

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://chromedriver.chromium.org/downloads>
(<https://chromedriver.chromium.org/downloads>)
 - <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 -> 시스템 변수의 Path 더블클릭 -> chromedriver path를 추가
 - 예: C:\Users\%(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

[1]:

1 !pip install selenium

```

Collecting selenium
  Downloading selenium-4.7.2-py3-none-any.whl (6.3 MB)
    ----- 6.3/6.3 MB 31.0 M
B/s eta 0:00:00
Requirement already satisfied: certifi>=2021.10.8 in c:\Wprogramdata\Wanaconda3\lib\site-packages (from selenium) (2022.9.14)
Collecting trio~0.17
  Downloading trio-0.22.0-py3-none-any.whl (384 kB)
    ----- 384.9/384.9 kB ? e
ta 0:00:00
Collecting trio-websocket~0.9
  Downloading trio_websocket-0.9.2-py3-none-any.whl (16 kB)
Requirement already satisfied: urllib3[socks]~1.26 in c:\Wprogramdata\Wanaconda3\lib\site-packages (from selenium) (1.26.11)
Requirement already satisfied: sortedcontainers in c:\Wprogramdata\Wanaconda3\lib\site-packages (from trio~0.17->selenium) (2.4.0)
Requirement already satisfied: cffi>=1.14 in c:\Wprogramdata\Wanaconda3\lib\site-packages (from trio~0.17->selenium) (1.15.1)
Collecting outcome
  Downloading outcome-1.2.0-py2.py3-none-any.whl (9.7 kB)
Collecting async-generator>=1.9
  Downloading async_generator-1.10-py3-none-any.whl (18 kB)
Requirement already satisfied: idna in c:\Wprogramdata\Wanaconda3\lib\site-packages (from trio~0.17->selenium) (3.3)
Requirement already satisfied: attrs>=19.2.0 in c:\Wprogramdata\Wanaconda3\lib\site-packages (from trio~0.17->selenium) (21.4.0)
Collecting exceptiongroup>=1.0.0rc9
  Downloading exceptiongroup-1.1.0-py3-none-any.whl (14 kB)
Requirement already satisfied: sniffio in c:\Wprogramdata\Wanaconda3\lib\site-packages (from trio~0.17->selenium) (1.2.0)
Collecting wsproto>=0.14
  Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Requirement already satisfied: PySocks!=1.5.7,<2.0,>=1.5.6 in c:\Wprogramdata\Wanaconda3\lib\site-packages (from urllib3[socks]~1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\Wprogramdata\Wanaconda3\lib\site-packages (from cffi>=1.14->trio~0.17->selenium) (2.21)
Collecting h11<1,>=0.9.0
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
    ----- 58.3/58.3 kB 3.0 M
B/s eta 0:00:00
Installing collected packages: outcome, h11, exceptiongroup, async-generator, wsproto, trio, trio-websocket, selenium
Successfully installed async-generator-1.10 exceptiongroup-1.1.0 h11-0.14.0 outcome-1.2.0 selenium-4.7.2 trio-0.22.0 trio-websocket-0.9.2 wsproto-1.2.0

```

[6]:

```
1 from selenium import webdriver
```

- 브라우저 열기 - 방법 1 직접 드라이버의 경로를 지정하여 실행

[7]:

```
1 driver = webdriver.Chrome("C:/Myexam/chromedriver/chromedriver.exe")
```

```
C:\Users\Wstudent\AppData\Local\Temp\Wipykernel_884\W252834201.py:
1: DeprecationWarning: executable_path has been deprecated, please
pass in a Service object
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

[3]:

```
1 # 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 확인하실 수 있습니다.
- 셀레니움은 브라우저를 자동화하는 툴로 테스트를 목적으로 웹 어플리케이션을 자동화하는 목적으로 만들어졌지만 테스트에 국한되지 않고 사용됩니다.
- 웹 기반의 작업들을 자동화하여 업무 효율성을 높일 수 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 확인하실 수 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 이동할 수 있습니다.
- 아래의 코드는 naver웹 페이지로 이동합니다.
- url은 http를 꼭 포함해야 합니다

[8]:

```
1 driver.get("https://www.naver.com/")
```

2.3 윈도우 사이즈 조절

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

[9]:

```
1 # (가로, 세로) - 1024 * 768 픽셀의 사이즈로 윈도우 브라우저를 열어줍니다.
2 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 이동시킬 수 있습니다.
- javascript의 window.scrollTo 함수의 첫번째 파라미터는 가로축 위치를 나타내며 아래 코드와 같이 200으로 설정을 하면 왼쪽에서 200픽셀만큼의 브라우저 스크롤 위치를 이동함을 의미합니다.
- 두번째 파라미터는 세로축의 위치를 나타내며 아래의 코드와 같이 300으로 지정을 하면 위에서 아래로 300픽셀 만큼 스크롤이 내려감을 의미합니다.

[11]:

```
1 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 다룰 줄 알아야 합니다

[12]:

```
1 # alert 체크
2 try:
3     alert = driver.switch_to.alert
4     print(alert.text)
5 except:
6     print('alert 없음')
```

alert 없음

[13]:

```
1 driver.execute_script("alert('selenium test');")
```

[14]:

```
1 # alert 체크
2 try:
3     alert = driver.switch_to.alert
4     print(alert.text)
5 except:
6     print('alert 없음')
```

selenium test

[15]:

```
1 # alert 확인 버튼 누르기
2 alert.accept()
```

2.7 버튼 클릭하기

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

[16]:

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

[17]:

```
1 driver.find_element(By.CSS_SELECTOR, ".btn_submit").click()
```

2.8 텍스트 데이터 가져오기

- text 변수를 사용
- find_element(By.CSS_SELECTOR, 'classname'): 여러개의 엘리먼트를 선택
- find_elements(By.CSS_SELECTOR, 'classname'): 여러개의 엘리먼트를 선택

[18]:

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

[19]:

```
1 # 테드 사이트로 접속
2 driver.get('https://www.ted.com/talks')
```

[20]:

```
1 # 메인 배너 타이틀 가져오기
2 driver.find_element(By.CSS_SELECTOR, '#banner-secondary').text
```

'Join TED Recommends to get the best ideas, selected just for you'

[21]:

```
1 # 콘텐츠 리스트 제목 가져오기
2 contents = driver.find_elements(By.CSS_SELECTOR, '#browse-results > div')
3 len(contents)
```

36

[22]:

```
1 # 셀렉터 확인
2 contents[0].find_element(By.CSS_SELECTOR, '.media > .media__message .ga')
```

'The bizarre world of parasitic wasps'

[23]:

```
1 # 전체 데이터 가져오기
2 titles = []
3 for content in contents:
4     title = content.find_element(By.CSS_SELECTOR, '.media > .media__message')
5     titles.append(title)
```

[24]:

```
1 titles[-5:]
```

['How will AI change the world?',
 'How to tap into your awareness – and why meditation is easier
 than you think',
 'Demystifying the wild world of crypto',
 'How can we escape soaring energy bills? Stop using fossil fuel
 s',
 'A disability-inclusive future of work']

[25]:

```

1 # 사용가능한 언어 옵션 리스트 가져오기 - text는 하위 엘리먼트의 문자열까지
2 languages = driver.find_element(By.CSS_SELECTOR, '#languages').text
3 languages = languages.split("\n")[1:-1]
4 languages

```

```

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

```

[26]:

```

1 # 한국어 선택 후 결과 콘텐츠의 제목 가져오기
2
3 # 셀렉트 박스를 선택
4 driver.find_element(By.CSS_SELECTOR, '#languages [lang="ko"]').click()

```

[27]:

```

1 import time

```

[28]:

```

1 # 콘텐츠 가져오기
2 time.sleep(3)
3 contents = driver.find_elements(By.CSS_SELECTOR, '#browse-results > div')
4 titles = []
5 for content in contents:
6     title = content.find_element(By.CSS_SELECTOR, '.media > .media__message')
7     titles.append(title)
8 titles[-5:]

```

```

['외계인 피라미드 수수께끼를 풀 수 있나요? - 헨리 피치오토',
'과연 빠른 체중 감량은 가능할까?',
'이 회사들에는 CEO가 없지만 번창하고 있습니다',
'화석 연료를 대체할 최고의 희망, 원자력',
'우정이 뇌에 미치는 영향 - 샤넌 오델( Shannon Ode11)']

```

2.9 속성 데이터 가져오기

- get_attribute 함수를 이용

[29]:

```

1 # 테드 에서 콘텐츠 링크 리스트 가져오기
2 time.sleep(3)
3 contents = driver.find_elements(By.CSS_SELECTOR, '#browse-results > div')
4 links = []
5 for content in contents:
6     link = content.find_element(By.CSS_SELECTOR, '.media > .media__message')
7     links.append(link)
8 links[-5:]

```

```

['https://www.ted.com/talks/henri_picciotto_can_you_solve_the_alien_pyramid_riddle?language=ko',
 'https://www.ted.com/talks/heiman_chan_is_it_possible_to_lose_weight_fast?language=ko',
 'https://www.ted.com/talks/ted_ed_these_companies_with_no_ceo_are_thriving?language=ko',
 'https://www.ted.com/talks/isabelle_boemeke_nuclear_power_is_our_best_hope_to_ditch_fossil_fuels?language=ko',
 'https://www.ted.com/talks/shannon_odell_how_friendship_affects_your_brain?language=ko']

```

2.10 윈도우 및 브라우저 종료하기

- selenium driver의 quit 함수를 이용하여 윈도우 창을 닫을수 있습니다

[30]:

```

1 # 브라우저 닫기
2 driver.quit()

```

[]:

1