

# 크롤링 함수 코드

2022.01.05

김혜정

[개발 – 자연어처리 팀]

# 1주차 진행 사항

- 논문 정독
- 크롤링 함수 코드 정리

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



# 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

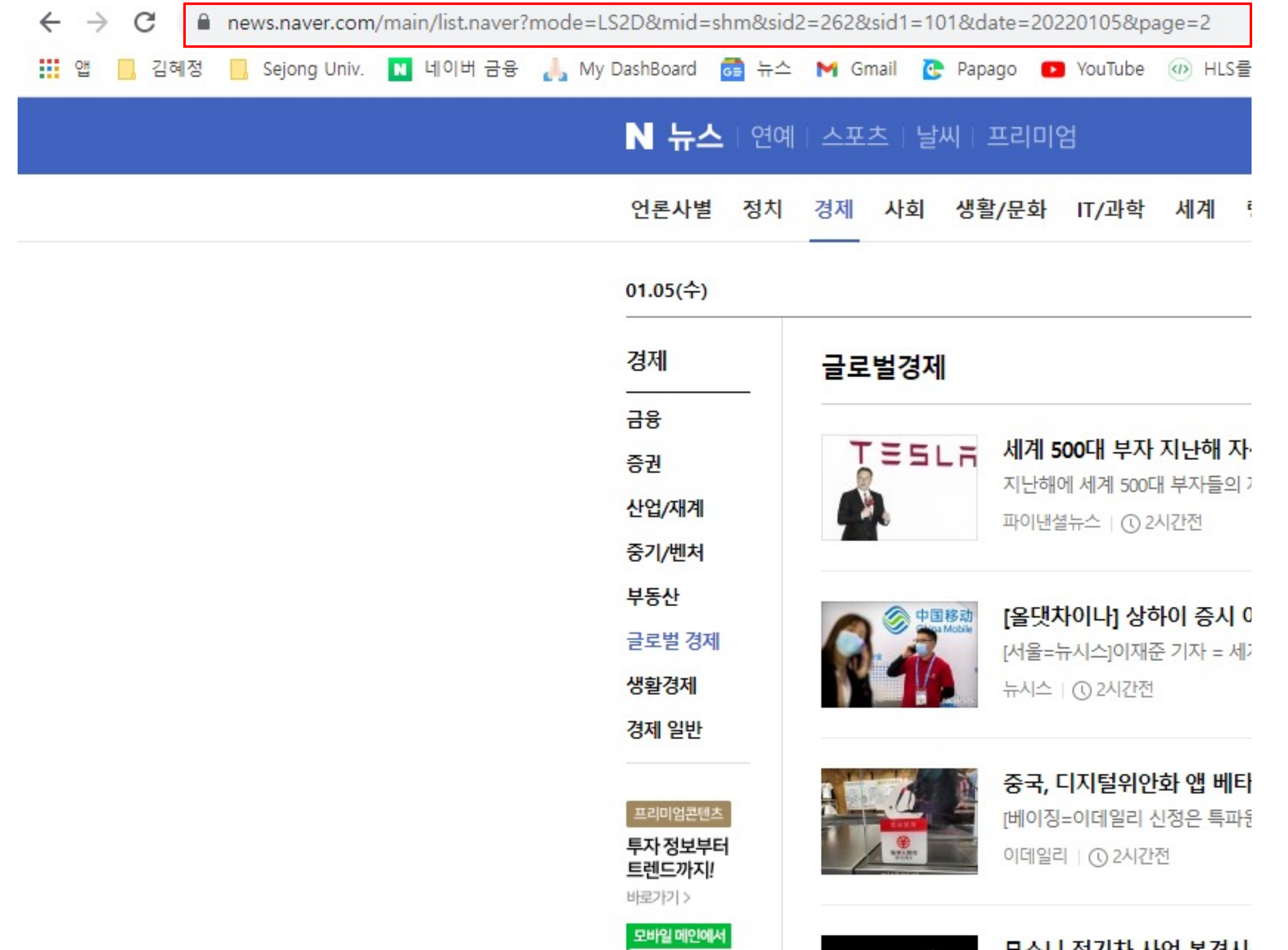
            # type06 html 가져오기 (아래 문단)
            ul_tag = html_news.find("ul",class_="type06")
            a_tag = ul_tag.findAll("a")

            #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
```





# 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

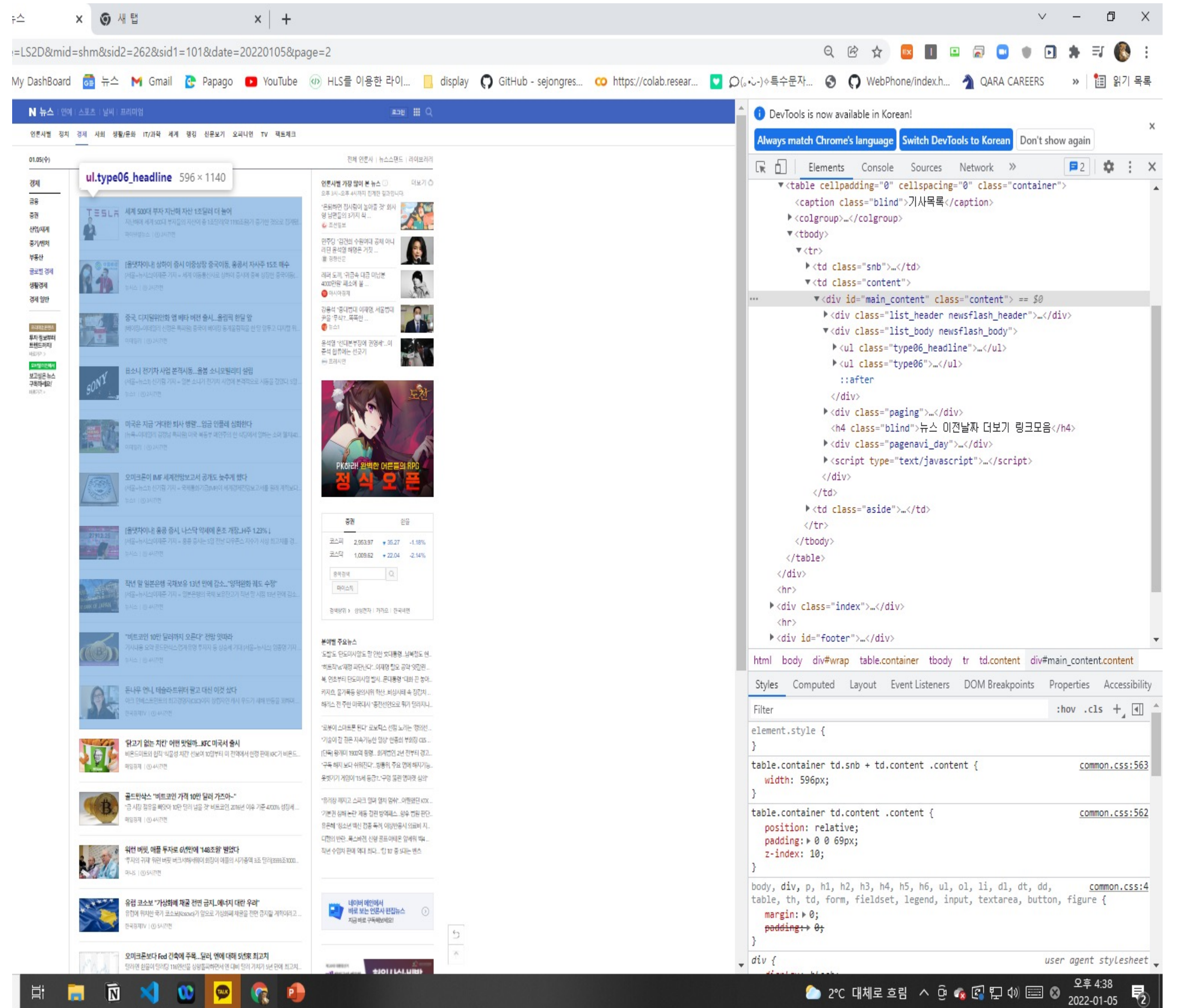
            # type06 html 가져오기 (아래 문단)
            ul_tag = html_news.find("ul",class_="type06")
            a_tag = ul_tag.findAll("a")

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

        return url_list # url list 반환

    except Exception as e:
        print('Exception occurred :', 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=LSD2D&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 (첫 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

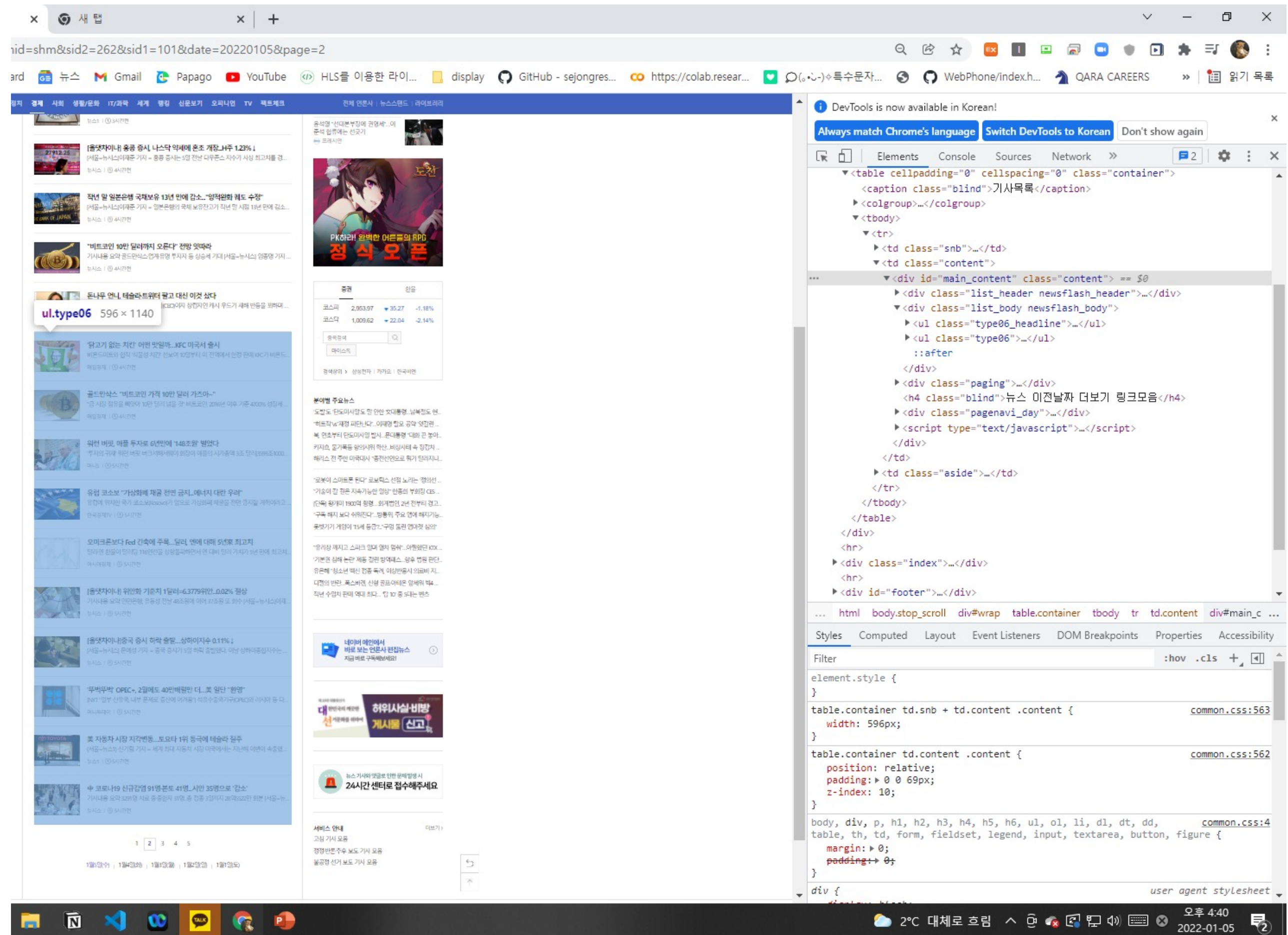
            # type06 html 가져오기 (아래 문단)
            ul_tag = html_news.find("ul",class_="type06")
            a_tag = ul_tag.findAll("a")

            #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

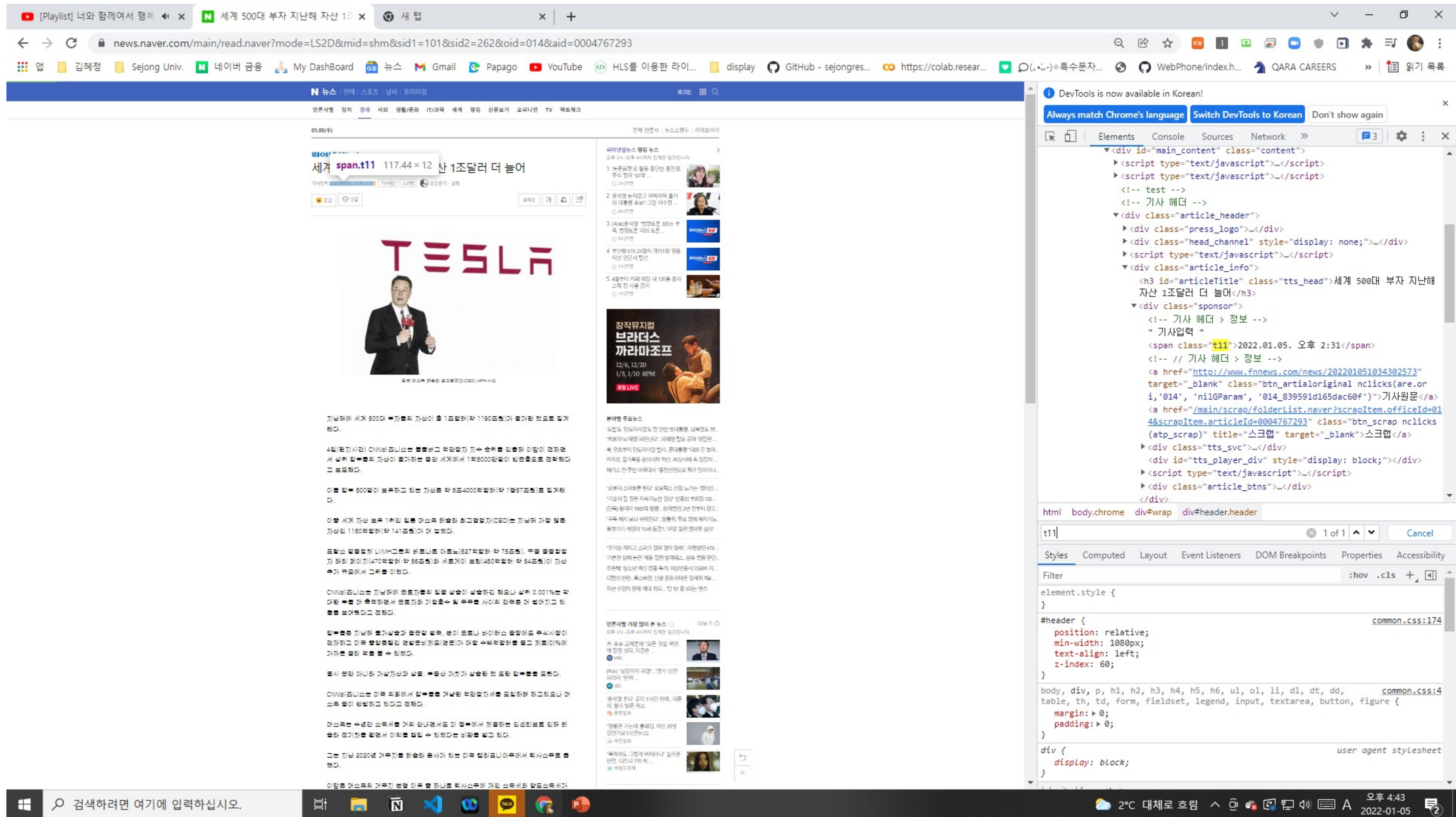
    except Exception as e: # 에러 발생 시, 에러 출력
        print('Exception occured :', str(e))
        return None
```





# News\_content\_crawling

## - Date 태그





# News\_content\_crawling - title 태그



# News\_content\_crawling

- articleBodyContents 태그

The screenshot displays a web browser window with the Naver news article "세계 500대 부자 지난해 자산 1조달러 더 늘어" (World's 500 richest families' assets increased by 1 trillion dollars last year). The article features a large image of Elon Musk and text in Korean. A tooltip points to the `div#articleBodyContents_article_body_contents.article_body_contents` element. The browser's address bar shows the URL: `news.naver.com/main/read.naver?mode=LS2D&mid=shm&sid1=101&sid2=262&oid=014&aid=0004767293`. The browser's taskbar at the bottom shows various open applications including VS Code, Edge, and Chrome. The Windows taskbar at the very bottom shows the date and time as 2022-01-05, 4:46 PM.

DevTools is now available in Korean!

Always match Chrome's language | Switch DevTools to Korean | Don't show again

Elements | Console | Sources | Network

```
i, '014', 'nilGParam', '014_839591d165dac60f')">기사원문</a>
<a href="/main/scrap/folderList.naver?scrapItem.officeId=014&scrapItem.articleId=0004767293" class="btn_scrap nclicks (atp_scrap)" title="스크랩" target="_blank">스크랩</a>
<div class="tts_svc">...</div>
<div id="tts_player_div" style="display: block;"></div>
<script type="text/javascript">...</script>
<div class="article_btns">...</div>
</div>
::after
</div>
<!-- // 기사 헤더 -->
<div class="article_body_font_setting_target size3 font1" id="articleBody">
  <div id="articleBodyContents" class="_article_body_contents article_body_contents" style="-webkit-tap-highlight-color: rgba(0,0,0,0)">...</div>
  <div class="byline">...</div>
  <div class="article_journalist">...</div>
  <script type="text/x-jindo-template" class="_recommend_layer_template">...</script>
  <script type="text/javascript">...</script>
  <div class="copyright">...</div>
  <!-- [D] .guide_categorization_title 내 링크가 클릭되면 .guide_categorization_ct display:block; 해주세요 -->
  <div class="guide_categorization">...</div>
  <div class="promotion">...</div>
```

html | head | title

articleBodyContents 1 of 2

Styles | Computed | Layout | Event Listeners | DOM Breakpoints | Properties | Accessibility

Filter :hov .cls +

```
element.style {
}
title {
  display: none;
}
Inherited from html
html {
  font-size: 10px;
}
```

margin -

border -

padding -

검색하려면 여기에 입력하십시오.

2°C 대체로 흐림

오후 4:46 2022-01-05



# 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, "lxml") #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 😊