from selenium import webdriver

from bs4 import BeautifulSoup as bs

import pandas as pd

from time import sleep

from collections import OrderedDict

driver = webdriver.Chrome('/Users/Byung-Jun/PycharmProjects/untitled2/Crawling/chromedriver')

driver.implicitly\_wait(3)

# 로그인 전용 화면

driver.get('https://nid.naver.com/nidlogin.login')

# 아이디와 비밀번호 입력

id = 'Naver Login ID'

pw = 'Naver Login PW'

#네이버 캡챠를 뚫기 위해 다음 코드를 사용한다.

#자동입력방지 문자(캡챠)가 떠오르면 한번 입력해 주자.

driver.execute\_script("document.getElementsByName('id')[0].value=\'" + id + "\'")

driver.execute\_script("document.getElementsByName('pw')[0].value=\'" + pw + "\'")

driver.find\_element\_by\_xpath('//\*[@id="frmNIDLogin"]/fieldset/input').click()

sleep(15)

"""

> NAVER CAFE URL Format : 'https://cafe.naver.com//ArticleList.nhn?search.clubid=까페ID'

1. URL과 카페ID로 원하는 카페로 이동합니다.

2. 여러 메뉴의 게시판 중 하나를 선택합니다. 이때 사용되는 것이 게시판ID인 menuid입니다.

3. 게시판에 올라와 있는 게시글의 리스트를 가져옵니다. 이떄 사용되는 것이 게시글이 있는 게시판ID(menuid)와 페이지 번호(page)입니다.

4. 게시글 하나하나에 대해서 URL을 가져와서 URL LIST를 생성합니다.

5. URL List에서 하나씩 URL을 가져온 후, 해당 URL에 대한 기사 내용을 가져옵니다. 가져온 csv파일에 넣어줍니다.

- 1. Test Cafe URL : https://cafe.naver.com/linuxcare , 카페명 : linuxcare

- 1. Cafe ID (clubid = 10098758)

- 2,3 . 여러 메뉴중 자유게시판의 ID(menuid = 3)

- 3. 자유게시판의 게시글이 올라와 있는 페이지(1부터 1000) 중에 테스트를 위해 1페이지의 정보만 사용함(page = 1)

"""

#Step 1~3

#base\_url = "자신이 크롤링하기를 원하는 카페의 게시판의 URL을 아래와 같이 넣어주세요"

base\_url = 'https://cafe.naver.com/ungsangstory/ArticleList.nhn?search.clubid=24272558'

driver.get(base\_url + '&search.menuLink=31&search.page=1')

#iFrame Switch : iframe 영역의 내부 html code로 진입하기 위해서는 frame switching 명령을 수행해야 합니다.

driver.switch\_to.frame('cafe\_main')

#print(driver.page\_source) #주석을 풀고 원하는 게시판의 page번호에 해당하는 게시글 리스트가 끌려왔는지 확인해 봅시다.

#Step 4

article\_list = driver.find\_elements\_by\_css\_selector('div.inner\_list > a.article') #이 태그 부분 카페마다 다르다던데....

article\_urls = [ i.get\_attribute('href') for i in article\_list ]

#아래 주석을 하나 씩 풀어서 확인해 보세요.

#print(len(quest\_urls))

# for article\_url in article\_urls:

# print (article\_url)

article\_list = []

for article\_url in article\_urls :

try : #게시글이 삭제되었을 경우가 있기 때문에 try-exception

driver.get(article\_url)

#게시글로 들어가면 또 한번 iframe이 나옵니다. 따라서, 해석하기 위해서는 frame switching을 한번더 수행해야 합니다.

driver.switch\_to.frame('cafe\_main')

soup = bs(driver.page\_source, 'html.parser')

#게시글에서 제목 추출

title = soup.select('div.tit-box span.b')[0].get\_text()

#게시글에서 내용 추출

#게시글은 <p>의 연속으로 이뤄져 있기 때문에 이를 합쳐줘야 합니다.

content\_tags = soup.select('#tbody')[0].select('p')

content = ' '.join([ tags.get\_text() for tags in content\_tags ])

#dict형태로 만들어 결과 리스트를 구성합니다.

content\_dict = OrderedDict()

content\_dict["title"] = title

content\_dict["content"] = content

article\_list.append(content\_dict)

except : # chrome alert창 처리해줌

driver.switch\_to\_alert.accpet()

driver.switch\_to\_alert

driver.switch\_to\_alert.accpet()

driver.close()