

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://sites.google.com/a/chromium.org/chromedriver/downloads>
 - <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\%(사용자계정이름)\chromdriver 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 [29]: !pip install selenium
```

Requirement already satisfied: selenium in c:\users\bobb\appdata\roaming\python\python311\site-packages (4.19.0)

Requirement already satisfied: urllib3<3,>=1.26 in c:\programdata\anaconda3\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (2.0.7)

Requirement already satisfied: trio~=0.17 in c:\users\bobb\appdata\roaming\python\python311\site-packages (from selenium) (0.25.0)

Requirement already satisfied: trio-websocket~=0.9 in c:\users\bobb\appdata\roaming\python\python311\site-packages (from selenium) (0.11.1)

Requirement already satisfied: certifi>=2021.10.8 in c:\programdata\anaconda3\lib\site-packages (from selenium) (2024.2.2)

Requirement already satisfied: typing_extensions>=4.9.0 in c:\programdata\anaconda3\lib\site-packages (from selenium) (4.9.0)

Requirement already satisfied: attrs>=23.2.0 in c:\users\bobb\appdata\roaming\python\python311\site-packages (from trio~=0.17->selenium) (23.2.0)

Requirement already satisfied: sortedcontainers in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (2.4.0)

Requirement already satisfied: idna in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (3.4)

Requirement already satisfied: outcome in c:\users\bobb\appdata\roaming\python\python311\site-packages (from trio~=0.17->selenium) (1.3.0.post0)

Requirement already satisfied: sniffio>=1.3.0 in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.3.0)

Requirement already satisfied: cffi>=1.14 in c:\programdata\anaconda3\lib\site-packages (from trio~=0.17->selenium) (1.16.0)

Requirement already satisfied: wsproto>=0.14 in c:\users\bobb\appdata\roaming\python\python311\site-packages (from trio-websocket~=0.9->selenium) (1.2.0)

Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in c:\programdata\anaconda3\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)

Requirement already satisfied: pycparser in c:\programdata\anaconda3\lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)

Requirement already satisfied: h11<1,>=0.9.0 in c:\users\bobb\appdata\roaming\python\python311\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)

In [30]: !pip install webdriver_manager

Requirement already satisfied: webdriver_manager in c:\users\bobb\appdata\roaming\python\python311\site-packages (4.0.1)

Requirement already satisfied: requests in c:\programdata\anaconda3\lib\site-packages (from webdriver_manager) (2.31.0)

Requirement already satisfied: python-dotenv in c:\programdata\anaconda3\lib\site-packages (from webdriver_manager) (0.21.0)

Requirement already satisfied: packaging in c:\programdata\anaconda3\lib\site-packages (from webdriver_manager) (23.1)

Requirement already satisfied: charset-normalizer<4,>=2 in c:\programdata\anaconda3\lib\site-packages (from requests->webdriver_manager) (2.0.4)

Requirement already satisfied: idna<4,>=2.5 in c:\programdata\anaconda3\lib\site-packages (from requests->webdriver_manager) (3.4)

Requirement already satisfied: urllib3<3,>=1.21.1 in c:\programdata\anaconda3\lib\site-packages (from requests->webdriver_manager) (2.0.7)

Requirement already satisfied: certifi>=2017.4.17 in c:\programdata\anaconda3\lib\site-packages (from requests->webdriver_manager) (2024.2.2)

```
In [42]: from selenium import webdriver
```

```
In [43]: from selenium.webdriver.chrome.service import Service as ChromeService  
from webdriver_manager.chrome import ChromeDriverManager
```

- 브라우저 열기 -

```
In [44]: driver = webdriver.Chrome(service=ChromeService(ChromeDriverManager().install()))
```

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/> 웹페이지에서 자세한 내용과 사용법을 확인하실 수 있습니다.
- 셀레니움은 브라우저를 자동화하는 툴로 테스트를 목적으로 웹 어플리케이션을 자동화하는 목적으로 만들어졌지만 테스트에 국한되지 않고 사용되지는 않습니다.
- 웹 기반의 작업들을 자동화하여 업무 효율성을 높일 수 있으며 다양한 언어와 다양한 브라우저를 지원합니다.
- <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 이동할수 있습니다.
- 아래의 코드는 naver웹 페이지로 이동합니다.
- url은 http를 꼭 포함해야 합니다

```
In [45]: 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 [46]: # (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 200 픽셀만큼의 브라우저 스크롤 위치를 이동함을 의미합니다.
- 두번째 파라미 터는 세로축의 위치를 나타내며 아래의 코드와 같이 300으로 지정을 하면 위에서 아래로 300픽셀 만큼 스크롤이 내려감을 의미합니다.

```
In [47]: 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 [48]: # alert 체크
try:
    alert = driver.switch_to.alert
    print(alert.text)
except:
    print('alert 없음')
```

alert 없음

```
In [49]: driver.execute_script("alert('selenium test');")
```

```
In [50]: # alert 체크
try:
    alert = driver.switch_to.alert
    print(alert.text)
except:
    print('alert 없음')
```

selenium test

```
In [51]: # alert 확인 버튼 누르기
alert.accept()
```

2.7 버튼 클릭하기

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

```
In [52]: from selenium.webdriver.common.by import By
```

```
In [53]: driver.find_element(By.CSS_SELECTOR, ".MyView-module__link_login__HpHMW").click()
```

2.8 윈도우 및 브라우저 종료하기

- selenium driver의 quit 함수를 이용하여 윈도우 창을 닫을수 있습니다

```
In [54]: # 브라우저 닫기  
driver.quit()
```

```
In [ ]:
```

```
In [ ]:
```