Chapter. 02

데이터 크롤링

Selenium

FASTCAMPUS

ONLINE

금융공학/퀀트I

강사. 서찬웅

이번 시간에 배울 내용

강의내용

1. 자바스크립트란?
2. Ajax 기술
3. Selenium
4. css 선택자 및 xpath
5. Chrome(firefox) 드라이버
6. PhantomJS

I 자바스크립트에 대해서 알아봅시다

- 자바스크립트란(JS)?
  자바스크립트(영어: JavaScript)는 객체 기반의 스크립트 프로그래밍 언어이며 웹페이지를 동적으로 제어하기 위해서 고안된 언어입니다. 웹 브라우저 내에서 주로 사용됩니다.

- 우리가 크롤링 하고 싶은 웹 페이지는 아래와 같이 구분할 수 있습니다
  - 데이터 출력을 모두 자바스크립트로 처리하는 페이지
  - 자바스크립트는 보조적인 역할을 하는 페이지

- Python의 표준 라이브러리는 자바스크립트를 실행할 수 없습니다.

- JS를 실행하는 크롤러를 만들기 위해서는 Selenium이라는 프로그램의 힘을 빌려야 합니다.

- JS를 실행하는 크롤러는 html만 해석하는 크롤러에 비해서 페이지를 처리하는 시간이 길어지고, 메모리도 많이 소비합니다. 일상적으로 사용하는 웹 브라우저처럼 외부 JS, CSS, 이미지등 페이지에 있는 모든 것들을 읽어 들인 후 자바스크립트 코드를 실행하기 때문입니다.

- 자바스크립트를 실행이 필요한 페이지는 불가피 하게 사용해야 하지만, 일반적인 단순한 페이지는 표준 라이브러리를 사용하는 것이 좋을 수도 있습니다.

I Ajax 및 Selenium에 대해서 알아보겠습니다.

- 웹에서 특정 버튼을 누르거나 어떤 행동을 했을 때 페이지가 새로 고치지 않고 해당 결과 부분만 출력되는 경험을 누구나 다 경험 했을 것입니다. (예 : 다음지도, 지메일, 댓글 더보기...)
- 새로고침을 하지 않고 폼을 전송하여 서버에서 정보를 가져오는 기술을 ajax라고 합니다.
  - ajax는 비동기 자바스크립트와 XML의 약자입니다.
  - ajax는 서버에 별도의 페이지를 요청하지 않고 정보를 주고 받기 위해서 사용됩니다. 이렇게 자바스크립트로 된 ajax 기술을 사용한 페이지는 requests 라이브러리로 정보를 가져올 수 없습니다.
- Selenium은 웹 애플리케이션을 테스트 하기 위한 자동화 도구입니다. 최근에는 selenium을 사용하여 파이썬에서 ajax(자바스크립트)로 구현된 정보를 가져올 수 있습니다.
- Selenium은 브라우저를 조작하기 위해 드라이버라는 기능을 제공하며, 드라이버를 통해 웹 브라우저(파이어폭스, 크롬등)을 제어할 수 있습니다.
- 크롤링을 할 때 웹 브라우저를 통해서 데이터를 가져오지만, 항상 웹 브라우저가 실행되어야 한다는 단점이 있습니다. 이 단점을 극복하기 위해서 '화면이 없는 웹 브라우저'인 헤드리스 브라우저 PhantomJS를 사용합니다.

I Selenium에 대해서 정리하겠습니다.

- Selenium은 웹사이트 테스트 목적으로 개발. 활용하면 강력한 웹 스크레핑 도구로 사용할 수 있습니다.
- Selenium은 Chrome이나 Explorer같은 브라우저가 웹사이트를 불러오고, 필요한 데이터를 가져오고, 웹 페이지 상에서 특정한 행동(로그인, 메뉴 클릭등..)을 자동화 가능
- Selenium은 자체적으로 웹 브라우저가 내장되어 있지 않아 컴퓨터에 설치되어 있는 브라우저을 실행
  - Selenium을 chrome과 함께 사용하면 Python으로 Chrome에서 사람이 하는 행위를 모두 제어할 수 있음
  - 위와 같은 방법은 웹 사이트 상에서 어떤 일이 일어나는지 지켜보기 편하지만, 내가 하는 작업을 다른 사람이 볼 수도 있습니다.
  - 백그라운드에서 조용히 실행되는 것을 원하면 컴퓨터에 설치되어 있는 브라우저 대신 Phantomjs를 사용다.
- PhantomJS는 인터페이스가 없는 headless 브라우저입니다.
  - 웹 사이트를 메모리에 올려놓구 페이지의 자바스크립트를 실행하지만, 화면 출력은 전혀 없습니다.
  - Selenium과 PhantomJS를 결합하면 쿠키와 자바스크립트, 헤더, 그 외 필요한 모든 것을 쉽게 처리할 수 있는 강력한 크롤링 도구가 됩니다.

I Selenium을 설치 및 준비 작업을 진행합니다.

- Selenium은 Third Party LIbrary 이며, anaconda에 기본으로 설치된 라이브러리가 아닙니다.
- Jupyter CelI에서 Selenium을 설치해 보겠습니다.

!로 시작하는 줄은 느낌표 다음에 있는 내용을 시스템 쉘에서 실행하라는 의미입니다.

```
!pip install selenium
```

```
Collecting selenium
  Downloading https://files.pythonhosted.org/packages/41/c6/78a9a0d0150dbf43095c6f422fdf6f948e18453c5ebbf92384175b372ca2/selenium-3.13.0-py2
.py3-none-any.whl (946kB)
Installing collected packages: selenium
Successfully installed selenium-3.13.0
```

```
distributed 1.21.8 requires msgpack, which is not installed.
```

- 두 번째로 할일은 브라우저를 제어할 수 있는 드라이버를 설치하는 것입니다.
  - 크롬, 파이어폭스등 브라우저에 따른 드라이버는 별도로 존재합니다.
  - 크롬 : http://chromedriver.chromium.org/downloads
  - 파이어폭스 : https://github.com/mozilla/geckodriver/releases

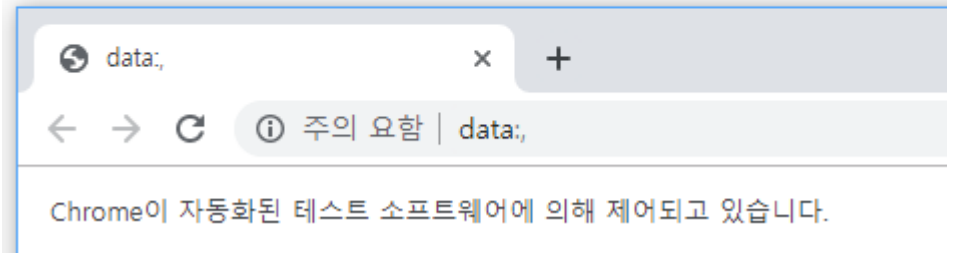
- 세 번째로 할일은 헤드리스 브라우저인 PhantomJS를 설치하는 것입니다. 아래 사이트에서 다운로드
  - http://phantomjs.org/download.html

I Selenium을 사용하기 위한 세팅 작업을 진행합니다.

- chrome을 기반으로 설명하겠습니다. (firefox도 거의 비슷합니다.)
- 현재 사용중인 chrome 버전인 **75.0.3770.90**의 드라이버를 다운로드하여 c:\chromedriver 폴더에 압축을 풉니다.
- 이번에는 spyder에서 작업을 진행하겠습니다.
- selenium 라이브러리 안에 webdriver를 호출하고 Chrome()을 실행합니다. 매개변수로 드라이버 실행파일의 경로를 적습니다. (상대경로 or 절대경로)

```
from selenium import webdriver

driver = webdriver.Chrome('c:/chromedriver/chromedriver.exe')
```



- 코드를 실행하면 Chrome이 자동으로 실행이 됩니다. 그리고 옆의 화면처럼 자동화된.... 제어되고 있습니다. 표시가 되어 있습니다.
- 현재 실행된 Chrome을 제어하여 웹 정보를 수집할 수 있습니다. Chrome과 같은 브라우저를 실행하지 않고 눈에 보이지 않은 브라우저를 headless라고 한다고 앞에서 이야기 했습니다. headless 브라우저인 PhantomJS는 Selenium의 기능을 리뷰하고 그 뒤에 설치해보겠습니다.

I Selenium의 메소드에 대해서 알아보겠습니다.

- selenium의 기능을 알아보겠습니다.
- 웹 페이지의 DOM의 정보를 바탕으로 선택자를 선택할 수 있습니다.
  - find_element_by_id() : id 값을 사용하여 위치를 찾는다.
  - find_element_by_css_selector () : css 요소의 표기 방법으로 된 위치를 찾는다.
  - find_element_by_xpath () : xml의 특정 요소로 지정된 위치를 찾는다.
- 선택자 함수를 사용하면 브라우저에 해당 항목을 선택한 것처럼 됩니다.
  - 선택자 함수들의 click(), send_keys() 메소드를 가지고 있습니다. , get() : 특정 사이트로 이동
  - click() : 선택자 함수로 선택된 곳을 실제로 클릭 , clear() : 선택된 곳에 타이핑 된 내용을 삭제
  - implicitly_wait() : 브라우저에서 웹 페이지 요소들이 로딩될 수 있도록 특정 시간 기다리는 함수
  - send_keys() : 선택자 함수로 선택된 곳에 키보드로 타이핑하는 것처럼 데이터 전송
  - page_source : drive가 보고 있는 웹 데이터를 가지고 있는 문자열 변수
- 사이트에 로그인 할 때 로그인 부분을 클릭하고, id, passwd 입력하는 것을 파이썬 코드로 똑같이 재현하는 것이라고 생각하면 됩니다. 일종의 자동화입니다.

I PhantomJS를 설치해 보겠습니다.

- chrome 브라우저를 사용하여 driver를 이용하여 접근하는 방법은 항상 브라우저가 같이 실행되어야 합니다. 이번에는 headless 방식으로 화면에 브라우저가 보이지 않은 PhantomJS를 설치해보겠습니다.
- PhantomJS를 다운로드 받아 C 드라이브에 압축을 푼 예제입니다. 이전 Chrome하고 차이가 있다면 webdriver에서 호출하는 메소드가 PhantomJS로 변경되었다는 것입니다. 그리고 매개변수로 압축을 푼 곳에 있는 phantomjs.exe 파일 위치를 넘겨줍니다.
- 실행이 되면 화면에 추가적인 창이 띄지 않습니다.
- 이제 Seleinium이라는 강력한 도구를 Python과 함께 구동하는 방법을 알아봤습니다.

```
from selenium import webdriver

driver = webdriver.PhantomJS('c:/phantomjs-2.1.1/bin/phantomjs.exe')
```

정리

- 자바스크립트가 있는 사이트에서 데이터 가져오는 방법
- 셀레니엄 사용법
- css 선택자
- 브라우저 드라이버
- headless 브라우저

감사합니다

FAST CAMPUS
ONLINE
서찬웅 강사.