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]:

```
103
```

확인

In [4]:

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

Out[4]:

```
'국제수사'
```

In [5]:

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

```
103 ['담보', '언힌지드', '그린랜드', '국제수사', '테넷', '애프터: 그 후', '극장판 포켓몬스터 뮤츠의 역습 EVOLUTION', '브레이크 더 사일런스: 더 무비', '극장판 미니특공대: 햄버거괴물의 습격', '마르지엘라', '해수의 아이', '디바', '밥정', '검객', '트롤킹', '죽지않는 인간들의 밤', '트라이얼 오브 더 시카고 7', '아웃포스트', '뮬란', '교실 안의 야크', '남매의 여름밤', '도망친 여자', '극장판 엉덩이 탐정: 텐텐마을의 수수께끼', '우리가 이별 뒤에 알게 되는 것들', '프란시스 하', '기기괴괴 성형수', '너의 이름은.', '소년시절의 너', '마음 울적한 날엔', '강철비2: 정상회담', '제리 맥과이어', '공포분자', '블레이드 러너 2049', '다시 만난 날들', '보테로', '나를 구하지 마세요', '날씨의 아이', '69세', '알제리 전투', '스파이더맨: 뉴 유니버스', '카일라스 가는 길', '제로 다크 서티', '피아니스트', '브리짓 존스의 일기', '테스와 보낸 여름', '에이바', '마티아스와 막심', '홀리 모터스', '오! 문희', '경계선', '루스 베이더 긴즈버그 : 나는 반대한다', '고스트 오브 워', '타샤 튜더', '퍼스널 쇼퍼', '후쿠오카', '낙엽귀근', '다만 악에서 구하소서', '드라이브', '동아시아반일무장전선', '반교: 디텐션', '백년의 기억', '보리밭을 흔드는 바람', '비독: 파리의 황제', '소년 아메드', '워터 릴리스', '치어리딩 클럽', '파밍 보이즈', '하워즈 엔드', '500일의 썸머', '걸작', '구르는 수레바퀴', '굿타임', '나의 산티아고', '디트로이트', '라라랜드', '라붐', '리스본행 야간열차', '미스터 스마일', '베로니카의 이중 생활', '사랑과 영혼', '사랑하는 시바여 돌아오라', '샤인', '셰이프 오브 뮤직: 알렉상드르 데스플라', '스프링 브레이커스', '십계', '아이 캔 온리 이매진', '위크엔드인 파리', '이십일세기 소녀', '작은 아씨들', '천국보다 낯선', '타오르는 여인의 초상', '톰보이', '피터와 드래곤', '하얀 리본', '다운폴', '더 파티', '미스 사이공: 25주년 특별 공연', '베를린 천사의 시', '분노의 질주: 홉스&쇼', '비투스', '영원과 하루', '위대한 쇼맨', '포드 V 페라리']
```

In [6]:

```
## 평점
soup_ul_li[0].find("span", class_="num").text

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

Out[6]:

```
'3,322'
```

In [7]:

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

Out[7]:

```
'6.35'
```

In [8]:

```
empty_107 = soup_ul_li[107].find_all("dl", class_="info_exp")
empty_107
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-8-9d18f8b7ba4c> in <module>
----> 1 empty_107 = soup_ul_li[107].find_all("dl", class_="info_exp")
      2 empty_107

IndexError: list index out of range
```

In [ ]:

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

In [ ]:

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

In [ ]:

```
## 감독 8번째, 2명
dirA = soup_ul_li[7].find_all("dl", class_="info_txt1")[0].find_all("dd")[1].text
dirA = dirA.translate(  { ord('\n'):"", ord('\r'):"", ord('\t'):""  } )
dirA
```

In [ ]:

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

In [ ]:

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

In [ ]:

```
# 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)
```

In [ ]:

```
import pandas as pd
```

In [ ]:

```
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)
```

실습 과제

- (1) 참여수에 ','를 빼고 넣기
- (2) 출연정보 가져오기