**웹크롤링**

**0. 파일제어**

**f= open(“파일 명”, “모드”, encoding=”utf8”)**

#기본적인 메소드, 파일을 연다

#모드 w, r, a, w+, rb, wb

#a의 경우 파일의 끝에서부터 작성, w는 파일의 내용을 지우고 처음부터 다시 작성 w+는 a와 같다. wb, rb는 바이트로 읽는다.

#encoding방식을 바꿔서 글자가 깨지는 현상을 해결할 수 있다.

**f.write(“text”)**

#text를 쓴다.

**f.readline()**

#개행문자가 나올 때 까지 읽는다. 여러 번 호출하면 그 다음줄을 차례로 읽는다.

**f.readlines()**

#모든 라인을 읽고 리스트 형태로 반환한다.

**f.seek(0)**

#처음부터 다시 읽는다. ()안의 숫자는 단위에 맞게 적절히 설정, 그 위치부터 읽는다

**f.close()**

**1. 파일을 읽고 쓰기**

**fr = open(‘test.txt’, ‘r’, encoding = ‘utf8’)**

**fw= open(‘text\_copy.txt’, ‘w’, encoding = ‘uft8’)**

**fw.write(fr.read())**

#fw에 fr에서 읽어온 값을 쓴다.

#파일이 사진인 경우 rb, wb로 읽고 쓴다. r과 w는 텍스트 읽기, 쓰기 모드

**with open(‘test.txt’, ‘r’, encoding = ‘utf8’) as fr:**

**with open(‘text\_copy.txt’, ‘w’, encoding = ‘uft8’) as fw:**

#statement가 끝나면 자동으로 close()해준다.

**2. 웹에서 사진 다운받기**

**inport urllib.request as req**

#urllib.request 라이브러리 사용 너무 길어서 별칭으로 req 설정

**rep = req.urlopen(‘URL’)**

#requst에 맞는 response가 생긴다. object로 생김.

**print(rep.getheaders())**

#headers에는 중요한 정보들이 저장되어 있다.

**print(rep.getheader(“Connection”))**

#headers안에서 ()안의 내용에 대해 출력

**rep.status**

#요청이 반환된 상태를 말함. 200의 경우 정상

**f = open(‘url.html’, ‘w’, encoding = ‘uft8’)**

**f.write(rep.read().decode(‘utf8’) )**

#response를 읽어서 f에 저장한다 즉 f에는 url이 담겨서 url.html 파일이 생성됨 그 안에는 url의 정보가 저장

**data = rep.read().decode(‘utf8’)**

#data를 가지고 필요한 정보들만 가져오자.

=>정규식 사용 **import re**

**result = re.findall(‘[0-9a-zA-Z]+,jpg’, data)**

#data안에서 .jpg로 끝나고 0~9, a~z, A~Z 들로 시작하는 데이터를 찾아라. 단순하게 [\w]로 쓸 수 있다. (특수문자는 안되므로 [./-\_\w]로 쓰자)

**for link in result**

**idx = link.rfind(‘/’)**

#link에는 result의 값들이 하나씩 저장, rfind는 오른쪽에서부터 괄호 안의 값이 처음 나오는 index

**with open( link[idx+1:], “wb”) as f:**

#슬라이싱을 통하여 \ 다음부터 나오는 문자열을 파일 명으로 한다.

**pic = req.urlopen(link)**

#link 라는 url의 정보를 pic에 저장

**f.write( pic.read() )**

#data에 들어있는 url의 사진 정보들이 만들어진다.

**3. Beautifulsoup 사용법**

터미널에서 **pip install beautifulsoup4** 입력

#beautifulsoup 설치

**from bs4 import BeautifulSoup**

#모듈 사용

[*https://www.youtube.com/watch?v=DBK9QdX6Yw&list=PLRB\_0OYQAHMOF60j0XP7A1yjYo41q5PRF&index=4*](https://www.youtube.com/watch?v=DBK9QdX6Yw&list=PLRB_0OYQAHMOF60j0XP7A1yjYo41q5PRF&index=4)

에서 파이썬으로 html을 간편하게 작성하는 방법을 소개해줌. 참고

**res = req.urlopen(‘http://~’)**

#res에서 url주소를 연다

**soup = BeautifulSoup(res, ‘html.parser’)**

#res가 담고있는 url을 beautifulSoup로 제어한다. print(soup.title) 하면 url에서 title 태그를 출력

**print(soup.title.text)**

#title태그에서 text만 얻어옴

**print(soup.body.div.p.text)**

#body 안에 div 안에 p태그에서 p는 unique하기에 soup.p로 할 수 있다 text만 얻어옴

#태그가 unique하지 않은 경우 제일 첫 번째 태그를 찾음

**print(soup.find\_all(‘div’))**

#div 태그를 모두 찾는다. 일반적인 경우 div태그는 굉장히 많아서 불편하다

=> **print(soup.find\_all(id = “test1”))**

#id가 text1인 것을 모두 찾아라 id는 속성이므로 “”나 \_를 붙이지 않아도 된다.

**print(soup.find\_all(class\_=”myclass”))**

#class 가 myclass인 것을 모두 찾아라. 단 class는 예약어이므로 \_를 붙인다.

**print(soup.find(‘div’)[‘id’])**

#div 태그에서 속성이 id인 것을 찾아라

**for tag in soup.find\_all(‘a’):**

**print(tag[‘href’])**

#a태그에 걸려있는 모든 링크 주소를 가져온다.

html에 copy selector를 하고

**soup.select(‘붙여넣기’)** => 붙여넣기 하면 #test1 > img 로 되는데 이는 id가 test1인 것의 자식 img태그를 select한다.

#select 복사를 리스트 형태로 붙여넣기 한다.

**soup.select\_one(‘붙여넣기’)**

#하나만 복사

**4. BeautifulSoup 실전**

EX) 한 학교의 식단 정보를 크롤링해서 메신저로 물어보고 대답하거나 정시에 식단 정보를 보내주는 프로그램

**from bs4 import BeautifulSoup**

**import urllib.request as req**

**baseURL = ‘~~’**

#baseURL에 저장 할 때 마지막 date정보는 삭제하고 저장한다

**date = 0701**

#이 URL은 마지막의 숫자에 따라서 예를 들어 0701 이면 7월 1일의 급식 표시

**res = req.urlopen(‘baseURL + date’)**

**meals = soup.find\_all(class\_=’tch-lnc-wrap’)**

#크롤링 할 부분의 태그를 잘 찾아서 unique 한 부분을 find\_all을 이용하여 찾자

**print(meals.ul.text)**

#meals에서 ul태그의 자식들에 대해서 text로 출력하라

참고) **print(repr(meal.ul.text))**

#meal.ul을 개행문자처럼 제어문자를 포함하여 출력한다

개행 문자 없애기

**st = meal.ul.text**

**st = st.replace(‘\r\n’,’’)**

#\r\n을 아무것도 없는(‘’)로 바꾼다.

#금요일에는 석식이 없고 방학 기간에는 중식만 있다

=>

**dic = {}**

**for meal in meals:**

**dic.update({meal.dt.text : meal.ul.text})**

#key 와 value 순으로 update

**print(dic[‘조식’])**

#dic에서 조식만 프린트, 조식이 없는데 조식을 찾는 경우 if문을 통해 없는 식단 정보라고 출력할 수 있다

**5. Selenium**

Javascript가 html 렌더링을 마음대로 해서 Javascript로 만든 사이트의 경우 리퀘스트를 보내서 받은 리스폰스로 구현할 수 없어서 브라우저 자동화 모듈 selenium을 사용한다.

*크롬 => 도움말 => 정보에서 크롬의 버전 확인*

*selenium chrome driver 검색 후 크롬 버전에 맞게 다운로드*

*vscode의 작업 폴더에 압축 해제 후 터미널에서* ***pip install selenium***

**from selenium import webdriver**

**import time**

**chrome = webdriver.Chrome(‘chromedriver.exe’)**

#괄호 안에는 chrome 드라이버 위치에 맞게 작성. 예시에서는 같은 폴더에 크롬이 있어서 경로 생략

**chrome.get(‘https://google.com’)**

#chrome으로 구글을 연다

**time.sleep(3)**

**chrome.quit()**

#3초 뒤에 크롬창을 종료한다

자동 검색 후 로그인 만들기

**from selenium import webdriver**

**import time**

**chrome = webdriver.Chrome(‘chromedriver.exe’)**

#괄호 안에는 chrome 드라이버 위치에 맞게 작성. 예시에서는 같은 폴더에 크롬이 있어서 경로 생략

**chrome.get(‘https://google.com’)**

#chrome으로 구글을 연다

**time.sleep(3)**

#구글의 검색창에 우클릭-검사를 해서 input이 있는 태그를 찾고 copy selector

**input\_box = chrome.find\_element\_by\_css\_selector()**

#find에는 다양한 종류가 있는데 css\_selector로 찾자 괄호 안에는 위에서 복사한 것 붙여넣기

**input\_box.send\_keys(‘충북대학교’)**

#검색창에 충북대학교라는 입력까지만 마친 상태이다. 엔터 키나 검색 버튼을 누르는 것을 코드로 구현해야 한다.

**input\_box.submit()**

#브라우저 자동화로 제출

#OR 엔터키를 구현하려면?

**from selenium.webdriver.common.keys import Keys**

**input\_box.send\_keys(Keys.ENTER)**

#sned\_key를 보내는데 문자열이 아닌 Keys의 ENTER을 보낸다.

#검색 후 검색 화면에서 첫 번째 링크를 들어가보자.

#들어갈 링크를 우클릭-검사해서 a태그를 셀렉터로 copy

**link = chrome.find\_element\_by\_css\_selector()**

#괄호 안에는 위에서 복사한 a태그 붙여넣기 이제 클릭을 구현해야함

**link.click()**

#팝업창이 뜨는 경우 팝업창의 브라우저 컨트롤도 신경써야함

#로그인 버튼 클릭하게 만들기

#로그인 우클릭-검사 태그 복사

**link = chrome.find\_element\_by\_css\_selector()**

**link.click()**

#아이디 부분에 우클릭-검사-복사

**id\_input = chrome.find\_element\_by\_css\_selector()**

#비밀번호 부분에 우클릭-검사-복사

**pass\_input = chrome.find\_element\_by\_css\_selector()**

#제출 버튼에 우클릭-검사-복사

**submit = chrome.find\_element\_by\_css\_selector()**

**id\_input.send\_keys(id\_text)**

**pass\_input.send\_keys(pass\_text)**

**submit.click()**

#아이디 비밀번호 입력 후 submit 누름

#비밀번호가 틀리면 경고창이 뜨는데, 자바스크립트로 경고창이 뜨는 경우 해결해야 한다

**alert = chrome.switch\_to.alert**

**print(alert.text)**

#경고 메세지가 무엇인지 프린트

**alert.accept()**

#경고창 확인

**alert.dismiss()**

#경고창 취소

#팝업창을 제어해야 하는 경우

**main\_window, sub.window = chrome.window\_handles**

#현재 열려있는 윈도우들의 제어권을 언패킹으로 받아준다

**chrome.switch\_to.window(sub\_window)**

#제어권을 팝업으로 이전

**chrome.close()**

#팝업을 닫는다

**time.sleep(2)**

#작업이 꽤 걸리므로 time sleep해준다

#chrome.switch\_to.window(main\_window)

#제어권을 다시 메인으로 돌려준다

#위의 경고창 제어도 윈도우 제어권을 이용하여 아래의 문장으로도 할 수 있다

**chrome.switch\_to.alert.accept()**

chrome.get(‘url’)

#url주소로 이동한다.

source = chrome.page\_source

#url주소의 page source를 가져온다

from bs4 import Beautifulsoup

#Beautifulsoup 사용

soup = Beautifulsoup(source, ‘html.parser’)

#soup에 Beautifulsoup를 이용하여 page source를 가져왔고 이를 3번에서 다룬 예제와 같이 page source를 크롤링한다.

#현재까지 만든 프로그램은 자동으로 크롬을 열어서 충북대학교 웹페이지에 들어가서, 로그인 한 후 원하는 url(예를 들어 식단 정보)으로 이동한 후 자동으로 크롤링 해 온다.

#주의해야 할 점은 윈도우 분할이 일어 날 때(경고창, 팝업) 제어권을 어떻게 넘기는지