PYTHON 프로그래밍 실습

Web Scraping

2019-01-11

용어

- Web Scraping (웹 스크래핑)

  정해진 형태의 웹 페이지에서 원하는 정보를 추출하는 작업

- Web Crawling (웹 크롤링)
  - 자동화 봇(bot)인 웹 크롤러(web crawler)가 정해진 규칙에 따라 복수 개의 웹 페이지들을 수집하는 작업

  예) 검색 엔진은 데이트의 최신 상태 유지를 위해 웹 크롤링을 함

- 스크래핑(Scraping) 방법

  (1) 원하는 웹 페이지에 request를 보내어, 결과 html을 받는다.

  (2) 받은 html을 파싱한다.

  (3) 필요한 정보만 추출한다.

- 필요한 모듈

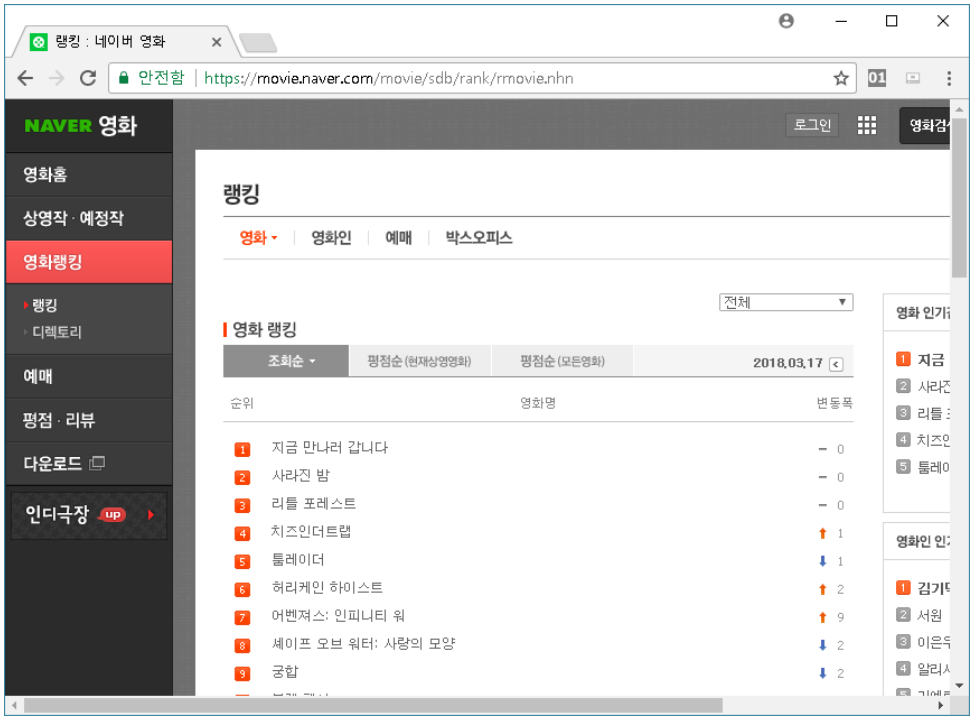
  - http request/response를 다루는 모듈

  - html 파싱하는 모듈

실습1 (영화랭킹)

- 네이버 영화랭킹 읽어오기

https://movie.naver.com/movie/sdb/rank/rmovie.nhn



1위 지금 만나러 갑니다
2위 사라진 밤
3위 리틀 포레스트
4위 툴레이더
5위 치즈인더트랩
6위 셰이프 오브 워터: 사랑의 모양
7위 궁합
8위 허리케인 하이스트
9위 블랙 팬서
10위 월요일이 사라졌다
11위 플로리다 프로젝트
12위 퍼시픽 림: 업라이징
13위 골든슬럼버
14위 로건 럭키
15위 쓰리 빌보드

실습1 (영화랭킹)

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

url = 'http://movie.naver.com/movie/sdb/rank/rmovie.nhn'

page = urlopen(url)
soup = BeautifulSoup(page, "html.parser")

result = soup.find_all("div", attrs={"class":"tit3"})

i = 1
for movie in result:
    print("%2d위" % i , end=' ')
    print(movie.get_text().strip())
    i += 1
```

웹 페이지 읽어오기

- import urllib (내장 함수 사용)

```
from urllib.request import urlopen

url = 'http://movie.naver.com/movie/sdb/rank/rmovie.nhn'

page = urlopen(url)

#print(page.read())
```

BeautifulSoup

- 많이 쓰이는 파이썬용 파서로 html, xml을 파싱할 때 주로 사용

- 파싱 (parsing)
  - 가공되지 않은 데이터에서 원하는 특정한 문자열을 추출하여 의미 있는 데이터로 만드는 과정

- BeautifulSoup 모듈 설치

```
pip install beautifulsoup4
```

BeautifulSoup

- 예제

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

url = 'http://movie.naver.com/movie/sdb/rank/rmovie.nhn'

page = urlopen(url)
soup = BeautifulSoup(page, "html.parser")
                                    # page의 내용을 html.parser로 전달하여 html로 해석
#print(soup)
```

HTML (1)

```
page = '''
<html>
<head>
    <title>제목</title>
</head>
<body>
    <h1>제목1</h1>
        <p>내용1</p>
    <h1>제목2</h1>
        <p>내용2</p>
</body>
</html>
'''
soup = BeautifulSoup(page,"html.parser")
```

- Hypertext markup language
- 웹 페이지를 작성하는데 사용
- 태그로 HTML 요소를 만든다
- 일반적인 태그 구조

<태그명 속성명1="속성값1" 속성명2="속성값2">
텍스트</태그명>

- 실습 (soup.태그명)

>>> soup.title

>>> soup.title.text

>>> soup.title.get_text()

find(), find_all() (1)

- HTML 페이지에서 원하는 Tag를 다양한 속성에 따라 쉽게 필터링

- **find**('태그명', {'속성명':'값'...})
  - 해당 조건에 맞는 하나의 태그 반환, 여러 개인 경우 첫 번째 태그 반환

```
>>> soup.find('h1')
```

- **find_all**('태그명', {'속성명':'값'...})
  - 해당 조건에 맞는 태그 모두 반환
  - 예) soup.find_all('h1')

- get_text(): 태그와 태그 사이의 값 반환

```
In [5]: soup.find('h1')
Out[5]: <h1>제목1</h1>

In [7]: titleList = soup.find_all('h1')

In [8]: for title in titleList:
            print(title.get_text())
제목1
제목2
```

HTML (2)

```
html_doc = """
<html><head><title>The Dormouse's story</title></head>
<body>
<p class="title"><b>The Dormouse's story</b></p>

<p class="story">Once upon a time there were three little sisters; and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>

<p class="story">...</p>
</body>
</html>
"""
```

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

find(), find_all() (2)

- 태그와 속성 이용

  예) 속성 id가 link2값을 가진 <a>태그 모두 반환

```
<p class="story">Once upon a time there were three little sisters; and their names were
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>,
<a href="http://example.com/lacie" class="sister" id="link2">Lacie</a> and
<a href="http://example.com/tillie" class="sister" id="link3">Tillie</a>;
and they lived at the bottom of a well.</p>
```

```
soup.find_all("a", id="link2")
```

```
[<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>]
```

```
soup.find_all("a", {"id":{"link2"}})
```

```
[<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>]
```

```
soup.find_all("a", attrs={"id":"link2"})
```

```
[<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>]
```

- 예) 속성 id가 link2 또는 link3의 값을 가진 <a>태그 모두 반환

```
soup.find_all("a", id={"link2", "link3"})
```

```
[<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,
 <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

```
soup.find_all("a", {"id":{"link2", "link3"}})
```

```
[<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,
 <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

```
soup.find_all("a", attrs={"id":{"link2", "link3"}})
```

```
[<a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>,
 <a class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

find(), find_all() (3)

- 속성에 접근하기
  - <a> 태그에 있는 href 속성에 있는 url 추출

```
<a href="http://example.com/elsie" class="sister" id="link1">Elsie</a>
```

```
aList = soup.find_all('a')
print(aList)
```

```
[<a class="sister" href="http://example.com/elsie" id="link1">Elsie</a>, <
a class="sister" href="http://example.com/lacie" id="link2">Lacie</a>, <a
class="sister" href="http://example.com/tillie" id="link3">Tillie</a>]
```

```
for aTag in aList:
    print(aTag.get('href'))
```

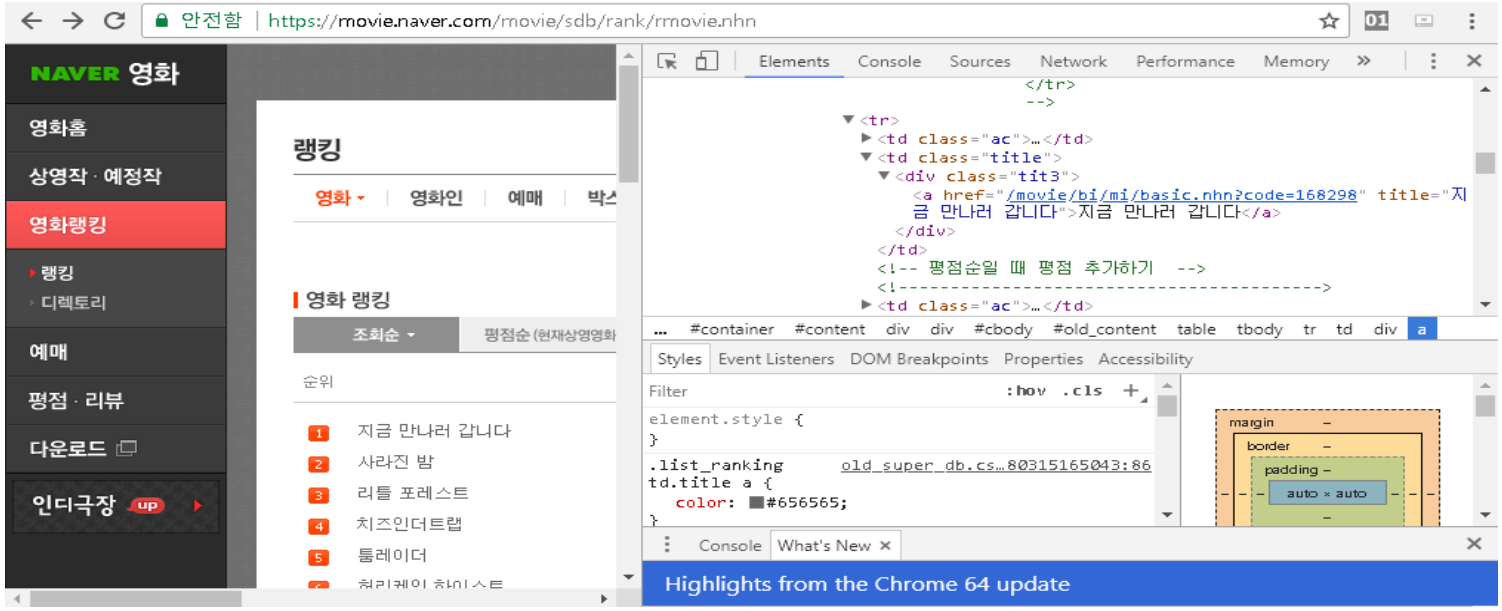
```
http://example.com/elsie
http://example.com/lacie
http://example.com/tillie
```

실습1 (영화랭킹)

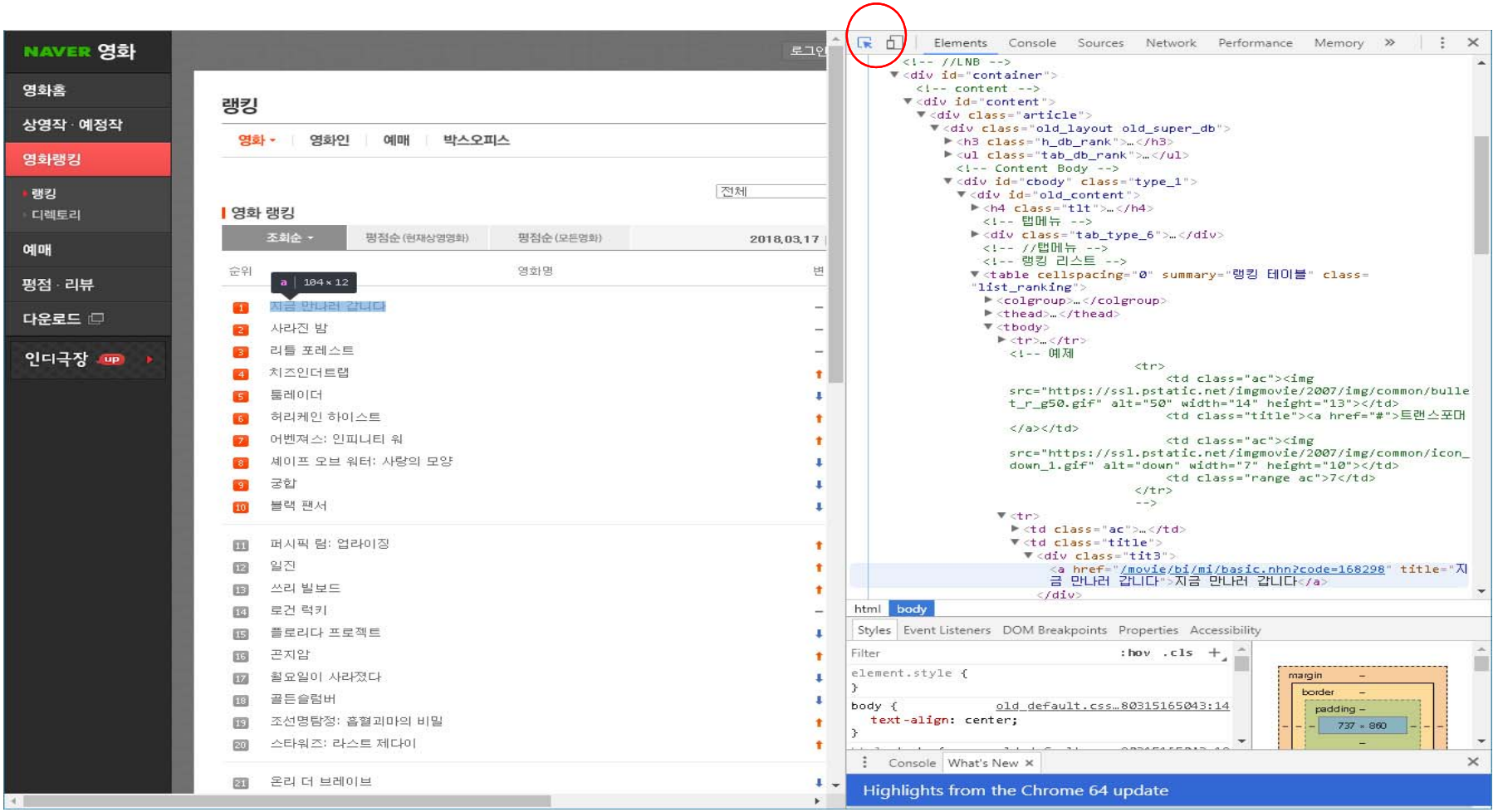
- 웹 사이트 구조 파악하기
  - Chrome Developer Tool 이용

    (1) https://movie.naver.com/movie/sdb/rank/rmovie.nhn 이동

    (2) 영화제목을 선택 후, 마우스 오른쪽 버튼 눌러서 <검사> 선택



실습1 (영화랭킹)


NAVER 영화
영화홈
상영작 · 예정작
영화랭킹
랭킹
디렉토리
예매
평점 · 리뷰
다운로드
인디극장 up
랭킹
영화
영화인
예매
박스오피스
전체
영화 랭킹
조회순
평점순 (현재상영영화)
평점순 (모든영화)
2018.03.17
순위
영화명
a | 104 × 12
1 지금 만나러 갑니다
2 사라진 밤
3 리틀 포레스트
4 치즈인더트랩
5 툼레이더
6 허리케인 하이스트
7 어벤져스: 인피니티 워
8 셰이프 오브 워터: 사랑의 모양
9 궁합
10 블랙 팬서
11 퍼시픽 림: 업라이징
12 일진
13 쓰리 빌보드
14 로건 럭키
15 플로리다 프로젝트
16 곤지암
17 철요일이 사라졌다
18 골든슬럼버
19 조선명탐정: 흡혈괴마의 비밀
20 스타워즈: 라스트 제다이
21 온리 더 브레이브
Elements
Console
Sources
Network
Performance
Memory
<!-- //LNB -->

<!-- content -->

<h3 class="h_db_rank">…</h3>
<ul class="tab_db_rank">…
<!-- Content Body -->

<h4 class="tlt">…</h4>
<!-- 탭메뉴 -->
…
<!-- //탭메뉴 -->
<!-- 랭킹 리스트 -->
<table cellspacing="0" summary="랭킹 테이블" class="list_ranking">
<colgroup>…</colgroup>
<thead>…</thead>
<tbody>
<tr>…</tr>
<!-- 예제
<tr>
<td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/common/bullet_r_g50.gif" alt="50" width="14" height="13"></td>
<td class="title">트랜스포머</td>
<td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/common/icon_down_1.gif" alt="down" width="7" height="10"></td>
<td class="range ac">7</td>
</tr>
-->
<tr>
<td class="ac">…</td>
<td class="title">

지금 만나러 갑니다

html
body
Styles
Event Listeners
DOM Breakpoints
Properties
Accessibility
Filter
:hov .cls
element.style {
}
body {
old_default.css…80315165043:14
text-align: center;
}
margin
border
padding
737 × 860
Console
What's New
Highlights from the Chrome 64 update

실습1 (영화랭킹)

```
▼ <tr>
  ▶ <td class="ac">…</td>
  ▼ <td class="title">
    ▼ <div class="tit3">
        <a href="/movie/bi/mi/basic.nhn?code=168298"
        title="지금 만나러 갑니다">지금 만나러 갑니다</a>
      </div>
    </td>
    <!-- 평점순일 때 평점 추가하기  -->
    <!----------------------------------------->
  ▶ <td class="ac">…</td>
    <td class="range ac">0</td>
  </tr>
▼ <tr>
  ▶ <td class="ac">…</td>
  ▼ <td class="title">
    ▼ <div class="tit3">
        <a href="/movie/bi/mi/basic.nhn?code=165026"
        title="사라진 밤">사라진 밤</a>
      </div>
    </td>
    <!-- 평점순일 때 평점 추가하기  -->
    <!----------------------------------------->
  ▶ <td class="ac">…</td>
    <td class="range ac">0</td>
  </tr>
```

```
… 생략 …

soup = BeautifulSoup(page, "html.parser")


# 영화 제목 추출

result = soup.find_all("div", attrs={"class":"tit3"})

#읽어온 html 소스에서 div 태그 안에 있는 class 가 tit3인 모든 정보를 가져옴

i = 1

for movie in result:
    print("%2d위" % i , end=' ')
    print(movie.get_text().strip())
    i += 1
```

파일 다운로드

- urllib.request.**urlretrieve**()

- 실습 (영화 포스터 다운로드)

(1) 왼쪽의 이미지를 클릭하면 이미지가 포함된 html 문서가 열린다
이미지 소스 부분의 url을 추출해 보자.

파일 다운로드

- urllib.request.**urlretrieve**()

- 실습 – 영화 포스터 다운로드

```
from urllib.request import urlretrieve

img_url = "https://movie-phinf.pstatic.net/20181109_245/1541740985337rsPka_JPEG/movie_image.jpg"

urlretrieve(img_url, "./images/movie.jpg")
```

내 컴퓨터에 저장할 파일 경로와 이름

도전과제

- 영화 랭킹 1위~5위 까지 5개의 이미지를 내 컴퓨터에 다운로드 받는 프로그램을 작성하라. (2초 간격(?)으로 다운로드 할 것)
- 수동으로 url을 입력하는 것이 아니라, 추출해서 받도록 하자.

1. 주먹왕 랄프 2: 인터넷 속으로
2. 아쿠아맨
3. 말모이
4. PMC: 더 벙커
5. 언니

로봇 배제 규약

- 무분별한 크롤링을 막고 제어하기 위한 규약

https://kimdoky.github.io/python/2017/06/14/python-robots.html

http://www.robotstxt.org/