숙명여대 학과정보 수집

```
# 필요한 모듈 참조
import requests
from bs4 import BeautifulSoup
from pandas import DataFrame
```

```
# 수집할 컨텐츠가 있는 웹 페이지의 주소
url = "https://www.sookmyung.ac.kr/sookmyungkr/1030/subview.do"
```

```
# 접속객체 생성
session = requests.Session()

# 접속객체에 부가정보(header) 삽입하기
session.headers.update({
    "Referer": "",
    "User-Agent": "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTMl
})

# 생성한 접속객체를 활용하여 API에 접속
r = session.get(url)

# 접속에 실패한 경우
if r.status_code != 200:
    # 에러코드와 에러메시지 출력
    msg = "[%d Error] %s 에러가 발생함" % (r.status_code, r.reason)
    # 에러를 강제로 생성시킴
    raise Exception(msg)

r.encoding = "utf-8"
#print(r.text)
soup = BeautifulSoup(r.text)
soup
```

응답결과로부터 데이터 추출하기

```
# 수집한 정보를 저장할 빈 리스트
학과목록_리스트 = []

college_list = soup.select(".college_list")
#print(college_list)

for item in college_list:
    #print(item)
    #print("-" * 50)

    # 학과이름
```

```
    h5El = item.select("h5")
    name = h5El[0].text.strip()
    print(name)
    #print("-" * 50)

    # 상세보기 URL
    # 가져온 페이지 주소의 예) /sookmyungkr/1012/subview.do
    # 같은 사이트 안에서 이동할 때는 앞부분 "https://www.sookmyung.ac.kr"을 생략 가능하지만
    # 외부에서 직접 접근할 때는 반드시 전체 주소가 필요함.
    # 가져온 값이 전체주소가 아니므로 검사하여 누락된 부분을 덧붙여야 한다.
    viewEl = item.select(".view")
    view = viewEl[0].attrs['href']

    if view.find("https://www.sookmyung.ac.kr") == -1:
        view = "https://www.sookmyung.ac.kr" + view

    #print(view)
    #print("-" * 50)

    # 학과소개 pdf
    pdfEl = item.select(".info")
    pdf = pdfEl[0].attrs['href']
    #print(pdf)
    #print("-" * 50)

    # 학과홈페이지
    homepageEl = item.select(".homepage")
    homepage = homepageEl[0].attrs['href']
    #print(homepage)
    #print("-" * 50)

    # 수집한 값들을 딕셔너리로 묶은 후 리스트에 추가
    college_dict = {"학과이름": name, "상세페이지": view, "홈페이지": homepage}


    #-----------------------------------------------------
    # 상세보기 URL을 새롭게 수집
    #-----------------------------------------------------
    r = session.get(view)

    if r.status_code != 200:
        # 에러코드와 에러메시지 출력
        msg = "[%d Error] %s 에러가 발생함" % (r.status_code, r.reason)
        print(msg)
        continue

    r.encoding = "utf-8"
    detailSoup = BeautifulSoup(r.text)
    #print(detailSoup)

    # 같은 구조를 갖는 전화번호, 팩스번호, 위치, 이메일주소를 가져온다.
    infoEl = detailSoup.select(".college_info_data dl")
    #print(infoEl)
    #print("-" * 50)

    for info in infoEl:
```

```
        #print(info)
        dt = info.select("dt")[0].text.replace(":", "").strip()
        dd = info.select("dd")[0].text.replace(":", "").strip()
        #print(dt, dd)
        #print("-" * 50)
        college_dict[dt] = dd
    #-----------------------------------------------------
    # 상세보기 URL을 새롭게 수집 (끝)
    #-----------------------------------------------------

    학과목록_리스트.append(college_dict)

    # pdf파일 다운로드 -> URL을 저장하고 있는 변수는 pdf
    r = session.get(pdf, stream=True)
    if r.status_code == 200:
        r.encoding = "utf-8"
        with open("%s.pdf" % name , 'wb') as f:
            f.write(r.raw.read())

df = DataFrame(학과목록_리스트)
df.to_excel("result.xlsx")
df
```