

노트북: 첫 번째 노트북

만든 날짜: 2020-10-26 오전 8:31

URL: <https://continuous-development.tistory.com/112?category=736681>

Python

[Python] python 에서 Seleium을 통한 동적 크롤링 - 1

2020. 10. 8. 15:50 수정 삭제 공개

Seleium을 통한 동적 크롤링

첫 번째로는 기본적인 함수를 넣어준다.

```
In [48]: from selenium import webdriver  
path = './chromedriver.exe'  
driver = webdriver.Chrome(path)  
driver
```

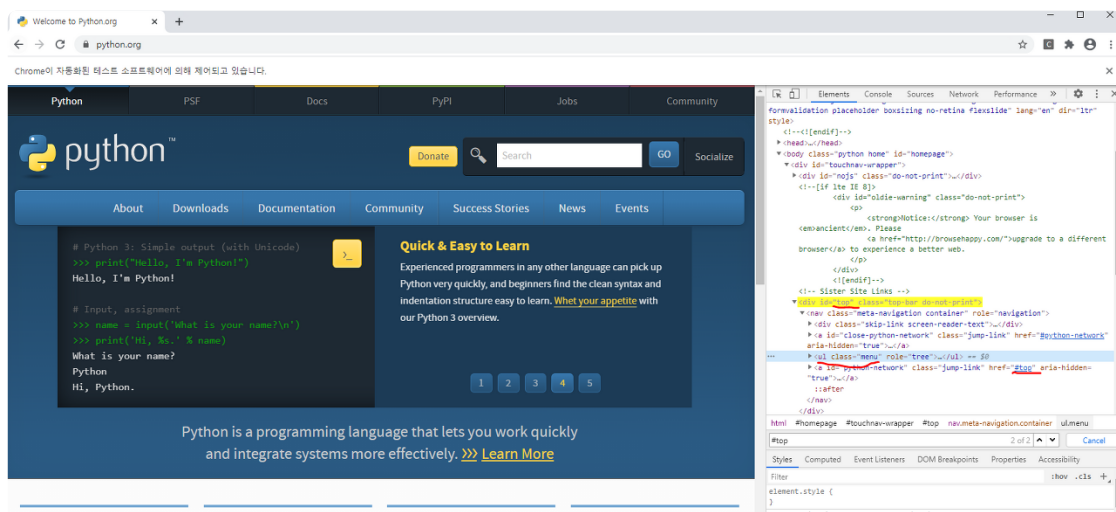
```
Out [48]: <selenium.webdriver.chrome.webdriver.WebDriver (session="94f2f1beb314c7cd475b38ac6b13ee76")>
```

- Selenium 특정요소를 찾는 함수
- find_element_*
- find_elements_*
- find_element_by_id() # id
- find_element_by_name() # name
- find_element_by_css_selector() # class
- 리턴 시켜주는 타입 WebElement 객체를 리턴 text

위에 간단하게 셀레늄을 통해 요소를 찾는 방법들을 정리했다.

```
In [50]: driver.get('https://www.python.org')
```

driver.get('url')이라는 명령어 구문으로 해당 사이트를 접속한다.



해당 부분에서 나는 메뉴에 따라서 크롤링을 하려 한다. 그러기 위해서 해당 부분이 어떤 것으로 이뤄졌는지 확인한다.

```

In [52]: menus = driver.find_elements_by_css_selector('#top ul.menu li')
menus

Out [52]: [<selenium.webdriver.remote.webelement.WebElement (session="94f2f1beb314c7cd475b38ac6b13ee76", element="4afa90fe-c15e-4d5c-b824-492b427ea755")>,
<selenium.webdriver.remote.webelement.WebElement (session="94f2f1beb314c7cd475b38ac6b13ee76", element="b81822cb-8559-4450-ba70-664c77e07373")>,
<selenium.webdriver.remote.webelement.WebElement (session="94f2f1beb314c7cd475b38ac6b13ee76", element="3c793b1b-3eb2-4fd6-a7f0-e0c88895a683")>,
<selenium.webdriver.remote.webelement.WebElement (session="94f2f1beb314c7cd475b38ac6b13ee76", element="e5d7641d-5a2f-4ce1-82b8-e47444120a9e")>,
<selenium.webdriver.remote.webelement.WebElement (session="94f2f1beb314c7cd475b38ac6b13ee76", element="4fca0200-1286-4224-9a81-a4b45d2dc1e2")>,
<selenium.webdriver.remote.webelement.WebElement (session="94f2f1beb314c7cd475b38ac6b13ee76", element="c1c4bdcf9-8b5e-4eed-9643-f73f6993c98f")>]

```

해당되는 모든 요소를 가져오기 위해 find_elements를 사용하였고 나는 class를 가져오기 때문에 뒤에 추가적으로 _by_css_selector을 붙여줬다. 그리고 가져오는 부분의 이름을 적어주고 경로까지 적어주었다.

이런 방식으로 가져온다. menus 를 가져온다.

그다음 가져온 menus를 하나씩 꺼내어 menu를 클릭하는 로직을 구현한다.

```

In [62]: click_menu = None
for menu in menus:
    # print(menu.text)

    if menu.text == 'Python':
        click_menu = menu
click_menu.click()
time.sleep(5)
driver.close()

```

여기서는 menu 중에 Python 이 있으면 그 메뉴를 click_menu로 저장하고. click()를 통해 클릭 이벤트를 발생시킨다. 그 후 5 초뒤에 close 하게 한다.

이렇게 클릭 이벤트를 할 수 있다.

'Python' 카테고리의 다른 글

[Python] python 에서 Seleium을 통한 동적 크롤링 - 3

[Python] python 에서 Seleium을 통한 동적 크롤링 - 2

[Python] python 에서 Seleium을 통한 동적 크롤링 - 1

[Python] python 에서 Seleium을 설치 방법

[Python] BeautifulSoup을 통한 이미지 블로그 스크래핑하기

[Python] BeautifulSoup을 통한 이미지 스크래핑 하기

python click 이벤트

python Seleium

python 동적 크롤링



나무늘보스

혼자 끄적끄적하는 블로그 입니다.