

Selenium 1

1. Install Selenim and Chrome Driver
2. Selenium의 기본 사용법

1. Install Selenim and Chrome Driver

- 크롬 드라이버 다운로드
 - 크롬 브라우저를 사용하기 위해 크롬 드라이버를 다운로드 해야 합니다.
 - <https://sites.google.com/a/chromium.org/chromedriver/downloads> (<https://sites.google.com/a/chromium.org/chromedriver/downloads>)
 - <https://chromedriver.storage.googleapis.com/index.html?path=2.42/> (<https://chromedriver.storage.googleapis.com/index.html?path=2.42/>)
- (참고) 환경변수 추가
 - 내컴퓨터 오른쪽 클릭 -> 속성 -> 고급시스템설정 -> 환경변수 -> 시스템변수의 Path 더블클릭 -> chromedriver path를 추가
 - 예: C:\Users\W(사용자계정이름)\chromedriver 으로 추가
 - 환경변수 추가 후에도 드라이버가 실행되지 않는 경우에는 아래의 두가지 방법의 사용이 가능
 - 크롬 드라이버 파일을 동일한 디렉토리에 위치해서 실행
 - 크롬 드라이버를 호출하는 함수에 아규먼트로 크롬 드라이버의 파일경로를 추가

패키지 호출

```
In [1]: from selenium import webdriver
```

- 브라우저 열기 - 방법 1 직접 드라이버의 경로를 지정하여 실행

```
In [2]: driver = webdriver.Chrome("C:/Myexam/chromedriver/chromedriver.exe")
```

- 브라우저 열기 : 방법 2
 - 환경변수 설정이 되어 있는 상태에서 실행
 - 드라이버 파일이 같은 디렉토리에 있는 경우에 실행

```
In [3]: # driver = webdriver.Chrome()
```

2. Selenium의 기본 사용법

1. 셀레니움이란?
2. 웹 드라이버 객체 만들기 및 페이지 이동
3. 윈도우 사이즈 조절
4. 브라우저 스크롤 이동
5. Alert 다루기
6. 입력창에 문자열 입력하기
7. 버튼 클릭하기
8. 텍스트 데이터 가져오기
9. 속성 데이터 가져오기
10. 윈도우 및 브라우저 종료하기

2.1 셀레니움이란?

- 셀레니움은 <http://www.seleniumhq.org/> (<http://www.seleniumhq.org/>) 웹페이지에서 자세한 내용과 사용법을 확인하실수 있습니다.
- 셀레니움은 브라우저를 자동화하는 툴로 테스트를 목적으로 웹 어플리케이션을 자동화하는 목적으로 만들어졌지만 테스트에 국한되지 사용 되지는 않습니다.
- 웹 기반의 작업들을 자동화하여 업무 효율성을 높일수 있으며 다양한 언어와 다양한 브라우저를 지원합니다.
- <http://www.seleniumhq.org/about/platforms.jsp> (<http://www.seleniumhq.org/about/platforms.jsp>) 웹 페이지에서 지원하는 브라우저와 지원하는 OS와 지원하는 언어를 확인하실수 있습니다.
- 셀레니움은 Firefox, Internet Explorer, Safari, Opera Chrome 브라우저를 지원합니다.
- 지원하는 OS로는 Microsoft Windows, Apple OS X, Linux를 지원하며, 지원하는 언어는 C#, Haskell, Java, Javascript, Object-C, Perl, PHP, Python, R, Ruby를 지원합니다.

2.2 웹 드라이버 객체 만들기 및 페이지 이동

- driver의 get 함수를 이용하여 크롬 브라우저에서 페이지를 이동할수 있습니다.
- 아래의 코드는 naver웹 페이지로 이동합니다.
- url은 http를 꼭 포함해야 합니다

```
In [4]: driver.get("https://www.naver.com/")
```

2.3 윈도우 사이즈 조절

- driver의 set_window_size 함수를 이용하여 윈도우 사이즈를 조절할수 있습니다.
- 함수내의 첫번째 파라미터는 가로 사이즈를 나타내며 두번째 파라미터는 세로 사이즈를 나타냅니다.
- 사이즈의 단위 는 픽셀입니다.

```
In [5]: # (가로, 세로) - 1024 * 768 픽셀의 사이즈로 윈도우 브라우저를 열어줍니다.
driver.set_window_size(1024, 768)
```

2.4 브라우저의 스크롤 위치 이동

- 브라우저의 스크롤 위치 이동은 selenium의 webdriver에 따로 기능이 없기 때문에 javascript 코드를 실행 하여 브라우저 스크롤 위치를 이동시킬수 있습니다.
- javascript의 window.scrollTo 함수의 첫번째 파라미터는 가로축 위치를 나타내며 아 래 코드와 같이 200으로 설정을 하면 왼쪽에서 200픽셀만 큼의 브라우저 스크롤 위치를 이동함을 의미합니다.
- 두번째 파라미 터는 세로축의 위치를 나타내며 아래의 코드와 같이 300으로 지정을 하면 위에서 아래로 300픽셀 만큼 스크롤이 내려감을 의미 합니다.

```
In [6]: driver.execute_script("window.scrollTo(200, 300);")
```

2.5 Alert 다루기

- 크롤링이나 자동화를 하는 중간에 alert 메시지가 나오는 경우가 있습니다.
- 이럴때 크롤링이나 자동화를 계속 하게 하기 위해 alert 메시지를 다룰줄 알아야 합니

```
In [7]: # alert 체크
try:
    alert = driver.switch_to.alert
    print(alert.text)
except:
    print('alert 없음')
```

alert 없음

```
In [8]: driver.execute_script("alert('selenium test');")
```

```
In [9]: # alert 체크
try:
    alert = driver.switch_to.alert
    print(alert.text)
except:
    print('alert 없음')
```

selenium test

```
In [10]: # alert 확인 버튼 누르기
alert.accept()
```

2.7 버튼 클릭하기

- selenium driver의 css selector와 click 함수를 이용하여 웹페이지의 input 태그에 문자열을 입력할수 있습니다.

```
In [11]: driver.find_element_by_css_selector(".btn_submit").click()
```

2.8 텍스트 데이터 가져오기

- text 변수를 사용
- find_elements_by_css_selector : 여러개의 엘리먼트를 선택

```
In [13]: # 테드 사이트로 접속
driver.get('https://www.ted.com/talks')
```

```
In [14]: # 메인 배너 타이틀 가져오기
driver.find_element_by_css_selector('#banner-secondary').text
```

Out[14]: 'Join TED Recommends to get the best ideas, selected just for you'

```
In [15]: # 컨텐츠 리스트 제목 가져오기
contents = driver.find_elements_by_css_selector('#browse-results > div > .col')
len(contents)
```

Out[15]: 36

```
In [16]: # 셀렉터 확인
contents[0].find_element_by_css_selector('.media > .media__message .ga-link').text
```

Out[16]: 'The secret society of the Great Dismal Swamp'

```
In [17]: # 전체 데이터 가져오기
titles = []
for content in contents:
    title = content.find_element_by_css_selector('.media > .media__message .ga-link').text
    titles.append(title)
```

```
In [19]: titles[-5:]
```

Out[19]: ["History's deadliest king",
"A cleanse won't detox your body – but here's what will",
'What should humans take to space (and leave behind)?',
'Could we build a wooden skyscraper?',
"Why aren't there more Native American restaurants?"]

```
In [23]: # 사용가능한 언어 옵션 리스트 가져오기 - text는 하위 엘리먼트의 문자열까지 가져옵니다.
languages = driver.find_element_by_css_selector('#languages').text
languages = languages.split("\n")[1:-1]
languages
```

Out[23]: ['English', 'Español', '日本語', 'Português brasileiro', '中文 (繁體)', '한국어']

```
In [24]: # 한국어 선택 후 결과 컨텐츠의 제목 가져오기

# 셀렉트 박스를 선택
driver.find_element_by_css_selector('#languages [lang="ko"]').click()
```

```
In [25]: import time
```

```
In [26]: # 컨텐츠 가져오기
time.sleep(3)
contents = driver.find_elements_by_css_selector('#browse-results > div > .col')
titles = []
for content in contents:
    title = content.find_element_by_css_selector('.media > .media__message .ga-link').text
    titles.append(title)
titles[-5:]
```

Out[26]: ['인터넷에서 마음의 평화를 되찾는 법',
'트로이 전쟁의 수수께끼를 풀 수 있나요? - 데니스 E. 샤샤(Dennis E. Shasha)',
'똥을 하루에 한 번씩 누지 않아도 되는 이유',
'질병 치료법을 바꿔 놓을 전자약',
'태양을 바라본 여인 - 알렉스 젠들러']

2.9 속성 데이터 가져오기

- get_attribute 함수를 이용

```
In [27]: # 테드 에서 컨텐츠 링크 리스트 가져오기
time.sleep(3)
contents = driver.find_elements_by_css_selector('#browse-results > div > .col')
links = []
for content in contents:
    link = content.find_element_by_css_selector('.media > .media__message .ga-link').get_attribute('href')
    links.append(link)
links[-5:]
```

Out[27]: ['https://www.ted.com/talks/naomi_shimada_tips_for_reclaiming_your_peace_of_mind_online?language=ko',
'https://www.ted.com/talks/dennis_shasha_can_you_solve_the_trojan_war_riddle?language=ko',
'https://www.ted.com/talks/jen_gunter_why_you_shouldn_t_worry_about_pooping_once_a_day?language=ko',
'https://www.ted.com/talks/khalil_ramadi_electronic_pills_that_could_transform_how_we_treat_disease?language=ko',
'https://www.ted.com/talks/alex_gendler_the_woman_who_stared_at_the_sun?language=ko']

2.10 윈도우 및 브라우저 종료하기

- selenium dirver의 quit 함수를 이용하여 윈도우 창을 닫을수 있습니다

In [28]:

브라우저 닫기
driver.quit()

In []: