Chap 7. 다양한 SNS 정보 수집하기

1. 이번 장에서 배울 내용 소개

여러분들은 어떤 Social Network Service를 사용하고 계신 가요?
우리가 살고 있는 지금 이 시대에는 아주 다양한 Social Network Service 가 존재합니다.
대표적으로 많이 사용하고 있는 소셜 네트워크 미디어로는 인스타그램, 페이스북, 트위터, 네이버 블로그, 다음 블로그, 카카오 스토리, 티스토리, 유투브 등 아주 많죠. 아마도 지금 이 책을 보시는 여러분도 이런 소셜 미디어 중에서 최소 1개 이상에 여러분들의 생각이나 다양한 일상들을 남기고 계시죠?

현대를 살아가는 수많은 사람들이 다양한 소셜 미디어에 다양한 생각이나 의견을 남기고 소셜 미디어로 서로 소통을 하고 있기에 많은 기업이나 연구 주체에서 소셜 미디어의 정보를 수집하여 사업에 활용도 하고 연구에 참고하기도 합니다.

그래서 이런 니즈들이 아주 많기 때문에 이번 챕터에서는 소셜 미디어의 정보를 수집하는 내용을 전해 드리겠습니다.
다양한 소셜 미디어 중에서 이 책에서는 인스타그램(https://www.instagram.com) 사이트에 자동으로 로그인 한 후 특정 키워드로 검색하여 여러 건의 해시 태그를 추출하여 파일에 저장하는 내용을 다루겠습니다.

인스타그램에는 사진을 비롯하여 다양한 정보들이 있는데 다른 사람의 사진을 무단으로 다운로드 받을 경우 불법이 될 수 있습니다.
그래서 이 책에서는 해시 태그만 수집하여 저장하는 방법을 설명하고 있습니다.
하지만 실전에서는 해시 태그 이외에도 리뷰 내용이나 날짜 정보 등 다양한 정보를 추출하는 경우가 아주 많이 있으니 이번 챕터의 내용을 공부한 후에 직접 도전해 보시는 것도 실력 향상에 많은 도움이 될 것 같습니다~^^

2. 전체 코드 미리 보기

(아래 코드는 저자가 제공하는 소스 코드를 사용하세요)

****

```
#Step 1. 필요한 모듈과 라이브러리를 로딩합니다.
from bs4 import BeautifulSoup
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.service import Service
import time
import math
import os
import random
import unicodedata          # 해시태그 수집 중 자음/모음 분리현상 방지용 모듈
import urllib.request
import urllib
import pandas as pd

#Step 2. 사용자에게 필요한 정보들을 입력 받기
print("=" *80)
print("인스타그램 해쉬태그와 이미지 수집하기")
print("=" *80)

v_id = input("1.인스타그램의 ID를 입력하세요: ")
v_passwd = input("2.인스타그램의 비밀번호를 입력하세요: ")
query_txt = input("3.검색할 해쉬태그를 입력하세요(예: 강남맛집): ")
try :
    cnt = int( input('4.수집할 건수는 총 몇 건입니까?(기본값:10): '))
except ValueError :
    cnt = 10
    print('기본값인 10 건으로 수집을 진행합니다.')
page_cnt = math.ceil( cnt / 10)
f_dir=input('5.파일이 저장될 경로만 쓰세요(기본경로 : c:₩₩py_temp₩₩ ) : ')
if f_dir =='' :
    f_dir = "c:₩₩py_temp₩₩"

#Step 3.결과를 저장할 폴더명과 파일명을 설정하고 폴더를 생성하기
n = time.localtime()
s = '%04d-%02d-%02d-%02d-%02d-%02d' % (n.tm_year, n.tm_mon, n.tm_mday, n.tm_hour, n.tm_min, n.tm_sec)
```

```
37
38  sec_name='인스타그램'
39  img_dir = f_dir+s+'-'+query_txt+'-'+sec_name+'₩₩'+'image'
40
41  os.makedirs(img_dir)
42  os.chdir(img_dir)
43
44  fc_name=f_dir+s+'-'+query_txt+'-'+sec_name+'₩₩'+s+'-'+query_txt+'-'+sec_name+'.csv'
45  fx_name=f_dir+s+'-'+query_txt+'-'+sec_name+'₩₩'+s+'-'+query_txt+'-'+sec_name+'.xlsx'
46
47  # Step 4. 인스타그램 접속 후 자동 로그인 하기
48  s_time = time.time( )
49  s = Service("c:/py_temp/chromedriver.exe")
50  driver = webdriver.Chrome(service=s)
51  url = "https://www.instagram.com/"
52  driver.get(url)
53  time.sleep(random.randrange(1,5))
54
55  print("₩n")
56  print("요청하신 데이터를 추출중이오니 잠시만 기다려 주세요~~~~^^")
57  print("₩n")
58
59  #ID와 비번 입력후 로그인하기
60  eid = driver.find_element(By.NAME,'username')
61  for a in v_id :
62      eid.send_keys(a)
63      time.sleep(0.3)
64  epwd = driver.find_element(By.NAME,'password')
65  for b in v_passwd :
66      epwd.send_keys(b)
67      time.sleep(0.5)
68
69  driver.find_element(By.XPATH,'//*[@id="loginForm"]/div/div[3]/button/div').click()
70  time.sleep(5)
71
72  # Step 5. 검색할 키워드 입력하기
73  element = driver.find_element(By.XPATH,'//*[@id="react-root"]/section/nav/div[2]/div/div/div[2]/input')
74  for c in query_txt :
75      element.send_keys(c)
```

```
76          time.sleep(0.2)
77      time.sleep(3)
78      element.send_keys("₩n")
79      element.send_keys("₩n")
80      time.sleep(5)
81
82      # Step 6. 전체 게시물의 원본 URL 추출하기
83      item=[ ]        # 인스타그램 URL 주소 저장할 리스트
84      item2=[ ]       # 중복값을 제거한 최종 URL 주소를 저장할 리스트
85
86      # 자동 스크롤다운 함수
87      def scroll_down(driver):
88          driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")
89          time.sleep(5)
90
91      print('요청하신 데이터를 수집중이니 잠시만 기다려 주세요~^^')
92      print()
93
94      a = 1
95      while (a <= page_cnt):
96          scroll_down(driver)
97
98          html = driver.page_source
99          soup = BeautifulSoup(html, 'html.parser')
100
101         all_a = soup.find('article','KC1QD').find_all('a')
102
103         for i in all_a:
104             url = i['href']
105             item.append(url)
106             item2 = pd.Series(item).drop_duplicates()
107
108             if len(item2) >= cnt :
109                 break
110         a += 1
111         print('요청하신 데이터를 수집중이니 잠시만 더 기다려 주세요~^^')
112
113 # 추출된 URL 사용하여 전체 URL 완성하기
114 full_url=[ ]
```

```
115 url_cnt = 1
116 print('= 수집될 인스타그램 주소는 아래와 같습니다 =========')
117 for x in item2 :
118         url = 'https://www.instagram.com' + x
119         full_url.append(url)
120         print(url_cnt,':',url)
121
122         if url_cnt > cnt:
123             break
124         url_cnt += 1
125 print('=========================================')
126 print()
127
128 #Step 7. 각 페이지별로 이미지와 해쉬태그를 수집하기
129 count = 1          # 추출 데이터 건수 세기
130 no2= [ ]           # 번호 저장
131 url2=[ ]           # 수집완료된 url 저장
132 hash2 = [ ]        # 해쉬 태그 저장
133
134 count = 1
135
136 for c in full_url :
137         print()
138         driver.get(c)
139         time.sleep(random.randrange(3,9))
140
141         html = driver.page_source
142         soup = BeautifulSoup(html, 'html.parser')
143         tags = soup.find('div','EtaWk')
144
145         try :
146             tags_1 = tags.find_all('a')
147         except :
148             continue
149         else :
150             print('%s번째 게시물의 대표 이미지와 해쉬태그를 수집합니다~~~' %count)
151             print('게시물 URL:' , c )
152             no2.append(count)
153             url2.append(c)
```

```
154
155         #해당 페이지의 대표 이미지 수집
156         img_src = soup.find('div','KL4Bh').find('img')['src']
157         urllib.request.urlretrieve(img_src , str(count)+'.jpg')
158         print(img_dir,'아래에 %s번째 이미지 저장 완료===' %count)
159
160         # 해당 페이지의 해시태그 수집
161         # 비트맵 이미지 아이콘을 위한 대체 딕셔너리를 만들기
162         import sys
168         bmp_map = dict.fromkeys(range(0x10000, sys.maxunicode + 1), 0xfffd)
164
165         hash_tags=[]
166         for d in tags_1 :
167             tags = d.get_text()
168             tags_11 = tags.translate(bmp_map)
169             tags_2 = unicodedata.normalize('NFC', tags_11)
170
171             if tags_2[0:1]=='#':
172
173                 hash_tags.append(tags_2)
174
175         print(hash_tags)
176         hash2.append(hash_tags)  # 각 게시물의 해시태그를 리스트 형태로 저장하기
177
178     count += 1
179
180
181 #Step 8. 수집된 해시태그를 csv , xls 형식으로 저장하기
182 # xls , csv로 저장하기 위해 데이터 프레임 생성하기
183 insta = pd.DataFrame( )
184 insta['번호'] = no2
185 insta['URL주소'] = url2
186 insta['해쉬태그'] = pd.Series(hash2)
187
188 # csv 형태로 저장하기
189 insta.to_csv(fc_name,encoding="utf-8-sig",index=False)
190
191 # 엑셀 형태로 저장하기
192 insta.to_excel(fx_name ,index=False , engine='openpyxl')
```

```
193
194 #Step 9. 요약 정보 출력하기
195 e_time = time.time( )
196 t_time = e_time - s_time
197
198 print("=" *120)
199 print("1.총 소요시간: %s 초" %round(t_time,1))
200 print("2.총 저장 건수: %s 건 " %count)
201 print("3.csv파일 저장 경로: %s" %fc_name)
202 print("4.xls파일 저장 경로: %s" %fx_name)
203 print("5.이미지파일 저장 경로: %s" %img_dir)
204 print("=" *120)
205 driver.close( )
```

****

코드가 약간 길죠?
다음 절에서 위 코드를 자세하게 설명하니까 걱정하지 마세요~^^

3. 소스코드 설명

이번 장에서 배우는 인스타그램 웹 크롤러도 앞에서 살펴봤던 다른 크롤러들과 비슷한 부분이 많습니다. 그래서 앞의 다른 웹 크롤러와 겹치는 부분은 최대한 간결하게 설명하겠습니다.

```
47  # Step 4. 인스타그램 접속 후 자동 로그인 하기
48  s_time = time.time( )
49  s = Service("c:/py_temp/chromedriver.exe")
50  driver = webdriver.Chrome(service=s)
51  url = "https://www.instagram.com/"
52  driver.get(url)
53  time.sleep(random.randrange(1,5))
54
55  print("₩n")
56  print("요청하신 데이터를 추출중이오니 잠시만 기다려 주세요~~~~^^")
57  print("₩n")
```

인스타그램은 계정과 비밀번호로 로그인을 해야 합니다.
그래서 Step 2의 21,22번 행에서 사용자에게 로그인 ID와 비밀번호를 입력 받아서 변수에 저장한 후 59번행부터 70번까지 로그인 하는 코드가 추가되었습니다.
먼저 60번 행부터 63번 행까지 ID를 입력하는 부분부터 보겠습니다.

[ID 입력 필드 정보 확인]


```
59  #ID와 비번 입력후 로그인하기
60  eid = driver.find_element(By.NAME,'username')
61  for a in v_id :
62          eid.send_keys(a)
63          time.sleep(0.3)
```

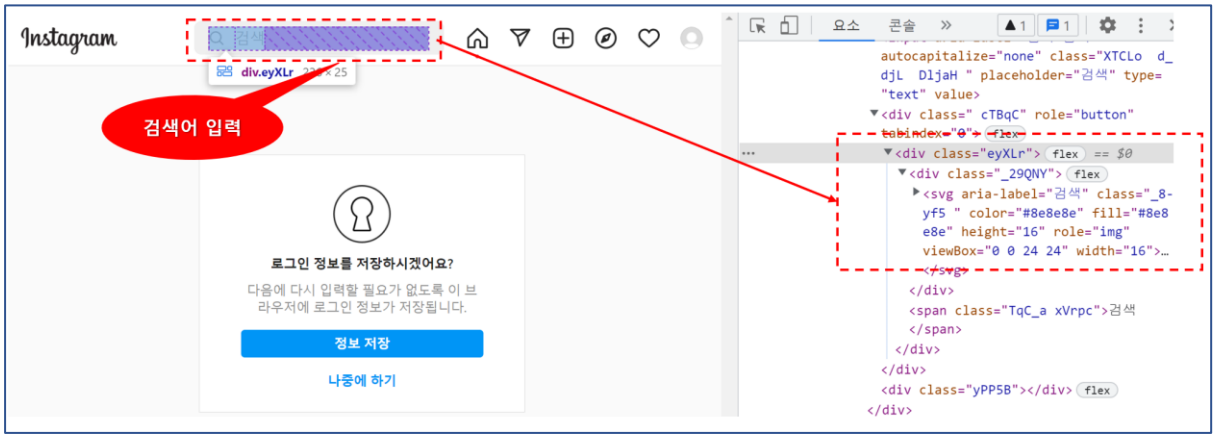
[비밀번호 입력 필드 정보 확인]

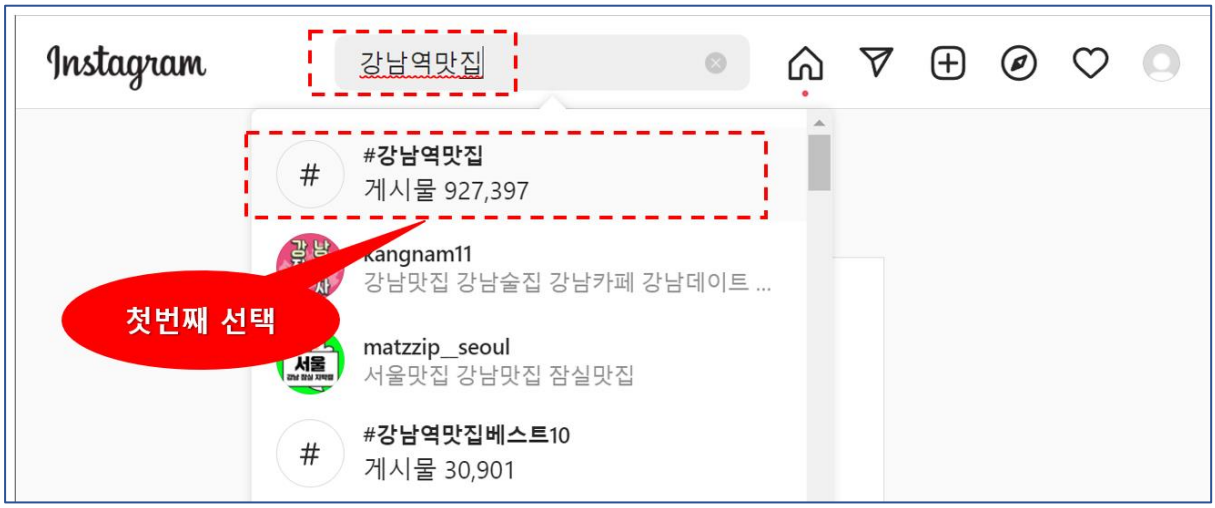

```
64  epwd = driver.find_element(By.NAME,'password')
65  for b in v_passwd :
66          epwd.send_keys(b)
67          time.sleep(0.5)
68
69  driver.find_element(By.XPATH,'//*[@id="loginForm"]/div/div[3]/button/div').click()
70  time.sleep(5)
```

위 코드의 60번 행부터 67번 행까지 ID와 비밀번호를 입력하는 부분입니다.
대부분의 사이트가 비슷한 경우가 많은데 ID나 비밀번호를 입력할 때 너무 빠른 속도로 입력할 경우 입력한 글자가 누락이 되는 경우가 자주 발생합니다.
그래서 위 코드와 같이 1글자 타이핑한 후 잠시 기다렸다가 다음 글자를 칠 수 있도록 반복문을 이용하였습니다.
이렇게 ID와 비밀번호를 모두 입력 후 로그인 버튼을 클릭하면 되는데 이때 로그인 버튼의 xpath 값을 사용하였습니다.

이제 잠시 기다리면 로그인이 완료되고 검색어를 입력할 수 있는 화면이 되는데 상단의 검색어를 입력하는 곳에 사용자가 입력한 키워드를 입력합니다.
이때 인스타그램은 사용자가 검색어를 입력하면 해당 검색어와 관련된 추천 검색어 목록을 하단에 보여 주는데 가장 게시글이 많은 검색어가 맨 위에 나옵니다
그래서 검색어를 입력한 후 추천 검색어 중에서 맨 위에 있는 검색어를 선택하면 되는데 이 부분을 아래의 코드와 같이 구현을 했습니다.

```
72  # Step 5. 검색할 키워드 입력하기
73  element = driver.find_element(By.XPATH,'//*[@id="react-root"]/section/nav/div[2]/div/div/div[2]/input')
74  for c in query_txt :
75          element.send_keys(c)
76          time.sleep(0.2)
77  time.sleep(3)
78  element.send_keys("₩n")
79  element.send_keys("₩n")
80  time.sleep(5)
```

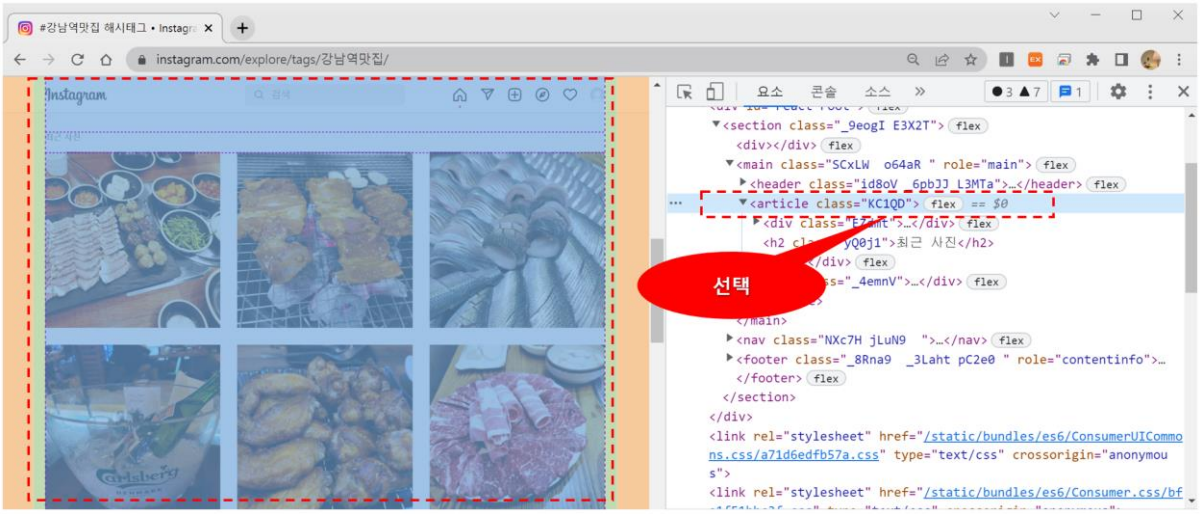




위 코드의 73번 행에서 검색어를 입력하는 창의 정보를 xpath 값을 이용해서 지정했습니다.
그리고 74번 행부터 76번 행까지 검색어를 천천히 입력했습니다.
중요한 것은 78 , 79번 행인데 특정 키워드를 입력하면 나오는 추천 키워드 중에서 맨 위의 것을 선택하기 위해서 엔터키를 2번 실행했습니다.

위 과정까지 진행하면 화면에 검색어와 관련된 다양한 게시물들이 보입니다.
이 게시물들을 하나씩 클릭하고 들어가서 대표 이미지와 # 기호로 시작하는 해시태그들을 수집해야 합니다. 이 작업을 하기 위해서는 각 게시물들의 URL 주소를 추출해야 하는데 아래와 같이 자동 스크롤 하는 함수를 만들어서 화면을 아래로 이동시킨 후 원하는 게시물들의 URL 주소를 추출하여 리스트에 저장하였습니다.
위의 설명 부분을 아래와 같이 코드로 작성하였습니다.

```
82  # Step 6. 전체 게시물의 원본 URL 추출하기
83  item=[ ]      # 인스타그램 URL 주소 저장할 리스트
84  item2=[ ]     # 중복값을 제거한 최종 URL 주소를 저장할 리스트
85
86  # 자동 스크롤다운 함수
87  def scroll_down(driver):
88      driver.execute_script("window.scrollTo(0,document.body.scrollHeight);")
89      time.sleep(5)
90
91  print('요청하신 데이터를 수집중이니 잠시만 기다려 주세요~^^')
92  print()
93
94  a = 1
95  while (a <= page_cnt):
96      scroll_down(driver)
97
98      html = driver.page_source
99      soup = BeautifulSoup(html, 'html.parser')
100
101     all_a = soup.find('article','KC1QD').find_all('a')
102
103     for i in all_a:
104         url = i['href']
105         item.append(url)
106         item2 = pd.Series(item).drop_duplicates()
107
108         if len(item2) >= cnt :
109             break
110     a += 1
111     print('요청하신 데이터를 수집중이니 잠시만 더 기다려 주세요~^^')
```



위 그림에서처럼 <article class="KC1QD"> 아래에 모든 게시물들의 정보들이 들어 있어서
위 코드의 101번 행에서 게시물들의 'a' 태그들을 전부 수집해서 all_a 라는 변수에 넣었습니다.
그리고 103번 행에서 반복문을 통해 각각의 a 태그에 있는 href 속성으로 지정된 URL 주소를 추출해서 105번 행에서 item 리스트에 추가를 했습니다.
그런데 보통 인스타그램 같은 SNS들은 한명의 사용자가 여러개의 게시글을 올리기에 중복된 게시물이나 사용자가 있을 수도 있습니다.

그래서 각 게시물들의 URL 주소값을 비교하여 중복값 여부를 체크한 후 중복되는 URL은 삭제하는 작업이 필요합니다. 이런 작업이 pandas 모듈에 있는 drop_duplicates( ) 함수가 수행합니다.
위 코드의 106번 행에서 이 함수를 사용하여 중복되는 URL 주소를 삭제한 후 item2 리스트에 저장하였습니다.
그런데 이렇게 수집된 URL 주소를 보면 도메인 주소가 생략된 형태입니다.
그래서 사용자가 도메인 주소를 추가해서 완전한 형태의 URL 주소로 생성해야 사용할 수 있는데 이 작업이 아래 코드에서 진행됩니다.

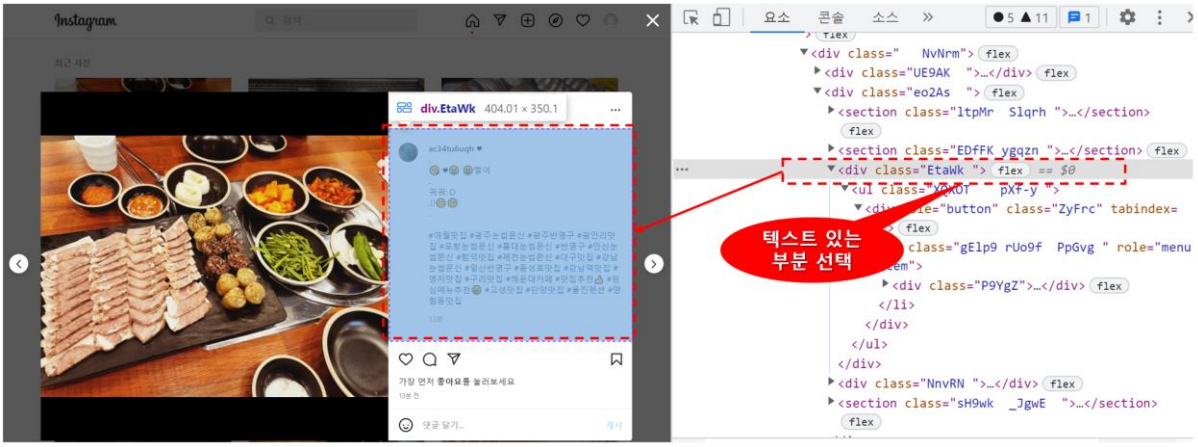
```
113 # 추출된 URL 사용하여 전체 URL 완성하기
114 full_url=[ ]
115 url_cnt = 1
116 print('= 수집될 인스타그램 주소는 아래와 같습니다 ==========')
117 for x in item2 :
118         url = 'https://www.instagram.com' + x
119         full_url.append(url)
120         print(url_cnt,':',url)
121
122         if url_cnt > cnt:
123                 break
124         url_cnt += 1
125 print('=========================================')
```

위 코드의 118번 행에서 도메인 주소와 수집된 URL 주소를 합쳐서 완전한 형태의 URL 주소를 생성합니다. 그리고 full_url 이라는 리스트에 추가합니다.
122,123 행은 사용자가 요청한 건수만큼의 URL 주소가 만들어지면 작업을 종료하는 부분입니다.

URL 정보를 추출했으니 이제 하나씩 접속해서 이미지와 해시태그값을 추출하겠습니다.

```
136 for c in full_url :
137         print()
138         driver.get(c)
139         time.sleep(random.randrange(3,9))
140 
141         html = driver.page_source
142         soup = BeautifulSoup(html, 'html.parser')
143         tags = soup.find('div','EtaWk')
144 
145         try :
146             tags_1 = tags.find_all('a')
147         except :
148             continue
149         else :
150             print('%s번째 게시물의 대표 이미지와 해쉬태그를 수집합니다~~~' %count)
151             print('게시물 URL:' , c )
152             no2.append(count)
153             url2.append(c)
```

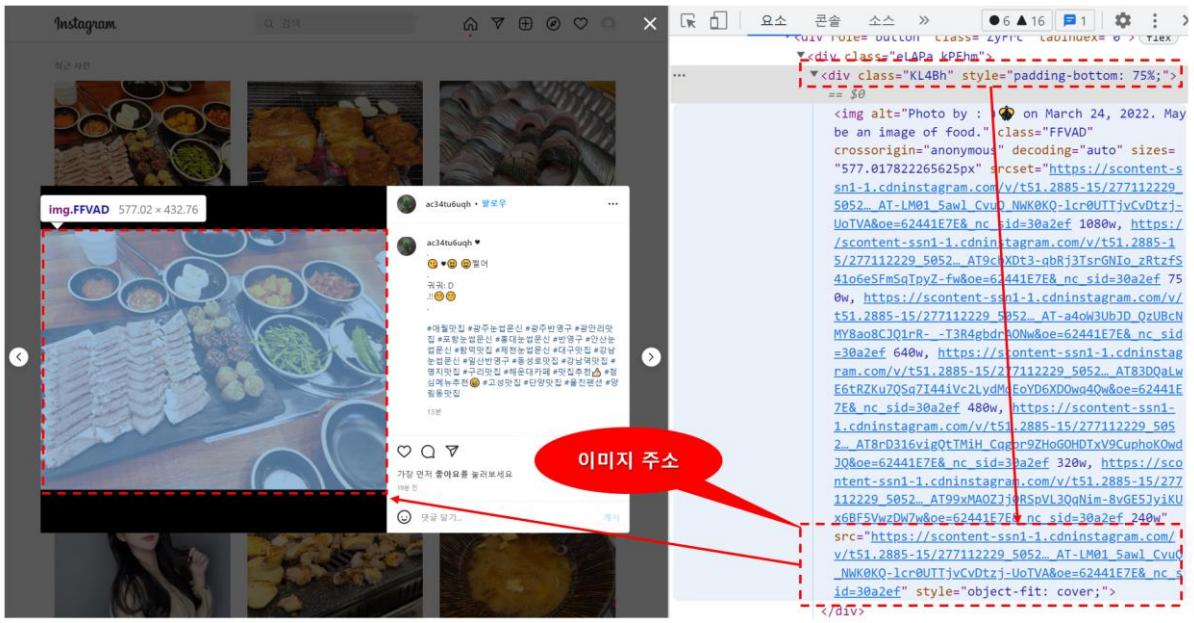
위 코드의 143번 행에서 각 게시물에서 텍스트가 저장되어 있는 태그를 찾아서 tags 변수에 지정했습니다. 아래 그림으로 텍스트가 저장되어 있는 태그를 확인할 수 있습니다.



이제 이미지와 해시 태그값을 추출하겠습니다.
먼저 이미지부터 저장하겠습니다.
아래 코드를 보세요.

```
155         #해당 페이지의 대표 이미지 수집
156         img_src = soup.find('div','KL4Bh').find('img')['src']
157         urllib.request.urlretrieve(img_src , str(count)+'.jpg')
158         print(img_dir,'아래에 %s번째 이미지 저장 완료===' %count)
159
```

아래 그림을 보면 이미지가 저장되어 있는 태그 값을 알 수 있습니다.
위 코드의 156번 행에서 해당 게시글의 이미지 URL 주소를 추출하여 img_src 변수에 할당한 후 157번 행에서 해당 이미지를 가져와서 저장하고 있습니다.



이미지를 저장했으니 이제 해시태그를 추출하여 저장하겠습니다.
해시 태그를 저장할 때 주의해야 할 부분은 2가지인데 첫번째는 저장하기 어려운 이모티콘이나 이미지를 만났을 때 어떻게 하는가입니다.
이런 경우를 처리하기 위해서 아래와 같이 저장 불가능한 이모티콘같은 비트맵 이미지를 만나면 특정 기호로 저장하도록 미리 사전을 만들어 두어야 합니다.

```
161         # 비트맵 이미지 아이콘을 위한 대체 딕셔너리를 만들기
162         import sys
168         bmp_map = dict.fromkeys(range(0x10000, sys.maxunicode + 1), 0xfffd)
```

두번째 주의 사항은 자음과 모음의 깨지는 현상을 해결해야 합니다.
인스타그램에 특정 태그나 내용을 등록한 후 수정을 하게 되면 자음과 모음의 분리 현상이 생깁니다. 인스타그램 페이지에서는 잘 보이는 텍스트들이 크롤링을 해서 저장을 하면 자음과 모음이 분리가 되어서 추출되고 저장됩니다.
당연히 이런 경우는 글자가 안되기 때문에 사용할 수도 없습니다.
이런 문제를 해결해주는 모듈이 바로 unicodedata 모듈입니다.
아래 코드를 보세요

```
165         hash_tags=[ ]
166         for d in tags_1 :
167             tags = d.get_text()
168             tags_11 = tags.translate(bmp_map)
169             tags_2 = unicodedata.normalize('NFC', tags_11)
170
171             if tags_2[0:1]=='#':
172
173                 hash_tags.append(tags_2)
174
175         print(hash_tags)
176         hash2.append(hash_tags)  # 각 게시물의 해시태그를 리스트 형태로 저장하기
```

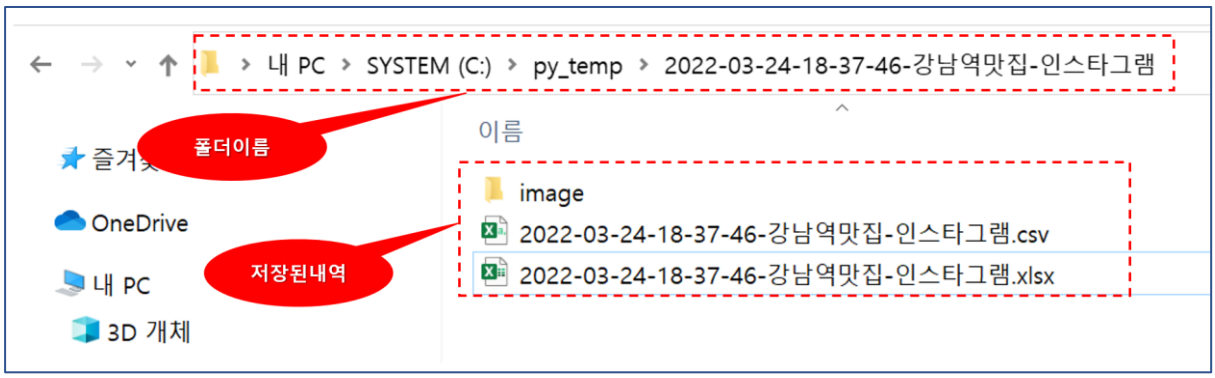
위 코드의 167번 행에서 텍스트를 추출합니다.
그리고 168번 행에서 추출된 텍스트가 표현 불가한 것일 때 168번 행에서 미리 선언해 둔 문자로 대체하여 저장합니다. 그리고 169번 행에서 자음과 모음이 분리되는 현상이 발생할 경우 수정하여 정상적인 글자로 변환해서 tags_2 변수에 할당합니다.
해시태그란 첫 글자가 # 기호로 시작하는 글자이므로 171번 행에서 이 부분을 체크하여 173번 행에서 첫 글자가 # 기호로 시작할 경우 hash_tags 리스트에 추가하도록 작성했습니다.
그리고 176번행은 1건의 게시물에 해시 태그가 여러 개인 경우에 여러개의 해시 태그를 1개의 리스트에 담아서 게시물 1건당 리스트 1개로 저장하였습니다.
이렇게 수집된 내용들을 아래와 같이 저장하였습니다.

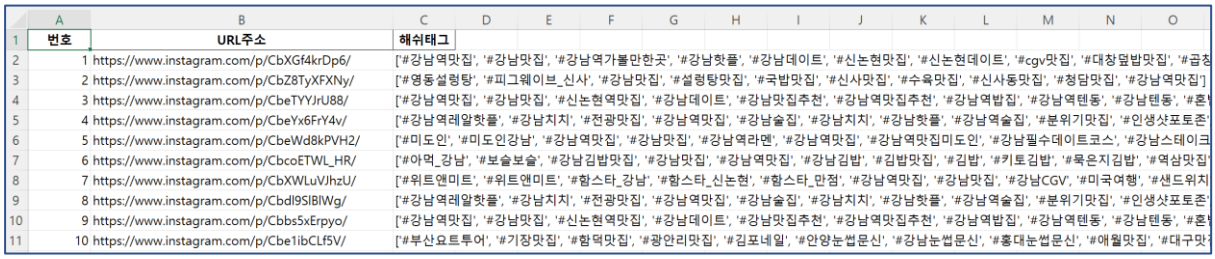
```
181 #Step 8. 수집된 해시태그를 csv , xls 형식으로 저장하기
182 # xls , csv로 저장하기 위해 데이터 프레임 생성하기
183 insta = pd.DataFrame( )
184 insta['번호'] = no2
185 insta['URL주소'] = url2
186 insta['해쉬태그'] = pd.Series(hash2)
187
```

위와 같이 데이터프레임으로 생성한 후 아래와 같이 csv , xls 형태로 저장하였습니다.

```
188 # csv 형태로 저장하기
189 insta.to_csv(fc_name,encoding="utf-8-sig",index=False)
190
191 # 엑셀 형태로 저장하기
192 insta.to_excel(fx_name ,index=False , engine='openpyxl')
```

저장된 결과를 볼까요?





	A	B	C
1	번호	URL주소	해쉬태그
2	1	https://www.instagram.com/p/CbXGf4krDp6/	['#강남역맛집', '#강남맛집', '#강남역가볼만한곳', '#강남핫플', '#강남데이트', '#신논현맛집', '#신논현데이트', '#cgv맛집', '#대창덮밥맛집', '#곱창
3	2	https://www.instagram.com/p/CbZ8TyXFXNy/	['#영동설렁탕', '#피그웨이브_신사', '#강남맛집', '#설렁탕맛집', '#국밥맛집', '#신사맛집', '#수육맛집', '#신사동맛집', '#청담맛집', '#강남역맛집']
4	3	https://www.instagram.com/p/CbeTYYJrU88/	['#강남역맛집', '#강남맛집', '#신논현역맛집', '#강남데이트', '#강남맛집추천', '#강남역맛집추천', '#강남역밥집', '#강남역텐동', '#강남텐동', '#혼
5	4	https://www.instagram.com/p/CbeYx6FrY4v/	['#강남역레알핫플', '#강남치치', '#전광맛집', '#강남역맛집', '#강남술집', '#강남치치', '#강남핫플', '#강남역술집', '#분위기맛집', '#인생샷포토존'
6	5	https://www.instagram.com/p/CbeWd8kPVH2/	['#미도인', '#미도인강남', '#강남역맛집', '#강남맛집', '#강남역라멘', '#강남역맛집', '#강남역맛집미도인', '#강남필수데이트코스', '#강남스테이크
7	6	https://www.instagram.com/p/CbcoETWL_HR/	['#아먹_강남', '#보슬보슬', '#강남김밥맛집', '#강남맛집', '#강남역맛집', '#강남김밥', '#김밥맛집', '#김밥', '#키토김밥', '#묵은지김밥', '#역삼맛집'
8	7	https://www.instagram.com/p/CbXWLuVJhzU/	['#위트앤미트', '#위트앤미트', '#함스타_강남', '#함스타_신논현', '#함스타_만점', '#강남역맛집', '#강남맛집', '#강남CGV', '#미국여행', '#샌드위치
9	8	https://www.instagram.com/p/Cbdl9SlBlWg/	['#강남역레알핫플', '#강남치치', '#전광맛집', '#강남역맛집', '#강남술집', '#강남치치', '#강남핫플', '#강남역술집', '#분위기맛집', '#인생샷포토존'
10	9	https://www.instagram.com/p/Cbbs5xErpyo/	['#강남역맛집', '#강남맛집', '#신논현역맛집', '#강남데이트', '#강남맛집추천', '#강남역맛집추천', '#강남역밥집', '#강남역텐동', '#강남텐동', '#혼
11	10	https://www.instagram.com/p/Cbe1ibCLf5V/	['#부산요트투어', '#기장맛집', '#함덕맛집', '#광안리맛집', '#김포네일', '#안양눈썹문신', '#강남눈썹문신', '#홍대눈썹문신', '#애월맛집', '#대구맛

이번 챕터에서도 다양한 내용들을 많이 배웠습니다.
열심히 연습해서 꼭 실력으로 만드세요~~

4. 연습문제로 실력 굳히기

1. 네이버 블로그에서 작성자 / 작성일자 / 본문내용 / 본문에 사용된 이미지 정보를 수집하여 txt 형식의 파일로 저장하는 문제입니다.

크롤링할 블로그 URL : https://blog.naver.com/hy820715/221514204265

위 주소의 블로그에 접속 후 예시와 같은 내용을 수집해서 저장하세요.

[크롤러 실행 화면 예시]

```
================================================================================
 연습문제 : 네이버 블로그 상세 내역과 댓글 정보 추출하여 저장하기
================================================================================
1.크롤링할 블로그 주소를 입력하세요: https://blog.naver.com/hy820715/221514204265
2.결과 파일을 저장할 폴더명만 쓰세요(예:c:₩py_temp₩):
```

[크롤러 수집 화면 예시]

```
 블로그 데이터를 수집합니다==========
================================================================================
1.블로그주소:  https://blog.naver.com/hy820715/221514204265
2.작성자 닉네임:  가치랩장입니다
3.작성일자: 2019. 4. 15. 16:27
4.블로그내용:
 웹 크롤링 진짜 많이 재미있죠?웹 크롤링에 대한 다양한 예제는 바로 이 책에 들어 있어요~  이 책안에는 다양한 유형의 웹사이트를 파이썬과 셀레니움을 활용하여 수집하는 노하우들이 다 들어 있습니다~하나씩 하나씩 따라하다 보면 금방 실력자가 되어 있으실 거예요~~^^그리고 수집된 데이터를 분석할 때  파이썬도 많이 사용하지만 R 프로그램도 많이 사용합니다~R 프로그램을 공부하시려면 아래의 책을 추천해드려요~~  위의 책에는 R 프로그램을 사용하여 텍스트 데이터를 분석하는 다양한 방법들부터 R 프로그램을 활용한 시각화 ,  지도 작업 , 정형 데이터를 핸들링하는 다양한 패키지 설명까지 제공되고 있어서 쉽고 빠르게 R 프로그램 사용 방법을 배우실 거예요~~무엇보다도 절대로 포기하지 말고 열공하는 자세가 가장 중요합니다~~^^열공해 주세요~~^^가치랩장 드림.


https://postfiles.pstatic.net/MjAx0TA3MjIfMTQ0/MDAxNTY0MzcwMDYxNzYy.3iQRsQbrL8btKFujR9tFXCT9_PKu3DsiM6up1YBFhAcg.Yj62i09BekL90Y8AnBmP2Fkc9kkfWbH0cPZh2rqf3k4g.JPEG.hy820715/완친파_표지_최종.jpg?type=w966
1 -이미지 저장 완료
https://postfiles.pstatic.net/MjAx0TA0MTVfNDgg/MDAxNTU1MzEzMDg3NTM1.biuF2sH30dCYq9ZITLLjl3rZNoanAYXM2dT4_0qS89sg.ttQz7rYhTwklVdSM_y09UEbU5h35n096W6DdR10GXLQg.JPEG.hy820715/책표지_-_한국정보인재개발원_jpg.jpg?type=w966
2 -이미지 저장 완료


================================================================================
총 소요시간은 7.2 초 입니다
파일 저장 완료: txt 파일명 : c:₩py_temp₩2022-03-26-08-12-24-블로그댓글수집₩2022-03-26-08-12-24-블로그댓글수집.txt
파일 저장 완료: csv 파일명 : c:₩py_temp₩2022-03-26-08-12-24-블로그댓글수집₩2022-03-26-08-12-24-블로그댓글수집.csv
파일 저장 완료: xls 파일명 : c:₩py_temp₩2022-03-26-08-12-24-블로그댓글수집₩2022-03-26-08-12-24-블로그댓글수집.xls
================================================================================
```

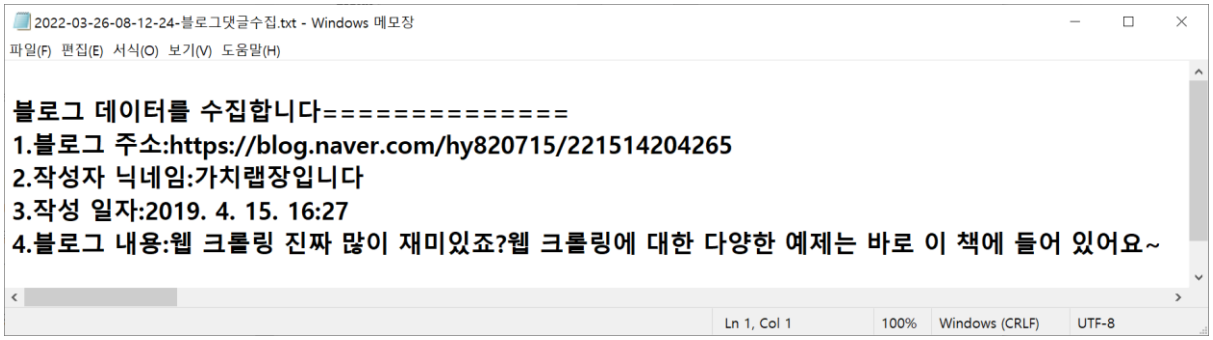
[수집된 결과 화면 예시]

1.jpg

2.jpg

2022-03-26-08-12-24-블로그댓글수집.txt

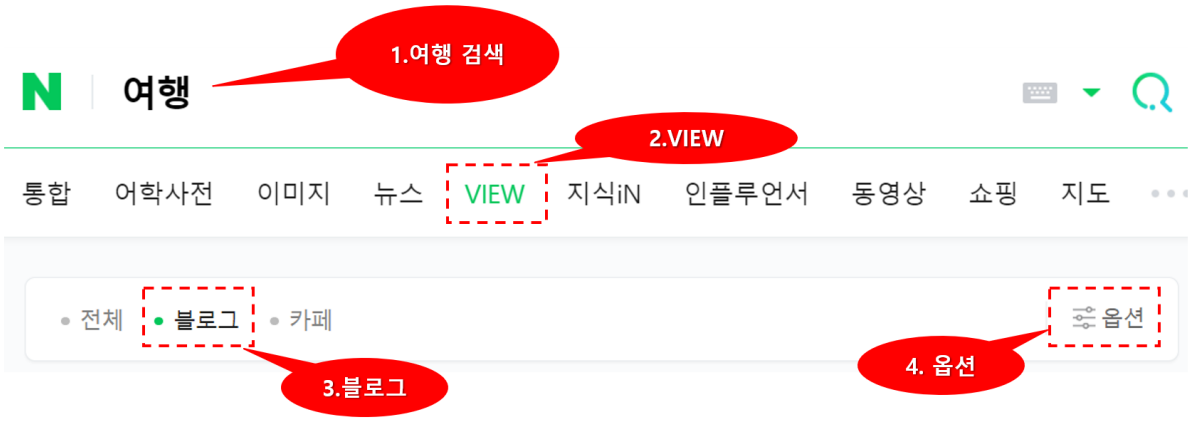
[txt 파일 내용 예시]



2. 네이버에서 특정 키워드로 검색 후 여러 건의 블로그의 이미지와 텍스트 정보 수집하기

이번 연습문제는 네이버에서 "여행" 키워드로 검색한 후 View -> 블로그를 선택하고 조회를 시작할 날짜와 종료할 날짜를 옵션에서 선택하여 해당 블로그의 이미지 데이터와 텍스트 데이터를 수집한 후 텍스트 데이터들은 txt , xls , csv 형태로 저장하는 문제입니다.

그림으로 살펴볼까요?



위 그림에서 4.옵션 메뉴를 클릭하면 아래 그림과 같이 여러가지 옵션이 나옵니다.

위 그림에서 기간을 맨 오른쪽의 직접입력을 클릭한 후 조회시작일에 해당되는 년 , 월 , 일을 직접 클릭하고 조회 종료일도 년 , 월 , 일을 직접 클릭해야 합니다(날짜를 타이핑할 수 없습니다)

테스트로 사용하는 기간은 2021-01-01 ~ 2021-12-31 로 하세요~

검색된 결과가 아래와 같이 많이 나옵니다.

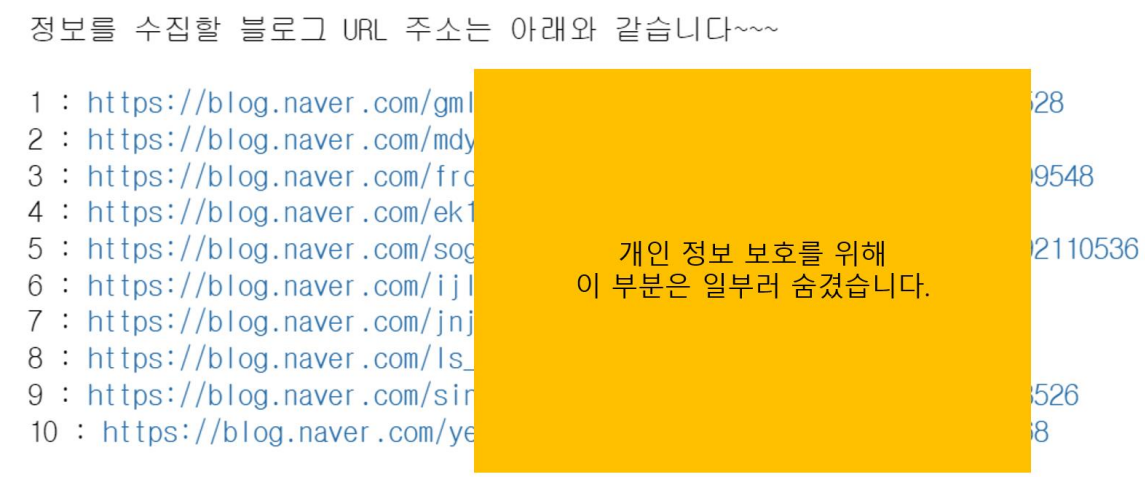
개인정보를 보호하기 위해 화면은 일부러 모자이크 처리했습니다.
위 목록들의 블로그를 하나씩 들어가서 아래와 같이 상세 내역을 추출한 후 xls , csv 형태의 파일로 저장하면 됩니다.

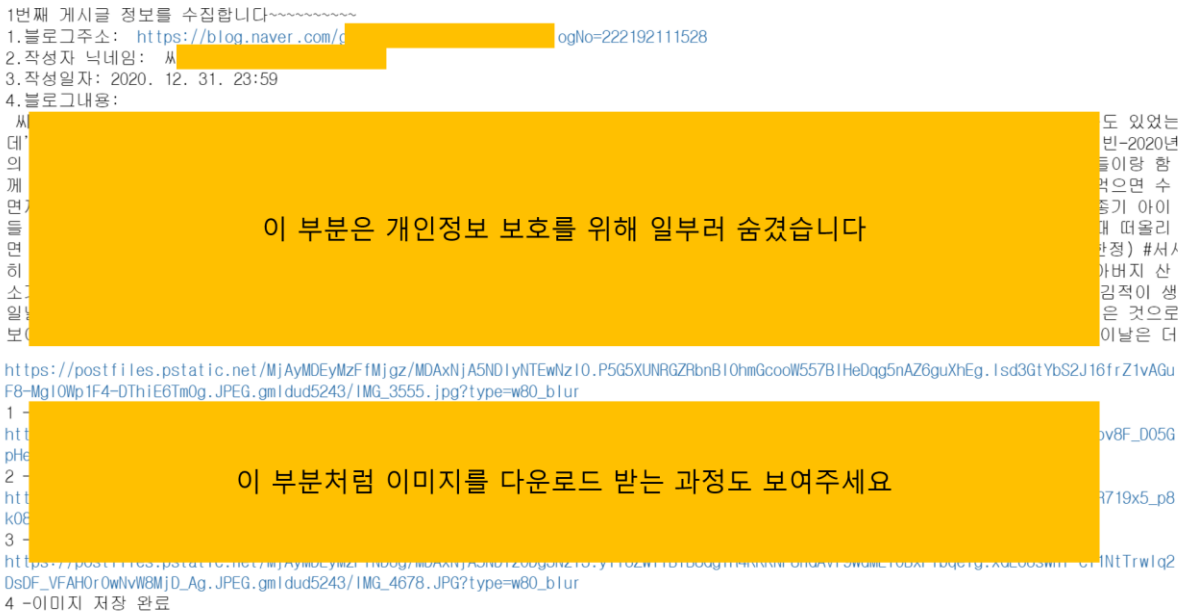
[웹 크롤러 실행 화면 예시]

```
================================================================================
 연습문제: 블로그  크롤러 : 네이버 view -> 블로그 정보 수집하기
================================================================================
1.정보를 수집할 키워드는 무엇입니까?: 제주도여행
2.조회를 시작할 날짜를 입력하세요(예:2017-01-01) :2020-01-01
3.조회를 종료할 날짜를 입력하세요(예:2017-12-31): 2020-12-31
4.몇 건의 정보를 수집할까요? :10
5.파일을 저장할 폴더명만 쓰세요(기본값:c:₩py_temp₩):c:₩py_temp₩
```

위 그림과 같이 키워드와 조회시작일 , 조회종료일 , 수집할 건수 , 저장할 폴더명을 입력 받도록 코드를 작성하세요.

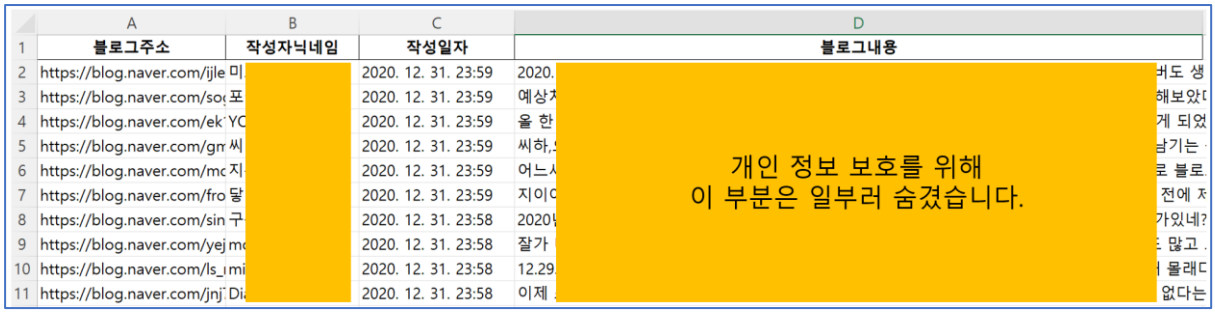
[웹 크롤링 진행 과정 예시]



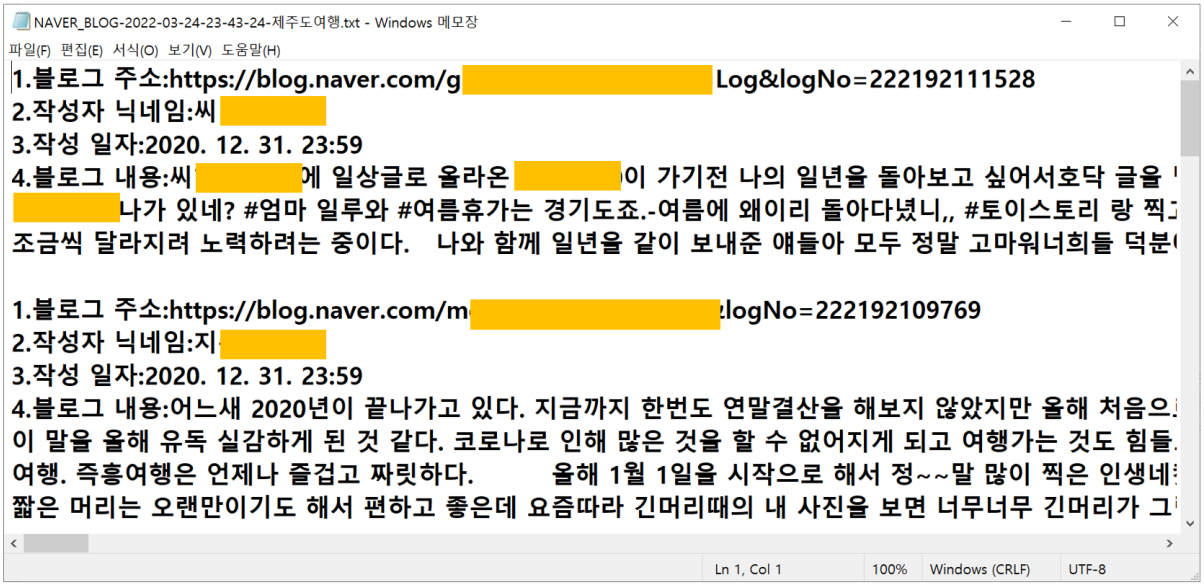


[파이썬 능력자 너도 될 수 있어~! - 서진수 저 -]

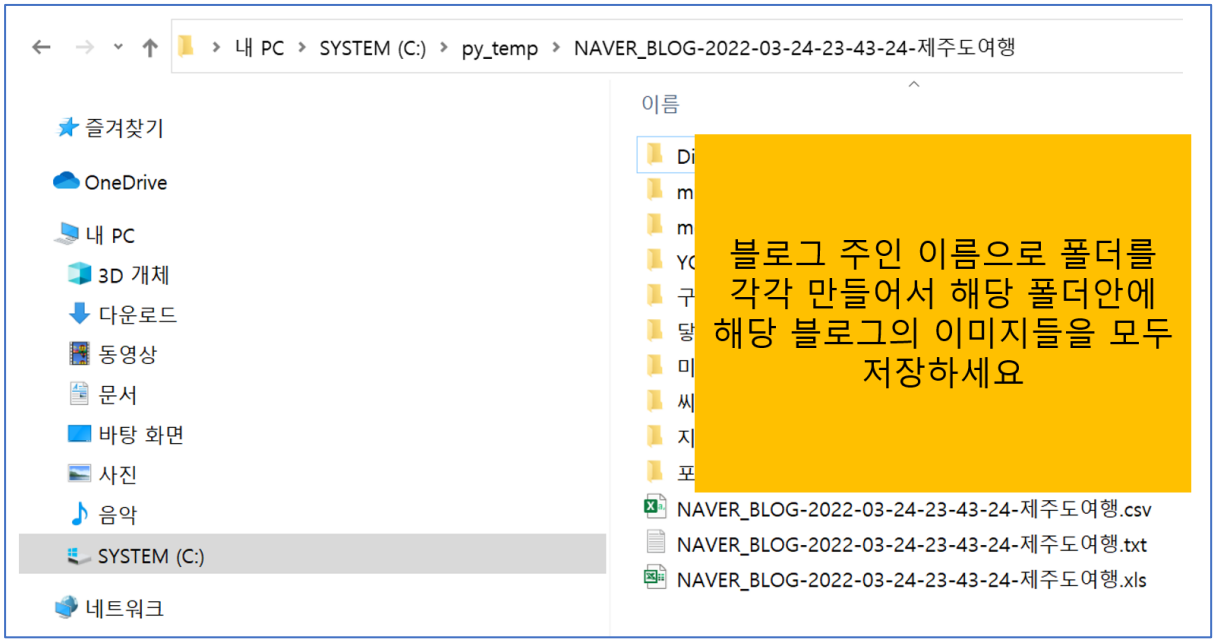
[xls / csv 파일 저장 예시]



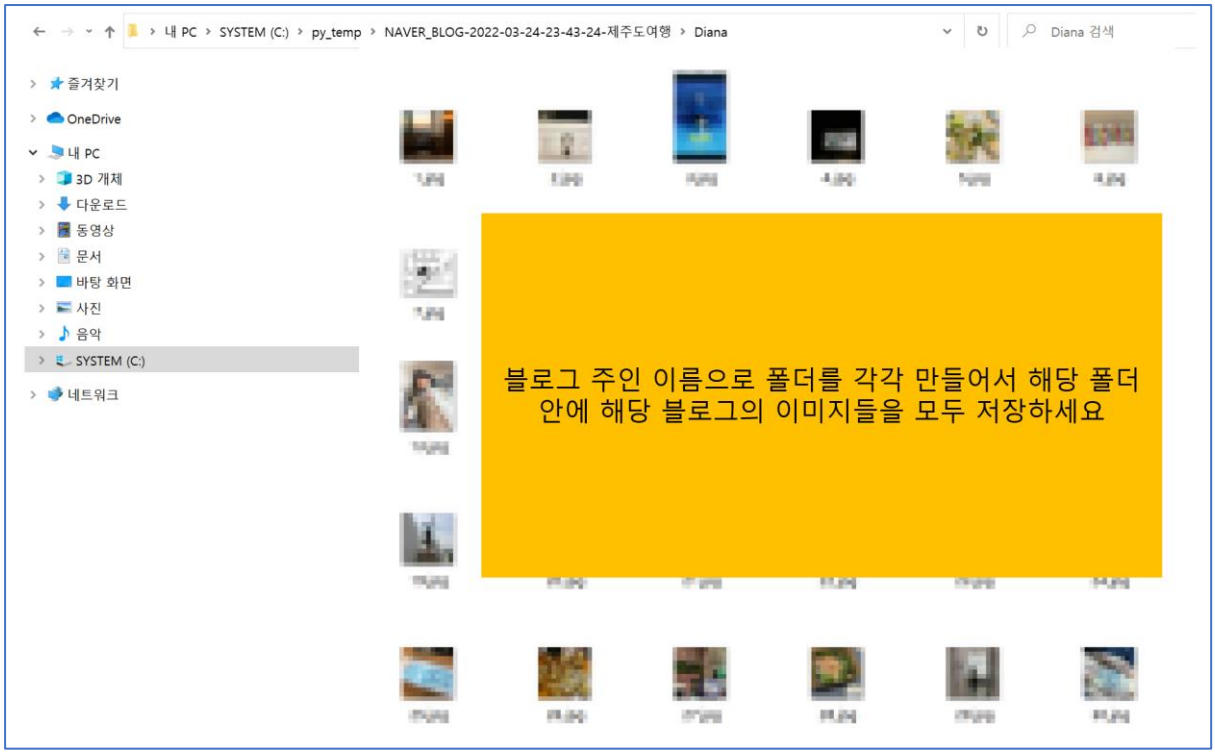
[txt 파일 저장 예시]



[폴더 저장 예시]



[파이썬 능력자 너도 될 수 있어~! - 서진수 저 -]



이번 챕터의 내용도 열심히 공부하셔서 꼭 실력으로 만드세요!!