분석용 데이터
획득 전략

웹 데이터 요약 정보 추출 기법

학습 목표

학습 목표

- Beautiful Soup 프로그램의 원리를 이해하고 설정할 수 있다.
- find( ) / find_all( ) / select( ) 함수 사용방법을 설명할 수 있다.
- Beautiful Soup 프로그램을 활용하여 데이터를 추출할 수 있다.

학습 내용

- Beautiful Soup 원리와 데이터 추출
- Beautiful Soup를 활용한 데이터 추출하기

Beautiful Soup 이해와 설정

1) Beautiful Soup의 역할

- Selenium이 가져온 전체 데이터에서 필요한 데이터만 뽑아냄

예시

Beautiful Soup 이해와 설정

2) Beautiful Soup의 설치와 사용

(1) Beautiful Soup의 설치

Anaconda를 사용할 경우

Anaconda Prompt를 관리자 권한으로 실행한 후

➔ pip install bs4 명령 실행

파이썬을 개별적으로 설치한 경우

윈도우의 cmd(명령프롬프트)를 관리자 권한으로 실행 후

➔ pip install bs4 명령 실행

(2) Beautiful Soup 사용 순서

1. import 명령으로 Beautiful Soup 불러오기
2. Beautiful Soup( ) 함수로 HTML 파싱하여 변수에 저장하기
3. 위 2단계에서 저장한 변수에서 원하는 값 추출하기

파싱(Parsing)

전체 데이터를 분석해서 원하는 형태로 변환하는 것

find()/find_all()/select() 함수 사용법

1) find() 함수 사용법

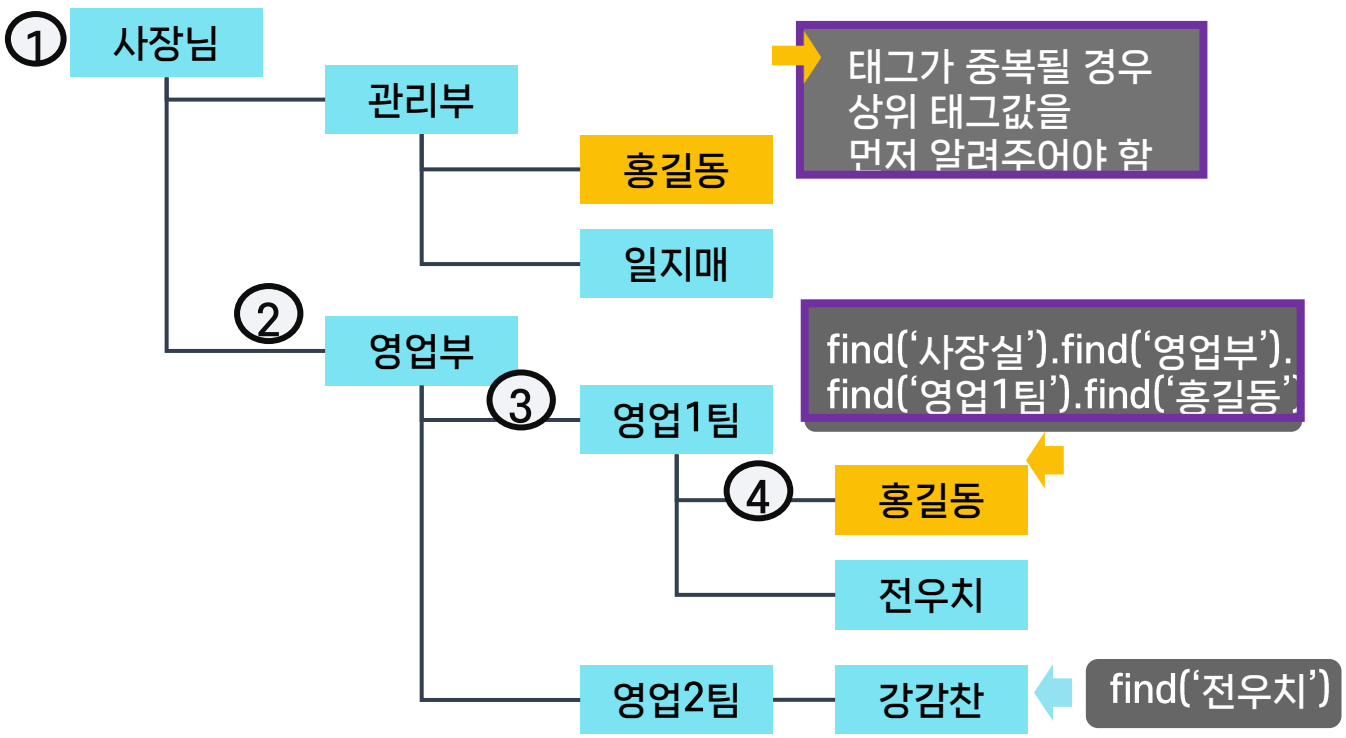
find() 함수란, 사용자가 요청한 데이터 1건 추출하는 함수

find() 함수 사용 예시

[find('HTML 태그이름', 속성 종류='속성값')]

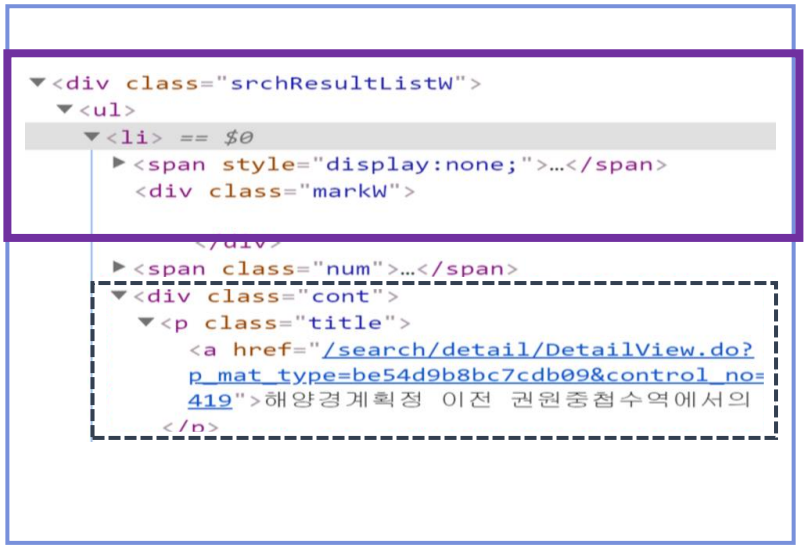
html 예시

<p class="title">
p가 태그이름, class가 속성 종류, title이 속성 값에 해당
<서진수 고향="경남">처럼 속성은 태그를 설명함



find()/find_all()/select()함수사용법

➡ 상위태그 : 위쪽에 있으면서 왼쪽으로 내어쓰기 되어 있는 태그



- BeautifulSoup예제1

➡ find( ) : 주어진 조건을 만족하는 첫 번째 태그값만 가져오기

```
#Beautiful Soup 예제 1
from bs4 import BeautifulSoup
ex1 = '''
<html>
    <head>
        <title> HTML 연습 </title>
    </head>
    <body>
        <p align="center"> text 1 </p>
        <img src="c:₩₩temp₩₩image₩₩솔개.png">
    </body>
<html> '''

soup = BeautifulSoup(ex1, 'html.parser')
print( soup.find('title') )
print( soup.find('p') )
```

```
<title> HTML 연습 </title>
<p align="center"> text 1 </p>
```

find()/find_all()/select()함수사용법

- BeautifulSoup예제2
  - 중복값이있어도첫번째값만가져옴

```
1  #Beautiful Soup 예제 2
2  from bs4 import BeautifulSoup
3  ex1 = '''
4  <html>
5      <head>
6          <title> HTML 연습 </title>
7      </head>
8      <body>
9          <p align="center"> text 1 </p>
10         <p align="right"> text 2 </p>
11         <p align="left"> text 3 </p>
12         <img src="c:₩₩temp₩₩image₩₩솔개.png">
13     </body>
14 <html> '''
15
16 soup = BeautifulSoup(ex1, 'html.parser')
17 print( '첫번째 태그만 추출:' soup.find('p') )
18 print( '속성값을 지정하여 추출:', soup.find('p',align="right") )
```

```
첫번째 태그만 추출: <p align="center"> text 1 </p>
속성값을 지정하여 추출: <p align="right"> text 2 </p>
```

find() / find_all() / select() 함수 사용법

2) Find_all() 함수 사용법

find_all() 함수란, 주어진 조건을 만족하는 모든 값 가져오기

Find_alll() 함수 사용 예시

[Find all('HTML 태그이름', 속성 종류='속성값')]

html 예시

<p class="title">
- p가 태그이름, class가 속성 종류, title이 속성 값에 해당
<서진수 고향="경남">처럼 속성은 태그를 설명함

find()/find_all()/select()함수사용법

- BeautifulSoup예제3
  - 태그가여러개있을경우전체값을다가져옴

```
#Beautiful Soup 예제 3
from bs4 import BeautifulSoup
ex1 = '''
<html>
    <head>
        <title> HTML 연습 </title>
    </head>
    <body>
        <p align="center"> text 1 </p>
        <p align="center"> text 2 </p>
        <p align="center"> text 3 </p>
        <img src="c:₩₩temp₩₩image₩₩솔개.png">
    </body>
<html> '''

soup = BeautifulSoup(ex1, 'html.parser')
print( '1건만가져오기:', soup.find('p') )
print( '전부가져오기:', soup.find_all('p') )
print( '첫번째가져오기:',soup.find_all('p')[0] )
print( '두번째가져오기:',soup.find_all('p')[1] )
print( '세번째가져오기:',soup.find_all('p')[2] )
```

```
1건만가져오기: <p align="center"> text 1 </p>
전부가져오기: [<p align="center"> text 1 </p>, <p align="center"> text 2 </p>, <p align="center"> text 3
첫번째가져오기: <p align="center"> text 1 </p>
두번째가져오기: <p align="center"> text 2 </p>
세번째가져오기: <p align="center"> text 3 </p>
```

인덱싱(Indexing)

문자열에 번호를 매겨 특정 문자를 찾을 수 있는 기능

find() / find_all() / select() 함수 사용법

3) select() 함수 사용법

- select() 함수 사용 예시

  [select('HTML 태그이름', 속성 종류='속성값')]

- html 예시

  <p class="title">

  - p가 태그이름, class가 속성 종류, title이 속성 값에 해당

  <서진수 고향="경남">처럼 속성은 태그를 설명함

- select() 함수의 장점

  ➔ 다양한 옵션을 사용하여 데이터 추출 가능

find()/find_all()/select()함수사용법

- 실습용HTML 소스코드

```
#select( ) 함수 사용하기
# 연습용 html 만들기
ex2='''
<html>
    <head>
        <h1> 사야할 과일
    </head>
    <body>
        <h1> 시장가서 사야할 과일 목록
            <div><p id='fruit1' class='name1' title='바나나'> 바나나
                <span class='price'> 3000원 </span>
                <span class='count'> 10개 </span>
                <span class='store'> 바나나가게 </span>
                <a href='https://www.banana.com'> banana.com </a>
                </p>
            </div>
             <div><p id='fruit2' class='name2' title='체리'> 체리
                <span class='price'> 100원 </span>
                <span class='count'> 50개 </span>
                <span class='store'> 체리가게 </span>
                <a href='https://www.cherry.com'> cherry.com </a>
                </p>
            </div>
             <div><p id='fruit3' class='name3' title='오렌지'> 오렌지
                <span class='price'> 500원 </span>
                <span class='count'> 20개 </span>
                <span class='store'> 오렌지가게 </span>
                <a href='https://www.orange.com'> banana.com </a>
                </p>
            </div>
        </body>
    </html> '''
```

find()/find_all()/select() 함수 사용법

(1) select('.태그이름')

- 사용예시

```
# select('태그이름')
soup2 = BeautifulSoup(ex2, 'html.parser')

soup2.select('p')
```

(2) select('.클래스이름')

- 사용예시

```
#select('.클래스이름')
soup2.select('.name1')
```

- 출력 결과

```
[<p class="name1" id="fruit1" title="바나나"> 바나나
                <span class="price"> 3000원 </span>
 <span class="count"> 10개 </span>
 <span class="store"> 바나나가게 </span>
 <a href="https://www.banana.com"> banana.com </a>
 </p>]
```

주의사항

select('.클래스이름') 사용시 점(.) 생략하지 않게 주의

find()/find_all()/select() 함수 사용법

(3) select('상위태그 > 하위태그 > 하위태그')

- 사용예시

```
#select('상위태그 > 하위태그 > 하위태그')
soup2.select('div > p > span')
```

- 출력 결과

```
[<span class="price"> 3000원 </span>,
 <span class="count"> 10개 </span>,
 <span class="store"> 바나나가게 </span>,
 <span class="price"> 100원 </span>,
 <span class="count"> 50개 </span>,
 <span class="store"> 체리가게 </span>,
 <span class="price"> 500원 </span>,
 <span class="count"> 20개 </span>,
 <span class="store"> 오렌지가게 </span>]
```

주의사항

부등호 표시 앞뒤로 띄어쓰기 반드시 필요

find()/find_all()/select() 함수 사용법

(4) select('상위태그.클래스이름 > 하위태그.클래스이름')

- 사용예시

```
#select('상위태그.클래스이름 > 하위태그.클래스이름')
soup2.select('p.name1 > span.store')
```

- 출력 결과

```
[<span class="store"> 바나나가게 </span>]
```

(5) select('#아이디명')

- 사용예시

```
#select('#아이디명')
soup2.select('#fruit1')
```

- 출력 결과

```
[<p class="name1" id="fruit1" title="바나나"> 바나나
                <span class="price"> 3000원 </span>
 <span class="count"> 10개 </span>
 <span class="store"> 바나나가게 </span>
 <a href="https://www.banana.com"> banana.com </a>
 </p>]
```

주의사항

아이디명 앞에 # 붙여서 입력

find()/find_all()/select() 함수 사용법

(6) select('태그명[속성1=값1]')

- 사용예시

```
#select('태그명[속성1=값1]')
soup2.select('a[href]')
```

- 출력 결과

```
[<a href="https://www.banana.com"> banana.com </a>,
 <a href="https://www.cherry.com"> cherry.com </a>,
 <a href="https://www.orange.com"> banana.com </a>]
```

find()/find_all()/select()함수사용법

- 텍스트데이터만추출하기

① get_text()사용안할경우

- 사용예시

```
#태그뒤의텍스트만추출하기
1txt3=soup2.find_all('p')
2foriintxt3:
3print(i)
```

- 출력결과

```
<p class="name1" id="fruit1" title="바나나"> 바나나
                <span class="price"> 3000원 </span>
<span class="count"> 10개 </span>
<span class="store"> 바나나가게 </span>
<a href="https://www.banana.com"> banana.com </a>
</p>
```

find() / find_all() / select() 함수 사용법

② get_text() 함수로 텍스트만 추출하기

- 사용예시

```
# 태그 뒤의 텍스트만 추출하기
1 txt3 = soup2.find_all('p')
2 for i in txt3 :
3 print(i.get_text().replace('₩n', '') )
```

- 출력 결과

바나나	3000원	10개	바나나가게	Banana.com
체리	100원	50개	체리가게	Cherry.com
오렌지	500원	20개	오렌지가게	Orange.com

데이터 추출 실습

- find() 함수를 이용한 데이터 추출

```
#Beautiful Soup 예제 1
from bs4 import BeautifulSoup
ex1 = '''
<html>
    <head>
        <title> HTML 연습 </title>
    </head>
    <body>
        <p align="center"> text 1 </p>
        <img src="c:\\temp\\image\\솔개.png">
    </body>
<html> '''

soup = BeautifulSoup(ex1, 'html.parser')
print( soup.find('title') )
print( soup.find('p') )
```

```
ex1 = '''
<html>
    <head>
        <title> HTML 연습 </title>
    </head>
    <body>
        <p align="center"> text 1 </p>
        <img src="c:\\temp\\image\\솔개.png">
    </body>
<html> '''

soup = BeautifulSoup(ex1, 'html.parser')
print( soup.find('title') )
print( soup.find('p') )
```

```
<title> HTML 연습 </title>
<p align="center"> text 1 </p>
```

데이터 추출 실습

- find() 함수를 이용한 데이터 추출

```
1  #Beautiful Soup 예제 2
2  from bs4 import BeautifulSoup
3  ex1 = '''
4  <html>
5      <head>
6          <title> HTML 연습 </title>
7      </head>
8      <body>
9          <p align="center"> text 1 </p>
10         <p align="right"> text 2 </p>
11         <p align="left"> text 3 </p>
12         <img src="c:\\temp\\image\\솔개.png">
13     </body>
14 <html> '''
15
16 soup = BeautifulSoup(ex1, 'html.parser')
17 print( '첫번째 태그만 추출:',soup.find('p') )
18 print( '속성값을 지정하여 추출:', soup.find('p',align="right") )
```

```
첫번째 태그만 추출: <p align="center"> text 1 </p>
속성값을 지정하여 추출: <p align="right"> text 2 </p>
```

데이터 추출 실습

- find_all() 함수를 이용한 데이터 추출

```
#Beautiful Soup 예제 3
from bs4 import BeautifulSoup
ex1 = '''
<html>
    <head>
        <title> HTML 연습 </title>
    </head>
    <body>
        <p align="center"> text 1 </p>
        <p align="center"> text 2 </p>
        <p align="center"> text 3 </p>
        <img src="c:\\temp\\image\\솔개.png">
    </body>
<html> '''

soup = BeautifulSoup(ex1, 'html.parser')
print( '1건만가져오기:', soup.find('p') )
print( '전부가져오기:', soup.find_all('p') )
print( '첫번째가져오기:',soup.find_all('p')[0] )
print( '두번째가져오기:',soup.find_all('p')[1] )
print( '세번째가져오기:',soup.find_all('p')[2] )
```

```
1건만가져오기: <p align="center"> text 1 </p>
전부가져오기: [<p align="center"> text 1 </p>, <p align="center"> text 2 </p>, <p align="center"> text 3 </p>]
첫번째가져오기: <p align="center"> text 1 </p>
두번째가져오기: <p align="center"> text 2 </p>
세번째가져오기: <p align="center"> text 3 </p>
```

데이터 추출 실습

- select() 함수를 이용한 데이터 추출

```
#select( ) 함수 사용하기
# 연습용 html 만들기
ex2='''
<html>
    <head>
        <h1> 사야할 과일
    </head>
    <body>
        <h1> 시장가서 사야할 과일 목록
            <div><p id='fruit1' class='name1' title='바나나'> 바나나
                <span class='price'> 3000원 </span>
                <span class='count'> 10개 </span>
                <span class='store'> 바나나가게 </span>
                <a href='https://www.banana.com'> banana.com </a>
                </p>
            </div>
             <div><p id='fruit2' class='name2' title='체리'> 체리
                <span class='price'> 100원 </span>
                <span class='count'> 50개 </span>
                <span class='store'> 체리가게 </span>
                <a href='https://www.cherry.com'> cherry.com </a>
                </p>
            </div>
             <div><p id='fruit3' class='name3' title='오렌지'> 오렌지
                <span class='price'> 500원 </span>
                <span class='count'> 20개 </span>
                <span class='store'> 오렌지가게 </span>
                <a href='https://www.orange.com'> orange.com </a>
                </p>
```

데이터 추출 실습

- select('.태그이름')

```
In [12]:
# select('태그이름')
soup2 = BeautifulSoup(ex2 , 'html.parser')

soup2.select('p')
```

```
Out[12]: [<p class="name1" id="fruit1" title="바나나"> 바나나
                <span class="price"> 3000원 </span>
<span class="count"> 10개 </span>
<span class="store"> 바나나가게 </span>
<a href="https://www.banana.com"> banana.com </a>
</p>,
<p class="name2" id="fruit2" title="체리"> 체리
                <span class="price"> 100원 </span>
<span class="count"> 50개 </span>
<span class="store"> 체리가게 </span>
<a href="https://www.cherry.com"> cherry.com </a>
</p>,
<p class="name3" id="fruit3" title="오렌지"> 오렌지
                <span class="price"> 500원 </span>
```

- select(.클래스이름)

※클래스이름 앞에 점(.) 반드시 입력

```
<span class="count"> 20개 </span>
<span class="store"> 오렌지가게 </span>
<a href="https://www.orange.com"> orange.com </a>
</p>]
```

```
In [13]:
#select('.클래스이름')
soup2.select(' .name1 ')
```

```
Out[13]: [<p class="name1" id="fruit1" title="바나나"> 바나나
                <span class="price"> 3000원 </span>
<span class="count"> 10개 </span>
<span class="store"> 바나나가게 </span>
<a href="https://www.banana.com"> banana.com </a>
</p>]
```

데이터 추출 실습

- select('상위태그>하위태그>하위태그')
  ※부등호사이에 공백한칸이상 꼭 입력

```
In [14]:
1 #select( ' 상위태그 > 하위태그 > 하위태그 ' )
2 soup2.select(' div > p > span')
```

```
Out[14]: [<span class="price"> 3000원 </span>,
 <span class="count"> 10개 </span>,
 <span class="store"> 바나나가게 </span>,
 <span class="price"> 100원 </span>,
 <span class="count"> 50개 </span>,
 <span class="store"> 체리가게 </span>,
 <span class="price"> 500원 </span>,
 <span class="count"> 20개 </span>,
 <span class="store"> 오렌지가게 </span>]
```

- select('상위태그.클래스이름>하위태그.클래스이름')

```
In [15]:
1 # select( ' 상위태그.클래스이름 > 하위태그.클래스이름 ' )
2 soup2.select(' p.name1 > span.store ')
```

```
Out[15]: [<span class="store"> 바나나가게 </span>]
```

데이터 추출 실습

- select('#아이디명')

```
In [15]:
# select( '상위태그.클래스이름 > 하위태그.클래스이름' )
soup2.select(' p.name1 > span.store ')
```

```
Out[15]: [<span class="store"> 바나나가게 </span>]
```

```
In [16]:
# select( '#아이디명' )
soup2.select(' #fruit1')
```

```
Out[16]: [<p class="name1" id="fruit1" title="바나나"> 바나나
                  <span class="price"> 3000원 </span>
<span class="count"> 10개 </span>
<span class="store"> 바나나가게 </span>
<a href="https://www.banana.com"> banana.com </a>
</p>]
```

- select('태그명[속성1=값1]')

```
In [17]:
# select( '태그명[속성1=값1]' )
soup2.select('a[href]')
```

```
Out[17]: [<a href="https://www.banana.com"> banana.com </a>,
 <a href="https://www.cherry.com"> cherry.com </a>,
 <a href="https://www.orange.com"> orange.com </a>]
```

데이터 추출 실습

- 텍스트 데이터만 추출하기
  - get_text() 사용 안 할 경우

In [18]:
```
# 태그 뒤의 텍스트만 추출하기
txt3 = soup2.find_all('p')
for i in txt3 :
    print(i)
```

```
<p class="name1" id="fruit1" title="바나나"> 바나나
                <span class="price"> 3000원 </span>
<span class="count"> 10개 </span>
<span class="store"> 바나나가게 </span>
<a href="https://www.banana.com"> banana.com </a>
</p>
<p class="name2" id="fruit2" title="체리"> 체리
                <span class="price"> 100원 </span>
<span class="count"> 50개 </span>
<span class="store"> 체리가게 </span>
<a href="https://www.cherry.com"> cherry.com </a>
</p>
<p class="name3" id="fruit3" title="오렌지"> 오렌지
                <span class="price"> 500원 </span>
<span class="count"> 20개 </span>
<span class="store"> 오렌지가게 </span>
<a href="https://www.orange.com"> orange.com </a>
</p>
```

  - get_text() 함수로 텍스트만 추출하기

In [19]:
```
# 태그 뒤의 텍스트만 추출하기
txt3 = soup2.find_all('p')
for i in txt3 :
    print(i.get_text().replace('\n','') )
```

```
바나나                3000원  10개  바나나가게  banana.com
체리                 100원  50개  체리가게  cherry.com
오렌지                500원  20개  오렌지가게  orange.com
```

데이터 추출 실습

- riss.kr 사이트에서 특정 키워드로 자동 검색하기

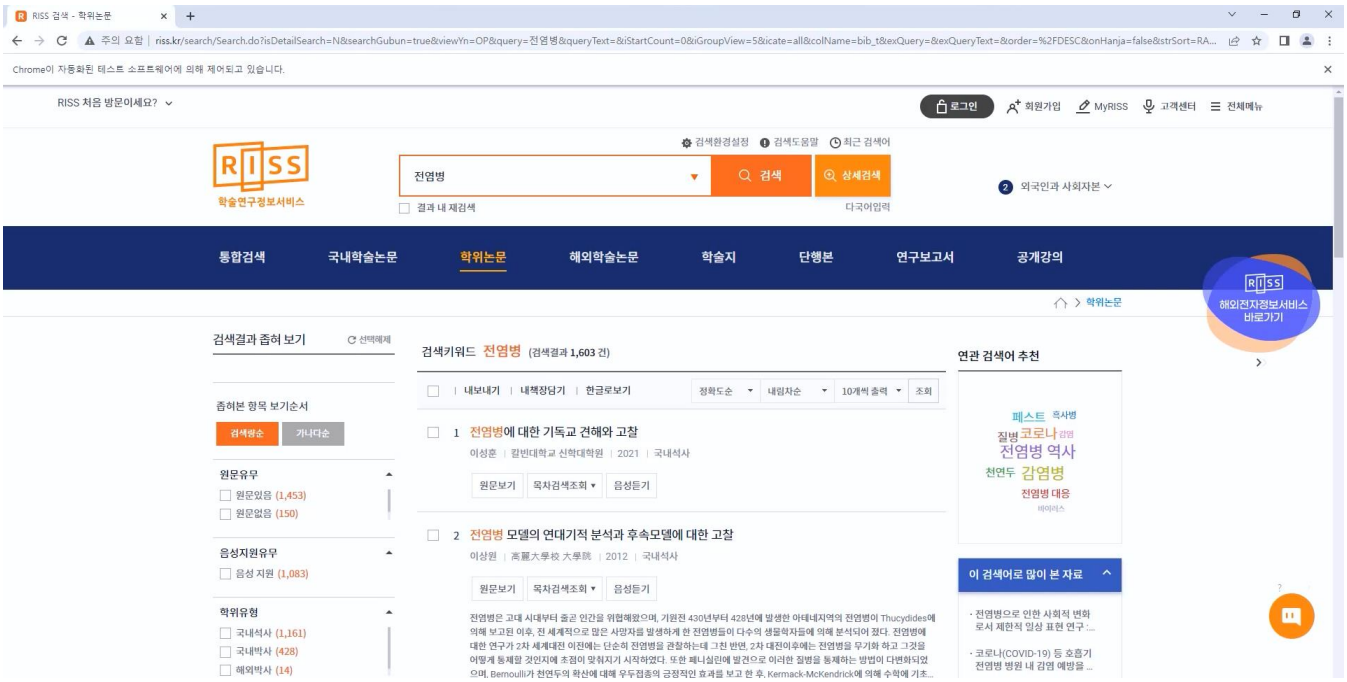
In [ ]:

```
#3차시 : riss.kr 사이트에서 특정 키워드로 자동 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 자료 수집용 웹크롤러입니다.")
print("=" *100)
query_txt = input('1.수집할 자료의 키워드는 무엇입니까?(예: 전염병): ')
print("\n")

#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")

#Step 5.학위 논문 선택하기
driver.find_element(By.LINK_TEXT,'학위논문').click()
time.sleep(2)

#Step 6.Beautiful Soup 로 본문 내용만 추출하기
from bs4 import BeautifulSoup
html_1 = driver.page_source #현재 페이지의 전체 소스코드를 다 가져오기
soup_1 = BeautifulSoup(html_1, 'html.parser')

content_1 = soup_1.find('div','srchResultListW').find_all('li')
for i in content_1 :
    print(i.get_text().replace("\n"," ").strip())
    print("\n")
```

```
================================================================================
 이 크롤러는 riss 사이트의 논문 자료 수집용 웹크롤러입니다.
================================================================================
1.수집할 자료의 키워드는 무엇입니까?(예: 전염병): 전염병
```

데이터 추출 실습

- riss.kr 사이트에서 특정 키워드로 자동 검색하기




```
38 html_1 = driver.page_source #현재 페이지의 전체 소스코드를 다 가져오기
39 soup_1 = BeautifulSoup(html_1, 'html.parser')
40
41 content_1 = soup_1.find('div','srchResultListW').find_all('li')
42 for i in content_1 :
43     print(i.get_text().replace("\n"," ").strip())
44     print("\n")
```

```
==========
 이 크롤러는 riss 사이트의 논문 자료 수집용 웹크롤러입니다.
==========
1.수집할 자료의 키워드는 무엇입니까?(예: 전염병): 전염병

1 전염병에 대한 기독교 견해와 고찰 이성훈 칼빈대학교 신학대학원 2021 국내석사 RANK : 27772927 원문보기 목차검색조회 음성듣기

원문보기

목차검색조회

음성듣기

2 전염병 모델의 연대기적 분석과 후속모델에 대한 고찰 이상원 高麗大學校 大學院 2012 국내석사 RANK : 27772927 원문보기 목차검색조회 음성듣기
```

```
In [ ]: 1
```