

웹 스크래핑

네이버증권 실시간 검색상위종목 스크래핑

강사 : 경남대학교 신병주 교수





INDEX

1. 웹스크래핑 사전지식
2. 웹스크래핑 도구
3. 웹스크래핑 실습
4. 데이터 저장

1. 웹스크래핑 사전지식

웹스크래핑 (Web Scraping)

- 인터넷 웹페이지로부터 원하는 데이터를 추출하여 수집하는 데이터수집기법

스크랩(수집)



파싱(가공)



데이터저장

스크랩
- 해당 웹페이지
내용을 가져옴

파싱
- 가져온 내용중
서 필요한 내용만
추출

데이터저장
- 분석에 필요한
데이터타입을 설
정하고 원하는 형
태로 저장

2. 웹스크래핑 도구

I. 웹스크래핑 라이브러리 설치

```
!pip install beautifulsoup4
!pip install selenium
!pip install undetected-chromedriver
!pip install pandas
!pip install openpyxl
```

II. 웹스크래핑 라이브러리 불러오기

```
import pandas as pd
import undetected_chromedriver as uc
from selenium.webdriver.common.by import By
from selenium import webdriver
from bs4 import BeautifulSoup as bs
```

- **beautifulsoup4** : 스크랩한 데이터를 파싱할 때 사용

- **selenium** : 크롬과 같은 웹 브라우저를 조종하고 웹 페이지를 스크랩할 때 사용

- **undetected-chromedriver** : 실제 크롬 브라우저를 사용할 수 있는 드라이버를 자동으로 불러옴

- **pandas** : 데이터 분석 라이브러리

- **openpyxl** : pandas를 엑셀파일로 저장하기 위해 설치되어 있어야 되는 라이브러리

2. 웹스크래핑 도구

개발자 도구 (DevTools)

- 웹 브라우저에서 제공하는 기능으로, 웹 개발자들이 웹 페이지를 디버깅, 편집, 분석하고 성능을 측정하는 데 도움을 주는 도구 모음으로 **F12**를 누르면 확인할 수 있음
- 웹페이지의 **URL** 주소와 **HTML** 구조를 확인

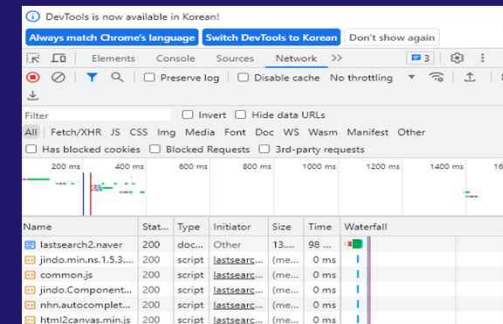
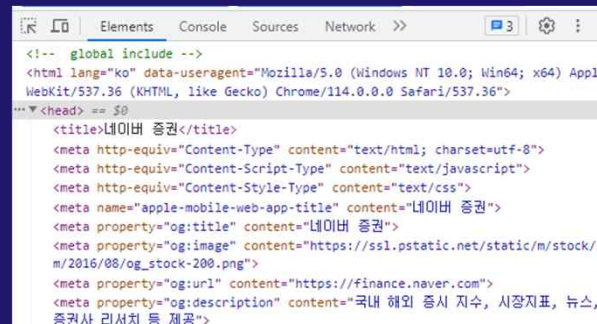
- url = 인터넷 웹페이지의 주소

예) <https://www.naver.com>



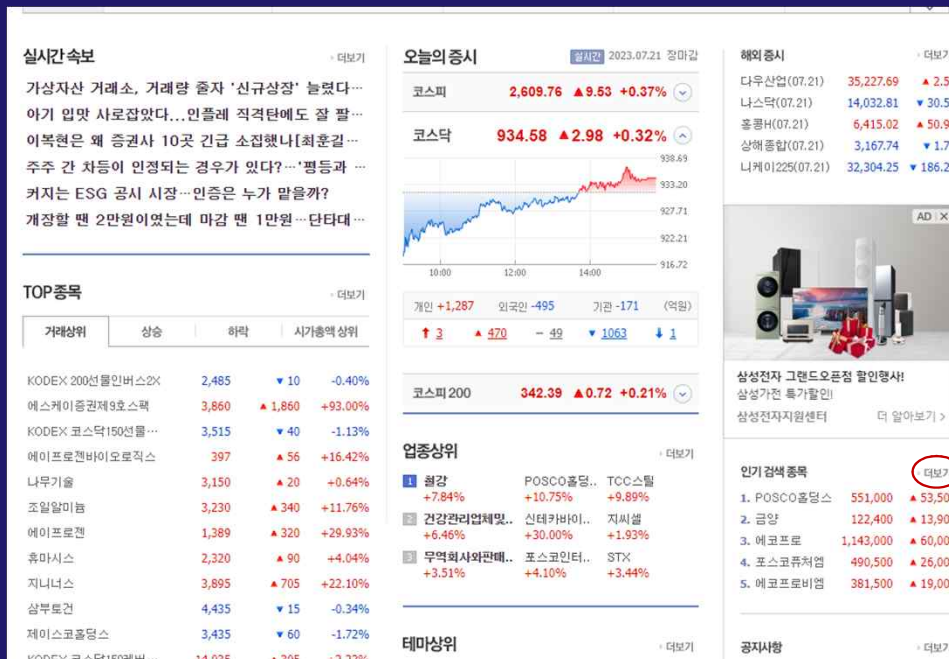
- HTML = 웹 페이지의 구조와 내용을 구성하고 있는 언어

해당 웹페이지가 어떤 HTML 구조를 가지고 있는지 파악해야 파싱을 할 수 있음



3. 웹스크래핑 실습

I. 네이버 증권 검색상위 종목 URL 주소 얻기



네이버 증권의 검색상위 URL

<https://finance.naver.com/sise/lastsearch2.naver>

증권종 > 국내증시 > 검색상위 종목

검색상위 종목

인기를 자유롭게 변경하실 수 있습니다. | 최대 6개까지 설정 가능합니다.

☒ 거래량
 ☐ 매수호가
 ☐ 거래대금(백만)
 ☐ 시가총액(억)
 ☐ 영업이익(억)
 ☒ PER(배)

☒ 시가
 ☐ 매도호가
 ☐ 전일거래량
 ☐ 자산총계(억)
 ☐ 영업이익증가율
 ☒ ROE(%)

☒ 고가
 ☐ 매수증진량
 ☐ 외국인비율
 ☐ 부채총계(억)
 ☐ 당기순이익(억)
 ☐ ROA(%)

☒ 저가
 ☐ 매도증진량
 ☐ 상장주식수(천주)
 ☐ 매출액(억)
 ☐ 주당순이익(원)
 ☐ PBR(배)

☐ 매출증가율
 ☐ 보유주당당금(원)
 ☐ 유보율(%)

적용하기 | 초기 항목으로

순위	종목명	검색비율	현재가	전일비	동작률	거래량	시가	고가	저가	PER	ROE
1	POSCO홀딩스	3.61%	551,000	▲ 53,500	+10.75%	3,423,271	486,000	555,000	483,500	21.85	6.11
2	에코프로	3.05%	1,143,000	▲ 60,000	+5.54%	1,344,156	1,036,000	1,190,000	1,035,000	77.39	2.94
3	에코프로비엠	1.80%	381,500	▲ 19,000	+5.24%	3,977,402	348,500	389,500	347,500	133.30	24.26
4	금강	1.53%	122,400	▲ 13,900	+12.81%	9,381,999	111,700	128,900	110,900	-198.06	-24.34
5	삼성전자	1.39%	70,300	▼ 700	-0.99%	16,496,206	70,400	70,400	69,400	10.61	17.07
6	SK이노베이션	1.25%	176,700	▲ 3,000	+1.73%	778,585	173,400	178,300	170,300	25.71	8.49
7	엘앤에프	0.83%	268,000	▼ 5,500	-2.01%	2,217,802	263,000	277,000	259,000	52.33	28.26
8	LG화학	0.83%	694,000	▲ 9,000	+1.31%	229,682	681,000	694,000	672,000	33.30	6.95
9	LG디스플레이	0.83%	14,080	▼ 360	-2.49%	1,944,032	14,370	14,370	14,020	-1.17	-26.71
10	이수퍼타시스	0.83%	36,500	▼ 650	-1.75%	5,150,973	35,550	36,900	35,100	22.19	58.92
11	코스모화학	0.83%	61,700	▲ 500	+0.82%	1,640,268	59,900	63,400	59,100	2,682.61	6.13
12	포스코퓨처엠	0.83%	490,500	▲ 26,000	+5.60%	2,070,312	448,000	496,000	447,500	313.62	4.87
13	신테카바이오	0.69%	12,870	↑ 2,970	+30.00%	3,260,022	10,200	12,870	10,120	-25.79	-15.25
14	유비온	0.69%	2,145	↑ 493	+29.94%	14,947,444	1,622	2,145	1,530	35.75	65.14
15	포스코DX	0.69%	31,250	▲ 200	+0.64%	16,323,249	30,450	32,100	30,200	84.92	11.84

3. 웹스크래핑 실습

II. HTML 구조 분석

- 수집할 데이터가 HTML 구조에 포함되어 있는지 확인하는 단계
- 크롬을 열고 **F12**를 누르면 개발자 도구를 열 수 있음
- 개발자 도구에서 C trl + F를 눌러 수집할 데이터가 HTML에 존재하는지 확인

순위	종목명	금액비율	현재가	전일비	동량률	거래량	시가	고가	저가	PER	ROE
1	에코프로	3.52%	1,143,000	▲ 60,000	+5.54%	1,344,156	1,036,000	1,190,000	1,035,000	77.39	2.94
2	POSCO홀딩스	2.58%	551,000	▲ 53,500	+10.75%	3,423,271	486,000	555,000	483,500	21.85	6.11
3	포스코DX	1.76%	31,250	▲ 200	+0.64%	16,323,249	30,450	32,100	30,200	84.92	11.84
4	금양	1.41%	122,400	▲ 13,900	+12.81%	9,381,999	111,700	128,900	110,900	-198.06	-24.34
5	엘앤에프	1.29%	268,000	▼ 5,500	-2.01%	2,217,802	263,000	277,000	259,000	52.33	28.26

개발자 도구를 열고 순위, 종목명 등 데이터가 HTML에 있는지 확인

```

<tr class="type1"> == $0
  <th>순위</th>
  <th>종목명</th>
  <th>경기비율</th>
  <th>현재가</th>
  <th class="tr" style="padding-right:8px">전월비</th>
  <th>동락률</th>
  <th>거래량</th>
  <th>시가</th>
  <th>고가</th>
  <th>저가</th>
  <th>PER</th>
  <th>ROE</th>
</tr>
<tr> <td></td>
<tr> <td></td>

```

```
<tr>
  <td class="no">2</td>
  <td>
    <a href="/item/main.naver?code=086520" class="title">에코프린
    </a> == $0
  </td>
  <td class="number">3.60%</td>
  <td class="number">1,143,000</td>
  > <td class="number">☹</td>
  > <td class="number">☹</td>
  <td class="number">1,344,156</td>
  <td class="number">1,036,000</td>
  <td class="number">1,190,000</td>
  <td class="number">1,035,000</td>
  <td class="number">77.39</td>
  <td class="number">2.94</td>
```

3. 웹스크래핑 실습

II. HTML 구조 분석

- 원하는 데이터 위치 파악



순위	종목명	금액비율	현재가	전일비	동락률	거래량	시가	고가	저가	PER	ROE
1	POSCO홀딩스	3.73%	551,000	▲ 53,500	+10.75%	3,423,271	486,000	555,000	483,500	21.85	6.1
2	에코프로	3.60%	1,143,000	▲ 60,000	+5.54%	1,344,156	1,036,000	1,190,000	1,035,000	77.39	2.9
3	에코프로비엠	1.52%	381,500	▲ 19,000	+5.24%	3,977,402	348,500	389,500	347,500	133.30	24.2
4	금강	1.38%	122,400	▲ 13,900	+12.81%	9,381,999	111,700	128,900	110,900	-198.06	-24.3
5	SK이노베이션	1.11%	176,700	▲ 3,000	+1.73%	778,585	173,400	178,300	170,300	25.71	8.4
6	코스모화학	1.11%	61,700	▲ 500	+0.82%	1,660,268	59,900	63,400	59,100	2,682.61	6.1
7	엘앤에프	0.97%	268,000	▼ 5,500	-2.01%	2,217,802	263,000	277,000	259,000	52.33	28.2
8	포스코DX	0.97%	31,250	▲ 200	+0.64%	16,323,249	30,450	32,100	30,200	84.92	11.8
9	삼성전자	0.83%	70,300	▼ 700	-0.99%	16,496,206	70,400	70,400	69,400	10.61	17.0
10	포스코인터내셔널	0.69%	50,800	▲ 2,000	+4.10%	3,630,532	47,800	51,300	46,500	11.34	15.7
11	조일월미늄	0.69%	3,230	▲ 340	+11.76%	31,287,294	3,055	3,270	2,980	-248.46	10.0
12	삼성중공업	0.69%	8,590	▲ 60	+0.70%	8,818,932	8,530	8,770	8,480	-14.97	-16.0
13	포스코엡텍	0.69%	33,500	▲ 650	+1.98%	4,665,408	31,700	33,850	31,350	620.37	4.4
14	에코프로에미치켄	0.55%	91,400	▲ 5,100	+5.91%	5,620,980	86,300	95,600	85,500	33.43	43.3
15	루닛	0.55%	170,000	▼ 14,000	-7.61%	1,074,196	183,500	183,500	170,000	-103.60	-110.0

각 종목들은 class가 'type_5'인 table안에 tr이라는 태그안에 3번째부터 들어있음

```
...
<table cellpadding="0" cellspacing="0" class="type_5">== $0
  <colgroup>...</colgroup>
  <tbody>
    <tr class="type1">...</tr>
    <tr>...</tr>
    <tr>...</tr>
  <tr>
    <td class="no">2</td>
    <td>
      <a href="/item/main.naver?code=086520" class="title">에코프로
      </a>
    </td>
  </tr>
</table>

```


3. 웹스크래핑 실습

III. 코드작성

- selenium을 사용하여 페이지 불러오기

```
driver = uc.Chrome()
```

```
url = 'https://finance.naver.com/sise/lastsearch2.naver'
```

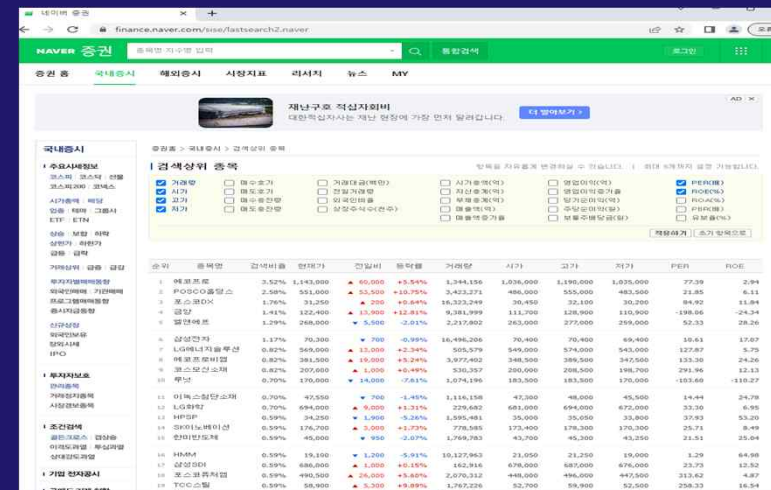
```
driver.get(url)
```

- 페이지 HTML 가져오기

```
html = driver.page_source
```

```
print(html)
```

크롬 드라이버 결과화면



HTML 가져오기 결과화면



3. 웹스크래핑 실습

III. 코드작성

- BeautifulSoup를 사용하여 클래스가 'type_5'인 'table'을 파싱

```
soup = bs(html)
type_5 = soup.find('table', {'class':'type_5'})
type_5
```

- 파싱한 'type_5'에서 여러개의 'tr' 태그들을 파싱

```
tr_list = type_5.find_all('tr')
print(tr_list[2])
print(tr_list[3])
```

find를 사용하면 제일 위에 있는 1개의 HTML을 파싱함

```
Out [24]: <table cellpadding="0" cellspacing="0" class="type_5">
  <colgroup><col width="4%" />
  <col width="*" />
  <col width="7%" />
  <col width="7%" />
  <col width="9%" />
  <col width="7%" />
  <col width="9%" />
  <col width="9%" />
  <col width="9%" />
  <col width="9%" />
  <col width="9%" />
  </colgroup><tbody><tr class="type1">
  <th>순위</th>
  <th>종목명</th>
  <th>검색비율</th>
  <th>현재가</th>
  <th class="tr" style="padding-right:8px">전일비</th>
```

find_all을 사용하여 여러 개의 tr 태그들을 모두 리스트로 가져온 모습

```
<tr>
  <td class="no">1</td>
  <td><a class="title" href="/item/main.naver?code=005490">P0000홀딩스</a></td>
  <td class="number">2.39</td>
  <td class="number">551,000</td>
  <td class="number">
    <span class="tah pill red02">
      53,500
    </span>
  </td>
  <td class="number">
    <span class="tah pill red01">
      +10.75%
    </span>
  </td>
  <td class="number">3,423,271</td>
  <td class="number">495,000</td>
  <td class="number">555,000</td>
  <td class="number">489,500</td>
  <td class="number">21.85</td>
  <td class="number">6.11</td>
</tr>
  <td class="no">2</td>
  <td><a class="title" href="/item/main.naver?code=247540">에코프로비엠</a></td>
```

3. 웹스크래핑 실습

III. 코드작성

- tr_list는 tr이 들어있는 **리스트**형태이므로 슬라이싱이 가능함
- 종목 데이터는 'tr' 태그중에서 3번째부터 시작하므로 1,2번째는 제거
- 반복문을 돌면서 tr_list안에 있는 'tr'태그들을 차례대로 꺼냄

```
tr_list = tr_list[2:]
for tr_index in tr_list :
    print(tr_index )
```

- tr 내부에서는 순위 , 종목명 등이 **여러 개의 td 태그**로 이루어져 있는것을 확인
- 즉 1개의 tr 태그 안에는 여러 개의 td 태그가 존재

반복문을 돌며 차례대로 출력되는 tr 태그 결과화면

```
<tr>
<td class="no">1</td>
<td><a class="title" href="/item/main.naver?code=005490">POSCO홀딩스</a></td>
<td class="number">2.45%</td>
<td class="number">551,000</td>
<td class="number">
<span class="tah p11 red02">
53,500
</span>
</td>
<td class="number">
<span class="tah p11 red01">
+10.75%
</span>
</td>
<td class="number">3,423,271</td>
<td class="number">486,000</td>
<td class="number">555,000</td>
<td class="number">489,500</td>
```

3. 웹스크래핑 실습

III. 코드작성

- tr 태그 내부의 여러 개의 td 태그들을 find_all을 통해 td_list에 저장

```
for tr_index in tr_list :  
    td_list = tr_index.find_all('td')  
    print(td_list)
```

- 반복문을 통해 td_list 안의 내용을 출력
- text는 태그의 내용물을 가져오고 strip()은 공백을 제거함

```
for tr_index in tr_list :  
    td_list = tr_index.find_all('td')  
    print('-----')  
    for td_index in td_list:  
        print(td_index.text.strip())
```

tr 내부의 여러 개의 td 태그 결과화면

```
for tr_index in tr_list :  
    td_list = tr_index.find_all('td')  
    print(td_list)
```

[<td class="no">1</td>, <td>POSCO홀딩스</td>, <td class="number">2.39%</td>, <td class="number">551,000</td>, <td class="number">

53,500

</td>, <td class="number">

+10.75%

</td>, <td class="number">3,423,271</td>, <td class="number">486,000</td>, <td class="number">
<td class="number">483,500</td>, <td class="number">21.85</td>, <td class="number">6.11</td>]
[<td class="no">2</td>, <td>에코프로비엠</td>, <td class="number">2.12%</td>, <td class="number">381,500</td>, <td class="number">

19,000

</td>]

text로 td 태그의 내용물과 strip()을 통해 공백을 제거한 모습

```
for tr_index in tr_list :  
    td_list = tr_index.find_all('td')  
    print('-----')  
    for td_index in td_list:  
        print(td_index.text.strip())
```

1

POSCO홀딩스
2.39%
551,000
53,500
+10.75%
3,423,271
486,000
555,000
483,500
21.85
6.11

2
에코프로비엠
2.12%
381,500
19,000
15,240

3. 웹스크래핑 실습

III. 코드작성

- 엑셀에 저장하기 위한 코드 작성

```
Data = pd.DataFrame(columns=['순위','종목명','검색비율','현재가',  
                             '전일비','등락률','거래량','시가','고가','저가','PER','ROE'])  
  
Data_num = 0  
for tr_index in tr_list:  
    td_list = tr_index.find_all('td')  
    print('-----')  
    data_list = []  
    for td_index in td_list:  
        print(td_index.text.strip())  
        data_list.append(td_index.text.strip())  
    Data.loc[Data_num] = data_list  
    Data_num += 1
```

Data에 5순위인 삼성전자까지만 저장된 모습

순위		종목명	검색비율	현재가	전일비	등락률	거래량	시가	고가	저가	PER	ROE
0	1	POSCO홀딩스	2.39%	551,000	53,500	+10.75%	3,423,271	486,000	555,000	483,500	21.85	6.11
1	2	에코프로비엠	2.12%	381,500	19,000	+5.24%	3,977,402	348,500	389,500	347,500	133.30	24.26
2	3	에코프로	1.59%	1,143,000	60,000	+5.54%	1,344,156	1,036,000	1,190,000	1,035,000	77.39	2.94
3	4	HMM	1.06%	19,100	1,200	-5.91%	10,127,963	21,050	21,250	19,000	1.29	64.98
4	5	삼성전자	1.06%	70,300	700	-0.99%	16,496,206	70,400	70,400	69,400	10.61	17.07

해당 코드를 실행하면 오류가 뜨게 되는데 HTML을 자세히보면 td에 잘못된 데이터인 선이 들어가있는 것을 볼 수 있음

선은 데이터프레임에 저장 안 됨

td.blank_06 810x6		1.06%	19,100	▼ 1,200	-5.91%	10,127,963	21,050	21,250	19,000	1.29	64.98
		1.06%	70,300	▼ 700	-0.99%	16,496,206	70,400	70,400	69,400	10.61	17.07
6	하나기술	0.80%	126,000	▲ 6,200	+5.18%	751,878	117,300	133,500	117,200	1,272.73	2.14
7	LX세미콘	0.80%	112,800	▼ 3,700	-3.18%	204,360	115,300	115,800	111,300	10.66	26.70
8	엘앤에프	0.80%	268,000	▼ 5,500	-2.01%	2,217,802	263,000	277,000	259,000	52.33	28.26
9	NAVER	0.80%	206,000	▲ 1,500	+0.73%	627,606	202,500	206,500	200,500	52.47	3.29
10	SK텔레콤	0.80%	45,500	▲ 350	+0.78%	289,713	45,150	45,800	45,100	10.04	7.97

3. 웹스크래핑 실습

성공적으로 Data 에 저장되었음

III. 코드작성

- 파이썬 try except 구문을 활용
- 데이터프레임이 저장이 안 돼서 에러가 발생할 경우 무시하고 진행하는 코드

```
for td_index in td_list:
    print(td_index.text.strip())
    data_list.append(td_index.text.strip())
try:
    Data.loc[Data_num] = data_list
except:
    continue
Data_num += 1
```

Data												
순위		종목명	검색비율	현재가	전일비	등락률	거래량	시가	고가	저가	PER	ROE
0	1	POSCO홀딩스	2.39%	551,000	53,500	+10.75%	3,423,271	486,000	555,000	483,500	21.85	6.11
1	2	에코프로비엠텐	2.12%	381,500	19,000	+5.24%	3,977,402	348,500	389,500	347,500	133.30	24.26
2	3	에코프로	1.59%	1,143,000	60,000	+5.54%	1,344,156	1,036,000	1,190,000	1,035,000	77.39	2.94
3	4	HMM	1.06%	19,100	1,200	-5.91%	10,127,963	21,050	21,250	19,000	1.29	64.98
4	5	삼성전자	1.06%	70,300	700	-0.99%	16,496,206	70,400	70,400	69,400	10.61	17.07
5	6	하나기술	0.80%	126,000	6,200	+5.18%	751,878	117,300	133,500	117,200	1,272.73	2.14
6	7	LX세미콘	0.80%	112,800	3,700	-3.18%	204,360	115,300	115,800	111,300	10.66	26.70
7	8	엘앤에프	0.80%	268,000	5,500	-2.01%	2,217,802	263,000	277,000	259,000	52.33	28.26
8	9	NAVER	0.80%	206,000	1,500	+0.73%	627,606	202,500	206,500	200,500	52.47	3.29
9	10	SK텔레콤	0.80%	45,500	350	+0.78%	289,713	45,150	45,800	45,100	10.04	7.97
10	11	신세계	0.80%	191,000	400	+0.21%	28,994	189,400	191,300	188,500	4.80	10.34
11	12	SK하이닉스	0.80%	115,100	1,800	-1.54%	2,726,182	114,700	115,100	113,200	-35.97	3.56
12	13	에코프로에이치엔	0.53%	91,400	5,100	+5.91%	5,620,980	86,300	95,600	85,500	33.43	43.37
13	14	LG에너지솔루션	0.53%	569,000	13,000	+2.34%	505,579	549,000	574,000	543,000	127.87	5.75
14	15	와이팜	0.53%	4,890	60	+1.24%	962,059	4,710	4,955	4,710	543.33	-13.67
15	16	SK바이오팜	0.53%	85,400	0	0.00%	233,409	85,400	87,000	84,000	-62.75	-36.66
16	17	제이엘케이	0.53%	31,600	700	-2.17%	1,830,526	34,000	35,700	31,000	-73.15	-44.28
17	18	케이엔케이	0.53%	23,500	1,050	-4.28%	112,190	23,850	23,850	23,000	14.56	20.42
18	19	KODEX 미국달러 선물레버리지	0.53%	12,035	245	+2.08%	139,905	11,905	12,045	11,885	N/A	N/A
19	20	메드락토	0.53%	15,170	820	+5.71%	340,825	14,510	15,350	14,420	-7.12	-94.26
20	21	스튜디오산타클로 스	0.53%	465	18	+4.03%	6,523,614	441	513	435	-1.01	-53.22
21	22	에이디테크놀로지	0.53%	27,600	1,300	-4.50%	1,792,061	28,700	29,300	27,200	-484.21	3.61
22	23	풍산	0.53%	40,100	700	-1.72%	89,579	40,600	40,650	39,900	6.04	9.94
23	24	SK이노베이션	0.53%	176,700	3,000	+1.73%	778,585	173,400	178,300	170,300	25.71	8.49
24	25	오킨스전자	0.53%	9,780	4,180	-29.94%	1,553,314	13,720	13,920	9,780	63.10	8.36
25	26	케이엔솔	0.53%	20,200	1,980	+10.87%	4,388,493	18,050	21,900	18,040	27.86	10.51
26	27	포스코인터내셔널	0.53%	50,800	2,000	+4.10%	3,630,532	47,800	51,300	46,500	11.34	15.79
27	28	카카오	0.53%	50,600	100	+0.20%	1,192,125	50,000	50,800	49,750	176.92	13.54
28	29	SK	0.53%	148,200	400	+0.27%	134,873	147,100	148,300	144,100	-7,410.00	5.13
29	30	포스코DX	0.53%	31,250	200	+0.64%	16,323,249	30,450	32,100	30,200	84.92	11.84

4. 데이터 저장

I. 데이터 확인

- 저장된 데이터의 타입을 확인하고 수정

```
print(Data.info())
```

- 비교 분석이 편하도록 데이터 타입을 수정해줄 필요성이 있음

순위 -> **int** 정수형 타입

PER, ROE -> N/A를 제거하고 **float** 소수점 타입

검색비율, 등락률 -> %를 제거하고 **float** 소수점 타입

현재가, 전일비, 거래량, 시가, 고가, 저가 -> ','를 제거하고 **int** 정수형 타입

Data.info() 결과 -> 모든 열이 object로 되어있음

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 30 entries, 0 to 29
Data columns (total 12 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   순위        30 non-null    object
1   종목명      30 non-null    object
2   검색비율    30 non-null    object
3   현재가      30 non-null    object
4   전일비      30 non-null    object
5   등락률      30 non-null    object
6   거래량      30 non-null    object
7   시가        30 non-null    object
8   고가        30 non-null    object
9   저가        30 non-null    object
10  PER         30 non-null    object
11  ROE         30 non-null    object
dtypes: object(12)
memory usage: 4.1+ KB
None
```

4. 데이터 저장

II. 데이터 타입변경 후 엑셀로 저장

```
Data['순위'] = Data['순위'].astype('int')  
Data[['PER', 'ROE']] = Data[['PER', 'ROE']].replace('N/A', None, regex=True).astype('float')  
Data[['검색비율', '등락률']] = Data[['검색비율', '등락률']].replace('%', '', regex=True).astype('float')  
Data[['현재가', '전일비', '거래량', '시가', '고가', '저가']] =  
\Data[['현재가', '전일비', '거래량', '시가', '고가', '저가']].replace(',', '', regex=True).astype('int')
```

```
print(Data.info())  
Data.to_excel('네이버증권_검색상위종목.xlsx', index=False)  
  
<class 'pandas.core.frame.DataFrame'>  
Int64Index: 30 entries, 0 to 29  
Data columns (total 12 columns):  
#   Column      Non-Null Count  Dtype  
---  -  
0   순위        30 non-null      int32  
1   종목명      30 non-null      object  
2   검색비율    30 non-null      float64  
3   현재가      30 non-null      int32  
4   전일비      30 non-null      int32  
5   등락률      30 non-null      float64  
6   거래량      30 non-null      int32  
7   시가        30 non-null      int32  
8   고가        30 non-null      int32  
9   저가        30 non-null      int32  
10  PER         29 non-null      float64  
11  ROE         28 non-null      float64  
dtypes: float64(4), int32(7), object(1)  
memory usage: 3.3+ KB  
None
```



A	B	C	D	E	F	G	H	I	J	K	L
순위	종목명	검색비율	현재가	전일비	등락률	거래량	시가	고가	저가	PER	ROE
1	POSCO홀	2.05	551000	53500	10.75	3423271	486000	555000	483500	21.85	6.11
2	SK바이오	0.88	77500	1200	1.57	362648	76900	79500	76800	73.67	
3	에코프로	0.88	381500	19000	5.24	3977402	348500	389500	347500	133.3	24.26
4	SK이노베	0.88	176700	3000	1.73	778585	173400	178300	170300	25.71	8.49
5	코스모신	0.88	207000	1000	0.49	530357	200000	208500	198700	291.96	12.13
6	포스코퓨	0.88	490500	26000	5.6	2070312	448000	496000	447500	313.62	4.87
7	노을	0.58	5200	630	13.79	3058936	4570	5790	4455	-3.8	-206.45
8	비율	0.58	6970	150	-2.11	955003	7070	7250	6950	28.33	35.06
9	피플바이	0.58	9100	410	4.72	500039	8690	9630	8550	-6.48	-136.26
10	메지온	0.58	41800	1400	3.47	412426	40000	42100	39800	-32.3	-81.54
11	에코프로	0.58	1143000	60000	5.54	1344156	1036000	1190000	1035000	77.39	2.94
12	리튬포어	0.58	19650	700	-3.44	639792	20000	20500	19640	2.64	82.34
13	셀트리온	0.58	151000	900	0.6	403039	149900	153100	149900	37.79	13.35
14	엘앤에프	0.58	268000	5500	-2.01	2217802	263000	277000	259000	52.33	28.26
15	새로닉스	0.58	46300	1400	-2.94	311921	45850	47700	44550	11.92	58.93
16	에코바이	0.58	5450	50	-0.91	89436	5490	5530	5350	-12.33	-6.63
17	LG디스플	0.58	14080	360	-2.49	1944032	14370	14370	14020	-1.17	-26.71
18	기업은행	0.58	10240	80	0.79	505387	10130	10240	10130	2.88	9.85

감사합니다.