

# 셀레니움 실습(동적)을 통해 알아보기

## 학습 내용

- (1) [웹 브라우저를 자동으로 띄워보기](#)
- (2) [웹 페이지에 접속해 보기](#)
- (3) [id를 이용하여 웹 페이지 정보 가져오기](#)
- (4) [태그이름을 이용하여 접근하기](#)
- (5) [name을 이용하여 접근하기](#)
- (6) [클래스 이름을 이용한 찾기](#)
- (7) [selector를 이용한 접근하기](#)
- (8) [Link Text를 이용하여 접근하기](#)
- (9) [웹 제어\(마우스 클릭, 텍스트 입력\)](#)
- (10) [과제 실습해 보기\(스타일 난다\)](#)

## 01. 기본 함수

- 참조 URL : <https://selenium-python.readthedocs.io/locating-elements.html> (<https://selenium-python.readthedocs.io/locating-elements.html>)

## 하나의 DOM(객체)에 접근 - element

```
find_element_by_id      : id값을 이용하여 접근
find_element_by_name    : 속성 name를 이용하여 접근
find_element_by_xpath   : xpath를 이용하여 접근
find_element_by_link_text : a태그의 href를 이용하여 수집(쿼리 스트링 부분)
find_element_by_partial_link_text : a태그의 href를 이용하여 수집(path를 이용)
find_element_by_tag_name : 태그이름을 이용하여 접근
find_element_by_class_name : 클래스 이름을 이용하여 접근
find_element_by_css_selector : css selector를 이용하여 접근
```

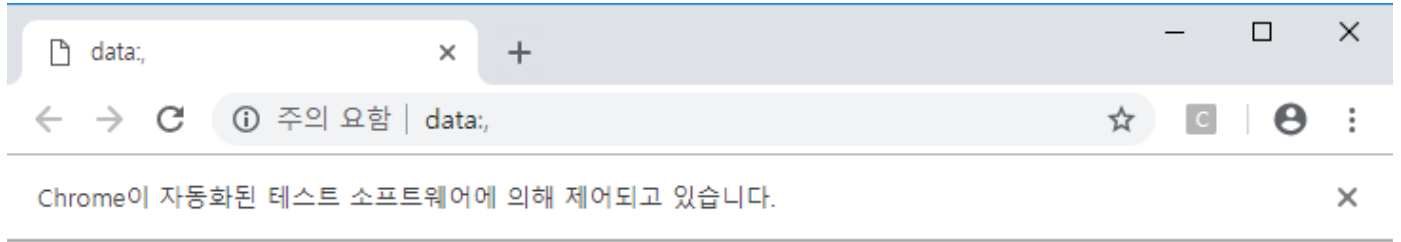
## 여러개의 DOM(객체)에 접근 - elements

```
find_elements_by_name
find_elements_by_xpath
find_elements_by_link_text
find_elements_by_partial_link_text
find_elements_by_tag_name
find_elements_by_class_name
find_elements_by_css_selector
```

## (1) 웹 브라우저를 자동으로 띄워보기

In [1]:

```
from selenium import webdriver  
  
driver = webdriver.Chrome('chromedriver_246')
```



## (2) 웹 페이지에 접속해 보기

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

In [2]:

```
url = 'https://pythonstart.github.io/web/'  
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")
  - id의 값과 일치하는 첫번째 요소를 가지고 온다.
- find\_elements\_by\_id("id명")
  - id의 값과 일치하는 전체 요소를 가지고 온다.

In [3]:

```
url = 'https://pythonstart.github.io/web/'
driver.get(url)
```

In [5]:

```
selected_id = driver.find_element_by_id('rank')
print(selected_id)
print(selected_id.tag_name) # 해당 연결의 tag이름
print(selected_id.text)    # 해당 태그의 text정보
```

```
<selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.5156981997915995-1")>
```

a

10. 랭킹 정보 가져오기(웹 크롤링)

#### (4) 태그이름을 이용하여 접근

```
find_element_by_tag_name("태그명") # 하나
find_elements_by_tag_name("태그명") # 여러개
```

### my web page

01. 제목 가져오기(title)

02. 텍스트 가져오기(text)

03. 링크 가져오기(a)

04. 이미지 정보 가져오기(img)

05. 리스트 정보 가져오기(ul,ol)

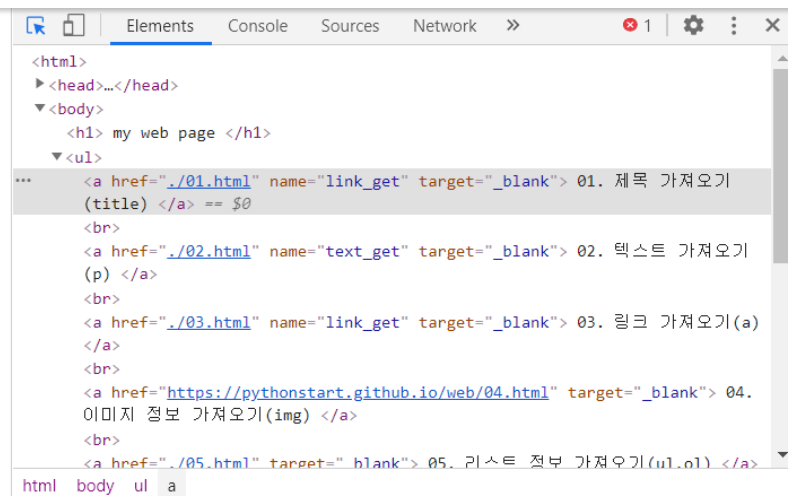
06. 텍스트 가져오기(p)

07. 텍스트 가져오기(div)

08. 텍스트 가져오기(h1)

09. 텍스트 가져오기(h2)

10. 랭킹 정보 가져오기(웹 크롤링)



In [6]:

```
from selenium import webdriver

url = 'https://pythonstart.github.io/web/'
# driver = webdriver.Chrome('chromedriver')
# driver = webdriver.Chrome('chromedriver_80') _80은 버전을 의미
driver.get(url)

selected_tag_p = driver.find_element_by_tag_name('h1')
print(selected_tag_p)
print(selected_tag_p.tag_name)
print(selected_tag_p.text)
```

```
<selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-1")>
h1
my web page
```

In [7]:

```
## 전체 a태그 정보 가져오기
selected_tags_p = driver.find_elements_by_tag_name('a')
print(selected_tags_p)
```

```
[<selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-2")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-3")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-4")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-5")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-6")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-7")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-8")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-9")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-10")>, <selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.019464183444975358-11")>]
```

## (5) name을 이용하여 접근

- find\_element\_by\_name("이름") # 하나
- find\_elements\_by\_name("이름") # 여러개

In [8]:

```
## meta 태그 - name='description' 가져오기
from selenium import webdriver

url = 'https://pythonstart.github.io/web/'
# driver = webdriver.Chrome('chromedriver')
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

# find_elements_by_name 이용해 보기
# name이 'q'인것이 하나이기에 길이가 1인 리스트가 반환됨.
selected_names = driver.find_elements_by_name('link_get')
print(len(selected_names))
```

```
<selenium.webdriver.remote.webelement.WebElement (session="94c558e4c75e6a1494f04f76f6ab7108", element="0.5803408285373712-1")>
a
2
```

## (6) 클래스 이름을 이용한 찾기

- find\_element\_by\_class\_name과
- find\_elements\_by\_class\_name

## (7) selector를 이용한 접근

- find\_element\_by\_css\_selector과
- find\_elements\_by\_css\_selector

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

```
content = driver.find_element_by_css_selector('p.content')
```

In [9]:

```
url = 'https://pythonstart.github.io/web/'
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 접근

- 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>
```

```
continue_link = driver.find_element_by_link_text('Continue')
continue_link = driver.find_element_by_partial_link_text('Conti')
```

In [10]:

```
url = 'https://pythonstart.github.io/web/'
driver.get(url)

# link_test라는 링크를 가져온다.
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

- 뉴스를 선택해서 해당 페이지로 이동해 보기

In [11]:

```

from selenium import webdriver

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

# driver = webdriver.Chrome('chromedriver')
driver.get(url)

# //*[@id="today_main_news"]/div[2]/ul/li[1]/div[1]/a
selected_selector = driver.find_element_by_xpath('//*[@id="today_main_news"]/div[2]/ul/li[1]/div[1]/a')
print(selected_selector.tag_name)
print(selected_selector.text)
selected_selector.click()

```

a

“보좌관 전화 건 건 사실인 듯” …野 “추미애 사퇴하고 특검해야”

## 실습 1. 네이버 뉴스에서 원하는 기사를 검색 후, 해당 페이지로 이동하기

### 9-2 키보드 제어

- send\_keys() 함수를 이용

In [12]:

```

from selenium import webdriver

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

# driver = webdriver.Chrome('chromedriver')
driver.get(url)

# //*[@id="lnb.searchForm"]/fieldset/input[1] # 검색 내용
# //*[@id="lnb.searchForm"]/fieldset/button/span # 검색
# text_today_main_news_801001 > li:nth-child(1) > div > a > strong
selected_xpath = driver.find_element_by_xpath('//*[@id="lnb.searchForm"]/fieldset/input[1]')
selected_xpath_search = driver.find_element_by_xpath('//*[@id="lnb.searchForm"]/fieldset/button/span')
selected_xpath.send_keys('미세먼지')
#print(selected_xpath.tag_name)
print(selected_xpath.text)
selected_xpath_search.click()

```

Copyright 2020 LIM Co.(예영Edu Co.) all rights reserved.

교육용으로 작성된 것으로 배포 및 복제시에 사전 허가가 필요합니다.

### history

- 2020/09 v14 : 정보 업데이트, 일부 개정

In [ ]: