[AI] 8/1 실습

SeSAC - 파이썬 데이터 처리 프로그래밍 2일차

2023.08.01

실습 - 지식인에서 파이썬 검색해서 제목과 링크 가져오기

```
import requests
from fake_useragent import UserAgent
from bs4 import BeautifulSoup

ua = UserAgent()
headers = {
    "User-Agent" : ua.random
}

res = requests.get("<https://search.naver.com/search.naver?where=kin&sm=tab_jum&query
=%ED%8C%8C%EC%9D%B4%EC%8D%AC>", headers=headers)

bs = BeautifulSoup(res.text, 'html.parser')
area = bs.select_one(".lst_total")
# area = bs.select_one(".lst_total._list") # 두가지 클래스명을 다 가진것을 가져올때
elem = area.select(".question_text")

for e in elem:
    print(e.text)
    print(e.attrs["href"])
```


pagenation을 이용해서 여러 페이지 크롤링하기

네이버에서 파이썬을 검색한 첫번째 페이지의 주소를 확인해보자.

https://github.com/kimbap918/TIL/assets/75712723/4af4a623-7652-437a-8863-8b99283a8e57

페이지의 주소는 아래와 같다.

```
# kin_start=1
<https://search.naver.com/search.naver?where=kin&kin_display=10&qt=&title=0&&answer=0&
grade=0&choice=0&sec=0&nso=so%3A-1%2Ca%3A%2Cp%3Aall&query=%ED%8C%8C%EC%9D%B4%EC%8D%AC&
c_id=&c_name=&sm=tab_pge&kin_start=1&kin_age=0>
```

그렇다면 두번째 페이지도 확인해보자.

https://github.com/kimbap918/TIL/assets/75712723/b1dcb530-e10a-497b-8659-0e80430296d1

```
# kin_start=11
<https://search.naver.com/search.naver?where=kin&kin_display=10&qt=&title=0&&answer=0&grade=0&choice=0&sec=0&nso=so%3A-1%2Ca%3A%2Cp%3Aall&query=%ED%8C%8C%EC%9D%B4%EC%8D%AC&c_id=&c_name=&sm=tab_pge&kin_start=11&kin_age=0>
```

페이지가 증가함에 따라 kin_start가 10씩 증가한다.

이것을 이용해서 여러 페이지를 한번에 크롤링 할 수 있다.

여러 페이지 크롤링

```
import requests
from fake_useragent import UserAgent
from bs4 import BeautifulSoup

ua = UserAgent()
headers = {
    "User-Agent" : ua.random
}

# 크롤링 하려는 페이지의 url
url = "<https://search.naver.com/search.naver?where=kin&kin_display=10&qt=&title=0&&answer=0&grade=0&choice=0&sec=0&nso=so%3A-1%2Ca%3A%2Cp%3Aall&query=%ED%8C%8C%EC%9D%B4%EC%8D%AC&c_id=&c_name=&sm=tab_pge&kin_start=1&kin_age=0>"

# 크롤링 페이지의 수
for page in range(3):
    # 페이지를 계산
    request_url = f"{url}&kin_display=10&kin_start={(page*10)+1}"
    res = requests.get(url, headers=headers)
    bs = BeautifulSoup(res.text, 'html.parser')

    area = bs.select_one(".lst_total")
    elem = area.select(".question_text")

    for e in elem:
        print(e.text)
        print(e.attrs["href"])
```


pymysql

jupyter에서 mysql 실행하기

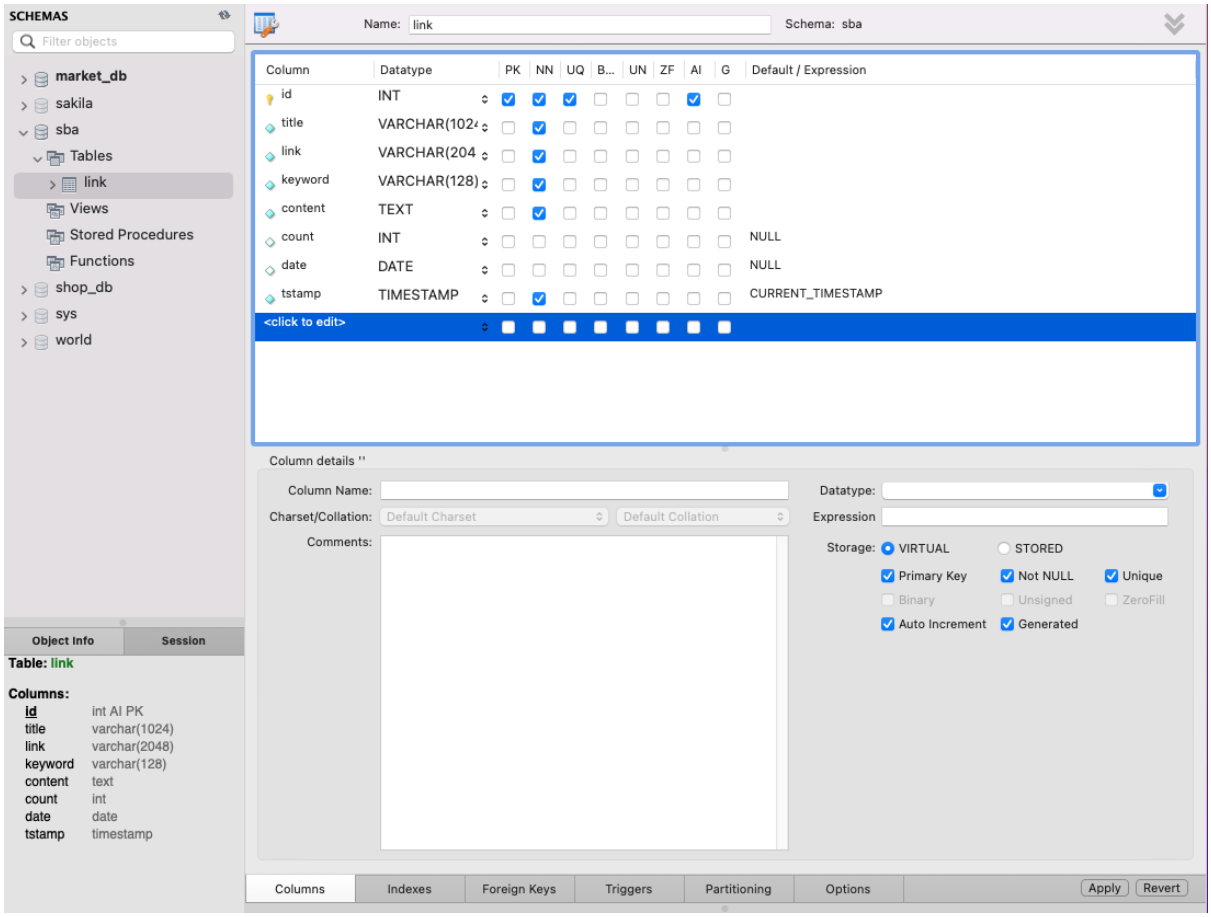
```
import pymysql

db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="market_db")
cursor = db.cursor() # cursor로 sql 쿼리를 가져온다.

sql = """
SELECT * FROM member;
"""

cursor.execute(sql) # 커서 실행
result = cursor.fetchmany(size=100) # fetchone = 하나만, fetchmany = 여러개
for data in result:
    print(data)

db.close()
```


값 삽입하기

```
import pymysql

db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="market_db")
cursor = db.cursor()

sql = """
insert into member values("ABC", "에이비씨", "10", '경기', '031', '11122233', '170', '202
3-08-01');
"""

cursor.execute(sql)

db.commit()
db.close()
```


테이블 생성해서 값 삽입해보기

1. sba 스키마 생성 후 아래와 같이 테이블 생성



1. jupyter에서 값 삽입

```
import pymysql

db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="sba")
cursor = db.cursor()

sql = """
insert into link
values(NULL, 'title-test', 'link-test', 'keyword-test', 'content-test', 0, '2022-12-1
2', '2023-08-01 00:00:00');
"""

cursor.execute(sql)

db.commit()
db.close()
```


실습 - 파이썬 뉴스 크롤링해서 DB에 넣기

```
# 파이썬 뉴스 5페이지까지 수집
# title = 뉴스 제목, link = 뉴스의 링크, keyword = 검색어, 파이썬, content = 요약내용, count =
 요약 내에서 키워드가 들어간 횟수
# date = 안넣어도됨, timestamp = now()

import requests
from fake_useragent import UserAgent
from bs4 import BeautifulSoup
import pymysql

db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="sba")
cursor = db.cursor()

ua = UserAgent()
headers = {
    "User-Agent" : ua.random
}

# 파이썬 뉴스 페이지
url = "<https://search.naver.com/search.naver?where=news&sm=tab_pge&query=%ED%8C%8C%E
C%9D%B4%EC%8D%AC&sort=0&photo=0&field=0&pd=0&ds=&de=&cluster_rank=62&mynews=0&office_t
ype=0&office_section_code=0&news_office_checked=&nso=so:r,p:all,a:all>"

# 5페이지를 가져오기 위해 range(5)
for page in range(5):
    # url 뒤에 페이지 계산
    request_url = f"{url}&start={(page*10)+1}"
    res = requests.get(request_url, headers=headers)
    bs = BeautifulSoup(res.text, 'html.parser')

    # 필요한 정보가 있는 태그 클래스
    area = bs.select_one(".list_news")
    elem = area.select(".news_area")


    for e in elem:
        # 제목의 정보를 텍스트로, DB에 삽입 시 작은따옴표가 오류가 생겨서 큰 따옴표로 치환
        title = e.select_one(".news_tit").text.replace("'", '"')
        # 제목의 정보에서 href 링크를 저장
        link = e.select_one(".news_tit").attrs["href"]
        # 키워드는 파이썬
        keyword = "파이썬"
        # elem 내의 .dsc_txt_wrap 클래스 정보를 텍스트로, 작은따옴표 오류가 생겨서 큰 따옴표로 치환
        content = e.select_one(".dsc_txt_wrap").text.replace("'", '"')
        # 컨텐츠 내의 파이썬 단어 개수를 저장
        cnt = content.count(keyword)

        sql = f"""
        insert into link
        values(NULL, '{title}', '{link}', '{keyword}', '{content}', {cnt}, NULL, now
());
        """
        cursor.execute(sql)
```

```
db.commit()
db.close()
```


실습 - 뉴스 기사를 모두 방문해서 "파이썬"이 들어간 개수 확인하기

- 작성한 답안

```
import requests
from fake_useragent import UserAgent
from bs4 import BeautifulSoup
import pymysql

db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="sba")

ua = UserAgent()
headers = {
    "User-Agent" : ua.random
}

url = "<https://search.naver.com/search.naver?where=news&sm=tab_pge&query=%ED%8C%8C%E
C%9D%B4%EC%8D%AC&sort=0&photo=0&field=0&pd=0&ds=&de=&cluster_rank=62&mynews=0&office_t
ype=0&office_section_code=0&news_office_checked=&nso=so:r,p:all,a:all>"
cursor = db.cursor()

for page in range(2):
    request_url = f"{url}&start={(page*10)+1}"
    res = requests.get(request_url, headers=headers)
    bs = BeautifulSoup(res.text, 'html.parser')

    area = bs.select_one(".list_news")
    elem = area.select(".news_area")

    for e in elem:

        title = e.select_one(".news_tit").text.replace("'", '"')
        link = e.select_one(".news_tit").attrs["href"]
        # 링크 안에 있는 내용을 읽어오기
        res2 = requests.get(link, headers=headers)
        bs2 = BeautifulSoup(res2.text, 'html.parser')
        content = bs2.select_one("html").text

        keyword = "파이썬"
        cnt = content.count(keyword)

        sql = f"""
        insert into link
        values(NULL, '{title}', '{link}', '{keyword}', '{content}', {cnt}, NULL, now
());
        """
        cursor.execute(sql)
```

```
db.commit()
db.close()
```

- 풀이

```
# 파이썬 뉴스 5페이지까지 수집
# title = 뉴스 제목, link = 뉴스의 링크, keyword = 검색어, 파이썬, content = 요약내용, count =
 요약 내에서 키워드가 들어간 횟수
# date = 안넣어도됨, timestamp =

import requests
import time
from fake_useragent import UserAgent
from bs4 import BeautifulSoup
import pymysql

db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="sba")

ua = UserAgent()
headers = {
    "User-Agent" : ua.random
}

url = "<https://search.naver.com/search.naver?where=news&sm=tab_pge&query=%ED%8C%8C%E
C%9D%B4%EC%8D%AC&sort=0&photo=0&field=0&pd=0&ds=&de=&cluster_rank=62&mynews=0&office_t
ype=0&office_section_code=0&news_office_checked=&nso=so:r,p:all,a:all>"
cursor = db.cursor()

for page in range(2):
    request_url = f"{url}&start={(page*10)+1}"
    res = requests.get(request_url, headers=headers)
    bs = BeautifulSoup(res.text, 'html.parser')

    area = bs.select_one(".list_news")
    elem = area.select(".news_area")

    for e in elem:

        title = e.select_one(".news_tit").text.replace("'", '"')
        link = e.select_one(".news_tit").attrs["href"]
        keyword = "파이썬"

        # 링크 안에 있는 내용을 읽어오기
        res2 = requests.get(link, headers=headers)
        content = res2.text
        cnt = content.count(keyword)

        sql = f"""
        insert into link
        values(NULL, '{title}', '{link}', '{keyword}', '{content}', {cnt}, NULL, now
());
        """
        cursor.execute(sql)
```

```
        time.sleep(0.1)

db.commit()
db.close()
```


openpylx로 엑셀 파일 조작하기

https://openpyxl.readthedocs.io/en/stable/

- 쓰기

```
from openpyxl import Workbook

wb = Workbook()
ws = wb.active

for row in range(10):
    ws.append([row, f"{row}-data"])

# A1 셀에 Test-data 삽입
ws["A1"] = "Test-data"
wb.save("result.xlsx")
```


- 읽기

```
from openpyxl import load_workbook

wb = load_workbook("result.xlsx")
ws = wb.active

for row in ws.iter_rows():
    print(row[0].value, row[1].value)
```


실습 - 뉴스 기사 엑셀 파일로 저장하기

```
# DB관련 내용 삭제후 result.xlsx에 동일한 내용을 저장하도록 변경하기

import requests
from fake_useragent import UserAgent
from bs4 import BeautifulSoup
from openpyxl import Workbook
```

```
wb = Workbook()
ws = wb.active

ua = UserAgent()
headers = {
    "User-Agent" : ua.random
}

url = "<https://search.naver.com/search.naver?where=news&sm=tab_pge&query=%ED%8C%8C%E
C%9D%B4%EC%8D%AC&sort=0&photo=0&field=0&pd=0&ds=&de=&cluster_rank=62&mynews=0&office_t
ype=0&office_section_code=0&news_office_checked=&nso=so:r,p:all,a:all>"

for page in range(2):
    request_url = f"{url}&start={(page*10)+1}"
    res = requests.get(request_url, headers=headers)
    bs = BeautifulSoup(res.text, 'html.parser')

    area = bs.select_one(".list_news")
    elem = area.select(".news_area")

    for e in elem:

        title = e.select_one(".news_tit").text.replace("'", '"')
        link = e.select_one(".news_tit").attrs["href"]
        keyword = "파이썬"

        # 링크 안에 있는 내용을 읽어오기
        res2 = requests.get(link, headers=headers)
        content = res2.text
        cnt = content.count(keyword)

        # 제목, 링크, 키워드, 내용, 카운트
        ws.append([title, link, keyword, content, cnt])

# result.xlsx로 저장
wb.save("result.xlsx")
```


실습 - 멜론 데이터 가져와서 DB에 삽입, Excel에 저장하기

- 순위, 제목, 가수, 앨범, 좋아요, 변동 넣어보기

테스트 데이터 DB에 넣어보기

```
# 순위, 제목, 가수, 앨범, 좋아요

import pymysql

# 테스트 데이터 넣어보기
db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="sba")
```

```
cursor = db.cursor()

sql = """
insert into melon
values(NULL, 101, "바보", "최준혁", "최준혁2집", -1, -100);
"""

cursor.execute(sql)

db.commit()
db.close()
```


- 크롤링 및 DB삽입

```
# 순위, 제목, 가수, 앨범, 좋아요
# rank, title, singer, album, like, diff
import requests
from fake_useragent import UserAgent
from bs4 import BeautifulSoup
from openpyxl import Workbook
import pymysql

# 5. DB 연결 설정
db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="sba")
cursor = db.cursor()

# 8. workbook
wb = Workbook()
ws = wb.active

# 1. userAgent
ua = UserAgent()
headers = {
    "User-Agent" : ua.random
}

# 2. 가져올 url
url = "<https://www.melon.com/chart/index.htm>"
res = requests.get(url, headers=headers)
bs = BeautifulSoup(res.text, 'html.parser')

# 3. 가져올 정보 클래스
area = bs.select_one(".service_list_song")
elem = area.select("div > table > tbody > tr")

# 4. 순위, 제목, 가수, 앨범, 좋아요
for e in elem:
    rank = e.select_one(".rank").text
    title = e.select_one(".ellipsis.rank01 > span > a").text.replace("'", '"') # repla
ce는 DB insert 시 작은따옴표 처리를 위해 사용
    singer = e.select_one(".ellipsis.rank02 > a").text.replace("'", '"')
    album = e.select_one(".ellipsis.rank03 > a").text.replace("'", '"')
```

```
    diff = e.select_one(".rank_wrap").text.replace("'", '"')

    # 6. DB에 저장
    sql = f"""
    insert into melon
    values(NULL, {rank}, '{title}', '{singer}', '{album}', 0, '{diff}');
    """
    cursor.execute(sql)

    # 9. Excel에 저장
    ws.append([rank, title, singer, album, 0, diff])

# 7. DB commit
db.commit()
db.close()

# 10. excel 저장
wb.save("melon.xlsx")

    # print("순위 : ", rank)
    # print("제목 : ", title)
    # print("가수 : ", singer)
    # print("앨범 : ", album)
    # print("변동 : ", diff)
```


실습2 - 순위 변동 표시하기

```
# 순위 변동 추가 변동없음, - +2 -3 형태
# python 안에서 쿼리 짜서 출력
# 각 가수별로 top100에 올라간 곡 수, 가수명을 출력하고 순서대로 정렬해서 출력

import requests
from fake_useragent import UserAgent
from bs4 import BeautifulSoup
from openpyxl import Workbook
import pymysql

# DB
db = pymysql.connect(host="localhost", port=3306, user="root", password="jen401018&",
 db="sba")
cursor = db.cursor()

# workbook
wb = Workbook()
ws = wb.active

# userAgent
ua = UserAgent()
headers = {
    "User-Agent" : ua.random
}
```

```
# 가져올 url
url = "<https://www.melon.com/chart/index.htm>"
res = requests.get(url, headers=headers)
bs = BeautifulSoup(res.text, 'html.parser')

# 가져올 정보 클래스
area = bs.select_one(".service_list_song")
elem = area.select("div > table > tbody > tr")

# 순위, 제목, 가수, 앨범, 좋아요
for e in elem:
    rank = e.select_one(".rank").text
    title = e.select_one(".ellipsis.rank01 > span > a").text.replace("'", '"')
    singer = e.select_one(".ellipsis.rank02 > a").text.replace("'", '"')
    album = e.select_one(".ellipsis.rank03 > a").text.replace("'", '"')
    # 슬라이싱
    diff_icon = e.select_one(".rank_wrap").text[:6]
    diff = e.select_one(".rank_wrap").text[7]

    # 순위가 동일하면 -, 단계 상승이면 +n, 단계 하락이면 -n
    if "순위 동일" in diff_icon:
        diff = "-"
    elif "단계 상승" in diff_icon:
        diff = "+" + diff
    elif "단계 하락" in diff_icon:
        diff = "-" + diff
    else:
        diff = "new"
```


실습3 - python에서 쿼리 실행하기

```
sql = """
SELECT count(singer) 곡수, singer
FROM melon
GROUP BY singer
ORDER BY 곡수 DESC;
"""

cursor.execute(sql)
result = cursor.fetchmany(size=100) # fetchone = 하나만, fetchmany = 여러개
for data in result:
    print(data)
```