

Selenium 시작하기

최희윤 강사

selenium 개요

✓ selenium이란?

- 웹 브라우저를 자동화 하기 위한 오픈소스 라이브러리
- 브라우저를 직접 제어하여 사용자의 행동(클릭, 입력, 탐색, 등)을 자동화할 수 있음
- 주로 웹 애플리케이션 테스트(테스트 자동화 구현)와 웹 크롤링에 사용 됨
- 다양한 언어와 호환 됨 (Python, Java, C#, JavaScript, 등)

✓ 특징

- 정적 페이지 뿐만 아니라 동적 웹 페이지 (JavaScript로 렌더링 되는 페이지)도 제어 가능
- Chrome, Firefox, Edge, Safari등 여러 브라우저 지원
- 다양한 플랫폼 지원(Windows, macOS, Linux, 등)
- 간단한 API로 직관적 ㄴ인 명령어를 제공하여 사용하기 쉬움

selenium 설치



pip install selenium

터미널을 열어서 가상환경을 확인한 후 설치해주세요.

selenium webdriver

✓ webdriver란?

- 웹 브라우저를 직접 제어할 수 있도록 설계된 API
- WebDriver는 브라우저에 직접 명령어를 전달하여 사람이 브라우저를 사용하는 것처럼 행동하게 함
- 브라우저를 열고, 웹페이지 탐색, 버튼 클릭, 텍스트 입력 등의 다양한 작업 자동화 가능

✓ 작동 원리

- selenium이 명령을 브라우저 드라이버에 전달 (Chrome브라우저를 제어하려면 ChromeDriver 필요)
- 브라우저 드라이버는 Selenium명령을 해석하여 해당 브라우저에 전달
- 브라우저 드라이버는 명령 실행 결과를 Selenium으로 반환

selenium webdriver

```
from selenium import webdriver
```

```
driver = webdriver.Chrome() # 크롬 웹드라이버를 실행하기 위해 webdriver객체 생성
```

```
driver.get('https://www.naver.com') # get()메서드를 통해 브라우저를 열고 특정 페이지로 이동
```

```
driver.quit() # 브라우저를 닫고 드라이버 프로세스 종료
```

접속 및 객체 셀렉터

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

driver = webdriver.Chrome()
driver.get('https://crawlingstudy-dd3c9.web.app/01/ ')

driver.find_element(By.CSS_SELECTOR, 'css_selector') # 제일 첫 태그 갖고 오기
driver.find_elements(By.CSS_SELECTOR, 'css_selector') # 모든 태그 갖고 와서 list로 return

element.text # 태그 내 텍스트
element.get_attribute('속성명') # 옵션(속성)
```

wait & sleep

- ✓ wait: driver.implicitly_wait(초)
 - 모든 요소 갖고 올 때 까지 기다리기 (요소나 브라우저 상태 체크)
 - 특정 요소가 로드되거나 나타날 때 까지 최대 지정된 시간만큼 기다림
 - 특정 요소가 클릭 가능할 때 까지나, 텍스트가 특정 요소에 표시될 때 까지 기다림
 - 조건이 충족되면 즉시 실행 재개 (명시된 시간을 무조건 채우지 않음)
 - 대기 시간 초과 시 예외 발생

- ✓ sleep: time.sleep(초)
 - 무조건 고정된 시간 동안 대기
 - 지정된 시간 동안 프로그램의 실행을 일시 중지
 - 브라우저나 요소의 상태 체크 없이 무조건 대기하기 때문에, 로드가 지연되면 오류가 발생할 수있음

wait & sleep

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

driver = webdriver.Chrome()
driver.implicitly_wait(2) # 2초동안 요소가 나타날 때 까지 대기

driver.get('https://crawlingstudy-dd3c9.web.app/01/')

time.sleep(3) # 3초동안 프로그램 일시 중지

driver.quit()
```


클릭

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

driver = webdriver.Chrome()
driver.implicitly_wait(2)

driver.get('https://crawlingstudy-dd3c9.web.app/01/')

a_tag = driver.find_element(By.CSS_SELECTOR, 'a') # 클릭할 대상인 a태그를 찾음
a_tag.click() # a태그에 해당하는 요소 클릭
time.sleep(3)

driver.quit()
```

키보드 입력

```
driver = webdriver.Chrome()
driver.implicitly_wait(2)

driver.get('https://www.naver.com') # 네이버에 로그인 해보기

a_tag = driver.find_element(By.CSS_SELECTOR, 'a.MyView-module__link_login__HpHMW')
a_tag.click() # 로그인 버튼 클릭
time.sleep(3) # 로그인 페이지가 열릴 때 까지 다음 코드가 실행되지 않도록 대기

id_tag = driver.find_element(By.CSS_SELECTOR, 'div.input_item.id input#id').send_keys('test')
pw_tag = driver.find_element(By.CSS_SELECTOR, 'div.input_item.pw input#pw').send_keys('test')
time.sleep(1)
login_tag = driver.find_element(By.CSS_SELECTOR, 'div.btn_login_wrap button.btn_login').click()
time.sleep(3) # 로그인 결과 확인해보기 (단순 확인용 sleep)

driver.quit()
```

스크롤 내리기

✓ 페이지 크기 갖고 오기 (스크롤 내릴 때 필요)

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

driver = webdriver.Chrome()
driver.implicitly_wait(2)

driver.get('https://www.musinsa.com/categories/item/001?device=mw')

height = driver.execute_script('return document.body.scrollHeight') # 페이지 크기 갖고 오기
print(height)

driver.quit()
```

스크롤 내리기

✓ 스크롤 내리기

```
# 스크롤 세 번 내리기
for i in range(3):
    # 0에서 페이지 마지막 길이까지
    driver.execute_script('window.scrollTo(0, document.body.scrollHeight);')
    time.sleep(2)
    new_height = driver.execute_script('return document.body.scrollHeight')
    print(new_height)

driver.quit()
```

Page source 갖고 오기

- ✓ selenium으로 html 소스 갖고오기

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

driver = webdriver.Chrome()
driver.get('https://crawlingstudy-dd3c9.web.app/01/')

response = driver.page_source # page_source 속성을 통해 html 소스 갖고 올 수 있음
print(response)

driver.quit()
```

Selenium 실습

최희운 강사

실습 1

✓ 사이트내에 인기검색종목과 주요해외지수를 각각 크롤링하여 **종목명**과 **주가지수**를 아래와 같이 리스트로 정리해주세요

결과:

['씨니전자', '5,000'], ['삼성전자', '55,200'], ['안랩', '81,000'], ['케이엠더블.', '57,300'], ['피피아이', '12,600'],
['KT&G', '92,500'], ['삼성전자우', '45,600'], ['대양금속', '10,550'], ['SK하이닉스', '94,700'], ['SK텔레콤', '234,000']]

['다우산업', '28,647.43'], ['나스닥', '9,015.03'], ['홍콩H', '11,320.56'], ['상해종합', '3,085.20'], ['니케이225', '23,656.62']]

실습 2

- ✓ Selenium으로 **비동기 웹 크롤링** 해보기
- ✓ 아래 사이트를 크롤링하여 아래와 같이 각각 글에 **id**와 **title** 그리고 글마다 **코멘트내용**을 리스트형식으로 담고 최종 json 파일 형태로 저장해보세요.

결과 :

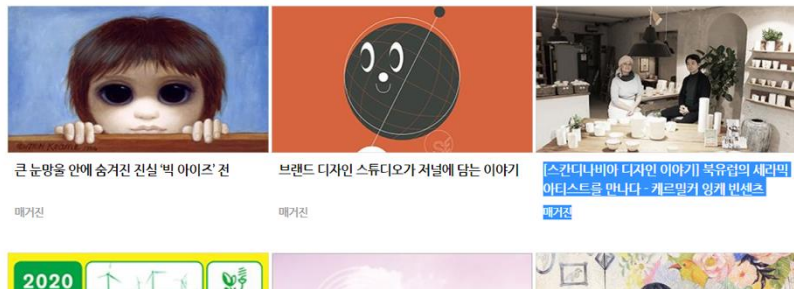
```
[{'id': 1,
  'title': 'sunt aut facere repellat provident occaecati excepturi optio reprehenderit',
  'comment': [
    'laudantium enim quasi est quidem magnam voluptate ipsam eos\ntempora quo necessitatibus\ndolor quam
    autem quasi\nreiciendis et nam sapiente accusantium',
    'est natus enim nihil est dolore omnis voluptatem numquam\net omnis occaecati quod ullam at\involuptatem
    error expedita pariatur\nnihil sint nostrum voluptatem reiciendis et',
    ... ]},
 {'id': 2
  ...
}]
```


실습 3

✓ 아래 사이트의 메인페이지

인피니티 스크롤 크롤링의 제목, 카테고리 10페이지 크롤링하여 아래 결과와 같이 출력 후, json 파일로 저장

<https://www.jungle.co.kr/>



결과 :

[큰 눈망울 안에 숨겨진 진실 '빅 아이즈' 전](#)
매거진

[브랜드 디자인 스튜디오가 저널에 담는 이야기](#)
매거진

[\[스칸디나비아 디자인 이야기\] 북유럽의 세라믹 아티스트를 만나다 - 케르밀커 잉케 빈센츠](#)
매거진