3rdWeek Practice

2018102168 김보의 10

■ Practice1

url 변수와 함께 analyze_html함수가 실행

```
savepath = download_file(url)
if savepath is None: return
if savepath in proc_files: return
proc_files[savepath] = True
print("analyze_html=",url)
```

해당 url의 웹페이지 다운 후 경로 리턴

해당 파일 이미 처리한 것 인지 확인

처리한 것 아닐 시 표시 남기고 처리

```
links = enum_links(html, url)
```

해당 웹페이지의 href를 모두 가져옴

```
if link_url.find(root_url) != 0:
    if not re.search(r".css$", link_url): continue

if re.search(r".(html|html)$", link_url):
    analyze_html(link_url, root_url)
    continue

download_file(link_url)
```

Root_url에서 파생된 게 아니면서 css가 아니면 패스, css면 웹페이지 다운

Root_url에서 파생된거면 재귀적으로 analyze_html실행

def download_file은 해당 url의 경로와 같은 디렉토리가 만들어졌는지 확인후 없으면 만들고 다운

있으면 다운로드 실행후 성공시 다운로드한 파일 경로 반환

def enum_links는 다운로드한 html파일의 href속성을 찾아서 값을 전부 절대경로로 변경하여 분석에 이용

```
(venv) boui@boui-virtual-machine:~/BigDataProject/3RDWeek/venv/code$ python3 P1.py
analyze_html= https://docs.python.org/3.5/library/
mkdir= ./docs.python.org/3.5/_static
download= https://docs.python.org/3.5/_static/pydoctheme.css
download= https://docs.python.org/3.5/_static/pygments.css
download= https://docs.python.org/3.5/library/intro.html
analyze_html= https://docs.python.org/3.5/library/intro.html
download= https://docs.python.org/3.5/library/functions.html
analyze_html= https://docs.python.org/3.5/library/functions.html
download= https://docs.python.org/3.5/library/constants.html
analyze_html= https://docs.python.org/3.5/library/constants.html
download= https://docs.python.org/3.5/library/stdtypes.html
analyze_html= https://docs.python.org/3.5/library/stdtypes.html
download= https://docs.python.org/3.5/library/exceptions.html
analyze_html= https://docs.python.org/3.5/library/exceptions.html
download= https://docs.python.org/3.5/library/text.html
analyze_html= https://docs.python.org/3.5/library/text.html
download= https://docs.python.org/3.5/library/string.html
analyze_html= https://docs.python.org/3.5/library/string.html
download= https://docs.python.org/3.5/library/re.html
analyze_html= https://docs.python.org/3.5/library/re.html
download= https://docs.python.org/3.5/library/difflib.html
analyze_html= https://docs.python.org/3.5/library/difflib.html
download= https://docs.python.org/3.5/library/textwrap.html
analyze_html= https://docs.python.org/3.5/library/textwrap.html
download= https://docs.python.org/3.5/library/unicodedata.html
```

■ Practice2

```
login = driver.find_element(By.CSS_SELECTOR, ".btn-seconda
login.click()

idbox = driver.find_element(By.CSS_SELECTOR, "#username")
idbox.send_keys("boui2000@naver.com")

pwbox = driver.find_element(By.CSS_SELECTOR, "#password")
pwbox.send_keys("12311231")

login_button = driver.find_element(By.CSS_SELECTOR, "#orga
login_button.click()

apply_button = driver.find_element(By.CSS_SELECTOR, ".jobs
apply_button.click()
```

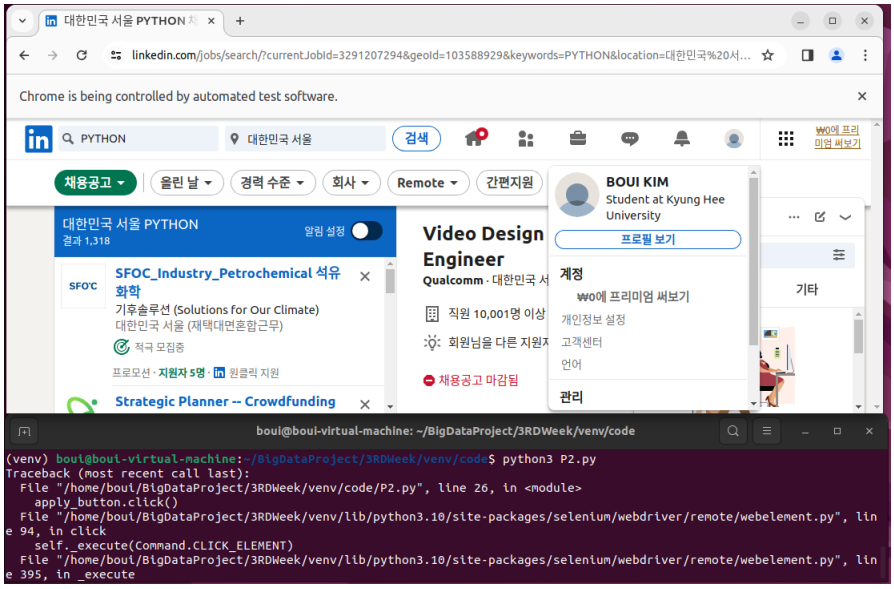
selenium을 통해 chrome을 제어하기 위해 chromedriver 설치.

selenium 라이브러리를 통해 페이지를 조작할 객체 생성 후 크롬 웹 드라이버 세팅.

Selenium을 통해 조작할 웹페이지 url 입력.

해당 페이지내에서 조작될 요소를 driver가 찾을수 있게 CSS_SELECTOR 값을 입력.

객체에 각종 메소드를 통해 해당 요소를 이용



마지막 'apply_butten'객체가 이용할 요소에 CSS_SELECTOR값은 ".jobs-apply-button"인데 찾지 못함. 해당 CSS_SELECTOR값을 가진 요소가 웹페이지 어디에 또는 html안에 어느 태그에 위치해 있는지 확인하려 했는데 찾지 못함. 또한 다른 기업에 대한 '지원' 버튼은 CSS_SELECTOR값이 페이지에 들어갈 때마다 유동적으로 변하여 이용이 어려움

■ Practice3

```python
import requests
import json

apikey = "daf764df314a58e005edce3b73c6af22"


cities = ["Seoul", "Tokyo", "New York"]

api = "https://api.openweathermap.org/data/2.5/weather?q={city name}&appid={API key}"


k2c = lambda k: k - 273.15

for city in cities:
    response = requests.get(f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid={apikey}")
    data = json.loads(response.text)

    print("+ City = " + str(data["name"]))
    print("| WEATHER = " + str(data["weather"][0]["description"]))
    print("| MAX_TEMP = " + str(k2c(data["main"]["temp_max"])))
    print("| MIN_TEMP = " + str(k2c(data["main"]["temp_min"])))
    print("| HUMIDITY = " + str(data["main"]["humidity"]))
    print("| PRESSURE = " + str(data["main"]["pressure"]))
    print("| DEG = " + str(data["wind"]["deg"]))
    print("| SPEED = " + str(data["wind"]["speed"]))
```

openweathermap에서 data를 받아오기 위해 여러 API중 '도시이름'과 'API'키를 입력하여 데이터를 받아오는 포맷을 이용.

data를 json형식으로 읽어서 data를 이용.

```
(venv) boui@boui-virtual-machine:~/BigDataProject/3RDWeek/venv/code$ python3 P3.py
+ City = Seoul
| WEATHER = moderate rain
| MAX_TEMP = 8.760000000000048
| MIN_TEMP = 8.660000000000025
| HUMIDITY = 76
| PRESSURE = 1016
| DEG = 60
| SPEED = 5.14
+ City = Tokyo
| WEATHER = moderate rain
| MAX_TEMP = 12.79000000000002
| MIN_TEMP = 10.920000000000016
| HUMIDITY = 87
| PRESSURE = 1013
| DEG = 30
| SPEED = 1.54
+ City = New York
| WEATHER = clear sky
| MAX_TEMP = 3.32000000000005
| MIN_TEMP = -0.6899999999999977
| HUMIDITY = 61
| PRESSURE = 1032
| DEG = 22
| SPEED = 7.15
```

■ Practice4

```python
#!usr/bin/env python3
from bs4 import BeautifulSoup
import urllib.request
import ssl
import datetime


now = datetime.datetime.now().replace(microsecond=0)
now = now.strftime("%Y-%m-%d %H:%M:%S")

url = "https://finance.naver.com/marketindex/"
res = urllib.request.urlopen(url)
data = res.read()

soup = BeautifulSoup(data, 'html.parser')
price = (soup.select_one('#exchangeList > li.on > a.head.usd > div > span.value')).string


print("DATE: " + now + " 미국 USD: " + price)
```

해당 url의 html파일을 읽어와 CSS_SELECTOR값으로 원하는 요소를 제어

```
Ln 14, Col 1   Spaces: 4   UTF-8   LF   Python
* * * * * /home/boui/BigDataProject/3RDWeek/venv/bin/python3 /home/boui/BigDataProject
/3RDWeek/venv/code/P4.py >> /home/boui/BigDataProject/3RDWeek/venv/code/P4.log 2>&1
(venv) boui@boui-virtual-machine:~/BigDataProject/3RDWeek/venv/code$
```

Crontab이라는 Linux Scheduler를 이용해 주기적이고, 자동적으로 작업을 수행

```
(venv) boui@boui-virtual-machine:~/BigDataProject/3RDWeek/venv/code$ cat P4.log
DATE: 2024-03-24 20:17:01 미국 USD: 1,346.00
DATE: 2024-03-24 20:18:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:39:02 미국 USD: 1,346.00
DATE: 2024-03-24 21:40:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:41:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:42:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:43:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:44:02 미국 USD: 1,346.00
DATE: 2024-03-24 21:45:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:46:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:47:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:48:01 미국 USD: 1,346.00
DATE: 2024-03-24 21:49:02 미국 USD: 1,346.00
DATE: 2024-03-24 21:50:02 미국 USD: 1,346.00
```