marshall Islands      221 (2018)
189  182         Kiribati             195
190  183            Nauru             118
191  184           Tuvalu              47

[192 rows x 3 columns]
```

[97]:
```
df_us = pd.DataFrame(data[table_headers[2]])
df_us
```

[97]:
```
    Rank Country/Territory GDP(US$million)
0    NaN               NaN             NaN
1              World[25]      85,085,189
2      1    United States      20,580,223
3      2       China[n 5]      13,608,152
4      3            Japan       4,971,323
..   ...              ...             ...
210  190  Marshall Islands             214
211  191         Kiribati             189
212  192            Nauru             127
213    -       Montserrat              64
214  193           Tuvalu              46

[215 rows x 3 columns]
```

- Lưu dữ liệu 3 bảng vào 3 tập tin *.csv

[98]:
```
df_international.to_csv(table_headers[0] + '.csv')
df_work_bank.to_csv(table_headers[1] + '.csv')
df_us.to_csv(table_headers[2] + '.csv')
```

[99]:
```
#import csv
```

```
# for name, table in data.items():
#     with open(f"{name}.csv", 'w') as out_file:
#         writer = csv.DictWriter(out_file, headers)
#         writer.writeheader()
#         for row in table:
#             if row:
#                 writer.writerow(row)
```

4 Bài tập

- Lấy về thông tin sách (tên sách, tác giả, rating, giá bán) của những sách đang bán chạy trên trang Amazon: https://www.amazon.com/gp/bestsellers/books/