2020 D&A

DEEP SESSION

3. 크롤링

D&A

Contents

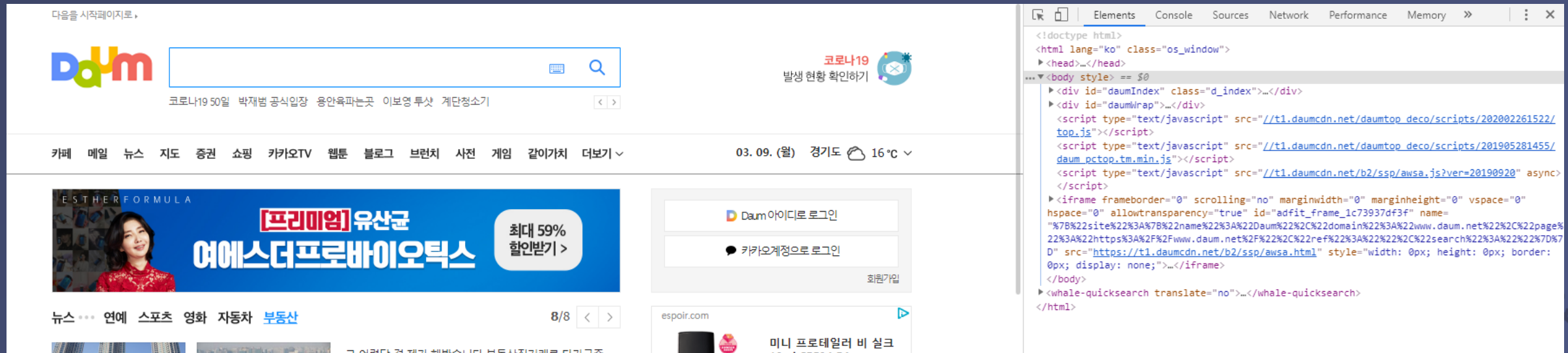
D&A

01 크롤링(Crawling)이란?

소프트웨어가 웹을 돌아다니며
필요한 정보를 긁어오는 것을 말함
(web scraping)

웹페이지는 html의 태그로 구성되어 있음!
크롤링은 각각의 태그에 저장되어 있는
데이터를 긁어오는 일을 함

F12를 누르면 확인 가능!



02 HTML의 구조

일반적으로 태그는 <html></html> 구조

정보를 찾는 과정
<h1>D&A</h1>
-> h1태그에 D&A라는 정보가 들어있음!

태그의 종류 (우리는 이를 이용해 정보를 찾을 것이니 외울 필요는 없음)

태그	설명	예	속성
h1, h2, h3, h4, h5, h6	글자의 크기를 조절	D&A D&A 2020 D&A	
b	굵게	D&A **D&A**	
u	밑줄	<u>D&A</u> D&A	
br	줄바꿈	 D&A</br> D&A	
a	링크 연결	네이버 네이버	href(연결할링크), title(확인문구)
span	문서 요소를 묶음	딥세션 딥세션	
li	목록의 리스트	2020 2020	
div	레이아웃		
ul	순서가 없는 목록		
ol	순서가 있는 목록		
button	클릭할 버튼 생성	<button>버튼</button>	type(종류)
p	한 단락	Deep Session 화이팅 Deep Session 파이팅	
img	이미지 파일 넣기	<img src= 'microphone.jpg' width= '100%' >	src(이미지주소), width(가로길이), height(세로길이)
input	입력도구 만들기	<input type= 'text' >	type(종류), value(초기값), placeholder(안내문)
textarea	큰 입력영역 만들기	<textarea rows= '5' cols= ' 10' ></textarea>	rows(행수), cols(열수)

★ a
★ img

02 HTML의 구조

똑같은 태그가 엄청 많음

※ 여러 개가 존재하는 태그에는
대부분 class나 id로 구분함!!

id : #을 사용해 정보 찾기

class : . 을 사용해 정보 찾기

class는 여러 개가 존재할 수 있으며 각각의 클래스는 띄어쓰기로 구분됨

※ 태그는 상하관계가 있기 때문에
찾고자 하는 여러 태그가 존재한다면 이를 이용

자식 관계 : 바로 한단계 아래 하위태그. > 사용

자손관계 : 모든 하위태그. 띄어쓰기 사용

선택자 > *하면 모든 자식선택자 선택

D&A

Requests

웹상의 데이터를 가져올 수 있는 패키지

```
!pip install requests
import requests
```

BeautifulSoup4

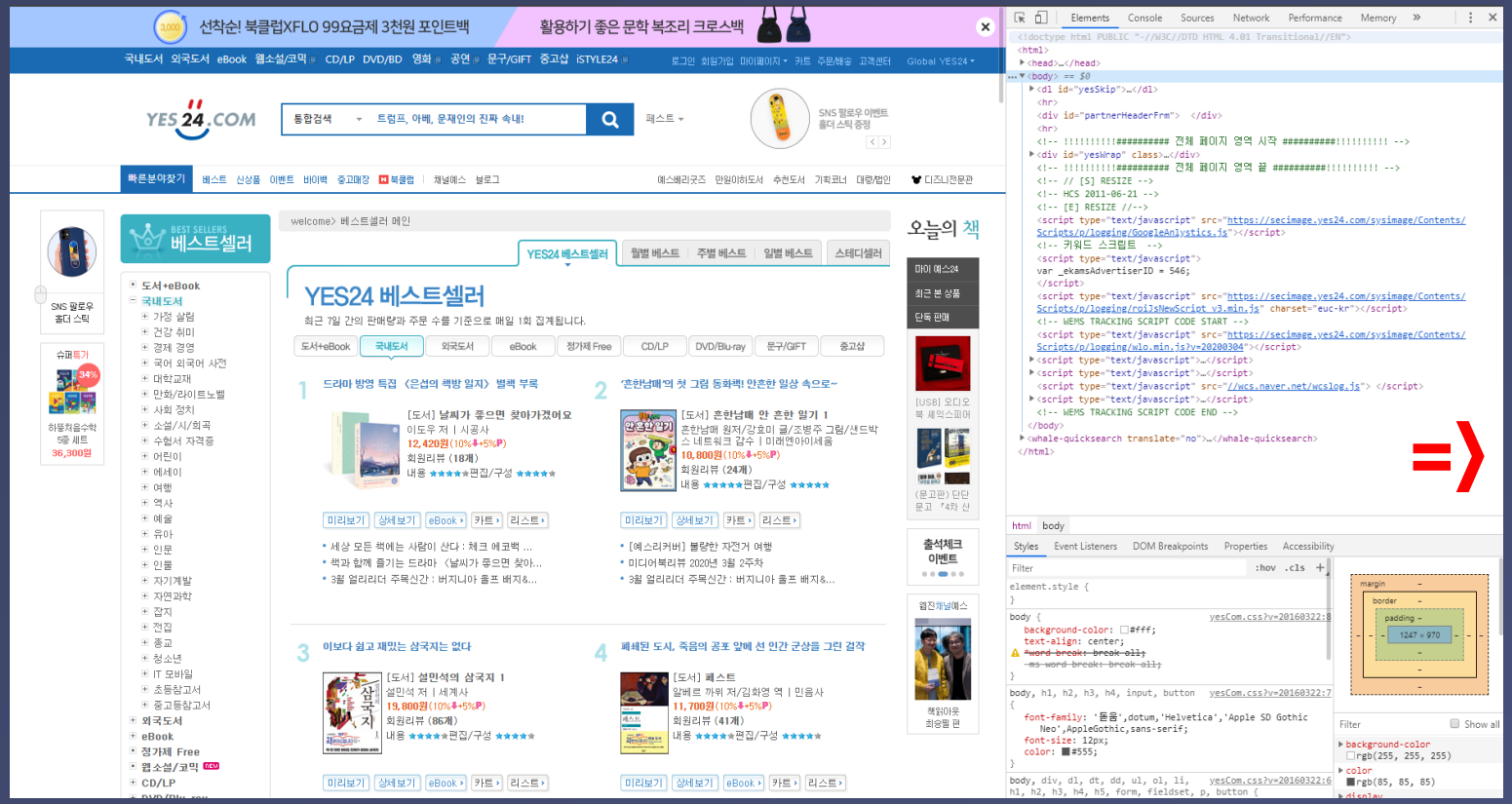
html코드를 파싱해 원하는 데이터를 추출할 수 있는 패키지

```
!pip install BeautifulSoup4
from bs4 import BeautifulSoup
```

03 Requests와 BeautifulSoup4

```
In [2]: raw = requests.get('http://www.yes24.com/24/Category/BestSeller')
        print(raw)

<Response [200]>
```

get('웹페이지주소') : 웹페이지를 접속해 정보를 가져옴



=>

```
In [3]: raw.text
```

text : 가져온 웹페이지의 소스코드

```
Out[3]: '₩r₩n    <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">₩r₩n    <html>₩r₩n₩r₩n<head><base href="http://www.yes24.com/24/" />₩r₩n<meta http-equiv="X-UA-Compatible" content="IE=5" />₩r₩n₩r₩n    <meta http-equiv="Content-Type" content="text/html;charset=euc-kr" />₩r₩n₩r₩n<meta name="viewport" content="width=1170" />₩r₩n₩r₩n<title>YES24 | 대한민국 대표 인터넷서점 | 베스트셀러</title>₩r₩n₩r₩n
<meta name="title" content="YES24 - 대한민국 대표 인터넷서점" />₩r₩n    <meta name="description" content="YES24는 대한민국 1위 인터넷 온라인 서점 입니다. 국내 최대의 도서정보를 보유하고 있으며, 음반, DVD, 공연, 영화까지 다양한 문화 콘텐츠 및 서비스를 제공합니다." />₩r₩n    <meta name="keywords" content="인터넷 서점, 온라인 쇼핑, 상품 추천, 쇼핑몰, 상품 검색, 도서 정보, 국내도서, 외국도서, 전자책, eBook, 이북, 크레마, 공연, 콘서트, 뮤지컬, 영화, 음반, 예매, DVD, 블루레이, 예스24, YES24, 교보문고, 알라딘" />₩r₩n    <meta property="og:image" content="https://secimage.yes24.com/sysimage/renew/logo_meta.png" />₩r₩n₩r₩n<script type="text/javascript" src="https://secimage.yes24.com/sysimage/Contents/Scripts/p/jquery/jquery-1.2.6.min.js"></script>₩r₩n<script type="text/javascript" src="https://secimage.yes24.com/sysimage/Contents/Scripts/p/jquery/jquery.menu-aim.js?v=20140801"></script>₩r₩n<script type="text/javascript" src="https://secimage.yes24.com/sysimage/Contents/Scripts/p/jquery/jquery.easing.1.3.min.js?v=20140801"></script>₩r₩n<script type="text/javascript" src="http://www.yes24.com/JavaScript/util.js?v=20191127"></script>₩r₩n
```

03 Requests와 BeautifulSoup4

```
In [4]: html = BeautifulSoup(raw.text,'html.parser')
        print(html)
```

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">

<html>
<head><base href="http://www.yes24.com/24/"/>
<meta content="IE=5" http-equiv="X-UA-Compatible"/>
<meta content="text/html;charset=utf-8" http-equiv="Content-Type"/>
<meta content="width=1170" name="viewport"/>
<title>YES24 | 대한민국 대표 인터넷서점 | 베스트셀러</title>
<meta content="YES24 - 대한민국 대표 인터넷서점" name="title"/>
<meta content="YES24는 대한민국 1위 인터넷 온라인 서점 입니다. 국내 최대의 도서정보를 보유하고 있으며, 음반, DVD, 공연, 영화까지 다양한
문화 콘텐츠 및 서비스를 제공합니다." name="description"/>
<meta content="인터넷 서점, 온라인 쇼핑, 상품 추천, 쇼핑몰, 상품 검색, 도서 정보, 국내도서, 외국도서, 전자책, eBook, 이북, 크레마, 공
연, 콘서트, 뮤지컬, 영화, 음반, 예매, DVD, 블루레이, 예스24, YES24, 교보문고, 알라딘" name="keywords"/>
<meta content="https://secimage.yes24.com/sysimage/renew/logo_meta.png" property="og:image"/>
<script src="https://secimage.yes24.com/sysimage/Contents/Scripts/p/jquery/jquery-1.2.6.min.js" type="text/javascript"></script>
<script src="https://secimage.yes24.com/sysimage/Contents/Scripts/p/jquery/jquery.menu-aim.js?v=20140801" type="text/javascript"></scri
pt>
<script src="https://secimage.yes24.com/sysimage/Contents/Scripts/p/jquery/jquery.easing.1.3.min.js?v=20140801" type="text/javascript">
```

BeautifulSoup(웹페이지의 소스코드, 'html.parser')
웹페이지의 소스코드를 html단위로 구분해줌

D&A

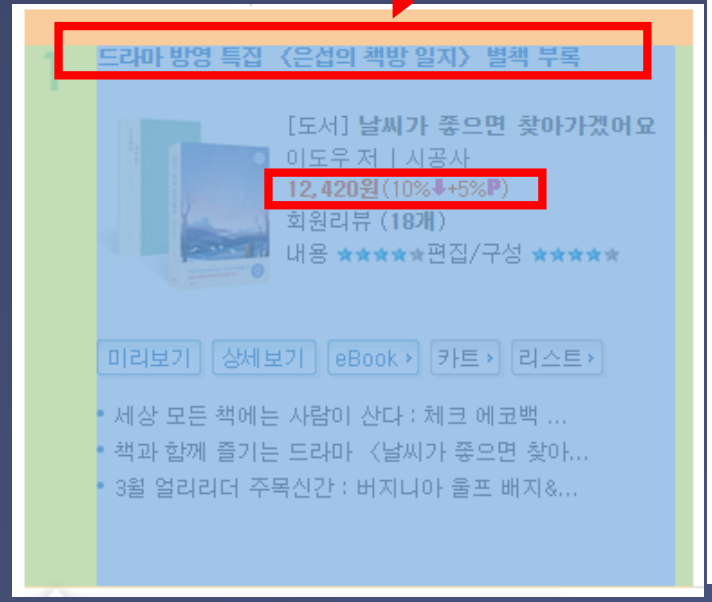
03 Requests와 BeautifulSoup4

1. 컨테이너 구조를 파악 ※ 페이지 검사를 통해 필요한 정보가 다 담긴 컨테이너를 찾아야 함

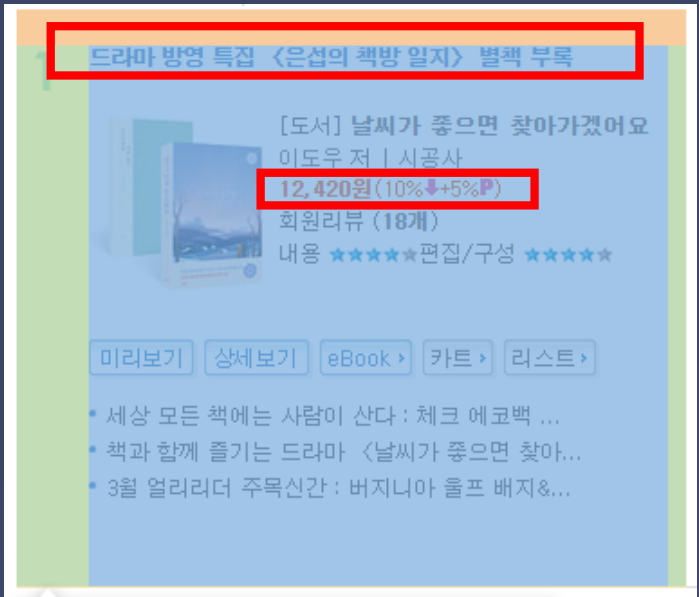
2. 필요한 정보가 어떤 태그에 담겨져 있는지 확인

3. 다른 정보가 담겨져 있는 중복된 태그가 없는지 확인

4. 크롤러 만들기



03 Requests와 BeautifulSoup4



컨테이너의 태그
div#bestList ol > li

홍보물 카피의 태그
div#bestList ol > p.copy

가격의 태그
div#bestList ol > p.price

```
In [5]: books = html.select('div#bestList ol > li')
        print(books)
```

select : html의 태그를 선택함 그 안의 모든 태그를 가져옴

```
[<li class="num1">
<p class="copy"><a href="/Product/Goods/86895523">드라마 방영 특집 〈은섭의 책방
일지〉 별책 부록</a></p>
<p class="image" id="location_0">
<a href="/Product/Goods/86895523">
<img alt="날씨가 좋으면 찾아가겠어요" src="http://image.yes24.com/goods/86895523/
S">
</img></a>
</p>
<p>[도서] <a href="/Product/Goods/86895523">날씨가 좋으면 찾아가겠어요</a></p>
<p class="aupu"><a href="http://www.yes24.com//SearchCorner/Result?domain=ALL&amp;
author_yn=Y&amp;query=&amp;auth_no=147966" target="_blank">이도우</a> 저 | <a href
="http://www.yes24.com//SearchCorner/Result?domain=ALL&amp;company_yn=Y&amp;query=
시공사">시공사</a></p>
<p class="price"><strong>12,420원</strong>(10%<img src="http://image.yes24.com/sys
image/wel/i_dc.gif"/>+5%<img src="http://image.yes24.com/sysimage/wel/i_p.gif"/>)
</p>
<p>회원리뷰 (<a href="/Product/Goods/86895523#Review">18개</a>)</p>
<p>
```

03 Requests와 BeautifulSoup4

```
In [10]: copy = books[0].select_one('p.copy')
         print(copy)
```

select_one : html의 태그를 선택. 첫번째 하나의 태그만 가져옴

```
<p class="copy"><a href="/Product/Goods/86895523">드라마 방영 특집 〈은섭의 책방 일지〉 별책 부록</a></p>
```

```
In [11]: copy = books[0].select_one('p.copy').text
         print(copy)
```

```
드라마 방영 특집 〈은섭의 책방 일지〉 별책 부록
```

```
In [7]: price = books[0].select_one('p.price').text
        print(price)
```

```
12,420원(10%+5%)
```

```
In [9]: for book in books:
            copy = book.select_one('p.copy').text
            price = book.select_one('p.price').text
            print(price,copy)
```

```
12,420원(10%+5%) 드라마 방영 특집 〈은섭의 책방 일지〉 별책 부록
10,800원(10%+5%) '흔한남매' 의 첫 그림 동화책! 안흔한 일상 속으로~
19,800원(10%+5%) 이보다 쉽고 재밌는 삼국지는 없다
11,700원(10%+5%) 폐쇄된 도시, 죽음의 공포 앞에 선 인간 군상을 그린 걸작
19,800원(10%+5%) 이보다 쉽고 재밌는 삼국지는 없다
14,400원(10%+5%) 전승환이 꼽은 인생 문장
14,400원(10%+5%) 하버드 성공 공식 "신호를 차단하고 깊이 몰입하라"
14,400원(10%+5%) 하루 1분이면 교양 시민
24,120원(10%+5%) 리더는 타고나는가 만들어지는가
10,800원(10%+5%) 설쌤과 함께 위인들의 어린 시절을 만나요!
12,150원(10%+5%) 현실에서 딱 1cm 벗어나 행복찾기!
14,850원(10%+5%) 영화 〈작은 아씨들〉 공식 오리지널 커버 디자인
10,800원(10%+5%) 흔한 일상 속에 숨겨진 반짝거리는 웃음과 재미!
11,610원(10%+5%) 2020 최신판으로 토익 기출 보카 업그레이드
17,820원(10%+5%) 지대넓얕 0권, 모든 지식의 시작
16,650원(10%+5%) 우리는 코스모스에서 태어났다! 인류를 품은 거대한 우주, 그 기원과 신비, 인간의 본질을 밝히다
11,700원(10%+5%) '다시 한 번 사랑해보기로 한' 모든 이를 향한 따뜻한 시선
10,800원(10%+5%) 고통과 공감의 능력을 깨우치게 하는 소설
25,200원(10%+5%) 서울대 도서관 대출도서 부동의 1위! 퓰리처상 수상작품
```

이렇게 for문을 통해
여러 개의 데이터를
한 번에 수집할 수 있음!

D&A

03 Requests와 BeautifulSoup4

불러온 웹페이지에서 다른 페이지로 넘어가 정보를 가져올 때

```
In [30]: raw2 = requests.get('http://yes24.com'+books[0].select_one('p.copy a').attrs['href'])

In [32]: html2 = BeautifulSoup(raw2.text,'html.parser')
         title = html2.select_one('h2.gd_name').text
         print(title)

날씨가 좋으면 찾아가겠어요
```

특정 태그에 속해있는 href태그를 불러오면 연결된 페이지의 정보가 불러와짐

이미지 정보를 가져올 때

```
In [33]: from urllib.request import urlretrieve

In [36]: img = books[0].select_one('ol li p.image img').attrs['src']

In [39]: urlretrieve(img,'책1.png')
```

src로 불러와짐

D&A

03 Requests와 BeautifulSoup4

※ 검색 자동화

www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%C0%FC%C3%BC&query=python


```
In [16]: keyword = 'python'

In [17]: raw = requests.get('http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%C0%FC%C3%BC&query='+keyword)
```

for문을 통해 자동화 가능

```
In [19]: keyword = ['python','java','r']

In [28]: for k in keyword:
             raw = requests.get('http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%C0%FC%C3%BC&query='+k)
             html = BeautifulSoup(raw.text,'html.parser')
             books = html.select('td.goods_infogrp')
             for book in books[:20]:
                 title = book.select_one('a').text
                 writer = book.select_one('div.goods_info a').text
                 price = book.select_one('div.goods_price em.yes_b').text
                 print(title,writer,price)
```

혼자 공부하는 파이썬 윤인성 16,200
Do it! 점프 투 파이썬 박응용 16,920
밑바닥부터 시작하는 딥러닝 사이토 고키 21,600
파이썬 머신러닝 완벽 가이드 권철민 34,200
밑바닥부터 시작하는 딥러닝 2 사이토 고키 26,100
파이썬과 리액트를 활용한 주식 자동거래 시스템 구축 박재현 25,200
케라스 창시자에게 배우는 딥러닝 프랑소와 숄레 29,700
파이썬으로 배우는 프로그래밍 기초 크리스 로피 13,000
머신 러닝 교과서 with 파이썬, 사이킷런, 텐서플로 세바스찬 라시카 29,700
파이썬 기초 문법 이원하 0
파이썬 Phython 김영천 25,000
파이썬 연습 김기용 26,000
알기 쉬운 파이썬 SQL 코딩하기 정현희 13,500
파이썬으로 배우는 수치 데이터 처리 김동근 19,800
딥러닝 AI프로젝트 실사례 Santanu Pattanayak 29,000
예제로 배우는 자연어 처리 기초 쇼홈 고시 31,500
모두의 파이썬 이승찬 10,800
Do it! 점프 투 파이썬 + 딥러닝 입문 박응용 34,740
실무자를 위한 파이썬 Python 100제 오승환 18,000
모두의 데이터 분석 with 파이썬 송석리 16,200
글자 바로 쓰기 초등 국어 1-1 (2020년용) 미래엔 초등 국어 연구회 7,650
하루 한장 국어 + 수학 1-1 세트 (2020년용) 미래엔콘텐츠연구회 26,460
글자 바로 쓰기 초등 국어 2-1 (2020년용) 미래엔 초등 국어 연구회 7,650

D&A

03 Requests와 BeautifulSoup4

※ 페이지 넘기기 자동화

www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%C0%FC%C3%BC&query=python

www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%c0%fc%c3%bc&query=python&PageNumber=2&scode=012

아까와 마찬가지로 페이지 주소를 살피면 답이 나옴!


```
In [32]: for k in keyword:
             for num in range(1,10):
                 raw = requests.get(
                     'http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%C0%FC%C3%BC&query='+k
                     +'&PageNumber='+str(num))
                 html = BeautifulSoup(raw.text,'html.parser')
                 books = html.select('td.goods_infogrp')
                 for book in books[:20]:
                     title = book.select_one('a').text
                     writer = book.select_one('div.goods_info a').text
                     price = book.select_one('div.goods_price em.yes_b').text
                     print(title,writer,price)
```

```
혼자 공부하는 파이썬 윤인성 16,200
Do it! 점프 투 파이썬 박응용 16,920
밑바닥부터 시작하는 딥러닝  사이토 고키 21,600
파이썬 머신러닝 완벽 가이드 권철민 34,200
밑바닥부터 시작하는 딥러닝 2 사이토 고키 26,100
파이썬과 리액트를 활용한 주식 자동거래 시스템 구축 박재현 25,200
케라스 창시자에게 배우는 딥러닝 프랑소와 숄레 29,700
파이썬으로 배우는 프로그래밍 기초 크리스 로피 13,000
머신 러닝 교과서 with 파이썬, 사이킷런, 텐서플로 세바스찬 라시카 29,700
파이썬 기초 문법 이원하 0
파이썬 Phython 김영천 25,000
파이썬 연습 김기용 26,000
알기 쉬운 파이썬 SQL 코딩하기 정현희 13,500
파이썬으로 배우는 수치 데이터 처리  김동근 19,800
딥러닝 AI프로젝트 실사례 Santanu Pattanayak 29,000
예제로 배우는 자연어 처리 기초 쇼홈 고시 31,500
모두의 파이썬 이승찬 10,800
Do it! 점프 투 파이썬 + 딥러닝 입문 박응용 34,740
실무자를 위한 파이썬 Python 100제 오승환 18,000
```

D&A

03 Requests와 BeautifulSoup4

※ 크롤링하려고 하던 페이지가 막혀서 접근불가일 때
해결방안 : headers={'User-Agent':'접근자'} 를 설정

```
In [33]: raw = requests.get('https://movie.naver.com/movie/running/current.nhn',headers={'User-Agent':'Mozilla/5.0'})
         html = BeautifulSoup(raw.text,'html.parser')
```

Mozilla/5.0의 탈을 쓰고 웹페이지에 접근한다는 뜻

※ 똑같은 이름의 태그라 원하는 정보를 수집하지 못할 때

```
In [36]: movies = html.select('dl.lst_dsc')

         for movie in movies:
             title = movie.select_one('dt.tit a').text
             genre = movie.select_one('dl.info_txt1 dd a').text
             actor = movie.select_one('dl.info_txt1 dd a').text
             print(title,genre,actor)
```

```
인비저블맨 공포 공포
다크 워터스 드라마 드라마
1917 드라마 드라마
더 보이 2: 돌아온 브람스 공포 공포
지푸라기라도 잡고 싶은 짐승들 범죄 범죄
작은 아씨들 드라마 드라마
정직한 후보 코미디 코미디
시원찮은 그녀를 위한 육성방법 피날레 애니메이션 애니메이션
찬실이는 복도 많지 드라마 드라마
스타 이즈 본 드라마 드라마
울프 콜 액션 액션
악몽 미스터리 미스터리
젠틀맨 범죄 범죄
타오르는 여인의 초상 드라마 드라마
리암 갤러거 다큐멘터리 다큐멘터리
작가 미상 드라마 드라마
하이, 젝시 코미디 코미디
슬럼독 밀리어네어 범죄 범죄
비긴 어게인 드라마 드라마
```

```
▼<dl class="info_txt1">
   <dt class="tit_t1">개요</dt>
  ▼<dd> == $0
   ▼<span class="link_txt">
      <a href="/movie/sdb/browsing/bmovie.nhn?genre=1">드라마</a>
      <!-- N=a:nol.genre,r:1 -->
    </span>
```

```
▼<dl class="info_txt1">
   <dt class="tit_t1">개요</dt>
  ▶<dd>…</dd>
   <dt class="tit_t2">감독</dt>
  ▶<dd>…</dd>
   <dt class="tit_t3">출연</dt>
  ▼<dd>
   ▼<span class="link_txt"> == $0
      <a href="/movie/bi/pi/basic.nhn?code=6176">마크 러팔로</a>
      <!-- N=a:nol.actor,r:1 -->
      ",
                                                    "
```

Q&A

03 Requests와 BeautifulSoup4

해결방안 : 같은 태그 뒤에 :nth-of-type(태그의 순서)를 덧붙임

```
In [37]: movies = html.select('dl.lst_dsc')

         for movie in movies:
             title = movie.select_one('dt.tit a').text
             genre = movie.select_one('dl.info_txt1 dd:nth-of-type(1) a').text
             actor = movie.select_one('dl.info_txt1 dd:nth-of-type(3) a').text
             print(title,genre,actor)
```

info_txt1 다음의 dd가 여러 개 있는 게 겹치는 것이기 때문에 dd:nth-of-type사용

```
인비저블맨 공포 엘리자베스 모스
다크 워터스 드라마 마크 러팔로
1917 드라마 조지 맥케이
더 보이 2: 돌아온 브람스 공포 케이티 홈즈
지푸라기라도 잡고 싶은 짐승들 범죄 전도연
작은 아씨들 드라마 시얼샤 로넌
정직한 후보 코미디 라미란
시원찮은 그녀를 위한 육성방법 피날레 애니메이션 마츠오카 요시츠구
찬실이는 복도 많지 드라마 강말금
스타 이즈 본 드라마 브래들리 쿠퍼
울프 콜 액션 프랑수아 시빌
악몽 미스터리 오지호
젠틀맨 범죄 매튜 맥커너히
타오르는 여인의 초상 드라마 아델 하에넬
리암 갤러거 다큐멘터리 리암 갤러거
작가 미상 드라마 톰 쉘링
하이, 젝시 코미디 아담 드바인
슬럼독 밀리어네어 범죄 데브 파텔
비긴 어게인 드라마 키이라 나이틀리
```

※ 정적페이지와 동적페이지?

정적페이지는 웹페이지의 화면이
클릭이나 입력을 추가했을 때 웹페이지 주소가 변경되는 것

동적페이지는 웹페이지 주소가 변경되지는 않지만
보이는 웹페이지가 변경되는 페이지

javascript는 외관적인 페이지를 변경해주는 용도이기 때문에 동적페이지 수집방법으로 수집

Selenium

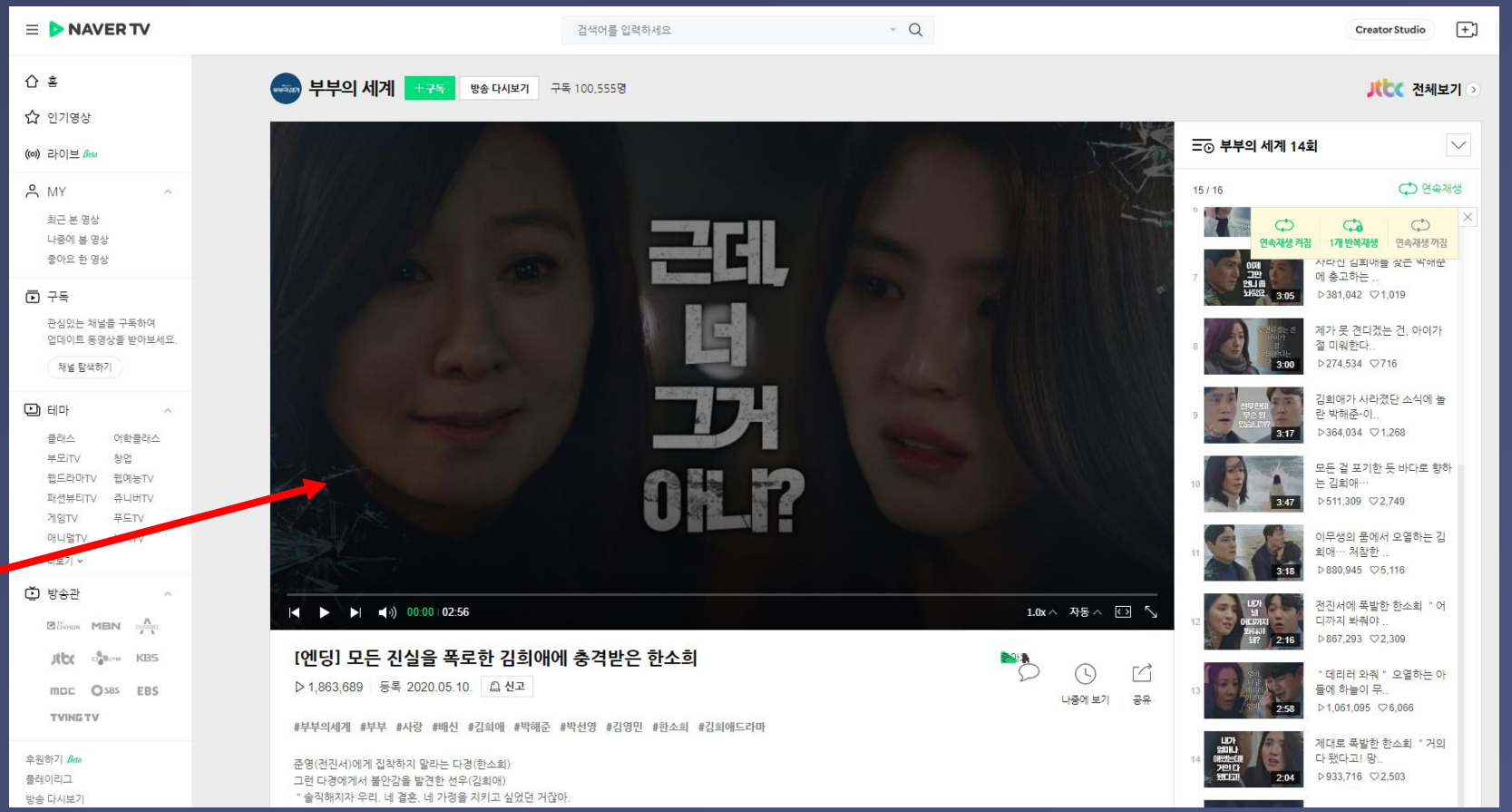
웹브라우저를 컨트롤하여 웹을 자동화시켜주는 패키지

!pip install selenium

웹자동화를 시키기 위한 크롬드라이버를 먼저 설치해야함
크롬 실행후 도움말에서 Chrome정보 클릭후 버전확인 후 다운받기
(http://chromedriver.chromium.org/downloads)

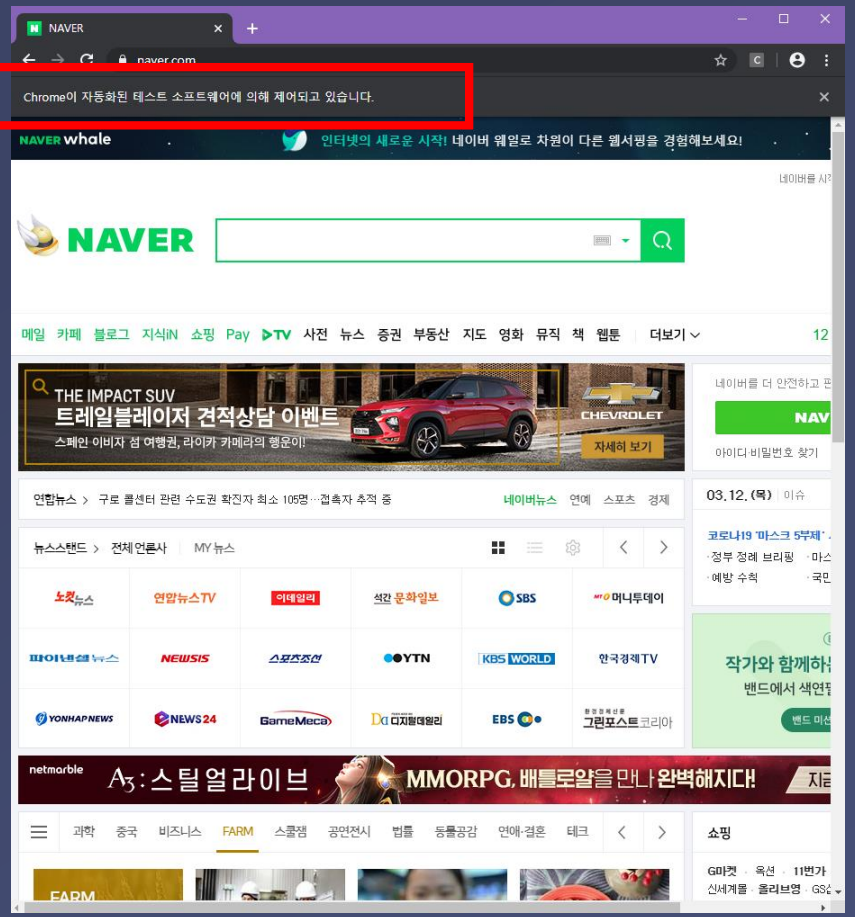
04 Selenium

클릭을해서 정보를 추가로 수집할 수 있음



04 Selenium

```
In [48]: from selenium import webdriver

In [52]: driver = webdriver.Chrome('./chromedriver')
         driver.get('https://www.naver.com')
         driver.close()
```



실행만 시키면 크롬창이 실행되며 원하는 페이지로 자동으로 이동

※ 웹드라이버는 페이지를 로드, 이동하는데
시간이 걸리기 때문에 성급하게 페이지를 닫기보다는
기다리세요!
(로드까지 1~3초정도의 시간이 소요됨
네트워크환경에 따라 다를 수 있음)

D&A

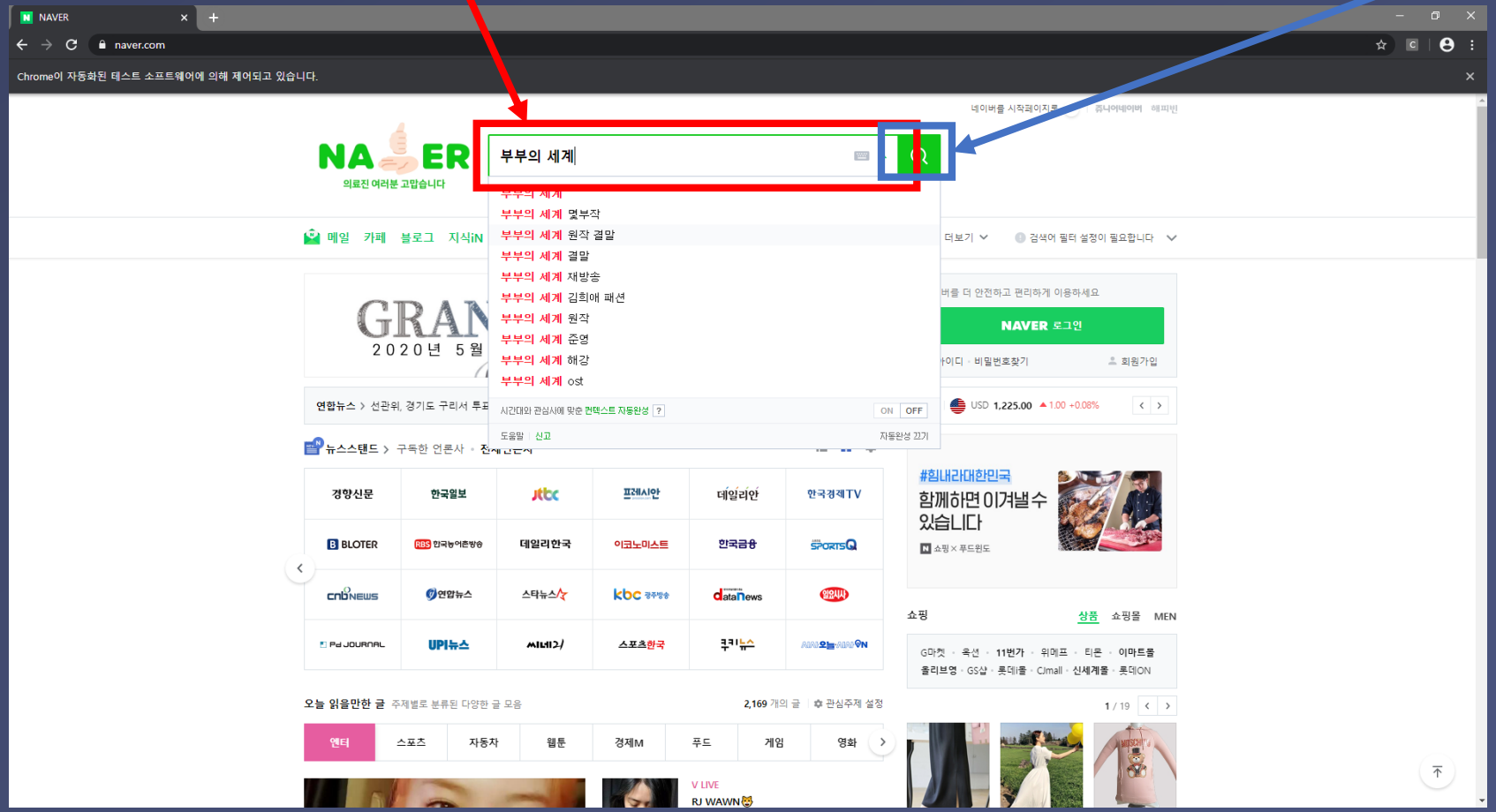
04 Selenium

send_keys() : input 태그에 값을 입력해주는 메소드

```
In [50]: keyword = driver.find_element_by_css_selector('span.green_window input')
         keyword.send_keys('부부의 세계')
```

click() : button 태그를 클릭해주는 메소드

```
In [67]: search = driver.find_element_by_css_selector('button#search_btn')
         search.click()
```



D&A

Requests와 마찬가지로 for문을 사용해 자동화를 시킬 수 있음

```
In [53]: video = driver.find_elements_by_css_selector('div.formula_video li')
         for v in video:
             title = v.find_element_by_css_selector('strong a').text
             date = v.find_element_by_css_selector('div.bt_desc span.update').text
             view = v.find_element_by_css_selector('div.bt_desc span.play_count').text
             print(title,date,view)

[Jtalk 인터뷰_박해준 편] 박해준이 생각하는 〈부부의 세계〉 '이태오' 2020.05.11. 재생수
9,150
[15회 예고] 망상에 빠진 건 그 여자가 아니라, 나였어… 2020.05.09. 재생수
2,070,949
[엔딩] 모든 진실을 폭로한 김희애에 충격받은 한소희 2020.05.09. 재생수
1,908,645
```

※ 간혹 나는 잘 찾았는데 선택자를 찾을 수 없다는 에러가 발생
대표원인 : 대부분 드라이버가 실행되는 시간 보다
파이썬의 코드가 읽히는 시간이 더 짧기 때문
해결방안 :

```
import time
time.sleep(5)
```

데이터가 없을 때 오류가 발생할 수 있음

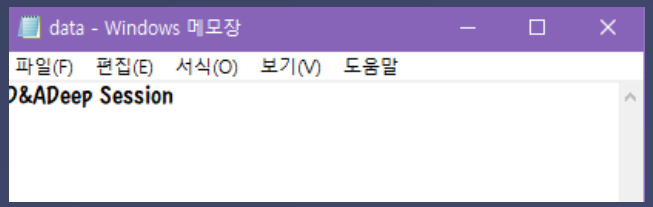
```
In [79]: video = driver.find_elements_by_css_selector('div.formula_video li')
         for v in video:
             title = v.find_element_by_css_selector('strong a').text
             date = v.find_element_by_css_selector('div.bt_desc span.update').text
             try:
                 view = v.find_element_by_css_selector('div.bt_desc span.play_count').text
             except:
                 view = '정보없음'
             print(title,date,view)
```

=> 예외처리(try, except)를 사용하여 에러 해결

05 데이터 저장하기

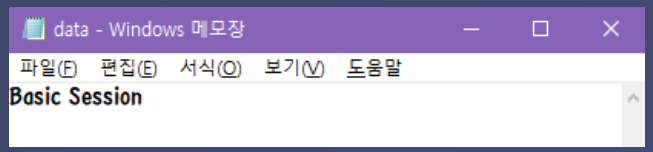
'w' : 쓰기모드
'a' : 추가모드
'r' : 읽기모드

```
In [80]: f = open('data.txt','w')
         f.write('D&A')
         f.write('Deep Session')
         f.close()
```



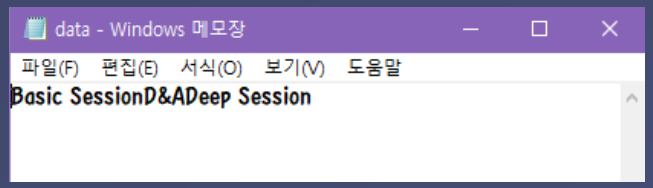
write를 사용하면 원래 데이터에 이어서 작성됨

```
In [89]: f = open('data.txt','w')
         f.write('Basic Session')
         f.close()
```



다시 데이터를 오픈해서 write를 하면 이전 데이터는 사라지고 덮어쓰기 함

```
In [90]: f = open('data.txt','a')
         f.write('D&A')
         f.write('Deep Session')
         f.close()
```



a 모드를 사용해야 내용을 추가할 수 있음

```
In [91]: f = open('data.txt','r')
         data = f.read()
         print(data)
         f.close()
Basic SessionD&ADeep Session
```

읽기모드는 쓰거나 추가할 수는 없음

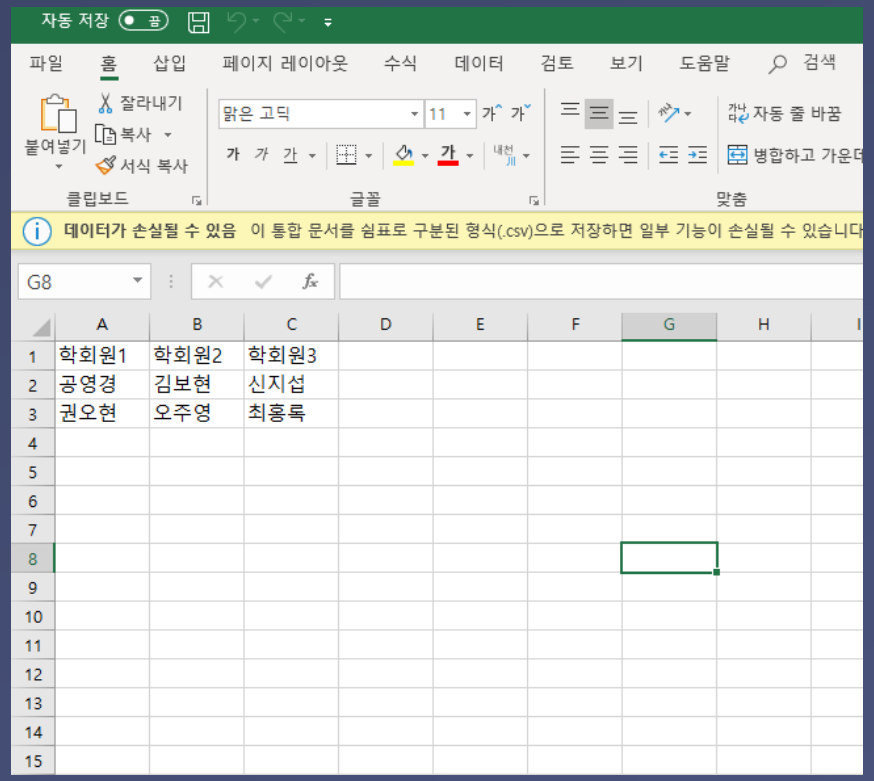
D&A

05 데이터 저장하기

In [103]:
```
f = open('data.csv','w')
f.write('학회원1,학회원2,학회원3')
f.write('₩n공영경,김보현,신지섭')
f.write('₩n권오현,오주영,최홍록')
f.close()
```

In [104]:
```
f = open('data.csv','r')
data = f.read()
print(data)
f.close()
```

```
학회원1,학회원2,학회원3
공영경,김보현,신지섭
권오현,오주영,최홍록
```

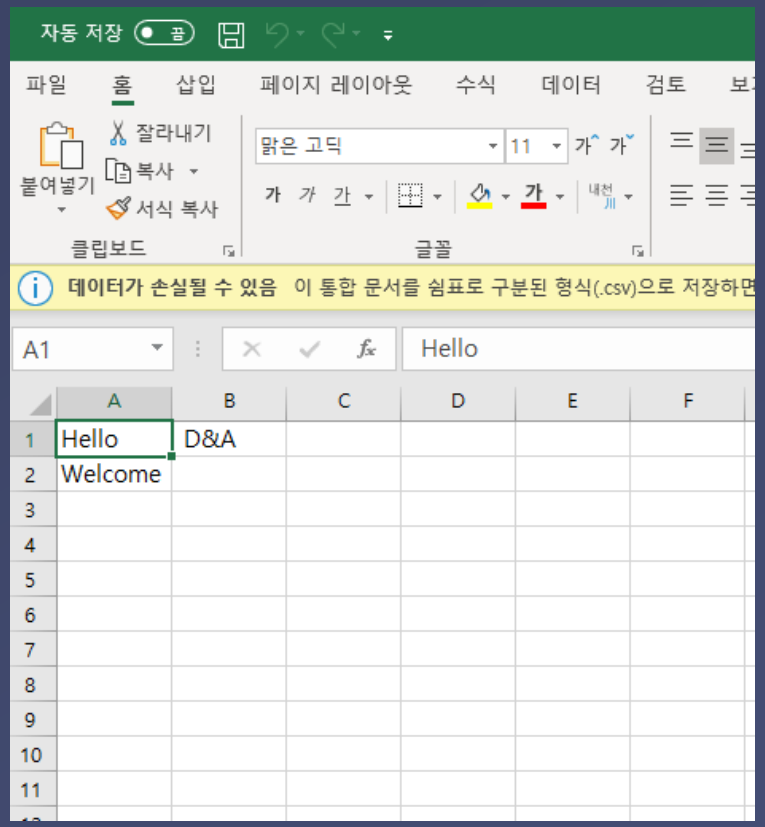


D&A

05 데이터 저장하기

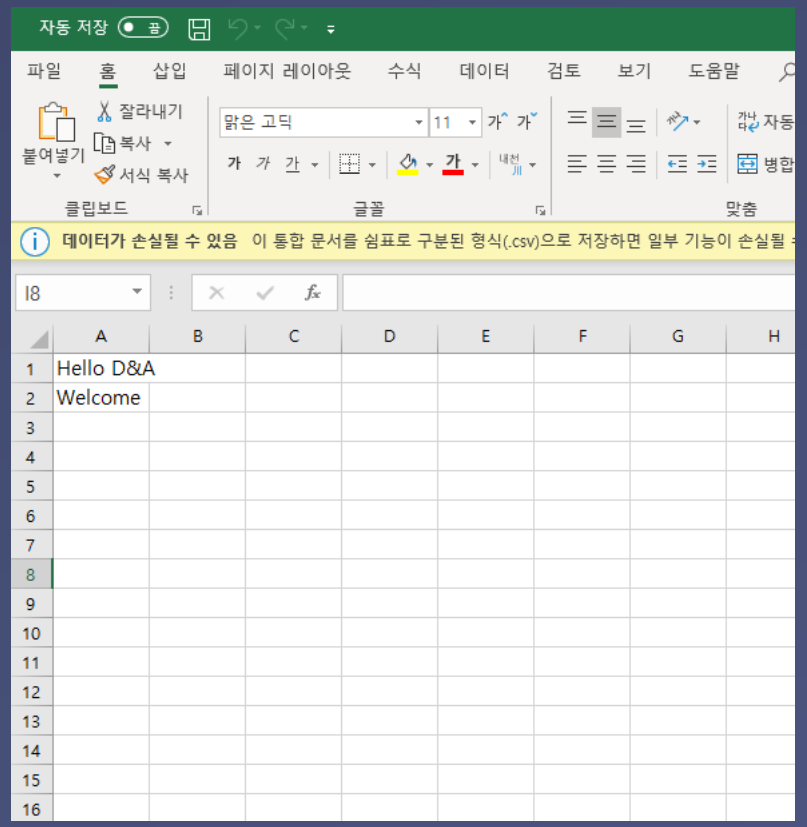
```
In [108]: a1 = 'Hello, D&A'
          a2 = '\nWelcome'
          f = open('data.csv','w')
          f.write(a1)
          f.write(a2)
          f.close()
```

csv로 저장하면 콤마때문에 a1 셀이 나눠지게 된다.



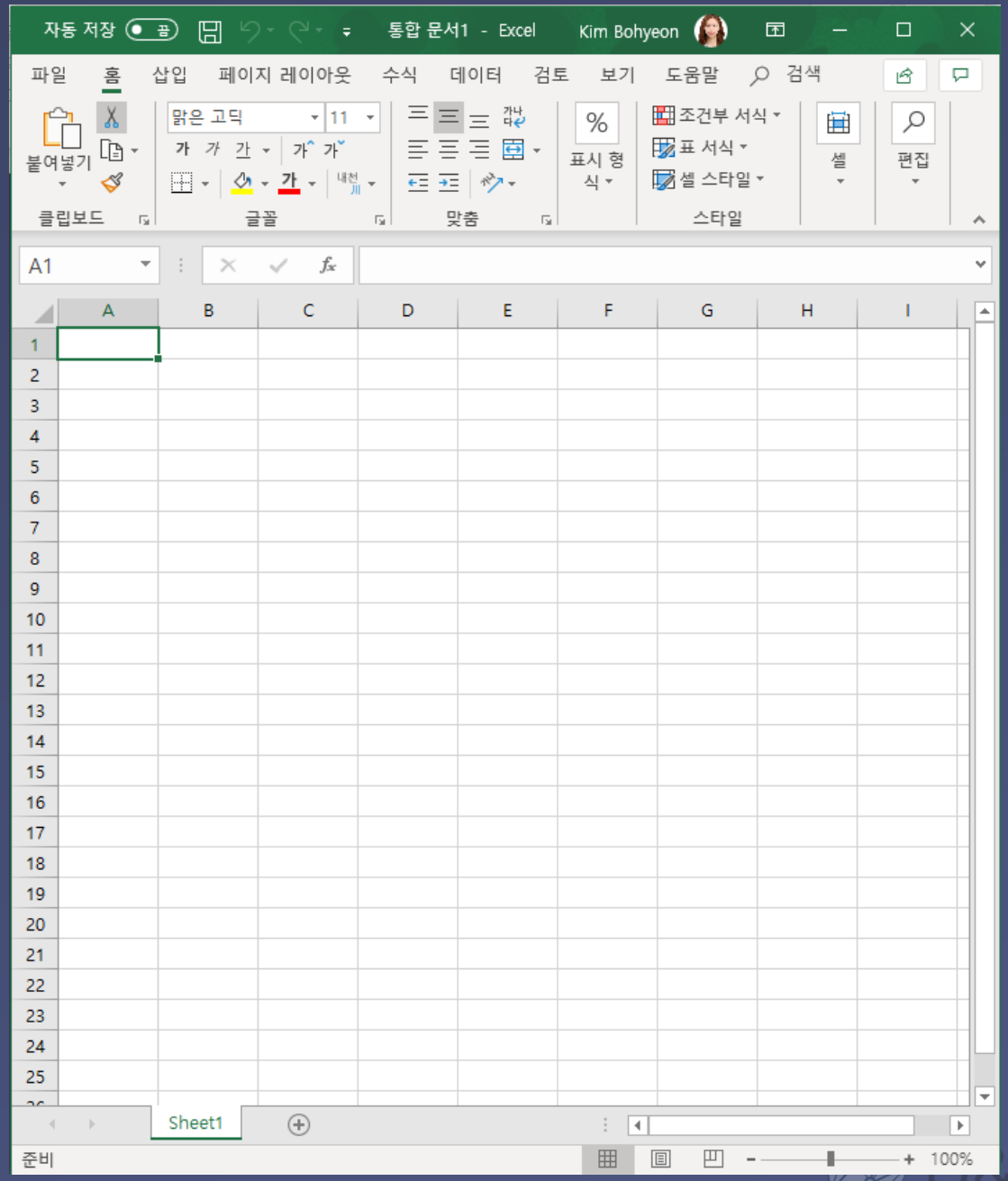

```
In [110]: a1 = 'Hello, D&A'
          a2 = '\nWelcome'
          f = open('data.csv','w')
          f.write(a1.replace(',',''))
          f.write(a2)
          f.close()
```

replace를 통해 교체를 해줘야함



05 데이터 저장하기

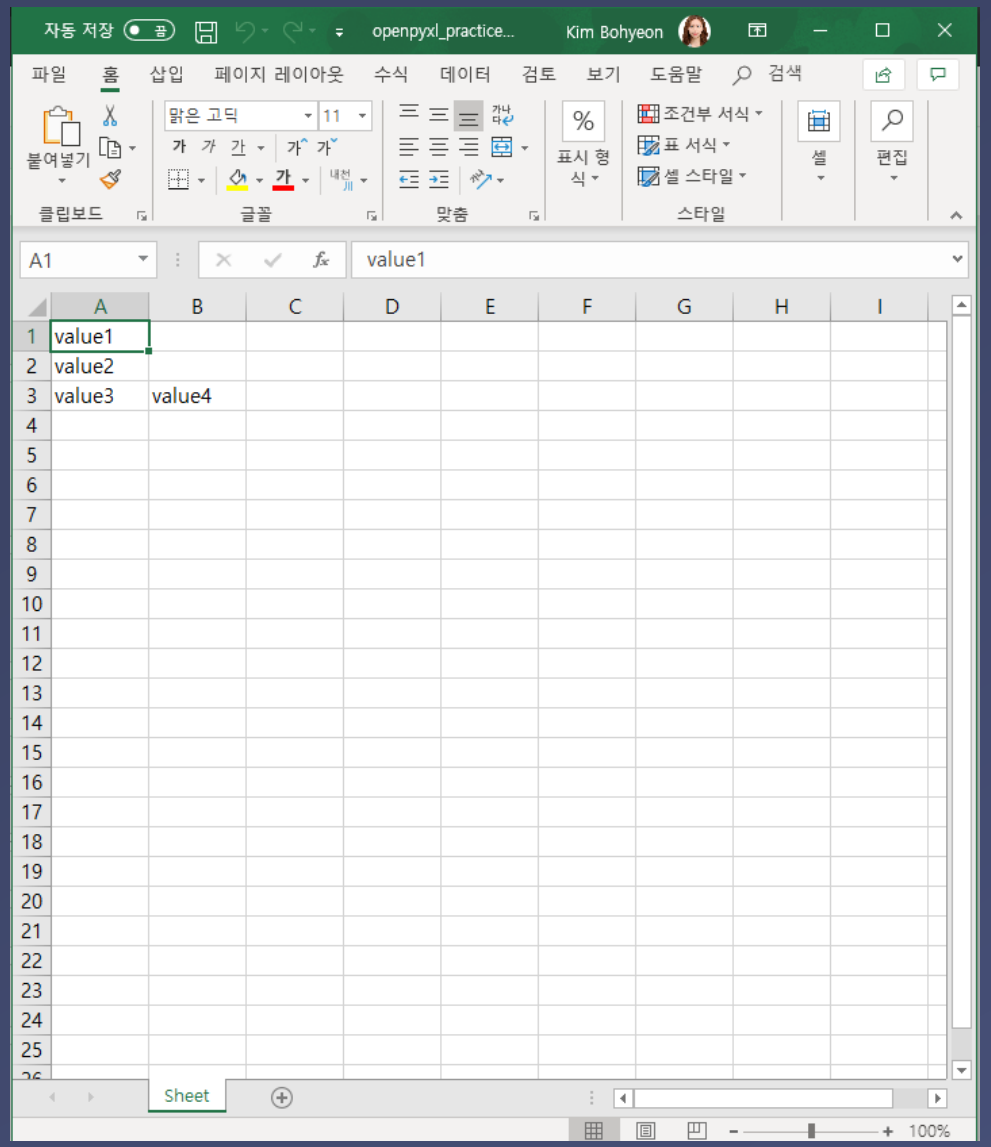
```
In [112]: import openpyxl

          wb = openpyxl.Workbook()
          sheet = wb.active
```



openpyxl.Workbook()
새로운 엑셀 워크북을 열기

wb.active
엑셀 워크시트를 활성화시켜
데이터를 넣을 수 있게 함

05 데이터 저장하기



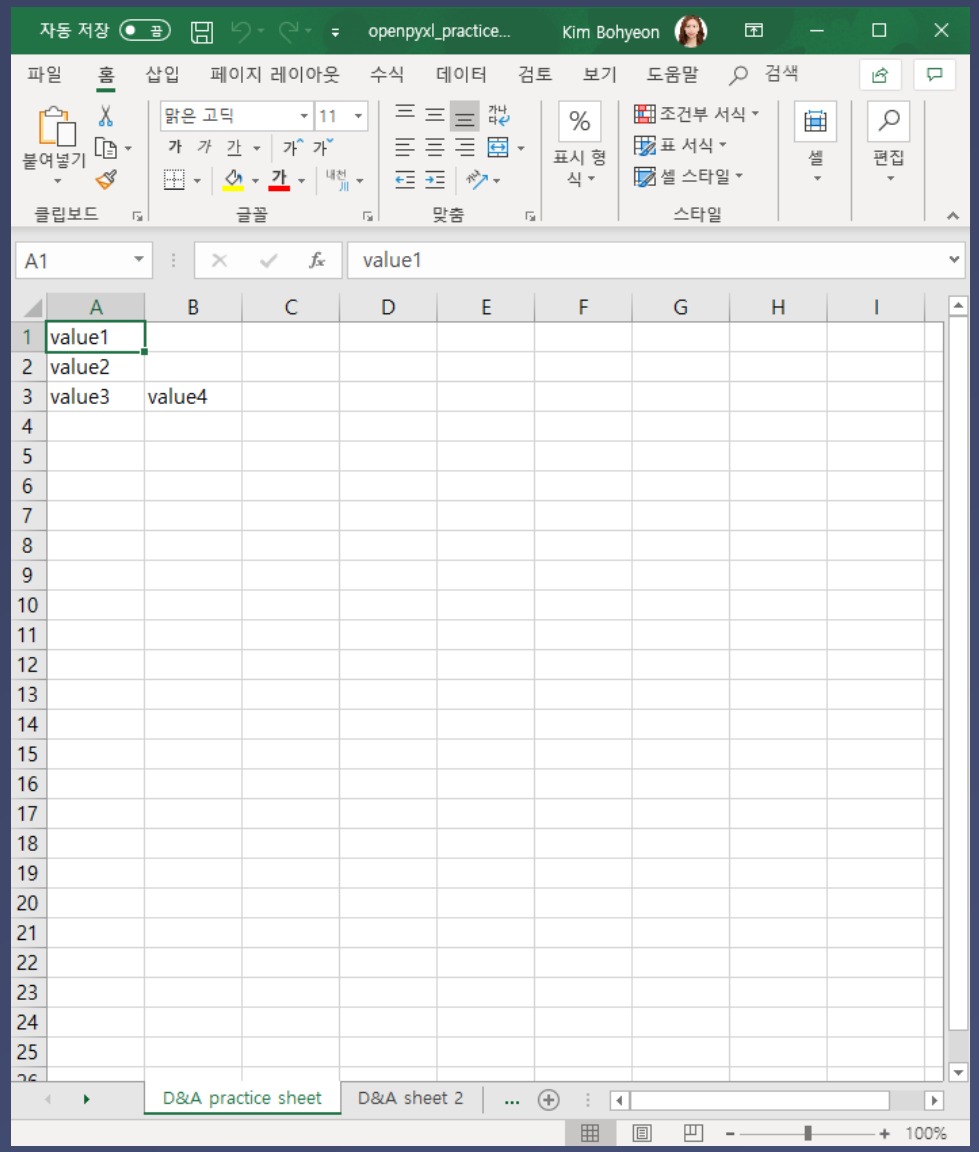

```
In [132]: sheet['A1'] = 'value1'

In [133]: sheet.cell(row=2,column=1).value = 'value2'

In [134]: sheet.append(['value3','value4'])
```

값을 채우는 여러 가지 방법

sheet['셀이름'] = 값
sheet.cell(row=n,column=n).value = 값
sheet.append(['값'])

D&A

05 데이터 저장하기




```
In [138]: sheet.title = 'D&A practice sheet'

In [139]: sheet2 = wb.create_sheet('D&A sheet 2')
          wb.save('openpyxl_practice.xlsx')
```

시트 이름 변경도 가능

sheet.title : 기존의 시트 이름을 변경
create_sheet('값') : 새로운 시트 이름 지정

D&A

05 데이터 저장하기

```
In [121]: try:
              wb = openpyxl.load_workbook('D&A.xlsx')
              sheet = wb.active
          except:
              wb = openpyxl.Workbook()
              sheet = wb.active
```

openxl.load_workbook('파일이름')
이미 존재하는 파일을 불러오기
없으면 에러가 발생

=> 예외처리로 해결

D&A

과제

네이버 쇼핑에서 원하는 상품을 검색하고, 500개 이상의 데이터를 수집하는 코드를 작성하고, csv파일과 xlsx파일로 저장하시오

- 제품사진, 제품명, 가격, 리뷰수, 구매건수, 찜한수, 사이트 링크를 모두 포함해야함
- 제품사진은 폴더를 따로 만들어 저장한 후 캡처본으로 제출
- 패키지 이용에는 관여하지 않음
- 제일 처음 받는 웹페이지의 주소는 네이버 쇼핑의 첫 화면이어야 함
- 제출 목록 : ipynb파일, csv파일, excel파일, 사진저장 폴더의 캡처본