

Crawling Day 4

네이버 영화 순위/영화/평점 크롤링

by 이혁주

```
import requests
from bs4 import BeautifulSoup as bs

html = bs(requests.get("https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=cur&date=20230302").text)
title = html.select('table.list_ranking > tbody > tr > td.title')
point = html.select('table.list_ranking > tbody > tr > td.point')

✓for i in range(len(title)):
|   print(i + 1, title[i].text.strip(), point[i].text.strip())
✓ 0.2s
```

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```
1 아임 히어로 더 파이널 9.93
2 탑건: 매버릭 9.76
3 다음 소희 9.47
4 더 퍼스트 슬램덩크 9.45
5 올빼미 9.30
6 극장판 주술회전 0 9.21
7 러브레터 9.14
8 탄생 9.06
9 비긴 어게인 9.04
10 영웅 8.99
11 극장판 소드 아트 온라인 -오디널 스케일- 8.95
12 하녀 8.91
13 극장판 5등분의 신부 8.91
14 카운트 8.90
15 서치 2 8.90
16 장화신은 고양이: 끝내주는 모험 8.86
17 엘비스 8.83
18 너의 이름은. 8.81
19 여름날 우리 8.77
20 항거:유관순 이야기 8.66
21 아바타: 물의 길 8.65
22 헤어질 결심 8.64
```

스타벅스 크롤링 & 분석

```
# 모듈 임포트

from selenium import webdriver
import time
from bs4 import BeautifulSoup as bs
from selenium.webdriver.common.by import By
import pandas as pd

# 연결 주소 설정
```

```

url = '<https://www.starbucks.co.kr/store/store_map.do>'
driver = webdriver.Chrome('chromedriver.exe')
driver.get(url)

# 웹페이지에서 단추를 눌러서 원하는 자료로 이동

area_btn = '#container > div > form > fieldset > div > section > article.find_store_co
nt > article > header.loca_search > h3 > a'
driver.find_element(By.CSS_SELECTOR, area_btn).click()

time.sleep(2)
seoul_btn = '#container > div > form > fieldset > div > section > article.find_store_c
ont > article > article:nth-child(4) > div.loca_step1 > div.loca_step1_cont > ul > li:
nth-child(1) > a'
driver.find_element(By.CSS_SELECTOR, seoul_btn).click()

time.sleep(2)
all_btn = '#mCSB_2_container > ul > li:nth-child(1) > a'
driver.find_element(By.CSS_SELECTOR, all_btn).click()

# 웹 페이지 긁어오기

txt = driver.page_source
html = bs(txt)

# 전체 코드에서 필요한 부분만 추출

shops = html.select('ul.quickSearchResultBoxSidoGugun > li.quickResultLstCon')

# 한개 지점만 일단 추출

shop = shops[0]
shop

# 필요한 데이터만 추출

# 위도 : data-lat 딕셔너리 형태로 추출 즉 키값 밸류값
lat = shop['data-lat']

# 경도 : data-long 딕셔너리 형태로 추출 즉 키값 밸류값
lng = shop['data-long']

# 지점명 : data-name 딕셔너리 형태로 추출 즉 키값 밸류값
name = shop['data-name']

# <p class="result_details">서울특별시 마포구 양화로 105 (서교동)<br/>1522-3232</p>

# 주소 : 딕셔너리 형태가 아니기 때문에 select으로 추출
addr = str(shop.select('p.result_details')[0]).split('<br/>')[0].split('>')[1]

# 전화번호 : 주소와 전화번호는 묶여 있으므로 복잡하게 추출
tel = str(shop.select('p.result_details')[0]).split('<br/>')[1].split('<')[0]

# 매장종류 : 딕셔너리 형태가 아니기 때문에 select으로 추출
stype = shop.select('i.pin_general')[0].text

# 모든 지점의 필요 데이터 추출

```

```

starbucks_list = []

for shop in shops:
    lat = shop['data-lat']
    lng = shop['data-long']
    name = shop['data-name']
    addr = str(shop.select('p.result_details')[0]).split('<br/>')[0].split('>')[1]
    tel = str(shop.select('p.result_details')[0]).split('<br/>')[1].split('<')[0]
    stype = shop.select('i')[0].text

    time.sleep(0.01)
    starbucks_list.append([name, lat, lng, addr, tel, stype])

print(starbucks_list)

# 데이터 프레임으로 전환
df = pd.DataFrame(starbucks_list, columns = ['매장명', '위도', '경도', '주소', '전화번호', '매장종류'])
df

```

```

#####
#                               데이터 분석                               #
#####

#!pip install folium

import folium

# 기본 지도 그리기 연습

starmap = folium.Map(
    location=[37.5666805, 126.9784147], # 구글에서 검색한 서울 시청 좌표
    zoom_start=11,
    tiles='Stamen Terrain'
)

folium.CircleMarker(
    location=[37.5666805, 126.9784147],
    fill = True
).add_to(starmap)

starmap

# 스타벅스 600개 지점 지도 그리기

import folium

starmap = folium.Map(
    location=[37.5666805, 126.9784147], # 구글에서 검색한 서울 시청 좌표
    zoom_start=11,
    tiles='Stamen Terrain'
)

```

```

# 데이터 프레임에서 위도와 경도 값을 가져와서 600번 반복

    for idx in df.index:
        lat = df.loc[idx, '위도']
        lng = df.loc[idx, '경도']

        folium.Marker(
            location=[lat, lng],
            fill = True
        ).add_to(starmap)

    starmap

# 권두현
    for shop in shops:
        lat = shop['data-lat']
        lng = shop['data-long']
        name = shop['data-name']
        addr = str(shop.select('p.result_details')[0]).split('<br/>')[0].split('>')
[1]
[0]
        tel = str(shop.select('p.result_details')[0]).split('<br/>')[1].split('<')
        shoptype = shop.select('i')[0].text

        time.sleep(0.1)
        starbucks_list.append([name, lat, lng, addr, tel, shoptype])

        folium.Marker(
            location = [lat, lng],
            fill = True
        ).add_to(starmap)

```

```

# 각 지점에서 '구'명 추출

gu_names = []

for addr in df['주소']:
    gu = addr.split()[1]
    gu_names.append(gu)

print(gu_names)
print(len(gu_names))

# 기존의 데이터프레임에 열을 하나 새로만들어서 추가한다.
df['시군구명'] = gu_names
df

# 구별 스타벅스 매장 수 계산

starbucks_count = df.pivot_table(index='시군구명',
                                   values='매장명',
                                   aggfunc='count')

```

```
starbucks_count
```

```
seoul_sgg = pd.read_excel('seoul_sgg_stat.xlsx', thousands=',')  
seoul_sgg  
  
seoul_sgg = pd.merge(seoul_sgg, starbucks_count, how='left', on='시군구명')  
seoul_sgg
```