

[개 발 - 자 연 어 처 리 팀]

1 주 차 진 행 사 항

- 참 고 논 문 정 독
- 크롤링 함수 코드 정리

(1) 참 고 논 문 정 독

참고 논문 정독

- 전체 모델 구조 및 개발 계획

1 월

2 월

뉴스 / SNS 데이터 수집
시스템

감성사전 구축 시스템

감성 분석 시스템

기계 학습 모델

• 크롤링 및 정제

• 형태소 분석
• 빈도수 및 긍정지수 계산

• 일별 긍정지수 계산

• 라벨링 및 학습

< 1 월 주차 별 세부 계획 >

주차	내용
1주차	뉴스 데이터 크롤링 함수 (1년 데이터)
2주차	SNS 데이터 크롤링 함수 (1년 데이터)
3주차	빈도수 및 긍정지수 계산 함수
4주차	레포트 주차

(1) 빈 도 수 계 산 공 식

$$include(i,j)=\begin{cases} 1 & \text{(기사 } j \text{ 에 단어 } i \text{ 가 포함된 경우)} \\ 0 & \text{(그 외의 경우)} \end{cases}$$

(1)

$$frequency(i)=\sum_{j=1}^n include(i,j), \text{ } n=\text{전체 기사의 수}$$

(2)

(2) 긍 정 지 수 계 산 공 식

$$NSP(j)=\begin{cases} 1 & \text{(기사 } j \text{ 가 게재된 후 익일 주가가 상승한 경우)} \\ 0 & \text{(그 외의 경우)} \end{cases}$$

(3)

$$positive(i)=\sum_{j=1}^n \{include(i,j) \times NSP(j)\}, \text{ } n=\text{전체 기사의 수}$$

(4)

$$P(i)=\frac{\sum_{j=1}^n \{include(i,j) \times NSP(j)\}}{\sum_{j=1}^n include(i,j)}, \text{ } n=\text{전체 기사의 수}$$

(5)

EX> 뉴스 10개를 크롤링한 경우

- **I n c l u d e (i , j)** :: 뉴스 1개 (j) 에 단 어 1 개 (i) 가 포 함 된 경 우 1
- **F r e q u e n c y (i)** :: 단 어 1 개 (i) 가 뉴 스 1 0 개 (n) 에 포 함 된 빈 도 수

- **N S P (j)** :: 기 사 1 개 (j) 와 익 일 주 가 의 관 계 익 일 주 가 가 상 승 한 경 우 1
- **p o s i t i v e (i)** :: 단 어 1 개 (i) 와 익 일 주 가 의 관 계 기 사 1 0 개 (n) 에 단 어 1 개 (i) 가 포 함 되 고 , 익 일 주 가 가 상 승 한 경 우 를 c o u n t i n g
- **P (i)** :: 단 어 1 개 (i) 의 빈 도 수 대 비 긍 정 수
→ (긍 정 지 수)

(3) 일 별 긍 정 지 수 계 산 공 식

$$match(i,j)=\begin{cases} 1 & \text{(텍스트 } i \text{ 에 포함된 명사 } j \text{ 가 감성사전에 존재 할 경우)} \\ 0 & \text{(그 외의 경우)} \end{cases}$$

(6)

$$PT(i)=\frac{\sum_{j=1}^n\{match(i,j)\times P(j)\}}{\sum_{j=1}^nmatch(i,j)}, \text{ } n=\text{텍스트 } i \text{ 에 포함된 단어의 수}$$

(7)

$$DP(k)=\frac{\sum_{i=1}^nPT(i)}{n}, \text{ } n=k \text{ 일에 게재된 텍스트의 수}$$

(8)

E X > 일 일 뉴 스 1 0 개 를 크 롤 링 한 경 우

- **m a t c h (i , j)** :: 뉴 스 1 개 (i) 의 단 어 1 개 (j) 가 감 성 사 전 에 포 함 된 경 우 1
- **P T (i)** :: 뉴 스 1 개 의 긍 정 지 수 (포 함 된 단 어 의 긍 정 지 수 들 의 평 균)
- **D P (k)** :: 뉴 스 1 0 개 (n) 의 긍 정 지 수 평 균
→ (일 별 긍 정 지 수)

(2) 크 롬 링 코 드 공 유

크롤링 전체 구조

Read_naver_news

각 기사 별 url 크롤링

Url_list 반환

News_content_crawling

Url에 접근하여 필요한
데이터 크롤링

데이터 -> csv 파일로 저장

url list

01.05(수)	
경제	글로벌경제
금융	
증권	<div><p>OPEC+, 2월에도 하루 40만 배럴 증산규모 유지 [이데일리 장영은 기자] 석유수출국기구(OPEC)와 러시아 등 비(非)OPEC 주요 산유국들의... 이데일리 ① 2분전</p></div>
산업/재계	
중기/벤처	
부동산	<div><p>전력난 인니, 750만t 석탄 추가 확보...수출 금지 해제되나 (서울=뉴스1) 정윤미 기자 = 인도네시아가 750만톤(t) 규모 석탄 공급량을 추가 확보해 ... 뉴스1 ① 5분전</p></div>
글로벌 경제	
생활경제	
경제 일반	<div><p>“글로벌 공급망 혼란 점점 짙고 완화될 것” [이데일리 신채연 인턴기자] 글로벌 공급망 혼란이 정점에 다다랐다는 전망이 나왔다. ... 이데일리 ① 21분전</p></div>
프리미엄콘텐츠	
투자 정보부터 트렌드까지!	
바로가기 >	
모바일 메인에서 보고싶은 뉴스 구독하세요!	<div><p>대만 코로나 신규환자 26명·본토 1명 총 1만7155명...17일째 사망 無 [서울=뉴스1]이재준 기자 = 지난해 5월 중순 이래 코로나19가 급속히 퍼졌다가 진정세... 뉴스1 ① 30분전</p></div>
바로가기 >	
	<div><p>버핏, 시총 3조 달러 애플 투자 '대박'...6년 평가차익 150조원 (서울=뉴스1) 신기림 기자 = '투자의 귀재' 워런 버핏 버크셔해셔웨이 회장이 애플 투자... 뉴스1 ① 35분전</p></div>
	<div><p>달러약세·인플레이션·지정학적 불안... 연내 금값 2100달러 가능성 국제금값이 올해 31.1g(온스)당 2100달러까지 상승할 것이라는 전망이 나왔다. 4일(현지... 파이낸셜뉴스 ① 45분전</p></div>
	<div><p>찰리 멩거, 알리바바 주식만 857억원...최근 2배 늘려 [서울=뉴스1] 최영민 기자 = 워런 버핏의 오른팔로 불리는 찰리 멩거가 알리바바 주식을 2배 늘려 ... 뉴스1 ① 1분전</p></div>

Read_naver_news

```
def read_naver_news(self): #뉴스 url list 반환 함수
    date = (datetime.today() - timedelta(8)).strftime("%Y%m%d") # 12월 14일 기준
    page = 1 # 초기 페이지
    last_page = 8 # 최종 페이지
    count = 0 # 반복 횟수 카운팅
    url_list = [] # 각 뉴스의 url 리스트

    try:
        while True:

            # 종료 조건
            if count == 10 : break # 10 page 크롤링 후, 종료

            # url 업데이트
            if page > last_page : # last_page이면 이전 날짜로 업데이트
                date = (datetime.today() - timedelta(9)).strftime("%Y%m%d") # 12월 13일
                page = 1

            # 네이버 금융 글로벌 경제 url
            url=f"https://news.naver.com/main/list.naver?mode=LS2D&sid2=262&mid=shm&sid1=101&date={date}&page={page}"
            html_news = BeautifulSoup(requests.get(url, verify=False, headers = {'User-agent': 'Mozilla/5.0'}).text, 'lxml')

            # type06 headline html 가져오기 (윗 문단)
            ul_tag = html_news.find("ul",class_="type06_headline")
            a_tag = ul_tag.findAll("a")
            #각 기사별 url 가져오기
            for i in range(len(a_tag)):
                news_url = a_tag[i]['href'] # url 원소 1개씩 저장
                if i == 0:
                    url_list.append(news_url)
                elif a_tag[i-1]['href'] != a_tag[i]['href'] : # url 중복 방지
                    url_list.append(news_url)

            if page == last_page: # last page 에 type 06이 없어서 continue
                count+=1
                page+=1
                continue

            # type06 html 가져오기 (아래 문단)
            ul_tag = html_news.find("ul",class_="type06")
            a_tag = ul_tag.findAll("a")

            #각 기사별 url 가져오기
            for i in range(len(a_tag)):
                news_url = a_tag[i]['href']
                if i == 0:
                    url_list.append(news_url)
                elif a_tag[i-1]['href'] != a_tag[i]['href'] :
                    url_list.append(news_url)

            count+=1 # url 크롤링 완료 후, 데이터 업데이트
            page+=1

        return url_list # url list 반환

    except Exception as e: # 에러 발생 시, 에러 출력
        print('Exception occured :', str(e))
        return None
```

← → ↺

news.naver.com/main/list.naver?mode=LS2D&mid=shm&sid2=262&sid1=101&date=20220105&page=2

앱

김혜정

Sejong Univ.

N 네이버 금융

My DashBoard

뉴스

Gmail

Papago

YouTube

HLS를

N 뉴스

연예

스포츠

날씨

프리미엄

언론사별

정치

경제

사회

생활/문화

IT/과학

세계

기타

01.05(수)

경제

금융

증권

산업/재계

중기/벤처

부동산

글로벌 경제

생활경제

경제 일반


프리미엄콘텐츠

투자 정보부터 트렌드까지!


바로가기 >

모바일 앱에서


글로벌경제




세계 500대 부자 지난해 자
지난해에 세계 500대 부자들의
파이낸셜뉴스 | ① 2시간전



[올댓차이나] 상하이 증시 C
[서울=뉴시스]이재준 기자 = 세
뉴시스 | ① 2시간전



중국, 디지털위안화 앱 베타
[베이징=이데일리 신정은 특파원
이데일리 | ① 2시간전



러시아, 전기차 시어 부경시

Read_naver_news

```
def read_naver_news(self): #뉴스 url list 반환 함수
    date = (datetime.today() - timedelta(8)).strftime("%Y%m%d") # 12월 14일 기준
    page = 1 # 초기 페이지
    last_page = 8 # 최종 페이지
    count = 0 # 반복 횟수 카운팅
    url_list = [] # 각 뉴스의 url 리스트

    try:
        while True:

            # 종료 조건
            if count == 10 : break # 10 page 크롤링 후, 종료

            # url 업데이트
            if page > last_page : # last_page이면 이전 날짜로 업데이트
                date = (datetime.today() - timedelta(9)).strftime("%Y%m%d") # 12월 13일
                page = 1

            # 네이버 금융 글로벌 경제 url
            url=f"https://news.naver.com/main/list.naver?mode=LS2D&sid2=262&mid=shm&sid1=101&date={date}&page={page}"
            html_news = BeautifulSoup(requests.get(url, verify=False, headers = {'User-agent' : 'Mozilla/5.0'}),text,"lxml")

            # type06 headline html 가져오기 (위 문단)
            ul_tag = html_news.find("ul",class_="type06_headline")
            a_tag = ul_tag.findAll("a")
            #각 기사별 url 가져오기
            for i in range(len(a_tag)):
                news_url = a_tag[i]['href'] # url 원소 1개씩 저장
                if i == 0:
                    url_list.append(news_url)
                elif a_tag[i-1]['href'] != a_tag[i]['href'] : # url 중복 방지
                    url_list.append(news_url)

            if page == last_page: # last page 에 type 06이 없어서 continue
                count+=1
                page+=1
                continue

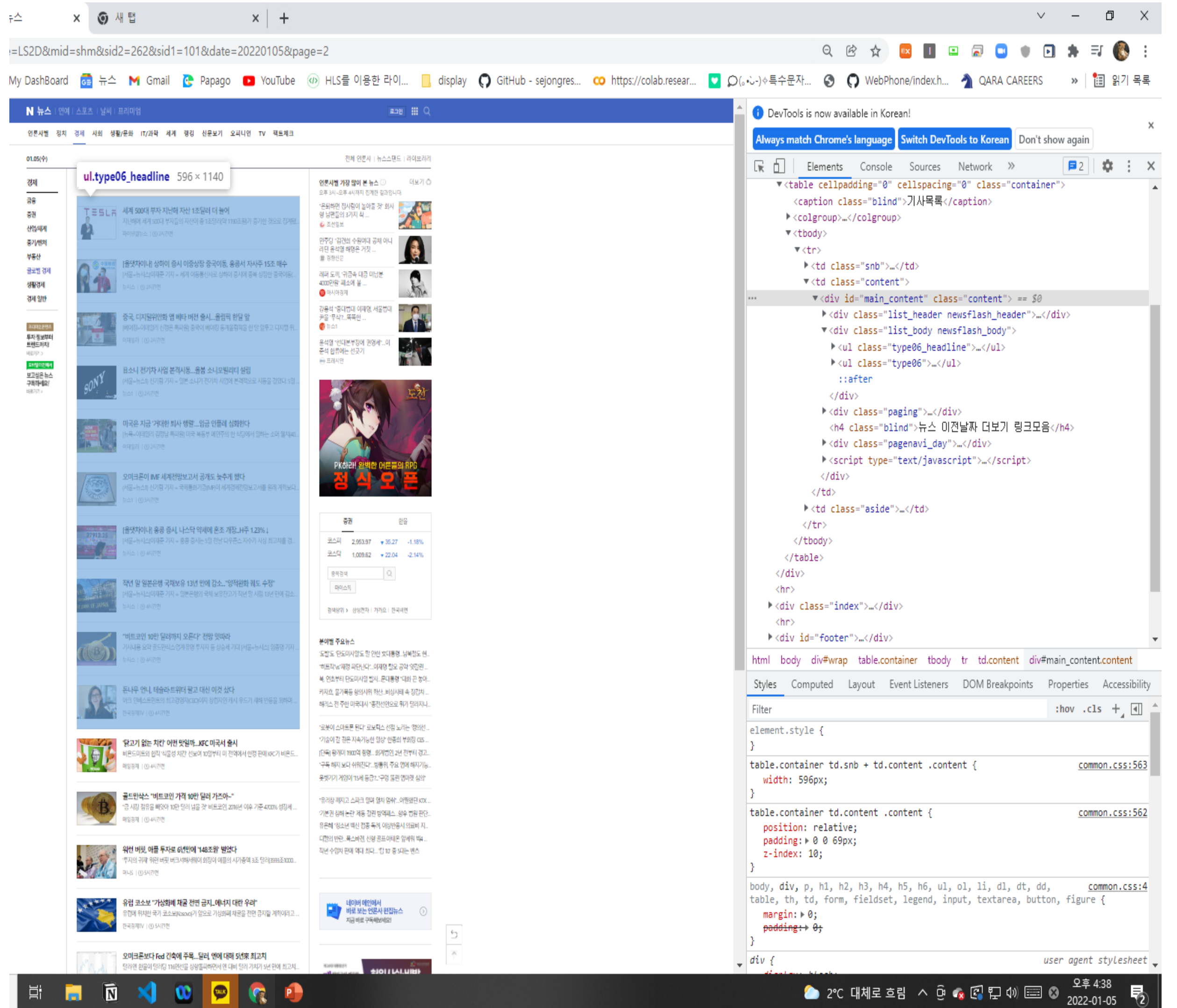
            # type06 html 가져오기 (아래 문단)
            ul_tag = html_news.find("ul",class_="type06")
            a_tag = ul_tag.findAll("a")

            #각 기사별 url 가져오기
            for i in range(len(a_tag)):
                news_url = a_tag[i]['href']
                if i == 0:
                    url_list.append(news_url)
                elif a_tag[i-1]['href'] != a_tag[i]['href'] :
                    url_list.append(news_url)

            count+=1 # url 크롤링 완료 후, 데이터 업데이트
            page+=1

        return url_list # url list 반환

    except Exception as e:
        print('Exception occured :', str(e))
        return None # 에러 발생 시, 에러 출력
```



Read_naver_news

```
def read_naver_news(self): #뉴스 url list 반환 함수
    date = (datetime.today() - timedelta(8)).strftime("%Y%m%d") # 12월 14일 기준
    page = 1 # 초기 페이지
    last_page = 8 # 최종 페이지
    count = 0 # 반복 횟수 카운팅
    url_list = [] # 각 뉴스의 url 리스트

    try:
        while True:

            # 종료 조건
            if count == 10 : break # 10 page 크롤링 후, 종료

            # url 업데이트
            if page > last_page : # last_page이면 이전 날짜로 업데이트
                date = (datetime.today() - timedelta(9)).strftime("%Y%m%d") # 12월 13일
                page = 1

            # 네이버 금융 글로벌 경제 url
            url=f"https://news.naver.com/main/list.naver?mode=LSD2&sid2=262&mid=shm&sid1=101&date={date}&page={page}"
            html_news = BeautifulSoup(requests.get(url, verify=False, headers = {'User-agent' : 'Mozilla/5.0'}),text,"lxml")

            # type06 headline html 가져오기 (위 문단)
            ul_tag = html_news.find("ul",class_="type06_headline")
            a_tag = ul_tag.findAll("a")
            #각 기사별 url 가져오기
            for i in range(len(a_tag)):
                news_url = a_tag[i]['href'] # url 원소 1개씩 저장
                if i == 0:
                    url_list.append(news_url)
                elif a_tag[i-1]['href'] != a_tag[i]['href'] : # url 중복 방지
                    url_list.append(news_url)

            if page == last_page: # last page 에 type 06이 없어서 continue
                count+=1
                page+=1
                continue

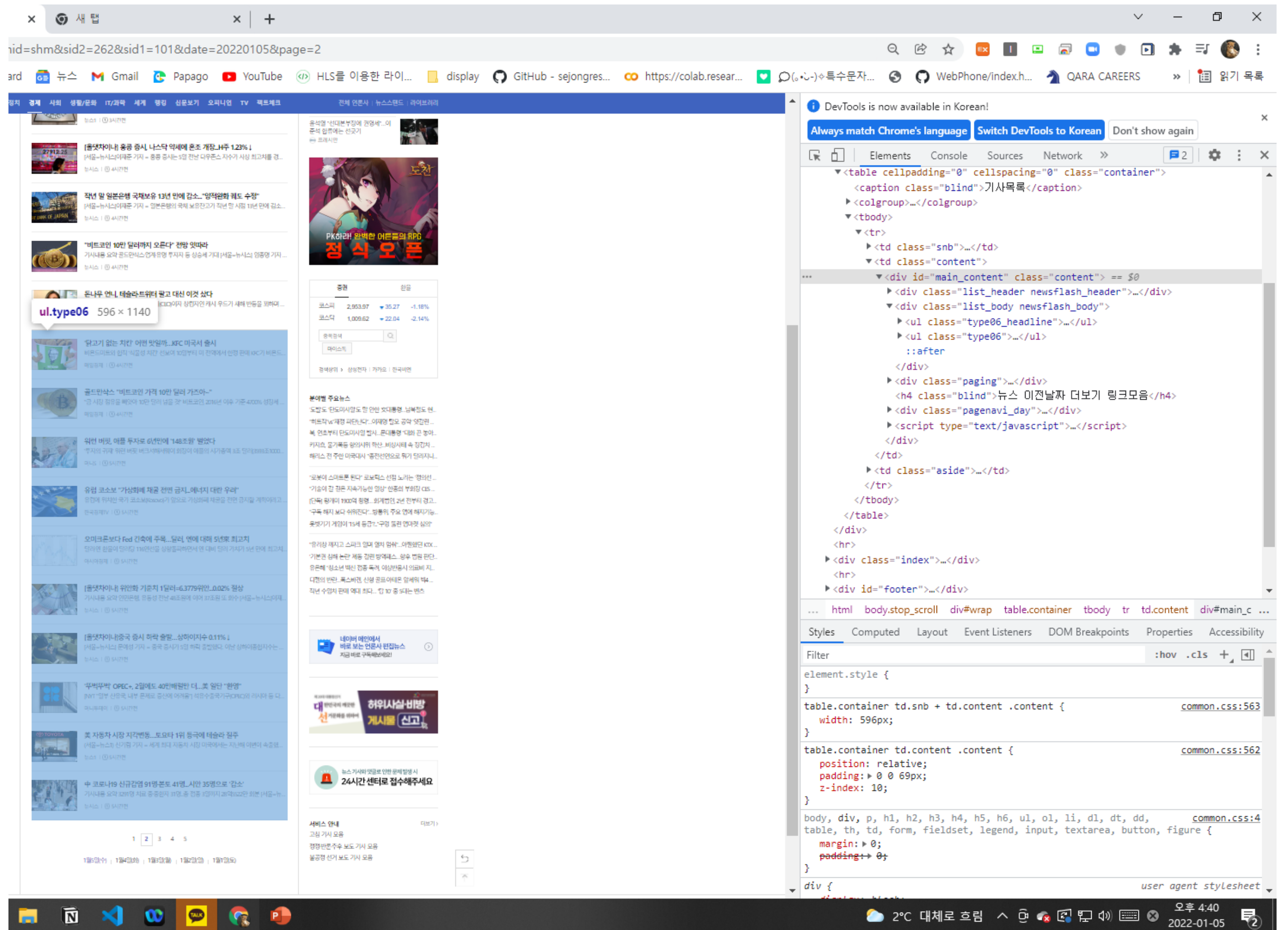
            # type06 html 가져오기 (아래 문단)
            ul_tag = html_news.find("ul",class_="type06")
            a_tag = ul_tag.findAll("a")

            #각 기사별 url 가져오기
            for i in range(len(a_tag)):
                news_url = a_tag[i]['href']
                if i == 0:
                    url_list.append(news_url)
                elif a_tag[i-1]['href'] != a_tag[i]['href'] :
                    url_list.append(news_url)

            count+=1 # url 크롤링 완료 후, 데이터 업데이트
            page+=1

        return url_list # url list 반환

    except Exception as e:
        print('Exception occured :', str(e))
        return None # 에러 발생 시, 에러 출력
```



News_content_crawling - Date 태그

News_content_crawling - title 태그

</

News_content_crawling - articleBodyContents 태그

News_content_crawling - 해당 태그 찾기

```
def news_content_crawling(self): # 뉴스 데이터 크롤링 함수
    (module) pd
    df = pd.DataFrame(columns=["date","title","content"]) # df 생성
    url_list = news.read_naver_news() # url 읽어오기

    for i in range(len(url_list)):

        url=url_list[i] # 1개의 url 가져오기
        html_news = BeautifulSoup(requests.get(url, verify=False, headers = {'User-agent' : 'Mozilla/5.0'}).text,"xml") #parser

        time = html_news.find("span",class_="t11").get_text() # time 긁어오기
        title = html_news.find("title").get_text() # title 긁어오기
        content = html_news.find(id="articleBodyContents").get_text() # 뉴스 본문 긁어오기

        data_list = {"date":time, "title":title, "content":content} # series 형식으로 데이터 저장
        df = df.append(data_list, ignore_index=True) # 데이터 프레임에 추가

    print('{} / {} pages are downloading...'.format(i+1,len(url_list))) # 다운로드 현황 출력


    df = df[["date","title","content"]] # column 순서 맞추기
    print(df)

    #csv 파일로 저장
    dataframe = pd.DataFrame(df)
    dataframe.to_csv("C:/Users/user/OneDrive/바탕 화면/AI_QUANT/news_raw.csv",header=False,index=False)

    #self.replace_into_db(df) # DB 업데이트 함수 실행
```

감사합니다 😊