실제 웹 사이트에 접속해 웹 스크레이핑을 수행하는 이 책의 예제 코드는 해당 웹 사이트가 변경될 경우에는 제대로 동작하지 않을 수 있습니다. 하지만 이 책에서 설명하는 웹 사이트의 데이터를 가져와 가공하는 기본적인 원리를 이해한다면 해당 웹 사이트가 변경되더라도 예제 코드를 수정해 원하는 데이터를 얻어올 수 있을 것입니다.

영화 순위

영화 감상은 많은 사람들이 사랑하는 대표적인 취미입니다. 영화를 보기 전 대부분의 사람들은 관심 있는 영화에 대해 알아보기 위해 영화를 소개하는 웹 사이트를 방문하기도 하고 평점이나 예매율을 참고하기도 합니다. 또한 인기 있는 영화를 보기 위해 영화 순위를 알아보기도 합니다. 네이버 영화에서는 영화와 관련된 다양한 정보와 영화의 순위 정보를 제공합니다. 이번에는 네이버 영화에서 영화 순위별로 영화 제목과 해당 영화의 링크를 추출해 보겠습니다.

먼저 웹 브라우저로 네이버 영화(`https://movie.naver.com`)에 접속한 후 오른쪽에 있는 [영화랭킹]을 클릭하면 그림 14-10처럼 오늘 날짜를 기준으로 영화 랭킹을 볼 수 있습니다. 네이버의 영화 랭킹은 검색어를 바탕으로 선정된 것으로, 영화관의 예매율을 기준으로 선정된 순서와는 차이가 있을 수 있습니다. 웹 브라우저의 주소창을 보면 URL이 `https://movie.naver.com/movie/sdb/rank/rmovie.naver`인 것을 볼 수 있습니다.

그림 14-10 네이버 영화의 영화 랭킹

웹 브라우저에서 날짜를 변경하면 날짜까지 포함한 URL(https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=cnt&tg=0&date=20220910)이 생성되는 것을 볼 수 있습니다. 즉, 오늘 날짜의 영화 순위를 알고 싶다면 URL을 https://movie.naver.com/movie/sdb/rank/rmovie.naver로 지정하고, 특정 날짜의 영화 순위를 알고 싶다면 URL을 https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=cnt&tg=0&date=YYYYMMDD 형식으로 지정하면 됩니다. 여기서 YYYYMMDD는 날짜로 20220910처럼 연도 4자리, 월 2자리, 일 2자리를 의미합니다. 이 URL을 이용하면 특정 날짜의 영화 순위를 가져올 수 있습니다.

위의 URL을 이용해 2022년 9월 10일의 영화 순위가 담긴 HTML 코드를 가져오려면 다음과 같이 수행합니다.

```
In: import requests
    from bs4 import BeautifulSoup

    base_url = 'https://movie.naver.com/movie/sdb/rank/rmovie.naver'
    date = '20220910'
    url = base_url + '?sel=cnt&tg=0&date=' + date # 날짜를 지정해 URL 생성

    html = requests.get(url).text
    soup = BeautifulSoup(html, 'lxml')

    print(url)
    https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=cnt&tg=0&date=20220910
```

```
Out: https://movie.naver.com/movie/sdb/rank/rmovie.naver?sel=cnt&tg=0&date=20220910
```

웹 브라우저에서 영화 제목에 대해 요소 검사를 실시하면 그림 14-12처럼 『육사오(6/45)』 요소가 찾아집니다. 이제 요소 검사 결과를 이용해 웹 사이트의 순위를 가져오도록 'BeautifulSoup.select('태그 및 속성')'의 '태그 및 속성'을 선택합니다. 여기서 a 태그를 이용하면 너무 많은 요소가 선택되기 때문에 한 단계 상위 요소인 『 육사오(6/45) 』를 이용합니다. 결과적으로 div 태그와 class 속성이 tit3인 요소를 모두 찾기 위해 select(div.tit3)를 이용해 영화 제목을 가져올 수 있습니다.

그림 14–12 영화 랭킹의 영화 제목에 대해 요소 검사를 수행한 결과

다음은 앞의 요소 검사 분석 결과를 적용한 코드입니다.

```
In: movies = soup.select('div.tit3')
    movies[0:5] # 전체 중 일부만 출력
```

```
Out: [<div class="tit3">
      <a href="/movie/bi/mi/basic.naver?code=204640" title="육사오(6/45)">육사오(6/45)</a>
      </div>,
      <div class="tit3">
      <a href="/movie/bi/mi/basic.naver?code=201641" title="공조2: 인터내셔날">공조2: 인터내셔날</a>
      </div>,
      <div class="tit3">
      <a href="/movie/bi/mi/basic.naver?code=195758" title="헌트">헌트</a>
      </div>,
      <div class="tit3">
      <a href="/movie/bi/mi/basic.naver?code=194196" title="한산: 용의 출현">한산: 용의 출현</a>
      </div>,
```

```
<div class="tit3">
<a href="/movie/bi/mi/basic.naver?code=214552" title="놉">놉</a>
</div>]
```

출력 결과를 살펴보면 ~ 가 포함된 요소를 잘 가져온 것을 볼 수 있습니다. 이 요소는 영화 제목과 관련된 a 태그의 요소를 포함합니다. 여기서 a 태그 안에 있는 영화 제목 관련 정보만 가져오려면 리스트의 각 항목에 대해 find('a')를 수행합니다. 다음은 movies 변수에서 첫 번째 항목에 대해 find('a')를 수행하는 예입니다.

```
In: movies[0].find('a')
```

```
Out: <a href="/movie/bi/mi/basic.naver?code=204640" title="육사오(6/45)">육사오(6/45)</a>
```

출력 결과를 보면 a 태그 안에는 영화 URL 정보와 제목 정보가 있습니다. 태그 안에 있는 속성의 내용을 가져오려면 요소반환결과['속성'] 혹은 요소반환결과.get('속성')을 이용합니다. 다음은 a 태그가 담긴 요소를 추출한 후 title과 href 속성을 추출하는 예입니다.

```
In: ranking_title = movies[0].find('a')['title'] # a 태그의 요소를 추출한 후 title 속성을 추출
    ranking_href = movies[0].find('a')['href']   # a 태그의 요소를 추출한 후 href 속성을 추출

    [ranking_title, ranking_href]
```

```
Out: ['육사오(6/45)', '/movie/bi/mi/basic.naver?code=204640']
```

모든 영화에 대해 제목과 링크를 추출하는 코드를 작성하면 다음과 같습니다. 추출한 결과는 pandas의 DataFrame 데이터 형식으로 변환했습니다.

```
In: import pandas as pd

    title_hrefs = [] # 빈 리스트 생성
    base_url = 'https://movie.naver.com'
    for movie in movies:
        ranking_title = movie.find('a')['title'] # a 태그의 요소를 추출한 후 title 속성을 추출
        ranking_href = movie.find('a')['href']   # a 태그의 요소를 추출한 후 href 속성을 추출

        title_hrefs.append([ranking_title, base_url + ranking_href]) # 리스트의 항목 추가
```

```
# 리스트를 이용해 DataFrame 데이터 생성(영화 순위를 index로 지정)
ranking = range(1, len(movies)+1) # 영화 순위 생성
df_movie = pd.DataFrame(title_hrefs, index=ranking, columns=['영화 제목', '링크'])

# head()를 이용해 일부만 출력 (링크 열 너비 지정)
df_movie.head(10).style.set_properties(subset=['링크'], **{'width': '400px'})

# df_movie # 전체 출력
```

Out:

	영화 제목	링크
1	육사오(6/45)	https://movie.naver.com/movie/bi/mi/basic.naver?code=204640
2	공조2: 인터내셔날	https://movie.naver.com/movie/bi/mi/basic.naver?code=201641
3	헌트	https://movie.naver.com/movie/bi/mi/basic.naver?code=195758
4	한산: 용의 출현	https://movie.naver.com/movie/bi/mi/basic.naver?code=194196
5	놉	https://movie.naver.com/movie/bi/mi/basic.naver?code=214552
6	비상선언	https://movie.naver.com/movie/bi/mi/basic.naver?code=184519
7	탑건: 매버릭	https://movie.naver.com/movie/bi/mi/basic.naver?code=81888
8	리미트	https://movie.naver.com/movie/bi/mi/basic.naver?code=193324
9	블랙폰	https://movie.naver.com/movie/bi/mi/basic.naver?code=202465
10	불릿 트레인	https://movie.naver.com/movie/bi/mi/basic.naver?code=218056

출력 결과를 보면 영화 제목과 링크를 잘 가져온 것을 볼 수 있습니다. 참고로 위의 코드에서는 DataFrame 데이터(df_movie)를 출력할 때 링크 열에 있는 URL이 일부만 표시되지 않도록 DataFrame_data.style.set_properties()를 이용해 링크 열의 너비(width)를 400px로 지정했습니다. 여기서 px는 픽셀(pixel) 단위를 의미합니다.

주간 음악 순위

이번에는 파이썬을 이용해 음악 서비스를 제공하는 벅스(Bugs)에서 음악 순위 정보(곡명과 아티스트)를 가져오는 코드를 작성해 보겠습니다. 벅스의 벅스차트에서는 실시간, 일간, 주간 음악 순위를 제공하는데, 실시간 음악 순위는 계속 바뀌므로 이를 이용해 코딩하는 방법을 설명하면 이 책을 쓸 때의 음악 순위와 여러분이 코드를 실행할 때의 음악 순위가 달라서 혼동될 것입니다. 일간이나 주간은 날짜를 지정할 수 있는데 여기서는 주간 음악 순위를 이용해 데이터를 가져오는 예를 살펴보겠습니다.