네이버 영화 내용 가져오기

In [1]:

```
from bs4 import BeautifulSoup
from urllib.request import urlopen
```

In [2]:

```
url = "https://movie.naver.com/movie/running/current.nhn"
page = urlopen(url)
soup = BeautifulSoup(page, 'lxml')
```

상영작/예정작 제목만 뽑기

In [3]:

```
soup_ul_li = soup.find("ul", class_="lst_detail_t1").find_all("li")
len(soup_ul_li)
```

Out[3]:

93

제목 가져오기 하나 확인

In [5]:

```
soup_ul_li[3].find("dt", class_="tit").a.text
```

Out[5]:

'그린랜드'

In [6]:

```
all_title = []
for item in soup_ul_li:
    dat = item.find("dt", class_="tit").a.text
    all_title.append(dat)
print(len(all_title), all_title)
```

```
93 ['담보', '언힌지드', '테넷', '그린랜드', '국제수사', '극장판 포켓몬스터 뮤츠의 역습 EVOLUTION', '극장판 미니특공대: 햄버거괴물의 습격', '애프터: 그 후', '어디갔어, 버나뎃', '브레이크 더 사일런스: 더 무비', '검객', '트롤킹', '마샤와 곰: 최고 중에 최고', '죽지않는 인간들의 밤', '밥정', '피원에이치 : 새로운 세계의 시작', '해수의 아이', '마르지엘라', '부활: 그 증거', '디바', '트라이얼 오브 더 시카고 7', '뮬란', '아웃포스트', '극장판 엉덩이 탐정: 텐텐마을의 수수께끼', '교실 안의 야크', '강철비2: 정상회담', '예티: 신비한 동물 탐험대', '도망친 여자', '기기괴괴 성형수', '너의 이름은.', '남매의 여름밤', '프란시스 하', '우리가 이별 뒤에 알게 되는 것들', '소년시절의 너', '날씨의 아이', '카일라스 가는 길', '보테로', '다시 만난 날들', '공포분자', '블레이드 러너 2049', '스윈들러', '스파이더맨: 뉴 유니버스', '그대, 고맙소 : 김호중 생애 첫 팬미팅 무비', '제리 맥과이어', '다만 악에서 구하소서', '드라이브', '나를 구하지 마세요', '동아시아반일무장전선', '낙엽귀근', '오! 문희', '테스와 보낸 여름', '라라랜드', '브리짓 존스의 일기', '알제리 전투', '69세', '마티아스와 막심', '제로 다크 서티', '극장판 짱구는 못말려: 신혼여행 허리케인~ 사라진 아빠!', '리스본행 야간열차', '후쿠오카', '에이바', '홀리 모터스', '마음 울적한 날엔', '피아니스트', '보리밭을 흔드는 바람', '하워즈 엔드', '500일의 썸머', '걸작', '구르는 수레바퀴', '다운폴', '더 파티', '디트로이트', '라붐', '반교: 디텐션', '백년의 기억', '베로니카의 이중 생활', '베를린 천사의 시', '비독: 파리의 황제', '비투스', '사랑과 영혼', '사랑하는 시바여 돌아오라', '소년 아메드', '아무르', '영원과 하루', '워터 릴리스', '위크엔드 인 파리', '천국보다 낯선', '치어리딩 클럽', '미스 사이공: 25주년 특별 공연', '분노의 질주: 더 익스트림', '위대한 쇼맨', '타샤 튜더', '포드 V 페라리']
```

평점과 참여 명수 확인 및 예매율 확인

In [10]:

```
## 평점
soup_ul_li[70].find("span", class_="num").text
```

Out[10]:

```
'8.07'
```

In [11]:

```
## 참여명수
soup_ul_li[0].find("em").text
```

Out[11]:

```
'4,196'
```

In [12]:

```
## 예매율
soup_ul_li[3].find("dl", class_="info_exp").span.text
```

Out[12]:

'6.94'

맨 마지막은 예매율이 있는지 확인

In [17]:

```
length = len(soup_ul_li) - 1
print(length)
last_element = soup_ul_li[length].find_all("dl", class_="info_exp")
last_element
```

92

Out[17]:

[]

In [19]:

```
## 개요
soup_ul_li[2].find("span", class_= "link_txt").text
```

Out[19]:

'Wn액션, WrWnWtWtWtWtWtWtWtWtWrWnWtWtWtWtWtWtWtWtWtSFWn'

In [20]:

```
## 감독
dirA = soup_ul_li[0].find_all("dl", class_="info_txt1")[0].find_all("dd")[1].text
dirA = dirA.replace("Wn", "")
dirA
```

Out[20]:

'강대규'

In [21]:

```
## 감독 8번째, 2명
dirA = soup_ul_li[7].find_all("dl", class_="info_txt1")[0].find_all("dd")[1].text
dirA = dirA.translate(  { ord('Wn'):"", ord('Wr'):"", ord('Wt'):""  } )
dirA
```

Out[21]:

'로저 컴블'

In [22]:

```
#  상영시간
soup_ul_li[2].find("dl", class_= "info_txt1").dd
# soup_ul_li[2].find("span", class_= "link_txt")
```

Out[22]:

```
<dd>
<span class="link_txt">
<a href="/movie/sdb/browsing/bmovie.nhn?genre=19">액션</a><!-- N=a:nol.genre,r:1 --
>,

                                                                    <a href="/mo
vie/sdb/browsing/bmovie.nhn?genre=18">SF</a><!-- N=a:nol.genre,r:2 -->
</span>
<span class="split">|</span>
                                                                    150분




                                                                    <span class="split">
|</span>
                                                                    2020.08.26 개봉

                                                          </dd>
```

In [23]:

```
a = soup_ul_li[8].find("dl", class_= "info_txt1").dd.children
al = list(a)
print(al[-1], al[-3])
al[-1]
```

```
                                                                    2020.10.08 개봉


                                                                    109분
```

Out[23]:

```
'WrWnWtWtWtWtWtWtWtWt2020.10.08 개봉WrWnWtWtWtWtWtWtWtWrWnWtWtWtWtWtWt'
```

In [24]:

```
# 제목, 평점, 참여수, 예매율, 개요, 감독, 상영시간, 상영날짜
all_title = []
all_score = []
all_people = []
all_re_rate = []
all_category = []
all_dir = []
all_time = []
all_date = []

for item in soup_ul_li:
    all_title.append(item.find("dt", class_="tit").a.text) # 제목
    all_score.append(item.find("span", class_="num").text) # 평점
    all_people.append(item.find("em").text)  # 참여명수

    ## 예매율
    temp_re_rate = item.find("dl", class_="info_exp")
    if temp_re_rate is not None:
        ticking = temp_re_rate.find('span', class_='num').text
    else:
        ticking = "0"
    all_re_rate.append(ticking)

    # 개요
    tmp_cat = item.find("span", class_= "link_txt").text
    tmp_cat = tmp_cat.replace("₩n", "")
    tmp_cat = tmp_cat.replace("₩t", "")
    tmp_cat = tmp_cat.replace("₩r", "")
    all_category.append(tmp_cat)

    # 감독
    tmp_all_dir = item.find_all("dl", class_="info_txt1")[0].find_all("dd")[1].text
    tmp_all_dir = tmp_all_dir.translate(  { ord('₩n'):"", ord('₩r'):"", ord('₩t'):""} )
    all_dir.append(tmp_all_dir)

    # 상영시간
    tmp_all_dir = list(item.find("dl", class_= "info_txt1").dd.children)
    tmp_time = tmp_all_dir[-3]
    tmp_time = tmp_time.replace("₩n", "")
    tmp_time = tmp_time.replace("₩t", "")
    tmp_time = tmp_time.replace("₩r", "")
    tmp_time = tmp_time.replace("분", "")
    all_time.append(tmp_time)

    # 상영날짜
    tmp_date = tmp_all_dir[-1]
    tmp_date = tmp_date.replace("₩n", "")
    tmp_date = tmp_date.replace("₩t", "")
    tmp_date = tmp_date.replace("₩r", "")
    tmp_date = tmp_date.replace("개봉", "")
    all_date.append(tmp_date)

# 확인
print(len(all_date), all_date)
```

```
93 ['2020.09.29 ', '2020.10.07 ', '2020.08.26 ', '2020.09.29 ', '2020.09.29 ', '202
0.09.30 ', '2020.09.30 ', '2020.10.07 ', '2020.10.08 ', '2020.09.24 ', '2020.09.23
```

```
', '2020.10.07 ', '2020.10.08 ', '2020.09.29 ', '2020.10.07 ', '2020.10.08 ', '2020.
09.30 ', '2020.09.30 ', '2020.10.08 ', '2020.09.23 ', '2020.10.07 ', '2020.09.17 ',
'2020.09.23 ', '2020.09.24 ', '2020.09.30 ', '2020.07.29 ', '2020.10.08 ', '2020.09.
17 ', '2020.09.09 ', '2018.01.04 ', '2020.08.20 ', '2020.09.24 ', '2020.09.30 ', '20
20.07.09 ', '2020.05.21 ', '2020.09.03 ', '2020.09.24 ', '2020.09.24 ', '2020.09.17
', '2017.10.12 ', '2020.10.08 ', '2018.12.12 ', '2020.09.29 ', '2017.02.14 ', '2020.
08.05 ', '2020.09.03 ', '2020.09.10 ', '2020.08.20 ', '2020.09.24 ', '2020.09.02 ',
'2020.09.10 ', '2020.03.25 ', '2001.09.01 ', '2009.10.15 ', '2020.08.20 ', '2020.07.
23 ', '2013.03.07 ', '2020.08.20 ', '2014.06.05 ', '2020.08.27 ', '2020.09.09 ', '20
13.04.04 ', '2020.09.24 ', '2015.06.18 ', '2019.12.18 ', '2020.09.03 ', '2016.06.29
', '2019.07.01 ', '2020.09.23 ', '2014.01.23 ', '2018.12.20 ', '2018.05.31 ', '2013.
10.24 ', '2020.08.13 ', '2020.06.11 ', '2016.06.23 ', '1993.05.15 ', '2020.09.17 ',
'2008.04.09 ', '2017.12.27 ', '1957.01.19 ', '2020.07.30 ', '2012.12.19 ', '2004.11.
19 ', '2020.08.13 ', '2014.05.01 ', '2017.11.16 ', '2020.09.10 ', '2016.11.24 ', '20
20.03.19 ', '2020.05.21 ', '2018.09.13 ', '2019.12.04 ']
```

In [25]:

```
import pandas as pd
```

In [26]:

```
movie_info = {"제목":all_title, "평점":all_score, "참여수":all_people, "예매율":all_re_rate,
              "개요":all_category, "감독":all_dir, "상영시간":all_time, "상영날짜":all_date}
dat = pd.DataFrame(movie_info)
dat.to_excel("movie_naver.xlsx", index=False)
```

이후에 더 해볼만한 실습 내용

- (1) 참여수에 ','를 빼고 넣기
- (2) 출연정보 가져오기