

네이버 함께 많이 찾는 주제어 수집

- 정적(static) 웹페이지 데이터 수집
- BeautifulSoup을 이용하여 HTML 문자열 데이터 parsing

```
In [1]: import pandas as pd
import requests
from bs4 import BeautifulSoup
```

```
In [2]: # 1. 웹페이지 분석 : URL
```

```
In [3]: query = "삼성전자"
url = f"https://search.naver.com/search.naver?query={query}"
print(url)
```

https://search.naver.com/search.naver?query=삼성전자

```
In [4]: # 2. request(URL) > response : str(html)
```

```
In [5]: response = requests.get(url)
response
```

```
Out[5]: <Response [200]>
```

```
In [6]: response.text[:250]
```

```
Out[6]: '<!doctype html> <html lang="ko"><head> <meta charset="utf-8"> <meta name
="referrer" content="always"> <meta name="format-detection" content="telep
hone=no,address=no,email=no"> <meta property="og:title" content="삼성전자 : 네
이버 검색"/> <meta property="og:im'
```

```
In [7]: # 3. str(html) > bs object
```

```
In [8]: dom = BeautifulSoup(response.text, "html.parser")
type(dom)
```

```
Out[8]: bs4.BeautifulSoup
```

```
In [9]: # 4. bs object > .select(css-selector), .select_one(css-selector) > str(text)
```

```
In [10]: # 10개의 엘리먼트들 선택
# select() : list(Tag, Tag)
# select_one() : Tag
elements = dom.select(".fds-refine-query-grid a")
len(elements)
```

```
Out[10]: 6
```

```
In [11]: elements[0].text
```

```
Out[11]: '삼성전자주가'
```

```
In [12]: element = elements[0]
link = element.get("href")
link
```

```
Out[12]: '?where=nexearch&sm=tab_clk.rqT&query=%EC%82%BC%EC%84%B1%EC%A0%84%EC%9E%90%EC%A3%BC%EA%B0%80'
```

```
In [13]: element.text.strip()
```

```
Out[13]: '삼성전자주가'
```

```
In [14]: # 각각의 엘리먼트에서 text 데이터 수집
```

```
In [15]: keywords = []
for element in elements:
    keyword = element.text
    keywords.append(keyword)
print(keywords)
```

```
['삼성전자주가', '오늘 삼성전자 주가', '삼성전자주식가격', '삼성전자 주식', '삼성전자 주가 전
망', '삼성전자 주식전망']
```

```
In [16]: keywords = [element.text for element in elements]
print(keywords)
```

```
['삼성전자주가', '오늘 삼성전자 주가', '삼성전자주식가격', '삼성전자 주식', '삼성전자 주가 전
망', '삼성전자 주식전망']
```

```
In [17]: # 5. str(text) > DataFrame
```

```
In [18]: df = pd.DataFrame({"keywords": keywords})
df["query"] = query
df.tail(2)
```

```
Out[18]:
```

	keywords	query
4	삼성전자 주가 전망	삼성전자
5	삼성전자 주식전망	삼성전자

```
In [19]: # 현재 시간 데이터 추가
```

```
In [20]: from datetime import datetime
```

```
In [21]: now = datetime.now()
now = now.strftime("%Y-%m-%d %H:%M")
now
```

```
Out[21]: '2024-09-11 21:12'
```

```
In [22]: df["date_time"] = now
df.tail(2)
```

```
Out[22]:
```

	keywords	query	date_time
4	삼성전자 주가 전망	삼성전자	2024-09-11 21:12
5	삼성전자 주식전망	삼성전자	2024-09-11 21:12

```
In [23]: # query를 입력하면 데이터 프레임을 출력하는 함수
def naver_relate_keyword(query):
    url = f"https://search.naver.com/search.naver?query={query}"
    response = requests.get(url)
    dom = BeautifulSoup(response.text, "html.parser")
```

```

elements = dom.select(".fds-refine-query-grid a")
keywords = [element.text for element in elements]

df = pd.DataFrame({"keywords": keywords})
df["query"] = query

now = datetime.now()
now = now.strftime("%Y-%m-%d %H:%M")
df["date_time"] = now

return df

```

```

In [24]: query = "삼성전자"
df = naver_relate_keyword(query)
df.tail(2)

```

```

Out[24]:

```

	keywords	query	date_time
4	삼성전자 추가 전망	삼성전자	2024-09-11 21:12
5	삼성전자 주식전망	삼성전자	2024-09-11 21:12

```

In [25]: dfs = []
queries = ["삼성전자", "LG전자"]

for query in queries:
    print(query, end=" ")
    df = naver_relate_keyword(query)
    dfs.append(df)

result = pd.concat(dfs, ignore_index=True)
result

```

삼성전자 LG전자

```

Out[25]:

```

	keywords	query	date_time
0	삼성전자추가	삼성전자	2024-09-11 21:12
1	오늘 삼성전자 추가	삼성전자	2024-09-11 21:12
2	삼성전자주식가격	삼성전자	2024-09-11 21:12
3	삼성전자 주식	삼성전자	2024-09-11 21:12
4	삼성전자 추가 전망	삼성전자	2024-09-11 21:12
5	삼성전자 주식전망	삼성전자	2024-09-11 21:12
6	Lg전자서비스센터	LG전자	2024-09-11 21:12
7	lg전자 서비스센터 전화번호	LG전자	2024-09-11 21:12
8	lg전자추가전망	LG전자	2024-09-11 21:12
9	lg전자 추가	LG전자	2024-09-11 21:12
10	lg전자 에어컨 서비스센터	LG전자	2024-09-11 21:12
11	lg전자 목표주가	LG전자	2024-09-11 21:12