

selenium : 게시판 사이트 들어가서 글들 자동으로 가져오는

```
In [ ]: !pip install webdriver-manager
```

```
In [ ]: pip install selenium
```

```
In [10]: # selenium 할때마다 처음에 입력 할 것/환경조성
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup

# 크롬 브라우저를 selenium 사용하기 좋게 만들기
service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service)
```

```
In [11]: driver.get('https://www.clien.net/service/board/jirum')
```

```
In [ ]: for i in range(1,11):
    driver.find_element(By.CSS_SELECTOR, f'div.list_item:nth-child({i})').find_eleme

    # 페이지 안에 들어갔다 나올때 실행하는 코드
    soup = BeautifulSoup(driver.page_source, 'html.parser')
    try:
        print(soup.select_one('div.post_content a').get('href'))
    except:
        pass

    driver.back()
```

기상청 날씨 정보

```
In [ ]: from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By

options = webdriver.ChromeOptions()
options.add_argument('--start-maximized')

#options.add_argument('') 따옴표 안에 속성을 입력, 구글크롬 사용시 옵션
```

```
In [ ]: service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service, options=options)

driver.get('https://www.weather.go.kr/w/forecast/short-term.do')

driver.find_element(By.CSS_SELECTOR, '.sym-btn.txt-only.pop-open').click()
time.sleep(2)
driver.find_element(By.CSS_SELECTOR, 'a[data-name = "세종특별자치시"]').click()
time.sleep(2)
driver.find_element(By.CSS_SELECTOR, 'a[data-name = "세종특별자치시"]').click()
time.sleep(2)
driver.find_element(By.CSS_SELECTOR, 'a[data-name = "반곡동"]').click()
time.sleep(2)
driver.find_element(By.CSS_SELECTOR, 'div.btn-wrap > button').click()
time.sleep(2)
driver.find_element(By.CSS_SELECTOR, 'div.btn-wrap > a.view-table').click()
```

```
time.sleep(2)

html = driver.page_source # 가져온 웹데이터 html에 담기
driver.close()
```

```
In [ ]: list_ = []
soup = BeautifulSoup(html, 'html.parser')
weather = soup.select('div.item-wrap > ul.s-item, ul.vs-item')
for ul in weather:
    date = ul.get('data-date')
    time = ul.get('data-time')
    cloud = ul.select_one('.wic').text.strip()
    temp = ul.select_one('.feel').text.strip().split('°C')[0]
    rain = ul.select_one('.pcp > span:last-child').text.strip()
    if (rain == '-') or (rain == ''):
        rain = '0'
    elif rain.find('~') != -1:
        rain = rain[1:-2]
    else:
        rain = rain[:-2]
    wind = ul.select_one('.wspd').text.strip().replace('m/s', '')
    humid = ul.select_one('li:nth-child(8) > span:nth-child(2)').text.strip()[:-1]
    list_.append([date, time, cloud, temp, rain, wind, humid])
```

```
In [ ]: import time
time.sleep(1)
```

```
In [ ]: driver.find_element(By.CSS_SELECTOR, '.sym-btn.txt-only.pop-open').click()
driver.implicitly_wait(3) # 3초 기다리기
driver.find_element(By.CSS_SELECTOR, 'a[data-name = "서울특별시"]').click()
driver.implicitly_wait(3)
driver.find_element(By.CSS_SELECTOR, 'a[data-name = "서초구"]').click()
driver.implicitly_wait(3)
driver.find_element(By.CSS_SELECTOR, 'a[data-name = "서초1동"]').click()
driver.implicitly_wait(3)
driver.find_element(By.CSS_SELECTOR, 'div.btn-wrap > button').click()
```

```
In [ ]: rain = '10mm' # '-' or '~1mm' or '1mm'

if rain == '-':
    rain = 0
elif rain.find('~') != -1:
    rain = int(rain[1:-2])
else:
    rain = int(rain[:-2])
print(rain)
```