Selenium 기초

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

크롬브라우저 특화// selenium 사용시 가장 처음에 넣기

In [ ]:
```
# 처음에 입력! 브라우저 간의 차이 극복? 크롬 잘 동작하도록
from selenium import webdriver
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
from selenium.webdriver.common.by import By

service = Service(ChromeDriverManager().install())
driver = webdriver.Chrome(service=service)
```

In [ ]:
```
pip install selenium    # 브라우저 자동화 역할이 주
```

In [ ]:
```
from selenium import webdriver
from bs4 import BeautifulSoup
```

In [ ]:
```
# 1.크롬 드라이버 실행
driver = webdriver.Chrome()
```

In [ ]:
```
# 2.웹사이트 이동(브라우저에 사이트 접속)
driver.get('https://www.daum.net')
```

In [ ]:
```
# 3.추출한 웹데이터 html에 넣기
html=driver.page_source
```

In [ ]:
```
# 3.html 파싱
soup=BeautifulSoup(html,'html.parser')
```

In [ ]:
```
# 4.선택
soup.select('tr')
```

In [ ]:
```
# 페이지 사이즈 조절
driver.set_window_size(200,400)
```

In [ ]:
```
# 브라우저 스크롤 조절
driver.execute_script('window.scrollTo(300,300);')
```

In [ ]:
```
# alert 다루기 -간단한 알림창, 자바스크립트
driver.execute_script('alert("hello selenium!!!");')    #창 열기
alert = driver.switch_to.alert
alert.accept()
```

In [ ]:
```
# selenium 태그 찾기
from selenium.webdriver.common.by import By

driver.get('https://www.daum.net')
driver.find_element(By.CSS_SELECTOR, '#q').send_keys('셀레니움')
driver.find_element(By.CSS_SELECTOR, '.inner_search > .ico_pctop.btn_search').click(
```

In [ ]:
```
# 브라우저 종료
driver.close()
or
driver.quit
```

In [ ]:
```
# txt 저장/가능은 하나 너무 늦어서 잘 사용하지 않음
driver.find_element(By.CSS_SELECTOR, 'a.keyword').txt
```

In [ ]:
```
words=driver.find_element(By.CSS_SELECTOR, 'a.keword')
```

In [ ]:
```

```

In [ ]:
```
# xpath를 사용하여 찾기
driver.find_element(By.XPATH,'//*[@id="q"]').send.keys('빅데이터')  #키를 입력
driver.find_element(By.XPATH,'//*[@id="daumSearch"]/fieldset/div/div/button[3]').cli
```

In [ ]:
```
from selenium.webdriver.common.keys import Keys
Keys.ENTER
```

TED 사이트 크롤링

In [ ]:
```
driver = webdriver.Chrome()
driver.get('https://www.ted.com/talks')
```

In [ ]:
```
# 텍스트 문구 가져오기
words = driver.find_elements(By.CSS_SELECTOR, '#TopicsLaunchPad-Top-Default span.rel
for word in words:
    print(word.text)
```

In [ ]:
```
# 제목 제이터 가져오기
contents = driver.find_elements(By.CSS_SELECTOR, ".mb-1 > span")
len(contents)

# 제목 텍스트 가져오기
for content in contents:
print(content.text)

# 셀렉트 박스 선택후 데이터 가져오기 -한국어 선택
driver.find_element(By.CSS_SELECTOR,'#filterBarScrollArea div > button > span > p').
driver.find_element(By.CSS_SELECTOR,'#dropdown-menu > li:nth-child(6) > button > spa
driver.find_element(By.CSS_SELECTOR, '#k ul > li:nth-child(5) > button').click()

# 링크 데이터 가져오기
driver = webdriver.Chrome()
driver.get('https://naver.com')
login = driver.find_element(By.CSS_SELECTOR, '#account a').get_attribute('href')
driver.get(login)
```

headless : 브라우저 띄우지않고 크롤링하는 방법

In [ ]:
```
options = webdriver.ChromeOptions()
options.add_argument('headless')
driver = webdriver.Chrome(options=options)
driver.get("https://www.ted.com/talks")
words = driver.find_elements(By.CSS_SELECTOR,'#TopicsLaunchPad-Top-Default span.rela
for word in words:
print(word.text)
driver.quit()
```

크롬 드라이버 옵션들

In [ ]:
```
options.add_argument('--window-size= 200, 300')  #실행되는 브라우저 크기 지정
options.add_argument('--start-maximized')        #브라우저가 최대화된 상태로 실행
options.add_argument('--start-fullscreen')       #브라우저가 풀스크린 모드(F11)로 실
options.add_argument('--blink-settings=imagesEnabled=false') #이미지 로딩 안함
options.add_argument('--mute-audio')             #브라우저에 음소거 옵션 적용
options.add_argument('incognito')                #시크릿 모드의 브라우저 실행
```

스타벅스 사이트크롤링 /beautifulsoup 사용

In [ ]:
```
from selenium import webdriver
from selenium.webdriver.common.by import By
from bs4 import BeautifulSoup
```

In [ ]:
```
driver = webdriver.Chrome()
driver.get('https://www.istarbucks.co.kr/store/store_map.do')
```

In [ ]:
```
# 클라스명으로 버튼찾기(지역검색)
loca = driver.find_element(By.CLASS_NAME, 'loca_search')
loca.click()
# 전체 li 상위 태그를 찾아서 다중 검색으로 li태그 선택
sido = driver.find_element(By.CLASS_NAME, 'sido_arae_box')
li = sido.find_elements(By.TAG_NAME, 'li')
li[16].click()
# id 명으로 버튼 찾기(시군구 버튼)
gugun = driver.find_element(By.ID, 'mCSB_2_container')
li = gugun.find_elements(By.TAG_NAME, 'li')
li[1].click()
```

In [ ]:
```
# .page_source를 이용하여 변경된 html 소스 가져오기
source = driver.page_source
print(source[:200])

html = BeautifulSoup(source,'html.parser')
```

In [ ]:
```
# 검색 결과 찾기
search_list = html.select('ul.quickSearchResultBoxSidoGugun > li')
items = []
for item in search_list:
    items.append(item.find('p').text)
    print(items[0])
driver.quit()
```

In [ ]:
```

```