DEEP SESSION

#3 Crawling

DATA&ANALYTICS

0 지난시간 과제 리뷰

Cifar10 data로 만든 basic model에 대해
새로운 activation function, optimizer, dropout,
batch normalization, weight initialization을 사용해
학습 성능 올려 보기

평균적인 ACC > 50%

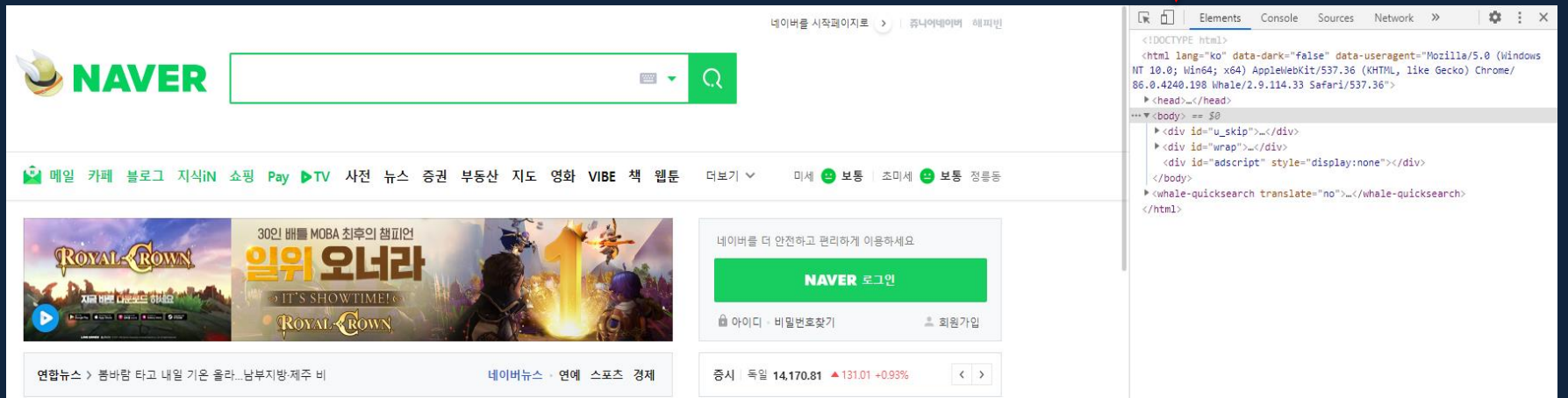
INDEX

1 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!
Or 마우스 오른쪽 클릭 후 검사



1 크롤링 예시

예시 : 지난 학회(2020) 크롤링 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

2 HTML의 구조

일반적으로 태그는 <html>(시작태그) …contents… </html>(종료태그) 구조

ex) <h1>D&A</h1>

-> h1태그에는 D&A라는 정보가 들어있음 !

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

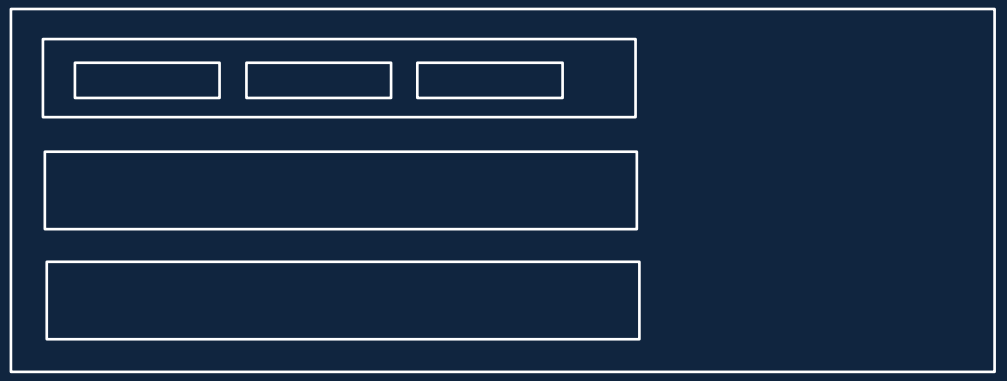
태그	설명	예	속성
h1, h2, h3, h4, h5, h6	글자의 크기를 조절	D&A D&A 2020 D&A	
b	굵게	D&A D&A	
u	밑줄	<u>D&A</u> D&A	
br	줄바꿈	 D&A</br> D&A	
a	링크 연결	네이버 네이버	href(연결할링크), title(확인문구)
span	문서 요소를 묶음	딥세션 딥세션	
li	목록의 리스트	2020 2020	
div	레이아웃		
ul	순서가 없는 목록		
ol	순서가 있는 목록		
button	클릭할 버튼 생성	<button>버튼</button>	type(종류)
p	한 단락	Deep Session 화이팅 Deep Session 화이팅	
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)

2 HTML의 구조

※ 태그는 대부분 class나 id로 구분함!!

똑같은 태그가 엄청 많음

```
<!doctype html>
<html lang="ko" class="os_window">
▶<head>…</head>
▼<body style>
  ▶<div id="daumIndex" class="d_index">…</div>
  ▼<div id="daumWrap">
    ▶<header id="daumHead" class="head_daum">…</header>
     <div id="tierContWrap" class="wrap_tiercont"></div>
    ▶<div id="adLeft" class="wrap_tiercont" style="z-index:auto">…</div>
     <hr class="hide">
    ▼<main id="daumContent">
      ▼<div id="cMain" class="cont_main">
        ▼<article id="mArticle" class="wrap_main">
          ▼<div class="feature_tmp">
            ▶<div id="adMain" class="advert_tmp">…</div>
            ▶<div class="bg_login login_tmp #loginbox">…</div>
             </div>
```



id : 하나의 웹페이지에 하나만 쓸 수 있는 고유한 이름

class : 비슷한 형태를 가진 요소에는 여러번 사용할 수 있는 이름

id : #을 사용해 정보 찾기

class : .을 사용해 정보 찾기

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

자식 관계 : 바로 한단계 아래 하위태그 > 사용

자손 관계 : 모든 하위태그. 띄어쓰기 사용

선택자 >* 하면 모든 자식선택자 선택

3 Requests와 BeautifulSoup4

Requests

웹상의 데이터를 가져올 수 있는 패키지

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

BeautifulSoup4

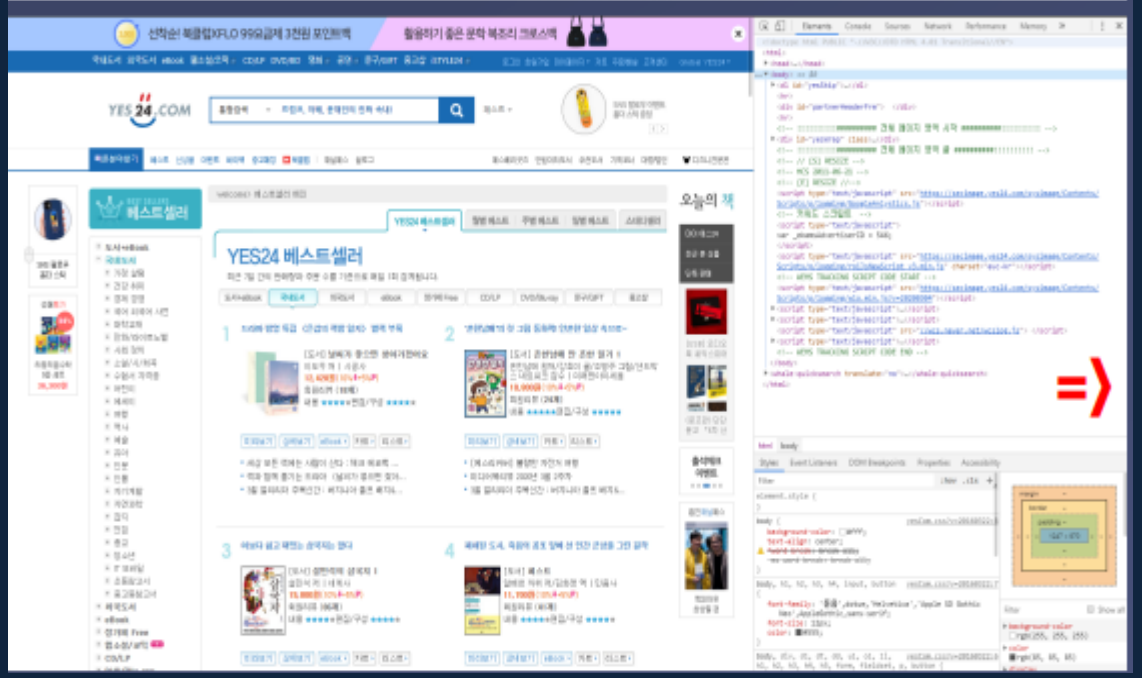
html코드를 파싱해 원하는 데이터를 추출할 수 있는 패키지

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

3 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<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
```

3 Requests와 BeautifulSoup4

```
soup = BeautifulSoup(raw.text, 'html.parser')
print(soup)
```

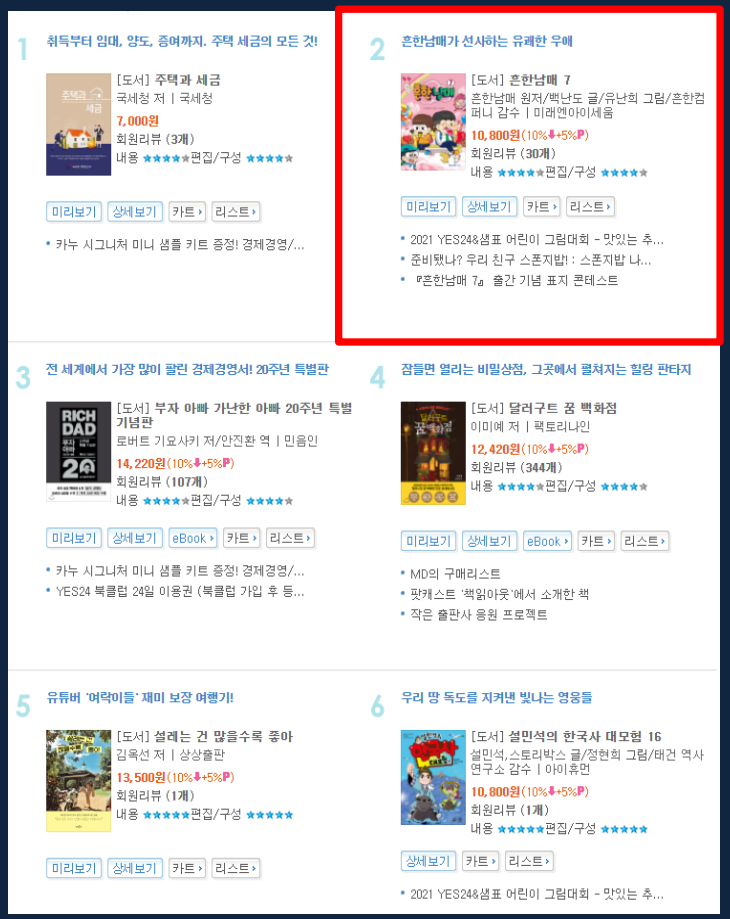
BeautifulSoup(웹페이지의 소스코드, 'html.parser')
웹페이지의 소드코드를 html단위로 구분해줌

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

<html>
<head><base href="http://www.yes24.com/24/"/>
<meta content="IE=Edge" http-equiv="X-UA-Compatible"/>
<meta content="text/html;charset=utf-8" http-equiv="Content-Type"/>
<meta content="dpr, width, viewport-width, rtt, downlink, ect, UA, UA-Platform, UA-Arch, UA-Model, UA-Mobile, UA-Full-Version" htt
p-equiv="Accept-CH"/>
<meta content="86400" http-equiv="Accept-CH-Lifetime"/>
<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,
공연, 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">
```

3 Requests와 BeautifulSoup4

yes24
베스트셀러 정보 크롤링



※ 페이지 검사를 통해 필요한
정보가 다 담긴 컨테이너를 찾아야 함

1. 구조를 파악

2. 필요한 정보가 어떤 태그에
담겨져 있는지 확인

3. 크롤러 만들기

3 Requests와 BeautifulSoup4

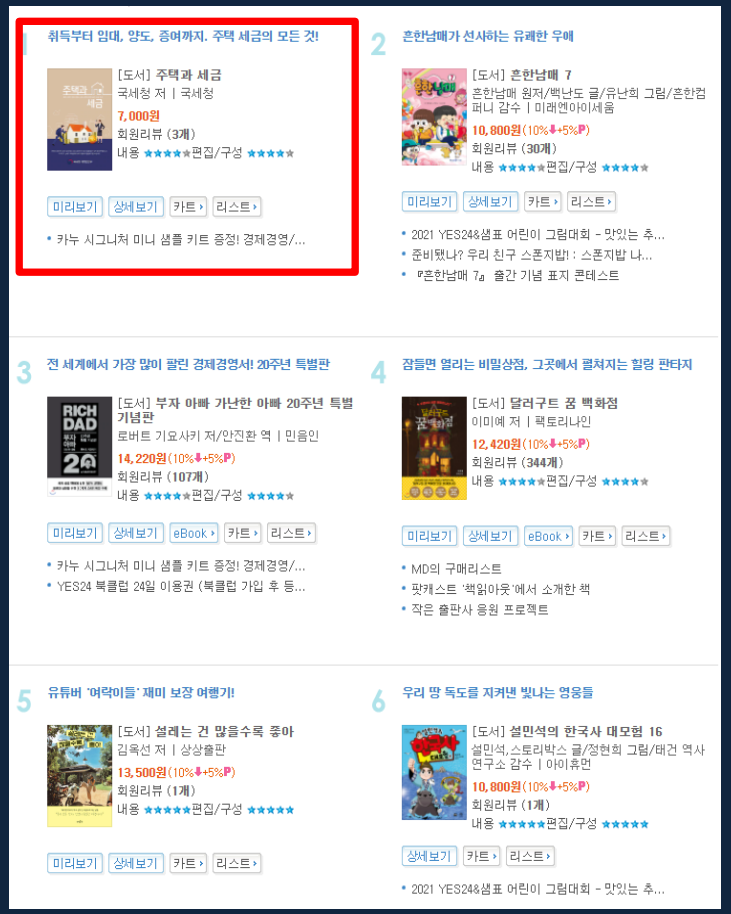
책 정보의 구조 div#bestList ol > li

```
books = soup.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/97784717">취득부터 임대, 양도, 증여까지. 주택 세금의 모든 것!</a></p>
<p class="image" id="location_0">
<a href="/Product/Goods/97784717">
<img alt="주택과 세금" src="http://image.yes24.com/goods/97784717/L">
</img></a>
</p>
<p>[도서] <a href="/Product/Goods/97784717">주택과 세금</a></p>
<p class="aupu"><a href="http://www.yes24.com//SearchCorner/Result?domain=ALL&amp;author_yn=Y&amp;query=&amp;auth_no=299299" target="_blank">국세청</a> 저 | <a href="http://www.yes24.com//SearchCorner/Result?domain=ALL&amp;company_yn=Y&amp;query=국세청">국세청</a></p>
<p class="price"><strong>7,000원</strong></p>
<p>회원리뷰 (<a href="/Product/Goods/97784717#Review">3개</a>)</p>
<p>
                내용 <img align="absmiddle" src="http://image.yes24.com/sysimage/interface/smallStarOn.gif"/><img align="absmiddle" src="http://image.yes24.com/sysimage/interface/smallStarOn.gif"/><img align="absmiddle" src="http://image.yes24.com/sysimage/interface/smallStarOn.gif"/><img align="absmiddle" src="http://
```

3 Requests와 BeautifulSoup4




```
# 첫번째 책 정보
books[0]
```

책 서브제목 / 가격 불러오기

select_one : html의 태그를 선택. 첫번째 태그 하나만 가져옴

```
# 서브 제목
sub_title = books[0].select_one('p.copy').text
print(sub_title)
```

```
취득부터 임대, 양도, 증여까지. 주택 세금의 모든 것!
```

```
# 가격
price = books[0].select_one('p.price').text
print(price)
```

```
7,000원
```

3 Requests와 BeautifulSoup4

```
for book in books:
    sub_title = book.select_one('p.copy').text
    price = book.select_one('p.price').text
    print(sub_title, price)
```

```
취득부터 임대, 양도, 증여까지. 주택 세금의 모든 것! 7,000원
흔한남매가 선사하는 유쾌한 우애 10,800원(10%+5%)
전 세계에서 가장 많이 팔린 경제경영서! 20주년 특별판 14,220원(10%+5%)
잠들면 열리는 비밀상점, 그곳에서 펼쳐지는 힐링 판타지 12,420원(10%+5%)
유튜버 '여락이들' 재미 보장 여행기!  13,500원(10%+5%)
우리 땅 독도를 지켜낸 빛나는 영웅들 10,800원(10%+5%)
주린이들을 위한 안전판! 염블리표 투자 바이블 16,200원(10%+5%)
국민 육아멘토 오은영 박사의 현실밀착 육아회화 15,750원(10%+5%)
돌아온 피터슨, 다시 인생 법칙을 말하다  16,020원(10%+5%)
비상! 나무 집에 '미확인 비행 눈알' 이 나타났다! 10,800원(10%+5%)
유서 깊은 역사와 찬란한 문화! 한국으로 GO GO! 10,800원(10%+5%)
전천당에 행운의 손님이 등장했다! 10,800원(10%+5%)
'인더숲 BTS편'에서 슈가, RM이 읽은 그 책! 10,800원(10%+5%)
드디어 카이도와의 전력 대전!  4,500원(10%+5%)
10년 후, 지금보다 더 거대한 변화가 우리 앞에 온다 16,200원(10%+5%)
2021년 한국사능력검정시험 준비는 큰별쌤과 함께 17,100원(10%+5%)
수학이 어려운 엄마를 위한 전략적 학습 로드맵! 15,120원(10%+5%)
일론 머스크가 자문하는 두뇌 전문가 '짐 퀵'의 책! 15,120원(10%+5%)
박완서 작가 10주기 에세이  14,400원(10%+5%)
'사랑할 때 알아야 할 것들' 김재식 작가의 관계 처방전  13,320원(10%+5%)
```

for문을 통해
여러 개의 데이터를
한 번에 수집할 수 있음!

3 Requests와 BeautifulSoup4

책 정보에서 url / 책 이미지 불러오기

책 정보를 클릭해서 넘어가지게 되는 url 가져오기

```
# 주소,  attrs는 attributions(속성)의 약자
books[0].select_one('p.copy a').attrs['href']
```

```
'/Product/Goods/96644794'
```

특정 태그에 속해 있는 href태그를 불러오면
연결된 페이지의 정보가 불러와짐

```
# 얻은 주소로 재접속
temp_req = requests.get('http://www.yes24.com' + books[0].select_one('p.copy a')

temp_html = BeautifulSoup(temp_req.text, 'html.parser')

temp_title = temp_html.select_one('h2.gd_name').text
print(temp_title)
```

```
주택과 세금
```

이미지 정보를 가져올 때

```
from urllib.request import urlretrieve
```

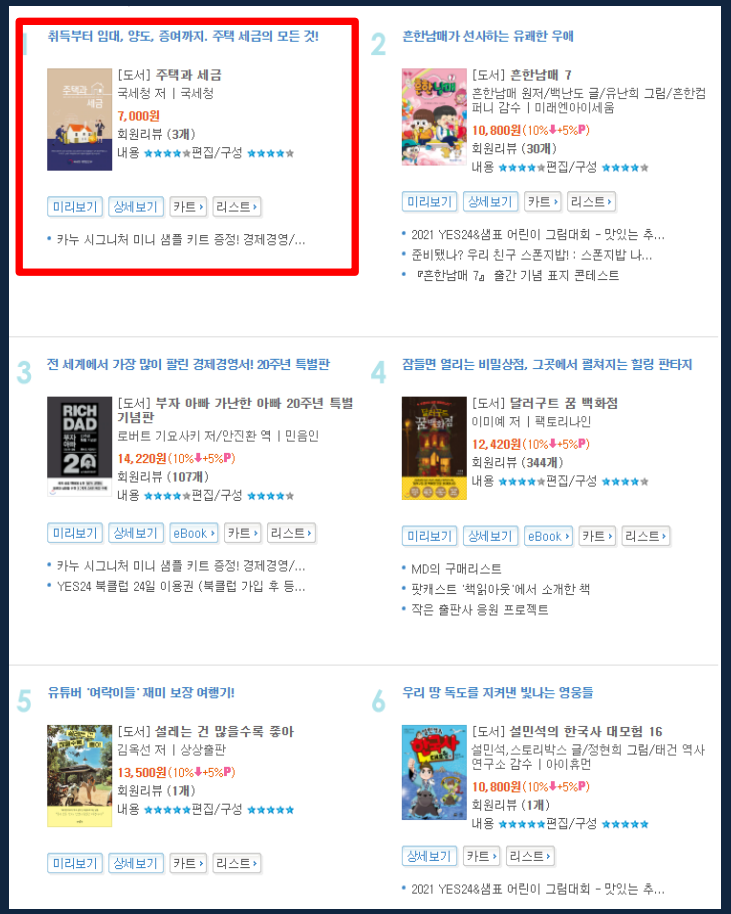
```
img = books[0].select_one('p.image a img').attrs['src']
print(img)
```

```
http://image.yes24.com/goods/97784717/L
```

```
urlretrieve(img, '책_1.png')
```

src로 불러와짐

3 Requests와 BeautifulSoup4



책 서브제목 / 가격 불러오기

책 url / 이미지 불러오기

우리가 원하는 것은 완전 자동화 !

1. 검색 자동화

2. 페이지 넘기기 자동화

3 Requests와 BeautifulSoup4

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

※ 1. 검색 자동화

검색하기

```
keyword = 'python'
```

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

```
<Response [200]>
```

for문을 통해 자동화 가능

```
keywords = ['python', 'deep learning', 'sql']
```

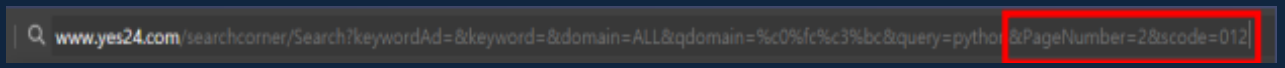
```
for keyword in keywords:
    print(f'------------------ {keyword} ------------------')
    req = requests.get('http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%C0%FC%C3%BC&query=' + keyw
    soup = BeautifulSoup(req.text, 'html.parser')
    books = soup.select('td.goods_infogrp')
    for book in books[:20]:
        title = book.select_one('p.goods_name a').text
        writer = book.select_one('div.goods_info a').text
        price = book.select_one('div.goods_price em').text
        print(f'{title}, {writer}, {price}')
```

```
------------------ python ------------------
Do it! 점프 투 파이썬, 박응용, 16,920
혼자 공부하는 파이썬, 윤인성, 16,200
혼자 공부하는 머신러닝+딥러닝, 박해선, 23,400
파이썬 증권 데이터 분석 , 김황후, 28,800
이것이 취업을 위한 코딩 테스트다 with 파이썬 , 나동빈, 30,600
두근두근 파이썬, 천인국, 24,000
밑바닥부터 시작하는 딥러닝 , 사이토 고키, 21,600
Do it! 파이썬 생활 프로그래밍, 김창현, 18,000
Do it! 점프 투 파이썬 + Do it! 파이썬 생활 프로그래밍, 박응용, 34,920
핸즈온 머신러닝, 오렐리앙 제롱, 49,500
선형대수와 통계학으로 배우는 머신러닝 with 파이썬, 장철원, 33,750
```

3 Requests와 BeautifulSoup4

※ 2. 페이지 넘기기 자동화

&PageNumber=2



아까와 마찬가지로 페이지 주소를 살피면 답이 나옴!

```
f'http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%c0%fc%c3%bc&query={keyword}&PageNumber={page_num}
```

```
# 검색하고 페이지를 넘어가면서 끌어온다
for keyword in keywords:
    print(f'₩n------------------ {keyword} ------------------')
    for page_num in range(1, 2):
        req = requests.get(
            f'http://www.yes24.com/searchcorner/Search?keywordAd=&keyword=&domain=ALL&qdomain=%c0%fc%c3%bc&query={keyword}&PageNumb
        soup = BeautifulSoup(req.text, 'html.parser')
        books = soup.select('td.goods_infogrp')[:20]
        for book in books:
            title = book.select_one('a').text
            writer = book.select_one('div.goods_info a').text
            price = book.select_one('div.goods_price em').text
            print(f'{title}, {writer}, {price}')
```

```
------------------ python ------------------
Do it! 점프 투 파이썬, 박응용, 16,920
혼자 공부하는 파이썬, 윤인성, 16,200
혼자 공부하는 머신러닝+딥러닝, 박해선, 23,400
두근두근 파이썬, 천인국, 22,800
파이썬 for Beginner, 우재남, 23,000
이것이 취업을 위한 코딩 테스트다 with 파이썬 , 나동빈, 30,600
밑바닥부터 시작하는 딥러닝 , 사이토 고키, 21,600
파이썬 증권 데이터 분석 , 김황후, 28,800
파이썬을 이용한 비트코인 자동매매, 조대표, 24,300
핸즈온 머신러닝, 오렐리앙 제롱, 49,500
파이썬 머신러닝 완벽 가이드, 권철민, 34,200
파이썬으로 배우는 컴퓨팅 사고, 김완섭, 17,000
파이썬 알고리즘 인터뷰, 박상길, 34,200
```

3 Requests와 BeautifulSoup4

종종 있는 문제점들

※ 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>
  </span>
```

genre

```
▼<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>
  ",
```

actor

3 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
```

Break Time

Review Time

4 Selenium

※ 정적페이지와 동적페이지?

정적페이지는 웹페이의 화면이
클릭이나 입력을 추가했을 때 웹페이지 주소가 변경되는 것

동적페이지는 웹페이지 주소가 변경되지는 않지만
보이는 웹페이지가 변경되는 페이지

4 Selenium

Selenium

웹드라이버가 웹브라우저를 컨트롤하여 웹을 자동화시켜주는 패키지

!pip install selenium

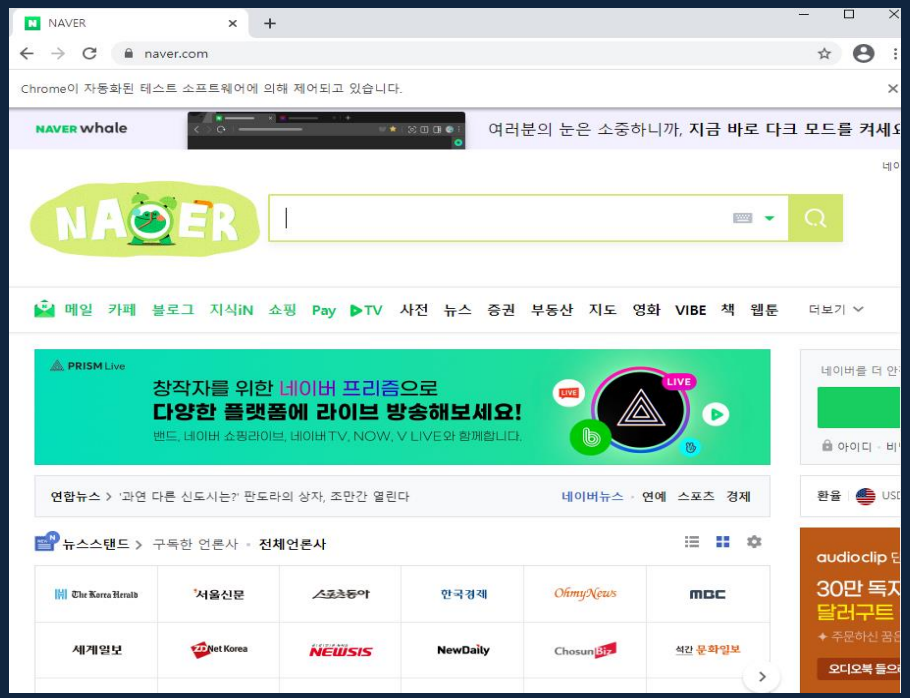
웹자동화를 시키기 위한 크롬드라이버를 먼저 설치해야함
크롬 실행후 도움말에서 Chrome정보 클릭후 버전확인 후 다운받기
다운 받은 파일을 주피터 노트북 작업 파일과 같은 경로에 넣어주기
(http://chromedriver.chromium.org/downloads)

4 Selenium

```
# !pip install selenium
from selenium import webdriver
```

```
# 드라이버 불러오기(크롬으로 빈 페이지가 불러와짐)
dr = webdriver.Chrome('./chromedriver_win32/chromedriver')
```

```
# 원하는 페이지 get메소드로 불러오기
dr.get('https://www.naver.com')
```



실행만 시키면 크롬창이 실행되며
get으로 원하는 페이지로 자동으로 이동

※ 웹드라이버는 페이지를 로드, 이동하는데
시간이 걸리기 때문에 조금 기다리기!
(로드까지 1~3초 정도의 시간이 소요됨
네트워크환경에 따라 다를 수 있음)

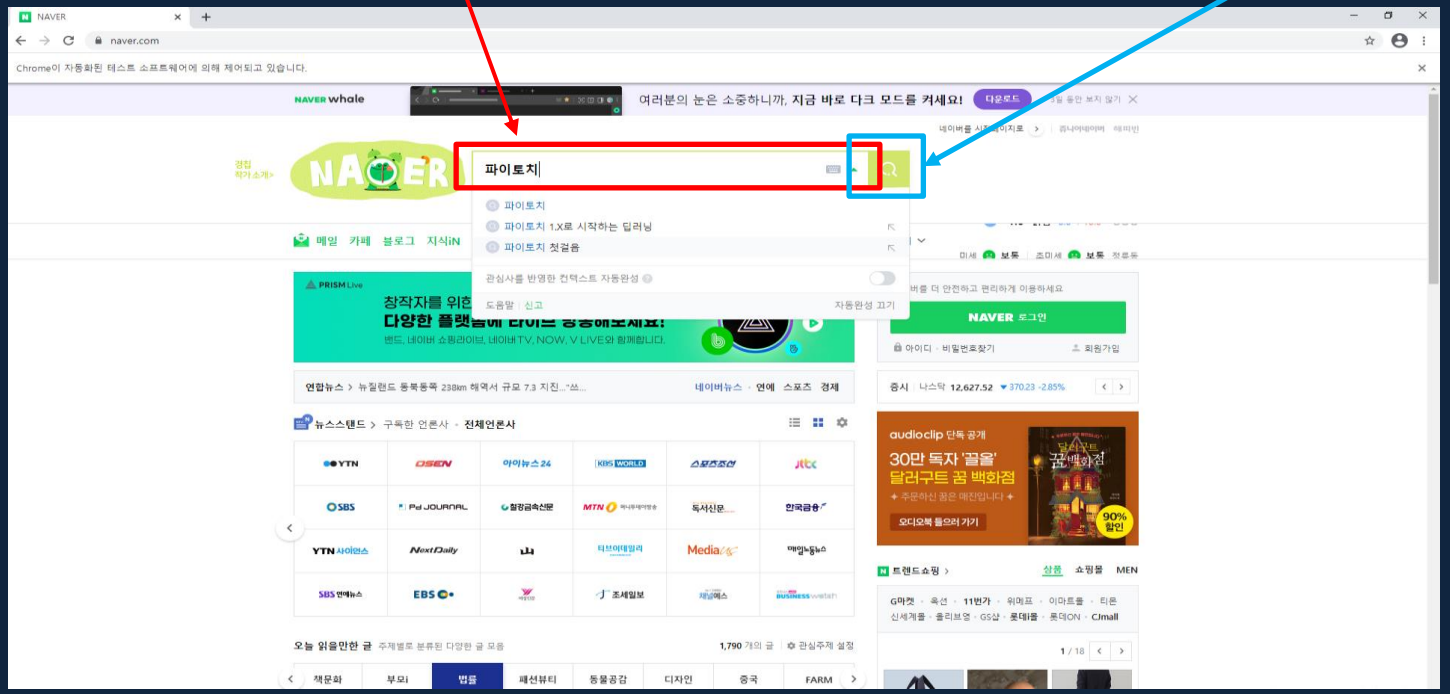
4 Selenium

send_keys() : input 태그에 값을 입력해주는 메소드

```
In [ ]:
# 검색입력
keyword = dr.find_element_by_css_selector('div.green_window input')
keyword.send_keys('파이토치')
```

click() : button 태그를 클릭해주는 메소드

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



4 Selenium

Requests와 마찬가지로 for문을 사용해 자동화를 시킬 수 있음

```
In [38]: books = dr.find_elements_by_css_selector('li.book_group')
for i in range(len(books)):
    title = books[i].find_element_by_css_selector('div.book_info a').text
    writer = books[i].find_element_by_css_selector('div.item_info.type_writer').te
    view = books[i].find_element_by_css_selector('div.item:nth-of-type(3) a').text
    print(f'{title}, {writer}, {view}')
    if (i+1) % 3 == 0:
        next_arrow = dr.find_element_by_css_selector('div.cmm_pgs a:nth-of-type(2)
        next_arrow.click()
```

```
파이썬 딥러닝 파이토치(Python Deep Learning PyTorch), 이경택, 방성수
외 1명, 15건
파이토치 첫걸음 (딥러닝 기초부터 RNN, 오토인코더, GAN 실전 기법까지), 최건호, 3
건
펭귄브로의 3분 딥러닝 파이토치맛 (PyTorch 코드로 맛보는), 김건우, 염상준, 2건
파이토치 1.X로 시작하는 딥러닝 (Deep Learning with PyTorch 1.x Second Edition),
```

books 불러오기

책 정보 불러오기

다음페이지 넘어가기

※ 셀레늄에서 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(2)
```

4 Selenium

데이터가 없을 때 오류가 발생할 수 있음

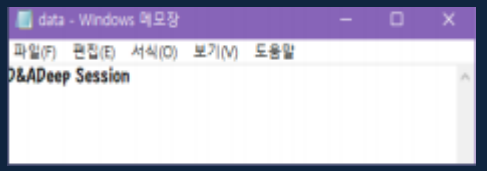
```
In [79]: video = driver.find_elements_by_css_selector('div.formula_video li')
         for v in video:
             title = v.find_element_by_css_selector('strong a').text
             date = v.find_element_by_css_selector('div.bt_desc span.update').text
             try:
                 view =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

5 데이터 저장하기

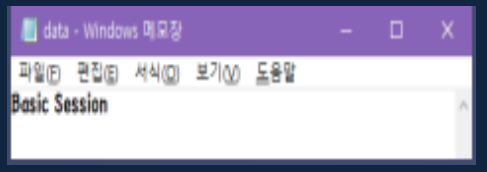
'w' : 쓰기모드
'a' : 추가모드
'r' : 읽기모드

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



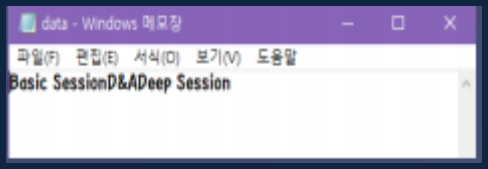
Write를 사용하면 원래 데이터에
이어서 작성됨

```
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()
```



A모드를 사용해야
내용을 추가할 수 있음

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

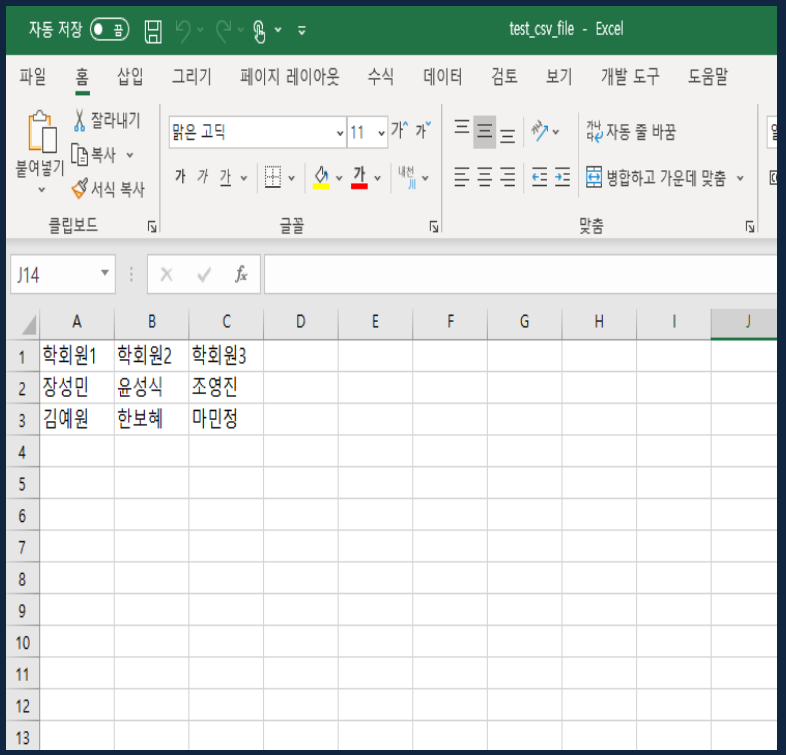
읽기모드는 쓰거나 추가할 수는 없음

5 데이터 저장하기

```
# 쓰기
f = open('test_csv_file.csv', 'w')
f.write('학회원1,학회원2,학회원3') # csv파일이라 쉼표로 셀 구분
f.write('₩n장성민,윤성식,조영진')
f.write('₩n김예원,한보혜,마민정')
f.close()
```

```
# 읽기
f = open('test_csv_file.csv', 'r')
sentence = f.read()
f.close()
print(sentence)
```

```
학회원1,학회원2,학회원3
장성민,윤성식,조영진
김예원,한보혜,마민정
```

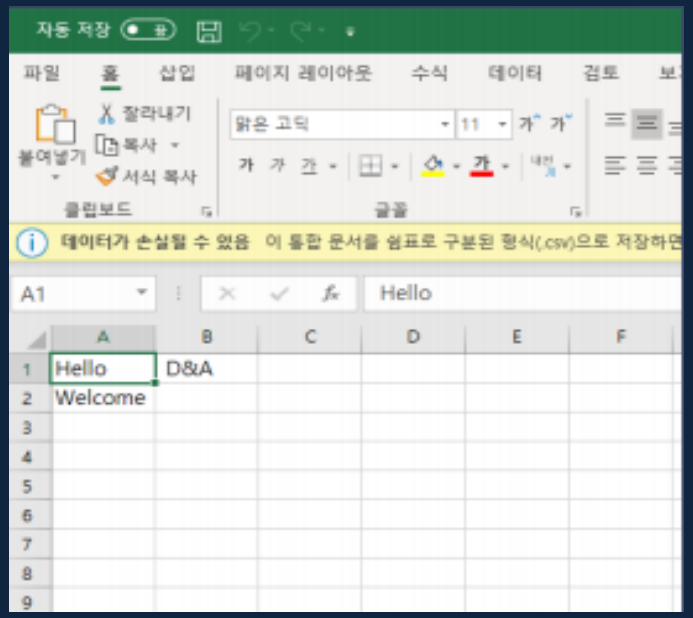


	A	B	C	D	E	F	G	H	I	J
1	학회원1	학회원2	학회원3							
2	장성민	윤성식	조영진							
3	김예원	한보혜	마민정							
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										

5 데이터 저장하기

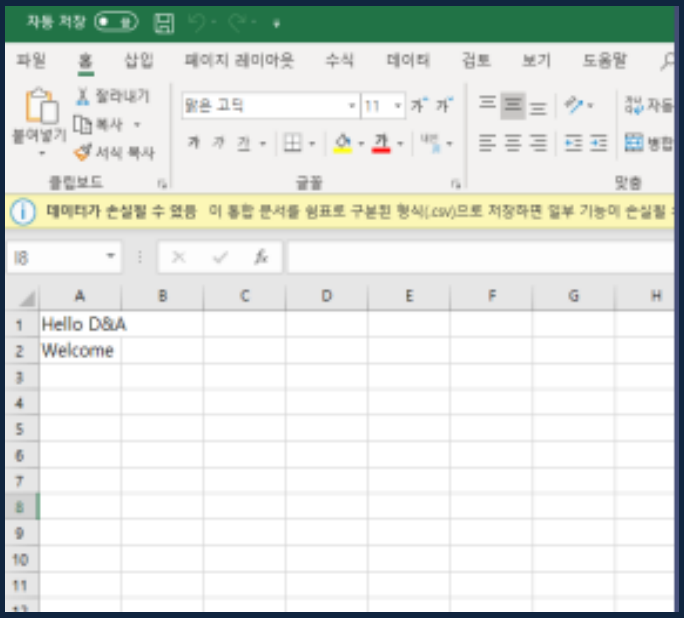
```
In [108]: a1 = 'Hello, D&A'
          a2 = '₩nWelcome'
          f = open('data.csv','w')
          f.write(a1)
          f.write(a2)
          f.close()
```

csv로 저장하면 콤마때문에
A1 셀이 나눠지게 된다.



replace를 통해
교체를 해줘야함

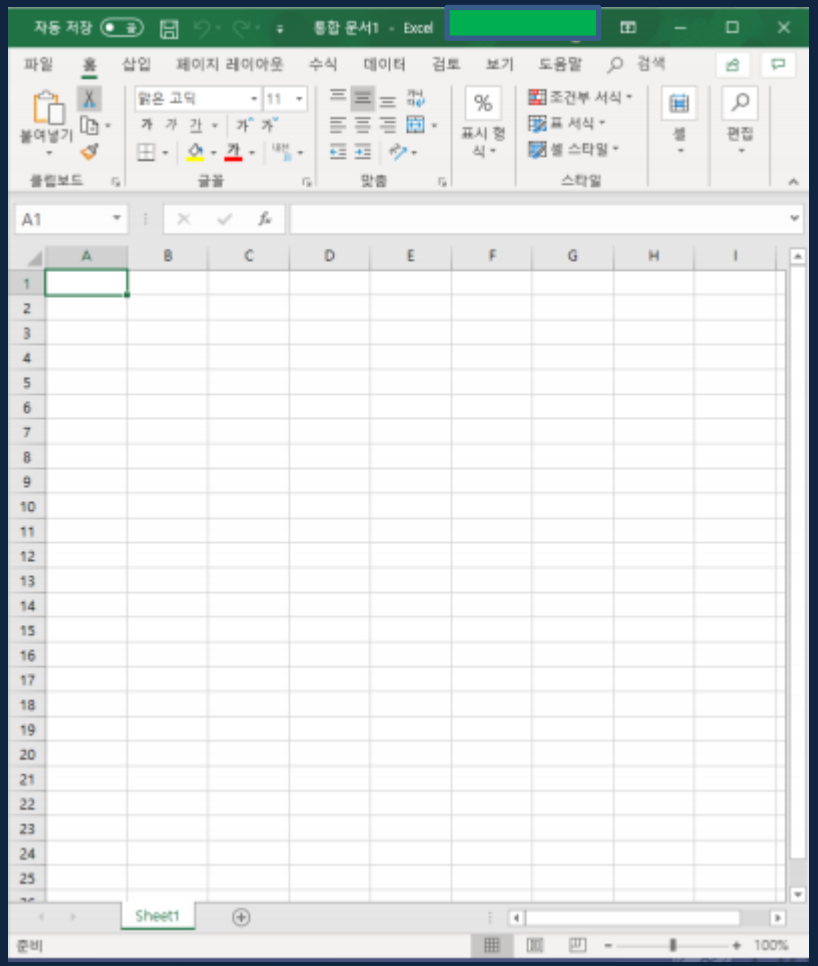
```
In [110]: a1 = 'Hello, D&A'
          a2 = '₩nWelcome'
          f = open('data.csv','w')
          f.write(a1.replace(',',''))
          f.write(a2)
          f.close()
```



5 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

5 데이터 저장하기

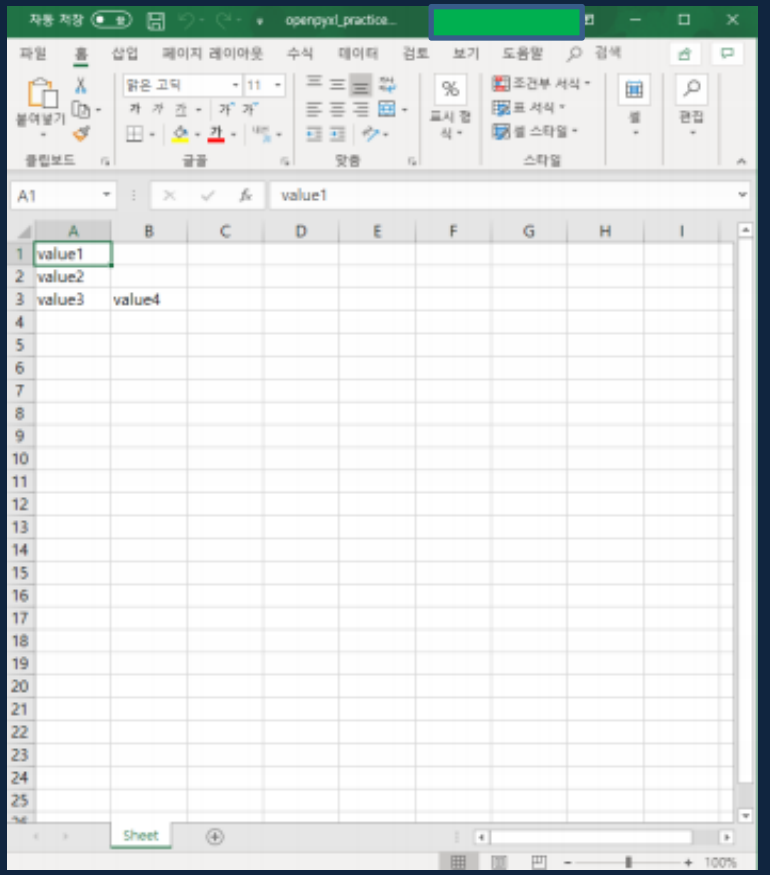
값을 채우는 여러 가지 방법

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

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

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

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



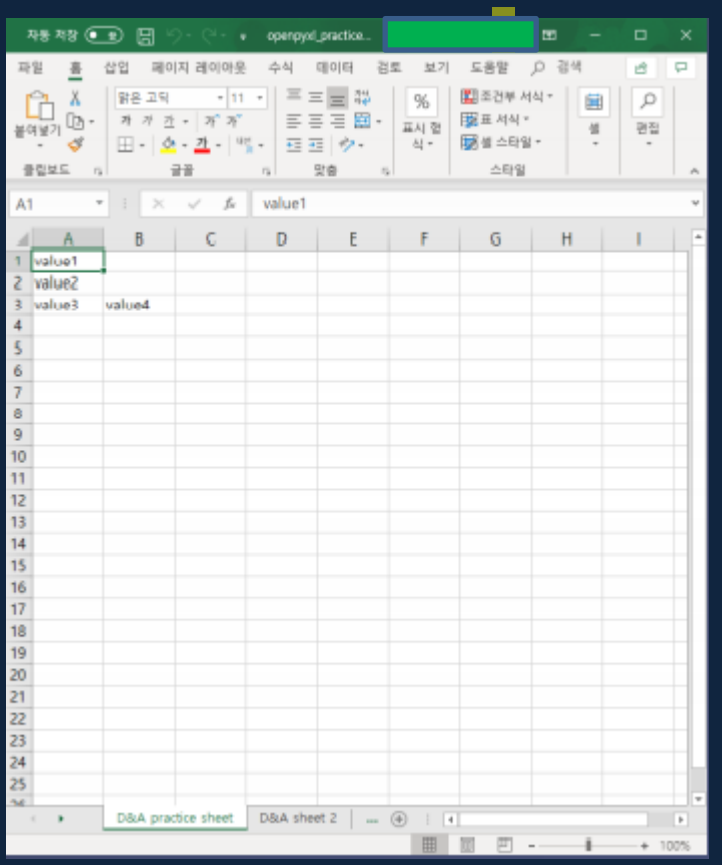
5 데이터 저장하기

시트 이름 변경도 가능

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

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

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



5 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

과제

Yes24에서 도서를 검색하고 200개 이상의 책 정보를
수집하는 코드를 작성하고, csv파일과 xlsx파일로 저장하시오

- 책 사진, 제목, 저자, 가격, 판매지수, 리뷰 수,
  별점 점수, 사이트 링크 모두 포함해야함

- 책 사진은 폴더를 따로 만들어 저장한 후 캡처본으로 제출

- 패키지(Beatifulsoup4, Selenium) 이용에는 관여하지 않음

- 제출 목록: ipynb파일, csv파일, excel파일, 사진저장 폴더의 캡처본
  (압축 X, 4개 파일 올리기)

THANK YOU