웹 데이터 수집 기본 다지기

학습 목표

- 웹 데이터 수집을 위한 기본 내용을 이해해 본다.

urlopen함수와 BeautifulSoup함수의 이해

학습내용

- urlopen() 함수 기본
- BeautifulSoup() 함수 기본
- lxml등을 이용한 html 정보를 구조화 시키기
- 정보 가져오는 여러가지 방법

학습 내용

1. urlopen() 함수 기본

목차로 이동하기

- urllib.request - url 처리와 관련된 모듈 패키지.
- [패키지].[모듈].[함수()]의 형식으로 실행.
- urlopen() 함수를 이용하여 URL로 HTTP 요청을 보내고 서버의 응답을 받는다.
- 국내증시 html 정보 가져오기
- url : https://finance.naver.com/sise/

In [2]:
```
from urllib.request import urlopen
```

In [3]:
```
## 정보 가져오기
url = "https://finance.naver.com/sise/"
page = urlopen(url)
page
```

Out[3]:
```
<http.client.HTTPResponse at 0x25fddd77550>
```

list() 함수를 활용한 획득 정보 일부를 확인

In [4]:
```
listpage = list(page)
listpage[0:20]
```

Out[4]:
```
[b"<script language='javascript'>\n",
 b'\n',
 b'function main_tab(tab_title, pst, tab_cnt)\n',
 b'{\n',
 b'\tfor(var i=0 ; i<tab_cnt ; i++)\n',
 b'\t{\n',
 b'\t\tif (i == pst)\n',
 b"\t\t\tdocument.getElementById(tab_title+'_title_tab_'+i).style.display =
'';\n",
 b'\t\telse\n',
 b"\t\t\tdocument.getElementById(tab_title+'_title_tab_'+i).style.display = 'no
ne';\n",
 b'\n',
 b'\n',
 b'\t\tif (i == pst)\n',
 b"\t\t\tdocument.getElementById(tab_title+'_tab_'+i).style.display = '';\n",
 b'\t\telse\n',
 b"\t\t\tdocument.getElementById(tab_title+'_tab_'+i).style.display = 'non
e';\n",
 b'\n',
 b'\t}\n',
 b'}\n',
 b'\n']
```

2. BeautifulSoup: HTML 및 XML 데이터 파싱을 위한 파이썬 라이브러리

목차로 이동하기

- 파싱(parsing)은 컴퓨터 과학에서 텍스트 데이터나 코드의 구조를 분석하고 이해하는 과정
- BeautifulSoup 는 파이썬 라이브러리입니다.
- HTML과 XML파일로부터 데이터를 쉽게 가져오기 위한 파이썬 라이브러리이다.
- Beautiful Soup을 사용하면 복잡한 HTML 구조를 간단하게 탐색하고 수정 가능합니다.
- bs4의 모듈 안에 있음.

In [7]:
```
from bs4 import BeautifulSoup
```

In [8]:
```
page = '''
<html>
<title>나의 홈페이지</title>
<body>
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com">google</a>
<img height="300" src="dog_01.jpg" width="500"/>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 </p>
```

```
<p id="p4"> 간단한 나의 홈페이지를 만들다.</p>
<p class="p3"> 강아지 사진과 네이버 링크222 </p>
</div>
</body>
</html>
'''
```

2.1 html을 파일로부터 읽을 때의 경우

목차로 이동하기

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

HTML/XHTML 텍스트 파일을 구문 분석하기 위해 여러 가지 파서를 사용할 수 있습니다. 파싱은 HTML 문법 규칙을 따른 문자열을 다른 문법 기준으로 단어의 의미나 구조를 분석하는 과정을 말합니다. 이를 수행하는 프로그램을 HTML 파서(HTML Parser)라고 합니다.

- HTML/XHTML을 텍스트 파일을 구문분석하기 몇가지 파서를 사용합니다.
  - HTML Parse란 : HTML 문법 규칙을 따른 문자열을 다른 문법을 기준으로 단어의 의미나 구조를 분석하는 것을 말함. 이를 수행하는 프로그램을 HTML Parser라 한다.

- HTML 파서의 종류
  - lxml : c로 구현된 가장 빠름.
  - html5lib : 웹 브라우저 방식으로 HTML 해석
  - html.parser :파이썬 표준 라이브러리로 제공되는 기본 HTML 파서

- 특정 parser의 경우, 특정 태그가 포함되지 않는 오류가 있을 수 있음.

In [9]:
```
soup = BeautifulSoup(page, 'lxml')
soup
```

Out[9]:
```
<html>
<head><title>나의 홈페이지</title>
</head><body>
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com">google</a>
<img height="300" src="dog_01.jpg" width="500"/>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 </p>
<p id="p4"> 간단한 나의 홈페이지를 만들다.</p>
<p class="p3"> 강아지 사진과 네이버 링크222 </p>
</div>
</body>
</html>
```

In [10]:
```
## soup 정보에서 head 부분 가져오기
soup.head
```

Out[10]:
```
<head><title>나의 홈페이지</title>
</head>
```

In [11]:
```
## soup 정보에서 body 부분 가져오기
soup.body
```

Out[11]:
```
<body>
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com">google</a>
<img height="300" src="dog_01.jpg" width="500"/>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 </p>
<p id="p4"> 간단한 나의 홈페이지를 만들다.</p>
<p class="p3"> 강아지 사진과 네이버 링크222 </p>
</div>
</body>
```

In [12]:
```
## soup 정보에서 body 부분안의 p태그 부분 가져오기
soup.body.p
```

Out[12]:
```
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
```

In [13]:
```
## soup 정보에서 body 부분안의 p태그 부분 텍스트 정보 가져오기
soup.body.p.text
```

Out[13]:
```
' 내가 가장 좋아하는 동물은 강아지입니다.'
```

In [14]:
```
print(soup.prettify())
```

```
<html>
 <head>
  <title>
   나의 홈페이지
  </title>
 </head>
 <body>
  <div>
   <a href="https://www.naver.com/">
    naver
   </a>
   <a href="https://www.google.com">
    google
   </a>
   <img height="300" src="dog_01.jpg" width="500"/>
   <p>
    내가 가장 좋아하는 동물은 강아지입니다.
   </p>
   <p>
    나는 그리고 네이버 홈페이지에 자주 갑니다.
   </p>
   <p class="p3">
    강아지 사진과 네이버 링크
   </p>
   <p id="p4">
    간단한 나의 홈페이지를 만들다.
   </p>
   <p class="p3">
    강아지 사진과 네이버 링크222
   </p>
  </div>
 </body>
</html>
```

2.2 자식 요소들의 정보를 반환 - children

목차로 이동하기

- children에 대해 알아보기
- 지정된 태그보다 한 레벨 아래 위치한 태그

In [15]:
```
soup.body.children
```

Out[15]:
```
<list_iterator at 0x25fdddc7010>
```

body 태그의 모든 자식 요소를 리스트 형태로 변환하여 반환

In [16]:
```
list(soup.body.children)
```

Out[16]:
```
['\n',
 <div>
 <a href="https://www.naver.com/">naver</a>
 <a href="https://www.google.com">google</a>
 <img height="300" src="dog_01.jpg" width="500"/>
 <p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
 <p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
 <p class="p3"> 강아지 사진과 네이버 링크 </p>
 <p id="p4"> 간단한 나의 홈페이지를 만들다.</p>
 <p class="p3"> 강아지 사진과 네이버 링크222 </p>
 </div>,
 '\n']
```

In [17]:
```
tmp = list(soup.body.children)[1]
tmp
```

Out[17]:
```
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com">google</a>
<img height="300" src="dog_01.jpg" width="500"/>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 </p>
<p id="p4"> 간단한 나의 홈페이지를 만들다.</p>
<p class="p3"> 강아지 사진과 네이버 링크222 </p>
</div>
```

In [18]:
```
list(tmp.children)[3]
```

Out[18]:
```
<a href="https://www.google.com">google</a>
```

2.3 find_all(), find()을 이용한 정보 얻기

목차로 이동하기

- find_all() : 조건이 맞는 전체 정보를 리스트로 가져오기
- find() : 조건이 맞는 정보 하나만 가져오기

In [19]:
```
soup.find_all('p')
```

Out[19]:
```
[<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>,
 <p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>,
 <p class="p3"> 강아지 사진과 네이버 링크 </p>,
 <p id="p4"> 간단한 나의 홈페이지를 만들다.</p>,
 <p class="p3"> 강아지 사진과 네이버 링크222 </p>]
```

In [20]:
```
soup.find('p')
```

Out[20]:
```
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
```

In [21]:
```
soup.find('div')
```

Out[21]:
```
<div>
<a href="https://www.naver.com/">naver</a>
<a href="https://www.google.com">google</a>
<img height="300" src="dog_01.jpg" width="500"/>
<p> 내가 가장 좋아하는 동물은 강아지입니다.</p>
<p> 나는 그리고 네이버 홈페이지에 자주 갑니다.</p>
<p class="p3"> 강아지 사진과 네이버 링크 </p>
<p id="p4"> 간단한 나의 홈페이지를 만들다.</p>
<p class="p3"> 강아지 사진과 네이버 링크222 </p>
</div>
```

- HTML 속성 정보인 class_, id로 활용하여 정보를 가져올 수 있음.

In [22]:
```
soup.find_all('p', class_='p3')
```

Out[22]:
```
[<p class="p3"> 강아지 사진과 네이버 링크 </p>, <p class="p3"> 강아지 사진과 네이버 링크222 </p>]
```

In [23]:
```
soup.find_all(id='p4')
```

Out[23]:
```
[<p id="p4"> 간단한 나의 홈페이지를 만들다.</p>]
```

2.4 p 태그의 문자들만 가져오기

목차로 이동하기

In [24]:
```
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 </p>
<p id="p4"> 간단한 나의 홈페이지를 만들다.</p>
<p class="p3"> 강아지 사진과 네이버 링크222 </p>
```

In [25]:
```
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
간단한 나의 홈페이지를 만들다.
강아지 사진과 네이버 링크222
```

2.5 링크 가져오기(link)

목차로 이동하기

In [26]:
```
soup.find_all('a')
```

Out[26]:
```
[<a href="https://www.naver.com/">naver</a>,
 <a href="https://www.google.com">google</a>]
```

In [27]:
```
links = soup.find_all('a')
print(links[1]['href'])
```

```
print(links[1].string)
```

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

In [28]:
```
for each in links:
    href = each['href']
    text = each.string
    print(text + ' -> ' + href)
```

```
naver -> https://www.naver.com/
google -> https://www.google.com
```