

2. 웹 정보 수집 - BeautifulSoup 이해

- history
 - v12 : 웹 기본이해 delete 2020/09

웹 정보수집 라이브러리

- 웹 데이터를 요청 - requests, urllib(내장 모듈)
- HTML 소스코드를 Python에서 쉽게 사용가능하도록 구조화- BeautifulSoup
- 동적 웹 수집 라이브러리 - selenium

01. 간단한 정보 가져오기 실습

- BeautifulSoup 는 파이썬 라이브러리입니다.
- HTML 및 XML 파일에서 데이터를 추출하기 위한 Python 라이브러리입니다.

1-1 기본 예제(lxml)

- lxml : 파서(Parser), 원시코드인 순수 문자열 객체를 해석할 수 있도록 분석.
 - lxml : c로 구현된 가장 빠름.
 - html5lib : 웹 브라우저 방식으로 HTML 해석
 - html.parser

In [1]:



```
from bs4 import BeautifulSoup

html = "<p>test</p>"
soup = BeautifulSoup(html, 'lxml')
print(soup)
```

```
<html><body><p>test</p></body></html>
```

1-2 기본 예제(html5lib)

In [2]:



```
from bs4 import BeautifulSoup

html = "<p>test</p>"
soup = BeautifulSoup(html, 'html5lib')
print(soup)
soup.prettify()
```

```
<html><head></head><body><p>test</p></body></html>
```

Out[2]:

```
'<html>\n <head>\n </head>\n <body>\n  <p>\n    test\n  </p>\n </body>\n</html>'
```

1-3 lxml과 html5lib

- lxml - Processing XML and HTML with Python
- lxml은 Python언어로 XML과 HTML을 처리할 수 있는 가장 기능이 풍부하고 쉬운 라이브러리이다.
- html5lib는 HTML 구문 분석을 위한 순수 파이썬 라이브러리이다.
- lxml이 html5lib에 비해 속도가 빠르다.(C언어와 Python)

1-4 HTML에서 정보 가져오기

- title 태그 정보 가져오기

In [3]:



```
from bs4 import BeautifulSoup

html = """
<html>
<head><title>나의 웹페이지</title></head>
<p>test1</p>
<p>test2</p>
<p>test3</p>
</html>
"""
```

In [4]:



```
soup = BeautifulSoup(html, 'lxml')
tag_title = soup.title # soup 내부의 title 정보를 가져온다. 가정 첫번째 것만 해당됨.
print(tag_title.text)  # 정보
print(tag_title.name)
print(type(soup))
```

나의 웹페이지

title

<class 'bs4.BeautifulSoup'>

1-5 HTML에서 정보 가져오기

- p 태그 정보 가져오기
- p 태그의 속성 정보 가져오기
- p 태그의 정보를 id로 가져오기
- p 태그의 정보를 class로 가져오기

In [5]:



```
from bs4 import BeautifulSoup

html = """
<html>
<head><title> test site </title></head>
<p class='class1' align="left">test3</p>
<p class='class1'>test2</p>
<p id='p1'>오늘의 주가지수 1500</p>
<span class='class3'>span tag text</span>
<p class='class4'>test3</p>
</html>
"""
```

In [6]:



```
soup = BeautifulSoup(html, 'lxml')
soup.p
```

Out[6]:

```
<p align="left" class="class1">test3</p>
```

속성 정보 가져오기

In [7]:



```
soup.p.attrs
```

Out[7]:

```
{'class': ['class1'], 'align': 'left'}
```

In [8]:



```
# 속성의 값을 가져오기
soup.p['align']
```

Out[8]:

```
'left'
```

In [9]:

```
## p 태그의 정보를 id로 가져오기
soup.p['align'] # p 태그내의 텍스트의 정렬 정보(align)가져오기

# span 태그의 정보를 가져오기
print( soup.span )
print( soup.span.attrs)
print( soup.span.text)
print( soup.span['class'])
```

```
<span class="class3">span tag text</span>
{'class': ['class3']}
span tag text
['class3']
```

In [10]:

```
### 1-6 만약 정보를 가져올 때, 에러가 발생하면 어떻게 될까?
soup = BeautifulSoup(html, 'lxml')
tag_p = soup.p
tag_title = soup.title

print(tag_p.attrs)
```

```
{'class': ['class1'], 'align': 'left'}
```

In [11]:

```
print(tag_p['style']) # style가 없어 에러 발생
```

```
-----
-
KeyError                                Traceback (most recent call last)
<ipython-input-11-888641df517d> in <module>()
----> 1 print(tag_p['style']) # style가 없어 에러 발생

~\Anaconda3\lib\site-packages\bs4\element.py in __getitem__(self, key)
    1069     """tag[key] returns the value of the 'key' attribute for the tag,
    1070     and throws an exception if it's not there."""
-> 1071     return self.attrs[key]
    1072
    1073     def __iter__(self):

KeyError: 'style'
```

In [12]:

```
print(tag_p.get('style')) # get 을 이용하면 이에 대한 에러를 방지할 수 있다.
```

None

1-6 HTML에서 정보 가져오기

- 태그와 태그 사이의 text 정보를 가져오기

- text와 string 를 이용하기
- text는 태그들의 하위 내용까지 값 전체 출력
- string은 정확히 선택된 태그에 대해서만 값 출력

In [13]:



```
from bs4 import BeautifulSoup
html = """
<html>
<head><title> text와 string의 차이 </title></head>
<p>
<span>test1</span>
<span>test2</span>
<span><b>test3</b></span>
</p>
</html>
"""
soup = BeautifulSoup(html, 'lxml')
tag_p = soup.p # soup 내부의 title 정보를 가져온다. 가정 첫번째 것만 해당됨.

data_text = tag_p.text
data_string = tag_p.string
data_span_str = tag_p.span.string

# text를 이용한 하위 정보 전체 출력
print(data_text, type(data_text) )

# string을 이용한 현재 내용에 대해서만 출력
print(data_string, type(data_string) )

# string을 이용한 span 태그의 첫번째 줄에 대해서만 출력
print(data_span_str)
```

```
test1
test2
test3
<class 'str'>
None <class 'NoneType'>
test1
```

1-7 find와 find_all를 이용하기

- open을 이용하여 파일을 열기
 - open([파일명], [읽기모드/쓰기모드], encoding='인코딩방식')
- p 태그의 전체 정보를 가져오기

In [14]:



```
from bs4 import BeautifulSoup

page = open("mypage.html", 'r', encoding="utf-8").read()
page
```

Out [14]:

```
'<html>
<title>나의 홈페이지</title>
<body>
  <div>
    <a href="https://www.naver.com/">naver</a>
    <a href="https://www.google.com/">google</a>
    <p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
    <p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
    <p class=W'p3W'> 강아지 사진과 네이버 링크 p3 그룹 1</p>
    <p class=W'p3W'> 강아지 사진과 네이버 링크 p3 그룹 2</p>
    <p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
    <pre>
      내가 가장 좋아하는 동물은 강아지입니다.
      나는 그리고 네이버 홈페이지에 자주 갑니다.
    </pre>
  </div>
  <div>
    <p class=W'p3W'> 강아지 사진과 네이버 링크 p3 그룹 3</p>
    <p class=W'p3W'> 강아지 사진과 네이버 링크 p3 그룹 4</p>
    <p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
  </div>
</body>
</html>'
```

In [15]:



```
soup = BeautifulSoup(page, 'lxml')
soup
```

Out [15]:

```
<html>
<head>
  <title>나의 홈페이지</title>
</head>
<body>
  안녕하세요
  <div>
    <a href="https://www.naver.com/">naver</a>
    <a href="https://www.google.com/">google</a>
    <p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
    <p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
    <p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
    <p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
    <p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
    <pre>
      내가 가장 좋아하는 동물은 강아지입니다.
      나는 그리고 네이버 홈페이지에 자주 갑니다.
    </pre>
  </div>
  <div>
    <p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>
    <p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>
    <p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
  </div>
</body>
</html>
```

실습1

- p태그의 id가 'p4_only'인 정보를 가지고 와 주세요.
- p태그의 class가 'p3'인 정보를 가지고 와 주세요.
- a태그의 href의 속성 정보를 가지고 와 주세요.
- pre태그의 정보를 가지고 와 주세요

In [16]:



```
print(soup.prettify())
```

```
<html>
<head>
  <title>
    나의 홈페이지
  </title>
</head>
<body>
  안녕하세요
  <div>
    <a href="https://www.naver.com/">
      naver
    </a>
    <a href="https://www.google.com/">
      google
    </a>
    <p>
      내가 가장 좋아하는 동물은 강아지입니다.
    </p>
    <p>
      나는 그리고 네이버 홈페이지에 자주 갑니다.
    </p>
    <p class="p3">
      강아지 사진과 네이버 링크 p3 그룹 1
    </p>
    <p class="p3">
      강아지 사진과 네이버 링크 p3 그룹 2
    </p>
    <p id="p4_only">
      강아지 사진과 네이버 ID p4
    </p>
  <pre>
내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
</pre>
  </div>
  <div>
    <p class="p3">
      강아지 사진과 네이버 링크 p3 그룹 3
    </p>
    <p class="p3">
      강아지 사진과 네이버 링크 p3 그룹 4
    </p>
    <p id="p5_only">
      강아지 사진과 네이버 ID p4
    </p>
  </div>
</body>
</html>
```

children 를 활용

- 자신의 요소의 자식들의 요소를 가지고 올 수 있다.
- content속성(or contents)을 이용하여 가져올수도 있음.

- 기타 `parents`, `next_sibling`, `next_elements` 등이 있음

In [17]:

```
soup.children
```

Out[17]:

```
<list_iterator at 0x226a8144710>
```

In [18]:

```
soup_children_list = list(soup.children)
soup_children_list
```

Out[18]:

```
[<html>
  <head><title>나의 홈페이지</title>
</head><body>
  안녕하세요
  <div>
    <a href="https://www.naver.com/">naver</a>
    <a href="https://www.google.com/">google</a>
    <p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
    <p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
    <p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
    <p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
    <p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
  <pre>
    내가 가장 좋아하는 동물은 강아지입니다.
    나는 그리고 네이버 홈페이지에 자주 갑니다.
  </pre>
</div>
<div>
  <p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>
  <p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>
  <p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
</div>
</body>
</html>]
```

In [19]:

```
soup_children_list[0].children
```

Out[19]:

```
<list_iterator at 0x226a81c0940>
```


In [20]:



```
tmp = list(soup_children_list[0].children)
tmp
```

Out[20]:

```
['\n', <head><title>나의 홈페이지</title>
</head>, <body>
안녕하세요
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com/">google</a>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
<pre>
내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
</pre>
</div>
<div>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>
<p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
</div>
</body>, '\n']
```

In [21]:



```
### body 부분 정보 얻기
Content_Body = soup.body
Content_Body
```

Out[21]:

```
<body>
안녕하세요
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com/">google</a>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
<pre>
내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
</pre>
</div>
<div>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>
<p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
</div>
</body>
```

In [22]:



```
Content_Body = list(soup_children_list[0].children)[2]
Content_Body
```

Out[22]:

```
<body>
안녕하세요
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com/">google</a>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
<pre>
내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
</pre>
</div>
<div>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>
<p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
</div>
</body>
```

find을 이용한 하나의 정보 얻기

In [23]:



```
soup
```

Out[23]:

```
<html>
<head><title>나의 홈페이지</title>
</head><body>
안녕하세요
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com/">google</a>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
<pre>
내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
</pre>
</div>
<div>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>
<p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
</div>
</body>
</html>
```

In [24]:



```
soup.find('p')
```

Out[24]:

```
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
```

In [25]:



```
soup.find('title')
```

Out[25]:

```
<title>나의 홈페이지</title>
```

In [26]:



```
soup.find('div')
```

Out[26]:

```
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com/">google</a>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
<pre>
내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
</pre>
</div>
```

1-8 find_all를 이용하기

- find가 하나의 정보를 가져오는 것이라면 find_all은 확인되는 전체 정보를 가지고 온다.
- find_all 은 반환되는 값의 형태는 리스트가 된다.
- limit 키워드를 사용하여 태그 수의 제한두기(..., limit=1)

In [27]:



```
soup
```

Out[27]:

```
<html>
<head><title>나의 홈페이지</title>
</head><body>
안녕하세요
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com/">google</a>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
<pre>
내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
</pre>
</div>
<div>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>
<p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
</div>
</body>
</html>
```

In [28]:



```
soup.find_all('p')
```

Out[28]:

```
[<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>,
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>,
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>,
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>,
<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>,
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>,
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>,
<p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>]
```

In [29]:



```
soup.find_all('p', class_ = 'p3')
```

Out[29]:

```
[<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>,
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>,
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>,
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>]
```

In [30]:



```
soup.find_all('p', id = 'p4_only')
```

Out[30]:

```
[<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>]
```

1-9 soup.find_all로 확인된 정보 하나 하나의 값에 접근하기

- 기본적으로 리스트 형태이기에 for문을 이용하여 접근이 가능하다.

In [31]:



```
for ptag in soup.find_all('p'):
    print(ptag)
```

```
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 1</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 2</p>
<p id="p4_only"> 강아지 사진과 네이버 ID p4 </p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 3</p>
<p class="p3"> 강아지 사진과 네이버 링크 p3 그룹 4</p>
<p id="p5_only"> 강아지 사진과 네이버 ID p4 </p>
```

In [32]:



```
for ptag in soup.find_all('p'):
    print(ptag.text)
```

내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
강아지 사진과 네이버 링크 p3 그룹 1
강아지 사진과 네이버 링크 p3 그룹 2
강아지 사진과 네이버 ID p4
강아지 사진과 네이버 링크 p3 그룹 3
강아지 사진과 네이버 링크 p3 그룹 4
강아지 사진과 네이버 ID p4

In [33]:



```
for ptag in soup.find_all('p'):
    print(ptag.get_text())
```

내가 가장 좋아하는 동물은 강아지입니다.
나는 그리고 네이버 홈페이지에 자주 갑니다.
강아지 사진과 네이버 링크 p3 그룹 1
강아지 사진과 네이버 링크 p3 그룹 2
강아지 사진과 네이버 ID p4
강아지 사진과 네이버 링크 p3 그룹 3
강아지 사진과 네이버 링크 p3 그룹 4
강아지 사진과 네이버 ID p4

1-10 링크를 가져오기

In [34]:



```
soup.find_all('a')
```

Out[34]:

```
[<a href="https://www.naver.com/">naver</a>,
 <a href="https://www.google.com/">google</a>]
```

In [35]:



```
links = soup.find_all('a')
print(links[1]['href'])
print(links[1].string)
```

```
https://www.google.com/ (https://www.google.com/)
google
```

In [36]:



```
for each in links:
    href = each['href']
    text = each.string
    print(text + ' -> ' + href)
```

naver -> <https://www.naver.com/> (<https://www.naver.com/>)
google -> <https://www.google.com/> (<https://www.google.com/>)

기타 알아보기

- select()를 활용한 검색 결과 확인해 보기
- extract()를 활용한 태그를 지우기
- bs4와 re의 조합으로 좀 더 효율적으로 정보를 찾아보기

Ref

In [37]:



```
import os
print(os.getcwd()) #현재 이 주피터노트북의 주소
```

C:\Users\Whp\Documents\GitHub\PythonBasic\PythonWeb

In [38]:



```
print(os.path.realpath("0720.html")) # file or symbolic link인 경우 원본 위치를 찾아줌
```

C:\Users\Whp\Documents\GitHub\PythonBasic\PythonWeb\0720.html

In []:

