2. 웹 정보 수집 - BeautifulSoup 이해

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']
```

1-6 만약 정보를 가져올 때, 에러가 발생하면 어떻게 될까?

In [45]:

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

print(tag_p.attrs)
```

```
{}
```

In [46]:

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

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-46-888641df517d> in <module>
----> 1 print(tag_p['style'])   # style가 없어 에러 발생

~₩anaconda3₩lib₩site-packages₩bs4₩element.py in __getitem__(self, key)
   1399         """tag[key] returns the value of the 'key' attribute for the Tag,
   1400         and throws an exception if it's not there."""
-> 1401         return self.attrs[key]
   1402
   1403     def __iter__(self):

KeyError: 'style'
```

In [47]:

```
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 [48]:

```
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 [49]:

```
from bs4 import BeautifulSoup

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

Out[49]:

```
'<!-- 나의 웹 페이지 프로그램 -->₩n<html>₩n<head>₩n  <title>내 페이지</title>₩n  <
style>₩n  h1 { background-color: powderblue; }₩n  #p1 { ₩nWtfont-size:50px;₩nWtbac
kground-color:rgb(255, 207, 71);₩n  }₩n  .ptag { font-size:35px; }₩n  </style>
₩n</head>₩n<body>₩nWt<h1>헤드라인1</h1>₩nWt<div>₩nWt<p id="p1" class="ptag">단락1</p
>₩nWt<p class="ptag">단락2</p>₩nWt<p class="ptag">단락3₩nWt단락4₩nWt단락5₩nWt</p>₩n
Wt<pre>₩nWt단락3₩nWt단락4₩nWt단락5₩nWt</pre>₩nWt<p  class="ptag" style="font-size:40
px; background-color:green;">단락4</p>₩nWt<p  class="ptag" title="IW'm a tooltip">단
락5</p>₩nWt</div>₩nWt<div>₩nWt단락6<br>₩nWt단락7<br>₩nWt</div>₩nWt<div>₩nWt<a href
="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a><br>₩nWt<a href="https://ww
w.youtube.com/?gl=KR&hl=ko">유튜브</a><br>₩nWt<a href="https://www.naver.com/">네이
버</a><br>₩nWt<a href="https://www.daum.net/">다음</a><br>₩nWt<a href="https://papag
o.naver.com/">파파고</a><br>₩nWt</div>₩nWt<div>₩nWt<img src="C:/Users/user/Document
s/GitHub/myrepo2/pug-690566__340.webp">₩nWt<img src="2018-11-27-14-45-17.jpg" width=
300 height=300>₩nWt</div>₩n</body>₩n</html>'
```

In [50]:

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

Out[50]:

```
<!-- 나의 웹 페이지 프로그램 --><html>
<head>
<title>내 페이지</title>
<style>
   h1 { background-color: powderblue; }
   #p1 {
        font-size:50px;
        background-color:rgb(255, 207, 71);
   }
   .ptag { font-size:35px; }
   </style>
</head>
<body>
<h1>헤드라인1</h1>
<div>
<p class="ptag" id="p1">단락1</p>
<p class="ptag">단락2</p>
<p class="ptag">단락3
        단락4
        단락5
        </p>
<pre>
        단락3
        단락4
        단락5
        </pre>
<p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
<p class="ptag" title="I'm a tooltip">단락5</p>
</div>
<div>
        단락6<br/>
        단락7<br/>
</div>
<div>
<a href="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a><br/>
<a href="https://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a><br/>
<a href="https://www.naver.com/">네이버</a><br/>
<a href="https://www.daum.net/">다음</a><br/>
<a href="https://papago.naver.com/">파파고</a><br/>
</div>
<div>
<img src="C:/Users/user/Documents/GitHub/myrepo2/pug-690566__340.webp"/>
<img height="300" src="2018-11-27-14-45-17.jpg" width="300"/>
</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 [51]:

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

```
<!-- 나의 웹 페이지 프로그램 -->
<html>
 <head>
  <title>
   내 페이지
  </title>
  <style>
   h1 { background-color: powderblue; }
   #p1 {
        font-size:50px;
        background-color:rgb(255, 207, 71);
   }
   .ptag { font-size:35px; }
  </style>
 </head>
 <body>
  <h1>
   헤드라인1
  </h1>
  <div>
   <p class="ptag" id="p1">
    단락1
   </p>
   <p class="ptag">
    단락2
   </p>
   <p class="ptag">
    단락3
        단락4
        단락5
   </p>
   <pre>
        단락3
        단락4
        단락5
        </pre>
   <p class="ptag" style="font-size:40px; background-color:green;">
    단락4
   </p>
   <p class="ptag" title="I'm a tooltip">
    단락5
   </p>
  </div>
  <div>
   단락6
   <br/>
   단락7
   <br/>
  </div>
  <div>
   <a href="https://scholar.google.co.kr/schhp?hl=ko">
    구글논문
   </a>
   <br/>
   <a href="https://www.youtube.com/?gl=KR&amp;hl=ko">
    유튜브
```

```
    </a>
    <br/>
    <a href="https://www.naver.com/">
     네이버
    </a>
    <br/>
    <a href="https://www.daum.net/">
     다음
    </a>
    <br/>
    <a href="https://papago.naver.com/">
     파파고
    </a>
    <br/>
   </div>
   <div>
    <img src="C:/Users/user/Documents/GitHub/myrepo2/pug-690566__340.webp"/>
    <img height="300" src="2018-11-27-14-45-17.jpg" width="300"/>
   </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 [52]:

```
soup.children
```

Out[52]:

```
<list_iterator at 0x25be9e050a0>
```

In [53]:

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

Out[53]:

```
[' 나의 웹 페이지 프로그램 ',
 <html>
 <head>
 <title>내 페이지</title>
 <style>
    h1 { background-color: powderblue; }
    #p1 {
        font-size:50px;
        background-color:rgb(255, 207, 71);
    }
    .ptag { font-size:35px; }
    </style>
 </head>
 <body>
 <h1>헤드라인1</h1>
 <div>
 <p class="ptag" id="p1">단락1</p>
 <p class="ptag">단락2</p>
 <p class="ptag">단락3
        단락4
        단락5
        </p>
 <pre>
        단락3
        단락4
        단락5
        </pre>
 <p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
 <p class="ptag" title="I'm a tooltip">단락5</p>
 </div>
 <div>
        단락6<br/>
        단락7<br/>
 </div>
 <div>
 <a href="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a><br/>
 <a href="https://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a><br/>
 <a href="https://www.naver.com/">네이버</a><br/>
 <a href="https://www.daum.net/">다음</a><br/>
 <a href="https://papago.naver.com/">파파고</a><br/>
 </div>
 <div>
 <img src="C:/Users/user/Documents/GitHub/myrepo2/pug-690566__340.webp"/>
 <img height="300" src="2018-11-27-14-45-17.jpg" width="300"/>
 </div>
 </body>
 </html>]
```

html 태그의 자식들

In [54]:

```
soup_children_list[1].children
```

Out[54]:

```
<list_iterator at 0x25be9e05280>
```

In [55]:

```
tmp = list(soup_children_list[1].children)
tmp
```

Out[55]:

```
['₩n',
 <head>
 <title>내 페이지</title>
 <style>
    h1 { background-color: powderblue; }
    #p1 {
        font-size:50px;
        background-color:rgb(255, 207, 71);
    }
    .ptag { font-size:35px; }
    </style>
 </head>,
 '₩n',
 <body>
 <h1>헤드라인1</h1>
 <div>
 <p class="ptag" id="p1">단락1</p>
 <p class="ptag">단락2</p>
 <p class="ptag">단락3
        단락4
        단락5
        </p>
 <pre>
        단락3
        단락4
        단락5
        </pre>
 <p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
 <p class="ptag" title="I'm a tooltip">단락5</p>
 </div>
 <div>
        단락6<br/>
        단락7<br/>
 </div>
 <div>
 <a href="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a><br/>
 <a href="https://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a><br/>
 <a href="https://www.naver.com/">네이버</a><br/>
 <a href="https://www.daum.net/">다음</a><br/>
 <a href="https://papago.naver.com/">파파고</a><br/>
 </div>
 <div>
 <img src="C:/Users/user/Documents/GitHub/myrepo2/pug-690566__340.webp"/>
 <img height="300" src="2018-11-27-14-45-17.jpg" width="300"/>
 </div>
 </body>,
 '₩n']
```

In [56]:

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

Out[56]:

```
<body>
<h1>헤드라인1</h1>
<div>
<p class="ptag" id="p1">단락1</p>
<p class="ptag">단락2</p>
<p class="ptag">단락3
        단락4
        단락5
        </p>
<pre>
        단락3
        단락4
        단락5
        </pre>
<p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
<p class="ptag" title="I'm a tooltip">단락5</p>
</div>
<div>
        단락6<br/>
        단락7<br/>
</div>
<div>
<a href="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a><br/>
<a href="https://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a><br/>
<a href="https://www.naver.com/">네이버</a><br/>
<a href="https://www.daum.net/">다음</a><br/>
<a href="https://papago.naver.com/">파파고</a><br/>
</div>
<div>
<img src="C:/Users/user/Documents/GitHub/myrepo2/pug-690566__340.webp"/>
<img height="300" src="2018-11-27-14-45-17.jpg" width="300"/>
</div>
</body>
```

In [57]:

```
Content_Body = list(soup_children_list[1].children)[3]
Content_Body
```

Out[57]:

```
<body>
<h1>헤드라인1</h1>
<div>
<p class="ptag" id="p1">단락1</p>
<p class="ptag">단락2</p>
<p class="ptag">단락3
        단락4
        단락5
        </p>
<pre>
        단락3
        단락4
        단락5
        </pre>
<p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
<p class="ptag" title="I'm a tooltip">단락5</p>
</div>
<div>
        단락6<br/>
        단락7<br/>
</div>
<div>
<a href="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a><br/>
<a href="https://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a><br/>
<a href="https://www.naver.com/">네이버</a><br/>
<a href="https://www.daum.net/">다음</a><br/>
<a href="https://papago.naver.com/">파파고</a><br/>
</div>
<div>
<img src="C:/Users/user/Documents/GitHub/myrepo2/pug-690566__340.webp"/>
<img height="300" src="2018-11-27-14-45-17.jpg" width="300"/>
</div>
</body>
```

find을 이용한 하나의 정보 얻기

In [58]:

```
soup
```

Out[58]:

```
<!-- 나의 웹 페이지 프로그램 --><html>
<head>
<title>내 페이지</title>
<style>
   h1 { background-color: powderblue; }
   #p1 {
        font-size:50px;
        background-color:rgb(255, 207, 71);
   }
   .ptag { font-size:35px; }
   </style>
</head>
<body>
<h1>헤드라인1</h1>
<div>
<p class="ptag" id="p1">단락1</p>
<p class="ptag">단락2</p>
<p class="ptag">단락3
        단락4
        단락5
        </p>
<pre>
        단락3
        단락4
        단락5
        </pre>
<p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
<p class="ptag" title="I'm a tooltip">단락5</p>
</div>
<div>
        단락6<br/>
        단락7<br/>
</div>
<div>
<a href="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a><br/>
<a href="https://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a><br/>
<a href="https://www.naver.com/">네이버</a><br/>
<a href="https://www.daum.net/">다음</a><br/>
<a href="https://papago.naver.com/">파파고</a><br/>
</div>
<div>
<img src="C:/Users/user/Documents/GitHub/myrepo2/pug-690566__340.webp"/>
<img height="300" src="2018-11-27-14-45-17.jpg" width="300"/>
</div>
</body>
</html>
```

In [59]:

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

Out[59]:

```
<p class="ptag" id="p1">단락1</p>
```

In [60]:

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

Out[60]:

```
<title>내 페이지</title>
```

In [61]:

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

Out[61]:

```
<div>
<p class="ptag" id="p1">단락1</p>
<p class="ptag">단락2</p>
<p class="ptag">단락3
        단락4
        단락5
        </p>
<pre>
        단락3
        단락4
        단락5
        </pre>
<p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
<p class="ptag" title="I'm a tooltip">단락5</p>
</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 [62]:

```
soup
```

Out[62]:

```
<!-- 나의 웹 페이지 프로그램 --><html>
<head>
<title>내 페이지</title>
<style>
   h1 { background-color: powderblue; }
   #p1 {
        font-size:50px;
        background-color:rgb(255, 207, 71);
   }
   .ptag { font-size:35px; }
   </style>
</head>
<body>
<h1>헤드라인1</h1>
<div>
<p class="ptag" id="p1">단락1</p>
<p class="ptag">단락2</p>
<p class="ptag">단락3
        단락4
        단락5
        </p>
<pre>
        단락3
        단락4
        단락5
        </pre>
<p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
<p class="ptag" title="I'm a tooltip">단락5</p>
</div>
<div>
        단락6<br/>
        단락7<br/>
</div>
<div>
<a href="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a><br/>
<a href="https://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a><br/>
<a href="https://www.naver.com/">네이버</a><br/>
<a href="https://www.daum.net/">다음</a><br/>
<a href="https://papago.naver.com/">파파고</a><br/>
</div>
<div>
<img src="C:/Users/user/Documents/GitHub/myrepo2/pug-690566__340.webp"/>
<img height="300" src="2018-11-27-14-45-17.jpg" width="300"/>
</div>
</body>
</html>
```

In [63]:

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

Out[63]:

```
[<p class="ptag" id="p1">단락1</p>,
 <p class="ptag">단락2</p>,
 <p class="ptag">단락3
        단락4
        단락5
        </p>,
 <p class="ptag" style="font-size:40px; background-color:green;">단락4</p>,
 <p class="ptag" title="I'm a tooltip">단락5</p>]
```

In [64]:

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

Out[64]:

```
[<p class="ptag" id="p1">단락1</p>,
 <p class="ptag">단락2</p>,
 <p class="ptag">단락3
        단락4
        단락5
        </p>,
 <p class="ptag" style="font-size:40px; background-color:green;">단락4</p>,
 <p class="ptag" title="I'm a tooltip">단락5</p>]
```

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

- 기본적으로 리스트 형태이기에 for문을 이용하여 접근이 가능하다.

In [65]:

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

```
<p class="ptag" id="p1">단락1</p>
<p class="ptag">단락2</p>
<p class="ptag">단락3
        단락4
        단락5
        </p>
<p class="ptag" style="font-size:40px; background-color:green;">단락4</p>
<p class="ptag" title="I'm a tooltip">단락5</p>
```

In [66]:

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

```
단락1
단락2
단락3
        단락4
        단락5

단락4
단락5
```

In [67]:

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

```
단락1
단락2
단락3
        단락4
        단락5

단락4
단락5
```

1-10 링크를 가져오기

In [68]:

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

Out[68]:

```
[<a href="https://scholar.google.co.kr/schhp?hl=ko">구글논문</a>,
 <a href="https://www.youtube.com/?gl=KR&amp;hl=ko">유튜브</a>,
 <a href="https://www.naver.com/">네이버</a>,
 <a href="https://www.daum.net/">다음</a>,
 <a href="https://papago.naver.com/">파파고</a>]
```

In [69]:

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

```
https://www.youtube.com/?gl=KR&hl=ko (https://www.youtube.com/?gl=KR&hl=ko)
유튜브
```

In [70]:

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

```
구글논문 -> https://scholar.google.co.kr/schhp?hl=ko (https://scholar.google.co.kr/s
chhp?hl=ko)
유튜브 -> https://www.youtube.com/?gl=KR&hl=ko (https://www.youtube.com/?gl=KR&hl=k
o)
네이버 -> https://www.naver.com/ (https://www.naver.com/)
다음 -> https://www.daum.net/ (https://www.daum.net/)
파파고 -> https://papago.naver.com/ (https://papago.naver.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 [71]:

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

```
C:₩Users₩hp₩Documents₩GitHub₩AI_enovation₩part02_library₩20201013_class
```