

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\(마우스 클릭, 텍스트 입력\)](#)

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 name을 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

In [4]:

```
from IPython.display import display
from PIL import Image
```



(1) 웹 브라우저를 자동으로 띄워보기

In [34]:

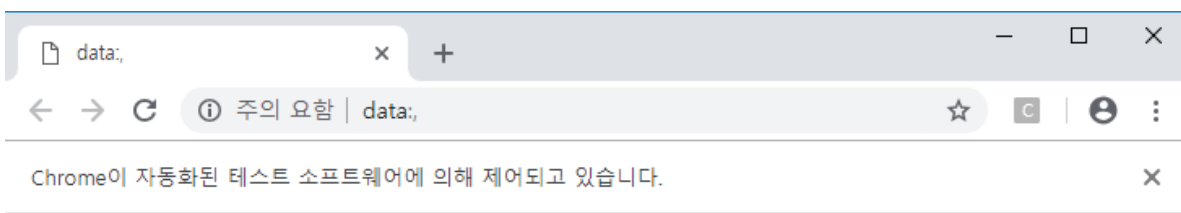
```
from selenium import webdriver

driver = webdriver.Chrome('../chromedriver_100') # 최신 버전 활용
```

<ipython-input-34-9e7b5e90302b>:3: DeprecationWarning: executable_path has been deprecated, please pass in a Service object
driver = webdriver.Chrome('../chromedriver_100') # 최신 버전 활용

In [6]:

```
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 [8]:

```
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 [9]:

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

In [10]:

```
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="bb7ead925815d301885d25065
2534be0", element="cd4c7c66-516c-419c-9ae2-9753496184fa")>
```

a

10. 랭킹 정보 가져오기(웹 크롤링)

```
<ipython-input-10-951e7883f22d>:1: DeprecationWarning: find_element_by_* commands ar
e deprecated. Please use find_element() instead
selected_id = driver.find_element_by_id('rank')
```

(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)

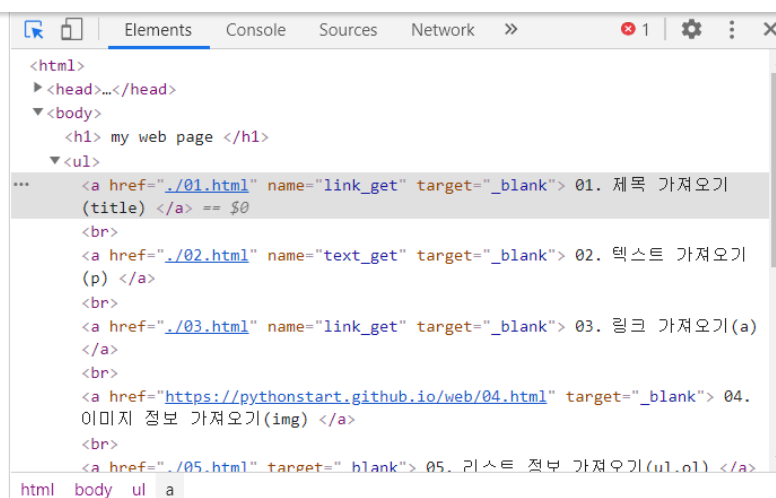
02. 텍스트 가져오기(text)

03. 링크 가져오기(a)

04. 이미지 정보 가져오기(img)

05. 리스트 정보 가져오기(ul,ol)

10. 랭킹 정보 가져오기(웹 크롤링)



In [17]:



```
from selenium import webdriver

url = 'https://pythonstart.github.io/web/'
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="bb7ead925815d301885d250652534be0", element="3c276fdc-1de5-4428-92f2-0def94ff8077")>
h1
my web page
```

```
<ipython-input-17-e827c5126619>:6: DeprecationWarning: find_element_by_* commands are deprecated. Please use find_element() instead
  selected_tag_p = driver.find_element_by_tag_name('h1')
```

In [18]:



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

In [21]:



```
## 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="bb7ead925815d301885d250652534be0", element="a7215a4a-c2d3-4faf-8ed8-a151df5a0c7e")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="80f7225b-69e2-4111-a1eb-95e42ba09906")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="d93676f3-a2d0-4197-bda3-96585935a0ad")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="c009efd1-f531-49a1-87cc-de2a05f635b3")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="7cda69cd-b4eb-4eb2-8d42-0d5f3bbd1404")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="d8a0e719-8dfa-49e7-922b-32c4312ef472")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="ec74e332-58ff-4e07-a490-4a97cbd21fb0")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="9ca521b1-f663-4f33-878c-9488413e15d1")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="a548fb41-c22b-4afb-ad7c-5282b1d2faec")>, <selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d250652534be0", element="be0e3846-ddba-47cb-be80-7616a3a70bda")>]
```

In [35]:



```
<h2><a id='5'><strong>(5) name을 이용하여 접근</strong></a></h2>
```

```
File "<ipython-input-35-19862e77afc8>", line 1
  <h2><a id='5'><strong>(5) name을 이용하여 접근</strong></a></h2>
  ^
```

SyntaxError: invalid syntax

(5) name을 이용하여 접근

- find_element_by_name("이름") or find_element(By.NAME) # 하나
- find_elements_by_name("이름") or find_elements(By.NAME) # 여러개

In [22]:



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

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

#selected_name = driver.find_element_by_name('text_get')
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')
selected_names = driver.find_elements(By.NAME, 'link_get')
print(len(selected_names))
```

```
<selenium.webdriver.remote.webelement.WebElement (session="bb7ead925815d301885d25065
2534be0", element="080db55a-ba8a-4707-896c-404ef3bca53a")>
```

```
a
2
```

(6) 클래스 이름을 이용한 찾기

- find_element_by_class_name or find_element(By.CLASS_NAME, "")
- find_elements_by_class_name or find_elements(By.CLASS_NAME, "")

(7) selector를 이용한 접근

- find_element_by_css_selector or find_element(By.CSS_SELECTOR, "")
- find_elements_by_css_selector or 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 [23]:



```
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 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('Continue') or find_element(By.LINK_TEXT, "")
find_element_by_partial_link_text('Conti') or find_element(By.PARTIAL_LINK_TEXT, "")
```

In [24]:



```
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

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

In [41]:



```
from selenium import webdriver

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

driver.get(url)
```

In [42]:



```
sel_search_A = driver.find_element(By.XPATH, '/html/body/section/header/div[1]/div/div/div[2]/div[3]')
print(sel_search_A.tag_name)
print(sel_search_A.text)
sel_search_A.click()

# //*[@id="Inb.searchForm"]/fieldset/input[1]
#sel_search = driver.find_element_by_xpath('//*[@id="Inb.searchForm"]/fieldset/input[1]')
sel_search_B = driver.find_element(By.XPATH, '//*[@id="u_hs"]/div/div/input')

print(sel_search_B.tag_name)
print(sel_search_B.text)

# //*[@id="Inb.searchForm"]/fieldset/button
#sel_btn = driver.find_element_by_xpath('//*[@id="Inb.searchForm"]/fieldset/button')
sel_btn = driver.find_element(By.XPATH, '//*[@id="u_hs"]/div/div/button[2]')

print(sel_btn.tag_name)
print(sel_btn.text)

sel_search_B.send_keys("패션")
sel_btn.click()
```

span

검색

input

button

뉴스검색

(10) 여러개의 정보를 xpath를 이용해서 가져와 보기

In [43]:



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

In [45]:



```
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,li)
06. id를 활용한 정보 획득
07. class를 활용한 정보 획득
08. 하나의 이미지 다운로드
09. 여러개의 이미지 다운로드

history

- 2020/09 v14 : 정보 업데이트, 일부
- 2021/09 v15 : 정보 업데이트
- 2022/04 v16 : 정보를 가져오는 함수의 변경 적용. (예) driver.find_element(By.XPATH, one_xpath)

REF

- warning message : find_element_by_* commands are deprecated in selenium
 - <https://stackoverflow.com/questions/69875125/find-element-by-commands-are-deprecated-in-seleniumm> (<https://stackoverflow.com/questions/69875125/find-element-by-commands-are-deprecated-in-seleniumm>)

교육용으로 작성된 것으로 배포 및 복제시에 사전 허가가 필요합니다.

Copyright 2021 LIM Co. all rights reserved.