2022 SMARCLE DATA ANALYSIS STUDY -

WEB CRAWLING USING BEAUTIFULSOUP4

- 네이버 영화 사이트에서 영화 순위 정보 수집하기

박시현 심재성 이윤서 최규민

크롤링이란?

- 인터넷에 존재하는 다양한 데이터는 다운로드 후 다음 개발 목적과 용도에 맞게 가공하는 과정이 필요. 이런 웹 페이지의 정보들을 추출하는 작업을 크롤링이라고 함.

네이버 영화 사이트에서 영화 순위 정보 수집하기

목표

1. 영화 정보 크롤링
2. 크롤링한 정보 csv파일로 저장

영화 정보

영화 순위
영화 제목
순위 변동 여부
순위 변동값

CODE DESCRIPTION

```
import urllib.request                  #urllib라이브러리의 .request 모듈
from bs4 import BeautifulSoup          #bs4라이브러리에서 BeautifulSoup 모듈
from pandas import DataFrame           #pandas 라이브러리에서 DataFrame 모듈

#각 라이브러리에서 특정 모듈만 사용할 경우,
from (라이브러리 이름) import (모듈)으로 임포트 가능
```

LIBRARY

- urllib: URL, 즉 웹 페이지와 관련된 데이터를 쉽게 다루기 위한 라이브러리
  ex. HTML 소스 코드 읽거나, 데이터 다운로드 등의 작업 수행.
- =>request 모듈: URLs을 열어서 읽어 들이고 웹을 통해 데이터 요청하는 모듈

- Beautifulsoup4: 크롤링을 해주는 대표적인 라이브러리. 이 라이브러리는 파이썬에서 제공하는 HTML 구문분석 라이브러리로서 웹의 태그나 속성의 값을 손쉽게 가져올 수 있도록 지원

- pandas: 엑셀처럼 표(데이터프레임)를 만들어주는 라이브러리. 크롤링한 결과물을 출력할 때 주로 사용됨.

CODE DESCRIPTION

```
url = "http://movie.naver.com/movie/sdb/rank/rmovie.naver"
#url 변수에 가져올 url(네이버 영화) 저장

html = urllib.request.u
#html 변수에 .urlopen 0

soup = BeautifulSoup(html, 'html.parser')
#beautifulsoup 이용하여 네이버영화 페이지에 대한 객체 soup 생성
```

“parsing”
HTML 코드를 파이썬이 이해하는 객체 구조로 변환.

CODE DESCRIPTION

tags = soup.findAll(‘div’, attrs = {‘class’: ‘tit3’})

태그 중 class 속성의 값이 tit3인 항목들 추출

HTML구문 중 하나로,
구분이

```
<td class="title">
    <div class="tit3">
        <a href="/movie/bi/mi/basic.naver?code=211161" title="늑대사냥">늑대사냥</a>
    </div>
</td>
```

print(‘-’ * 30)

CODE DESCRIPTION

print('\n영화 제목만 뽑아내기')

for tag in tags :

print(tag.a.string)

#soup.findall()로 추출

"<a>"
HTML에서, 한 줄 안에 텍스트를 삽입할 때 사용하는 태그
ex) <a>스마클 짱

CODE DESCRIPTION

```
앵커의 href 속성
--------------------------------------------------
https://movie.naver.com/movie/bi/mi/basic.naver?code=211161
--------------------------------------------------
https://movie.naver.com/movie/bi/mi/basic.naver?code=201641
--------------------------------------------------
https://movie.naver.com/movie/bi/mi/basic.naver?code=221031
```

```
print('\n앵커의 href 속성')
url_header = 'https://movie.naver.com
#영화 세부 페이지 url주소의 앞부분을  url_header 변수에 저장

for tag in tags:
        print('-' * 50)
        print(url_header + tag.a['href'])
#url_header의 값 뒤에 상세 주소 페이지를 tag.a['href']를 이용해 추가하여 출력
```

"href"
HTML에서 링크를 삽입할 때 사용하는 태그

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

CODE DESCRIPTION

mytrs = soup.find_all('tr')

#.find_all() 함수로 모든 <tr>태그를 찾아 mytrs에 저장

"<tr>"
table row. HTML에서 행(rows)을 정리할 때 사용하는 태그

CODE DESCRIPTION

for c

#.fin

저장

```
<tr>

        <td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/common/bullet_r_r03.gif" alt="03" width="14" height="13"></td>



    <td class="title">
        <div class "tit3">
            <a href="/movie/bi/mi/basic.naver?code=210852" title="정직한 후보2">정직한 후보2</a>
        </div>
    </td>
    <!-- 평점순일 때 평점 추가하기  -->

    <!--------------------------------------->

    <td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/common/icon_up_1.gif" alt="up" width="7" height="10" class="arrow"></td>
    <td class="range ac">1</td>


</tr>
```

CODE DESCRIPTION

```
if (mytd != None) :                    #mytd의 값이 존재하면
        no += 1                        #no값에 1을 누적하여 더하고
        newno = str(no).zfill(2)       #순위를 초기화

        my
        titl
        #m

        mytd = one_tr.select_one('td:nth-of-type(3)')
        #one_tr에서 세번째 <td>태그 선택
        myimg = mytd.find('img')       #myimg에 mytd의 img태그 저장
```

".zfill(n)"

자리수 n만큼 0을 채워주는 함수

Ex) 해당 코드에서는 str(no)로 정수형 변수인 no의 값이 문자형 1로 변환되었으므로, zfill(2)함수에 의해 결과값이 '1'이 아닌 '01'이 된다.

CODE DESCRIPTION

```
<tr>
    <td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/common/bullet_r_r03.gif" alt="03" width="14" height="13"></td>

    <td class="title">
        <div class="tit3">
            <a href="/movie/bi/mi/basic.naver?code=210852" title="정
        </div>
    </td>
    <!-- 평점순일 때 평점 추가하기 -->

    <!---------------------------------------->

    <td class="ac"><img src="https://ssl.pstatic.net/imgmovie/2007/img/common/icon_up_1.gif" alt="up" width="7" height="10" class="arrow"></td>
    <td class="range ac">1</td>

</tr>
```

mytd = one_tr.select_one('td:nth-of-type(3)')
== 세 번째 <td>

CODE DESCRIPTION

```
if myimg.attrs['alt'] == 'up’:
        up_down = ‘상승’
elif myimg.attrs['alt'] == 'down’:
        up_down = ‘강등’
else :
        up_down = '불변'
```

```
#myimg에서 alt의 값이 up이라면
#up_down값을 ‘상승 ' 으로 저장
#alt의 값이 down이라면
#up_down값을 ‘강등 ' 으로 저장
#이도저도 아니면 ‘불변 '
```

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

CODE DESCRIPTION

```
change = one_tr.find('td', attrs={'class':'range ac'})
#one_tr에 저장된 <td>태그 중 class가 'range ac'인 것을 change에 저장

if change == None :          #range ac값이 없으면 패쓰
        pass
else:
        change = change.string
        #있다면 .string함수 이용해 코드에서 써먹을 수 있도록 객체화
        totallist.append((newno, title, up_down, change))
        #totallist에 (순위, 제목, 순위 변동여부, 순위 변동값) 추가
```

CODE DESCRIPTION

```
mycolumn = ['순위', '제목', '변동', '변동값']
#열의 머리글을 순위, 제목, 변동, 변동값으로 설정

myframe = DataFrame(totallist, columns = mycolumn)
#pandas의 DataFrame 모듈 활용하여 데이터프레임 생성

fil                                                    설정
m                                                      alse)
#.

print(filename, '으로 저장되었습니다.', sep='')
print('finished')
```

"DataFrame(datalist, columns)"

#앞에서 구한 데이터들의 집합 totallist와 표의 열 값으로 mycolumn을 인자로.

#데이터프레임: 여러 유형의 데이터가 표 형태로 되어있는 2차원 형식의 데이터 구조

```
for one_tr in mytrs :      #반복문
#     print(one_tr)
#     print('@' * 30)

    title = ''
    up_down = ''
    mytd = one_tr.find('td', attrs={'class':'title'}) #find 함수를 이용해서 각 행'td'마다 클래스가 title인 것을 mytd에 저장한다.
    if (mytd != None) :  # mytd에 값이 있다면
        no += 1    #숫자를 하나 올리고
        newno = str(no).zfill(2) #순위를 초기화 zfill함수는 ()안에 숫자 자리수 만큼 0 채우기 no가 1이면 01 이 되게끔

        mytag = mytd.find('div', attrs={'class':'tit3'})  # find 함수를 이용해서 mytd에서 div태그에서 클래스가 tit3인것을 mytag에 저장한다.
        title = mytag.a.string # 하위<a> 태그 아래의 글자 영역을 title에 저장

        mytd = one_tr.select_one('td:nth-of-type(3)') # one_tr에서 td:nth-of-type(3) 태그 이름을 선택
        myimg = mytd.find('img')
        if myimg.attrs['alt'] == 'up': #만약 myimg에서 alt값 변동값이 up이라면 up_down 값을 상승으로
            up_down = '상승'
        elif myimg.attrs['alt'] == 'down':#만약 myimg에서 alt값 변동값이 down이라면 up_down 값을 강등으로
            up_down = '강등'
        else :
            up_down = '불변'#만약 myimg에서 alt값 변동값이 up,down둘다 아니면 up_down 값을 불변으로

        change = one_tr.find('td', attrs={'class':'range ac'}) #각 행마다 클래스가 range ac인것을 찾아 change 변수에 저장
        if change == None :
            pass
        else:
            change = change.string #change 옆에 숫자를(문자를) change에 저장 즉 변동값 설정.
            # print(newno + '/' + title + '/' + up_down + '/'  + change)
            totallist.append((newno, title, up_down, change))#totallist에 갱신된 순위 타이틀 순위변동 변동값을 추가
```

THANK YOU!

박시현 심재성 이윤서 최규민