

업무자동화

파이썬X웹크롤링 스터디

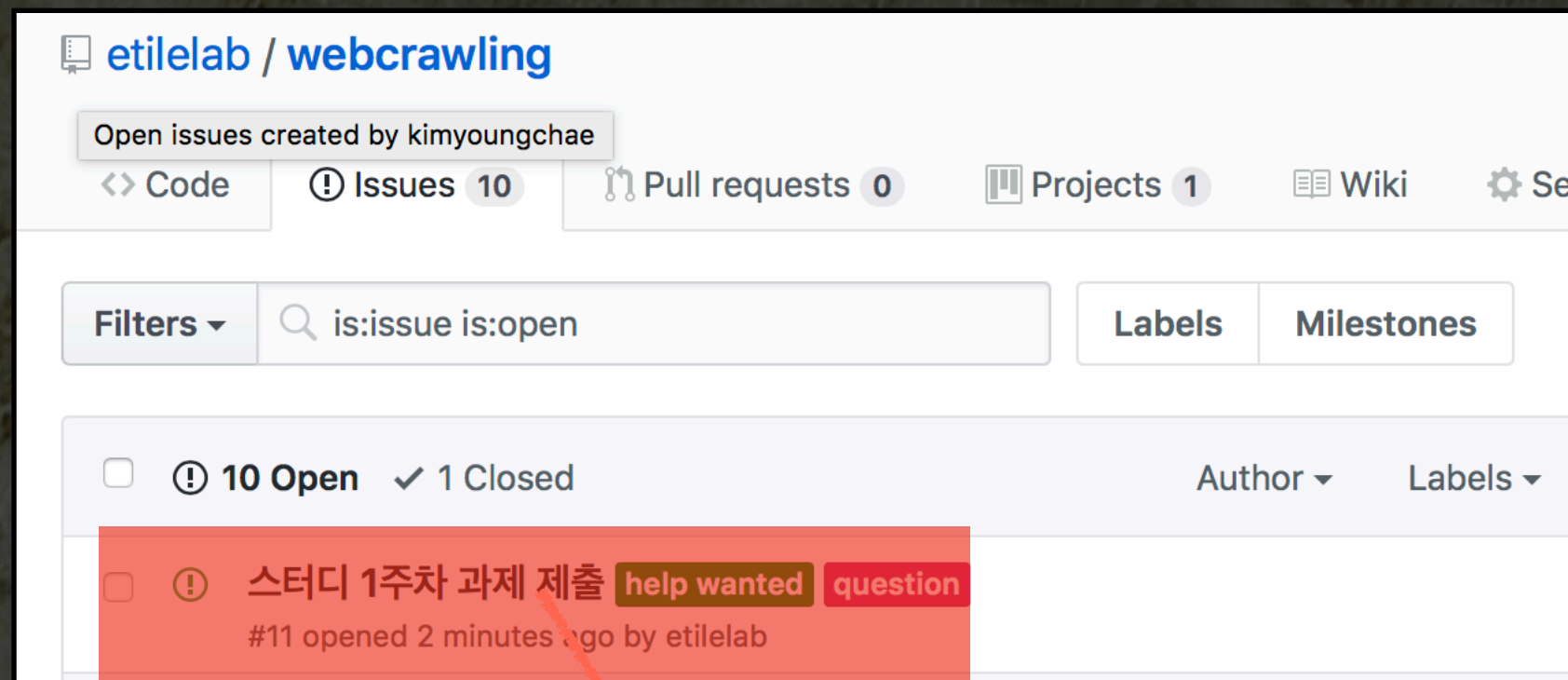
파이썬X웹크롤링 스터디 목표

- 파이썬X웹크롤링 업무자동화 강의의 심화학습
- 강의 수강 후 코드 리뷰를 통한 실력 향상
- 파이썬사용법, 깃허브사용법
- 스터디를 통한 다양한 지식 습득

파이썬X웹크롤링 스터디 참여방법

- 깃허브(<http://github.com>) 가입
- 카카오톡 오픈 채팅방(<https://open.kakao.com/o/g4E8GBA>) 방문 후 문의

파이썬X웹크롤링 스터디 과제제출법




매주 월요일 스터디 O주차 과제제출란 이슈가 생기면 클릭

파이썬X웹크롤링 스터디 과제제출법


스터디 1주차 과제 제출란 (여기에 해주세요) #11


🔔 Open etilelab opened this issue 3 minutes ago · 0 comments


 etilelab commented 3 minutes ago • edited Owner + 🗑️ ✎️


댓글로 소스코드를 zip파일로 압축하여 첨부해주시기 바랍니다.
제출 마감일은 9월 30일 토요일 오후 9시까지 입니다.

이미 다른 이슈를 통해 제출하신분은 재제출하실 필요없습니다.

 etilelab added **help wanted** **question** labels 3 minutes ago


 etilelab changed the title from 스터디 1주차 과제 제출 to 스터디 1주차 과제 제출란 24 seconds ago

 etilelab changed the title from 스터디 1주차 과제 제출란 to 스터디 1주차 과제 제출란 (여기에 해주세요) 10 seconds ago

 Write Preview AA ▾ B i “ <> 🔗 ⋮ ⋮ ⋮ ↩ @ ★

Leave a comment

Attach files by dragging & dropping or [selecting them](#).

 Styling with Markdown is supported Close issue Comment

과제제출기간 및
공지 확인

소스코드들을 압축하여, 드래그앤 드롭
각 문제별로 소스파일은 따로 만들것

업무자동화

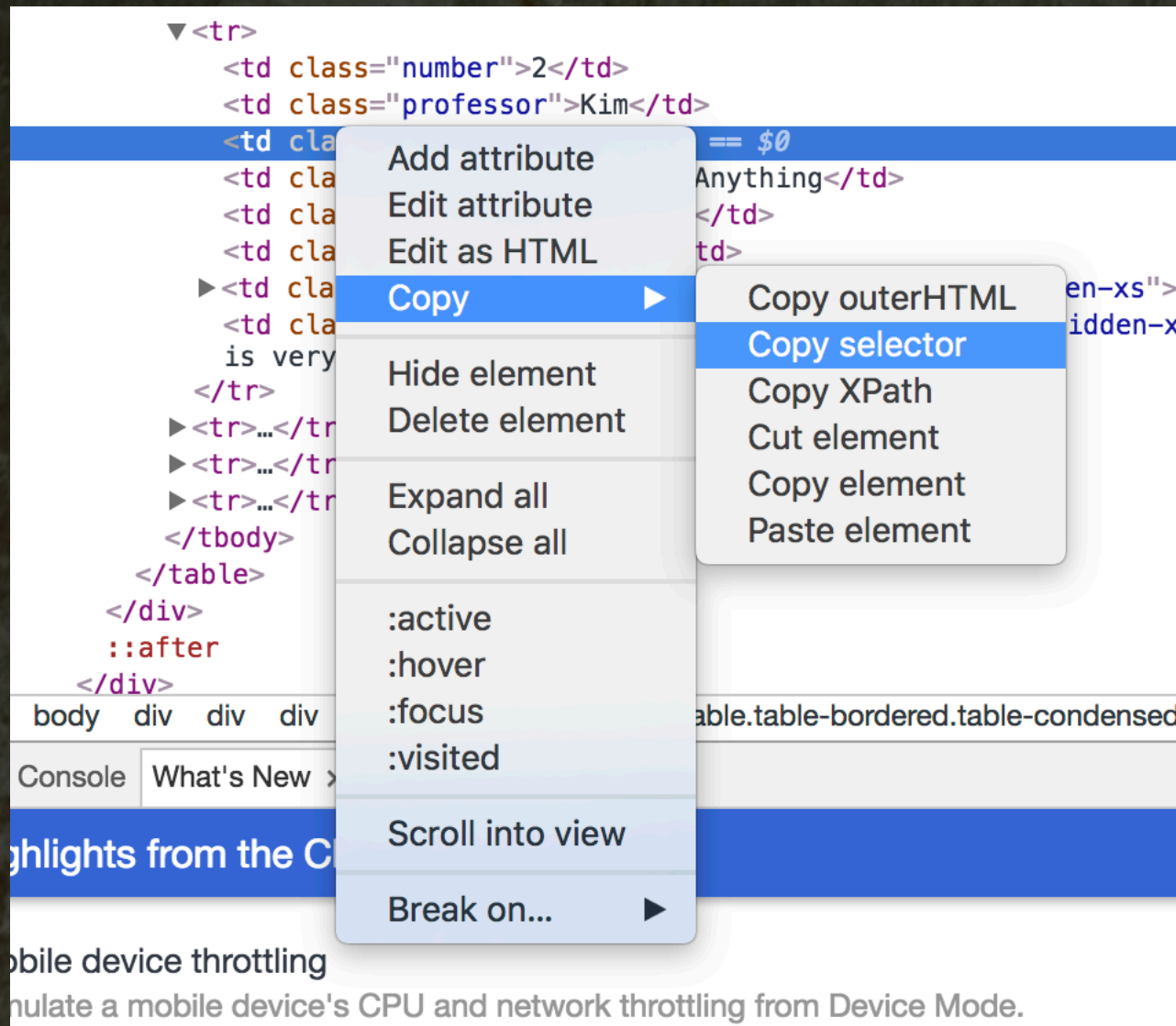
제1주차, 파이썬X웹크롤링 스터디

파이썬X웹크롤링 1주차 복습

- 기본적인 파이썬 문법(조건문, 반복문), 파이썬 설치
- beautifulsoup 라이브러리의 설치와 import
- beautifulsoup를 통한 임의의 정적페이지 크롤링

Css selector 사용하기

구글크롬 -> 홈페이지접속 -> 검사(inspect) -> 태그선택 -> 오른쪽마우스 -> copy -> copy selector



해당 태그의 **Css문법**이 복사됨 !


Soup.select("복사한 css")로 사용 가능

Id -> # / class -> . 으로 구별, 내부태그는 > 으로 해결

header 추가하기

```
headers = {  
    'User-Agent' : 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_4) AppleWebKit/537.36  
(KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36',  
    'referer': 'http://example.com'  
}
```

```
r=requests.get("http://example.com", headers=headers)  
c=r.content
```



원하는 header를 추가하여 통신가능

제1주차 스터디 과제

(<https://github.com/etilelab/webcrawling/blob/master/Week1/Study/practice01.pdf>)

문제1

깃허브에 가입후 Star(즐거찾기) + Watching(구독)을 눌린 후 자신의 깃허브 아이디를
스터디 카카오톡방에 올리시오.

문제2

홈페이지(<http://lambutan.dothome.co.kr>)에서 method, aboutexam 크롤링하시오.
(Beautifulsoup 라이브러리 외 다른 크롤링 라이브러리 사용불가)

문제3

2번의 크롤링 내용 중 구분자(lecturehowNow, test 등)을 제거하여 완전한
컨텐츠 내용으로 나타나게 하시오.

문제 해설

문제2 소스코드

```
# normal answer
```

```
import requests
from bs4 import BeautifulSoup
```

```
r=requests.get("http://lambutan.dothome.co.kr/") # 홈페이지 접속
c=r.content # content(내용) 받아옴
soup=BeautifulSoup(c,"html.parser") # beautifulsoup를 사용할수 있게 만들어 줌
```

```
all=soup.find("tbody") # tbody 라는 태그를 찾아 all이라는 변수에 저장
all2=all.find_all("tr",{"class":""}) # 각 행(tr태그이면서 class는 공백인)을 all2에 저장
```

```
for item in all2: # 각 행을 for 문으로 돌면서
    method=item.find("td",{"class":"method"}).text # td 라는 태그 class 는 method 이며 텍스트만 추출한다
    aboutexam=item.find("td",{"class":"aboutexam"}).text # td 라는 태그 class 는 aboutexam 이며 텍스트만 추출한다
    print("method : " + method + " \naboutexam : " + aboutexam) # 출력
```


문제2 CSS 소스코드

```
import requests
from bs4 import BeautifulSoup

r=requests.get("http://lambutan.dothome.co.kr/") # 홈페이지 접속
c=r.content # content(내용) 받아옴
soup=BeautifulSoup(c,"html.parser") # beautifulsoup를 사용할수 있게 만들어 줌

method=soup.select("#ltable > tbody > tr > td.method.hidden-lg.hidden-md.hidden-sm.hidden-xs")
aboutexam=soup.select("#ltable > tbody > tr > td.aboutexam.hidden-lg.hidden-md.hidden-sm.hidden-xs")

for item,item2 in zip(method,aboutexam): # 각 행을 for 문으로 돌면서
    print(item.text)
    print(item2.text)
```


문제3 소스코드

```
r=requests.get("http://lambutan.dothome.co.kr/") # 홈페이지 접속
c=r.content # content(내용) 받아옴
soup=BeautifulSoup(c,"html.parser") # beautifulsoup를 사용할수 있게 만들어 줌

all=soup.find("tbody") # tbody 라는 태그를 찾아 all이라는 변수에 저장
all2=all.find_all("tr",{"class":""}) # 각 행(tr태그이면서 class는 공백인)을 all2에 저장

for item in all2: # 각 행을 for 문으로 돌면서
    # td 라는 태그 class 는 method 이며 텍스트만 추출한다 / replace 를 통해 원하는 텍스트만 추출
    method=item.find("td",{"class":"method"}).text.replace("lecturehow","")
    aboutexam=item.find("td",{"class":"aboutexam"}).text.replace("test","")
    print("method : " + method + " \naboutexam : " + aboutexam) # 출력
```


업무자동화

제2주차, 파이썬X웹크롤링 스터디

파이썬X웹크롤링 2주차 복습

- beautifulsoup 라이브러리를 통한 정적페이지 크롤링
- beautifulsoup 라이브러리를 통한 동적페이지 크롤링
- beautifulsoup를 통해 네이버뉴스크롤링 프로그램 제작
- Wordcloud, 자연어처리 라이브러리를 통해 데이터 시각화

제2주차 스터디 과제

네이버뉴스 댓글 수집 프로그램

\$ keyword

Input keyword :

Input max news count :

Input max comment count :

→ 키워드와, 수집할 뉴스링크 갯수,
수집할 댓글 갯수를 사용자로부터 입력받음

\$ wordcloud

Input csv file :

→ csv파일을 열어 데이터 시각화,
가장 많이 쓰인 단어

\$ load

Input csv file :

문제1

사용자로부터 키워드(ex : 안철수)와 크롤링할 뉴스기사 갯수를 입력받고, 네이버 뉴스검색 페이지에서 네이버뉴스링크를 크롤링해오시오.



링크주소를 크롤링 해오는것

Hint ! 네이버뉴스에, 키워드(안철수) 검색시 나오는 링크

<https://search.naver.com/search.naver?ie=utf8&where=news&query=안철수&start=19>

문제2

문제1번에서 크롤링해온 네이버뉴스링크(ex : <http://news.naver.com/main/read.nhn?mode=LSD&mid=sec&sid1=100&oid=001&aid=0009573718>)에 접속해 아이디,댓글달린시각,댓글내용을 크롤링해오시오. 크롤링 한 후 csv파일로 저장하시오.

(이때 csv파일명은 키워드+현재시각으로 하시오. 각 열은 키워드,아이디,시각,댓글내용으로 설정하시오)

✓ 호감순 최신순 공감비율순 답글순 과거순 ⓘ

ucan****

원피아 = MB = MB아바타. 여름에 그리 더워도 전력난도 없드라.

2017-09-26 15:36 | 접기요청

답글

94 25

sneo****

신고리 중단했는데도 올해 블랙아웃 한번도 없었음 리명박그네정부에서 실수요량 뺏튀기해서 건설사들 배불리다
고 생썬 한거임

2017-09-26 15:33 | 접기요청

답글 1

97 27

ssg4**** 댓글모음 >

헛소리 그만해 5 6호기는 공론화 중이거든 니말에 귀쫂긋 유치원생인줄아냐 초딩아

2017-09-26 15:34 | 접기요청

답글 3

85 23

댓글내용을 크롤링 해오는것

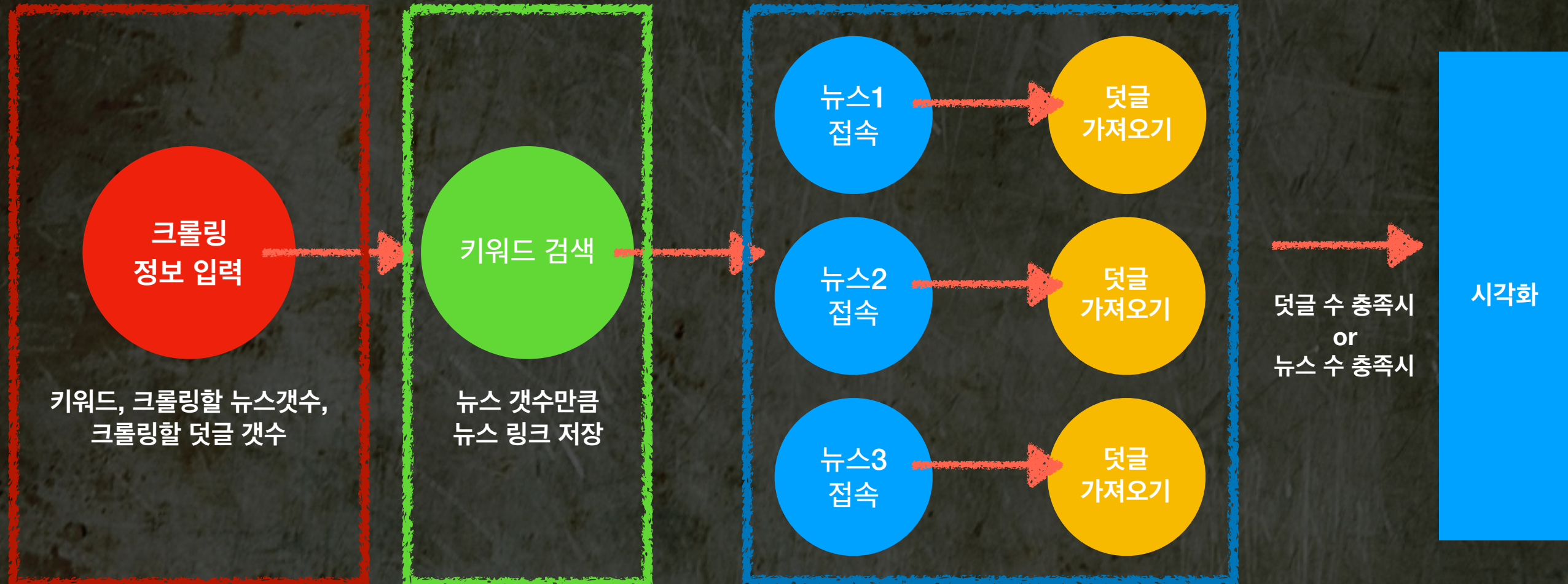
문제3

문제2에서 저장한 csv파일을 불러와 덧글의 단어를 워드클라우드를 사용해 시각화하라

1단계(main)

2단계(search_keyword)

3단계(get_comments)



메인 함수 설정

```

if __name__ == '__main__':
    comment_list = []

    keyword = input("$ 키워드를 입력해주세요 : ")
    news = input("$ 크롤링해올 뉴스 갯수를 입력해주세요 : ")
    comments = int(input("$ 크롤링해올 댓글 갯수를 입력해주세요 : "))
    news_links = keyword_search(keyword, news)

    comment_count = 0
    d = {}

    for news_link in news_links:
        l, flag = get_comments(news_link, comment_count, comments)
        comment_list.extend(l)
        comment_count = int(comment_count + len(l))

        if flag is True:
            break

    analyze(comment_list, keyword)

```

라이브러리 설정

```

import requests
from bs4 import BeautifulSoup
from urllib import parse
import nltk
from konlpy.tag import Twitter
from matplotlib import font_manager, rc
font_name = font_manager.FontProperties(fname="/Library/Fonts/AppleGothic.ttf").get_name()
rc('font', family=font_name)
from wordcloud import WordCloud
import matplotlib.pyplot as plt

```


keyword_search(keyword, page_count)**키워드 검색 및 뉴스링크 저장 함수**

```
def keyword_search(keyword, page_count):

    naver_news_links = []
    i = 0
    j = 1

    headers = {
        'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_4)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/61.0.3163.100 Safari/537.36'
    }

    while True:
        r = requests.get(
            "https://search.naver.com/search.naver?where=news&sm=tab_jum&query="
+ parse.quote(keyword) + "&start= " + str(j),
            headers=headers)
        c = r.content
        soup = BeautifulSoup(c, "html.parser")

        news_list = soup.find_all("a", {"class": "_sp_each_url"})

        for news_link in news_list:
            if news_link.text == "네이버뉴스":
                naver_news_links.append(news_link['href'])
                i = i + 1
                if i == int(page_count):
                    return naver_news_links

        j = j * 10
```


댓글 내용 크롤링 및 리스트화 3단계

```
def get_comments(news_link, comment_count, user_comment_count):

    comment_list = []

    headers = {
        'User-Agent': 'Mozilla/5.0 (Macintosh; Intel Mac OS X 10_12_4) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/
61.0.3163.100 Safari/537.36',
        'referer': 'http://news.naver.com/main/read.nhn?
mode=LPOD&mid=sec&oid=001&aid=0009572260&isYeonhapFlash=Y&rc=N&m_view=1&includeAllCount=true&m_url=%2Fcomment%2Fall.nhn%3Fservice
Id%3Dnews%26gno%3Dnews001%2C0009572260%26sort%3Dlikability'
    }

    aid = news_link.split("aid=")[1]
    oid = news_link.split("oid=")[1].split("&")[0]

    page = 1
    while True:
        r = requests.get(
            "https://apis.naver.com/commentBox/cbox/web_neo_list_jsonp.json?
ticket=news&templateId=default_politics&pool=cbox5&_callback=jQuery17023240944630416482_1506390886908&lang=ko&country=&objectId=n
ews" + oid + "%2C" + aid + "&categoryId=&pageSize=20&indexSize=10&groupId=&listType=OBJECT&page=" + str(page) +
"&sort=FAVORITE&current=1079250985&prev=1079229065&includeAllStatus=true&_1506390900990",
            headers=headers)
        c = r.content
        soup = BeautifulSoup(c, "html.parser")

        c_count = int(int(str(soup).split('{ "comment":') [1].split(",") [0]) / 20)

        contents = str(soup).split('"contents":')
        for i in range(1, len(contents)):
            user_name = contents[i].split('userName:') [1].split(",") [0]

            comment_content = contents[i].split('"", "userIdNo") [0]
            comment_time = contents[i].split('"modTime":') [1].split("'") [0]

            d["user_name"] = user_name
            d["time"] = comment_time
            d["comment_content"] = comment_content

            comment_count = comment_count + 1

            comment_list.append(comment_content)
            if int(user_comment_count) == int(comment_count):
                return comment_list, True

        if c_count < 1 or c_count == page:
            return comment_list, False
        else:
            page = page + 1
```


시각화

```
t=Twitter()
```

```
def analyze(content, keyword):  
    nouns = t.nouns(str(content))  
    ko=nltk.Text(nouns,name="분석")  
    ranking=ko.vocab().most_common(100)  
    tmpData=dict(ranking)  
    wordcloud=WordCloud(font_path="/Library/Fonts/  
AppleGothic.ttf",relative_scaling=0.2,background_color="white",)  
    .generate_from_frequencies(tmpData)  
    plt.figure(figsize=(16,8))  
    plt.imsave(keyword + ".png", wordcloud)  
    plt.imshow(wordcloud)  
    plt.axis("off")  
    plt.show()
```


시각화 결과



스터디 끝
파이썬/웹크롤링 개인과외 문의
카카오톡 : possible7202