웹 크롤링

2025.01.31

김자영 강사

웹 크롤링 vs 웹 스크래핑

웹크롤링(Web Crawling)	웹 스크래핑(Web Scraping)
✓ 자동화된 방식으로 웹페이지들을 순회하면서 링크를 따라 다니는 것 ✓ 검색 엔진의 봇(예: Googlebot)이 대표적인 예시 ✓ 주로 웹 사이트의 구조를 파악하고 새로운 페이지를 발견하는 것이 목적	✓ 웹페이지에서 원하는 데이터를 추출하는 기술 ✓ HTML 문서에서 특정 태그나 패턴을 찾아 정보를 수집 ✓ 가격 비교, 뉴스 수집, 제품 정보 추출 등에 활용

※ 위 두 기술이 함께 사용되어 웹 데이터를 수집.

크롤링 주의사항

법적 주의사항

- ✓ 웹사이트 콘텐츠는 대부분 저작권으로 보호됨
- ✓ 상당한 투자와 노력으로 구축된 데이터베이스는 제작자의 권리가 인정됨
- ✓ 무단 크롤링은 저작권법과 부정경쟁방지법 위반이 될 수 있음

기술적 주의사항

- ✓ 과도한 요청으로 서버에 부담을 주지 않도록 크롤링 간격 조정

  과도한 요청 시 영업방해로 인정될 수 있음.
- ✓ robots.txt 파일의 크롤링 규칙 확인 및 준수
- ✓ 기술적 제한을 우회하는 행위 금지

윤리적 고려사항

- ✓ 상업적 목적으로 사용 시 저작권자의 허가 필요
- ✓ 경쟁 업체의 데이터를 무단으로 활용하지 않도록 주의

크롤링 주의사항

▪ 대표적 소송 사례

야놀자 vs 여기어때 사건 (2023년)

- ✓ 여기어때가 야놀자의 숙박업소 정보를 무단 크롤링하여 사용
- ✓ 민사: 10억원 배상 판결 확정
- ✓ 형사: 여기어때 무죄 판결

잡코리아 vs 사람인 사건

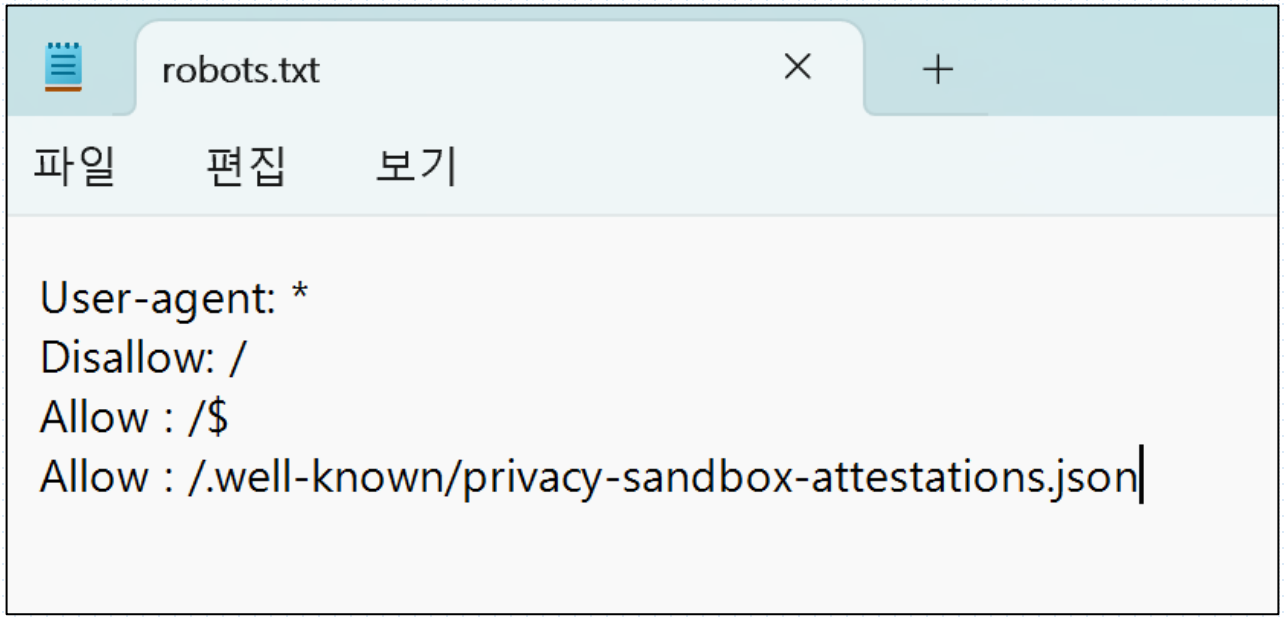
- ✓ 사람인이 잡코리아의 채용정보를 크롤링하여 웹사이트에 게시
- ✓ HTML 소스의 텍스트는 저작권 보호대상이 아니나, 데이터베이스 제작자 권리 인정
- ✓ 상당한 투자와 노력으로 구축한 데이터베이스의 무단 크롤링은 부정경쟁행위로 판단

크롤링 주의사항

▪ robot.txt 확인 방법

웹브라우저 주소창에 해당 웹사이트 주소 뒤에 /robots.txt를 입력하여 robots.txt 파일 다운로드

(예) http://www.naver.com/robots.txt



- **User-agent** : 규칙을 적용할 크롤러
- **Disallow** : 접근 차단할 경로
- **Allow** : 접근 허용할 경로
- **Crawl-delay** : 크롤링 요청 간 대기 시간(초)

※ robot.txt는 권고 사항이며 강제성은 없음
이 규칙을 준수하는 것은 웹 크롤링의 에티켓이나,
법정 분쟁 발생 시에는 고려 대상이 될 수 있음

크롤링을 위한 웹페이지의 유형

정적 페이지

한 번에 모든 HTML 문서를 응답 받는 페이지

사용 라이브러리

- requests
  - http 요청을 보내고 응답받음
  - 웹브라우저 열지 않음
- BeautifulSoup
  - html문서 파싱, 데이터 추출

동적 페이지

HTML을 응답 받은 후 일부 내용을 JavaScript가 동적으로생성하는 페이지

사용 라이브러리

- selenium
  - 웹브라우저를 열어 동적 콘텐츠 로드
  - 웹요소로부터 데이터 추출
  - html문서 파싱, 데이터 추출

동적 페이지 크롤링

❶ 웹드라이버 생성

❷ 웹드라이버의 브라우저를 이용하여
목표 URL로 이동

❸ 동적 콘텐츠 로딩

❹ 필요 시 상호작용(클릭, 스크롤, 검색 등)

❺ 데이터 추출

❻ 데이터 저장

라이브러리 설치하기

- Selenium

Selenium 설치하기

```
pip install selenium==4.17.0
```

셀레니움 버전 확인

```
pip show selenium
```

웹드라이버 객체 생성

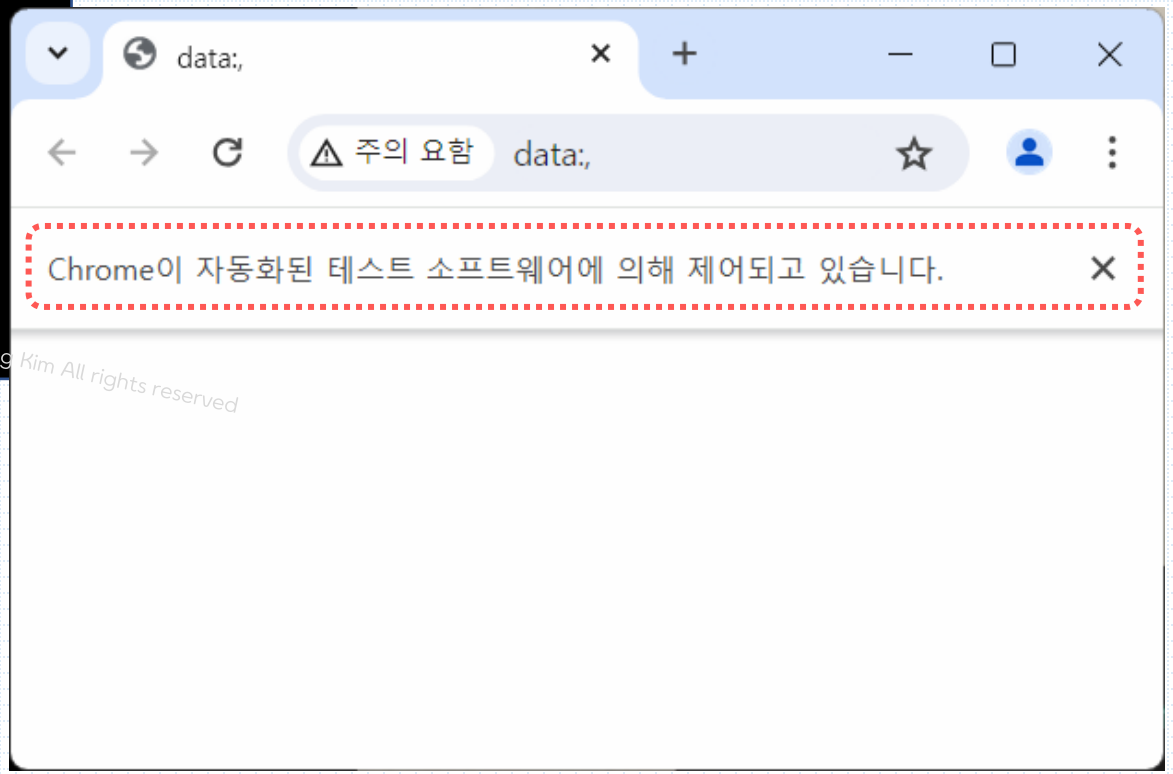
- 크롬 브라우저를 조작하기 위한 웹드라이버 객체를 생성합니다.


```
# 라이브러리 import : selenium 웹드라이버
from selenium.webdriver import Chrome


# 크롬 웹드라이버 객체 생성
driver = Chrome()
```


브라우저 조작을 위한 객체

⭐브라우저를 닫으면 안되요.



브라우저 제어하기

- 브라우저 크기 제어하기


```
# 브라우저 크기 최소화
driver.minimize_window()
```

```
# 브라우저 크기 최대화
driver.maximize_window()
```

브라우저 제어하기

- 원하는 페이지로 이동합니다.


```
# 원하는 페이지로 이동하기
driver.get("https://www.naver.com")


driver.get("https://www.google.com")


driver.get("https://www.daum.net")
```

브라우저 제어하기

- 이전 페이지로 이동, 다음 페이지로 이동, 새로고침


```
# 이전 페이지로 이동
driver.back()

# 다음 페이지로 이동
driver.forward()

# 새로고침
driver.refresh()
```

브라우저 제어하기

- 브라우저 닫기


```
# 웹드라이버로  생성한 현재 창 종료
driver.close()
```



```
# 웹드라이버로  생성한  모든 창 종료
driver.quit()
```

브라우저 제어하기

- error


```
NoSuchWindowException: Message: no such window: target window already closed
```

→ 웹드라이버로 생성한 브라우저를 닫았을 때 발생

웹페이지에서 가져올 요소 선택하기

▪ 웹페이지의 구성 요소

태그 (tag)	• html 요소를 정의하는 데 사용되는 코드. 시작태그, 종료태그가 있음. (예) hello , <img src="img.jpg">,
요소 (element)	• 태그와 태그 사이에 있는 내용을 합쳐서 '요소(element)'라고 부른다. • 내용을 갖는 요소, 내용을 갖지 않는 요소가 있음 (예) hello , <img src="img.jpg">,
속성 (attribute)	• 요소에 추가적인 정보를 제공하는 부분. • 열린 태그에서 정의. (속성이름=속성값) (예) <img src="img.png">
내용 (content)	• 시작태그와 종료태그 사이에 위치한 내용. 내용을 갖는 태그, 갖지 않는 태그가 있음. (예) hello

웹페이지에서 가져올 요소 선택하기

▪ CSS 선택자(CSS selector)

선택자 종류	스타일 적용 범위	사용법
tag 선택자 Type Selector	특정 태그를 사용한 모든 요소	태그명
class 선택자 Class Selector	동일한 class명을 사용하는 모든 요소	.class명
id 선택자 ID Selector	해당 id를 가진 단일 요소 ID는 페이지에서 유일해야 함	#id명
그룹 선택자 Group Selector	쉼표로 구분된 모든 선택자에 해당하는 요소	선택자1,…, 선택자n

https://developer.mozilla.org/ko/docs/Web/CSS/CSS_selectors

CSS 선택자(CSS selector)

▪ 형제 선택자

인접 형제 선택자 (Adjacent Sibling Selector)

선택자1 + 선택자2

선택자1 바로 뒤에 위치한 선택자2 선택

일반 형제 선택자 (General Sibling Selector)

선택자1 ~ 선택자2

선택자1 뒤에 위치한 모든 선택자2 선택

CSS 선택자(CSS selector)

▪ n번째 태그만 선택

특정 유형(type)의 형제요소 중 특정 순서에 있는 요소 선택

선택자:nth-of-type(n)


```
p:nth-of-type(3)        # 3번째 p요소 선택
p:nth-of-type(even)     # 짝수번째 p요소 선택
p:nth-of-type(odd)      # 홀수번째 p요소 선택
p:nth-of-type(3n)       # 3의 배수번째 p 요소 선택
```

CSS 선택자(CSS selector)

▪ 부정 선택자

괄호 안에 지정된 선택자와 일치하지 않은 요소 선택

찾을선택자:not(제외할선택자)


```
a:not(.example)              # a요소 중 example 클래스 제외
div:not(.example, #myid)     # div요소 중  exampl클래스, myid 아이디 제외
```

CSS 선택자(CSS selector)

▪ 속성 선택자

[속성]

요소[속성]

[속성=속성값]

[속성^=시작값]

[속성$=끝값]

[속성*=포함값]

웹페이지에서 가져올 요소 선택하기

- 브라우저 띄우기 : 웹드라이버 객체 생성하기

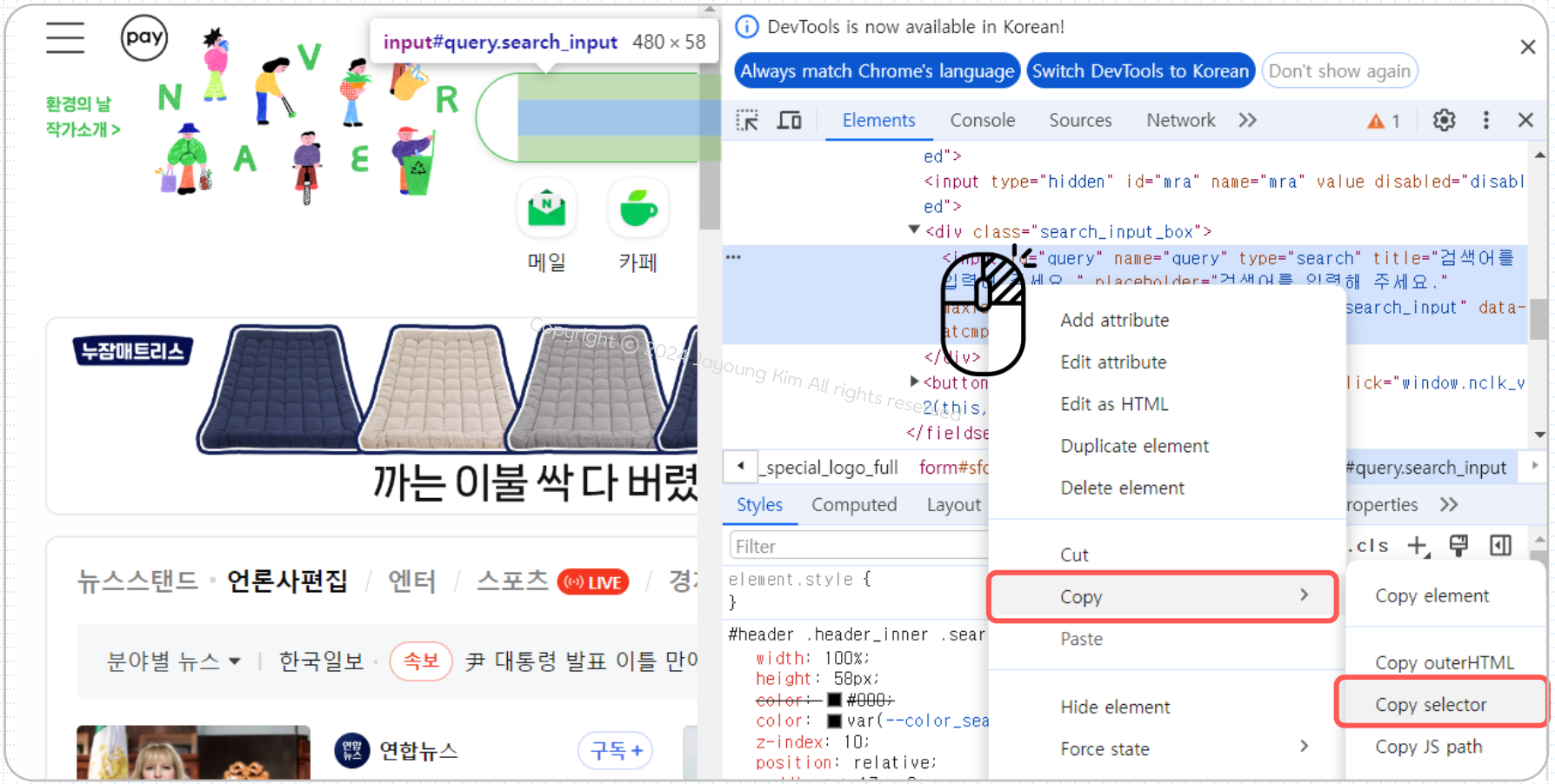

```
# 웹드라이버 객체 생성
from selenium import webdriver
driver = webdriver.Chrome()
driver.get('http://www.naver.com')
driver.maximize_window()
```

웹페이지에서 가져올 요소 선택하기

- 웹 요소 찾기 - 웹 요소의 CSS Selector 복사

웹페이지에서 가져올 요소 선택하기

- 웹 요소 찾기 - 웹 요소의 CSS Selector 복사



웹페이지에서 요소 가져오기

- CSS Selector로 요소 찾기

웹드라이버.find_element(요소지정방법, “선택자”)


```
# CSS선택자로 요소 객체 가져오기
from selenium.webdriver.common.by import By
search = driver.find_element(By.CSS_SELECTOR, "#query")
search
```



```
<selenium.webdriver.remote.webelement.WebElement
(session="9f04fdbf074f98cd3e4d61483c2c08b3",
element="f.AAA796954FE13F5F17CA3A5CC40D87F9.d.EC22A0E3A
13BFD8A6A3DE2BBBA9B5DED.e.93")>
```

반환된 요소 객체

- 웹요소를 선택하는 방법 지정
  - **By.CSS_SELECTOR**
  - By.ID
  - By.CLASS_NAME
  - By.NAME
  - By.XPATH
  - By.LINK_TEXT
  - By.PARTIAL_LINK_TEXT
  - By.TAG_NAME

웹페이지에서 요소 가져오기

- CSS Selector로 요소 찾기

웹드라이버.fine_element(요소지정방법, CSS Selector)

→ 첫번째 결과 반환

웹드라이버.fine_elements(요소지정방법, CSS Selector)

→ 모든 결과 리스트로 반환

웹페이지의 요소 제어하기

- 입력창에 검색어 입력하기


```
# 입력창 요소 클릭하기
search.click()
```

요소객체.click()


```
# 검색어(텍스트) 입력하기
search.send_keys("인공지능")
```

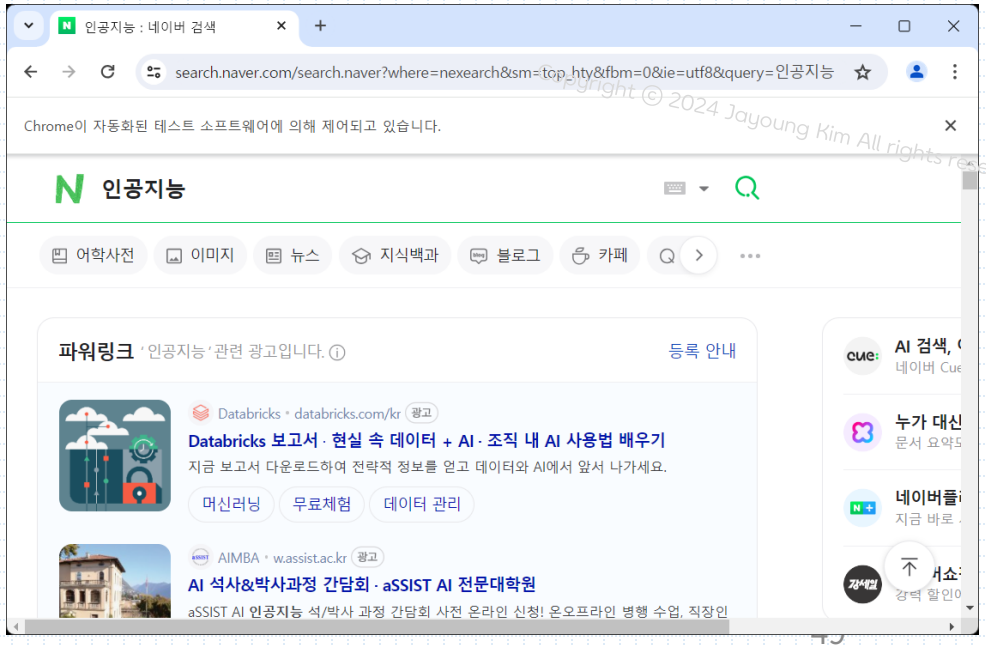
요소객체.send_keys(“텍스트”)

웹페이지의 요소 제어하기

▪ 엔터키 입력하기

요소.send_keys(특수키)


```
# 엔터키 입력하기
from selenium.webdriver.common.keys import Keys
search.send_keys(Keys.ENTER)
```



▪ 특수키 사용

- Keys.**ENTER**
- Keys.TAB
- Keys.SHIFT
- Keys.CONTROL
- Keys.ALT
- Keys.SPACE
- Keys.BACKSPACE
- Keys.DELETE
- Keys.LEFT
- Keys.RIGHT
- Keys.UP
- Keys.DOWN
- Keys.F1~Keys.F12

등…

※ dir(Keys)

웹페이지의 요소 제어하기

- 순차적으로 키 입력하기


```
# 순차적으로 키 입력하기
search.send_keys("인공지능", Keys.ENTER)
```

웹페이지의 요소 제어하기

- 브라우저 닫기


```
# 브라우저 닫기
driver.quit()
```

네이버증권 뉴스 크롤링

https://finance.naver.com/news/mainnews.naver

크롤링

- 제목
- 상세페이지링크
- 내용
- 언론사
- 날짜

웹페이지 접근

- Selenium으로 웹페이지 띄우기


```
url = "https://finance.naver.com/news/mainnews.naver"

from selenium.webdriver import Chrome
driver = Chrome()
driver.maximize_window()
driver.get(url)
```

아티클 제목만 추출하기

아티클 제목만 추출

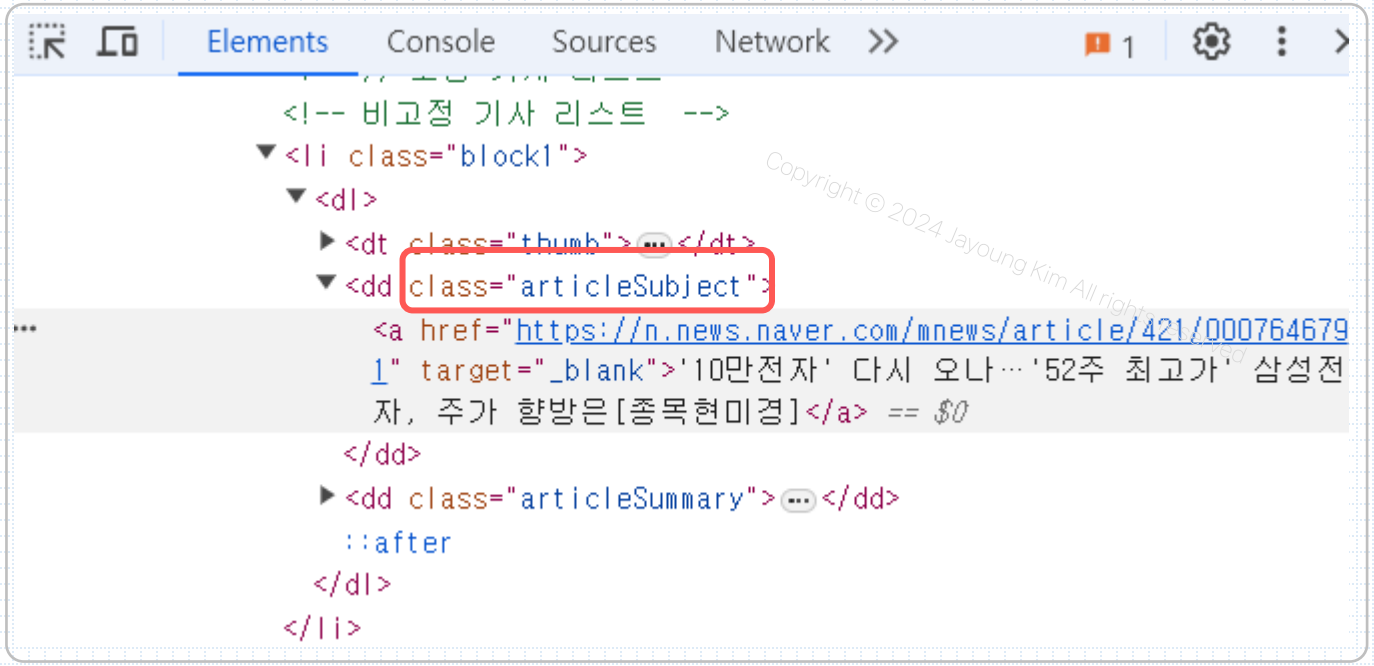
- 선택자 만들기

❶ 추출하고자 하는 요소에서 마우스 우클릭 > 검사

아티클 제목만 추출

- 선택자 만들기

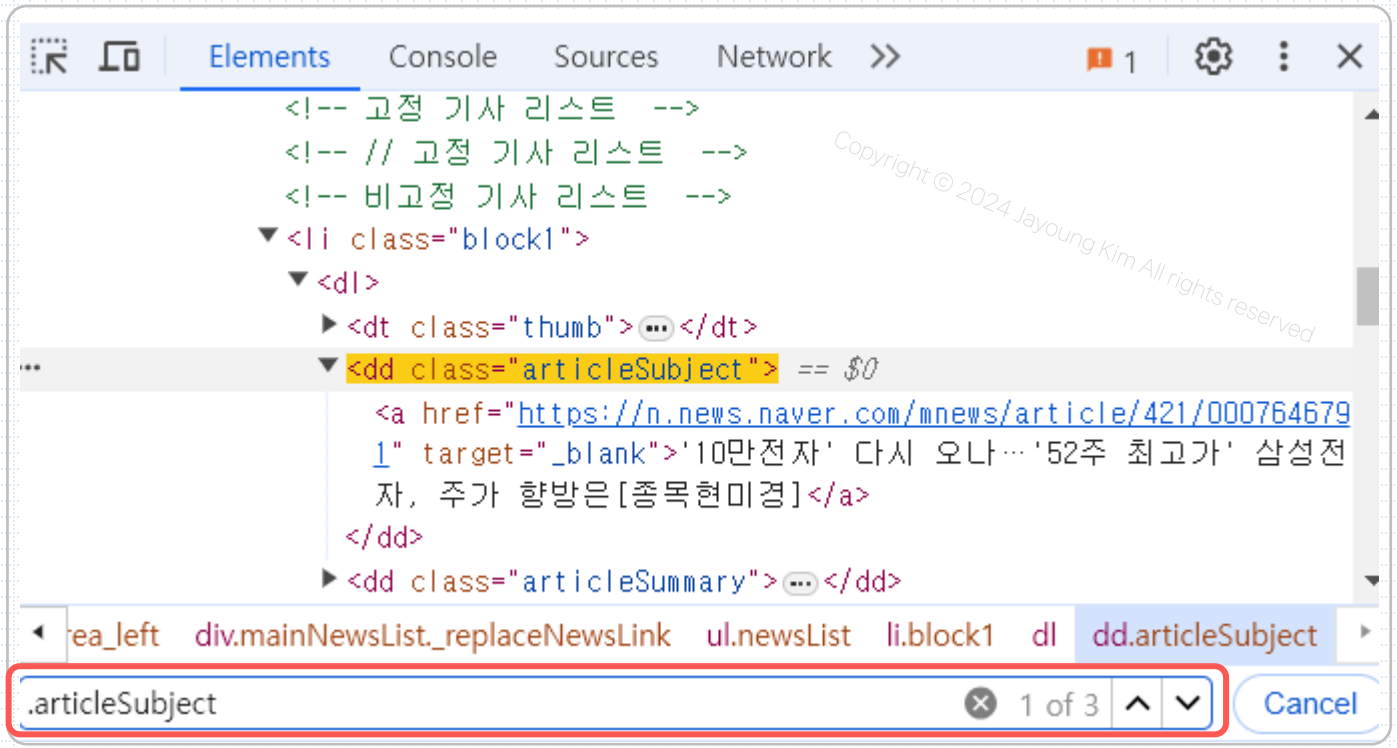
❷ CSS Selector 후보 찾기



아티클 제목만 추출

- 선택자 만들기

❸ Ctrl+F를 통하여 CSS Selector 확인



아티클 제목만 추출

- 제목 요소 추출


```
# 제목 요소 선택
from selenium.webdriver.common.by import By
subjects = driver.find_elements(By.CSS_SELECTOR, ".articleSubject")
subjects
```



```
[<selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.763")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.774")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.785")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.796")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.807")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.818")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.829")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.840")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.851")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.862")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.873")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.884")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.895")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.906")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.917")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.928")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.939")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.950")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.961")>,
 <selenium.webdriver.remote.webelement.WebElement (session="db0a325507bdb6eb8a623bce94ba2923", element="f.006C88143E6269123BEBF60CB1C18717.d.8656C3E63D32897B50EF015DFBD314AE.e.972")>]
```

아티클 제목만 추출

- 제목 요소의 텍스트 추출


```
# 제목 요소의 텍스트 추출
for subject in subjects:
    print(subject.text)
```

```
챗GPT·엔비디아·비트코인도 속절없이…AI 돌풍 일으킨 中 딥시크
"중국 AI 챗봇에 휘청"…나스닥 선물 4% 이상 급락 '충격'
美증시에 '트럼프'를 달고 있는 이 회사의 미래는? [홍키자의 빅테크]
로봇 기업 품는 삼성·LG…개발 경쟁 '활활'
임시 공휴일 '내수 효과'…도심 거리, 자영업자 만나봤더니
서학개미들, 설 연휴에도 '두근두근'…"이 종목 심상치 않다" [인터뷰+]
명절에 엔비디아 '급락' 날벼락…배경엔 챗GPT 제치고 앱스토어 1위한 中 '딥시크'
올해 '삼천피' 재도전 … 부진했던 반도체株 하반기에 날갯짓
빅테크가 주도하는 美증시, 상승 흐름 지속 … 1분기 변동성은 클 것
부양책 소용없나, 中 증시 춘절 장기 휴장 앞두고 하락
두산스코다파워, 내달 체코 프라하 증시 상장…"성장동력 확보"
딥시크 쇼크에 미국 주식 AI버블 붕괴하나.. "AI의 스푸트니크 모먼트"
KT '이통 대장주' 굳힐 세 가지 이유
함영주 하나금융 회장 연임…회추위 "검증된 리더십 절실"(종합2보)
'저PBR' 수혜 기대주였던 제주은행[급등주 지금은]
하나금융 차기 회장에 함영주 내정…회추위 연임 추천(종합)
"글로벌 금융위기 때와 판박이…국내증시 폭탄 세일 중" [인터뷰+]
불황 직격탄 맞은 식품주…해외 매출이 희비 가른다
'CES 효과' 끝나도 주가 달린다…돌격하는 개미 군단들
10억 간다고?…트럼프 '친 가상자산' 행보에 '코인 불장 올까'
```

아티클의 제목, 요약, 링크, 날짜, 언론사 추출하기

아티클에서 여러 요소 추출

- 아티클 영역 선택자 찾기

아티클에서 여러 요소 추출

- 아티클 요소 선택


```
# 아티클 영역 요소 선택
from selenium.webdriver.common.by import By
subjects = driver.find_elements(By.CSS_SELECTOR, ".articleSubject")
```

아티클에서 여러 요소 추출

- 아티클 요소에서 세부 요소 추출


```
# 아티클 영역 요소 선택
from selenium.webdriver.common.by import By
articles = driver.find_elements(By.CSS_SELECTOR, ".block1")

# 각 아티클 영역 요소에서 세부 요소 추출
for article in articles:
    subject = article.find_element(By.CSS_SELECTOR, ".articleSubject").text
    summary = article.find_element(By.CSS_SELECTOR, ".articleSummary").text
    press   = article.find_element(By.CSS_SELECTOR, ".press").text
    wdate   = article.find_element(By.CSS_SELECTOR, ".wdate").text
    print(subject)
```

아티클에서 여러 요소 추출

- 아티클 요소에서 link 추출을 위한 선택자 만들기

.articleSubject>a

.articleSubject의 자식인 a태그

```
<dd class="articleSubject">
  <a href="https://n.news.naver.com/mnews/article/277/0005539135" target="_blank">챗GPT · 엔비디아 · 비트
  코인도 속절없이…AI 돌풍 일으킨 中 딥시크</a> == $0
</dd>
```

아티클에서 여러 요소 추출

- 아티클 요소에서 세부 요소 추출


```
# 아티클 영역 요소 선택
from selenium.webdriver.common.by import By
articles = driver.find_elements(By.CSS_SELECTOR, ".block1")

# 아티클 요소에서 세부 요소 추출
for article in articles:
    subject = article.find_element(By.CSS_SELECTOR, ".articleSubject").text
    summary = article.find_element(By.CSS_SELECTOR, ".articleSummary").text
    press   = article.find_element(By.CSS_SELECTOR, ".press").text
    wdate   = article.find_element(By.CSS_SELECTOR, ".wdate").text
    link    = article.find_element(By.CSS_SELECTOR, ".articleSubject>a").get_attribute("href")
    print(link)
```


속성 추출

아티클에서 여러 요소 추출

- 데이터프레임 생성을 위해 list에 append


```
# 아티클 영역 요소 선택
from selenium.webdriver.common.by import By
articles = driver.find_elements(By.CSS_SELECTOR, ".block1")

# 아티클 요소에서 세부 요소 추출
article_list = []
for article in articles:
    subject = article.find_element(By.CSS_SELECTOR, ".articleSubject").text
    summary = article.find_element(By.CSS_SELECTOR, ".articleSummary").text
    press   = article.find_element(By.CSS_SELECTOR, ".press").text
    wdate   = article.find_element(By.CSS_SELECTOR, ".wdate").text
    link    = article.find_element(By.CSS_SELECTOR, ".articleSubject>a").get_attribute("href")
    article_list.append([subject, summary, press, wdate, link])
```

코드 합치고 delay 주기

```
url = "https://finance.naver.com/news/mainnews.naver"

from selenium.webdriver import Chrome
import time

driver = Chrome()
driver.maximize_window()
driver.get(url)
time.sleep(1) # 페이지가 로딩되는 시간 감안하여 delay

# 아티클 영역 요소 선택
from selenium.webdriver.common.by import By
articles = driver.find_elements(By.CSS_SELECTOR, ".block1")

# 아티클 요소에서 세부 요소 추출
article_list = []
for article in articles:
    subject = article.find_element(By.CSS_SELECTOR, ".articleSubject").text
    summary = article.find_element(By.CSS_SELECTOR, ".articleSummary").text
    press   = article.find_element(By.CSS_SELECTOR, ".press").text
    wdate   = article.find_element(By.CSS_SELECTOR, ".wdate").text
    link    = article.find_element(By.CSS_SELECTOR, ".articleSubject>a").get_attribute("href")
    article_list.append([subject, summary, press, wdate, link])
```

파일로 다운로드

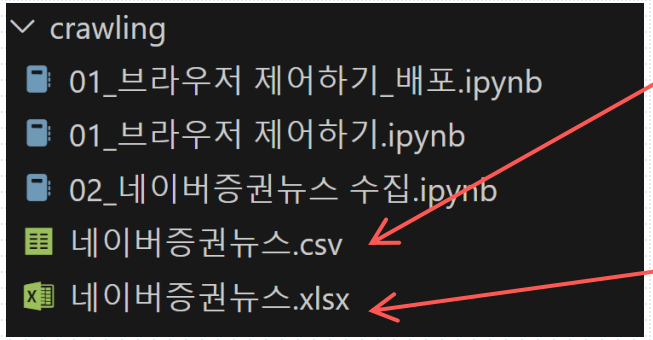
- 데이터프레임 생성


```
import pandas as pd
df = pd.DataFrame(article_list,
                  columns = ['subject','summary','press','wdate','link'])
df.head(3)
```


	subject	summary	press	wdate	link
0	챗GPT·엔비디아·비트코인도 속절없이... AI 돌풍 일으킨 中 딥시크	중국 인공지능(AI) 스타트업 딥시크(DeepSeek)의 AI 모델이 챗GPT를 제...	아시아경제	2025-01-27 21:10:14	https://n.news.naver.com/mnews/article/277/000...
1	"중국 AI 챗봇에 휘청"...나스닥 선물 4% 이상 급락 '충격'	중국 인공지능(AI) 업체 딥시크 충격으로 미국 자본시장이 흔들렸다. 지수 선물이 ...	한국경제	2025-01-27 20:54:14	https://n.news.naver.com/mnews/article/015/000...
2	美증시에 '트럼프'를 달고 있는 이 회사의 미래는? [홍키자의 빅테크]	1월 20일 도널드 트럼프 2기 행정부 시대가 본격 시작됐습니다. 지난해 11월 트...	매일경제	2025-01-27 20:01:06	https://n.news.naver.com/mnews/article/009/000...

파일로 다운로드

- 파일로 다운로드




```
df.to_csv('네이버증권뉴스.csv', encoding='utf-8', index=False)

import openpyxl
df.to_excel('네이버증권뉴스.xlsx', index=False)
```

특정 날짜의 기사 수집

https://finance.naver.com/news/mainnews.naver**?date=2025-01-23**

특정 날짜의 기사 수집

- 뉴스기사 날짜를 입력하여 해당 날짜의 뉴스기사 수집


```
import datetime
date = input("검색할 날짜(yyyymmdd) : ")
if date=="": date = datetime.datetime.now().strftime('%Y-%m-%d')
else: date = f'{date[:4]}-{date[4:6]}-{date[6:]}'
url = "https://finance.naver.com/news/mainnews.naver?date="+date

from selenium.webdriver import Chrome
import time

driver = Chrome()
driver.maximize_window()
driver.get(url)
```

특정 날짜의 기사 수집

- 파일명 관리


```
filename = date.replace('-','')
df.to_csv(f'네이버증권뉴스_{filename}.csv', encoding='utf-8', index=False)

import openpyxl
df.to_excel(f'네이버증권뉴스_{filename}.xlsx', index=False)
```

여러 페이지 크롤링

- 페이지 이동에 따른 url 구조 확인하기


```
https://finance.naver.com/news/mainnews.naver?date=2024-07-05&page=1
https://finance.naver.com/news/mainnews.naver?date=2024-07-05&page=2
https://finance.naver.com/news/mainnews.naver?date=2024-07-05&page=3
```

...

여러 페이지 크롤링

- 모든 페이지 크롤링

페이징 UI 구조 확인

1 2 3 4 5 6 맨뒤 ▸▸

◂◂ 맨앞 1 2 3 4 5 6

…

→ 마지막 페이지에는 [맨뒤] 없음


```
▼<td class="pgRR">
  ▼<a href="/news/mainnews.naver?date=2024-07-05&page=
   6"> == $0
     "맨뒤 "
     <img src="https://ssl.pstatic.net/static/n/cmn/bu_pgarRR.
     gif" width="8" height="5" alt border="0">
   </a>
 </td>
```

→ .pgRR을 이용하여 요소를 찾을 수 없을 때까지 반복

여러 페이지 크롤링

모든 페이지 크롤링


```
-- 생략 --
article_list = []
page = 1
while True:
    page_url = f'{url}&page={page}'
    driver.get(page_url)
    time.sleep(2) # 페이지가 로딩되는 시간 감안하여 delay

    # 아티클 영역 요소 선택
    from selenium.webdriver.common.by import By
    articles = driver.find_elements(By.CSS_SELECTOR, ".block1")

    # 아티클 요소에서 세부 요소 추출
    for article in articles:
        subject = article.find_element(By.CSS_SELECTOR, ".articleSubject").text
        summary = article.find_element(By.CSS_SELECTOR, ".articleSummary").text
        press   = article.find_element(By.CSS_SELECTOR, ".press").text
        wdate   = article.find_element(By.CSS_SELECTOR, ".wdate").text
        link    = article.find_element(By.CSS_SELECTOR, ".articleSubject>a").get_attribute("href")
        article_list.append([subject, summary, press, wdate, link])

    # 다음 페이지 버튼 확인
    try:
        driver.find_element(By.CSS_SELECTOR, ".pgRR")
        page += 1
    except:
        break
driver.quit()

import pandas as pd
df = pd.DataFrame(article_list,
                  columns = ['subject','summary','press','wdate','link'])

filename = date.replace('-','')
df.to_csv(f'네이버증권뉴스_{filename}.csv', encoding='utf-8', index=False)
import openpyxl
df.to_excel(f'네이버증권뉴스_{filename}.xlsx', index=False)
```

수고하셨습니다.