Chap 8. 다양한 인터넷 쇼핑몰 정보 수집하기

1. 이번 장에서 배울 내용 소개

이번 시간에는 지난 시간까지 배웠던 모든 지식을 동원해서 실제 업무에서 많이 사용될 수 있는 인터넷 쇼핑몰의 판매 정보를 수집하고 저장하는 방법을 배우겠습니다.
이번 시간에 사용할 예제 사이트는 한국의 쿠팡(https://www.coupang.com) 사이트입니다.
쿠팡 사이트에서 식품 카테고리에서 많이 판매된 Best Seller 상품의 목록을 추출하여 다양한 형식의 파일로 저장하는 방법을 알려 드리겠습니다.

*** 학습목표**

1. 사진을 포함한 다양한 정보를 수집하여 저장할 수 있다

2. 수집된 사진을 xls 파일에 추가할 수 있다.

이번 시간에 공부할 내용을 미리 살펴볼까요?

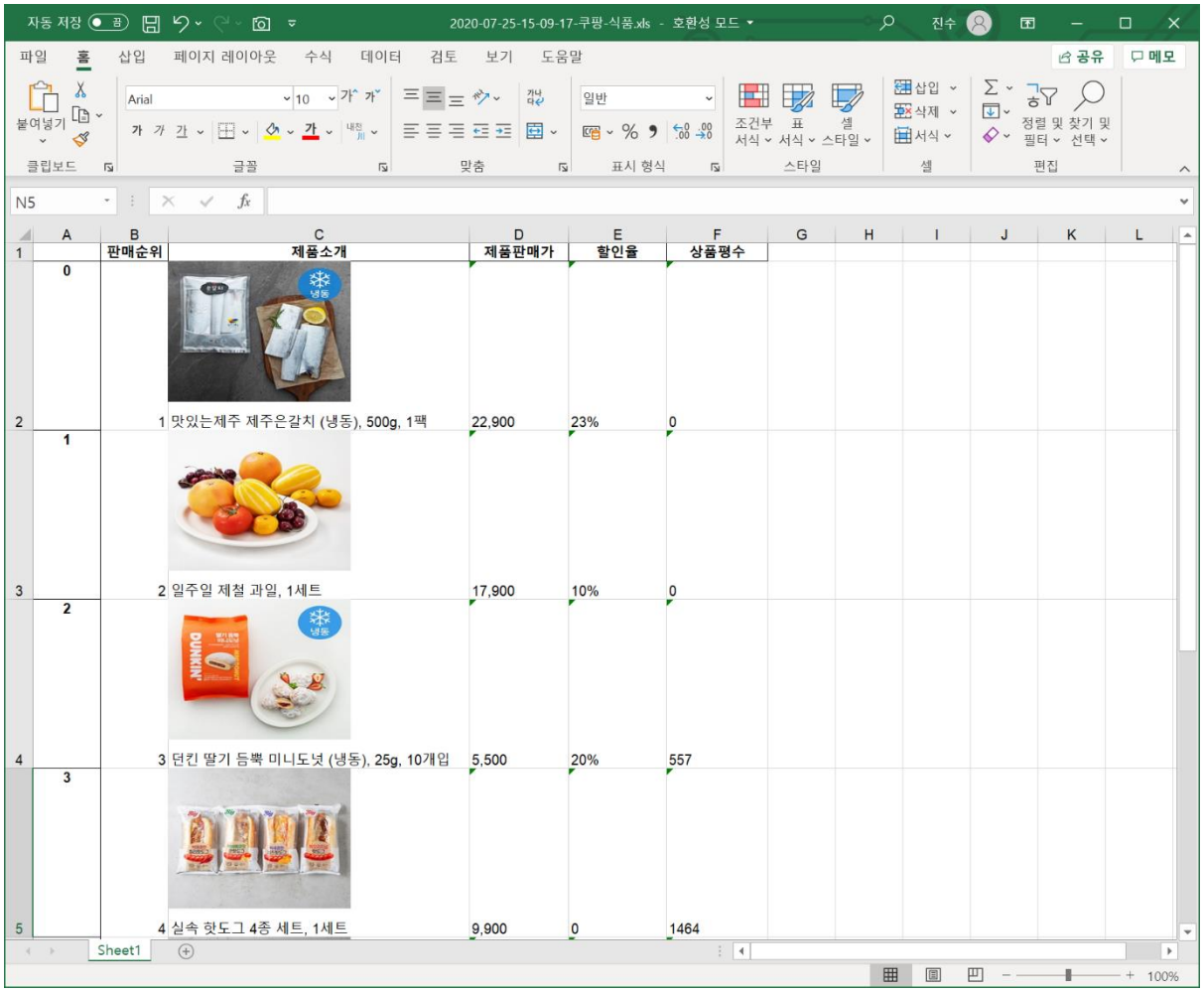
[실행화면 예시]

```
================================================================================
 쿠팡 사이트의 식품 카테고리 Best Seller 상품 정보 추출하기
================================================================================
1.크롤링 할 건수는 몇건입니까?: 200
2.파일을 저장할 폴더명만 쓰세요(기본경로:c:₩temp₩):
```

[수집 결과 예시]

```
--------------------------------------------------------------------
1.판매순위: 1
2.제품소개: 고디바 스페셜 빼빼로 데이 비스킷 세트, 밀크 비스킷 + 다크 비스킷 + 다크 핫 초콜렛사 1+1쿠폰, 1세
트
3.판매가격: 29,990
4:할인률: 10%
5.상품평 수:  0
--------------------------------------------------------------------
--------------------------------------------------------------------
1.판매순위: 2
2.제품소개: 빼빼로 자판기 세트, 아몬드맛 9개입 + 오리지널 9개입 + 누드초코 9개입 + 크런키 9개입, 1세트
3.판매가격: 31,500
4:할인률: 10%
5.상품평 수:  0
--------------------------------------------------------------------
--------------------------------------------------------------------
1.판매순위: 3
2.제품소개: 죠리퐁 마시멜로, 35g, 18개
3.판매가격: 14,220
4:할인률: 8%
5.상품평 수: 761
--------------------------------------------------------------------
```

[파일 저장 예시]



	A	B	C	D	E	F
1		판매순위	제품소개	제품판매가	할인율	상품평수
2	0	1	맛있는제주 제주은갈치 (냉동), 500g, 1팩	22,900	23%	0
3	1	2	일주일 제철 과일, 1세트	17,900	10%	0
4	2	3	던킨 딸기 듬뿍 미니도넛 (냉동), 25g, 10개입	5,500	20%	557
5	3	4	실속 핫도그 4종 세트, 1세트	9,900	0	1464

이번 시간에 공부하는 내용은 쿠팡 뿐만 아니라 국내외의 모든 온라인 쇼핑몰 데이터 수집에 바로 적용할 수 있는 즉 실무에서 바로 사용할 수 있는 내용이라서 소스 코드가 약간 길고 내용도 복잡하지만 열심히 공부해서 꼭 독자님의 실력으로 만들기 바랍니다.

2. 전체 코드 미리 보기

(아래 코드는 저자가 제공해 드린 코드를 사용하세요~)

****

```
1   #Step 1. 필요한 모듈과 라이브러리를 로딩합니다.
2   from bs4 import BeautifulSoup
3   from selenium import webdriver
4   from selenium.webdriver.common.by import By
5   from selenium.webdriver.common.keys import Keys
6   from selenium.webdriver.chrome.service import Service
7   import urllib.request
8   import urllib
9   import time
10  import pandas as pd
11  import os
12  import math
13
14  #Step 2. 사용자에게 검색어 키워드를 입력 받습니다.
15  print("=" *80)
16  print(" 쿠팡 사이트의 식품 카테고리 Best Seller 상품 정보 추출하기 ")
17  print("=" *80)
18
19  cnt = int(input('1.크롤링 할 건수는 몇건입니까?: '))
20  page_cnt = math.ceil(cnt/60)
21
22  f_dir = input("2.파일을 저장할 폴더명만 쓰세요(기본경로:c:₩₩py_temp₩₩):")
23  if f_dir == '' :
24      f_dir = "c:₩₩py_temp₩₩"
25
26  print("₩n")
27
28  if cnt > 30 :
29    print("      요청 건수가 많아서 시간이 제법 소요되오니 잠시만 기다려 주세요~~")
30  else :
31   print("      요청하신 데이터를 수집하고 있으니 잠시만 기다려 주세요~~")
32
33  #Step 3.저장될 파일 경로와 이름을 지정합니다
34  sec_name = '식품'
35  query_txt='쿠팡'
36
```

```
37  n = time.localtime()
38  s1 = '%04d-%02d-%02d-%02d-%02d-%02d' % (n.tm_year, n.tm_mon, n.tm_mday, n.tm_hour, n.tm_min, n.tm_sec)
39
40  os.makedirs(f_dir+s1+'-'+query_txt+'-'+sec_name)
41  os.chdir(f_dir+s1+'-'+query_txt+'-'+sec_name)
42
43  ff_dir=f_dir+s1+'-'+query_txt+'-'+sec_name
44  ff_name=f_dir+s1+'-'+query_txt+'-'+sec_name+'₩₩'+s1+'-'+query_txt+'-'+sec_name+'.txt'
45  fc_name=f_dir+s1+'-'+query_txt+'-'+sec_name+'₩₩'+s1+'-'+query_txt+'-'+sec_name+'.csv'
46  fx_name=f_dir+s1+'-'+query_txt+'-'+sec_name+'₩₩'+s1+'-'+query_txt+'-'+sec_name+'.xls'
47
48  # 제품 이미지 저장용 폴더 생성
49  img_dir = ff_dir+"₩₩images"
50  os.makedirs(img_dir)
51  os.chdir(img_dir)
52
53  s_time = time.time( )
54
55  #Step 4. 웹사이트 접속 후 해당 메뉴로 이동합니다.
56  s = Service("c:/py_temp/chromedriver.exe")
57  driver = webdriver.Chrome(service=s)
58
59  query_url='https://www.coupang.com/'
60  driver.get(query_url)
61  time.sleep(5)
62
63  # Access Denied 메시지가 나오면 아래코드로 쿠키를 삭제한다
64  driver.delete_all_cookies()
65  time.sleep(2)
66
67  # 카테고리 -> 식품 버튼을 눌러 페이지를 엽니다
68  driver.find_element(By.XPATH , '//*[@id="header"]/div').click( )
69  driver.find_element(By.XPATH , '//*[@id="gnbAnalytics"]/ul[1]/li[4]/a').click( )
70
71  #Step 5. 내용을 수집합니다
72  print("₩n")
73  print("===== 곧 수집된 결과를 출력합니다 ^^ ===== ")
74  print("₩n")
75
```

```
76  ranking2=[]           #제품의 판매순위 저장
77  title2=[]             #제품 정보 저장
78  p_price2=[]           #현재 판매가 저장
79  discount2 = []        #할인율 저장
80  sat_count2=[]         #상품평 수 저장
81
82  img_src2=[]    # 이미지 URL 저장변수
83  file_no = 0    # 이미지 파일 저장할 때 번호
84  count = 1       # 총 게시물 건수 카운트 변수
85
86  def scroll_down(driver):
87      driver.execute_script("window.scrollBy(0,1100);")
88      time.sleep(4)
89
90  for x in range(1,page_cnt + 1) :
91
92      for cc in range(1,7) :
93          scroll_down(driver)
94
95      html = driver.page_source
96      soup = BeautifulSoup(html, 'html.parser')
97
98      item_result = soup.find('ul','baby-product-list').find_all('li')
99
100     for li in item_result :
101         if cnt < count :
102             break
103
104         # 제품 이미지 다운로드 하기
105         try :
106             photo = li.find('dt','image').find('img')['src']
107         except AttributeError :
108             continue
109
110         file_no += 1
111         full_photo = 'https:' + photo
112         urllib.request.urlretrieve(full_photo,str(file_no)+'.jpg')
113         time.sleep(0.5)
114
```

```
115         #제품 내용 추출하기
116         f = open(ff_name, 'a',encoding='UTF-8')
117         f.write("-------------------------------------------------"+"₩n")
118         print("-" *70)
119
120         ranking = count
121         print("1.판매순위:",ranking)
122         f.write('1.판매순위:'+ str(ranking) + "₩n")
123
124         try :
125             t = li.find('div','name').get_text().replace("₩n","")
126         except :
127             title = '제품소개가 없습니다'
128             print(title.replace("₩n",""))
129             f.write('2.제품소개:'+ title + "₩n")
130       else :
131             title = t.replace("₩n","").strip()
131             print("2.제품소개:", title.replace("₩n","").strip())
132             f.write('2.제품소개:'+ title + "₩n")
133
134         try :
135             p_price = li.find('strong','price-value').get_text().replace("₩n","")
136         except :
137             p_price = '0'
138             print("3.판매가격:", p_price.replace("₩n",""))
139             f.write('3.판매가격:'+ p_price + "₩n")
140         else :
141             print("3.판매가격:", p_price.replace("₩n",""))
142             f.write('3.판매가격:'+ p_price + "₩n")
143
144         try :
145             discount = li.find('span','discount-percentage').get_text().replace("₩n","")
146         except   :
147             discount = '0'
148             print("4:할인률:", discount)
149             f.write('4.할인율:'+ discount + "₩n")
150         else :
151             print("4:할인률:", discount)
152             f.write('4.할인율:'+ discount + "₩n")
```

```
153
154             try :
155                 sat_count_1 = li.find('span','rating-total-count').get_text()
156                 sat_count_2 = sat_count_1.replace("(","").replace(")","")
157             except   :
158                 sat_count_2='0'
159                 print('5.상품평 수: ',sat_count_2)
160                 f.write('5.상품평 수:'+ sat_count_2 + "₩n")
161             else :
162                 print('5.상품평 수:',sat_count_2)
163                 f.write('5.상품평 수:'+ sat_count_2 + "₩n")
164
165             print("-" *70)
166
167             f.close( )
168             time.sleep(0.5)
169
170             ranking2.append(ranking)
171             title2.append(title.replace("₩n",""))
172
173             p_price2.append(p_price.replace("₩n",""))
174             discount2.append(discount)
175
176             try :
177                 sat_count2.append(sat_count_2)
178             except IndexError :
179                 sat_count2.append(0)
180
181             count += 1
182         x += 1
183         try :
184             driver.find_element(By.LINK_TEXT, '%s' %x).click() # 다음 페이지번호 클릭
185         except :
186             break
187
188 #step 6. csv , xls 형태로 저장하기
189 co_best_seller = pd.DataFrame()
190 co_best_seller['판매순위']=ranking2
191 co_best_seller['제품소개']=pd.Series(title2)
```

```
192 co_best_seller['제품판매가']=pd.Series(p_price2)
193 co_best_seller['할인율']=pd.Series(discount2)
194 co_best_seller['상품평수']=pd.Series(sat_count2)
195
196 # csv 형태로 저장하기
197 co_best_seller.to_csv(fc_name,encoding="utf-8-sig",index=False)
198
199 # 엑셀 형태로 저장하기
200 co_best_seller.to_excel(fx_name ,index=False , engine='openpyxl')
201
202 e_time = time.time( )
203 t_time = e_time - s_time
204
205 count -= 1
206 print("₩n")
207 print("=" *80)
208 print("1.요청된 총 %s 건의 리뷰 중에서 실제 크롤링 된 리뷰수는 %s 건입니다" %(cnt,count))
209 print("2.총 소요시간은 %s 초 입니다 " %round(t_time,1))
210 print("3.파일 저장 완료: txt 파일명 : %s " %ff_name)
211 print("4.파일 저장 완료: csv 파일명 : %s " %fc_name)
212 print("5.파일 저장 완료: xls 파일명 : %s " %fx_name)
213 print("=" *80)
214
215 #Step 7. xls 파일에 제품 이미지 삽입하기
216 import win32com.client as win32     #pywin32 , pypiwin32 설치후 동작
217 import win32api                      #파이썬 프롬프트를 관리자 권한으로 실행해야 에러없음
218
219 excel = win32.gencache.EnsureDispatch('Excel.Application')
220 wb = excel.Workbooks.Open(fx_name)
221 sheet = wb.ActiveSheet
222 sheet.Columns(2).ColumnWidth = 30
223 row_cnt = cnt+1
224 sheet.Rows("2:%s" %row_cnt).RowHeight = 120
225
226 ws = wb.Sheets("Sheet1")
227 col_name2=[]
228 file_name2=[]
229
230 for a in range(2,cnt+2) :
```

```
231         col_name='B'+str(a)
232         col_name2.append(col_name)
233
234 for b in range(1,cnt+1) :
235         file_name=img_dir+'₩₩'+str(b)+'.jpg'
236         file_name2.append(file_name)
237
238 for i in range(0,cnt) :
239         rng = ws.Range(col_name2[i])
240         image = ws.Shapes.AddPicture(file_name2[i], False, True, rng.Left, rng.Top, 130, 100)
241         excel.Visible=True
242         excel.ActiveWorkbook.Save()
243
244 driver.close( )
```

****

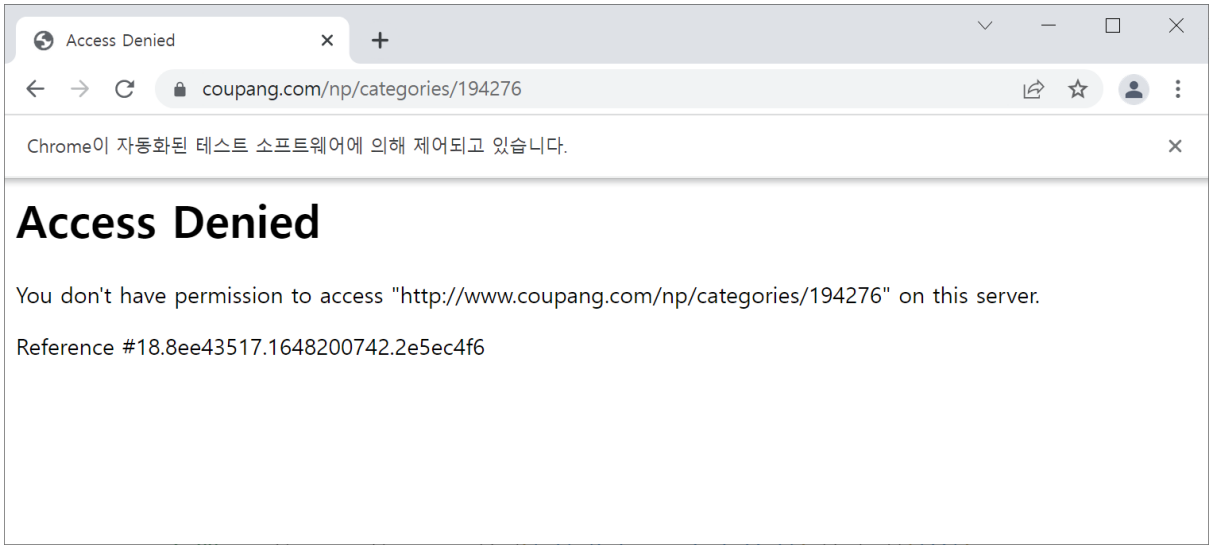
소스코드가 조금 길죠?
이제 자세하게 설명하겠습니다.

3. 소스 코드 설명

앞 부분의 모듈 불러오는 것과 디렉토리 설정하는 부분은 이미 잘 알고 계시죠?
이번 챕터에서 중요한 부분들을 발췌해서 설명하겠습니다.
소스코드에서 Step 4 부분에서 아주 중요한 내용이 있습니다.

```
55  #Step 4. 웹사이트 접속 후 해당 메뉴로 이동합니다.
56  s = Service("c:/py_temp/chromedriver.exe")
57  driver = webdriver.Chrome(service=s)
58
59  query_url='https://www.coupang.com/'
60  driver.get(query_url)
61  time.sleep(5)
62
63  # Access Denied 메시지가 나오면 아래코드로 쿠키를 삭제한다
64  driver.delete_all_cookies()
65  time.sleep(2)
```

위 코드에서 56번 행에서 61번 행까지는 다른 챕터에서 많이 본 크롬 드라이버를 설정하고 쿠팡 웹사이트에 접속하는 부분입니다.
그런데 셀레니움과 크롬 드라이버로 쿠팡 사이트에 그냥 접속을 하면 아래와 같은 화면이 뜨면서 오류가 발생합니다.



이 부분을 해결하기 위해서 위 소스코드에서 64번행과 65번 행을 사용하여 쿠키를 삭제해 주어야 합니다.
이 부분을 수행하지 않으면 셀레니움과 크롬 드라이버로 쿠팡에 접속할 수 없으니 꼭 수행해주세요.

쿠팡에 정상적으로 접속을 했으니 이제 카테고리를 클릭하고 식품 메뉴를 클릭해야 합니다.
이 메뉴들의 name 이나 id 값이 없어서 아래와 같이 xpath 값으로 지정하였습니다.

```
67  # 카테고리 -> 식품 버튼을 눌러 페이지를 엽니다
68  driver.find_element(By.XPATH , '//*[@id="header"]/div').click( )
69  driver.find_element(By.XPATH , '//*[@id="gnbAnalytics"]/ul[1]/li[4]/a').click( )
```

위 코드는 쿠팡 사이트에서 카테고리 메뉴를 누르고 식품 카테고리를 찾아서 클릭하는 코드입니다. 카테고리 메뉴에 대한 ID나 NAME 값이 없어서 xpath 값을 사용하여 작업하고 있습니다.

xpath 값을 찾는 방법은 위 그림과 같이 오른쪽의 HTML 코드 부분에서 마우스 오른쪽 버튼을 클릭 -> 단축메뉴에서 Copy -> Copy Xpath 를 선택하면 해당 엘리먼트의 Xpath 값이 메모리로 복사가 됩니다. 특정 엘리먼트의 ID 값이나 Name 값이 없을 때 아주 요긴하게 사용되는 방법입니다.

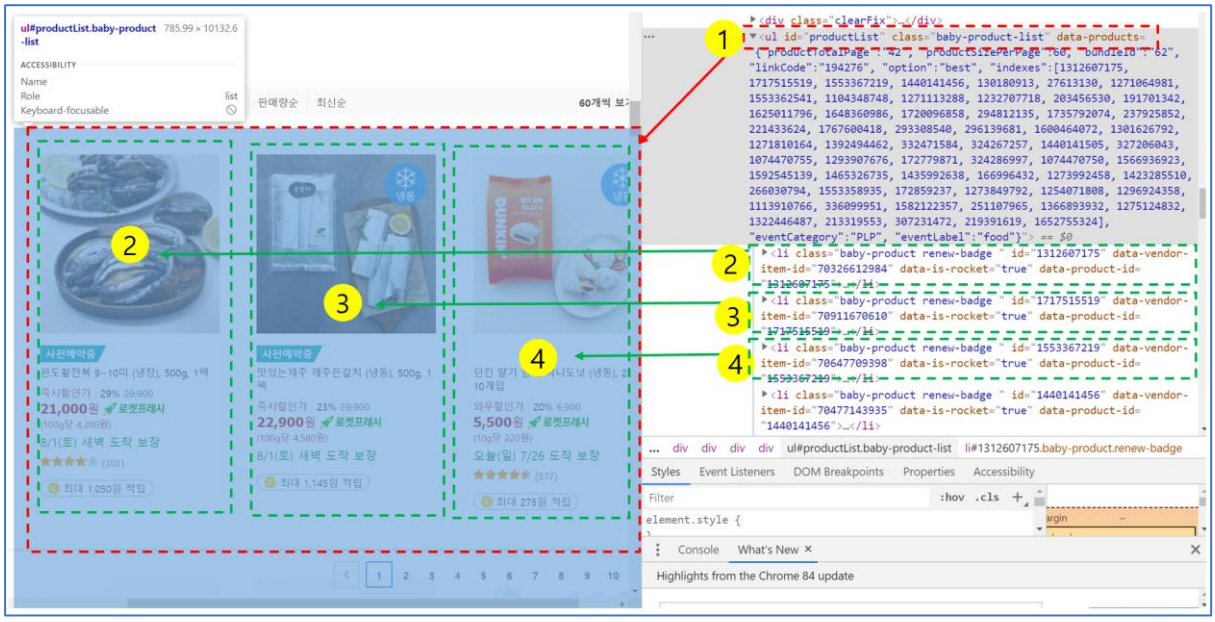
메뉴를 클릭하여 식품 카테고리에 접속했으니 이제 데이터를 수집해야겠죠?
쿠팡 사이트의 경우는 상단에는 제품 광고이고 중간 정도에 판매된 제품들의 현황이 보입니다.
그래서 화면을 아래로 스크롤 다운을 해야 해서 사용자 정의 함수를 아래와 같이 생성하였습니다.

```
71  #Step 5. 내용을 수집합니다
72  print("₩n")
73  print("===== 곧 수집된 결과를 출력합니다 ^^ ===== ")
74  print("₩n")
75
76  ranking2=[]          #제품의 판매순위 저장
77  title2=[]            #제품 정보 저장
78  p_price2=[]          #현재 판매가 저장
79  discount2 = []       #할인율 저장
80  sat_count2=[]        #상품평 수 저장
81
82  img_src2=[]    # 이미지 URL 저장변수
83  file_no = 0    # 이미지 파일 저장할 때 번호
84  count = 1       # 총 게시물 건수 카운트 변수
85
86  def scroll_down(driver):
87          driver.execute_script("window.scrollBy(0,1100);")
88          time.sleep(4)
```

위 코드에서 76번행부터 80번 행까지는 수집된 데이터를 저장할 리스트를 지정하는 부분입니다.
그리고 86번 행부터 88번 행은 현재 검색된 웹 페이지를 스크롤 다운해서 화면을 아래쪽으로 내려주는 사용자정의 함수를 만드는 부분입니다(사용자 정의 함수에 대해서는 이 책의 필수문법편을 참고하세요).
77번 행에 driver.execute_script( ) 함수를 사용한 것은 파이썬 코드에서 외부 OS 에 있는 특정 함수나 스크립트를 실행할 때 많이 사용하는 방법입니다. 즉 이 함수의 괄호안에 우리가 실행하고 싶은 OS 의 함수나 기능을 적으면 되는데 우리는 윈도의 마우스 스크롤 하는 기능을 실행하기 위해서 window.scrollTo( ) 함수를 사용한 것입니다.
window.scrollTo(x좌표, y좌표) 형식으로 사용하는데 예를 들어 window.scrollTo(0, 500) 이라고 적으면 500 픽셀만큼 아래로 화면을 이동시켜 줍니다.
이 함수와 비슷한 함수로 window.scrollBy(x좌표 , y좌표) 의 함수도 있는데 의미는 동일하지만 차이점은 window.scrollTo( ) 함수는 기준값이 절대 좌표이고 window.scrollBy( ) 함수는 기준값이 상대 좌표입니다. 화면 끝까지 이동하고 싶을 경우에는 document.body.scrollHeight 값을 사용하면 됩니다.
이 기능은 페이스북이나 인스타그램 , 유투브 등의 크롤러를 만들 때 반드시 사용하는 기능이므로 꼭 기억해 주세요.

```
90   for x in range(1,page_cnt + 1) :
91
92         for cc in range(1,7) :
93               scroll_down(driver)
94
95         html = driver.page_source
96         soup = BeautifulSoup(html, 'html.parser')
97
98         item_result = soup.find('ul','baby-product-list').find_all('li')
```

이제 데이터를 수집하겠습니다.
위 코드의 92번 – 93번 행은 화면을 아래로 천천히 스크롤 다운을 반복하는 부분입니다.
화면을 적절하게 스크롤 다운 한 후 95번 행에서 웹 페이지의 전체 소스코드를 가져온 후 96번 행에서 파싱을 수행하고 98번 행에서 제품 목록을 추출합니다.
아래 그림을 볼까요?

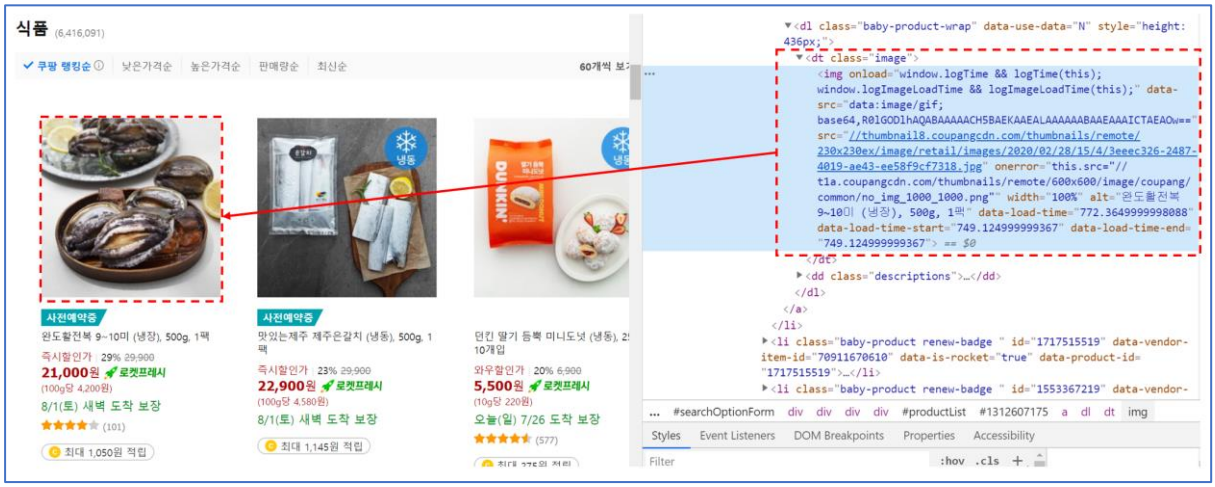


위 그림에서 왼쪽의 HTML 코드에서 1번을 보면 'ul' 태그에서 class='baby-product-list' 값을 가져온후 그 아래의 모든 'li' 태그를 가져오면 게시물 전체를 가져올 수 있다는 것이 확인됩니다. 그래서 위 코드에서 98번 행처럼 코드를 작성했습니다.

```
100     for li in item_result :
101         if cnt < count :
102             break
103
104         # 제품 이미지 다운로드 하기
105         try :
106             photo = li.find('dt','image').find('img')['src']
107         except AttributeError :
108             continue
109
110         file_no += 1
111         full_photo = 'https:' + photo
112         urllib.request.urlretrieve(full_photo,str(file_no)+'.jpg')
113         time.sleep(0.5)
```

위 코드는 상품 정보에서 상품의 이미지를 다운로드하는 부분입니다.
100번 행에서 item_result 변수에 1 페이지에 있는 60개 상품에 대한 목록이 들어 있는데 for 반복문을 사용해서 첫 번째부터 상품의 정보가 들어 있는 li 태그를 가져옵니다.
101번 – 102번 행에서 사용자가 요청한 건수보다 숫자가 크면 멈추고 그게 아닐 경우 105- 108번 행까지 제품 이미지의 URL 주소를 추출하여 photo 변수에 저장합니다.
그리고 추출한 URL 주소에 'https:' 를 붙여서 전체 URL 주소를 만들고 112번 행에서 이미지를 다운로드 받아서 지정된 폴더에 저장합니다.

아래 그림으로 제품 이미지가 저장된 태그와 속성값을 확인할 수 있습니다.



그리고 간혹 상품 설명에서 이미지가 없는 경우가 있어서 예외처리를 사용했습니다.

```
115         #제품 내용 추출하기
116         f = open(ff_name, 'a',encoding='UTF-8')
117         f.write("-------------------------------------------------"+"₩n")
118         print("-" *70)
119
120         ranking = count
121         print("1.판매순위:",ranking)
122         f.write('1.판매순위:'+ str(ranking) + "₩n")
```

위 코드의 116번 행은 수집 결과를 저장할 txt 형식의 파일을 지정하는 부분입니다.
제품의 판매 순위는 1번부터 시작하도록 지정했습니다.

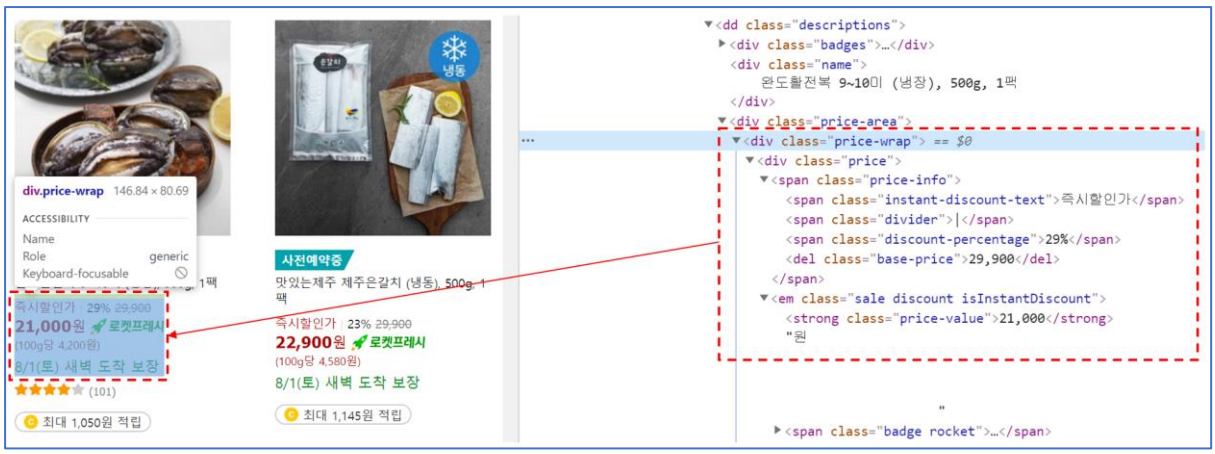
```
124         try :
125             t = li.find('div','name').get_text().replace("₩n","")
126         except :
127             title = '제품소개가 없습니다'
128             print(title.replace("₩n",""))
129             f.write('2.제품소개:'+ title + "₩n")
130       else :
131             title = t.replace("₩n","").strip()
131             print("2.제품소개:", title.replace("₩n","").strip())
132             f.write('2.제품소개:'+ title + "₩n")
```

위 코드는 제품 소개를 추출하는 부분입니다.
아래 그림을 보세요.

위 그림에서 'div' 태그에 class='name' 아래에 상품에 대한 설명이 있는 것이 확인됩니다.
그래서 위 코드의 125번행과 같이 코드를 작성했습니다.
그런데 간혹 상품에 대한 설명이 없는 부분이 있어서 예외처를 사용했습니다.

```
134             try :
135                 p_price = li.find('strong','price-value').get_text().replace("₩n","")
136             except :
137                 p_price = '0'
138                 print("3.판매가격:", p_price.replace("₩n",""))
139                 f.write('3.판매가격:'+ p_price + "₩n")
140             else :
141                 print("3.판매가격:", p_price.replace("₩n",""))
142                 f.write('3.판매가격:'+ p_price + "₩n")
143
144             try :
145                 discount = li.find('span','discount-percentage').get_text().replace("₩n","")
146             except   :
147                 discount = '0'
148                 print("4:할인률:", discount)
149                 f.write('4.할인율:'+ discount + "₩n")
150             else :
151                 print("4:할인률:", discount)
152                 f.write('4.할인율:'+ discount + "₩n")
```

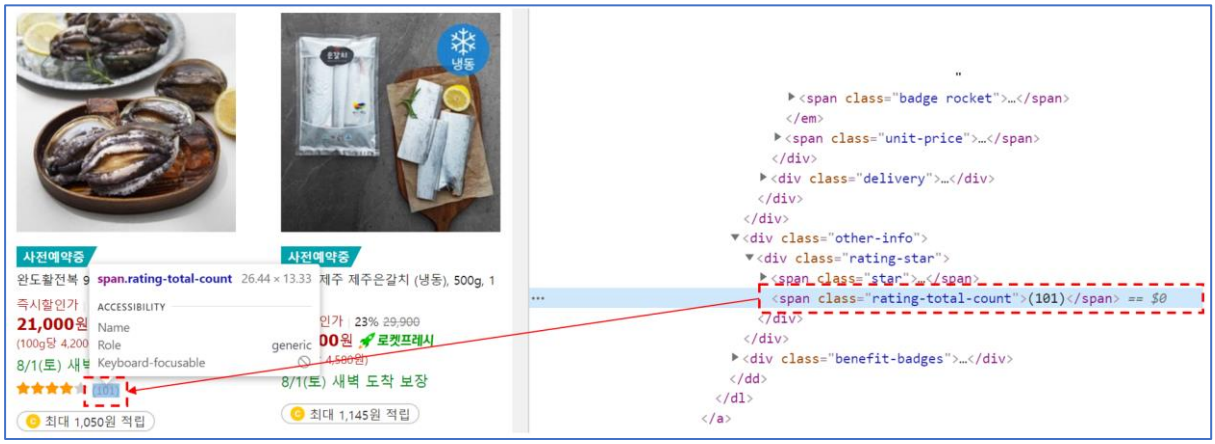
위 코드는 제품의 현재 판매가격과 할인율을 추출하는 코드입니다.
아래 그림을 보세요.



위 그림을 보면 제품의 판매 가격은 "strong" 태그에 class="price-value" 부분에 있는 것이 확인됩니다. 그래서 위의 소스코드에서 135번행과 같이 사용했습니다.
그리고 할인율은 "span" 태그에 class="discount-percentage" 부분에 있는 것이 확인되죠?
그래서 145번행과 같이 사용했습니다.
그리고 판매가격과 할인율 정보가 없는 상품이 있어서 예외처리를 사용했습니다.

```
154             try :
155                 sat_count_1 = li.find('span','rating-total-count').get_text()
156                 sat_count_2 = sat_count_1.replace("(","").replace(")","")
157             except   :
158                 sat_count_2='0'
159                 print('5.상품평 수: ',sat_count_2)
160                 f.write('5.상품평 수:'+ sat_count_2 + "₩n")
161             else :
162                 print('5.상품평 수:',sat_count_2)
163                 f.write('5.상품평 수:'+ sat_count_2 + "₩n")
164
165             print("-" *70)
166
167             f.close( )
168             time.sleep(0.5)
```

위 코드는 상품평 수를 수집하는 코드입니다. 아래 그림을 보세요.



위 그림을 보면 리뷰수가 "span" 태그에 class="rating-total-count" 부분에 있는 것이 확인됩니다. 그런데 리뷰가 괄호로 감싸져 있죠? 그래서 위 코드의 155번행과 156번행처럼 코드를 작성했습니다.

```
170             ranking2.append(ranking)
171             title2.append(title.replace("₩n",""))
172
173             p_price2.append(p_price.replace("₩n",""))
174             discount2.append(discount)
175
176             try :
177                 sat_count2.append(sat_count_2)
178             except IndexError :
179                 sat_count2.append(0)
180
181             count += 1
182         x += 1
183         try :
184             driver.find_element(By.LINK_TEXT, '%s' %x).click() # 다음 페이지번호 클릭
185         except :
186             break
```

위 코드는 수집된 데이터들을 미리 선언된 리스트에 추가하는 코드입니다.
그리고 184번행은 페이지번호를 바꾸는 부분입니다.

```
188 #step 6. csv , xls 형태로 저장하기
189 co_best_seller = pd.DataFrame()
190 co_best_seller['판매순위']=ranking2
191 co_best_seller['제품소개']=pd.Series(title2)
192 co_best_seller['제품판매가']=pd.Series(p_price2)
193 co_best_seller['할인율']=pd.Series(discount2)
194 co_best_seller['상품평수']=pd.Series(sat_count2)
195
196 # csv 형태로 저장하기
197 co_best_seller.to_csv(fc_name,encoding="utf-8-sig",index=False)
198
199 # 엑셀 형태로 저장하기
200 co_best_seller.to_excel(fx_name ,index=False , engine='openpyxl')
```

위 코드는 pandas 를 활용하여 수집된 데이터를 표 형태로 만든 후 csv , xls 형태의 파일로 저장하는 코드입니다 (pandas 에 대한 내용은 이 책의 필수문법편을 참고하세요)

이제 수집된 이미지와 저장된 xls 파일을 합성하는 작업을 진행하겠습니다.

```
215 #Step 7. xls 파일에 제품 이미지 삽입하기
216 import win32com.client as win32     #pywin32 , pypiwin32  설치
217 import win32api
218
219 excel = win32.gencache.EnsureDispatch('Excel.Application')
220 wb = excel.Workbooks.Open(fx_name)
221 sheet = wb.ActiveSheet
222 sheet.Columns(2).ColumnWidth = 30
223 row_cnt = cnt+1
224 sheet.Rows("2:%s" %row_cnt).RowHeight = 120
225
226 ws = wb.Sheets("Sheet1")
227 col_name2=[]
228 file_name2=[]
229
230 for a in range(2,cnt+2) :
231         col_name='B'+str(a)
232         col_name2.append(col_name)
233
234 for b in range(1,cnt+1) :
235         file_name=img_dir+'₩₩'+str(b)+'.jpg'
236         file_name2.append(file_name)
237
238 for i in range(0,cnt) :
239         rng = ws.Range(col_name2[i])
240         image = ws.Shapes.AddPicture(file_name2[i], False, True, rng.Left, rng.Top, 130, 100)
241         excel.Visible=True
242         excel.ActiveWorkbook.Save()
```

위 코드는 xls 형식의 파일을 열어서 앞에서 수집한 사진을 추가하는 코드입니다.
이 작업을 하기 위해 win32com 모듈을 사용하는데 216번 행과 217번 행에서 해당 모듈을 불러옵니다. 그리고 219번 행에서 엑셀 프로그램을 열고 220번 행에서 수집된 결과가 저장된 엑셀 파일을 엽니다.
222번 행과 224번 행은 사진을 저장할 컬럼의 행과 열의 크기를 지정하는 부분인데 이 부분은 사진의 크기에 따라 값이 달라지므로 사용하실 때 사이즈가 다를 경우 수정해서 사용하면 됩니다.
230번 행부터 232번 행까지는 사진을 추가할 "B" 컬럼 관련 설정을 하고 234번 행부터 236번행까지는 추가할 사진의 파일명을 설정합니다.

238번행부터 240번까지 엑셀 파일에 사진을 추가하고 241번행은 사진 추가 작업 후 결과를 화면에 보이게 설정하는 부분이며 242번행은 결과를 자동 저장하는 코드입니다.

여기까지 전체 코드에서 중요한 부분들을 설명했습니다.
나머지 코드들은 중복되는 내용이거나 이전 시간에 공부할 때 배운 코드이기 때문에 연습을 많이 해서 꼭 여러분의 실력으로 만드세요~

4. 연습 문제로 실력 굳히기

1. 아마존 닷컴 사이트에서 Best Seller 상품의 정보를 추출하는 크롤러를 만드는데 앞에서 배웠던 이미지를 다운로드 하는 방법을 사용하여 제품의 이미지까지 추출하여 엑셀 파일로 저장하도록 크롤러를 만드세요. (아래 예시 참고)

URL : https://www.amazon.com/bestsellers?ld=NSGoogle

[크롤러 실행 화면 예시]

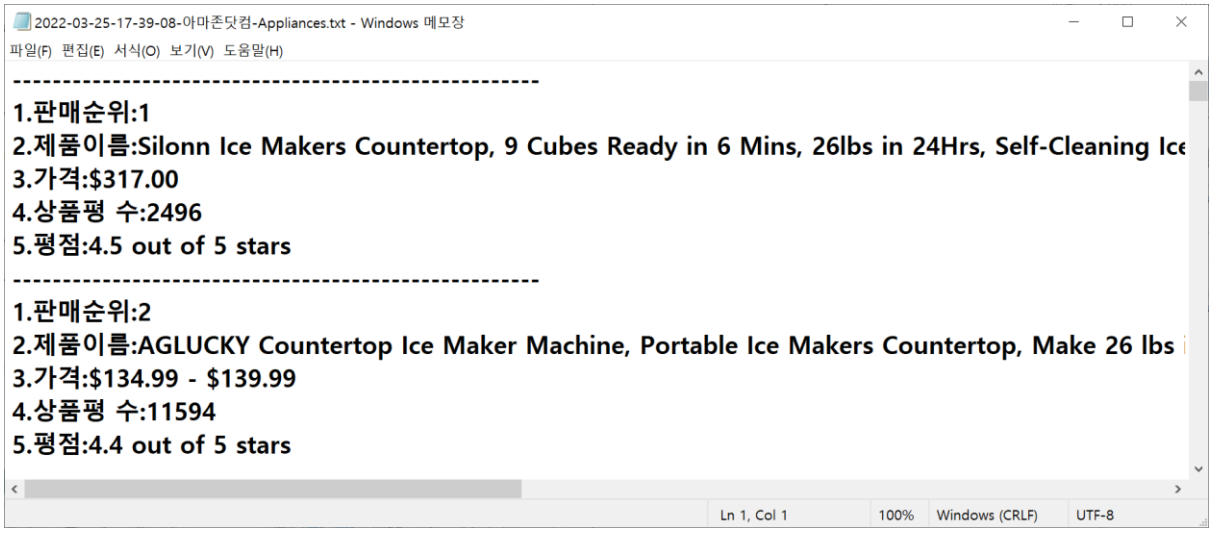
```
================================================================================
     아마존 닷컴의 분야별 Best Seller 상품 정보 추출하기
================================================================================

    1.Amazon Devices & Accessories    2.Amazon Launchpad          3.Appliances
    4.Apps & Games                    5.Arts, Crafts & Sewing     6.Audible Books & Originals
    7.Automotive                      8.Baby                      9.Beauty & Personal Care
    10.Books                          11.Camera & Photo Products   12.CDs & Vinyl


    1.위 분야 중에서 자료를 수집할 분야의 번호를  선택하세요: 3
    2.해당 분야에서 크롤링 할 건수는 몇건입니까?(1-100 건 사이 입력): 65
    3.파일을 저장할 폴더명만 쓰세요(예:c:₩py_temp₩):
```

[크롤러 실행 화면 예시]

```
1 페이지의 내용을 추출합니다~~
1번째 내용을 추출합니다~~
------------------------------------------------------------------------
1.판매순위: 1
2.제품이름: Silonn Ice Makers Countertop, 9 Cubes Ready in 6 Mins, 26lbs in 24Hrs, Self-Cleaning Ice Machine with Ice Scoop and Baske
t, 2 Sizes of Bullet Ice for Home Kitchen Office Bar Party
3.가격: $317.00
4.상품평 수: 2496
5.평점: 4.5 out of 5 stars
------------------------------------------------------------------------
2번째 내용을 추출합니다~~
------------------------------------------------------------------------
1.판매순위: 2
2.제품이름: AGLUCKY Countertop Ice Maker Machine, Portable Ice Makers Countertop, Make 26 lbs ice in 24 hrs,Ice Cube Rready in 6-8 Min
s with Ice Scoop and Basket (Black)
3.가격: $134.99 - $139.99
4.상품평 수: 11594
5.평점: 4.4 out of 5 stars
------------------------------------------------------------------------
3번째 내용을 추출합니다~~
------------------------------------------------------------------------
1.판매순위: 3
2.제품이름: 2-Pack Ice Machine Cleaner and Descaler 16 fl oz, Nickel Safe Descaler | Ice Maker Cleaner Compatible with All Major Brand
s (Scotsman, KitchenAid, Affresh) - Made in USA by Essential Values
3.가격: $19.99
4.상품평 수: 1138
5.평점: 4.5 out of 5 stars
```

(중간 내용은 분량이 많아 생략하겠습니다)

[txt 형식으로 저장된 결과 화면 예시]



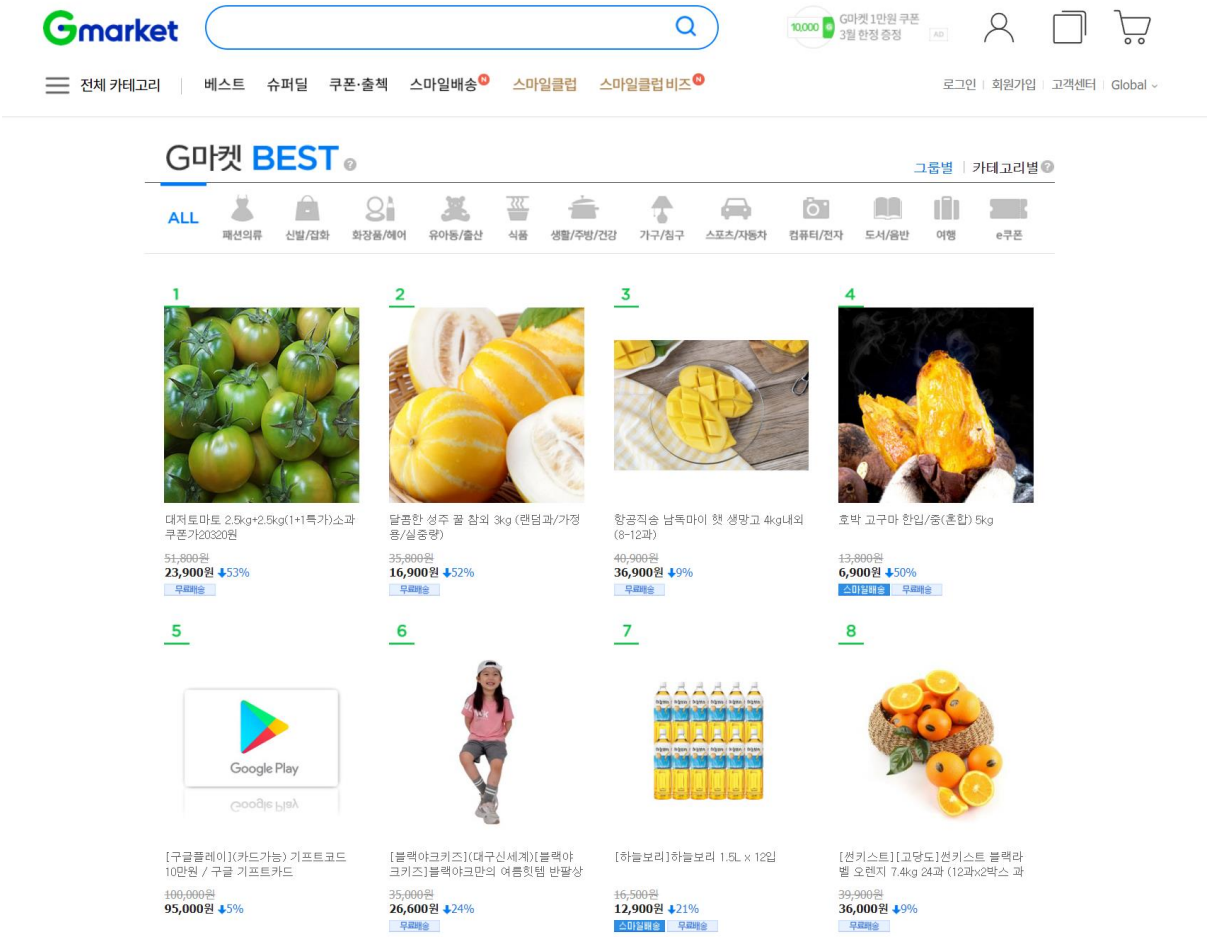
[xls 형식으로 저장된 결과 화면 예시]

	A	B	C	D	E	F	G
1		판매순위	제품소개	판매가격	상품평 갯수	상품평점	
2	0	1	Silonn Ice Makers Countertop, 9 Cu	$317.00	2496	4.5 out of 5 stars	
3	1	2	AGLUCKY Countertop Ice Maker Ma	$134.99 - $13	11594	4.4 out of 5 stars	
4	2	3	2-Pack Ice Machine Cleaner and De	$19.99	1138	4.5 out of 5 stars	

2. 지마켓 (www.gmarket.com) 사이트에서 Best Seller 상품의 정보를 수집하는 크롤러를 작성하세요. (아래의 예시 화면을 참고하세요)

URL 주소 : http://corners.gmarket.co.kr/Bestsellers

위 주소의 웹페이지에 접속하면 아래와 같이 전체 상품의 Best Seller 목록이 보입니다.



위 그림에서 특정 카테고리를 선택하지 말고 전체 카테고리에서 Best Seller 상품의 정보를 추출하는 크롤러를 만들면 됩니다.
위 그림에서 추출할 정보는 1.판매순위 / 2.제품소개 / 3.원래가격 / 4.판매가격 / 5할인율을 수집하면 됩니다.

[크롤러 실행 화면 예시]

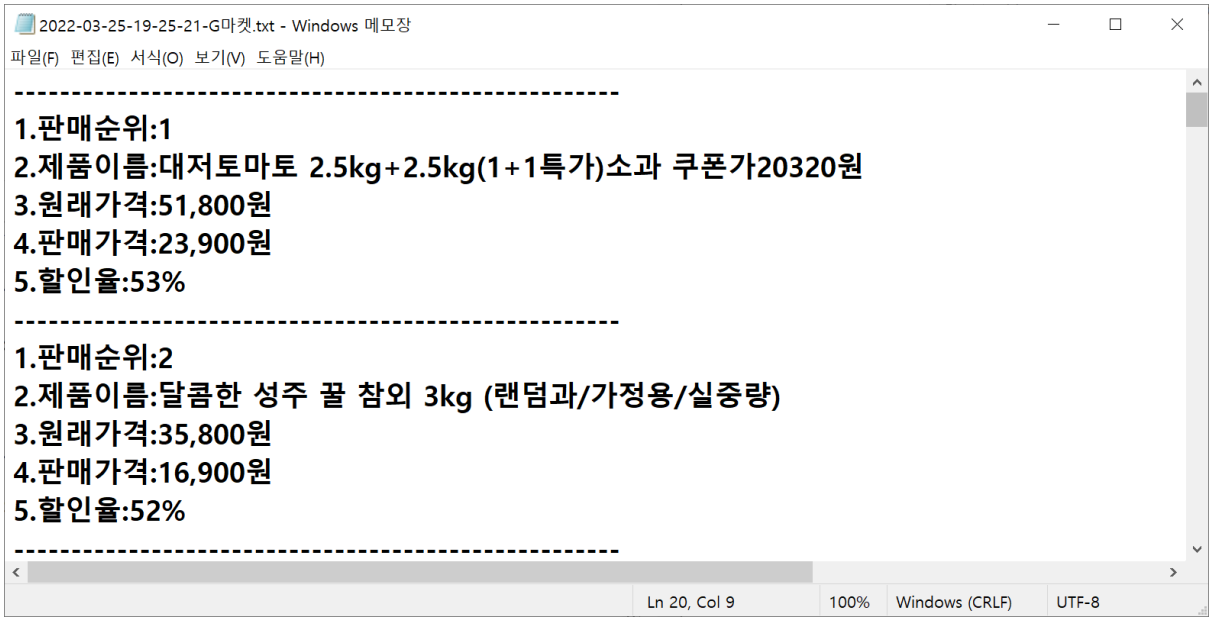
```
================================================================================
연습문제 : 지마켓 Best Seller 상품 정보 추출하기
================================================================================
1.크롤링 할 건수는 몇건입니까?(최대 200건): 25
2.파일을 저장할 폴더명만 쓰세요(예:c:₩py_temp₩):c:₩py_temp₩
```

[크롤링 과정 예시]

```
--------------------------------------------------------------------------
1.판매순위: 1
2.제품이름: 대저토마토 2.5kg+2.5kg(1+1특가)소과 쿠폰가20320원
3.원래가격: 51,800원
4.판매가격: 23,900원
5.할인율: 53%


--------------------------------------------------------------------------
1.판매순위: 2
2.제품이름: 달콤한 성주 꿀 참외 3kg (랜덤과/가정용/실중량)
3.원래가격: 35,800원
4.판매가격: 16,900원
5.할인율: 52%


--------------------------------------------------------------------------
1.판매순위: 3
2.제품이름: 항공직송 남독마이 햇 생망고 4kg내외(8-12과)
3.원래가격: 40,900원
4.판매가격: 36,900원
5.할인율: 9%
```

(내용은 많은데 지면 관계상 일부 내용만 보여 드립니다)

[txt 형식의 파일로 저장된 예시]



[xls 형식의 파일로 저장된 예시]

	A	B	C	D	E	F
1		판매순위	제품소개	원래가격	판매가격	할인율
2	0	1	 대저토마토 2.5kg+2.5kg(1+1특가)소과	51,800원	23,900원	53%
3	1	2	 달콤한 성주 꿀 참외 3kg (랜덤과/가정용	35,800원	16,900원	52%
4	2	3	 항공직송 남독마이 햇 생망고 4kg내외((	40,900원	36,900원	9%

지금까지 우리는 웹 크롤링의 원리와 환경 설정하는 방법을 배웠고 자동 검색을 수행하면서 개발자 도구를 활용하여 엘리먼트 정보를 확인하는 방법을 배웠습니다.

그리고 다양한 사례들을 통해 메뉴를 선택하고 페이지를 변경하면서 다양한 형식의 데이터를 수집한 후 csv , xls,, txt 형식으로 저장하는 방법들을 배웠습니다.

이 책에서 언급하지 못했지만 아주 많이 활용되는 웹 크롤러로 네이버 카페 정보를 수집하는 것이나 공동인증서(공인인증서)를 사용하여 로그인 한 후 데이터를 수집해야 하는 다양한 사이트용 웹 크롤러들이 있는데 여러가지 이유들로 이 책에는 담지 못했습니다.

혹시라도 이런 웹 크롤러가 필요하신 분들은 저자(seojinsu@gmail.com)에게 연락을 교육이나 웹 크롤러 제작을 도와드릴 수 있습니다. 다만 이 경우 별도의 비용이 발생할 수 있습니다.

필요한 데이터를 확보할 수 있는 능력이 독자님에게 아주 큰 힘이 될 것입니다.
이 책의 내용을 꼭 열심히 연습해서 독자님의 능력으로 가져 가시길 응원하겠습니다.