

Selenium

Selenium 이란?

Selenium은 주로 웹앱을 테스트하는데 이용하는 프레임워크다.

webdriver라는 API를 통해 운영체제에 설치된 Chrome등의 브라우저를 제어하게 된다.

브라우저를 직접 동작 시킨다는 것은 JavaScript를 이용해 비동기적으로 혹은 뒤늦게 불러와지는 컨텐츠들을 가져올 수 있다는 것이다. 즉, '눈에 보이는' 컨텐츠라면 모두 가져올 수 있다는 뜻이다. 우리가 requests에서 사용했던 .text의 경우 브라우저에서 '소스보기'를 한 것과 같이 동작하여, JS등을 통해 동적으로 DOM이 변화한 이후의 HTML을 보여주지 않는다. 반면 Selenium은 실제 웹 브라우저가 동작하기 때문에 JS로 렌더링이 완료된 후의 DOM결과물에 접근이 가능하다.

어떻게 설치하나?

pip selenium package

Selenium을 설치하는 것은 기본적으로 pip를 이용한다.

```
!pip install selenium
```

참고: Selenium의 버전은 자주 업데이트 되고, 브라우저의 업데이트 마다 새로운 Driver를 잡아주기 때문에 항상 최신버전을 깔아 주는 것이 좋다.

어떻게 설치하나?

Selenium은 webdriver라는 것을 통해 디바이스에 설치된 브라우저들을 제어할 수 있다. Chrome을 사용해 볼 예정이다.

Chrome WebDriver

크롬을 사용하려면 로컬에 크롬이 설치되어있어야 한다.



그리고 크롬 드라이버를 다운로드 받아주자.

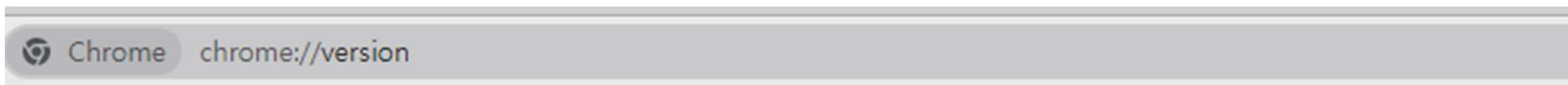
해당하는 크롬 드라이버를 받아야 한다.


<https://sites.google.com/chromium.org/driver/downloads>

어떻게 설치하나?

크롬에서는 현재 버전별 지정된 chromedriver를 받도록 안내하며, 버전에 일치하지 않는 드라이버를 사용하면 에러가 납니다.

현재 사용하는 크롬의 버전은 크롬 창에  chrome://version  이 URL을 주소창에 그대로 입력하면(http없이) 버전을 확인할 수 있습니다.



Chrome: 103.0.5060.114 (공식 빌드) (64비트) (cohort: Stable) 
개정: atc2360c5b02abd4d6ab33796ad8a268a6128226-refs/branch-heads/5060@{#1124}
OS: Windows 11 Version 21H2 (Build 22000.778)
JavaScript: V8 10.3.174.18
사용자 에이전트: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.5060.114 Safari/537.36
명령줄: "C:\Program Files\Google\Chrome\Application\chrome.exe" --flag-switches-begin --disable-features=BlockInsecurePrivateNetworkRequests --flag-switches-end
실행 가능 경로: C:\Program Files\Google\Chrome\Application\chrome.exe
프로필 경로: C:\Users\ceo\AppData\Local\Google\Chrome\User Data\Default
활성 버전: 8d569531-4f54ac4d
5e3a236d-4113a79e

<chrome://version/>

어떻게 설치하나?

The screenshot shows the ChromeDriver download page. A red arrow points from a Korean annotation to the link for ChromeDriver 77.0.3865.40 in the 'Current Releases' list.

ChromeDriver - WebDriver for Chrome

Search this site

Downloads

Current Releases

- If you are using Chrome version 78, please download [ChromeDriver 78.0.3904.11](#)
- If you are using Chrome version 77, please download [ChromeDriver 77.0.3865.40](#)
- If you are using Chrome version 76, please download [ChromeDriver 76.0.3809.126](#)
- For older version of Chrome, please see below for the version of ChromeDriver that supports it.

If you are using Chrome from Dev or Canary channel, please following instructions on the [ChromeDriver Canary](#) page.

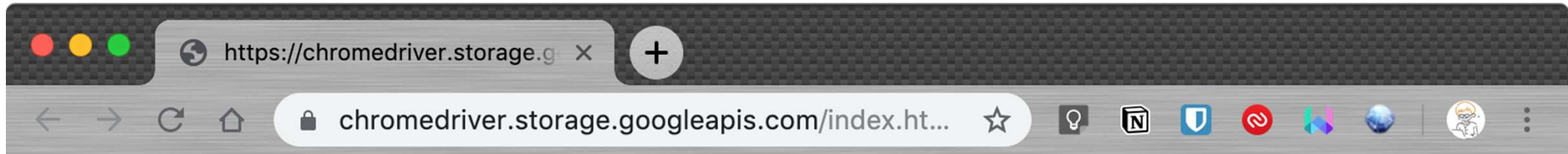
For more information on selecting the right version of ChromeDriver, please see the [Version](#)

Left Sidebar Navigation:

- CHROMEDRIVER
- CAPABILITIES & CHROMOPTIONS
- CHROME EXTENSIONS
- CHROMEDRIVER CANARY
- CONTRIBUTING
- ▼ DOWNLOADS
 - VERSION SELECTION
- ▼ GETTING STARTED
 - ANDROID
 - CHROMEOS
- ▼ LOGGING
 - PERFORMANCE LOG
- MOBILE EMULATION

버전을 클릭하면 아래와 같은 OS별 Driver파일이 나열되어있다. 사용하는 OS에 따른 driver를 받아주자.

어떻게 설치하나?



Index of /77.0.3865.40/

Name	Last modified	Size	ETag
Parent Directory	-	-	-
chromedriver_linux64.zip	2019-08-20 18:02:46	5.17MB	b4431816072192a2d36a10fa8cfde344
chromedriver_mac64.zip	2019-08-20 18:02:48	7.05MB	812570697aadcd7a9038041b27437054
chromedriver_win32.zip	2019-08-20 18:02:49	4.54MB	7e94b11b8157e856b918f64d1b4af424
notes.txt	2019-08-20 18:02:53	0.00MB	0609e0eff91a2087279a1600bb37198e

윈도우 버전을 다운 받아서 압축을 풀어두자. 경로를 기억해 둔다.

이 경로를 나중에 Selenium 객체를 생성할 때 지정해 주어야 한다. (그래야 python이 chromedriver를 통해 크롬 브라우저를 조작할 수 있다!)

Selenium으로 사이트 브라우징

Selenium은 webdriver api를 통해 브라우저를 제어한다.

우선 webdriver를 import해주자.

```
from selenium import webdriver
```

이제 driver라는 이름의 webdriver 객체를 만들어 주자.

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

Selenium은 기본적으로 웹 자원들이 모두 로드될때까지 기다려주지만, 암묵적으로 모든 자원이 로드될때 까지 기다리게 하는 시간을 직접 implicitly_wait을 통해 지정할 수 있다.

Selenium으로 사이트 브라우징

```
driver.implicitly_wait(3)
```

암묵적으로 웹 자원 로드를 위해 3초까지 기다려 준다.
이제 특정 url로 브라우저를 켜 보자.

```
driver.get('https://google.com')
```

만약 chromedriver의 위치가 정확하다면 새 크롬 화면이 뜨고 구글 첫 화면으로 들어가질 것이다.
Selenium은 driver객체를 통해 여러가지 메소드를 제공한다.

네이버 로그인 하기

네이버는 requests를 이용해 로그인하는 것이 어렵다. 프론트 단에서 JS처리를 통해 로그인 처리를 하기 때문인데, Selenium을 이용하면 보다 쉽게 로그인을 할 수 있다.

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

import time

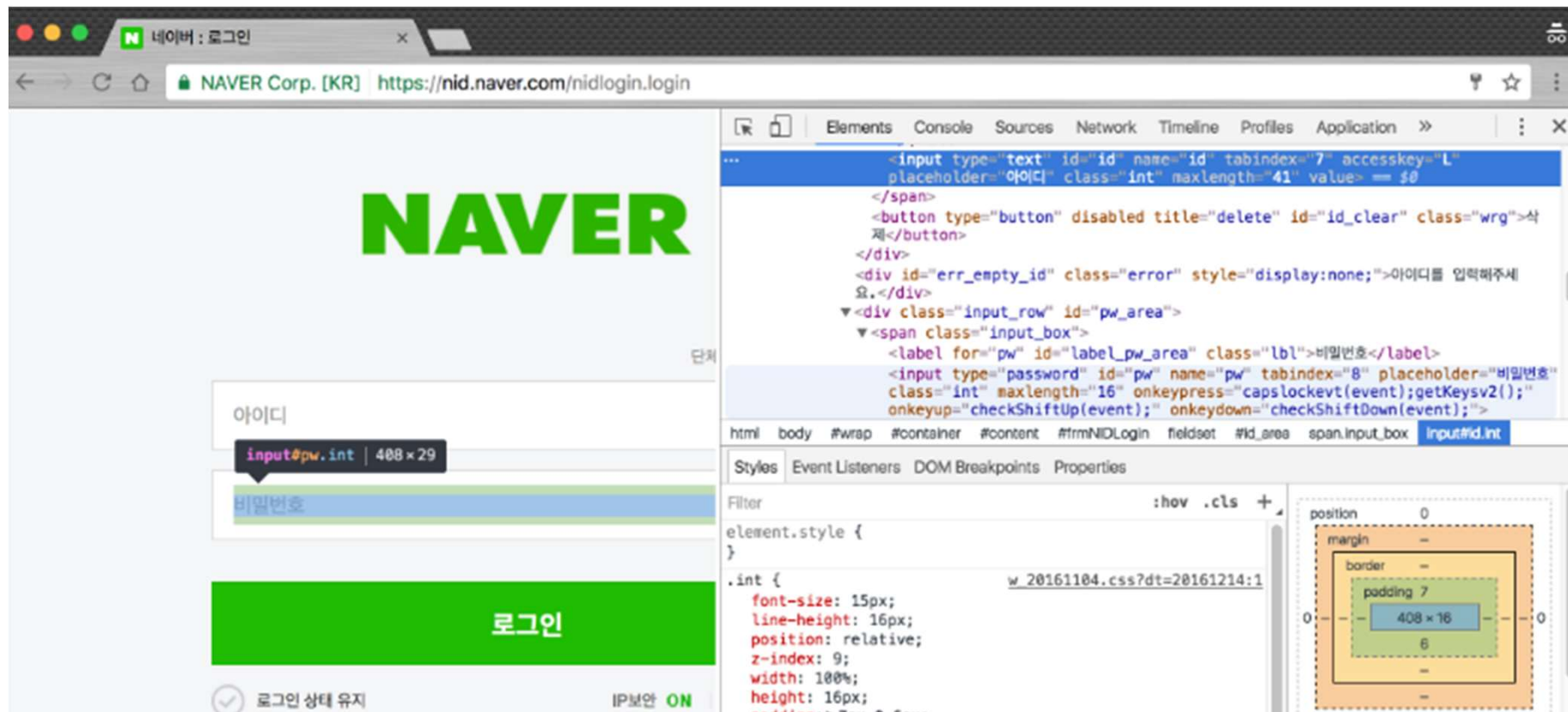
driver = webdriver.Chrome('./chromedriver')

time.sleep(2)

driver.get('https://nid.naver.com/nidlogin.login')
```

네이버 로그인 하기

네이버 로그인 화면을 확인 해 보면 아이디를 입력받는 부분의 name이 id, 비밀번호를 입력받는 부분의 name이 pw인 것을 알 수 있다.



네이버 로그인 하기

```
driver.get('https://nid.naver.com/nidlogin.login?mode=form&url=https%3A%2F%2Fwww.naver.com')  
time.sleep(1)
```

```
id = driver.find_element("id", "id")  
id.send_keys('아이디')  
time.sleep(1)
```

```
pw = driver.find_element("id", "pw")  
pw.send_keys('비밀번호')  
time.sleep(1)
```

```
btn = driver.find_element("id", "log.login")  
btn.click()
```

네이버 로그인 하기

The image shows the Naver login page. At the top center is the 'NAVER' logo in green. Below it, on the right, is the text '단체아이디 로그인 방법' in a small grey font. There are two input fields: the first is labeled 'naver_id' and the second contains a series of dots representing a password. Below these fields is a large green button with the white text '로그인' (Login).

성공적으로 값이 입력된 것을 확인할 수 있다.

하지만, 로봇으로 간주하고 캡차에서 막힐 수 있다.

이를 우회 하는 방법으로 복사/붙여넣기로 해결할 수 있다.

네이버 로그인 하기

사람이 로그인 한 것처럼 속이기 위해 복사/붙여넣기 모듈 설치

```
!pip install pyperclip
```

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

그리고, 이번에는 위처럼 By를 사용해서 좀 더 다양한 방식으로 아이디와 비밀번호 element를 찾아올 수 있도록 수정해보자!

네이버 로그인 하기

각 element에 따라 method를 따로 사용하는 것 보다 깔끔하게 정리하기 위해 By를 사용해 봅시다.

driver.find_element(By.<속성>, '<속성 값>')으로 사용합니다. 여러 element를 찾을 경우 find_elements로 할 수 있습니다.
사용은 아래와 같이 합니다.

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

driver.find_element(By.XPATH, '//button[text()="Some text"]')
driver.find_element(By.XPATH, '//button')
driver.find_element(By.ID, 'loginForm')
driver.find_element(By.LINK_TEXT, 'Continue')
driver.find_element(By.PARTIAL_LINK_TEXT, 'Conti')
driver.find_element(By.NAME, 'username')
driver.find_element(By.TAG_NAME, 'h1')
driver.find_element(By.CLASS_NAME, 'content')
driver.find_element(By.CSS_SELECTOR, 'p.content')

driver.find_elements(By.ID, 'loginForm')
driver.find_elements(By.CLASS_NAME, 'content')
```

네이버 로그인 하기

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
import time
import pyperclip

driver = webdriver.Chrome('./chromedriver')
time.sleep(2)

driver.get('https://nid.naver.com/nidlogin.login?mode=form&url=https%3A%2F%2Fwww.naver.com')
time.sleep(1)

id_value = ' 아이디 '
Pw_value = ' 비밀번호 '

id = driver.find_element("id", "id")
id.click()
pyperclip.copy(id_value)
id.send_keys(Keys.CONTROL, 'v')
```


네이버 로그인 하기

```
time.sleep(1)

pw = driver.find_element("id", "pw")
pw.click()
pyperclip.copy(pw_value)
pw.send_keys(Keys.CONTROL, 'v')

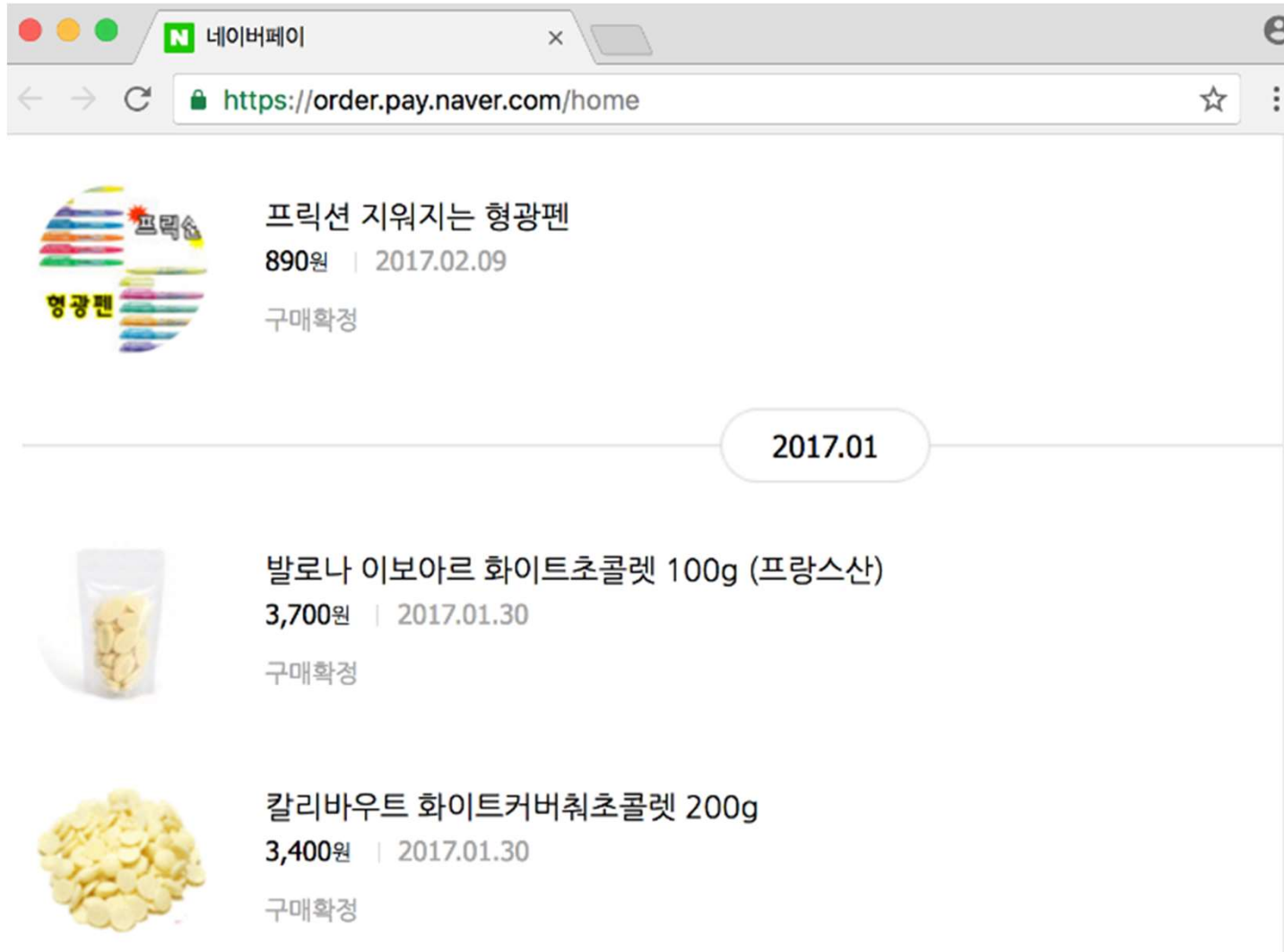
time.sleep(2)
driver.find_element(By.CLASS_NAME, 'btn_login').click()
```

Time.sleep()으로 시간차를 두어 마치 사람이 로그인 하는 것처럼 잘 조정해보자!

이제 로그인이 잘 될 것이다.


네이버페이 주문 내역 가져오기

로그인이 필요한 페이지인 네이버 페이의 주문내역 페이지를 가져와보자.





네이버페이

https://order.pay.naver.com/home

 프릭션 지워지는 형광펜
890원 | 2017.02.09
구매확정

2017.01

 발로나 이보아르 화이트초콜렛 100g (프랑스산)
3,700원 | 2017.01.30
구매확정

 칼리바우트 화이트커버칩초콜렛 200g
3,400원 | 2017.01.30
구매확정

네이버페이 주문 내역 가져오기

네이버 페이의 Url은 `https://order.pay.naver.com/home` 이다.
위 페이지의 알림 텍스트를 가져와 보자.

```
from bs4 import BeautifulSoup

driver.get('https://order.pay.naver.com/home')
html = driver.page_source
soup = BeautifulSoup(html, 'html.parser')
soup.text
```

Html 소스가 출력 될 것이다.
앞서 배운 방식으로 분석해서 제목을 다운받아 보자