

연습문제(2) - 쿠팡 상품 목록 수집

단독방에 주어진 URL은 쿠팡에서 "노트북" 키워드로 검색된 상품 목록에 대한 주소이다.

이 주소의 웹 페이지로부터 상품의 이름, 가격, 할인율, 할인가, 별점, 리뷰수, 상품이미지의 경로를 수집하여 엑셀 파일로 저장하시오.

```
# 필요한 모듈 참조
```

```
import requests
from bs4 import BeautifulSoup
from pandas import DataFrame
```

```
# 수집할 콘텐츠가 있는 웹 페이지의 주소
```

```
url = "https://www.coupang.com/np/search?q=%EB%85%B8%ED%8A%B8%EB%B6%81&listSize=72&cha
```

```
# 브라우저 버전정보
```

```
userAgent = "Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like
```

```
# 접속객체 생성
```

```
session = requests.Session()
```

```
# 접속객체에 추가정보(header) 삽입하기
```

```
session.headers.update({
    "Referer": "",
    "User-Agent": userAgent,
    "Accept-Language": "ko-KR,ko;q=0.8,en-US;q=0.5,en;q=0.3"
})
```

```
# 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)
```

```
# 인코딩 형식 지정하여 beautifulsoup 객체를 생성
```

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

```
product = soup.select(".search-product-wrap")
print("검색결과 수: ", len(product))
#product
```

```

# 추출할 데이터를 병합할 빈 리스트
data = []

for p in product:
    # 상품명
    nameEl = p.select(".name")
    if nameEl:
        name = nameEl[0].text.strip()
    else:
        name = None
    #print(name)

    # 가격(정가)
    basePriceEl = p.select(".base-price")
    if basePriceEl:
        basePrice = int(basePriceEl[0].text.strip().replace(", ", ""))
    else:
        basePrice = None
    #print(basePrice)

    # 할인율
    discountEl = p.select(".instant-discount-rate")
    if discountEl:
        discount = int(discountEl[0].text.strip().replace("%", ""))/100
    else:
        discount = None
    #print(discount)

    # 할인가
    priceEl = p.select(".price-value")
    if priceEl:
        price = int(priceEl[0].text.strip().replace(", ", ""))
    else:
        price = None
    #print(price)

    # 별점
    ratingEl = p.select(".rating")
    if ratingEl:
        rating = float(ratingEl[0].text.strip())
    else:
        rating = None
    #print(rating)

    # 리뷰수
    reviewEl = p.select(".rating-total-count")
    if reviewEl:
        review = reviewEl[0].text.strip()
        review = int(review[1:-1])
    else:
        review = None
    #print(review)

    # 상품 이미지 주소
    imageEl = p.select(".search-product-wrap-img")
    if imageEl:

```

```

        if "data-img-src" in imageEl[0].attrs:
            image = "https:" + imageEl[0].attrs['data-img-src']
        else:
            image = "https:" + imageEl[0].attrs['src']
    else:
        image = None
    #print(image)

    # 중고상품 판별 조건
    if not basePrice and not price:
        continue

    # 상품 하나에 대한 값들을 딕셔너리로 묶음
    item = {
        "상품명": name,
        "가격": basePrice,
        "할인율": discount,
        "할인가": price,
        "별점": rating,
        "리뷰수": review,
        "상품이미지": image
    }

    # 하나의 상품 정보를 미리 준비한 빈 리스트에 원소로 추가
    data.append(item)

    # 최종 정보를 데이터 프레임으로 변환
    df = DataFrame(data)
    df

```