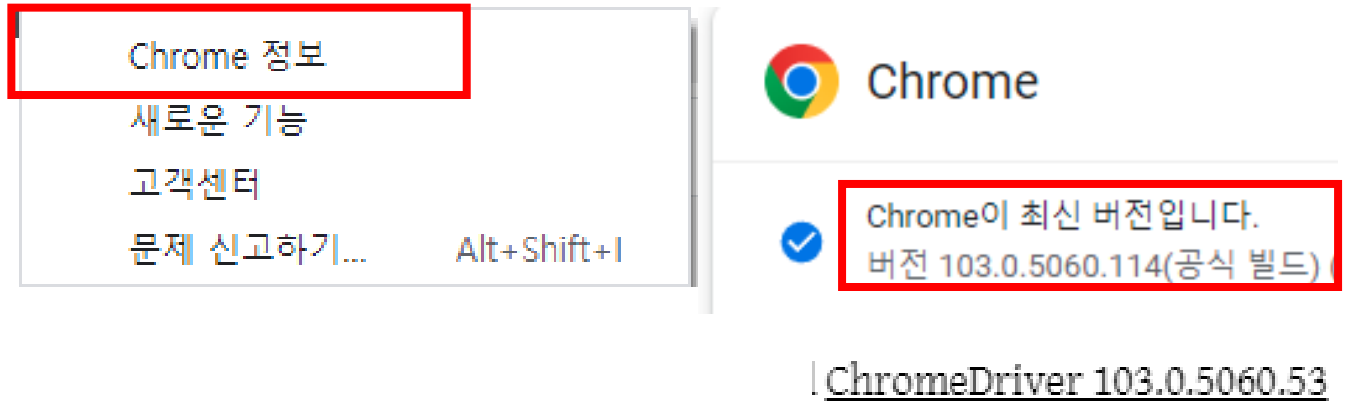
03_동적 페이지 크롤링

동적 페이지 크롤링

- Selenium
  - > https://www.selenium.dev
  - > 자동화를 목적으로 만들어진 다양한 브라우저와 언어를 지원하는 라이브러리
  - > Chromedriver
    - ~~크롬 브라우저 드라이버 다운로드가 필요함~~
    - ~~https://chromedriver.chromium.org/downloads~~
    - ~~사용하는 브라우저랑 동일한 버전 다운로드~~



동적 페이지 크롤링

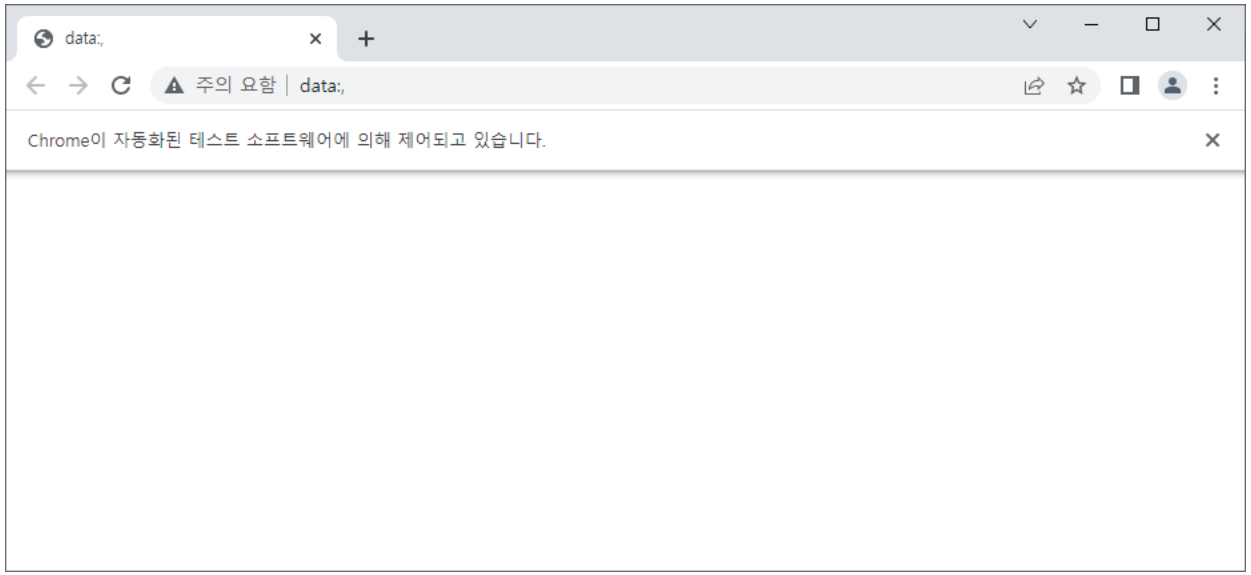
- Selenium

> pip install selenium

```
from selenium import webdriver
```

> 크롬 드라이버를 실행

```
driver = webdriver.Chrome()
```

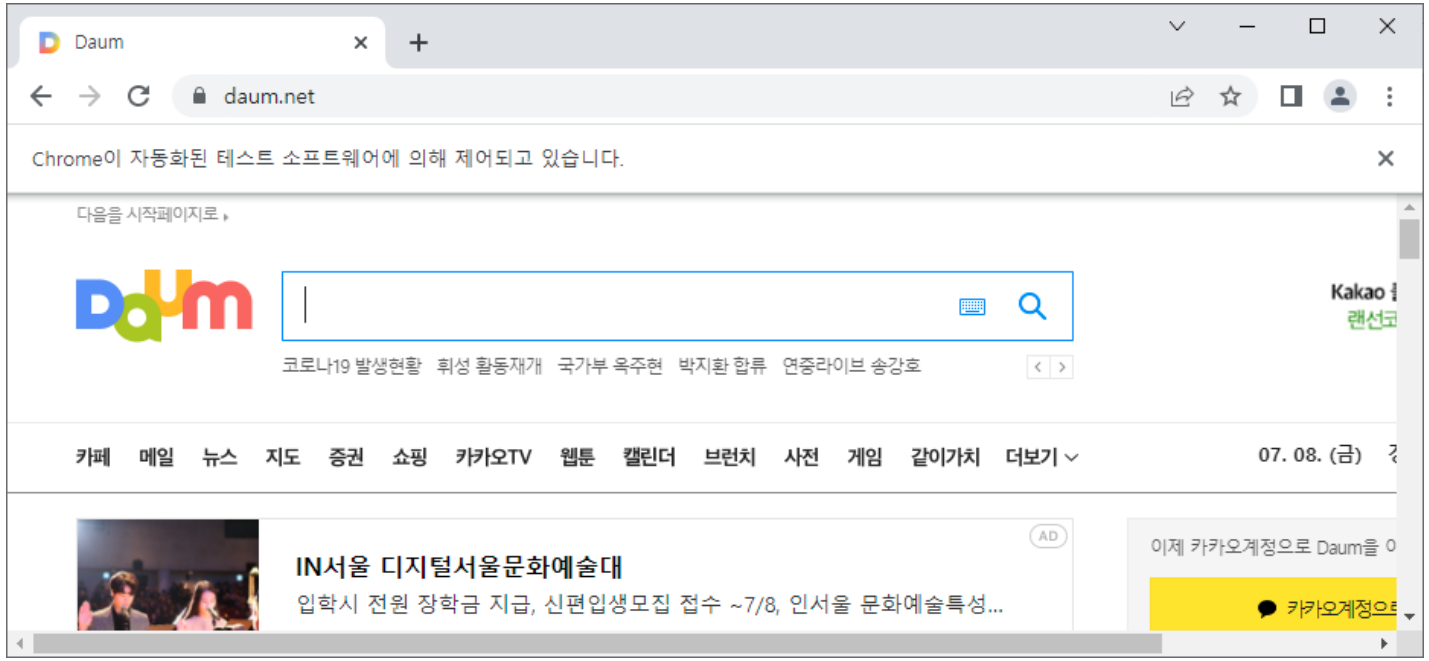


동적 페이지 크롤링

- Selenium

> 페이지 이동

```
driver.get('https://www.daum.net')
```



동적 페이지 크롤링

- Selenium

> 페이지 사이즈 조절

```
driver.set_window_size(200,600)
```

> 브라우저 스크롤 조절

```
driver.execute_script('window.scrollTo(200,300);')
```

> alert 다루기

```
driver.execute_script('alert('hello selenium!!!');')
```

```
alert = driver.switch_to.alert
alert.accept()
```

동적 페이지 크롤링

- Selenium

> 다양한 브라우저 조작 명령어

06

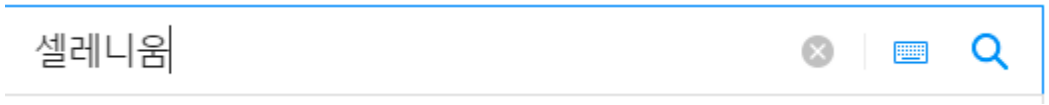
명령어	설명
add_cookie(cookie_dict)	쿠키값을 사전 형식으로 지정
back() / forward()	이전 페이지/ 다음페이지
close()	브라우저 닫기
current_url	현재 url
delete_all_cookies()	모든 쿠키 제거
delete_cookie(name)	name에 해당하는 쿠키 제거
execute(command, params)	브라우저 고유의 명령어 실행
execute_async_script(script, *args)	비동기 처리하는 자바스크립트를 실행
execute_script(script, *args)	동기 처리하는 자바스크립트를 실행
get(url)	웹 페이지를 읽어들임
get_cookie(name)	특정 쿠키 값을 추출
get_cookies()	모든 쿠키값을 사전 형식으로 추출
get_log(type)	로그를 추출(type: browser/driver/client/server)
get_screenshot_as_base64()	base64형식으로 스크린샷을 추출
get_screenshot_as_file(filename)	스크린샷을 파일로 저장
get_screenshot_as_png()	png형식으로 스키란샷의 바이너리를 추출
get_window_position(windowHandle ='current')	브라우저의 위치를 추출
get_window_size(windowHandle='current')	브라우저의 크기를 추출
implicitly_wait(sec)	최대 대기 시간을 초 단위로 지정해서 처리가 끈날때 까지 대기
quit()	드라이버를 종료 시켜 브라우저 닫기
save_screenshot(filename)	스크린샷 저장
set_page_load_timeout(time_to_wait)	페이지르르 읽는 타임아웃 시간을 지정
set_script_timeout(time_to_wait)	스크립트의 타임아웃 시간을 지정
set_window_position(x,y,windowHandle='current')	브라우저 위치를 지정
set_window_size(width, height, windowHandle='current')	브라우저 크기를 지정
title	현재 타이틀을 추출

동적 페이지 크롤링

- Selenium

> 태그 찾기

```
from selenium.webdriver.common.by import By
driver.find_element(By.CSS_SELECTOR, '#q').send_keys('셀레니움')
```




```
driver.find_element(By.CSS_SELECTOR,
                    '.inner_search > .ico_pctop.btn_search').click()
```

> 브라우저 종료

```
driver.close()
```

```
driver.quit()
```

동적 페이지 크롤링

- Selenium

> xpath를 사용하여 찾기

```
driver.find_element(By.XPATH, '//*[@id="q"]').send_keys('빅데이터')
```


```
driver.find_element(By.XPATH,
                    '//*[@id="daumSearch"]/fieldset/div/div/button[3]').click()
```

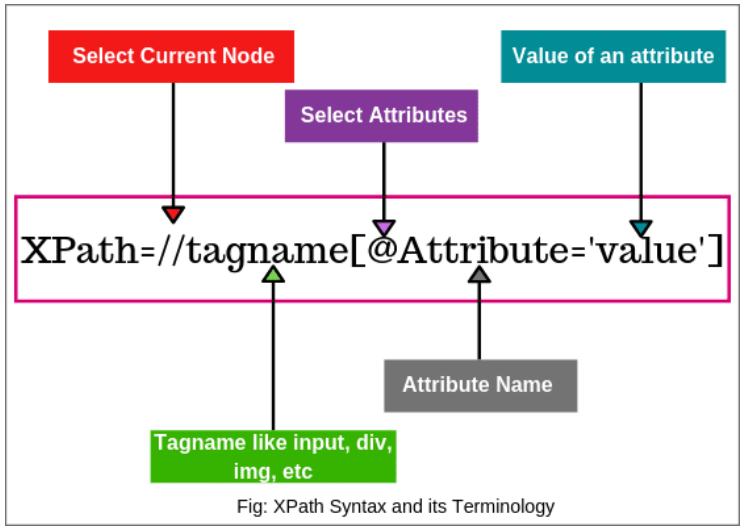


Fig: XPath Syntax and its Terminology

동적 페이지 크롤링

- Selenium

> 다양한 element 조작 명령어 & 키보드 명령어

메서드/ 속성	설명
clear()	글자를 지운다
click()	요소를 클릭
get_attribute(name)	요소 속성중 name에 해당하는 속성 값을 추출
is_displayed()	요소가 화면에 출력되는지 확인
is_enabled()	요소가 활성화돼 있는지 확인
is_selected()	체크박스 등의 요소가 선택된 상태인지 확인
screenshot(filename)	스크린샷
send_keys(value)	키를 입력
submit()	입력 양식을 전송
value_of_css_property(name)	name에 해당하는 css속성 값을 추출
id	id
location	요소의 위치
parent	부모요소
rect	크기와 위치 정보를 가진 사전자료형 리턴
screenshot_as_base64	스크린샷을 base64로 추출
screenshot_as_png	스크린샷을 png형식의 바이너리로 추출
size	요소의 크리
tag_name	태그 이름
text	요소의 내부 글자

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


Keys.ADD
```

```
ADD
ALT
ARROW_DOWN
ARROW_LEFT
ARROW_RIGHT
ARROW_UP
BACK_SPACE
BACKSPACE
CANCEL
CLEAR
```

동적 페이지 크롤링

- TED 사이트 크롤링
  - https://www.ted.com

> 브라우저를 실행하여 테드 사이트 열기

```
driver = webdriver.Chrome()
driver.get('https://www.ted.com/talks')
```

> 추천 텍스트 문구 가져오기

```
words = driver.find_elements(By.CSS_SELECTOR,
                             '#TopicsLaunchPad-Top-Default span.relative')
for word in words:
    print(word.text)
```

Out :

```
TED-Ed
Psychology
Leadership
Education
AI
Sleep
Mental Health
Business
Motivation
```

동적 페이지 크롤링

- TED 사이트 크롤링
  - https://www.ted.com

> 제목 데이터 가져오기

```
contents = driver.find_elements(By.CSS_SELECTOR, ".mb-1 > span")
len(contents)
```

Out : 24

> 제목 텍스트 데이터 가져오기

```
for content in contents:
    print(content.text)
```

Out :
```
How to create democracy - in an authoritarian country
The US vs. itself - and other top global risks in 2024
What would it take for generosity to go viral?
Is alternative meat the recipe for a healthier planet?
Enough red tape - we need to say yes to clean energy
Don't be a jerk to your barista - and other thoughts on frontline work
The vital data you flush down the toilet
```

동적 페이지 크롤링

- TED 사이트 크롤링
  - https://www.ted.com

> **셀렉트 박스 선택 후 데이터 가져오기**

  - 셀렉트 박스에서 한국어 선택

```
driver.find_element(By.CSS_SELECTOR,
                    '#filterBarScrollArea div > button > span > p').click()
```

```
driver.find_element(By.CSS_SELECTOR,
                    '#dropdown-menu > li:nth-child(6) > button > span').click()
```

```
driver.find_element(By.CSS_SELECTOR, '#k ul > li:nth-child(5) > button').click()
```

동적 페이지 크롤링

- TED 사이트 크롤링

> 링크 데이터 가져오기

  - 속성(href)값 가져오기


```
driver = webdriver.Chrome()
driver.get('https://naver.com')
login = driver.find_element(By.CSS_SELECTOR, '#account a').get_attribute('href')
driver.get(login)
```

Out : https://nid.naver.com/nidlogin.login?mode=form&url=https://www.naver.com/

동적 페이지 크롤링

- TED 사이트 크롤링
  - https://www.ted.com

> Headless

  - 브라우저를 화면에 띄우지 않고 크롤링하는 방법

```
options = webdriver.ChromeOptions()
options.add_argument('headless')
driver = webdriver.Chrome(options=options)
driver.get("https://www.ted.com/talks")
words = driver.find_elements(By.CSS_SELECTOR,
                             '#TopicsLaunchPad-Top-Default span.relative')
for word in words:
    print(word.text)
driver.quit()
```

- 크롬드라이버 옵션

> 기타 옵션들

```
#실행되는 브라우저 크기를 지정할 수 있습니다.
options.add_argument('--window-size= 200, 300')
#브라우저가 최대화된 상태로 실행됩니다.
options.add_argument('--start-maximized')
#브라우저가 풀스크린 모드(F11)로 실행됩니다.
options.add_argument('--start-fullscreen')
#브라우저에서 이미지 로딩을 하지 않습니다.
options.add_argument('--blink-settings=imagesEnabled=false')
#브라우저에 음소거 옵션을 적용합니다.
options.add_argument('--mute-audio')
#시크릿 모드의 브라우저가 실행됩니다.
options.add_argument('incognito')
```

동적 페이지 크롤링

- 스타벅스 사이트 크롤링
  - Selenium과 BeautifulSoup을 이용하여 크롤링 실습

> 필요 라이브러리 호출

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
```

> 스타벅스 Url 접속

```
driver = webdriver.Chrome()
driver.get('https://www.istarbucks.co.kr/store/store_map.do')
```

동적 페이지 크롤링

- 스타벅스 사이트 크롤링
  - Selenium과 BeautifulSoup을 이용하여 크롤링 실습

> **Class 명으로 버튼 찾기(지역 검색)**

```
loca = driver.find_element(By.CLASS_NAME, 'loca_search')
loca.click()
```

> **Class 명으로 버튼 찾기**
  - 전체 li 상위 태그를 찾아서 다중 검색으로 li 태그 선택

```
sido = driver.find_element(By.CLASS_NAME, 'sido_arae_box')
li = sido.find_elements(By.TAG_NAME, 'li')
li[16].click()
```

동적 페이지 크롤링

- 스타벅스 사이트 크롤링
  - Selenium과 BeautifulSoup을 이용하여 크롤링 실습

> **ID 명으로 버튼 찾기(시군구 버튼)**
  - 상위 태그를 ID로 검색
  - 하위 태그에서 li 태그 리스트로 받아오기

```
gugun = driver.find_element(By.ID, 'mCSB_2_container')
li = gugun.find_elements(By.TAG_NAME, 'li')
li[1].click()
```

동적 페이지 크롤링

- 스타벅스 사이트 크롤링
  - Selenium과 BeautifulSoup을 이용하여 크롤링 실습

> **page_source를 이용하여 html 소스 가져오기**
>
> - selenium을 통해 html 소스가 변경된 정보를 가져옴

```
source = driver.page_source
print(source[:200])
```

Out :

```
<html lang="ko"><head>


<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta charset="utf-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0,
maximum-scale=1.0,
```

동적 페이지 크롤링

- 스타벅스 사이트 크롤링
  - Selenium과 BeautifulSoup을 이용하여 크롤링 실습

> BeautifulSoup을 이용한 html parsing

```
html = BeautifulSoup(source,'html.parser')
```

> 검색 결과 찾기

```
search_list = html.select('ul.quickSearchResultBoxSidoGugun > li')
items = []
for item in search_list:
    items.append(item.find('p').text)
print(items[0])
driver.quit()
```

Out : 서울특별시 강남구 헌릉로 727 (세곡동)1522-3232