LESSON

BeautifulSoup 모듈

BeautifulSoup 모듈 정의

- 홈페이지 내 데이터를 쉽게 추출할 수 있도록 도와주는 파이썬 외부 라이브러리
- 웹 문서 내 수많은 HTML 태그들을 파서(parser)를 활용해 사용하기 편한 파이썬 객체로 만들어 제공

  -html 외에 xml 파서(parser) 도 제공

- 웹 문서 구조를 알고 있다면, 아주 편하게 원하는 데이터를 뽑아 활용할 수 있음

무작위로 작성된 HTML 코드를 태그들을 찾아 구조화 시킨 예제

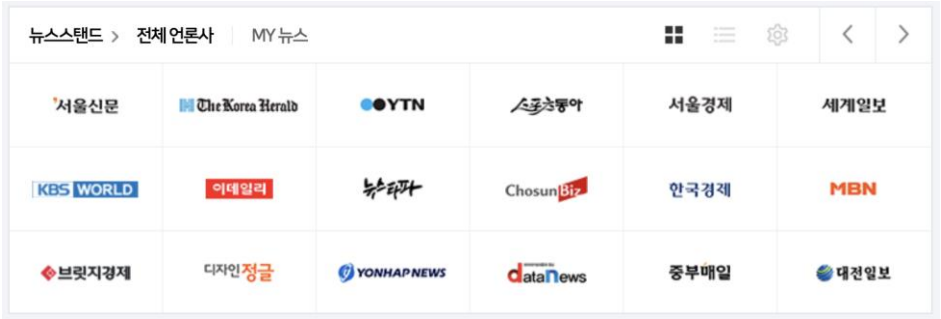
```
html_doc = """
<html><head><title>The Dormouse's story</title></head><body><p class="title">
<b>The Dormouse's story</b></p>
<p class="story">Once upon a time there were three little sisters;and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<p class="story">...</p>
"""
```

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')

print(soup.prettify())
```

```
<html>
 <head>
  <title>
   The Dormouse's story
  </title>
 </head>
 <body>
  <p class="title">
   <b>
    The Dormouse's story
   </b>
  </p>
  <p class="story">
   Once upon a time there were three little sisters;and their names were
   <a class="sister" href="http://example.com/elsie" id="link1">
    Elsie
   </a>
   ,
   <p class="story">
    ...
   </p>
  </p>
 </body>
</html>
```

기존 학습했던 방식과 차이점을 보여주는 예제 - Naver 메인 페이지의 언론사 목록 가져오기




```
<div class="an_panel_image _PM_newsstand_thumb" role="tabpanel" >
    <div class="api_list_wrap">
        <h3><span class="blind">언론사 목록</span></h3>
        <div class="flick-view">
            <div class="flick-container">
                <div class="flick-panel">
                    <ul class="api_list">


_038" class="api_item" data-pid="038">
api_link" href="http://newsstand.naver.com/?list=ct1&pcode=038" aria-haspopup="true" target="_blank">
https://s.pstatic.net/static/newsstand/up/2017/0424/nsd172837200.png" height="24" alt="한국일보" class=

="api_popup_btn_set" role="alertdialog">
```

BeautifulSoup 모듈 정의 : 기존 방식과의 차이점

- 기존 방식 : 정규 표현식, 문자열 함수 등을 활용하여 홈페이지 텍스트 내 패턴을 분석하여 하나씩 원하는 데이터를 찾아가는 형식

```
import requests
from bs4 import BeautifulSoup

req = requests.get("https://naver.com")
html =  req.text

html = html.split('<h3><span class="blind">언론사 목록</span></h3>')[1]
html = html.split('<i class="api_list_border_right" role="presentation"')[0]
html = html.split('alt="')
news_list = []
for i in html:
    news_list.append(i.split('"')[0])
print(news_list[1:])
```

```
['스포츠동아', '동아일보', '아이뉴스24', '일간스포츠', '노컷뉴스', '이데일리', '서울신문', '프레시안', '아시아경제', 'MBN', '시사인', 'YTN',
'베리타스알파', 'enews24', '산업일보', 'CNB뉴스', '충청투데이', '중부매일신문']
```

BeautifulSoup 모듈 정의 : 기존 방식과의 차이점

- BeautifulSoup 방식 : HTML 문서를 태그를 기반으로 구조화하여 태그로 원하는 데이터를 찾아가는 형식

```
import requests
from bs4 import BeautifulSoup

req = requests.get("https://naver.com")
html =  req.text
soup = BeautifulSoup(html, 'html.parser')

result = soup.find_all('a','api_link')
news_list = []
for i in result:
    news_list.append(i.find("img")["alt"])
print(news_list)
```

```
['스포츠동아', '동아일보', '아이뉴스24', '일간스포츠', '노컷뉴스', '이데일리', '서울신문', '프레시안', '아시아경제', 'MBN', '시사인', 'YTN',
'베리타스알파', 'enews24', '산업일보', 'CNB뉴스', '충청투데이', '중부매일신문']
```

BeautifulSoup 모듈 설치

- 외부 라이브러리로 pip을 활용해 설치 해주어야만 사용 가능

방법① 콘솔창에서 pip install beautifulsoup4 명령어로 설치

```
[zoostar@~$pip install beautifulsoup4
Requirement already satisfied: beautifulsoup4
```

BeautifulSoup 모듈 설치

- 외부 라이브러리로 pip을 활용해 설치 해주어야만 사용 가능

방법② Jupyter Notebook에서 !pip install beautifulsoup4 명령어로 설치

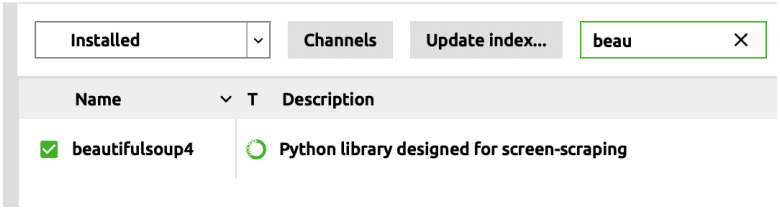
```
!pip install beautifulsoup4
```

```
Requirement already satisfied:
```

BeautifulSoup 모듈 설치

- 외부 라이브러리로 pip을 활용해 설치 해주어야만 사용 가능

방법③ Anaconda Navigator에서 검색 후 설치



BeautifulSoup 모듈 사용법

- from bs4 import BeautifulSoup로 모듈을 호출하여 사용

```
from bs4 import BeautifulSoup
soup = BeautifulSoup(html_doc, 'html.parser')
print(soup.prettify())
```

BeautifulSoup 모듈 사용법

- 모듈 내 BeautifulSoup( ) 클래스에 HTML 문서와 파서(parser)를 전달하여 분석 결과를 객체에 저장
  - 파서(parser)에 따라 HTML을 분석할 때, 태그를 추가, 무시, 강제 변경 등의 작업 수행

```
BeautifulSoup("<a></p>", "html.parser")
```

```
<a></a>
```

```
BeautifulSoup("<a></p>", "html5lib")
```

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

```
BeautifulSoup("<a></p>", "lxml")
```

```
<html><body><a></a></body></html>
```

```
BeautifulSoup("<a><b /></a>", "xml")
```

```
<?xml version="1.0" encoding="utf-8"?>
<a><b/></a>
```

BeautifulSoup 모듈 사용법

- BeautifulSoup은 HTML을 파싱하여 구조화하는 모듈로 urllib, requests 모듈 등과 함께 사용
  - 예) requests 모듈로 웹 문서를 텍스트로 가져온 뒤 BeautifulSoup 모듈로 분석

```
import requests

req = requests.get("https://naver.com")
html =  req.text
```

```
import urllib

response = urllib.request.urlopen("https://naver.com")
byte_data = response.read()
html = byte_data.decode()
```

```
from bs4 import BeautifulSoup

soup = BeautifulSoup(html, 'html.parser')
result = soup.find_all('a','api_link')
```

BeautifulSoup 모듈 사용법 : 검색

- 태그 : HTML의 해당 태그에 대한 첫 번째 정보를 가져옴
  - 태그['속성'] : HTML 해당 태그의 속성에 대한 첫 번째 정보를 가져옴

```
import requests
from bs4 import BeautifulSoup

req = requests.get("https://naver.com")
html =  req.text
soup = BeautifulSoup(html, 'html.parser')

print(soup.title)
print(soup.title.name)
print(soup.title.string)
```

```
<title>NAVER</title>
title
NAVER
```

```
print(soup.img)
print(soup.img['alt'])
print(soup.img['class'])
print(soup.img['height'])
```

```
<img alt="스포츠조선" class="api_logo" height="24" src=
4.png"/>
스포츠조선
['api_logo']
24
```

BeautifulSoup 모듈 사용법 : 검색

- find( ) : HTML의 해당 태그에 대한 첫 번째 정보를 가져옴
  - find(속성='값') : HTML 해당 속성과 일치하는 값에 대한 첫 번째 정보를 가져옴

```
print(soup.find('a'))
```

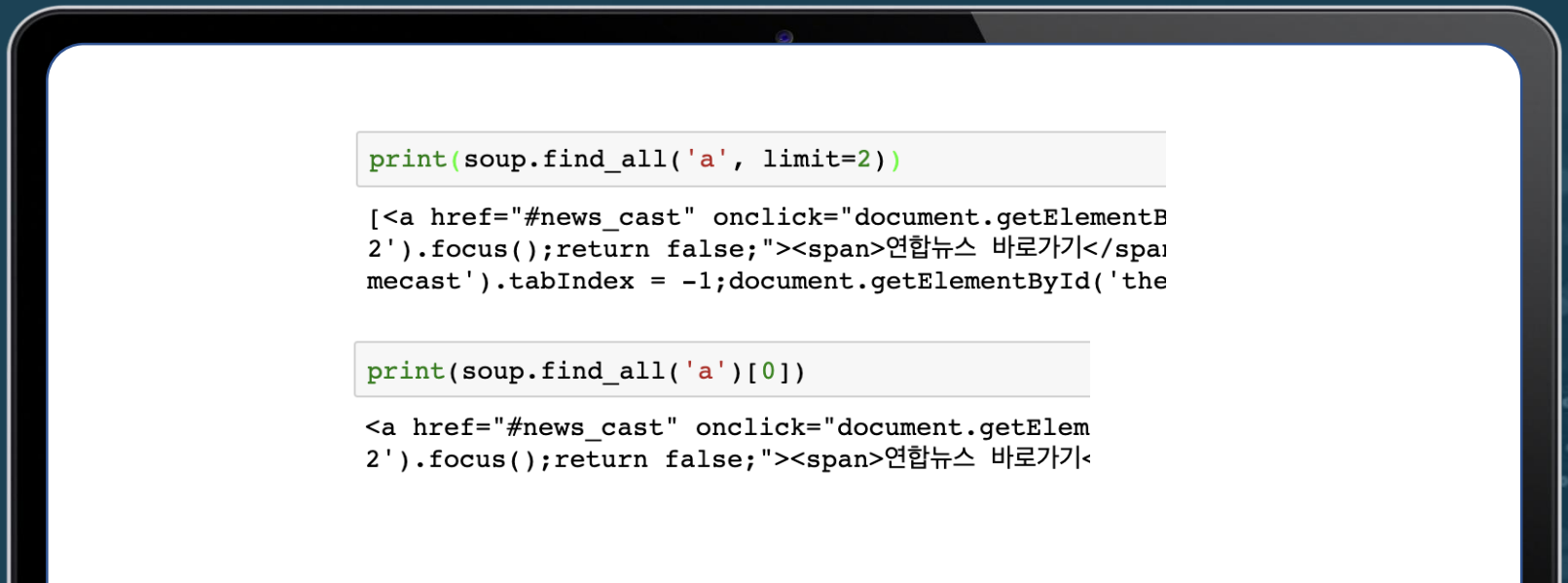
```
<a href="#news_cast" onclick="document.getElementById('new
2').focus();return false;"><span>연합뉴스 바로가기</span></a>
```

```
print(soup.find(id='search'))
```

```
<div class="search" id="search">
<!--자동완성 입력창-->
<form action="https://search.naver.com
<fieldset>
<legend class="blind">검색</legend>
```

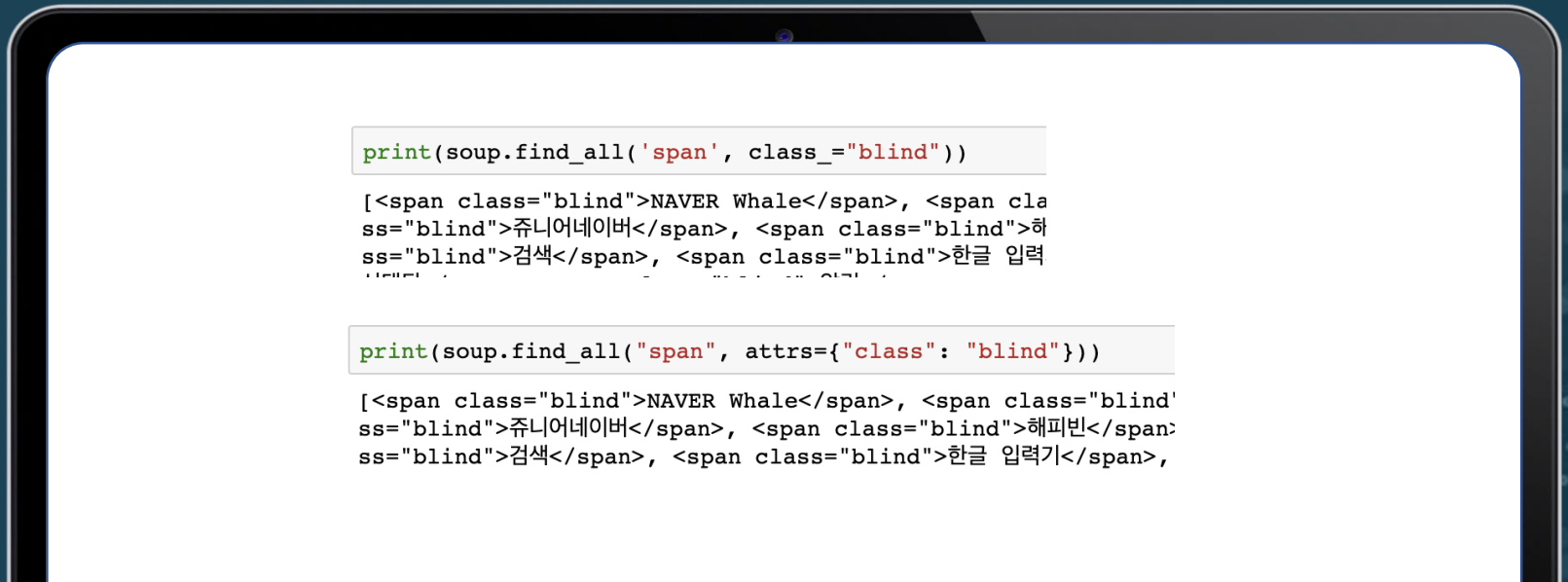
BeautifulSoup 모듈 사용법 : 검색

- find_all( ) : HTML의 해당 태그에 대한 모든 정보를 리스트 형식으로 가져옴
  - -limit 옵션으로 개수 지정 가능



BeautifulSoup 모듈 사용법 : 검색

- find_all( ) : HTML의 해당 태그에 대한 모든 정보를 리스트 형식으로 가져옴
  - -CSS 속성으로 필터링(class_로 클래스를 직접 사용 혹은 attrs에서 속성 = 값으로 필터링)



```
print(soup.find_all('span', class_="blind"))
```

```
[<span class="blind">NAVER Whale</span>, <span cla
ss="blind">쥬니어네이버</span>, <span class="blind">하
ss="blind">검색</span>, <span class="blind">한글 입력
```

```
print(soup.find_all("span", attrs={"class": "blind"}))
```

```
[<span class="blind">NAVER Whale</span>, <span class="blind'
ss="blind">쥬니어네이버</span>, <span class="blind">해피빈</span>
ss="blind">검색</span>, <span class="blind">한글 입력기</span>,
```

BeautifulSoup 모듈 사용법 : 검색

- find_all( ) : HTML의 해당 태그에 대한 모든 정보를 리스트 형식으로 가져옴
  - string으로 검색(해당 값이 있는지 없는지 검사 할 때 활용, 정규 표현식과 함께 활용)

```
print(soup.find_all(string='자동완성 끄기'))
```

```
['자동완성 끄기', '자동완성 끄기', '자동완성 끄기', '자동완성 끄기', '자동완성 끄기']
```

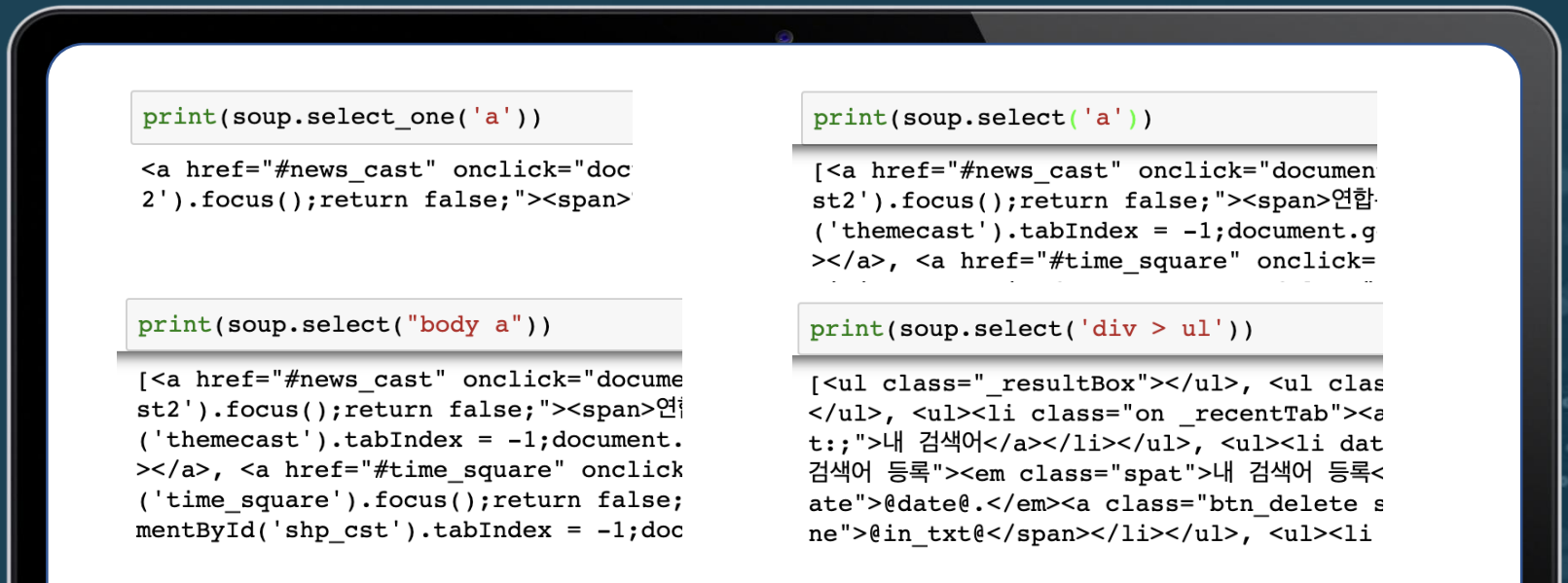
```
import re
print(soup.find_all(string=re.compile("네이버")))
```

```
[' 네이버 웨일로 차원이 다른 웹서핑을 경험해보세요!', '네이버',
'설정한 언론사의 기사들을 네이버 홈에서 바로 보실 수 있습니다.',
F', '네이버 랩스', '네이버 정책 및 약관', '네이버 정책']
```

BeautifulSoup 모듈 사용법 : 검색

- select_one( ) , select( )

  -CSS 선택자를 활용하여 원하는 정보를 가져옴(태그를 검색하는 find, find_all과 비슷함)



```
print(soup.select_one('a'))
```

```
<a href="#news_cast" onclick="doc
2').focus();return false;"><span>
```

```
print(soup.select('a'))
```

```
[<a href="#news_cast" onclick="documen
st2').focus();return false;"><span>연합
('themecast').tabIndex = -1;document.g
></a>, <a href="#time_square" onclick=
```

```
print(soup.select("body a"))
```

```
[<a href="#news_cast" onclick="docume
st2').focus();return false;"><span>연
('themecast').tabIndex = -1;document.
></a>, <a href="#time_square" onclick
('time_square').focus();return false;
mentById('shp_cst').tabIndex = -1;doc
```

```
print(soup.select('div > ul'))
```

```
[<ul class="_resultBox"></ul>, <ul clas
</ul>, <ul><li class="on _recentTab"><a
t:;">내 검색어</a></li></ul>, <ul><li dat
검색어 등록"><em class="spat">내 검색어 등록<
ate">@date@.</em><a class="btn_delete s
ne">@in_txt@</span></li></ul>, <ul><li
```

BeautifulSoup 모듈 사용법 : 가공

- get_text( ) : 검색 결과에서 태그를 제외한 텍스트만 출력

  -get('속성'): 해당 속성의 값을 출력

```
text = soup.find("span", attrs={"class": "blind"})
print(text)
print(text.get_text())
```

```
<span class="blind">NAVER Whale</span>
NAVER Whale
```

```
text = soup.find("span", attrs={"class": "blind"})
print(text)
print(text.get('class'))
```

```
<span class="blind">NAVER Whale</span>
['blind']
```

BeautifulSoup 모듈 사용법 : 가공

- string : 검색 결과에서 태그 안에 또 다른 태그가 없는 경우 해당 내용을 출력

```
text = """<li class="an_item">
<a class="an_a mn_checkout" data-clk="svc.pay" href="https://order.pay.naver.com/home">
<span class="an_icon"></span><span class="an_txt">네이버페이</span>
</a>
</li>"""
soup = BeautifulSoup(text, 'html.parser')
print(soup.string)
result=soup2.find('span',class_='an_txt')
print(result)
print(result.string)
```

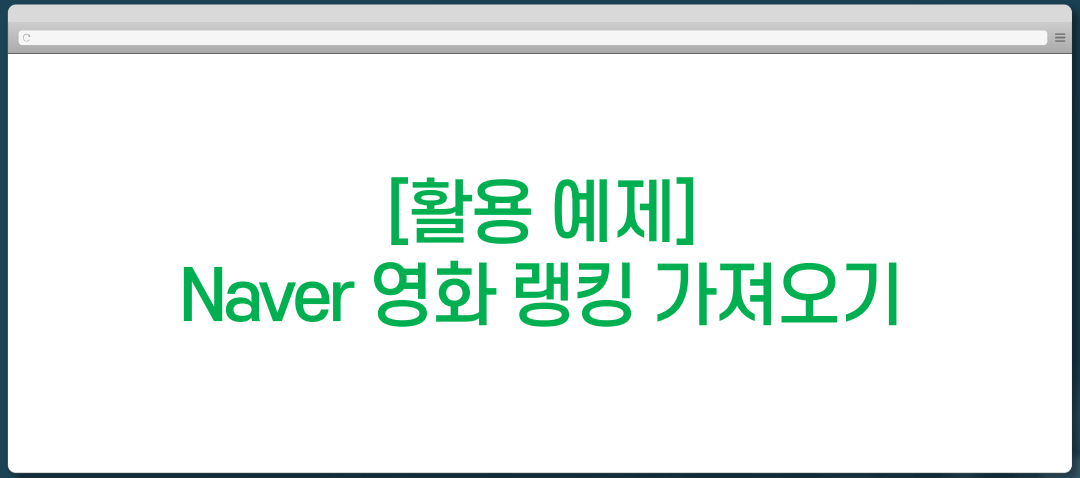
태그 안에 또 다른 태그가 있기 때문에 None 반환

유일한 태그이기 때문에 내용 출력

```
None
<span class="an_txt">네이버페이</span>
네이버페이
```

LESSON

BeautifulSoup 모듈 활용


[활용 예제]
Naver 영화 랭킹 가져오기

Naver 영화 랭킹 가져오기 Step1

① 홈페이지 텍스트 가져오기:
https://movie.naver.com/movie/sdb/rank/rmovie.nhn

Naver 영화 랭킹 가져오기 Step1

② BeautifulSoup으로 파싱하기

```
import requests
from bs4 import BeautifulSoup
req = requests.get("https://movie.naver.com/movie/sdb/rank/rmovie.nhn")
html =  req.text
soup = BeautifulSoup(html, 'html.parser')
```

Naver 영화 랭킹 가져오기 Step2

① 텍스트에서 영화 랭킹 찾기
② 영화 랭킹에 해당하는 부분의 태그 찾기

- ✓ td 태그, class는 title
- ✓ div 태그, class는 tit3
- ✓ a 태그, title 및 내용이 영화제목

```
<!-- 랭킹 리스트 -->
<table cellspacing="0" class="list_ranking">
    <caption class="blind">랭킹 테이블</caption>

<col width="6%"><col width="*"><col width="2%"><col width="4%">


<thead>
    <tr>
    <th scope="col">순위</th>
    <th scope="col">영화명</th>

    <th scope="col" colspan="2">변동폭</th>
    </tr>
</thead>
<tbody>
    <tr><td colspan="8" class="blank01"></td></tr>
    <!-- 예제
    <tr>
        <td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/common/bu
        <td class="title"><a href="#">트랜스포머</a></td>
        <td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/common/ic
        <td class="range ac">7</td>
    </tr>
    -->

    <tr>

            <td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/commo



        <td class="title">
            <div class="tit3">
                <a href="/movie/bi/mi/basic.nhn?code=187940" title="백두산">백두산</a>
            </div>
        </td>
        <!-- 평점순일 때 평점 추가하기  -->
```

Naver 영화 랭킹 가져오기 Step3

BeautifulSoup 함수로 데이터 가공

```
movie_ranking_list = soup.find_all('div',class_="tit3")
for i in range(len(movie_ranking_list)):
    print("{:2} 위 : {}".format(i+1,movie_ranking_list[i].get_text().strip()))
```

```
 1 위 : 백두산
 2 위 : 시동
 3 위 : 천문: 하늘에 묻는다
 4 위 : 캣츠
 5 위 : 포드 v 페라리
 6 위 : 겨울왕국 2
 7 위 : 나이브스 아웃
 8 위 : 쥬만지: 넥스트 레벨
 9 위 : 신비아파트 극장판 하늘도깨비 대 요르문간드
10 위 : 아내를 죽였다
```