

파이썬을 활용한 업무자동화

8회차: 업무자동화로 일 편하게 하기

목차

8회차: 자동으로 웹 사이트 다루기 (심화)

- 복습
- 웹 자동화 심화: 마우스를 활용
 - ▶ 인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기
- 웹 자동화 심화: iframe 다루기
 - ▶ 중고나라 검색결과 가져오기

복습

01

지난 강의 때 뭐했지..?

■ 자동으로 웹 사이트 다루기 (기본)

- 뉴스 Top10 가져오기
- 네이버 검색결과 수집하기

```
# 크롬 브라우저를 띄우기 위한 웹드라이버를 가져옴
from selenium import webdriver

# 크롬 드라이버로 크롬 실행
driver = webdriver.Chrome('./chromedriver')

try:
    # 네이버 뉴스 페이지로 이동
    driver.get('http://news.naver.com/')
    # 현재 타이틀 출력
    print(driver.title)

    # 오른쪽 최근 뉴스 목록을 가진 div 태그를 가져옴
    title_div = driver.find_element_by_id('right.ranking_contents')

    # 위 div안의 li태그를 전부 가져옴
    news_list = title_div.find_elements_by_tag_name('li')

    # 가져온 태그들에 대해 반복문을 수행하면서 각각 문자열을 출력
    for news in news_list:
        print(news.text)

except Exception as e:
    # 에러 발생시 에러 메시지를 출력
    print(e)
finally:
    # 브라우저 종료
    driver.quit()
```

웹 자동화 심화: 마우스 활용

03

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

■ 마우스를 자동화

- 이전 강의에서 진행된 내용은 동적으로 변경될때 추약할 수 있음
- 방법은?
 - ▶ 좀 더 사람 친화적인 접근을 할 수 있다!
 - ▶ “어떤 요소에게 클릭을 전달해줘!”
 - ▶ “어떤 요소가 있는 위치를 클릭해줘!”
- ActionChains 라는 클래스를 활용하여 위 동작을 할 수 있음!

웹 자동화 심화: 마우스 활용

04

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

■ 마우스를 자동화

- 이전 강의에서 진행된 내용은 동적으로 변경될때 취약할 수 있음
- 바버르?

```
from selenium.webdriver.common.action_chains import ActionChains

actions = ActionChains(driver)
actions.move_to_element_with_offset(elem, 20, 20)
actions.click()
actions.move_to_element(elem)
actions.click()
actions.key_down('test')
actions.perform()
```

웹 자동화 심화: 마우스 활용

05

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

■ 자동화 단계 분석

1. Instagram 에 접속
2. 아이디와 비밀번호를 입력하거나 Facebook 버튼을 눌러서 접속
3. #패스트캠퍼스 검색
4. 나오는 인기 게시물을 하나 눌러서 좋아요 자동으로 누르기

웹 자동화 심화: 마우스 활용

06

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

■ 0단계: Selenium 설정

```
import time
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
# 마우스로 해당 게시글 클릭을 위해 ActionChains 클래스 추가
from selenium.webdriver.common.action_chains import ActionChains

driver = webdriver.Chrome('./chromedriver')
```


웹 자동화 심화: 마우스 활용

07

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

■ 1단계: Instagram 접속

1. Instagram 에 접속!

```
try:  
    # 인스타그램 사이트로 이동  
    driver.get('https://www.instagram.com/')  
    print(driver.title)
```

웹 자동화 심화: 마우스 활용

08

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

■ 2단계: Facebook 접속 or 로그인

1. Facebook 로그인 버튼을 클릭

- 1.이 경우에는 Facebook 페이지로 이동됨
2. Facebook 로그인 이후에 자동으로 Instagram이 로그인됨
3. 경우에 따라 권한 설정을 물어보기도함

2. Facebook 로그인이 아닌 경우에는 맨 아래 로그인 버튼을 클릭

1. 맨 아래 로그인 링크를 누르면 로그인 가능한 페이지로 이동
2. 이메일과 비밀번호를 입력받고 로그인가능

웹 자동화 심화: 마우스 활용

09

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

2단계: Facebook 접속 or 로그인

1. Facebook 로그인 버튼을 클릭

```
1. # facebook 로그인 (둘중 하나만 선택하여 진행)
   buttons = driver.find_elements_by_tag_name('button')
2. btn = buttons[0]
   btn.click()
```

2. Fa

```
3. elem = driver.find_element_by_id('email')
   elem.send_keys('alghost.lee@gmail.com')

1. elem = driver.find_element_by_id('pass')
   elem.send_keys('password')
   elem.send_keys(Keys.RETURN)
```

2. 이메일과 비밀번호를 입력받고 로그인가능

웹 자동화 심화: 마우스 활용

10

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

2단계: Facebook 접속 or 로그인

1. Facebook 로그인 버튼을 클릭

1. 이 경우에는 Facebook 페이지로 이동된

2. Fa

```
1. 이 경우에는 Facebook 페이지로 이동된
# 인스타그램 로그인 (둘중 하나만 선택하여 진행)
elem = driver.find_element_by_link_text('로그인')
elem.click()
```

3. 경

```
elem = driver.find_element_by_name('username')
elem.send_keys('alghost.lee@gmail.com')
```

2. Face

```
elem = driver.find_element_by_name('password')
elem.send_keys('password')
```

1. 맨

```
elem.send_keys(Keys.RETURN)
```

2. 이메일과 비밀번호를 입력받고 로그인가능

웹 자동화 심화: 마우스 활용

11

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

3단계: #패스트캠퍼스 검색

1. 검색창에 검색어를 입력
2. 인스타그램의 경우 검색어 입력 후 검색창 하단에 리스트가 나온 후에
검색이 가능 => 따라서 기다려야함
3. 리스트가 나온 후에 검색!

3단계

1. 검색

2. 인스

검색

3. 리스

```
# 페이지가 로딩 될 때까지 기다린다
```

```
time.sleep(2)
```

```
# 유일한 클래스명으로 검색
```

```
elem = driver.find_element_by_class_name('_eduze')
```

```
ac = ActionChains(driver)
```

```
# 마우스를 해당 위치로 이동
```

```
ac.move_to_element(elem)
```

```
# 해당 위치를 클릭
```

```
ac.click()
```

```
# 해당 위치를 클릭
```

```
# 두번 클릭하는 이유는 앱다운로드 팝업창을 삭제하기 위함
```

```
ac.click()
```

```
# 해당 위치에서 키 입력
```

```
ac.send_keys('#파이썬')
```

```
# ac에 저장된 작업들을 수행 (클릭, 키입력 등)
```

```
ac.perform()
```

```
# 검색어 리스트가 나올때까지 기다림
```

```
time.sleep(2)
```

```
# 리스트가 나온 후에 원하는 위치로 가서 클릭
```

```
# 해당 위치는 첫번째 해시태그 검색 결과
```

```
ac.reset_actions()
```

```
ac.move_by_offset(0, 50)
```

```
ac.click()
```

```
ac.perform()
```

```
# 검색 결과가 나올때까지 기다림
```

```
time.sleep(2)
```

은 후에

웹 자동화 심화: 마우스 활용

13

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

■ 4단계: 나오는 인기 게시물 좋아요 누르기

1. 인기 게시물 영역의 글을 전부 가져옴
2. 글을 하나씩 클릭
3. 팝업으로 뜬 창에서 하트 버튼을 누름
4. 만약 이미 좋아요를 누른 글인 경우엔 좋아요를 취소

웹 자동화 심화: 마우스 활용

14

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

4단계: 나오는 인기 게시물 좋아요 누르기

1. 인기 게시물 영역의 글을 전부 가져옴

2. 글을

3. 팝업

4. 만약

```
# _cmdpi 클래스를 가진 영역이 두개 있어, 이를 일단 다 가져옴
elems = driver.find_elements_by_class_name('_cmdpi')
# 첫번째 영역이 인기게시물 영역
elem = elems[0]

# 해당 요소 안의 모든 게시글을 가져옴
# 각 게시글은 여러 클래스를 가지고 있는데, _f2mse 를 사용
posts = elem.find_elements_by_class_name('_f2mse')
```

```
...
  <div class="_cmdpi">
    <div class="_70iju">
      <div class="_mck9w _gvoze _f2mse">...</div> == $0
      <div class="_mck9w _gvoze _f2mse">...</div>
      <div class="_mck9w _gvoze _f2mse">...</div>
    </div>
    <div class="_70iju">...</div>
    <div class="_70iju">...</div>
  </div>
</div>
<h2 class="_nhq1x">최근 사진</h2>
```



```
# 위에서 찾은 링크들에 대해 반복문을 돌면서 클릭
ac = ActionChains(driver)
for post in posts:
    # 해당 요소로 이동하여 클릭
    ac.reset_actions()
    ac.move_to_element(post)
    ac.click()
    ac.perform()

    # 게시글이 뜰때까지 기다림
    time.sleep(1)

    try:
        # 좋아요를 링크 텍스트로 찾아서 누름
        # 만약 이미 좋아요가 눌려진 글이라면 해당 링크 텍스트가 없음
        # 따라서 try를 이용하여 처리
        like_btn = driver.find_element_by_link_text('좋아요')
        like_btn.click()
        # 좋아요가 반영될때까지 기다림
        time.sleep(1)
    except:
        print('이미 좋아요를 누른 게시글입니다.')
        like_btn = driver.find_element_by_link_text('좋아요 취소')
        like_btn.click()

    # ESC 키를 눌러 게시글을 닫음
    ac.reset_actions()
    ac.send_keys(Keys.ESCAPE)
    ac.perform()

    time.sleep(1)
```

웹 자동화 심화: 마우스 활용

16

인스타그램에 특정 태그로 검색한 후 나오는 글에 좋아요 누르기

■ 자동화 종료

```
input()

except Exception as e:
    print(e)
finally:
    driver.quit()
```

웹 자동화 심화: iframe 다루기

17

중고나라 검색결과 가져오기

■ 자동화 전에 iframe 을 이해해보자

- iframe은 웹 문서안에 또 다른 웹 문서를 포함시키고자 할 때 쓰임
- 문제가 있음
 - ▶ 웹 드라이버는 포함시킨 또 다른 문서의 안을 바로 알 수가 없음
 - ▶ 크롬으로 볼때는 하위 요소인 것처럼 보이지만..
 - ▶ 실제로 코드로 접근하려면 별도 작업이 필요

웹 자동화 심화: iframe 다루기

18

중고나라 검색결과 가져오기

The screenshot displays a web browser window with the URL `cafe.naver.com/joonggonara`. The page content includes a sidebar with user statistics (e.g., 15,606,969 members) and a main area listing various bicycles for sale. The Chrome DevTools console is open, showing the HTML structure of the page. A red box highlights the `<iframe>` element with the following attributes: `name="cafe_check", id="cafe_check", width="0", height="0", frameborder="0", title="카페확인"`. The console also shows the `<html>` and `<body>` tags within the iframe's document.

```
<!DOCTYPE html PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html lang="ko">
  <#shadow-root (open)
  <head>...</head>
  <body class=
    <div id="photoviewer" class="photo_view" style="position:
      absolute; display: none; top: 50px; left: 100px; width: 200px;
      height: 200px; z-index: 1000; cursor: pointer;">...</div>
    <link rel="stylesheet" type="text/css" href="/common/storyphoto/
      original_viewer.css">
    <div id="u_skip">...</div>
    <h1 class="d-none">중고나라</h1>
    <div id="cafe-body-skin">...</div>
    <input type="hidden" id="questionct" value="block">
    <iframe name="cafe_check" id="cafe_check" width="0" height="0"
      frameborder="0" title="카페확인">
      <#document
      <html> == $0
        <#shadow-root (open)
        <head></head>
        <body></body>
      </html>
    </iframe>
    <form id="cafeApply_check" action="/CafeApplyCheck.nhn" method=
      "post" target="cafe_check" style="margin:0;padding:0;"> </form>
  </body>
</html>
```

웹 자동화 심화: iframe 다루기

19

중고나라 검색결과 가져오기

■ 자동화 단계 분석

1. 중고나라 홈페이지로 이동
2. 검색어 입력후 검색
3. 검색 결과에서 제목만 뽑아내어 출력

웹 자동화 심화: iframe 다루기

20

중고나라 검색결과 가져오기

0단계: Selenium 설정

```
import time
from selenium import webdriver
from selenium.webdriver.common.keys import Keys

# 브라우저를 띄움
driver = webdriver.Chrome('./chromedriver')
```

웹 자동화 심화: iframe 다루기

21

중고나라 검색결과 가져오기

■ 1단계: 중고나라 홈페이지로 이동

1. driver.get()으로 중고나라 홈페이지로 이동

- ▶ 주소: <http://cafe.naver.com/joonggonara>

```
try:
    # 중고나라 주소로 이동
    driver.get('http://cafe.naver.com/joonggonara')
```

웹 자동화 심화: iframe 다루기

22

중고나라 검색결과 가져오기

■ 2단계: 검색어 입력후 검색

1. 검색어를 입력하는 input 태그 찾아, 검색어 입력하여 검색

```
# 검색어를 입력할 input 태그
elem = driver.find_element_by_id('topLayerQueryInput')
elem.send_keys('자전거')

# 검색어를 입력한 후에 엔터 전송
elem.send_keys(Keys.RETURN)
```


웹 자동화 심화: iframe 다루기

24

중고나라 검색결과 가져오기

■ 2단계: 검색 결과에서 제목만 뽑아내어 출력

- 1.iframe 태그를 찾음
- 2.해당 frame으로 switch 함
- 3.그 후 원하는 내용을 가져와 출력

웹

중고

2단

1. ifr

2. 해

3. 그

```
# 검색 결과를 감싸고 있는 iframe 가져오기
iframe = driver.find_element_by_id('cafe_main')

# iframe 안으로 들어가기
driver.switch_to_frame(iframe)

# 원하는 내용을 가져오기 위해 상위의 name을 가진 form을 가져옴
form = driver.find_element_by_name('ArticleList')

# 최하위에 숨어있는 tr을 피하기 위해 xpath로
# 원하는 tr을 명시해서 가져옴
rows = form.find_elements_by_xpath('./table/tbody/tr')
for row in rows:
    try:
        # 제목이 a태그로 표현되어 있어, a 태그를 가져옴
        # tr 태그중에 게시글이 아닌 하나의 선을 표현한 태그가 있기 때문에
        # 해당 태그를 건너뛰기 위해 try를 시도
        # 선을 표현한 태그는 하위에 a태그를 가지고 있지 않기 때문에
        a_tag = row.find_element_by_tag_name('a')
        print(a_tag.text)
    except:
        # 데이터가 없는 경우 반복문을 스킵
        continue

# 브라우저가 바로 꺼지지 않도록 input()을 추가
input()
except Exception as e:
    print(e)
finally:
    driver.quit()
```

웹 자동화 심화: iframe 다루기

26

중고나라 검색결과 가져오기

■ 자동화 단계 분석

1. 중고나라 홈페이지로 이동
2. 검색어 입력후 검색
3. 검색 결과에서 제목만 뽑아내어 출력
4. 페이지를 이동하면서 결과를 출력하고 싶다면?

웹 자동화 심화: iframe 다루기

27

중고나라 검색결과 가져오기

■ 3단계: 페이지를 이동하면서 결과를 출력

1. 기존에 제목을 가져오는 코드를 반복문으로 변경

1. 몇번째 페이지까지인지 모르기 때문에 무한반복

2. 반복문 안에서 제목을 가져오는 코드 수행

3. 제목을 가져오는 코드 수행을 마친 후, 페이지 번호 링크를 가져옴

4. 해당 링크를 누른 후에 다시 반복문 수행

웹 자동화 심화: iframe 다루기

28

중고나라 검색결과 가져오기

3단계: 페이지를 이동하면서 결과를 출력

1 기존에 제목은 가져오는 코드를 반복문으로 변경

```
curr_page = 1
while True:
```

2

```
# 원하는 내용을 가져오기 위해 상위의 name을 가진 form을 가져옴
form = driver.find_element_by_name('ArticleList')
```

3

```
# 최하위에 숨어있는 tr을 피하기 위해 xpath로
# 원하는 tr을 명시해서 가져옴
rows = form.find_elements_by_xpath('./table/tbody/tr')
```

4

```
for row in rows:
    try:
        # 제목이 a태그로 표현되어 있어, a 태그를 가져옴
        # tr 태그중에 게시글이 아닌 하나의 선을 표현한 태그가 있기 때문에
        # 해당 태그를 건너뛰기 위해 try를 시도
        # 선을 표현한 태그는 하위에 a태그를 가지고 있지 않기 때문에
        a_tag = row.find_element_by_tag_name('a')
        print(a_tag.text)
    except:
        # 데이터가 없는 경우 반복문을 스킵
        continue
```

웹 자동화 심화: iframe 다루기

29

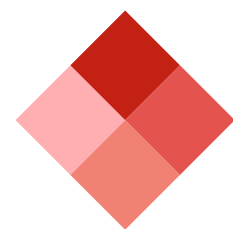
중고나라 검색결과 가져오기

```
for row in rows:
    try:
        # 제목이 a태그로 표현되어 있어, a 태그를 가져옴
        # tr 태그중에 게시글이 아닌 하나의 선을 표현한 태그가 있기 때문에
        # 해당 태그를 건너뛰기 위해 try를 시도
        # 선을 표현한 태그는 하위에 a태그를 가지고 있지 않기 때문에
        a_tag = row.find_element_by_tag_name('a')
        print(a_tag.text)
    except:
        # 데이터가 없는 경우 반복문을 스킵
        continue

# 페이지 네비게이션 정보가 있는 테이블을 가져옴
page_table = driver.find_element_by_class_name('Nnavi')
curr_page += 1
# 만약 현재 페이지가 10으로 나눠 떨어지는 페이지 (10, 20, 30, ...)라면 다음 클릭
if (curr_page-1) % 10 == 0:
    # 다음이라는 문자를 포함하는 링크를 가져옴
    page_link = page_table.find_element_by_partial_link_text('다음')
else:
    page_link = page_table.find_element_by_link_text(str(curr_page))

page_link.click()

time.sleep(1)
```



Good Bye

See you next time

Appendix

오늘 실습한 함수 정리

■ 사용된 함수 정리

함수명	설명	사용 클래스 변수
switch_to_frame	다른 iframe 으로 이동하는 함수	driver
switch_to_default_content	최초의 웹 문서로 돌아오는 함수	driver
move_to_element	해당 요소로 마우스를 이동하는 함수	ActionChains
move_by_offset	원하는 위치로 마우스를 이동하는 함수	ActionChains
move_to_element_with_offset	해당 요소에서 offset을 적용한 위치로 마우스를 이동하는 함수	ActionChains
click	해당 위치에서 클릭을 수행	ActionChains
perform	적용한 모든 작업을 수행 만약 이 함수를 호출하지 않으면 아무 동작하지 않음	ActionChains
reset_actions	저장된 모든 작업을 초기화	ActionChains