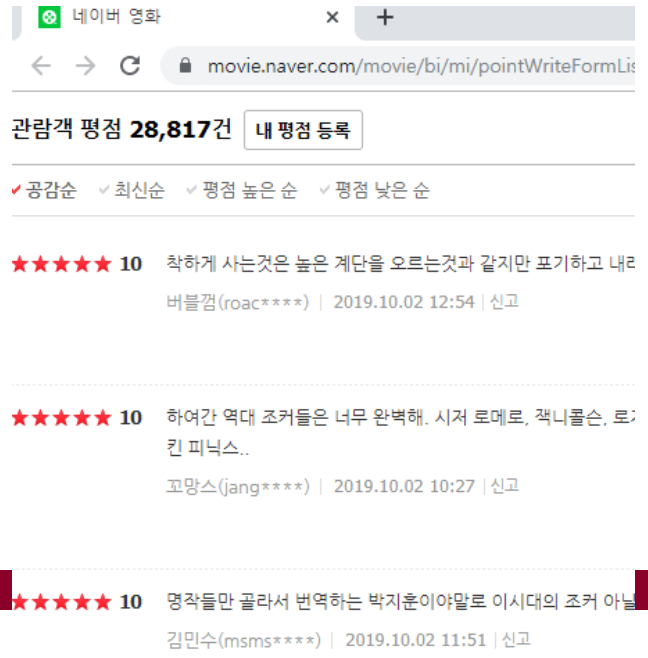
모두를 위한 파이썬 프로그래밍

14주 비정형 데이터의 웹 스크래핑

자연어 처리와 워드 크라우드 과정

Web crawling (beautiful soup)



형태소분석 (koNLPy)

```
[('영화', 'Noun'), ('끝나고도', 'Verb'), ('여운',
'Noun'), ('이', 'Josa'), ('가시', 'Noun'), ('지',
'Josa'), ('않아', 'Verb'), ('멍하게', 'Adjective')
, ('있었습니다', 'Adjective'), ('.',
'Punctuation'), ('화려한', 'Adjective'), ('액션',
'Noun'), ('대신', 'Noun'), ('숨막히는',
'Adjective'), ('설계', 'Noun'), ('가', 'Josa'), ('
있습니다', 'Adjective'), ('그리고', 'Conjunction')
```

시각화 (wordcloud)

들어가기

■ 고파스에서 식당 메뉴를 크롤링 해 보자

- 테이블 형태가 아닌 데이터들은 판다스로 가져올 수 없음

```
# 1. 필요한 라이브러리들을 import 합니다.
import requests
import pandas
# 2. 크롤링 대상 사이트의 html을 문자열 형태로 가져옵니다.
url = 'https://www.koreapas.com/bbs/sik.php?back=1'
# headers 를 수정하여 서버를 속이는 방법
headers = {'user-agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_2) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/86.0.4240.75 Safari/537.36'}
html = requests.get(url, headers=headers)

# 3. html 형식의 문자열들을 pandas를 이용해 리스트 Dataframe(table) 형태로 변환합니다. (parse)
tables = pandas.read_html(html.text)
print(tables)
```

```
[    0   1   2
0 NaN NaN NaN,
0  ¾È¾ÎÇÐ»ç »êÇÐ°ü ÀÚ¿¬°è ÇÐ»ý¾Ä´ç ±³¿ìÈ¸°ü ÇÐ»ýÈ...,
0  KU RESTAURANT  2020-11-30 ~ 2020-12-05  ÀÚ¿¬°è...,     0
0 NaN                                                NaN NaN
1 NaN                                                NaN NaN
2 NaN  °íÆÄ¾° ¾Ò°³ | ÀÌ¿ë¾à°ü | °³ÀÎÁ¤º¸Ãë±Þ³æÄ§ | ÀÌ... NaN]
```

학습내용

Beautiful soup을 이용한 비정형 데이터의 웹스크래핑/크롤링

- 웹 페이지의 비정형 데이터를 스크래핑하기 위한 패키지

필요한 패키지

urllib

- ✓ URL를 다루는 모듈로, 파이썬에 내장되어 있는 기본 모듈
- ✓ 최신 버전: urllib5

Beautiful Soup

- ✓ HTML과 XML 형식의 데이터를 보다 쉽게 parsing하고 다루는 모듈
- ✓ 최신 버전: bs4

웹 스크래핑 과정

웹사이트에 접근하여 HTML 가져오기

```
import urllib.request
html = urllib.request.urlopen('http://www.naver.com')
print(html.read())
```

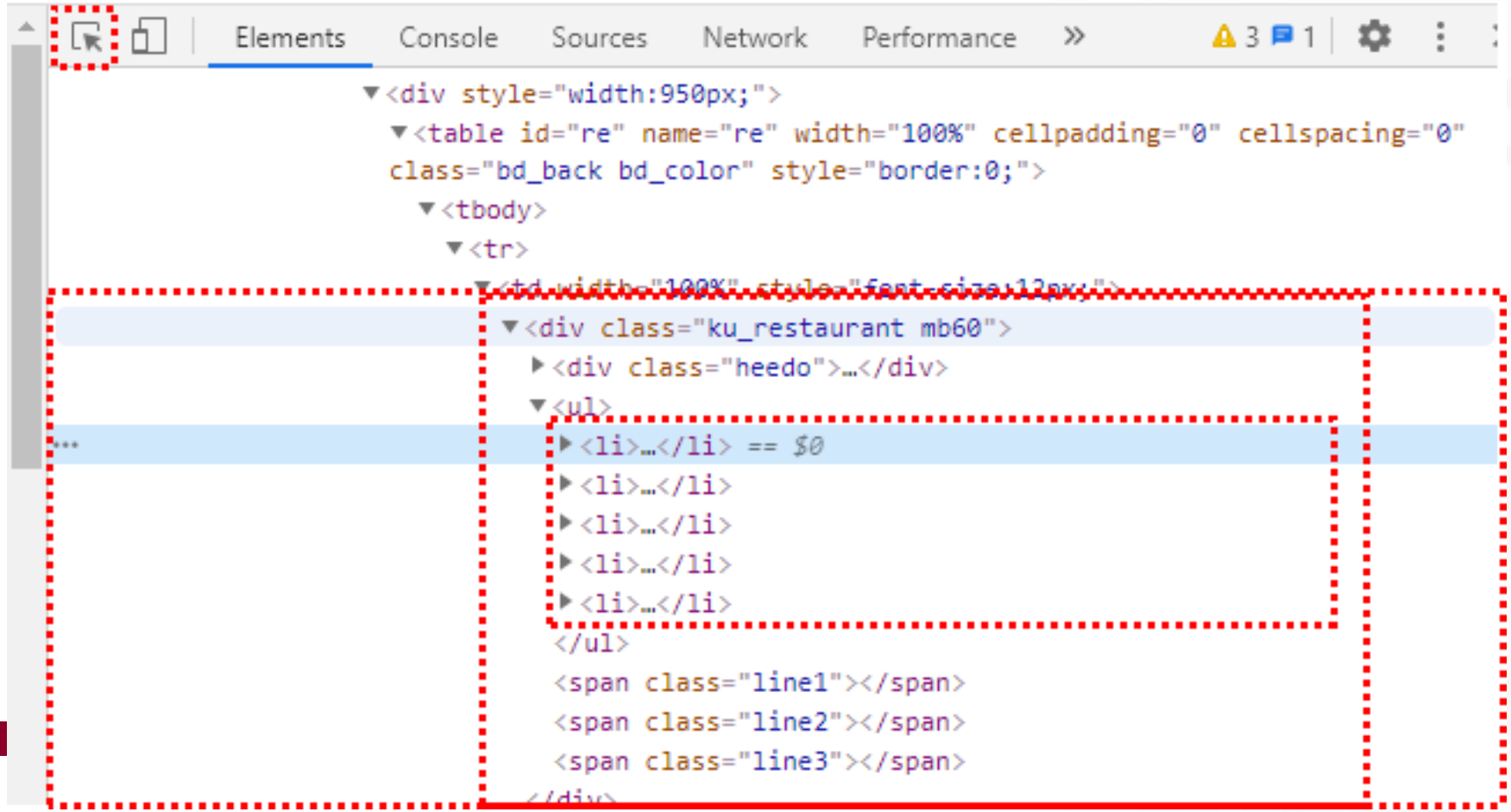
HTML 에서 원하는 부분만 가져오기

1. Beautiful Soup (bs4) 설치
   from bs4 **import** BeautifulSoup
2. html에서 원하는 부분 선택

   - 크롬 개발자 도구를 활용하여 원하는 요소를 추출

식당 메뉴 스크래핑

- **개발자 도구를 열어 원하는 부분의 태그를 확인 (웹브라우즈에서 F12)**
  - ✓ **요소 선택 단추 > 해당 영역 선택 > 클래스 정보 확인**



식당 메뉴 스크래핑

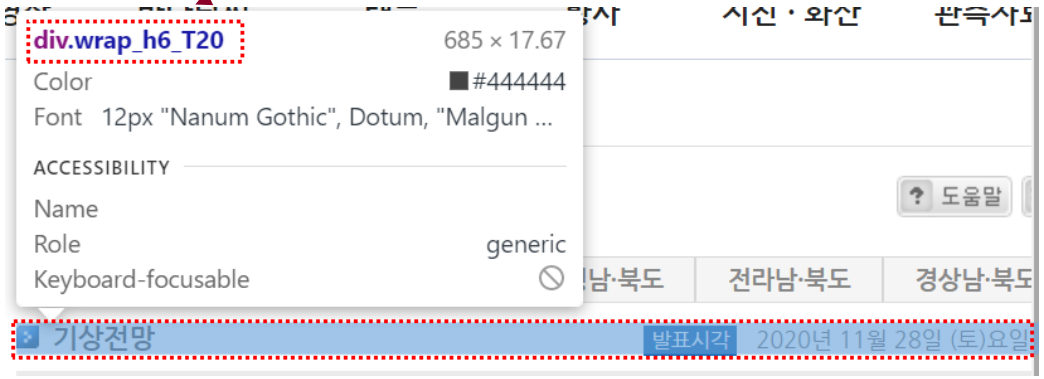
- **웹사이트에 접근하여 BeautifulSoup로 HTML를 parsing하고 원하는 데이터를 추출**

```
# 웹사이트에 접근하여 HTML 가져오기
from urllib.request import urlopen
from bs4 import BeautifulSoup

html = urlopen('https://www.koreapas.com/bbs/sik.php?back=1')
bs = BeautifulSoup(html, "html.parser")
weekly_menu = bs.find('div', {"class": "ku_restaurant mb60"})
print(weekly_menu)
print(weekly_menu.text)

# 파일로 저장하기
file = open('ku_restaurant.txt', 'w', encoding='utf-8')
file.write(weekly_menu.text)
file.close()
```

기상청 중기예보 스크래핑

■ 추출할 정보의 클래스 확인

✓ 해당 위치의 클래스 정보 확인하기



기상청 중기예보 스크래핑

■ 기상청 중기예보 스크래핑

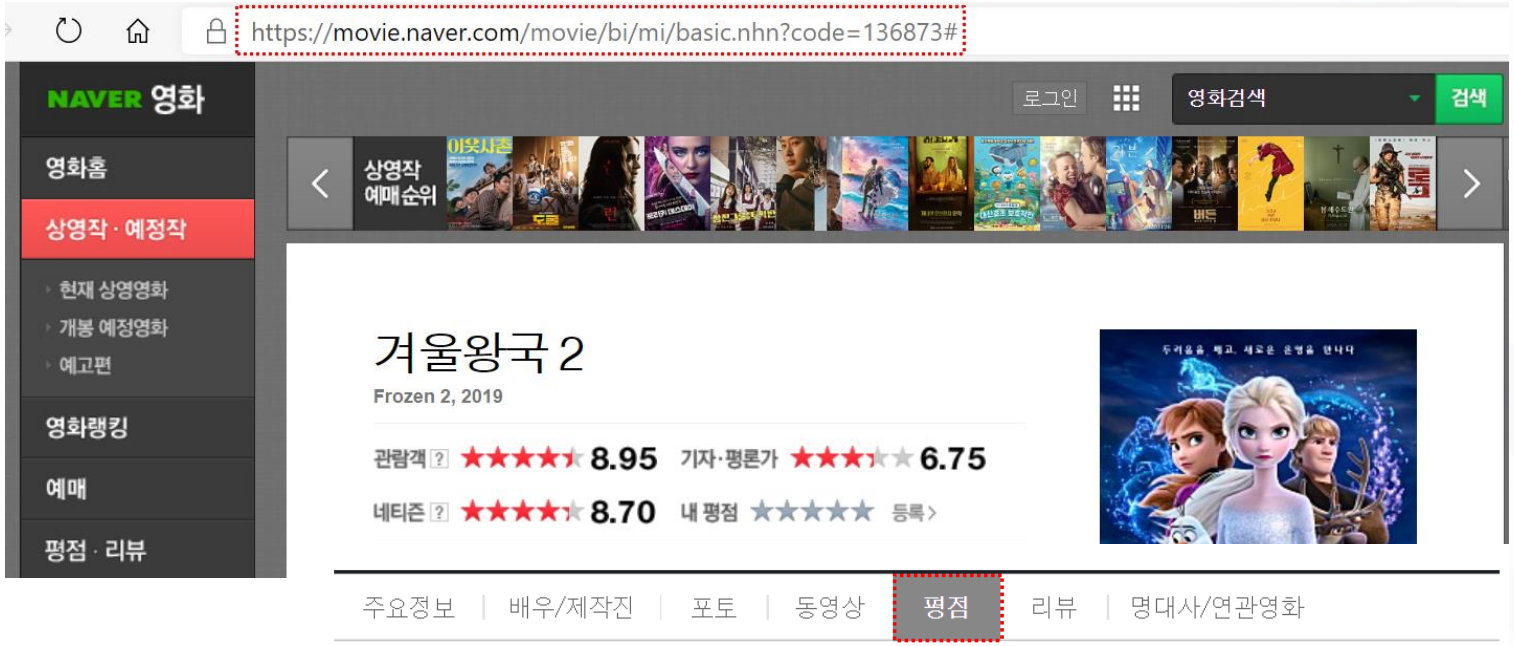
- ✓ 두 영역을 2번에 걸쳐 스크래핑하고 저장

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

html = urlopen('http://www.weather.go.kr/weather/forecast/mid-term_02.jsp')
bs = BeautifulSoup(html, "html.parser")
contents1 = bs.find('div', {"class": "wrap_h6_T20"})
                        # div.wrap_h6_T20 클래스를 찾아서 text만 가져옴
contents2 = bs.find('div', {"class": "bx_midterm"})
print(contents1.text)
print(contents2.text)
```

영화 감상평 수집하기(crawling)

■ 웹 사이트 분석

✓ 네이버 영화 접속 > 평점



■ 웹 사이트 댓글 데이터베이스 주소 확인

https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=136873&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&page=1

영화 감상평 수집하기(crawling)

■ 웹 사이트 html 소스 가져오기

- ✓ 댓글의 내용은 표처럼 정형화되어 있지 않기 때문에 beautifulsoup모듈 사용
- ✓ 웹 크롤링 모듈 2개 import

```
from urllib.request import urlopen # 웹서버에 접근 모듈
from bs4 import BeautifulSoup # 웹페이지 내용구조 분석 모듈
```

- ✓ 댓글 페이지 가져오기 테스트(1페이지만)

```
url='https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=136873&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&page=1'
html=urlopen(url)
# 맛나는수프를 이용하여 댓글 페이지를 utf-8형식으로 html 소스가져오기
html_source = BeautifulSoup(html,'html.parser',from_encoding='utf-8')
```

영화 감상평 수집하기(crawling)

웹페이지 소스보기와 동일한 형태로 추출되는 것 확인하기

```
print(html_source)
```

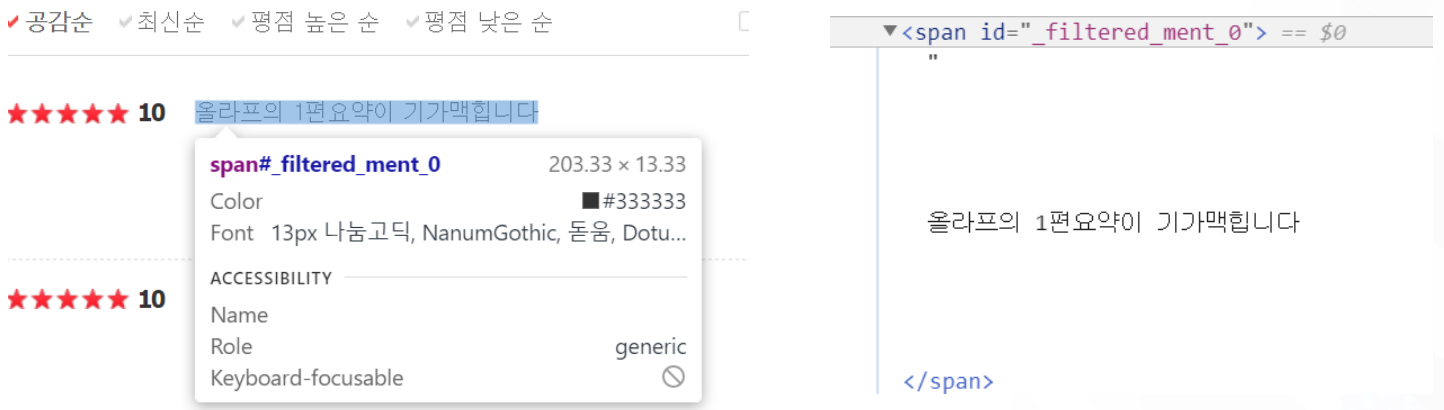
```
<!DOCTYPE html>
<html lang="ko" class="os_windows chrome pc version_78_0_3904_108">
<head>
<meta charset="utf-8">
<meta property="og:site_name" content="Daum 영화">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
 <title>겨울왕국 2 | 다음영화</title>



            <meta property="og:title" content="겨울왕국 2">
                <meta property="og:image" content="http://img1.daumcdn.net/thumb/C300x200/?
fname=http%3A%2F%2Ft1.daumcdn.net%2Fmovie%2F7de761c5bb66457385b4f140c0ce92cd1571277038272">
                <meta property="og:description" content="내 마법의 힘은 어디서 왔을까?나를 부르는 저 목소리는 누구지?어느 날 부턴가
의문의 목소리가 엘사를 부르고, 평화로운 아렌델 왕국을 위협한다. 트롤은 모든 것은 과거에서 시작되었음을 알려주며 엘사의 힘의 비밀과
진실을 찾아 떠나야한다고 조언한다.위험에 빠진 아렌델 왕국을 구해야만 하는 엘사와 안나는 숨겨진 과거의 진실을 ..">
                <meta property="og:type" content="video.movie">

    <meta name="plink" content=""/>
    <meta property="dg:plink" content=""/>
    <meta property="article:mobile_view_url" content="" /> <!-- mobile -->
    <meta property="article:pc_view_url" content="" /> <!-- pc -->
    <meta property="article:mobile_url" content="" /> <!-- mobile -->
    <meta property="article:pc_url" content="" /> <!-- pc -->
        <meta name="article:media_name" content="" />
    <meta property="article:txid" content="" />

    <meta name="article:service_name" content="다음영화" />
    <meta property="article:published_time" content="" />
    <meta property="og:regDate" content="" />

    <!-- 서비스홈 URLs -->
```

영화 감상평 수집하기(crawling)

■ 댓글부분에 해당하는 요소 확인 및 내용 추출하기

✓ 여기에서는 p 클래스 보다 span태그의 id 값을 추출하는 것이 가장 효율적

(태그:span , 속성명:id, 속성값: _filtered_ment_0)




```
html_reviews = html_source.find('span',{'id': '_filtered_ment_0'})
print(html_reviews)
```

영화 감상평 수집하기(crawling)

한 페이지에 대한 리뷰를 출력해 보자

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

url =
'https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=136873&type=after&isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&page=1'
html = urlopen(url)
html_source = BeautifulSoup(html, 'html.parser', from_encoding='utf-8')
# print(html_source)

for i in range(10):
    html_reviews = html_source.find('span',{'id': '_filtered_ment_'+str(i)})
    print(html_reviews)
```

영화 감상평 수집하기(crawling)

■ 불필요한 HTML 태그 제거하기

```
for i in range(10):
    html_reviews = html_source.find('span',{'id': '_filtered_ment_'+str(i)})
    print(html_reviews.text.strip())
```

 # 불필요한 HTML 태그 제거

```
올라프의 1편요약이 기가맥힙니다
크리스토퍼 뮤비에서 좀 흠칫함
미래가 보이지 않을 때는 지금 해야할 일을 해야 해
나는 개인적으로 2편이 더 좋았음. 더 깊어진 스토리에 아름다워진 영상미. 한번 더 볼 의향 있음.
엘사옷 보고 어머니들 긴장하는 영화
애기들 울고 떠들고 하는거 보고 엘사 마법으로 얼릴뻔 했네요
크리스토퍼 혹시 과거에 뮤비찍어본적 있니?왜 이렇게 잘해??...순간 당황했잖아
백마탄 엘사님 Show yourself 장면 진짜 개오집니다 단언컨데 제2의 렛잇고는 into the unknown 아니고 Show yourself 입니다
엘사 물 속에서 말타고 나올 때 대박ㅋㅋㅋㅋ
겨울왕국 역시는 역시였다. OST도 1편만큼이나 중독성있다고 생각함ㅋㅋㅋ1편만큼 존잼임
```

영화 감상평 수집하기(crawling)

- **불필요한 HTML 태그가 제거된 문장을 새로운 리스트 변수에 담기**

```
reviews_list=[]    # 리뷰를 담을 리스트 준비
for i in range(10):
    html_reviews = html_source.find('span',{'id': '_filtered_ment_'+str(i)})
    # print(html_reviews.get_text().strip())
    reviews_list.append(html_reviews.text.strip())          # 순수 리뷰 문장을 리스트에 추가

print(reviews_list)  # 리스트 확인
```

“ 지금까지 한 페이지에 대한 댓글 수집이 완료 되었습니다.”

영화 감상평 수집하기(crawling)

- **원하는 댓글 페이지 만큼 반복해서 처리하기(200페이지)**

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

reviews_list=[]
for j in range(1,11): # crawling ...1 to 10 page
    url = 'https://movie.naver.com/movie/bi/mi/pointWriteFormList.nhn?code=136873&type=after&
isActualPointWriteExecute=false&isMileageSubscriptionAlready=false&isMileageSubscriptionReject=false&page='+str(j)
    html = urlopen(url)
    html_source = BeautifulSoup(html, 'html.parser', from_encoding='utf-8')

    for i in range(10):
        html_reviews = html_source.find('span',{'id': '_filtered_ment_'+str(i)})
        reviews_list.append(html_reviews.text.strip())

print(reviews_list)
```

영화 감상평 수집하기(crawling)

맛나는 수프로 가공된 댓글내용 저장하기

■ **가공된 순수 댓글들을 파일로 저장하기 → 파일은 형태소 분석 단계에서 활용**

```
file = open('opinion.txt', 'w', encoding='utf-8')
for review in reviews_list: # 요소를 1개의 행으로 저장되도록 개행문자 추가
    file.write(review + '\n')  # 개행 문자 추가 --> Enter, 줄바꿈 효과
file.close()
```

```
1  올라프의 1편요약이 기가맥힙니다
2  크리스토퍼 뮤비에서 좀 흠칫함
3  미래가 보이지 않을 때는 지금 해야할 일을 해야 해
4  나는 개인적으로 2편이 더 좋았음. 더 깊어진 스토리에 아름다워진 영상미. 한번 더 볼 의향 있음.
5  엘사옷 보고 어머니들 긴장하는 영화
6  애기들 울고 떠들고 하는거 보고 엘사 마법으로 얼릴뻔 했네요
7  크리스토퍼 혹시 과거에 뮤비찍어본적 있니?왜 이렇게 잘해??...순간 당황했잖아
8  백마탄 엘사님 Show yourself 장면 진짜 개오집니다 단언컨데 제2의 렛잇고는 into the unknowr
9  엘사 물 속에서 말타고 나올 때 대박ㅋㅋㅋㅋ
10 겨울왕국 역시는 역시였다. OST도 1편만큼이나 중독성있다고 생각함ㅋㅋㅋ1편만큼 존잼임
11 스포 때문에 메인 넘버를 into the unknown으로 정한 느낌... 진짜 하이라이트는 show yoursel
```

opinion.txt 파일을 열어 확인해 봅시다

Thank you