분석용 데이터
획득 전략

대량 데이터 확보 기법

학습 목표

학습 목표

- 특정 메뉴를 선택하는 다양한 방법을 활용할 수 있다.
- 원하는 항목을 선택하여 추출할 수 있다.
- 페이지를 변경하는 다양한 방법을 활용할 수 있다.

학습 내용

- 메뉴 선택과 상세내역 출력 및 페이지 변경하기 – 이론
- 메뉴 선택과 상세내역 출력 및 페이지 변경하기 - 실습

메뉴 선택하기 - 이론

1) 특정 메뉴가 텍스트일 경우

선택하는 메뉴가 텍스트로 되어 있고
해당 메뉴에 하이퍼링크가 존재할 경우
LINK_TEXT() 함수를 활용하여 클릭 가능

LINK_TEXT()

하이퍼링크가 걸려 있을 때 그 글자를 클릭하라는 명령어

메뉴 선택하기 - 이론

2) 특정 메뉴가 텍스트가 아닐 경우

선택하고 싶은 메뉴가 텍스트가 아닐 경우에는
LINK_TEXT() 함수를 사용할 수 없다.

※ 이럴 경우 XPATH값을 사용하면 된다.

XPATH값 찾기

오른쪽 HTML에서 마우스 우측 버튼 클릭

➔ XPATH 클릭

➔ Copy XPATH 선택 후 메모장이나 소스코드에 붙여넣기

XPATH(XML Path Language)란,

W3C의 표준으로 확장 생성 언어 문서의 구조를 통해
경로 위에 지정한 구문을 사용하여 항목을 배치하고
처리하는 방법을 기술하는 언어

- XML 지시자 언어(XPointer)에 쓰이는 언어
- XSL 변환(XSLT)과 지시자 언어(Xpointer)에 쓰이는 언어
- XPATH는 XML 문서의 노드를 정의하기 위하여 경로식을 사용하며, 수학 함수와 기타 확장 가능한 표현들이 있음

정리

- ❖ 메뉴가 글자로 되어 있으면, LINK_TEXT 함수 사용
- ❖ 메뉴가 그림이나 다른 모양으로 되어 있으면, XPATH 사용

메뉴 선택하기 - 실습

- 입력한 정보 저장

```
# riss.kr 에서 특정 키워드로 논문 / 학술 자료 검색하기

#Step 1. 필요한 모듈을 로딩합니다
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.common.keys import Keys
from selenium.webdriver.chrome.service import Service
import time

#Step 2. 사용자에게 검색 관련 정보들을 입력 받습니다.
print("=" *100)
print(" 이 크롤러는 RISS 사이트의 논문 및 학술자료 수집용 웹크롤러입니다.")
print("=" *100)
query_txt = input('1.수집할 자료의 키워드는 무엇입니까? : ')
```

- 경로를 정확히 입력

```
#Step 3. 크롬 드라이버 설정 및 웹 페이지 열기
s = Service("c:/py_temp/chromedriver.exe")
driver = webdriver.Chrome(service=s)

url = 'https://www.riss.kr/'
driver.get(url)
time.sleep(5)
driver.maximize_window()

#Step 4. 자동으로 검색어 입력 후 조회하기
element = driver.find_element(By.ID,'query')
driver.find_element(By.ID,'query').click( )
element.send_keys(query_txt)
element.send_keys("\n")
```

메뉴 선택하기 - 실습

- 주소 앞에 프로토콜 반드시 입력

```
16 #Step 3. 크롬 드라이버 설정 및 웹 페이지 열기
17 s = Service("c:/py_temp/chromedriver.exe")
18 driver = webdriver.Chrome(service=s)
19
20 url = 'https://www.riss.kr/'
21 driver.get(url)
22 time.sleep(5)
23 driver.maximize_window()
24
25 #Step 4. 자동으로 검색어 입력 후 조회하기
26 element = driver.find_element(By.ID,'query')
27 driver.find_element(By.ID,'query').click( )
28 element.send_keys(query_txt)
29 element.send_keys("\n")
```

주의사항

- ❖ 느린 인터넷 상에서 접속 중 명령시 에러 발생
- ❖ 엘리먼트 이름은 사이트마다 다를 수 있기에 개발자 도구를 통해 탐색 후 사용

메뉴 선택하기 - 실습

- 대소문자 정확하게 입력

```
#Step 3. 크롬 드라이버 설정 및 웹 페이지 열기
s = Service("c:/py_temp/chromedriver.exe")
driver = webdriver.Chrome(service=s)

url = 'https://www.riss.kr/'
driver.get(url)
time.sleep(5)
driver.maximize_window()

#Step 4. 자동으로 검색어 입력 후 조회하기
element = driver.find_element(By.NAME,'query')
driver.find_element(By.ID,'query').click( )
element.send_keys(query_txt)
element.send_keys("\n")
```

주의사항

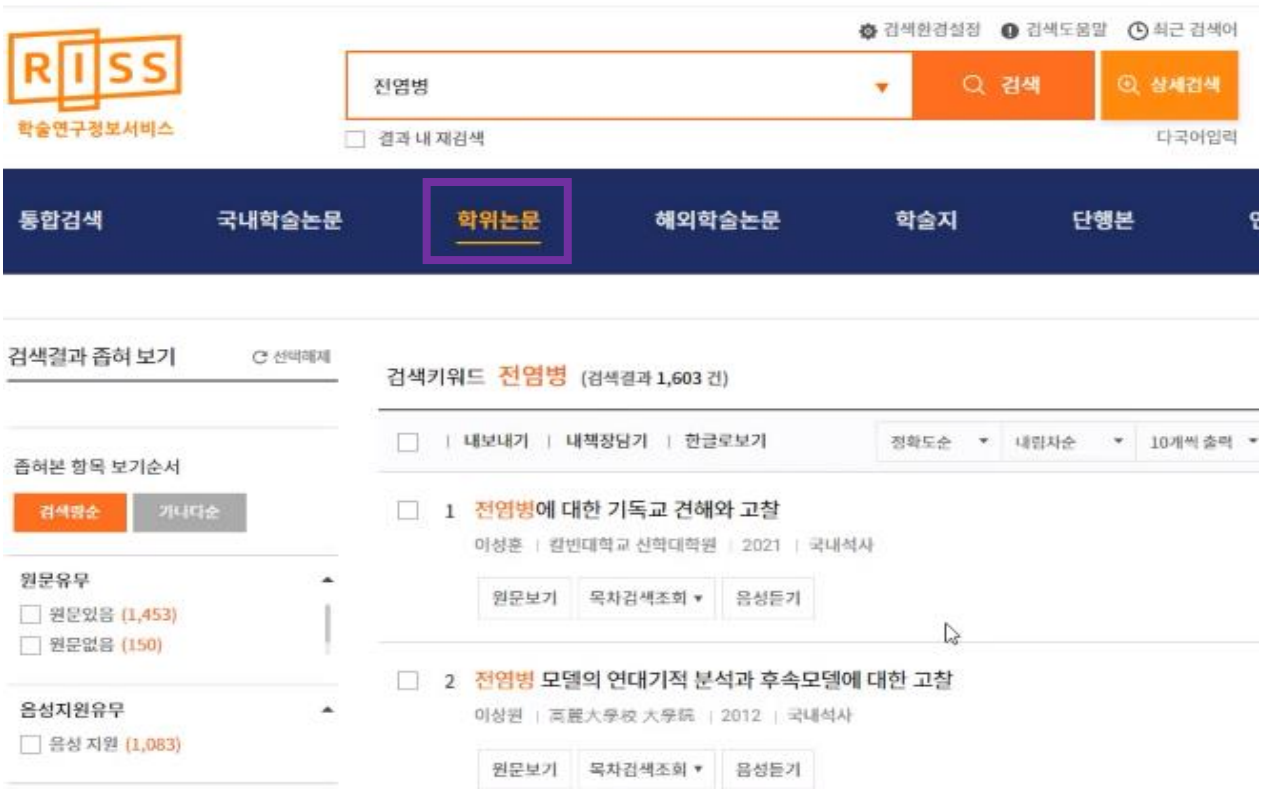
ID도 Name도 없는 사이트일 경우, XPATH값을 이용

상세 항목 추출하기

```
27 driver.find_element(By.ID,'query').click( )
28 element.send_keys(query_txt)
29 element.send_keys("\n")
```

```
이 크롤러는 RISS 사이트의 논문 및 학술자료 수집용 웹크롤러입니다.
1.수집할 자료의 키워드는 무엇입니까? : 전염병
```

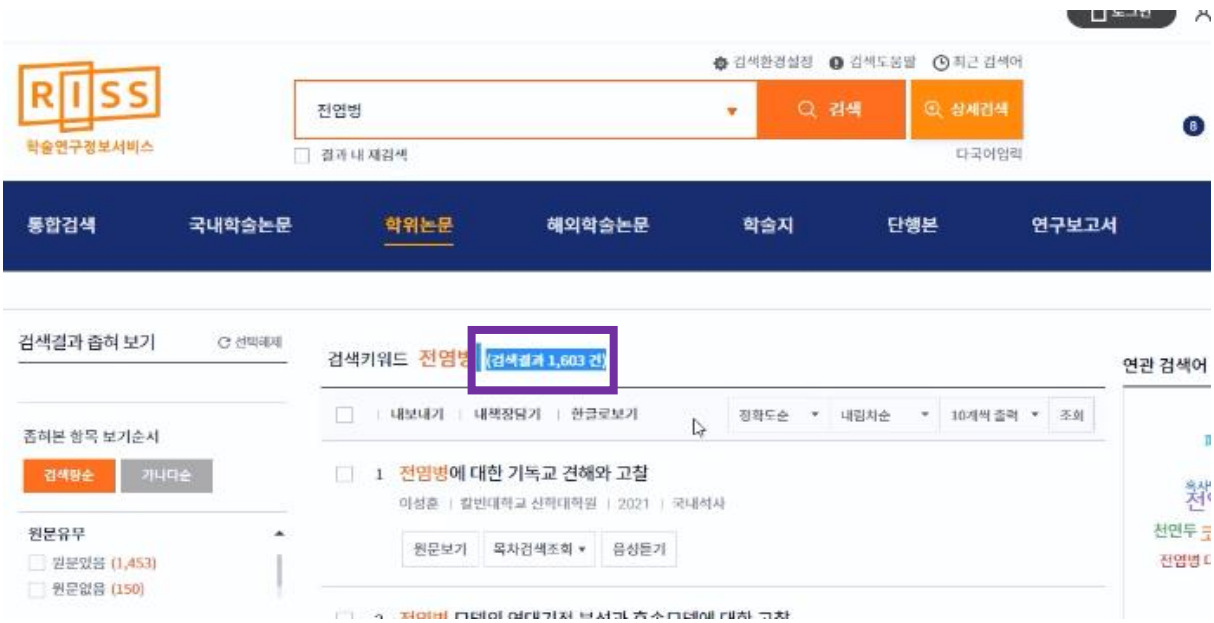
```
1 # Step 6.학위 논문 메뉴 선택하기
2 # 메뉴선택방법 1 - LINK_TEXT( ) 사용하기
3 driver.find_element(By.LINK_TEXT,'학위논문').click()
4 time.sleep(2)
5
6 # 메뉴 선택방법 2 - XPATH 사용하기
7 #driver.find_element(By.XPATH , '//*[@id="tabMenu"]/ul/li/div/ul/li[3]/a/span').click()
8 #time.sleep(2)
```



상세 항목 추출하기

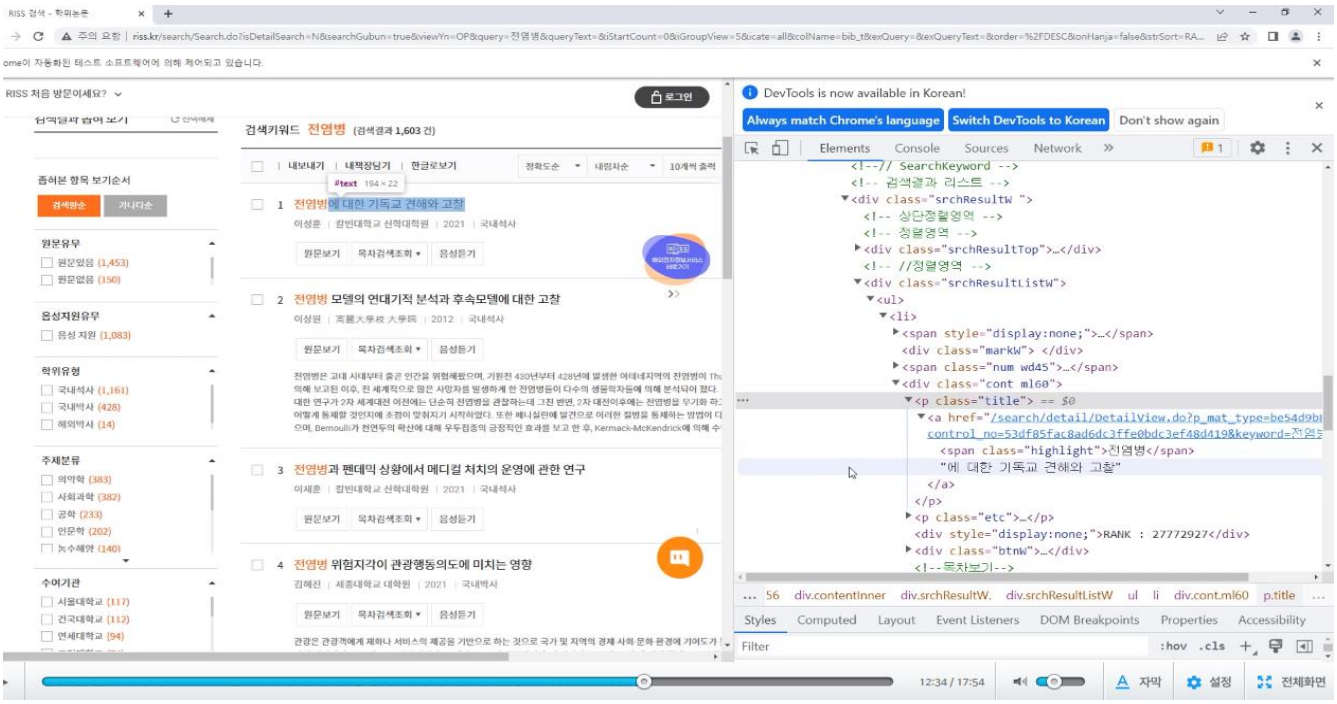
- 총 검색 건수 추출하기

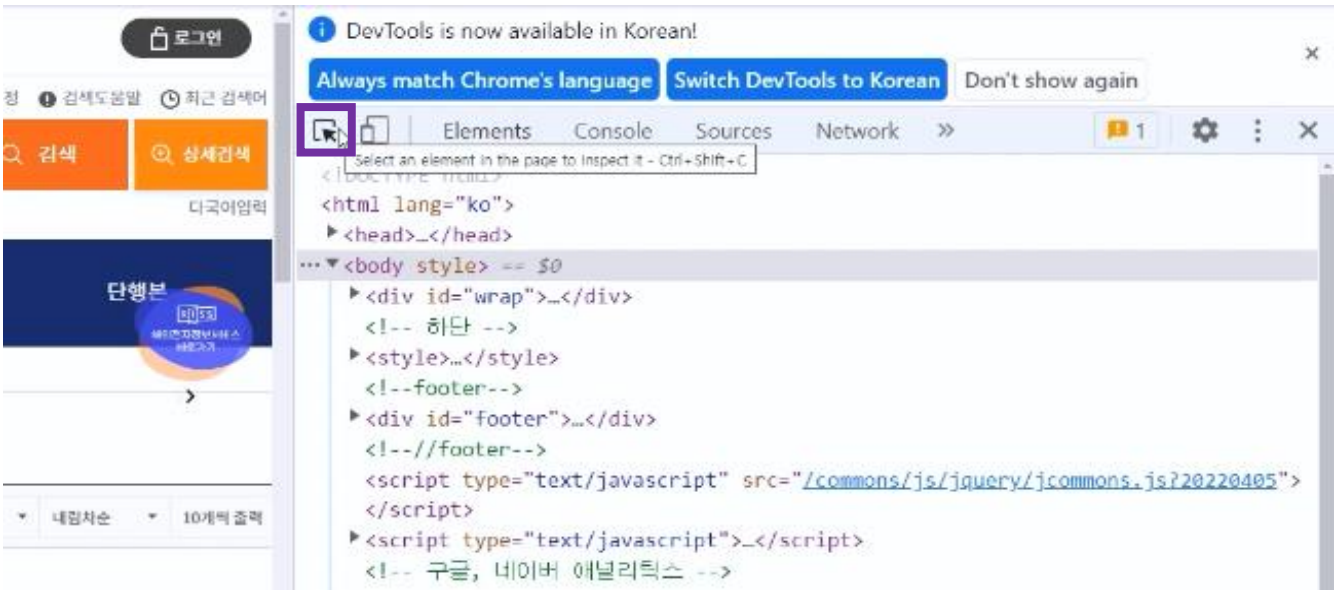
```
#Step 7.Beautiful Soup 로 본문 내용만 추출하기
from bs4 import BeautifulSoup
html_1 = driver.page_source
soup_1 = BeautifulSoup(html_1, 'html.parser')

# 총 검색 건수를 보여주고 수집할 건수 입력받기
import math
total_cnt = soup_1.find('div','searchBox pd').find('span','num').get_text()
print('검색하신 키워드 %s (으)로 총 %s 건의 학위논문이 검색되었습니다' %(query_
collect_cnt = int(input('이 중에서 몇 건을 수집하시겠습니까?: '))
collect_page_cnt = math.ceil(collect_cnt / 10)
print('%s 건의 데이터를 수집하기 위해 %s 페이지의 게시물을 조회합니다.' %(colle
print('=' *80)

#Step 8. 각 항목별로 데이터를 추출하여 리스트에 저장하기
no2 = [ ]          #번호 저장
title2 = [ ]       #논문제목 저장
author2 = [ ]      #논문저자 저장
company2 = [ ]          #소속기관 저장
date2 = [ ]        #발표년도 저장
hak2 = [ ]         #학위정보 저장
no = 1

for a in range(1, collect_page_cnt + 1) :

```



상세 항목 추출하기

- 개발자 도구 F12





메뉴 선택과 상세내역 출력 및 페이지 변경하기 - 실습

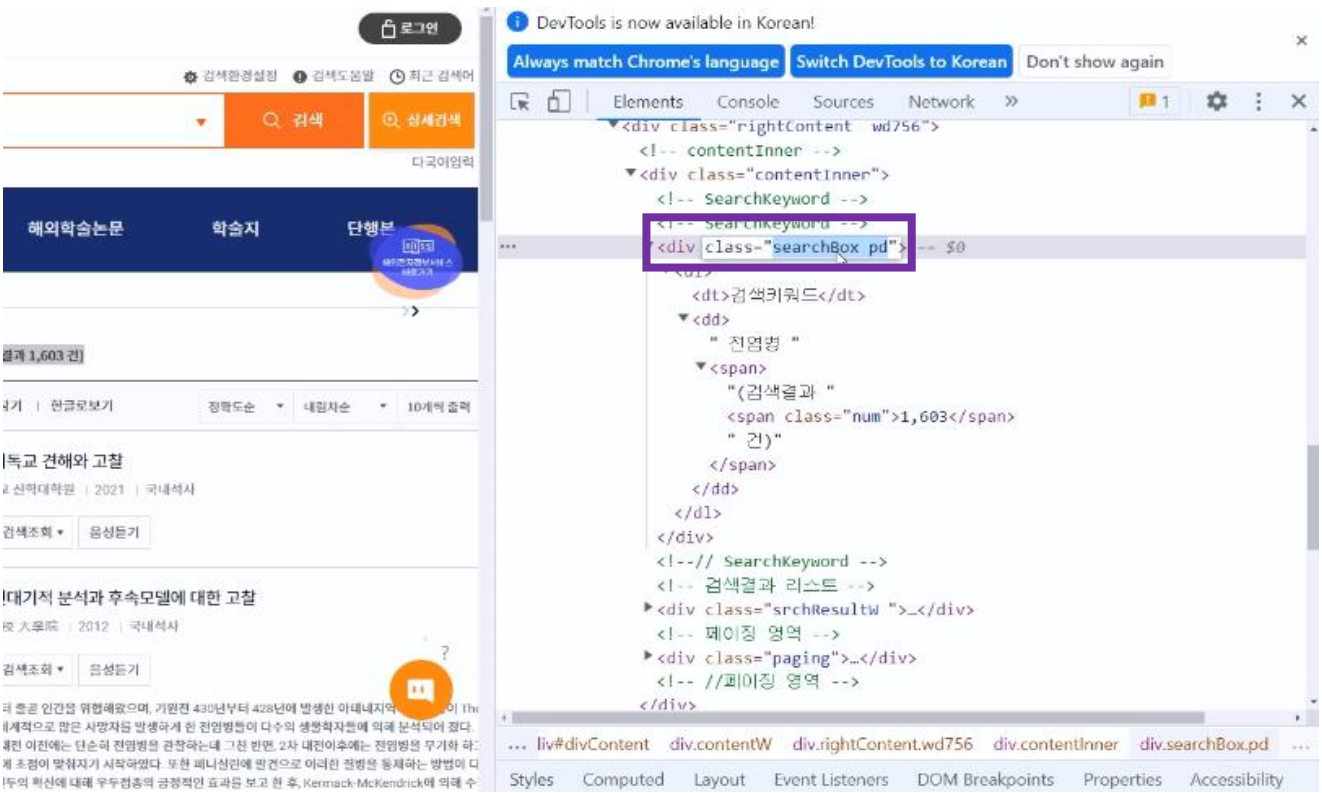
상세 항목 추출하기

```
#Step 7.Beautiful Soup 로 본문 내용만 추출하기
from bs4 import BeautifulSoup
html_1 = driver.page_source
soup_1 = BeautifulSoup(html_1, 'html.parser')

# 총 검색 건수를 보여주고 수집할 건수 입력받기
import math
total_cnt = soup_1.find('div', 'searchBox pd').find('span','num').get_text()
print('검색하신 키워드 %s (으)로 총 %s 건의 학위논문이 검색되었습니다' %(query_
collect_cnt = int(input('이 중에서 몇 건을 수집하시겠습니까?: '))
collect_page_cnt = math.ceil(collect_cnt / 10)
print('%s 건의 데이터를 수집하기 위해 %s 페이지의 게시물을 조회합니다.' %(colle
print('=' *80)
```

주의사항

searchBox pd : 대소문자, 공백 개수 등 정확히 입력해야 함

- 더블 클릭 → ctrl + c 눌러 복사 + 붙여넣기

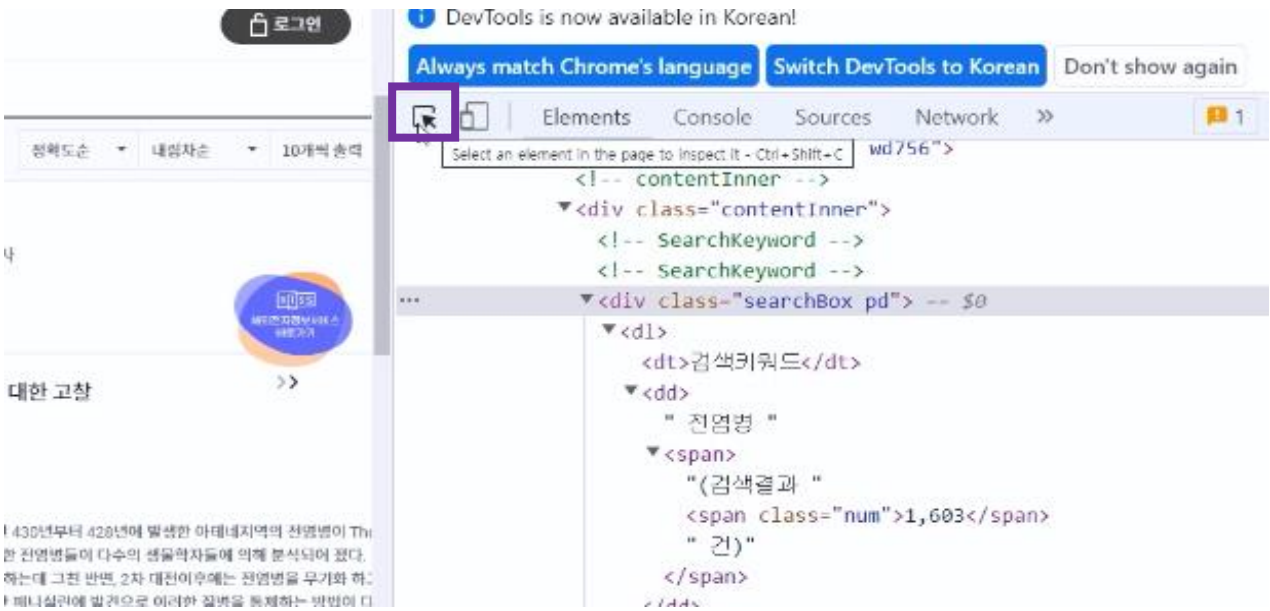


상세 항목 추출하기

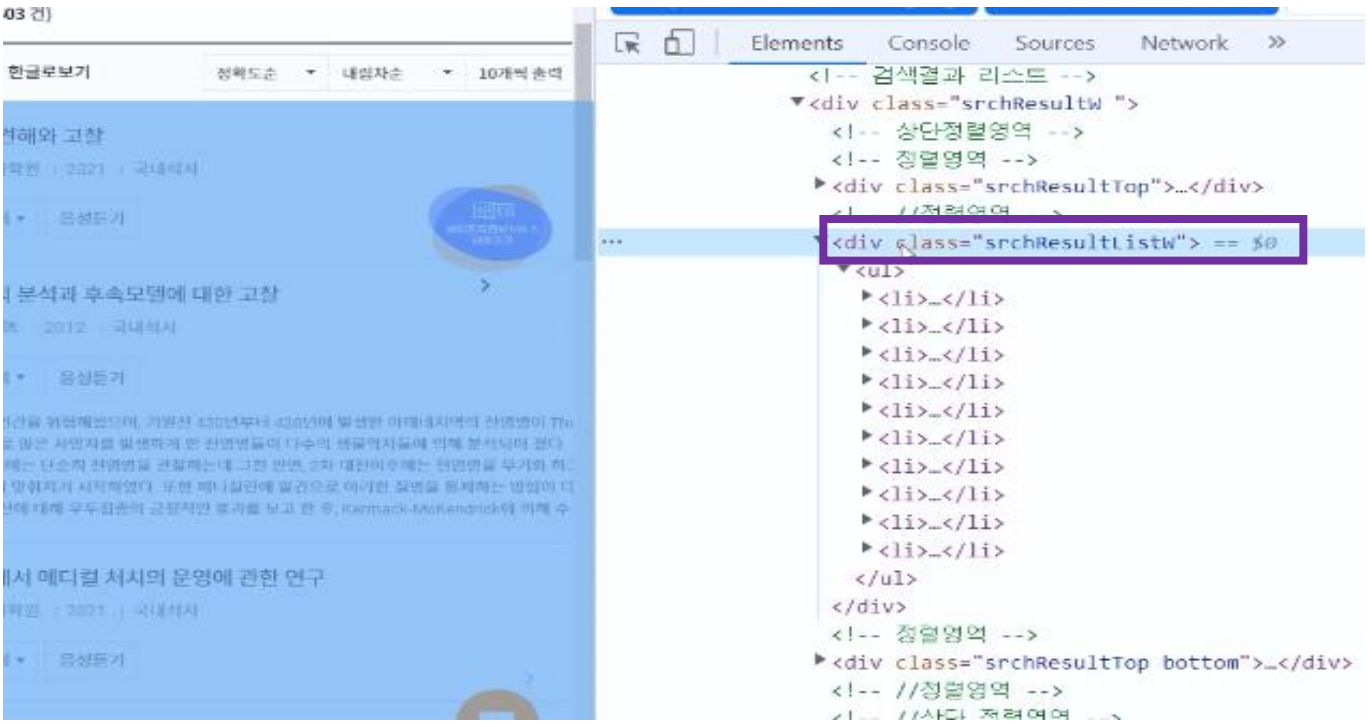
```
#Step 7.Beautiful Soup 로 본문 내용만 추출하기
from bs4 import BeautifulSoup
html_1 = driver.page_source
soup_1 = BeautifulSoup(html_1, 'html.parser')

# 총 검색 건수를 보여주고 수집할 건수 입력받기
import math
total_cnt = soup_1.find('div','searchBox pd').find('span','num').get_text()
print('검색하신 키워드 %s (으)로 총 %s 건의 학위논문이 검색되었습니다' %(query_txt,total_cnt))
collect_cnt = int(input('이 중에서 몇 건을 수집하시겠습니까?: '))
collect_page_cnt = math.ceil(collect_cnt / 10)
print('=' *80)

#Step 8. 각 항목별로 대이터를 추출하여 리스트에 저장하기
no2 = [ ]          #번호 저장
title2 = [ ]       #논문제목 저장
author2 = [ ]      #논문저자 저장
company2 = [ ]         #소속기관 저장
date2 = [ ]        #발표년도 저장
hak2 = [ ]         #학위정보 저장
no = 1

for a in range(1, collect_page_cnt + 1) :

    html_2 = driver.page_source
    soup_2 = BeautifulSoup(html_2, 'html.parser')

    content_2 = soup_2.find('div','srchResultListW').find_all('li')
```

ceil()

주어진 숫자보다 크고 가장 가까운 정수를 뽑아내는 함수



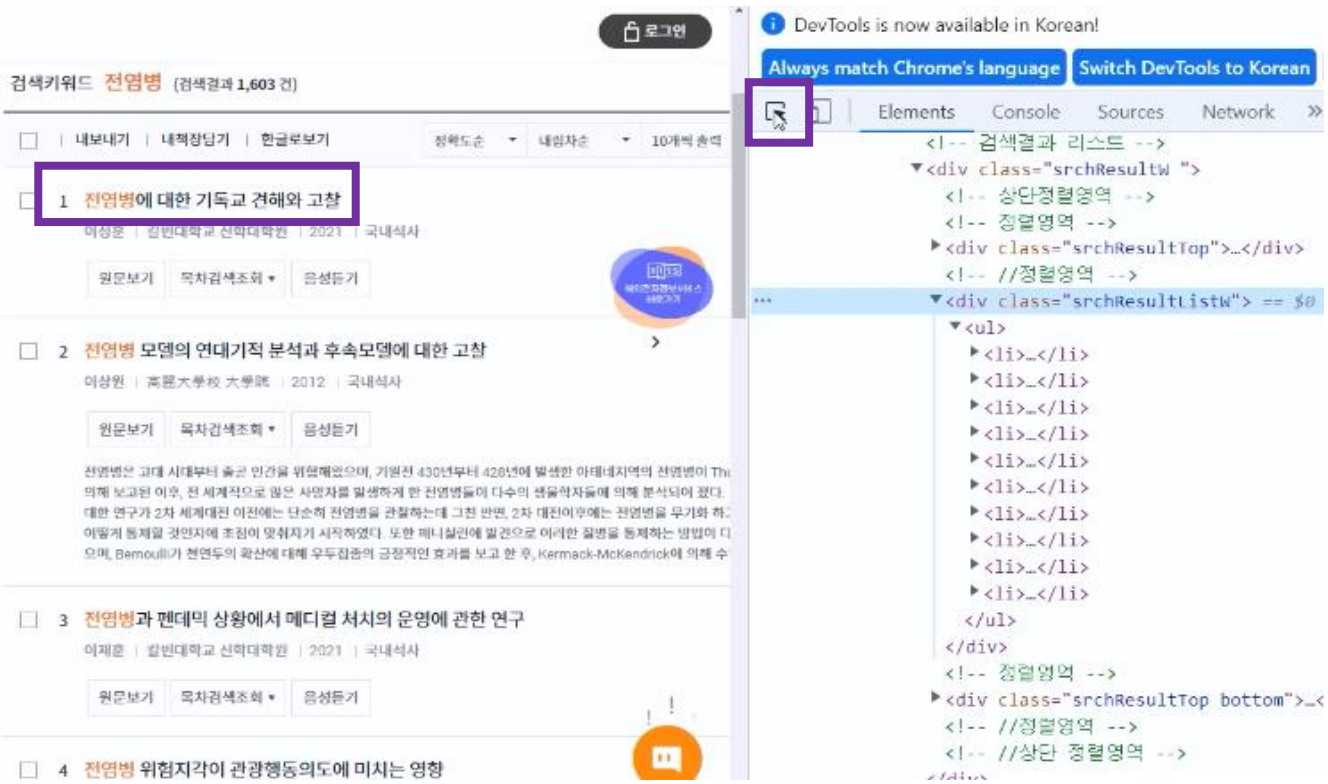
상세 항목 추출하기



상세 항목 추출하기

- 논문 제목 추출하기

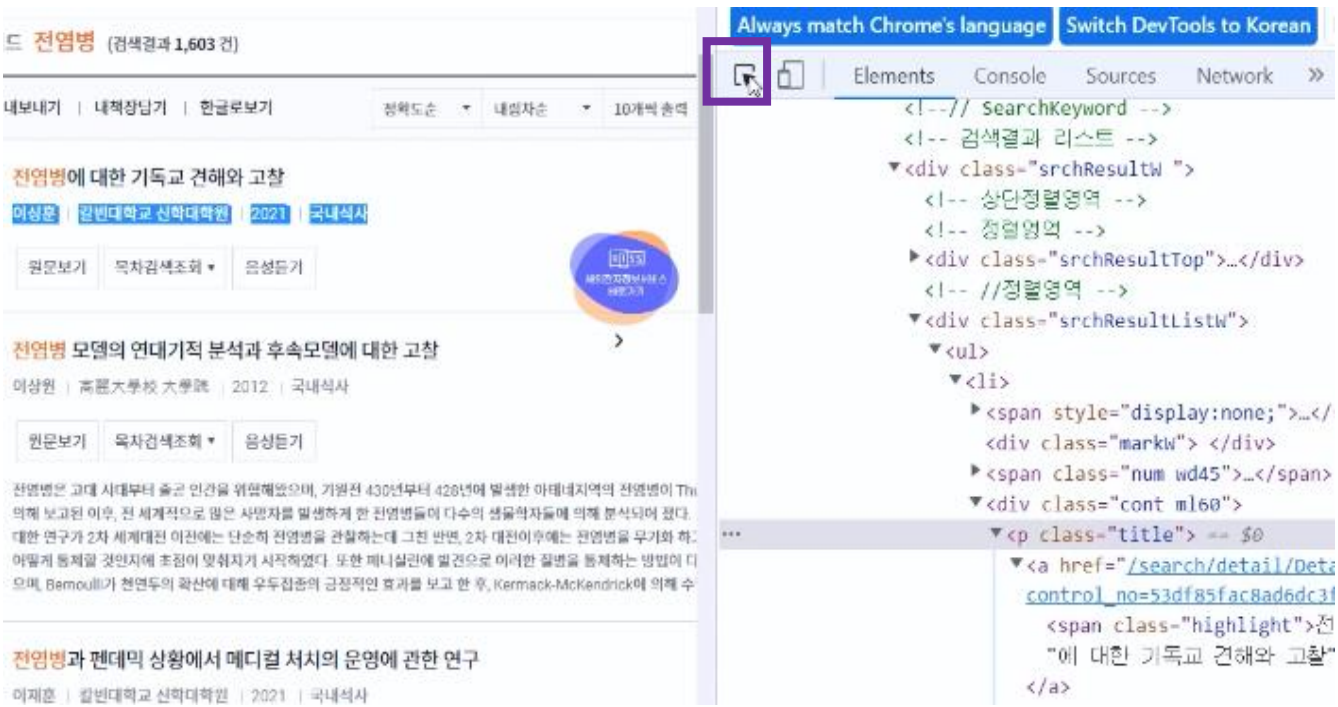
```
for a in range(1, collect_page_cnt + 1) :

    html_2 = driver.page_source
    soup_2 = BeautifulSoup(html_2, 'html.parser')

    content_2 = soup_2.find('div','srchResultListW').find_all('li')

    for i in content_2 :
        #1. 논문제목 있을 경우만
        try :
            title = i.find('div','cont').find('p','title').get_text()
        except :
            continue
        else :
            print('1.번호:',no)
            no2.append(no)

            print('2.논문제목:',title)
            title2.append(title)

            # 저자 정보
            all_data = i.find('p','etc').find_all('span')
            try :
                author=all_data[0].get_text().strip()
            except :
                author = '저자 정보가 없습니다'
                print("3.저자정보 : %s" %author)
```



상세 항목 추출하기

- 제목이 없는 경우 예외 처리

```
for a in range(1, collect_page_cnt + 1) :

    html_2 = driver.page_source
    soup_2 = BeautifulSoup(html_2, 'html.parser')

    content_2 = soup_2.find('div','srchResultListW').find_all('li')

    for i in content_2 :
        #1. 논문제목 있을 경우만
        try :
            title = i.find('div','cont').find('p','title').get_text()
        except :
            continue
        else :
            print('1.번호:',no)
            no2.append(no)

            print('2.논문제목:',title)
            title2.append(title)

            # 저자 정보
            all_data = i.find('p','etc').find_all('span')
            try :
                author=all_data[0].get_text().strip()
            except :
                author = '저자 정보가 없습니다'
                print("3.저자정보 : %s" %author)
                author2.append(author)
```



상세 항목 추출하기

- no = 추출한 건수
  collect_cnt = 사용자가 요구한 건수

```
78              # 학위정보
79              try :
80                  hak = all_data[3].get_text().strip()
81              except :
82                  hak='학위정보가 없습니다'
83                  hak2.append(hak)
84                  print("6.학위정보 : %s" %hak)
85              else :
86                  hak2.append(hak)
87                  print("6.학위정보 : %s" %hak)
88
89              no += 1
90              print("\n")
91
92              if no > collect_cnt :
93                  break
94
95              time.sleep(1)         # 페이지 변경 전 1초 대기
96
97      a += 1
98      b = str(a)
99
100     try :
101         driver.find_element(By.LINK_TEXT ,'%s' %b).click()
102     except :
103         driver.find_element(By.LINK_TEXT, ('다음 페이지로')).click()
104
105 print("요청하신 작업이 모두 완료되었습니다")
106
```

페이지 변경하기

```
            try :
                hak = all_data[3].get_text().strip()
            except :
                hak='학위정보가 없습니다'
                hak2.append(hak)
                print("6.학위정보 : %s" %hak)
            else :
                hak2.append(hak)
                print("6.학위정보 : %s" %hak)

            no += 1
            print("\n")

            if no > collect_cnt :
                break

            time.sleep(1)        # 페이지 변경 전 1초 대기

    a += 1
    b = str(a)

    try :
        driver.find_element(By.LINK_TEXT ,'%s' %b).click()
    except :
        driver.find_element(By.LINK_TEXT, ('다음 페이지로')).click()

print("요청하신 작업이 모두 완료되었습니다")
```