

Selenium Tutorial



I. Selenium 이란?

II. Selenium 및 ChromeDriver 설치

III. 웹 요소의 탐색과 조작

IV. 간단 실습 (홈페이지 게시글 크롤링)

V. 보안 및 예외 처리 사항

Selenium 이란?



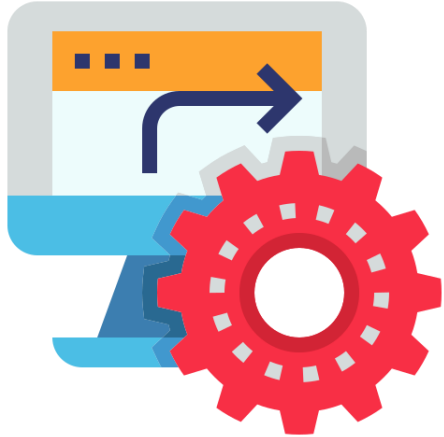
오픈 소스 웹 애플리케이션 테스트 도구로, 다양한 프로그래밍 언어를 지원하는 프레임워크이다.

브라우저에 대한 자동화를 지원하여 웹 애플리케이션의 동작을 테스트 할 수 있는 도구



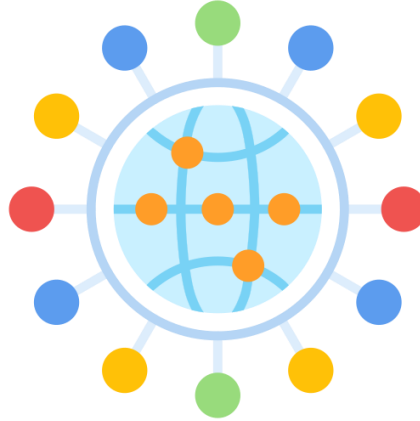
Selenium 이란?

Selenium의 대표적인 사용 사례



자동화 테스트

웹 애플리케이션을 테스트 하기 위해 사용자 시나리오를 자동으로 실행하여 버그를 식별하고 코드 변경에 따른 영향을 검증한다.



크로스 브라우징 테스트

다양한 브라우저와 플랫폼에서 웹 애플리케이션이 일관된 방식으로 동작하는지 확인합니다.



웹 스크래핑

웹 페이지에서 데이터를 수집하고 가공하는 데에도 사용된다.

Selenium은 앞선 사례의 활용 이 외에도 다양하게 활용 가능하다.

- 자동화 된 웹 태스크 수행
- 테스트 데이터 생성
- 웹 페이지의 스크린샷 및 동영상 녹화
- 웹 페이지의 성능 테스트

Selenium 및 ChromeDriver 설치

1. Python Selenium 라이브러리 설치

다음 명령어 중 하나를 이용해 Selenium 설치

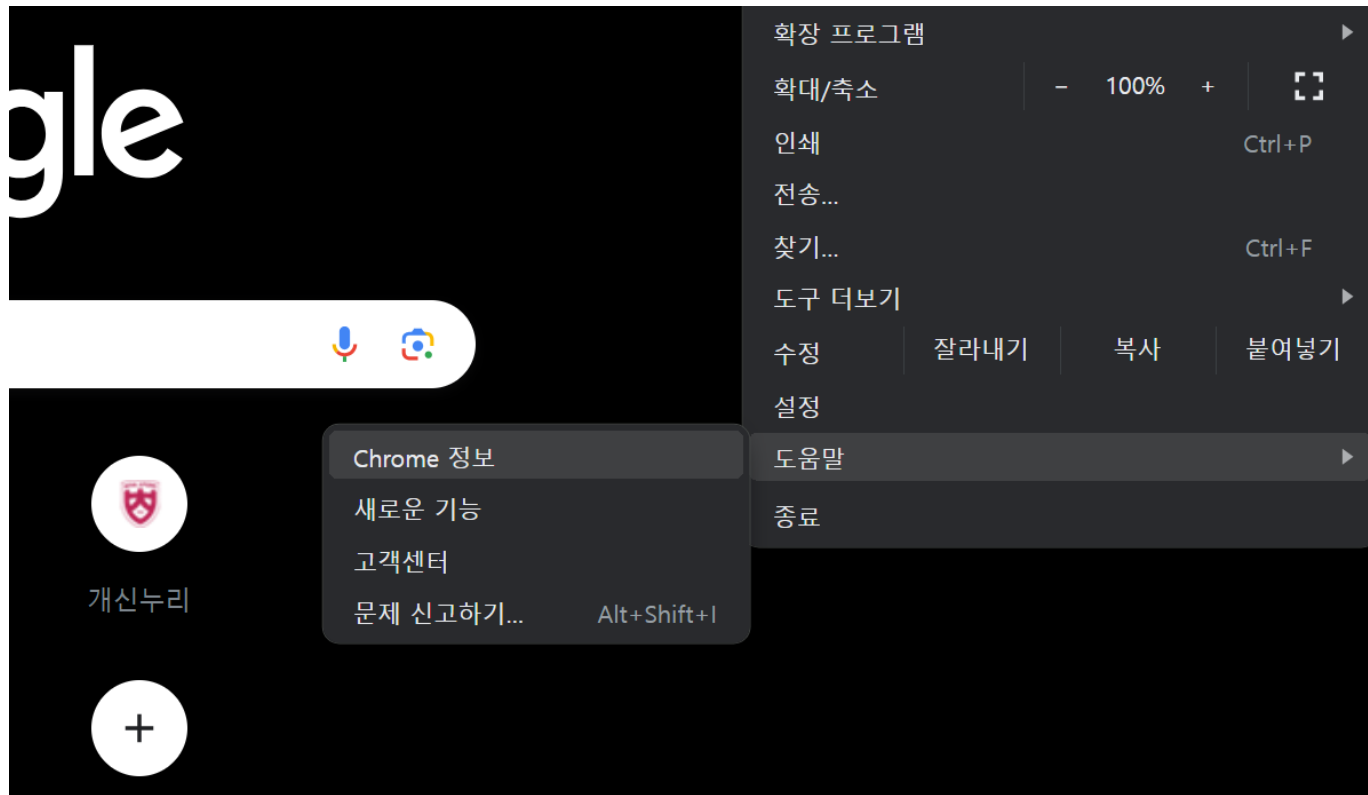
- > pip install selenium
- > conda install selenium

```
C:\Users\donggyu\Desktop\selenium>pip install selenium
Collecting selenium
  Obtaining dependency information for selenium from https://files.pythonhosted.org/packages/0e/59/aae37fa93e2d4292c3148efcc3066c8ecfe5cfaa72bf8c0b1a5614622cf7/selenium-4.15.2-py3-none-any.whl.metadata
Using cached selenium-4.15.2-py3-none-any.whl.metadata (6.9 kB)
Requirement already satisfied: urllib3[socks]<3,>=1.26 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from selenium) (1.26.13)
Requirement already satisfied: trio~=0.17 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from selenium) (0.23.1)
Requirement already satisfied: trio-websocket~=0.9 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from selenium) (0.11.1)
Requirement already satisfied: certifi>=2021.10.8 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from selenium) (2022.12.7)
Requirement already satisfied: attrs>=20.1.0 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from trio~=0.17->selenium) (23.1.0)
Requirement already satisfied: sortedcontainers in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from trio~=0.17->selenium) (3.4)
Requirement already satisfied: outcome in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from trio~=0.17->selenium) (1.3.0.post0)
Requirement already satisfied: sniffio>=1.3.0 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from trio~=0.17->selenium) (1.3.0)
Requirement already satisfied: cffi>=1.14 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from trio~=0.17->selenium) (1.15.1)
Requirement already satisfied: wsproto>=0.14 in c:\users\donggyu\appdata\local\programs\python\python311\lib\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:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from urllib3[socks]<3,>=1.26->selenium) (1.7.1)
Requirement already satisfied: pycparser in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from cffi>=1.14->trio~=0.17->selenium) (2.21)
Requirement already satisfied: h11<1,>=0.9.0 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from wsproto>=0.14->trio-websocket~=0.9->selenium) (0.14.0)
Using cached selenium-4.15.2-py3-none-any.whl (10.2 MB)
Installing collected packages: selenium
Successfully installed selenium-4.15.2

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

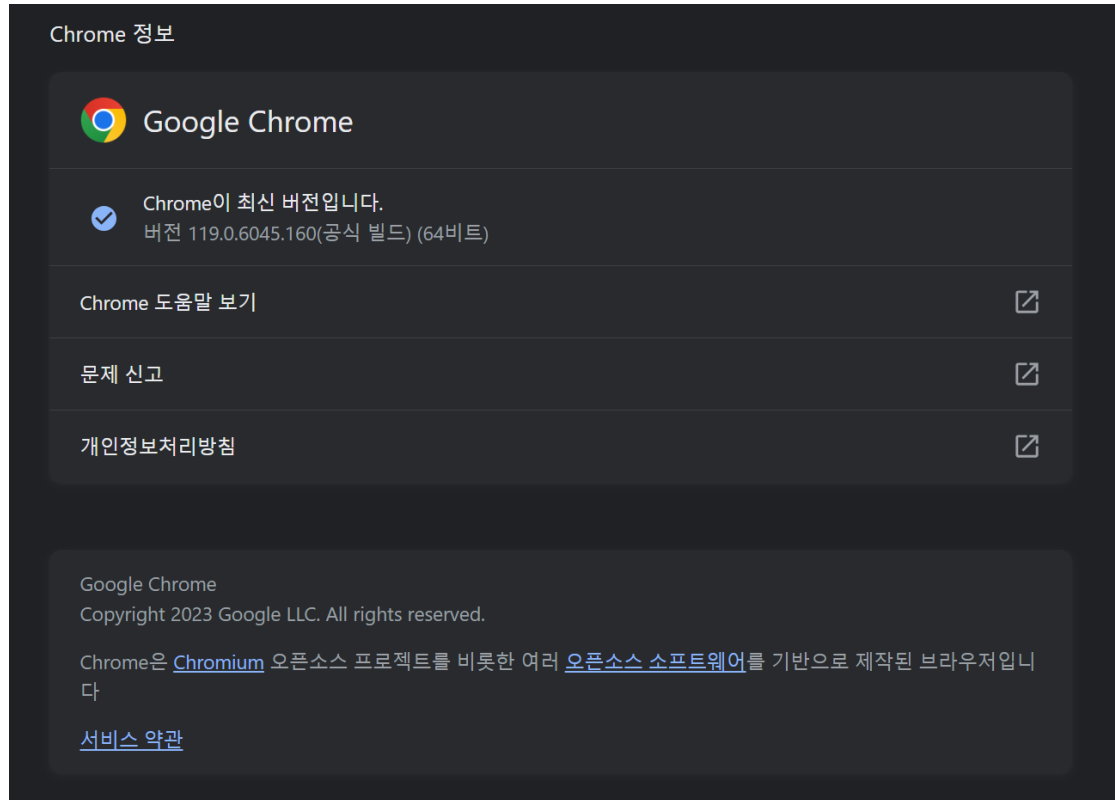

2. Chrome Driver 설치하기

1) 우측 점 세 개 -> 도움말 -> Chrome 정보



2. Chrome Driver 설치하기

2) Chrome 버전 확인



2. Chrome Driver 설치하기

3) 명령어를 통해 ChromeDriver 설치

> pip install webdriver_manager

```
C:\Users\donggyu\Desktop\selenium>pip install webdriver_manager
Collecting webdriver_manager
  Obtaining dependency information for webdriver_manager from https://files.pythonhosted.org/packages/b1/51/b5c11cf739ac4eecde611794a0ec9df420d0239d51e73bc19eb44f02b48b/webdriver_manager-4.0.1-py2.py3-none-any.whl.metadata
  Downloading webdriver_manager-4.0.1-py2.py3-none-any.whl.metadata (12 kB)
Requirement already satisfied: requests in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from webdriver_manager) (2.28.1)
Collecting python-dotenv (from webdriver_manager)
  Downloading python_dotenv-1.0.0-py3-none-any.whl (19 kB)
Requirement already satisfied: packaging in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from webdriver_manager) (23.1)
Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from requests->webdriver_manager) (2.1.1)
Requirement already satisfied: idna<4,>=2.5 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from requests->webdriver_manager) (3.4)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from requests->webdriver_manager) (1.26.13)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\donggyu\appdata\local\programs\python\python311\lib\site-packages (from requests->webdriver_manager) (2022.12.7)
Downloading webdriver_manager-4.0.1-py2.py3-none-any.whl (27 kB)
Installing collected packages: python-dotenv, webdriver_manager
Successfully installed python-dotenv-1.0.0 webdriver_manager-4.0.1

[notice] A new release of pip is available: 23.2.1 -> 23.3.1
[notice] To update, run: python.exe -m pip install --upgrade pip
```

3. 테스트 코드 실행

```
## test.py
import time
from selenium import webdriver
from selenium.webdriver.chrome.service import Service as ChromeService
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By

# 크롬 드라이버 생성
driver = webdriver.Chrome(service=ChromeService(
    ChromeDriverManager().install()))

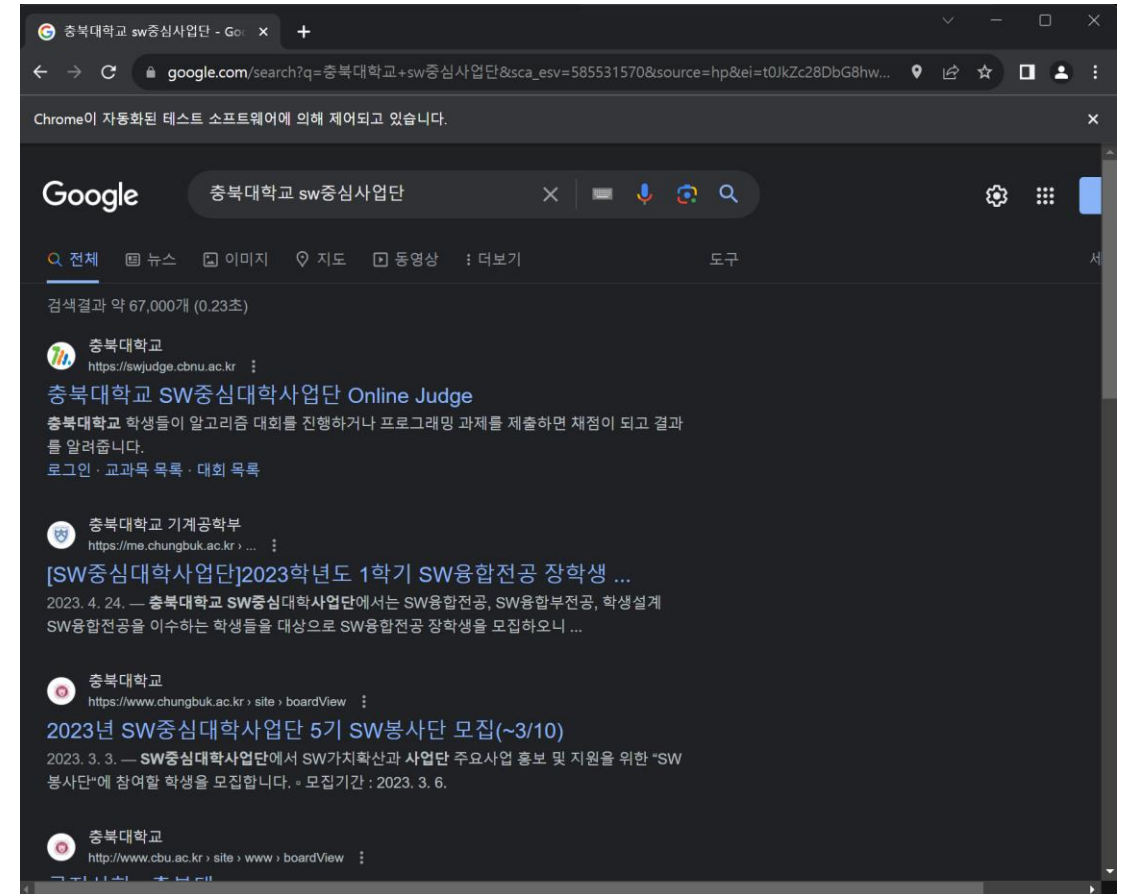
# Google 웹사이트에 접속
driver.get("https://www.google.com")

# 검색 입력창 찾기 (검색창의 이름이 'q')
search_box = driver.find_element(By.NAME, 'q')

# 검색어 입력
search_box.send_keys('충북대학교 sw중심사업단')

# 검색 실행
search_box.submit()
time.sleep(5)

# 브라우저 종료
driver.quit()
```



웹 요소의 탐색과 조작

웹 애플리케이션에 대한 자동화를 하려면, 웹 안의 요소를 탐색하고 조작할 수 있어야 한다.

Selenium에서 웹 요소를 선택하는 방법에는 “find_element” 방법이 있는데, 이는 버전에 따라 차이가 있다.

따라서 자신이 설치한 Selenium의 버전에 맞게 사용해야 한다.

selenium3

각 조건에 일치하는 첫 번째 요소를 찾는 메서드

<code>driver.find_element_by_class_name("")</code>	<code># class 이름으로 찾기</code>
<code>driver.find_element_by_id("")</code>	<code># id로 찾기</code>
<code>driver.find_element_by_css_selector("")</code>	<code># css selector로 찾기</code>
<code>driver.find_element_by_name("")</code>	<code># name으로 찾기</code>
<code>driver.find_element_by_tag_name("")</code>	<code># tag 이름으로 찾기</code>
<code>driver.find_element_by_xpath("")</code>	<code># xpath로 찾기</code>
<code>driver.find_element_by_link_text("")</code>	<code># link text</code>
<code>driver.find_element_by_partial_link_text("")</code>	<code># partial link text</code>

각 조건에 일치하는 모든 요소를 찾는 메서드(리스트로 반환)

`driver.find_elements_by~~`

selenium4

각 조건에 일치하는 첫 번째 요소를 찾는 메서드

```
driver.find_element(By.CLASS_NAME, "")
```

```
driver.find_element(By.ID, "")
```

```
driver.find_element(By.CSS_SELECTOR, "")
```

```
driver.find_element(By.NAME, "")
```

```
driver.find_element(By.TAG_NAME, "")
```

```
driver.find_element(By.XPATH, "")
```

```
driver.find_element(By.LINK_TEXT, "")
```

```
driver.find_element(By.PARTIAL_LINK_TEXT, "")
```

class 이름으로 찾기

id로 찾기

css 선택자로 찾기

name으로 찾기

tag 이름으로 찾기

xpath로 찾기

텍스트와 정확히 일치하는 링크 찾기

텍스트 일부분과 일치하는 링크 찾기

각 조건에 일치하는 모든 요소를 찾는 메서드(리스트로 반환)

```
driver.find_elements(By.~, "")
```

클릭 - click()

버튼 클릭 예제

```
button_element = driver.find_element(By.ID, "myButton")  
button_element.click()
```

텍스트 입력 - send_keys()

텍스트 필드에 텍스트 입력 예제

```
input_element = driver.find_element(By.NAME, "username")  
input_element.send_keys("my_username")
```


가져온 요소의 텍스트 추출 - get_text()

특정 요소의 텍스트 가져오기 예제

```
element = driver.find_element(By.ID, "myElement")  
element_text = element.text
```

가져온 요소의 속성 추출 - get_attribute()

특정 요소의 속성 가져오기 예제

```
element = driver.find_element(By.ID, "myElement")  
element_attribute_value = element.get_attribute("attribute_name")
```

예제 코드

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
import time

options = Options()
options.add_experimental_option("detach", True)
service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service, options=options)

url = "https://naver.com"

driver.get(url)

time.sleep(1)
```

다음 장의 코드 이어 붙이기 →

예제 코드

네이버 검색창 HTML 코드

"""

```
<input id="query" name="query" type="search" title="검색어를 입력해 주세요." placeholder="검색어를 입력해 주세요." maxlength="255"
autocomplete="off" class="search_input" data-atcmp-element="">
```

"""

class name을 이용한 검색

```
driver.find_element(By.CLASS_NAME, "search_input").send_keys("충북대학교")
```

```
time.sleep(1)
```

id를 이용한 검색

```
driver.find_element(By.ID, "query").send_keys("SW중심사업단")
```

```
time.sleep(1)
```

name을 이용한 검색

```
driver.find_element(By.NAME, "query").send_keys("소프트웨어학부")
```

```
time.sleep(1)
```

CSS Selector를 이용한 검색

```
driver.find_element(By.CSS_SELECTOR, "#query").send_keys("OSS")
```

```
time.sleep(1)
```

다음 장의 코드 이어 붙이기 →

예제 코드

네이버 검색창 HTML 코드

"""

```
<input id="query" name="query" type="search" title="검색어를 입력해 주세요." placeholder="검색어를 입력해 주세요." maxlength="255"
autocomplete="off" class="search_input" data-atcmp-element="">
```

"""

XPATH를 이용한 검색

```
driver.find_element(By.XPATH, '//*[@name="query"]').send_keys("운영보조원")
time.sleep(1)
```

Link Text를 이용한 검색

```
driver.find_element(By.LINK_TEXT, "뉴스스탠드").click()
time.sleep(1)
```

Partial Link Text를 이용한 검색

```
driver.find_element(By.PARTIAL_LINK_TEXT, "론사편").click()
time.sleep(1)
```

Class Name을 이용한 복수 선택

```
navs = driver.find_elements(By.CSS_SELECTOR, ".nav")
```

복수 선택한 요소들을 순회하며 출력

```
for nav in navs:
    print(nav.get_attribute("innerHTML"))
    print()
```

코드 마지막 부분

※ 코드 사용된 모듈 설명

```
from selenium import webdriver
```

=> Selenium 모듈

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

=> driver.find_element(By.ID, "query").send_keys("SW중심사업단") 에서 사용되는 By.ID 를 사용하기 위한 모듈

```
from selenium.webdriver.chrome.options import Options
```

=> Chrome 브라우저에 대한 옵션 모듈

```
from selenium.webdriver.chrome.service import Service
```

=> Chrome 드라이버 서비스를 위한 모듈

```
from webdriver_manager.chrome import ChromeDriverManager
```

=> Chrome 드라이버 관리자를 위한 모듈

```
import time
```

=> 대기 시간을 주기 위한 모듈

※ 드라이버 세팅 설명

```
options = Options()
```

```
options.add_experimental_option("detach", True)
```

=> Options 객체 생성(Chrome 브라우저의 옵션을 추가를 위해)

=> 브라우저를 닫아도 WebDriver 프로세스가 종료되지 않고 계속 실행되도록 하는 옵션

```
service = Service(ChromeDriverManager().install())
```

```
driver = webdriver.Chrome(service=service, options=options)
```

=> Chrome Driver 생성 및 세팅

```
url = https://naver.com
```

```
driver.get(url)
```

=> Driver를 지정한 url로 이동

```
time.sleep(1)
```

간단 실습

(홈페이지 게시물 크롤링)

1) 크롤링을 위한 Driver 세팅

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

from selenium.webdriver.chrome.options import Options
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
import time

options = Options()
options.add_experimental_option("detach", True)
service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service, options=options)
```


2) Driver를 웹 페이지로 이동

```
# 웹 페이지 열기
```

```
url = 'https://sw7up.cbnu.ac.kr/community/notice?page=1&limit=10&sort=-createdAt'
```

```
driver.get(url)
```

```
time.sleep(1)
```

3) 목록 요소를 찾아 가져오게 하기

공지사항 목록 가져오기

```
notice_elements = driver.find_elements(By.CSS_SELECTOR, '.notice-wrapper')
```

4) 가져온 목록에서 게시물 번호, 제목만 추출해서 출력

각 공지사항 출력

```
for index, notice_element in enumerate(notice_elements, start=1):  
    post_number = notice_element.find_element(By.CSS_SELECTOR, '.table-body.col-lg-1').text.strip()  
    post_title = notice_element.find_element(By.CSS_SELECTOR, '.title-container>span.mb-0').text.strip()  
    print(f'게시물 번호: {post_number}, 제목: {post_title}')
```

웹 드라이버 종료

```
driver.quit()
```

결과)

```
c:\Users\donggyu\Documents\pythonProject> c: && cd c:\Users\donggyu\Documents\pythonProject && cmd /C "C:\Users\donggyu\AppData\Local\Microsoft\Windows\Apps\python.exe c:\Users\donggyu\.vscode\extensions\ms-python.python-2023.20.0\pythonFiles\lib\python\debugpy\adapter\..\..\debugpy\launcher 60461 -- C:\Users\donggyu\Documents\pythonProject\pythonProject.py"
```

DevTools listening on ws://127.0.0.1:60473/devtools/browser/57dc228f-3c43-4cba-b844-fde051bb6bcd

게시물 번호: 1, 제목: [행사홍보] LG Aimers 프로그램 안내

게시물 번호: 2, 제목: 2023년 2학기 산학활동참여 마일리지 장학금 신청 (11/23~11/30)

게시물 번호: 3, 제목: SK실더스 2023년 동기 계절학기 및 2024년 1학기 학기제 표준 현장실습(인턴십) 참여학생 모집 (~11/29)

게시물 번호: 4, 제목: 2023년 웨스턴디지털코리아 하반기 신입사원 및 인턴십 채용(홍보)

게시물 번호: 5, 제목: SW전공트랙 이수 - 트랙 교과 확인

게시물 번호: 6, 제목: 2023년 SW중심대학사업단-KISTI 2차 HPC-AI 윈터스쿨 참가 학생 모집 (~12/11)

게시물 번호: 7, 제목: 2023학년도 2학기 마일리지 장학금 제출 안내(필독★)

게시물 번호: 8, 제목: 2023년도 2학기 STEP-Up 평가 멘토링 안내

게시물 번호: 9, 제목: 코딩다이어리 사용자 매뉴얼 안내

게시물 번호: 10, 제목: 2023학년도 2학기 마일리지 장학생 모집 안내 (신청일: 11/13~11/28)

보안 및 예외 처리 사항

Selenium을 통해 자동화 시나리오를 만들거나, 웹 크롤링을 할 때는 몇 가지 주의해야 되는 사항이 있습니다.

1. 크롤링 정책 및 robots.txt

웹 사이트는 robots.txt 파일을 통해 크롤러에 대한 접근 규칙을 명시할 수 있습니다.. 따라서 Selenium을 통해 타 사이트를 크롤링 할 때는 이 규칙을 준수 해야 합니다.

2. Headless 모드 사용

일부 웹 사이트는 브라우저 화면이 표시되지 않는 Headless 모드를 감지하고 브라우저를 표시하도록 요구할 수 있습니다. 이런 경우에는 User-Agent를 설정하거나, Selenium WebDriver의 옵션을 조정하여 Headless 모드를 사용하지 않도록 조정해야 합니다.

3. 웹 사이트의 로그인 및 세션 처리

웹 사이트에 로그인이 필요한 경우에는 사용자의 인증 정보(ex. id,pw)를 코드에 하드 코딩하면 안됩니다. 사용자 인증 정보에 대해 안전하게 저장하고, 가능하면 쿠키를 사용하여 세션을 관리하도록 해야 합니다.

4. 요청 간격과 부하 관리

웹 사이트는 과도한 요청으로 인한 부하를 방지하기 위해 일정한 요청 간격을 요구할 수 있습니다. 따라서 'time.sleep()' 등을 활용하여 적절한 요청 간격을 유지하도록 해야 합니다.

5. 예외 처리

WebDriver는 페이지가 로드되지 않거나, 요소가 찾아지지 않는 등 다양한 예외 상황이 나타날 수 있습니다. 따라서 'try'-'catch' 문 등을 사용하여 예외 처리를 구현하고, 오류에 대한 적절한 대응이 있어야 합니다.

THE END