셀레니움 실습(동적)을 통해 알아보기

학습 내용

(1) Selenium은 무엇인가요?

- Selenium은 웹 크롤링에 많이 사용되는 도구 중 하나입니다.
- 동적 웹 페이지 처리: Selenium은 JavaScript로 동적으로 생성되는 웹 페이지를 처리할 수 있습니다. 이는 단순한 HTML 파싱으로는 어려운 작업을 가능하게 합니다.
- 다양한 브라우저 지원: Selenium은 Chrome, Firefox, Safari, Edge 등 다양한 브라우저를 지원하므로, 크로스 브라우저 테스팅 및 크롤링이 가능합니다.

사전 준비

설치

```
pip install selenium
pip install webdriver-manager
```

01. 요소 찾기

- Selenium은 웹 페이지에서 요소를 찾기 위해 다음과 같은 다양한 방법을 제공합니다
- 참조 URL : https://selenium-python.readthedocs.io/locating-elements.html

하나의 DOM(객체)에 접근 - element

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

find_element(By.ID, "element_id"): 요소의 고유 ID로 찾기
find_element(By.NAME, "element_name"): 요소의 name 속성으로 찾기
find_element(By.XPATH, "//div[@class='element']"): XPath 표현식으로 찾기
```

find_element(By.LINK_TEXT, "Link Text"): 링크의 가시적인 텍스트로 찾기
find_element(By.PARTIAL_LINK_TEXT, "Partial Link"): 링크의 부분 텍스트로 찾기
find_element(By.TAG_NAME, "div"): HTML 태그 이름으로 찾기
find_element(By.CLASS_NAME, "element_class"): CSS 클래스 이름으로 찾기
find_element(By.CSS_SELECTOR, "div.element"): CSS 선택자로 찾기

여러개의 DOM(객체)에 접근 - elements

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

find_elements(By.NAME, "name"): 요소의 name 속성으로 여러 개의 요소를 찾습니다.
find_elements(By.XPATH, "xpath"): XPath 표현식을 사용하여 여러 개의 요소를 찾습니다.
find_elements(By.LINK_TEXT, "text"): 링크의 가시적인 텍스트로 여러 개의 요소를 찾습니다.
find_elements(By.PARTIAL_LINK_TEXT, "text"): 링크의 부분 텍스트로 여러 개의 요소를 찾습니다.
find_elements(By.TAG_NAME, "tag"): HTML 태그 이름으로 여러 개의 요소를 찾습니다.
find_elements(By.CLASS_NAME, "class"): CSS 클래스 이름으로 여러 개의 요소를 찾습니다.
find_elements(By.CSS_SELECTOR, "css"): CSS 선택자를 사용하여 여러 개의 요소를 찾습니다.

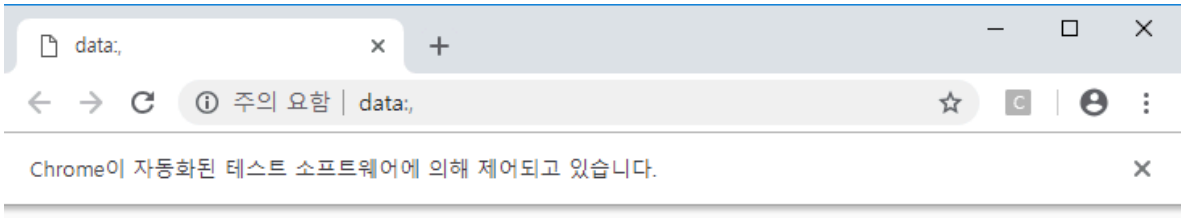
(1) 웹 브라우저를 자동으로 띄워보기

In [ ]:
```
# 작업에 필요한 패키지를 불러옵니다
from selenium import webdriver
from selenium.webdriver.chrome.service import Service as ChromeService
from webdriver_manager.chrome import ChromeDriverManager
```

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

In [22]:
```
url = "https://ldjwj.github.io/webPage/"
driver.get(url)
```

In [20]:
```
path = "../img/selenium01.png"
display(Image.open(path))
```



(2) 웹 페이지에 접속해 보기

- 웹 브라우저를 띄우고 크롤러를 만들기 위한 사이트에 접속
- get() 함수를 이용하면 지정된 url를 이용하여 사이트 접속이 가능

In [21]:
```
url = 'https://ldjwj.github.io/webPage/'
driver.get(url) # url 접속
```

(3) 일부 정보 id를 이용하여 가져오기

- 요소(element)의 id의 속성(attribute)를 알 때, 사용.
- find_element(By.ID, "id명"): 요소의 고유 ID 속성을 알 때 사용합니다. 이 메서드는 ID 값과 일치하는 첫 번째 요소를 반환합니다.
- find_elements(By.ID, "id명"): 요소의 고유 ID 속성을 알 때 사용합니다. 이 메서드는 ID 값과 일치하는 모든 요소를 리스트로 반환합니다.

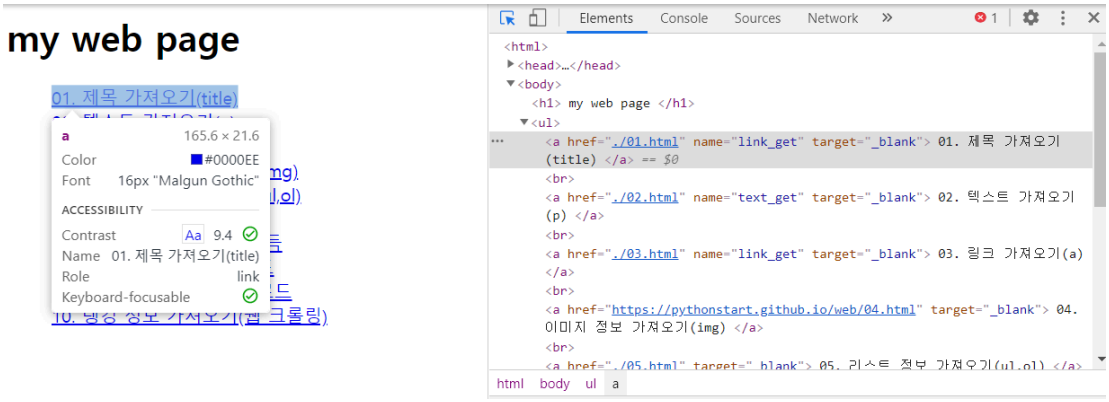
In [9]:
```
url = 'https://ldjwj.github.io/webPage/'
driver.get(url)
```

In [23]:
```
selected_id = driver.find_element(By.ID, 'rank')
print(selected_id)
print(selected_id.tag_name)  # 해당 요소의 태그 이름
print(selected_id.text)      # 해당 요소의 텍스트 정보
```

```
<selenium.webdriver.remote.webelement.WebElement (session="c9b318ebcedcd7dc79d194
1f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.A8CFFFCB0A785633E07E60
8CE7E4D0D7.e.2")>
a
10. 랭킹 정보 가져오기(웹 크롤링)
```

(4) 태그이름을 이용하여 접근

find_element(By.TAG_NAME, '태그명'): 태그 이름이 '태그명'인 첫 번째 요소를 찾습니다.
find_elements(By.TAG_NAME, '태그명'): 태그 이름이 '태그명'인 모든 요소를 찾습니다.



In [24]:
```
from selenium import webdriver

url = 'https://ldjwj.github.io/webPage/'
driver.get(url)

selected_tag_h1 = driver.find_element(By.TAG_NAME, 'h1')
print(selected_tag_h1)
print(selected_tag_h1.tag_name)
print(selected_tag_h1.text)
```

```
<selenium.webdriver.remote.webelement.WebElement (session="c9b318ebcedcd7dc79d1941f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE9A340F2C276E5C6E6434C6E87F36C2.e.4")>
h1
my web page
```

In [25]:
```
from selenium.webdriver.common.by import By

## 전체 a태그 정보 가져오기
# selected_tags_a = driver.find_elements_by_tag_name('a')
selected_tag_a = driver.find_elements(By.TAG_NAME, 'a')
print(selected_tag_a)
```

```
[<selenium.webdriver.remote.webelement.WebElement (session="c9b318ebcedcd7dc79d19
41f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE9A340F2C276E5C6E643
4C6E87F36C2.e.5")>, <selenium.webdriver.remote.webelement.WebElement (session="c9
b318ebcedcd7dc79d1941f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE
9A340F2C276E5C6E6434C6E87F36C2.e.6")>, <selenium.webdriver.remote.webelement.WebE
lement (session="c9b318ebcedcd7dc79d1941f88f35b41", element="f.E9F922E055FFC68C99
BA7D6F17DD25CE.d.DE9A340F2C276E5C6E6434C6E87F36C2.e.7")>, <selenium.webdriver.rem
ote.webelement.WebElement (session="c9b318ebcedcd7dc79d1941f88f35b41", element
="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE9A340F2C276E5C6E6434C6E87F36C2.e.8")>, <
selenium.webdriver.remote.webelement.WebElement (session="c9b318ebcedcd7dc79d1941
f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE9A340F2C276E5C6E6434C
6E87F36C2.e.9")>, <selenium.webdriver.remote.webelement.WebElement (session="c9b3
18ebcedcd7dc79d1941f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE9A
340F2C276E5C6E6434C6E87F36C2.e.10")>, <selenium.webdriver.remote.webelement.WebEl
ement (session="c9b318ebcedcd7dc79d1941f88f35b41", element="f.E9F922E055FFC68C99B
A7D6F17DD25CE.d.DE9A340F2C276E5C6E6434C6E87F36C2.e.11")>, <selenium.webdriver.rem
ote.webelement.WebElement (session="c9b318ebcedcd7dc79d1941f88f35b41", element
="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE9A340F2C276E5C6E6434C6E87F36C2.e.12")>,
<selenium.webdriver.remote.webelement.WebElement (session="c9b318ebcedcd7dc79d194
1f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE9A340F2C276E5C6E6434
C6E87F36C2.e.13")>, <selenium.webdriver.remote.webelement.WebElement (session="c9
b318ebcedcd7dc79d1941f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.DE
9A340F2C276E5C6E6434C6E87F36C2.e.14")>]
```

(5) name을 이용하여 접근

- find_element(By.NAME, "이름"): 요소의 name 속성이 "이름"인 첫 번째 요소를 찾습니다.
- find_elements(By.NAME, "이름"): 요소의 name 속성이 "이름"인 모든 요소를 찾습니다.

In [26]:
```
from selenium import webdriver
from selenium.webdriver.common.by import By

url = 'https://ldjwj.github.io/webPage/'
driver.get(url)

selected_name = driver.find_element(By.NAME, 'text_get')
print(selected_name)           # WebElement 객체 확인
print(selected_name.tag_name)  # 태그 이름 확인

selected_names = driver.find_elements(By.NAME, 'link_get')
print(len(selected_names))     # name이 'link_get'인 요소가 하나이므로 길이가 1인
```

```
<selenium.webdriver.remote.webelement.WebElement (session="c9b318ebcedcd7dc79d194
1f88f35b41", element="f.E9F922E055FFC68C99BA7D6F17DD25CE.d.375B43B6472AF4E494275D
71A61B6548.e.16")>
a
2
```

(6) 클래스 이름을 이용한 찾기

- find_element(By.CLASS_NAME, 'class_name'): 페이지에서 지정된 class 이름을 가진 첫 번째 요소를 찾습니다.

- find_elements(By.CLASS_NAME, 'class_name'): 페이지에서 지정된 class 이름을 가진 모든 요소를 찾습니다.

(7) selector를 이용한 접근

- find_element(By.CSS_SELECTOR, 'css_선택자'): 이 메서드는 페이지에서 지정된 CSS 선택자와 일치하는 첫 번째 요소를 찾습니다.
- find_elements(By.CSS_SELECTOR, 'css_선택자'): 이 메서드는 페이지에서 지정된 CSS 선택자와 일치하는 모든 요소를 찾습니다.

```
<html>
 <body>
  <p class="content">Content 부분</p>
</body>
<html>
```

In [27]:
```
url = 'https://ldjwj.github.io/webPage/'
driver.get(url)

content = driver.find_element(By.CSS_SELECTOR, 'body ul a#rank')
print(content.text)
```

```
10. 랭킹 정보 가져오기(웹 크롤링)
```

(8) Link Text를 이용하여 접근하기

- a태그(anchor tag)의 link text로 접근하려고 할때 사용.

```
<html>
 <body>
  <p>안녕하세요!</p>
  <a href="continue.html">Continue</a>
  <a href="cancel.html">Cancel</a>
</body>
<html>
```

find_element(By.LINK_TEXT, '')
find_element(By.PARTIAL_LINK_TEXT, '')

In [28]:
```
url = 'https://ldjwj.github.io/webPage/'
driver.get(url)

# '03. 링크 가져오기(a)'라는 링크 텍스트를 가진 요소를 찾습니다.
continue_link = driver.find_element(By.LINK_TEXT, '03. 링크 가져오기(a)')
print(continue_link.text)
```

```
03. 링크 가져오기(a)
```

(9) 웹 제어하기

- 마우스 제어
- 키보드 제어
- JavaScript 삽입

9-1 마우스 제어

- 뉴스 페이지 이동하여 뉴스 내용 중에 '패션'이라는 단어로 검색해 보기
- 입력 : send_keys() 함수를 이용
- 선택 : click() 함수를 이용

In [29]:
```
from selenium import webdriver
from selenium.webdriver.common.by import By

url = 'https://news.naver.com/'

# 웹 드라이버를 사용하여 지정된 URL로 이동합니다.
driver.get(url)
```

In [30]:
```
from selenium.webdriver.common.by import By

# 검색 아이콘 요소 찾기
search_icon = driver.find_element(By.XPATH, '/html/body/section/header/div[1]/di
print(search_icon.tag_name)
print(search_icon.text)
search_icon.click()

# 검색창 요소 찾기
search_input = driver.find_element(By.XPATH, '//*[@id="u_hs"]/div/div/input')
print(search_input.tag_name)
print(search_input.text)

# 검색 버튼 요소 찾기
search_button = driver.find_element(By.XPATH, '//*[@id="u_hs"]/div/div/button[2]
print(search_button.tag_name)
print(search_button.text)

# 검색어 입력 및 검색 실행
search_input.send_keys("패션")
search_button.click()
```

```
span
검색
input

button
뉴스검색
```

(10) 여러개의 정보를 xpath를 이용해서 가져와 보기

In [32]:
```
from selenium.webdriver.common.by import By

# 웹 페이지 URL 접속
```

```
url = 'https://ldjwj.github.io/webPage/'
driver.get(url)
```

In [33]:
```
base_xpath = '/html/body/ul/a['
end_xpath = ']'

for i in range(1,10,1):
    one_xpath = base_xpath + str(i) + end_xpath
    data = driver.find_element(By.XPATH, one_xpath)
    print(data.text)
```

01. 제목 가져오기(title)
02. 텍스트 가져오기(p)
03. 링크 가져오기(a)
04. 이미지 정보 가져오기(img)
05. 리스트 정보 가져오기(ul,ol)
06. id를 활용한 정보 획득
07. class를 활용한 정보 획득
08. 하나의 이미지 다운로드
09. 여러개의 이미지 다운로드

history

- 2024/11/01 최신 내용으로 적용