동적 페이지의 웹 크롤링 실습

OLEI
한국기술교육대학교
온라인평생교육원

학습목표

- JavaScript에 의해서 생성되는 웹 콘텐츠의 실전 예제를 통해 스크래핑 할 수 있다.
- 브라우저를 기동시킨 상태와 브라우저를 기동시키지 않은 상태에서 웹 페이지의 렌더링 결과를 캡처할 수 있다.

학습내용

- 카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제
- 웹 페이지의 화면 캡처

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

1 Selenium을 활용한 웹 크롤링과 스크래핑을 고려해야 하는 경우

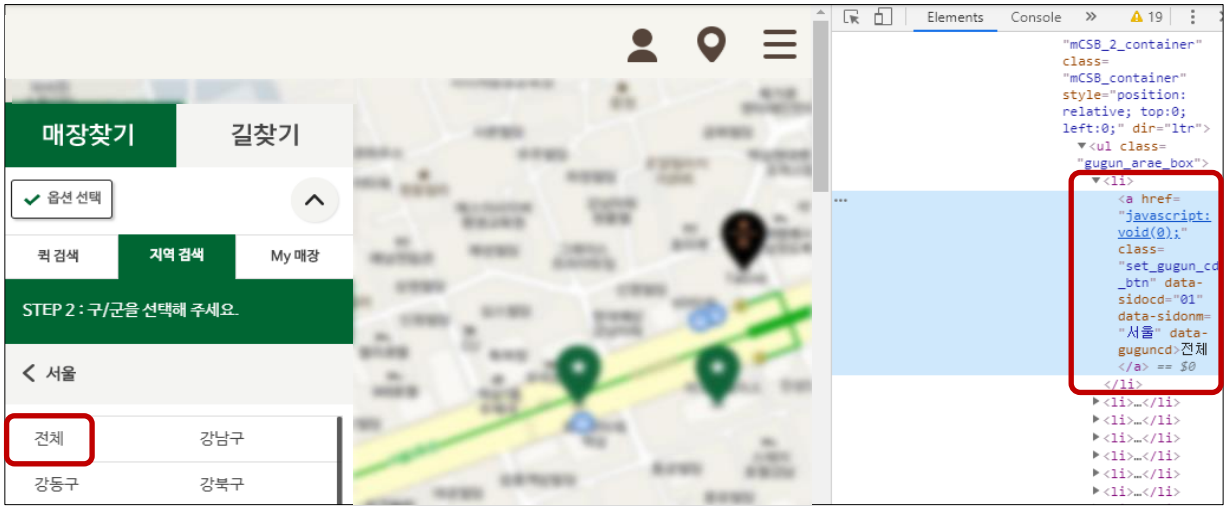
1. 화면에 렌더링된 웹 페이지의 내용을 서버로부터 전송된 소스 코드에서 찾을 수 없는 경우
2. 페이지 내의 링크를 클릭할 때 이동되는 페이지의 URL이 주소 필드에 출력되지 않는 경우
3. 웹 페이지를 크롤링하기 전에 로그인 과정을 거쳐서 인증 처리를 해야 하는 경우
4. 추출하려는 웹 페이지의 내용이 스크롤과 같은 이벤트가 발생해야 화면에 렌더링되는 경우
5. 버튼을 클릭해야 웹 페이지의 콘텐츠가 출력되는 경우

카페 및 서점 동적 웹페이지 크롤링 및 스크래핑 예제

2 관심 지역의 카페 매장 정보 읽어 오기

1 스크래핑 내용

1 전체의 WebElement 객체를 추출하여 클릭 이벤트 발생



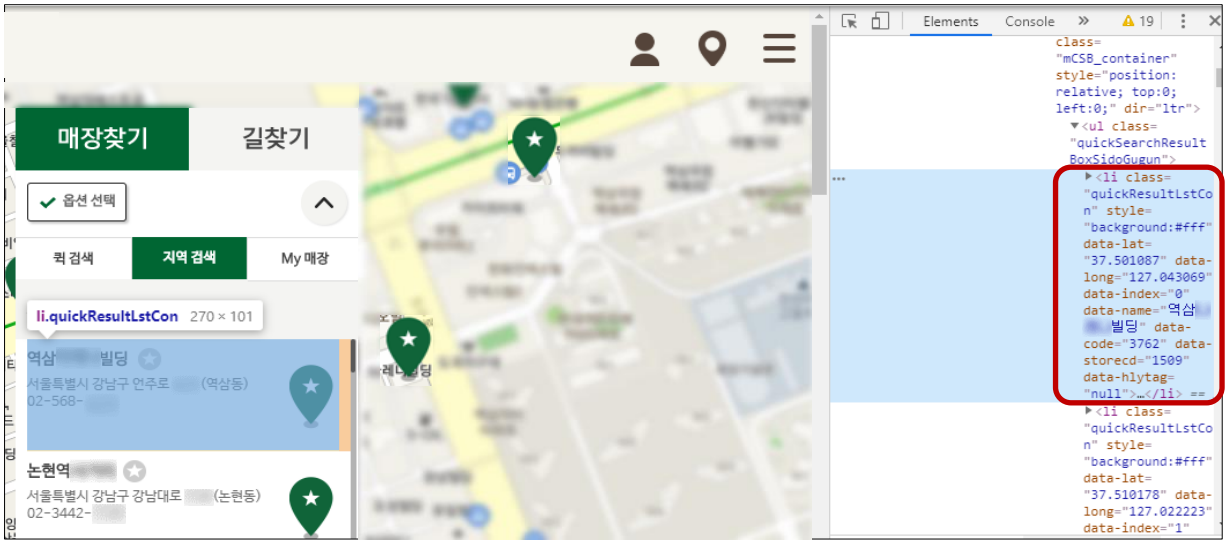

```
s_link = driver.find_element_by_css_selector("#mCSB_2_container > ul > li:nth-child(1) > a")
s_link.click()
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

2 관심 지역의 카페 매장 정보 읽어 오기

1 스크래핑 내용

2 매장 리스트가 출력되면 다음과 같은 CSS 선택자로 매장 리스트의 WebElement 객체를 리스트 객체로 추출




```
shopList =
driver.find_elements_by_css_selector("#mCSB_3_
container > ul > li")
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

2 관심 지역의 카페 매장 정보 읽어 오기

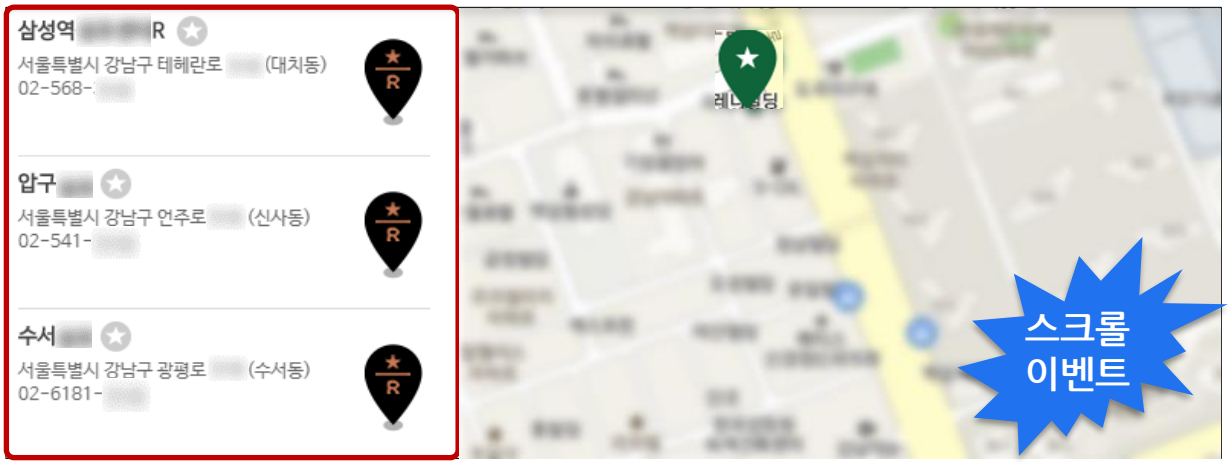
1 스크래핑 내용

3 해당 뷰 내에서 스크롤 이벤트 강제 발생

- 기본적으로 3개의 매장 정보 출력
  ➡ 해당 뷰의 스크롤을 아래로 내리면 그 다음 3개의 매장 정보 출력

- 화면에 렌더링된 매장 리스트의 세 번째 항목에서 JavaScript 코드 실행

```
driver.execute_script("var su = arguments[0];
var dom=document.querySelectorAll
                ( '#mCSB_3_container > ul > li' ) [su];
dom.scrollIntoView();", count)
```



카페 및 서점 동적 웹페이지 크롤링 및 스크래핑 예제

2 관심 지역의 카페 매장 정보 읽어 오기

2 소스

```
#파일명 : exam7_1.py
from selenium import webdriver
driver = webdriver.Chrome('C:/Temp/chromedriver')
driver.implicitly_wait(3)
driver.get("https://www.is******ks.co.kr/store/store_map.do")
import time
time.sleep(2)
loca = driver.find_element_by_class_name('loca_search')
loca.click()
time.sleep(2)
f_link = driver.find_element_by_css_selector("div.loca_step1_cont > ul > li:nth-child(1) > a")
f_link.click()
time.sleep(2)
s_link = driver.find_element_by_css_selector("#mCSB_2_container > ul > li:nth-child(1) > a")
s_link.click()
time.sleep(2)
shopList = driver.find_elements_by_css_selector("#mCSB_3_container > ul > li")
temp_list = []
time.sleep(3)
count = 0
total = len(shopList)
for shop in shopList :
    count += 1
    shoplat = shop.get_attribute("data-lat")
    shoplong = shop.get_attribute("data-long")
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

2 관심 지역의 카페 매장 정보 읽어 오기

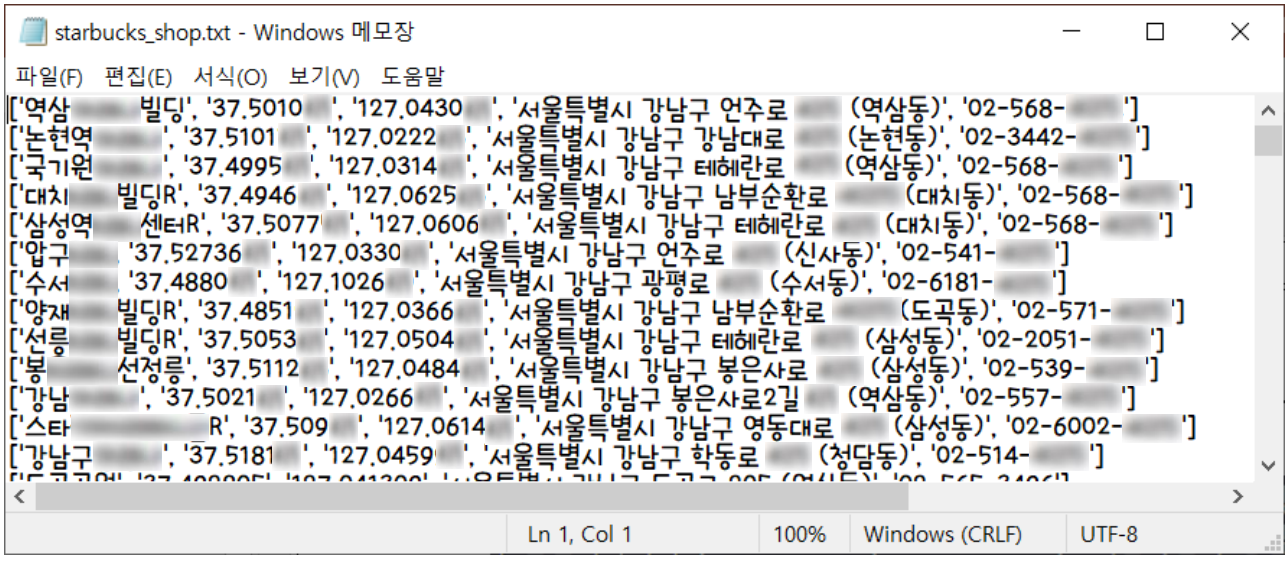
2 소스

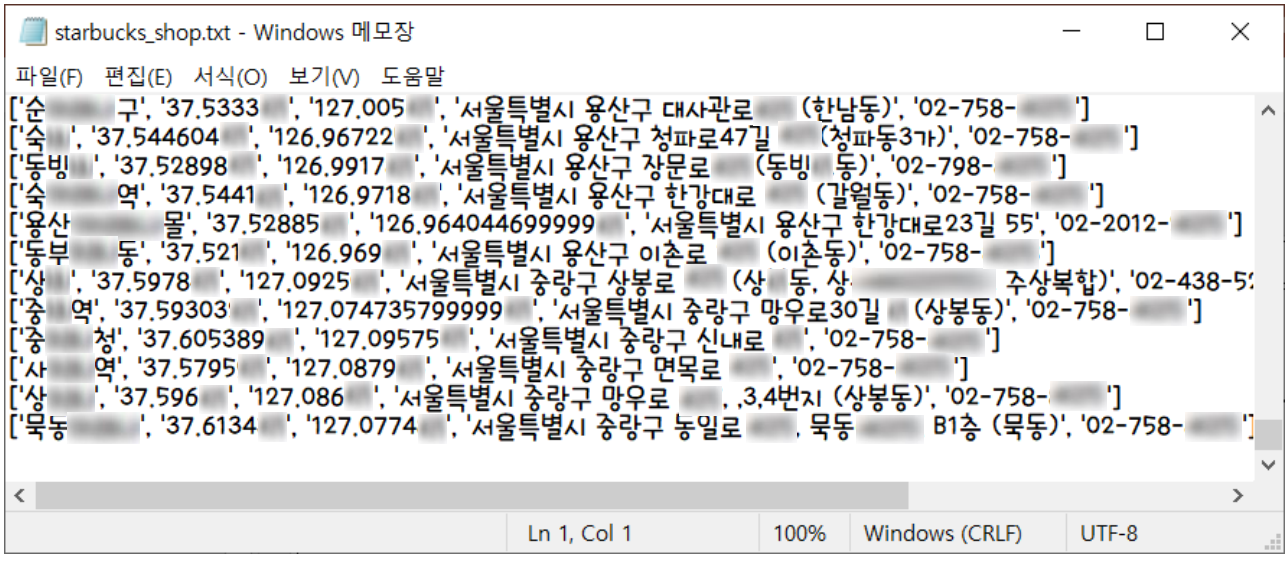
```
shopname = shop.find_element_by_tag_name("strong")
shopinfo = shop.find_element_by_tag_name("p")
splitinfo = shopinfo.text.split('₩n')
if(len(splitinfo) == 2):
   addr = splitinfo[0]
   phonenum = splitinfo[1]
temp_list.append([shopname.text, shoplat, shoplong, addr, phonenum])
if count != total and count % 3 == 0:
   driver.execute_script("var su = arguments[0]; var dom=
   document.querySelectorAll(
   '#mCSB_3_container > ul > li')[su]; dom.scrollIntoView();", count)

for item in temp_list :
    print(item)
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

2 관심 지역의 카페 매장 정보 읽어 오기

3 실행 결과



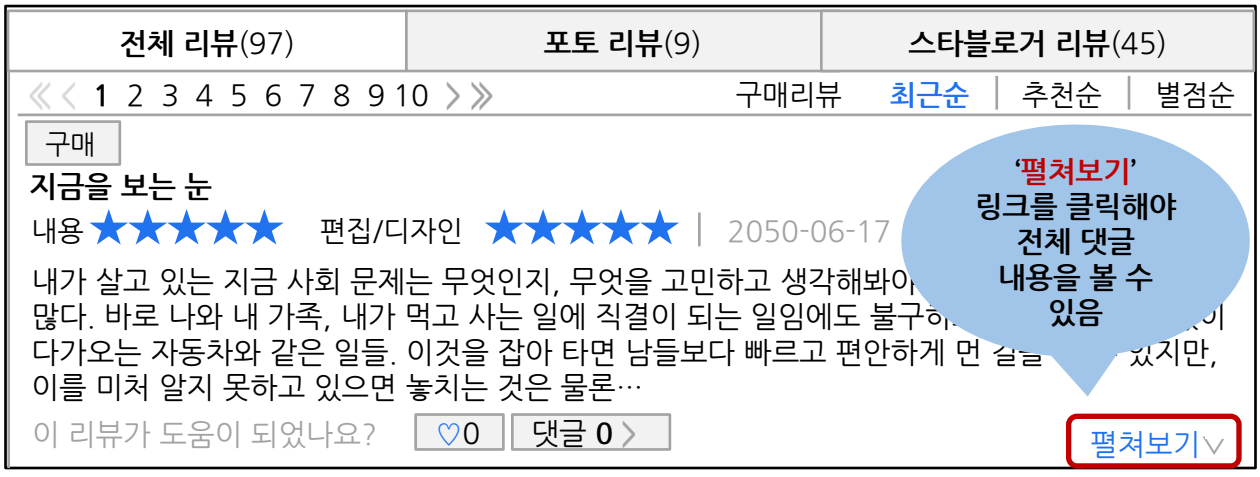


카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

1 도서 댓글 스크래핑

1 요약 댓글과 전체 댓글의 두 가지 타입 댓글로 구성



카페 및 서점 동적 웹페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

1 도서 댓글 스크래핑

2 '펼쳐보기'의 WebElement 객체들을 추출하여 각각 클릭 이벤트 발생

- 댓글 한 페이지당 5개의 댓글이 있음

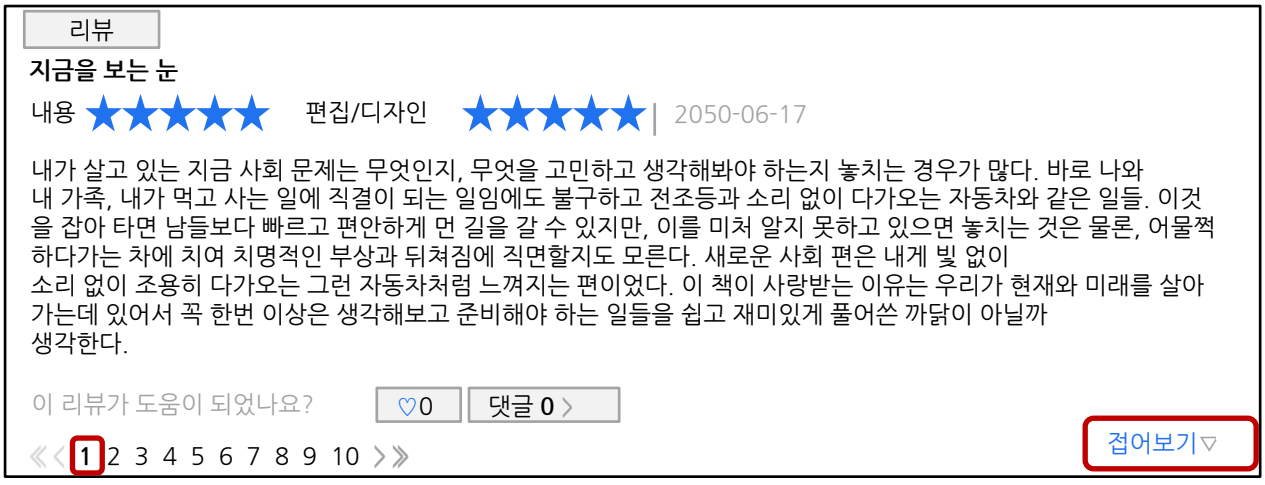
```
readLinks =
driver.find_elements_by_css_selector('#infoset_
reviewContentList  div.btn_halfMore > a') # 펼쳐보기
for readlink in readLinks :
    readlink.click()
    time.sleep(1)
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

1 도서 댓글 스크래핑

3 전체 댓글이 보여지면 댓글 추출



'펼쳐보기' 링크 문자열 → '접어보기' 링크 문자열로 변경 → 댓글 추출

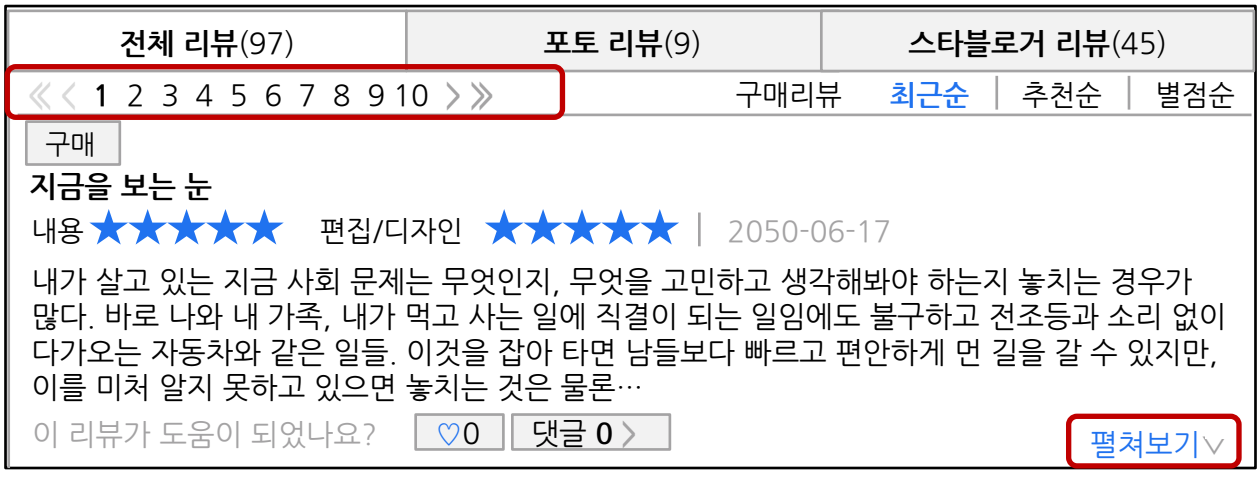
```
reviewList =
driver.find_elements_by_css_selector('#infoset_
reviewContentList div.reviewInfoBot.origin
div.review_cont')
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

1 도서 댓글 스크래핑

4 여러 페이지의 댓글 내용을 읽어오기 위해 **다음 페이지 링크 클릭**



카페 및 서점 동적 웹페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

1 도서 댓글 스크래핑

5 페이지 이동을 위해 각 페이지 링크 숫자에 적용되는 URL 규칙 파악

카페 및 서점 동적 웹페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

1 도서 댓글 스크래핑

5 페이지 이동을 위해 각 페이지 링크 숫자에 적용되는 URL 규칙 파악

```
#infoset_reviewContentList > div.review_sort.sortBot > div.review_sortLft > div > a:nth-child(4)
#infoset_reviewContentList > div.review_sort.sortBot > div.review_sortLft > div > a:nth-child(5)
#infoset_reviewContentList > div.review_sort.sortBot > div.review_sortLft > div > a:nth-child(11)
#infoset_reviewContentList > div.review_sort.sortBot > div.review_sortLft > div > a:nth-child(12)

#infoset_reviewContentList > div.review_sort.sortBot > div.review_sortLft > div > a.bgYUI.next
```

카페 및 서점 동적 웹페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

2 소스

```
#파일명 : exam7_2.py
from selenium import webdriver

driver = webdriver.Chrome('C:/Temp/chromedriver')
driver.implicitly_wait(3)

driver.get("http://www.y***4.com/Product/goods/40936880")
import time
time.sleep(2)
readLinks = driver.find_elements_by_css_selector
('#infoset_reviewContentList  div.btn_halfMore > a') # 펼쳐보기

for readlink in readLinks :
      readlink.click()
      time.sleep(1)
reviewList = driver.find_elements_by_css_selector
('#infoset_reviewContentList div.reviewInfoBot.origin div.review_cont')

temp_list = []
time.sleep(3)
for review in reviewList :
   temp_list.append(review.text)
stopFlag = False
while True :
  for n in range(4, 13) :
    linkurl = '#infoset_reviewContentList > div.review_sort.sortBot >
div.review_sortLft >       div > a:nth-child('+str(n)+')'
    linkNum = driver.find_element_by_css_selector(linkurl)
    linkNum.click()
    time.sleep(3)
readLinks = driver.find_elements_by_css_selector
                ('#infoset_reviewContentList  div.btn_halfMore > a')
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

2 소스

```
    for readlink in readLinks :
        #readlink.click()
        driver.execute_script("arguments[0].click();", readlink)
        time.sleep(3)

    reviewList = driver.find_elements_by_css_selector
    ('#infoset_reviewContentList div.reviewInfoBot.origin
    div.review_cont')
    time.sleep(2)

    for review in reviewList :
      temp_list.append(review.text)

    if len(reviewList) < 5 :
      stopFlag = True
      break

  if stopFlag == True :
    break

  nextPage = '#infoset_reviewContentList > div.review_sort.sortBot >
                                          div.review_sortLft > div >
a.bgYUI.next'
  linkNum = driver.find_element_by_css_selector(nextPage)
  linkNum.click()
  time.sleep(1)
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

2 소스

```
for item in temp_list :
    print(item)

wfile = open("c:/Temp/서점file.txt","w")
wfile.writelines(temp_list)
wfile.close()
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

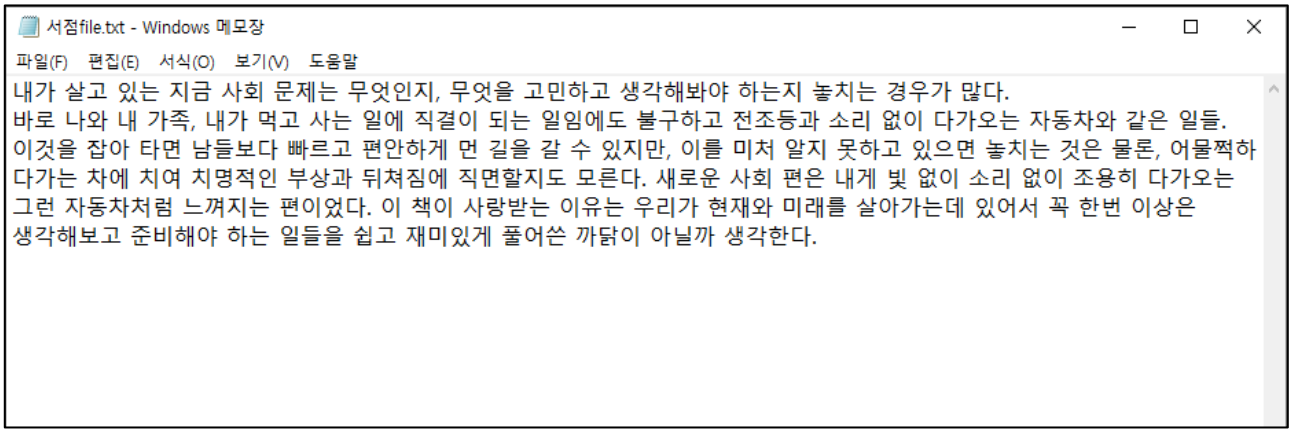
3 실행 결과 콘솔 화면

```
내가 살고 있는 지금 사회 문제는 무엇인지, 무엇을 고민하고 생각해봐야 하는지 놓치는 경우가 많다. 바로 나와 내 가족, 내가 먹고 사는 일에 직결이 되는 일임에도 불구하고 전조등과 소리 없이 다가오는 자동차와 같은 일들. 이것을 잡아 타면 남들보다 빠르고 편안하게 먼 길을 갈 수 있지만, 이를 미처 알지 못하고 있으면 놓치는 것은 물론, 어물쩍하다가는 차에 치여 치명적인 부상과 뒤쳐짐에 직면할지도 모른다. 새로운 사회 편은 내게 빛 없이 소리 없이 조용히 다가오는 그런 자동차처럼 느껴지는 편이었다. 이 책이 사랑받는 이유는 우리가 현재와 미래를 살아가는데 있어서 꼭 한번 이상은 생각해보고 준비해야 하는 일들을 쉽고 재미있게 풀어쓴 까닭이 아닐까 생각한다.
```

카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

3 서점 사이트의 도서 댓글 읽어 오기

4 파일 저장 화면



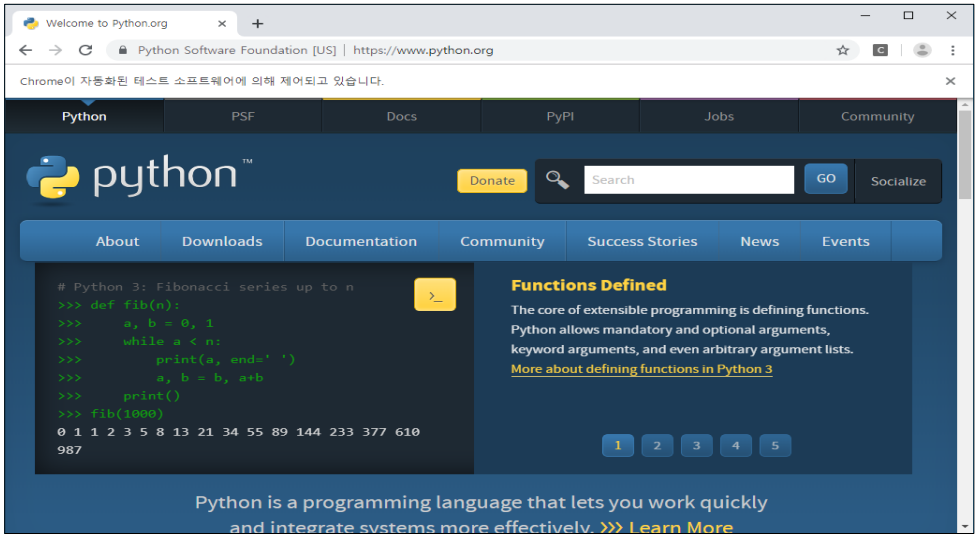
웹 페이지의 화면 캡처

1 웹 페이지 화면을 캡처하여 이미지 저장하기

1 Selenium에서 캡처 파일 이미지로 저장

- Selenium에 의해 기동된 크롬 브라우저의 페이지 렌더링 화면을 캡처하여 이미지 파일로 저장

```
driver.get_screenshot_as_file('c:/Temp/python_main.png')
```



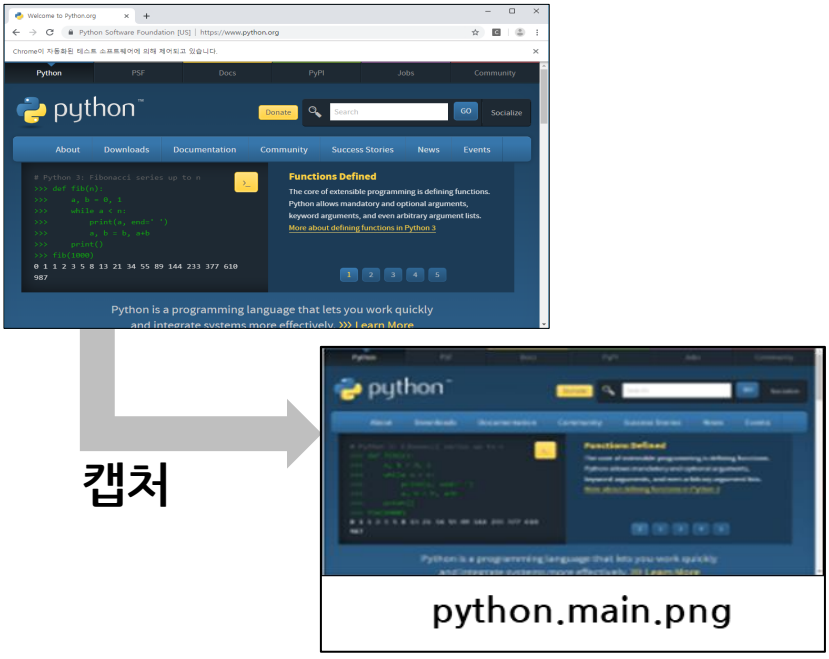
[출처 : www.python.org]

웹 페이지의 화면 캡처

1 웹 페이지 화면을 캡처하여 이미지 저장하기

2 소스

```
#파일명 : exam7_3.py
from selenium import webdriver

driver = webdriver.Chrome('C:/Temp/chromedriver')

driver.get('http://www.python.org/')
driver.implicitly_wait(3)
driver.get_screenshot_as_file('c:/Temp/python_main.png')
print('캡쳐 저장 완료')
import time
time.sleep(2)
driver.quit()
```



[출처 : www.python.org]

웹 페이지의 화면 캡처

1 웹 페이지 화면을 캡처하여 이미지 저장하기

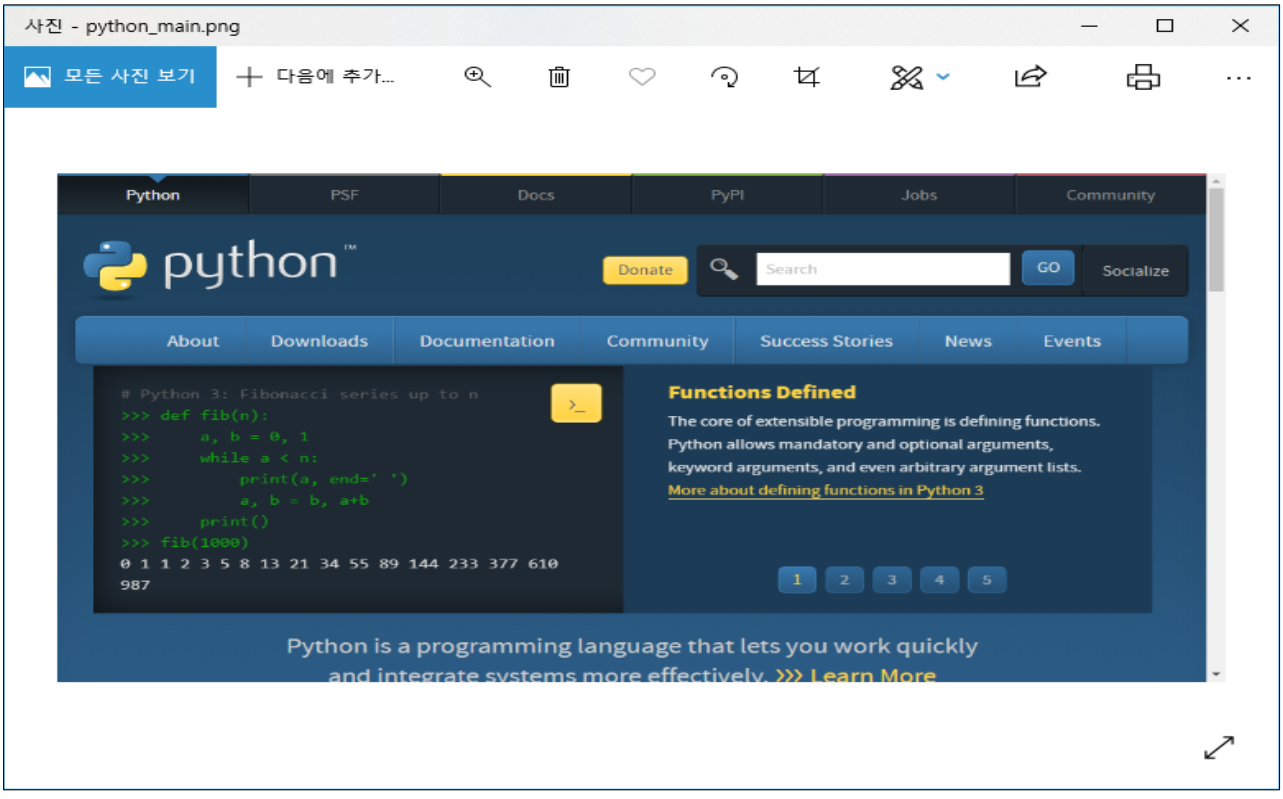
3 실행 결과 콘솔 화면

```
PS C:\example\myvscode> & C:/Users/UNICO/Anaconda3/python.exe c:/example/myvscode/
unit7/exam7_3.py

DevTools listening on ws://127.0.0.1:56345/devtools/browser/e754b8ab-3ff5-45fd-9f0
0-9e6323d5e8bc

KLIB_SelfTest return : KLR_OK
캡쳐 저장 완료
```

웹 페이지의 화면 캡처

1 웹 페이지 화면을 캡처하여 이미지 저장하기

4 실행 결과 이미지 파일 화면



[출처 : www.python.org]

웹 페이지의 화면 캡처

2 웹 브라우저를 기동시키지 않고 웹 페이지 화면을 캡처하여 이미지 저장하기

1 Headless 모드

Headless 모드	▪ 브라우저 창을 실제 운영체제의 '창'으로 띄우지 않고 웹 페이지의 화면을 그려주는 작업(렌더링)을 가상으로 진행해주는 방법 ▪ 실제 브라우저와 동일하게 동작하지만 창은 뜨지 않는 방식으로 동작 가능

윈도우 기준 크롬 59 버전

맥/리눅스 기준 크롬 60 버전

Headless Mode 정식 추가

- 브라우저가 최신이라면 크롬의 Headless모드를 쉽게 이용 가능

웹 페이지의 화면 캡처

2 웹 브라우저를 기동시키지 않고 웹 페이지 화면을 캡처하여 이미지 저장하기

1 Headless 모드

- 다음 코드 사용

```
options = webdriver.ChromeOptions()
options.add_argument('headless')
options.add_argument('window-size=1920x1080')
driver = webdriver.Chrome('C:/Temp/chromedriver',
options=options)
```

웹 페이지의 화면 캡처

2 웹 브라우저를 기동시키지 않고 웹 페이지 화면을 캡처하여 이미지 저장하기

2 소스

```
#파일명 : exam7_4.py
from selenium import webdriver
options = webdriver.ChromeOptions()
options.add_argument('headless')
options.add_argument('window-size=1920x1080')
driver = webdriver.Chrome('C:/Temp/chromedriver', options=options)

driver.get('http://www.python.org/')
driver.implicitly_wait(3)
driver.get_screenshot_as_file('c:/Temp/main_main_
headless.png')
print('캡처 저장 완료')
import time
time.sleep(2)
driver.quit()
```

브라우저 기동 없이

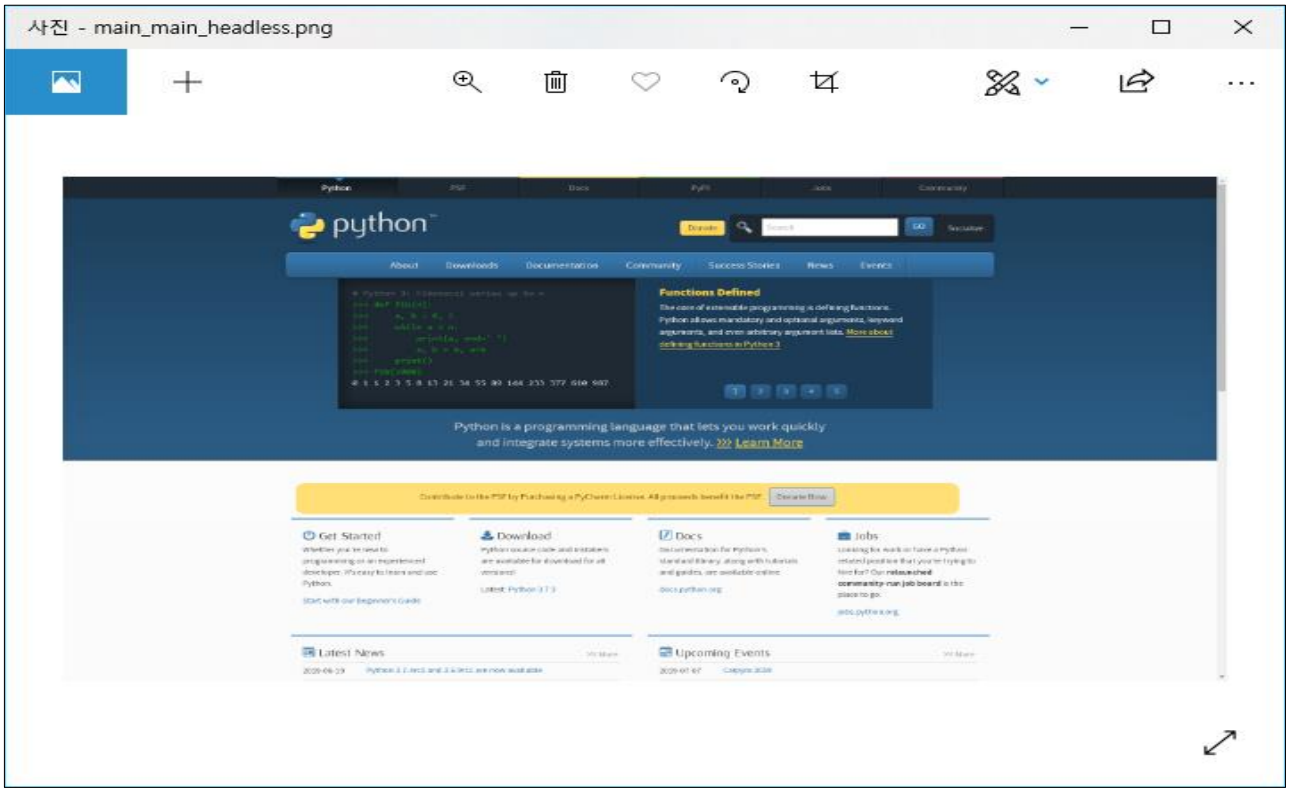
캡처

main_main_headless.png

웹 페이지의 화면 캡처

2 웹 브라우저를 기동시키지 않고 웹 페이지 화면을 캡처하여 이미지 저장하기

3 실행 결과 이미지 파일 화면



학습정리

1. 카페 및 서점 동적 웹 페이지 크롤링 및 스크래핑 예제

- 동적 웹 페이지 : 태그의 콘텐츠 즉, 웹 페이지의 내용을 브라우저에 함께 전송된 JavaScript 코드의 수행 결과로 생성시키는 페이지
- 태그에서 강제로 클릭 이벤트를 발생시킬 때 WebElement 객체의 click() 메서드를 호출

학습정리

2. 웹 페이지의 화면 캡처

- Selenium에 의해 제어되는 크롬 브라우저에 렌더링된 웹 페이지의 화면을 캡처하여 파일로 저장할 때 WebDrive 객체의 get_screenshot_as_file() 메서드 사용
- Headless 모드
  - 브라우저 창을 실제 운영체제의 '창'으로 띄우지 않고 웹 페이지의 화면을 그려주는 작업(렌더링)을 가상으로 진행해주는 방법
  - 실제 브라우저와 동일하게 동작하지만 창은 뜨지 않는 방식으로 동작