함수 실행순서

fetch\_reviews()

f = open(input\_save\_path(), 'w', *encoding*='utf-8')

result\_list = fectch\_score\_result(URL)

code = split\_url()

함수 설명

def split\_url():

url = input('url을 입력하세요')

code\_str = re.search('code=[0-9]\*', url).group()

code = re.search('[0-9]+', code\_str).group()

return code

url을 정규식에 따라 code에 담을수 있도록 하는 함수입니다.

def fectch\_score\_result(*URL*):

print(URL)

result = {}

result\_list = []

res = urllib.request.Request(URL, *headers*=hdr)

respone = urllib.request.urlopen(res)

html = respone.read()

soup = BeautifulSoup(html, 'html.parser')

score\_result = soup.find('div', *class\_*='score\_result').find('ul')

lis = score\_result.find\_all('li')

for li in lis:

score = li.find('div', *class\_*='star\_score').find('em').get\_text()

spectator = li.find('div', *class\_*='score\_reple').find(

'span').get\_text()

review = li.find('div', *class\_*='score\_reple').find('p').get\_text()

if spectator == '관람객':

review = review[3:]

result['score'] = score

result['review'] = review

result\_list.append({'score': score, 'review': review})

return result\_list

result 리스트를 반환합니다.

def fetch\_reviews():

code = split\_url()

f = open(input\_save\_path(), 'w', *encoding*='utf-8')

page = 1

while True:

count = int(input('게시물의 검색 개수를 입력하세요(10단위): '))

if count % 10 == 0:

break

l\_count = 1

isLoop = True

while isLoop:

URL = url1+code+url2+str(page)

print(URL)

result\_list = fectch\_score\_result(URL)

for r in result\_list:

f.write('=='\*40+'\n')

f.write('영화 평점:'+r['score']+'\n')

f.write('리뷰 내용:'+r['review']+'\n')

f.write('=='\*40+'\n')

l\_count += 1

if l\_count > count:

isLoop = False

break

page += 1

if not isLoop or l\_count == count:

isLoop = False

break

sleepTime = random.randint(4, 10)

time.sleep(sleepTime)

print(str(sleepTime)+'초 기다렸습니다.')

f.close()

리뷰를 가져와서 파일로 저장합니다.

