로그인이 필요한 웹페이지 크롤링 이해 및 실습

- javascript등 코드를 이해해야 해서 쉽지 않음
- 다른 방법이 있음

브라우저를 제어해서 크롤링을 하는 방법

> 브라우저에서 하는 작업을 프로그래밍으로 하도록 만들면 로그인도 쉽게 할 수 있음

Selenium

- Selenium: 웹을 테스트하기 위한 프레임워크
- 공식 홈페이지(http://www.seleniumhq.org/)
- Selenium with Python : http://selenium-python.readthedocs.io/index.html

사전준비 (Selenium 설치)

1. Selenium 인스톨: pip install selenium
2. 웹드라이버 인스톨: 웹 테스트 자동화를 위해 제공되는 툴(각 browser및 os 별로 존재)
    - selenium - 테스트 코드를 사용하여 브라우져에서의 액션을 테스트할 수 있게 해주는 툴
    - Firefox, chromedriver 등 각 브라우져마다 웹드라이버 다운로드 가능
    - https://sites.google.com/a/chromium.org/chromedriver/ (Chrome 브라우저용)
3. 설치 후, 다음 사이트에서 가장 최신 버전을 다운로드받아서, 덮어씌움
    - https://chromedriver.storage.googleapis.com/index.html
    - 윈도우: C:/dev_python/Webdriver/chromedriver.exe
    - 맥: /usr/local/Cellar/chromedriver/chromedriver

> chrome://version 으로 브라우저에서 확인 후, 버전에 맞는 드라이버를 설치해도 됨

Selenium 사용법

- selenium 로드

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

# 드라이버 생성
# chromedriver 설치된 경로를 정확히 기재해야 함
# chromedriver = 'C:/dev_python/Webdriver/chromedriver.exe' # 윈도우
chromedriver = '/usr/local/Cellar/chromedriver/chromedriver' # 맥
driver = webdriver.Chrome(chromedriver)
```

Selenium 사용법

- 크롤링 사이트 호출 및 확인

```
# 크롤링할 사이트 호출
driver.get("http://www.python.org")

# Selenium은 웹테스트를 위한 프레임워크로 다음과 같은 방식으로 웹테스트를 자동으로 진행함 (참고)
assert "Python" in driver.title
```

Selenium 사용법

- 브라우저 컨트롤
  - clear(): input 텍스트 초기화 하기
  - send_keys(키워드): 키보드 입력값 전달하기
    - Keys.RETURN - 엔터키
    - dir(Keys) 로 키에 대응되는 이름 찾기

```
# input 텍스트 초기화
elem.clear()

# 키 이벤트 전송
elem.send_keys("python")

# 엔터 입력
elem.send_keys(Keys.RETURN)
```

Selenium 사용법

- assert로 driver.page_source 에서 특정 키워드 확인하기
- time.sleep() 함수로 일정 시간 브라우저 내용 확인할 수 있도록 하기
- driver.quit() 함수로 브라우저 끝내기

```
assert "No results found." not in driver.page_source

# 명시적으로 일정시간을 기다릴 수 있음 (10초 기다림)
time.sleep(10)

# 크롬 브라우저 닫기 가능함
driver.quit()
```

Selenium 사용법

- 데이터 가져오기 주요 함수
  - find_element_by_id
  - find_element_by_name
  - find_element_by_tag_name
  - find_element_by_class_name
  - find_element_by_css_selector
  - find_element_by_xpath (XPATH 문법 이해 필요)

Selenium 사용법

- 데이터 가져오기 주요 함수
  - find_element_by_name(): 최초 발견한 name으로 가져오기
  - find_elements_by_name(): name이 동일한 모든 리스트를 가져오기

```
# <input id="id-search-field" name="q" class="" 검색창 name으로 검색하기
# 태그 name으로 특정한 태그를 찾을 수 있음
elem = driver.find_element_by_name("q")
```

> find_elements_ 와 같이 elements 로 호출할 경우, 일치되는 모든 데이터 리스트를 가져옴

Selenium 사용법

- 크롤링 사이트 정보 확인
  - 사이트 주소: driver.current_url
  - 사이트 타이틀: driver.title

```
print (driver.current_url)
print (driver.title)
```

Headless Chrome: 최신 크롤링 기술

> 가능한 모든 크롤링 기술도 다룹니다.

- Headless Chrome: PhantomJS와 유사한 기술로 크롬브라우저 기능으로 개발됨
  - 성능상 개선이 있다는 주장도 있음(https://hackernoon.com/benchmark-headless-chrome-vs-phantomjs-e7f44c6956c)

> headlesschrome.ipynb

쉬어가기

- 크롤링은 후킹(hooking) 기술에 가까움
- 웹사이트에 따라 될수도 안될 수도 있는 기술
    - 대부분 웹사이트는 크롤링을 막으려고 함
- 완벽한 기술이 있다기 보다, 여러 크롤링 기술을 케이스마다 번갈아가며 사용 필요
    - 예) Selenium 또는 Headless Chrome 사용법을 알고 둘다 번갈아가며 크롤링해보고, 잘되는 것으로 사용하는 것이 좋음

Headless Chrome 사용법

- 일전에 사용한 모든 기능/코드 동일
  - 크롬 브라우저 버전만 최소 60이상이면 기능 사용 가능
  - 다음 코드만 추가해주면 됨

```
from selenium import webdriver

options = webdriver.ChromeOptions()
options.add_argument('headless')
options.add_argument('window-size=1920x1080')
options.add_argument("disable-gpu")
options.add_argument("User-Agent:  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4
options.add_argument("lang=ko_KR")

chromedriver = '/usr/local/Cellar/chromedriver/chromedriver' # 맥
driver = webdriver.Chrome(chromedriver, options=options)
driver.get('http://v.media.daum.net/v/20170202185812986')

body = driver.find_element_by_id('harmonyContainer')
print (body.text)
driver.quit()
```

Headless Chrome 사용법

- 다음 코드가 headless chrome을 사용을 선언하는 것임

```
options.add_argument('headless')
```

Headless Chrome 사용법

> 크롤링을 막는 사이트들이 있고, Headless Chrome의 경우는 명백히 크롤링이므로 막을 수도 있음
> Headless Chrome이 사용률이 높지 않으므로, 참고로만 알아두고, 다음 코드 정도만 사용하면 됨

- 부수적인 코드
  - 브라우저 화면 사이즈
  - GPU(그래픽카드) 사용하지 않겠다는 의미 (가끔 에러를 일으키기 때문에...)
  - User Agent를 실제 사용자가 브라우저를 오픈한 것처럼 보이게 하기
  - 실제 사용자가 브라우저를 오픈하면, 언어 설정이 되지만, Headeless Chrome은 안되기 때문에 넣어줌

```
options.add_argument('window-size=1920x1080')
options.add_argument("disable-gpu")
options.add_argument("User-Agent:  Mozilla/5.0 (Macintosh; Intel Mac OS X 10_14_4
options.add_argument("lang=ko_KR")
```

Headless Chrome 사용법

- 다음 코드가 headless chrome을 사용을 선언하는
- chorme_options를 넣어줘야 함

```
driver = webdriver.Chrome(chromedriver, options=options)
```

실전 예제1: 다음 뉴스 기사 제목 가져오기

- 주요 함수 - find_element_by_tag_name(), find_elements_by_tag_name()
  - find_element_by_tag_name(): 최초 발견한 태그만 가져오기
  - find_elements_by_tag_name(): 모든 태그 리스트로 가져오기

```
from selenium import webdriver

chromedriver = '/usr/local/Cellar/chromedriver/chromedriver'
driver = webdriver.Chrome(chromedriver)
driver.get('http://v.media.daum.net/v/20170202185812986')

# 최초 발견한 태그만 검색
title = driver.find_element_by_tag_name('h3')
print (title.text)

# 모든 태그 검색
h3s = driver.find_elements_by_tag_name('h3')
for h3 in h3s:
    print (h3.text)
driver.quit()
```

실전 예제2: 다음 뉴스 기사 내용 가져오기

- 주요 함수 - find_element_by_id(), find_elements_by_id()
  - find_element_by_id(): 최초 발견한 아이디를 가진 태그만 가져오기
  - find_elements_by_id(): 아이디를 가진 모든 태그 리스트로 가져오기

```
from selenium import webdriver

chromedriver = '/usr/local/Cellar/chromedriver/chromedriver'
driver = webdriver.Chrome(chromedriver)
driver.get('http://v.media.daum.net/v/20170202185812986')

body = driver.find_element_by_id('harmonyContainer')
print (body.text)

driver.quit()
```

실전 예제3: 다음 뉴스 기사 내용 가져오기

- 주요 함수 - find_element_by_css_selector(), find_elements_by_css_selector()
  - find_element_by_css_selector(): CSS selector로 태그 가져오기
  - find_elements_by_css_selector(): CSS selector로 태그 리스트 가져오기

```
from selenium import webdriver

chromedriver = '/usr/local/Cellar/chromedriver/chromedriver'
driver = webdriver.Chrome(chromedriver)
driver.get('http://v.media.daum.net/v/20170202180355822')

# 클래스가 tit_view인 h3태그
title = driver.find_element_by_css_selector("h3.tit_view")
print (title.text)
driver.quit()
```

실전 예제4: 다음 뉴스 기사 내용 가져오기

- 주요 함수: 요소 내용 가져오기
  - head 태그 관련: get_attribute('text')
  - body 태그 관련: text

```
from selenium import webdriver

chromedriver = '/usr/local/Cellar/chromedriver/chromedriver'
headless_options = webdriver.ChromeOptions()
headless_options.add_argument('headless')
driver = webdriver.Chrome(chromedriver, options=headless_options)

driver.get('http://v.media.daum.net/v/20170202180355822')
title_data = driver.find_element_by_css_selector('html head title')
print(title_data.get_attribute('text'), title_data.text)

contents = driver.find_element_by_css_selector("div#harmonyContainer")
# body 안에 있는 태그 요소는 .text 로 추출할 수 있습니다. (출력이 잘 안되면, 둘다 써보셔도 좋습니다.
print(contents.get_attribute('text'), contents.text)

driver.quit()
```

실전 예제5: 다음 사이트 특정 태그 가져오기

- CSS Selector 로 특정 속성값을 가진 태그 가져오기

```
from selenium import webdriver

# driver = webdriver.PhantomJS('C:/dev_python/phantomjs-2.1.1-windows/bin/phanto
driver = webdriver.PhantomJS('/usr/local/Cellar/phantomjs/2.1.1/bin/phantomjs') #
driver.get('http://v.media.daum.net/v/20170202180355822')

# role attribute가 navigation인 div태그
nav = driver.find_element_by_css_selector("div[role='navigation']")
print(nav.text)
driver.quit()
```

실전 예제6: 트위터 사이트 로그인 해보기

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

# 드라이버 생성 방법1 (selenium)
# chromedriver = 'C:/dev_python/Webdriver/chromedriver.exe' # 윈도우
chromedriver = '/usr/local/Cellar/chromedriver/chromedriver' # 맥
driver = webdriver.Chrome(chromedriver)

driver.get("http://www.twitter.com")
elem = driver.find_element_by_name("session[username_or_email]")
elem.clear()
elem.send_keys("jhleeroot@gmail.com")
elem = driver.find_element_by_name("session[password]")
elem.send_keys("funcoding1")
elem.send_keys(Keys.RETURN)

time.sleep(5)

print (driver.current_url)
print (driver.title)

driver.quit()
```

참고: PhantomJS: 화면이 없는 브라우저

> 꼭 크롤링을 위해 페이지 화면까지 띄워서 볼 필요는 없으니, 화면은 띄우지 말자
> 시간이 단축될 것이라고 기대하지만, 실제로는 유사하거나 느린 경우도 있으므로 경우에 따라 선택 사용

- WebTesting을 위해 나온 화면이 존재하지 않는 브라우저
- 터미널환경에서 동작하는 크롤러의 경우 PhantomJS 브라우저 사용 권장

사전준비 (PhantomJS 설치)

1. 윈도우/맥: PhantomJS 다운로드 후 절적한 디렉토리에 압축을 품 (http://phantomjs.org/download.html)

> 맥의 경우 brew install phantomjs 터미널 명령을 통해서도 설치 가능 (단, brew 설치시)

- Homebrew 설치 (https://brew.sh/index_ko)

특정 사이트에서 검색 결과 가져오기

- 크롤링 사이트 호출 및 확인

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

# phantomJS 드라이버
# driver = webdriver.PhantomJS('C:/dev_python/phantomjs-2.1.1-windows/bin/phanto
driver = webdriver.PhantomJS('/usr/local/Cellar/phantomjs/2.1.1/bin/phantomjs') #

driver.get("http://www.python.org")
```