

---

# 데이터 수집 미니 프로젝트

---

이진규

# A Table of Contents.

- 1 과제 개요 및 프로세스 소개
- 2 과제 수행 코드 및 결과 발표
- 3 과제 수행 중 오류 해결 프로세스

---

Part 1, 과제 개요 및 프로세스 소개

---



# 과제 개요 및 프로세스 소개



## 마켓의 리뷰 데이터 요청

네이버 쇼핑 등 온라인 마켓의 리뷰 데이터와 상품 정보 등 상품의 경쟁력을 확인할 수 있는 데이터가 필요



## 데이터 수집을 위한 코드 작성

파이썬의 크롤링 및 스크래핑을 자동으로 진행하는 코드 작성으로 다수의 사이트에서 리뷰 데이터 수집을 자동화



## 수집한 데이터의 DB 입력

파이썬을 활용하여 수집한 데이터를 오라클 데이터베이스에 수집한 자료 유형에 따라 여러 테이블에 저장하여 활용

## 작업 진행 상세 프로세스

### STEP 1

작업 진행 대상 상품  
군 및 수집할 쇼핑 사  
이트 선정

&gt;&gt;

### STEP 2

수집할 데이터 및 사  
이트 구조를 확인하  
여 데이터 수집을 자  
동으로 진행하는 파  
이썬 코드 작성

&gt;&gt;

### STEP 3

수집한 데이터를 오  
라클 DB에 입력하기  
위한 파이썬 - 오라클  
연동 코드 작성 및 데  
이터 입력

&gt;&gt;

### STEP 4

입력한 데이터 확인  
및 작성 코드 재검토

# 과제 개요 및 프로세스 소개

## 작업 진행 중 사용할 툴 및 패키지

Step 1	Step 2	Step 3	Step 4
제품 및 사이트 선정	크롤링 코드 작성	데이터 - DB 연동 및 입력	수집 데이터 확인
웹 브라우저 <ul style="list-style-type: none"><li>Google 크롬</li></ul>	파이썬 프로그래밍 <ul style="list-style-type: none"><li>Selenium</li><li>Bs4</li></ul>	파이썬 - 오라클 연동 <ul style="list-style-type: none"><li>cx_Oracle</li></ul>	DB 관리 툴 <ul style="list-style-type: none"><li>SQLDeveloper</li></ul>

크롤링 및 데이터 - DB 연동시에는 각 프로세스를 클래스로 모듈화시켜 작업 진행

---

Part 2, 과제 수행 코드 및 결과

---



# 과제 수행 코드 및 결과

Step 1 : 제품 및 사이트 선정



갤럭시 버즈2프로 SM-R510 오늘출발

138,960원

LJK님만을 위한 혜택

최대 적립 포인트

7,397원 ?

기본적립

1,389원

N+ 멤버십 멤버십 추가 적립 >

5,558원

TIP. 포인트 더 받는 방법

+최대 2,779원

N 2% 네이버페이 머니로 결제 시 >

2,779원

연 최대 12만 포인트 네이버 현대카드 >

모바일할인 | 모바일 주문시 10원 추가할인

무이자할부 | 카드 자세히보기 ?

제품군

이어폰

플랫폼

네이버 쇼핑

비고

- 이어폰은 개인별 제조사 선호도가 다양하여 제품별로 리뷰수가 많음
- 이어폰은 보급이 대중화되어 있어서 수집할 리뷰 데이터도 충분히 많음
- 리뷰 수집 사이트는 네이버 쇼핑으로 일원화하여 코드 작성을 단순화하였음
- 리뷰는 페이지당 20개의 리뷰를 조회할 수 있으며 단일 url에서 소스가 변하는 방식이므로 Selenium을 활용해야 함
- 다만, 리뷰 긍정 (3점 이상)의 비율이 부정 비율에 비하여 지나치게 높아 분류분석 모델 생성으로는 적합한 데이터 샘플이 되지 못함



# 과제 수행 코드 및 결과

Step 1 : 제품 및 사이트 선정

## 선정 모델 및 사이트 url

1. 애플 에어팟 3세대 : <https://brand.naver.com/applestore/products/5985541143>
2. 애플 에어팟 프로 2세대 : <https://brand.naver.com/applestore/products/9360093290>
3. 삼성 갤럭시 버즈 2 : <https://smartstore.naver.com/uniyuni/products/6071556962>
4. 삼성 갤럭시 버즈 2 프로 : <https://smartstore.naver.com/o-ma/products/7363123499>
5. 소니 WF-1000XM5 : <https://brand.naver.com/sonystore/products/8932776097>
6. QCY T13 : <https://smartstore.naver.com/cotini/products/5357757813>

# 과제 수행 코드 및 결과

Step 2 : 쇼핑 사이트 데이터 수집

## 사이트 구조 분석

①

상세정보	리뷰 828	Q&A 63	반품/교환정보
<p>최소한으로 마진율받고 고객님들께 더저가 정보 알려 드립니다 총알배송해드리려 아침새벽밤낮없이 포장하구있답니다 !! 만족스러우셨다니 감사드려영~하얏</p>			

②

★★★★★ 5

lema\*\*\* · 23.12.13. | 신고

버즈2프로(색상): 화이트

네이버 최저가에 배송도 하루만에 왔네요. 음질은 이전 모델보다 많이 좋아졌어요. 특히 저음이 빵빵합니다.



0

★★★★★ 5

le\*\*\* · 23.12.18. | 신고

버즈2프로(색상): 화이트

좋은제품 빠른배송으로 받아보았습니다. 박스 모서리가 손상되어 있었지만 제품은 양호하네요.



0

★★★★★ 5

shgk\*\*\*\*\* · 23.12.10. | 신고

버즈2프로(색상): 화이트

버즈 플러스 쓰다가 잃어버리고 줄이어폰으로 전전긍긍하고 있었는데... 너무 편하고 음질 좋고 최고네요.. 살까말까 하다가 가격 반값되고 마침 줄이어폰 고장나서 샀는데... 행복합니다아ㅏ



0

③

1 2 3 4 5 6 다음 >

- 리뷰 영역은 사이트 페이지에서 10 ~ 15% 지점 스크롤을 내렸을 때 확인되는 리뷰 버튼으로 진입할 수 있음
- 리뷰는 페이지당 20개의 리뷰를 확인할 수 있으며 별점, 등록일자, 구매 옵션, 리뷰 본문을 확인할 수 있음
- 리뷰 페이지는 하단 번호 버튼으로 넘어갈 수 있음. 다만, 페이지 표기 방식이 각 제품별로 아래 2가지로 나뉨
  - 유형 1 : 10개 단위 페이지 ( 1 ~ 10, 11 ~ 20 ...)로 페이지 리스트를 표기
  - 유형 2 : 현재 조회중인 페이지를 기준으로 앞 뒤 5개 페이지를 표기 ( 6 ~ 16 )
  - 각 2가지 유형에 범용적으로 적용할 수 있는 코드 작성 필요

## 사이트 구조 분석

①

갤럭시 버즈2프로 SM-R510 오늘출발

②

138,960원

### LJK님만을 위한 혜택

최대 적립 포인트 7,397원 ?

↳ 기본적립 1,389원

↳ **N+** 멤버십 멤버십 추가 적립 > 5,558원**TIP.** 포인트 더 받는 방법 +최대 2,779원**N 2%** 네이버페이 머니로 결제 시 > 2,779원

연 최대 12만 포인트 네이버 현대카드 &gt;

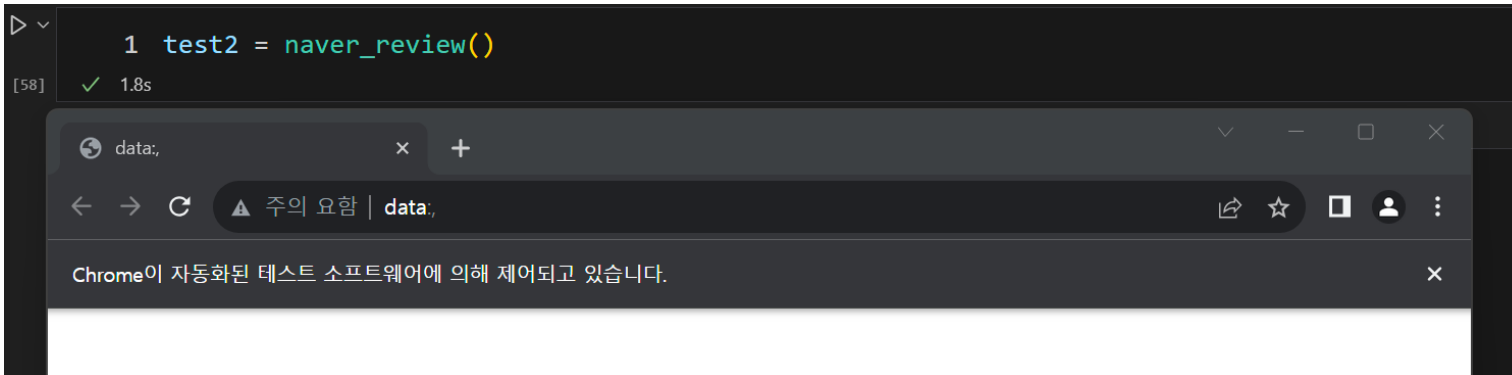
- DB에 상품의 기본적인 정보 입력을 위하여 추가적인 데이터 수집 진행

- 상품명 수집
- 상품 가격 수집

## 크롤링 수행 클래스 작성 (naver\_review)

메서드	입력데이터	내용	출력데이터
생성자 (__init__)	없음	크롬드라이버 버전 체크 및 실행 후 빈 페이지 상태로 대기	없음
connect_url	텍스트 형식의 url	입력받은 url 접속 후 리뷰 페이지로 이동	없음
rip_review	크롬드라이버 .page_source 객체	현재 조회중인 리뷰 페이지에서 최대 20개의 리뷰 데이터 수집	단일 페이지의 리뷰 정 보를 담은 리스트
rip_all	없음	현재 조회중인 페이지부터 시작하여 페이지 끝까지 연속하여 데이터를 수집 단일 페이지의 리뷰는 rip_review 메서드를 호출하여 리뷰를 수집	다수 페이지의 리뷰 정 보를 담은 리스트
page_summary	없음	현재 조회중인 페이지의 리뷰 정보와 상품명, 상품 가격을 수집 리뷰 정보는 rip_all 메서드를 호출하여 리뷰를 수집	페이지의 상품 정보와 리뷰를 담은 딕셔너리
rip_list	텍스트형식의 url을 담은 리스트	입력받은 리스트 내 모든 url을 탐색하여 상품 정보 및 리뷰를 모두 수집 단일 url에 대한 정보는 page_summary 메서드를 호출하여 수집	url별 모든 정보를 담은 리스트
dose	없음	크롬 드라이버 접속 종료	없음

```
class naver_review(): # 네이버 쇼핑에서 리뷰를 수집하는 클래스 객체
    def __init__(self): # 생성자에서 Selenium을 이용한 크롬드라이버 창 실행
        chromedriver_autoinstaller.install()
        self.driver = webdriver.Chrome()
        self.driver.implicitly_wait(2)
```

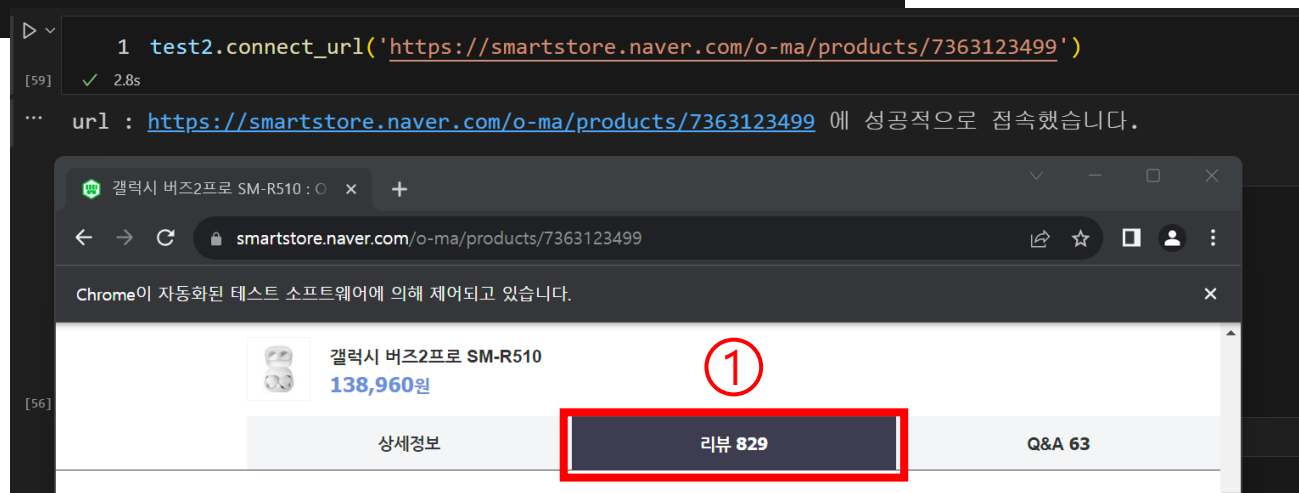


클래스 생성 시 생성자에 의하여 크롬 드라이버가 열림

```
def connect_url(self, url): # 특정 상품 페이지로 접속하는 메서드, url : 접속 대상 url을 텍스트 형식으로 입력
    self.driver.get(url) # url 접속
    time.sleep(1) # 페이지 로드 대기
    self.driver.execute_script("window.scrollTo(0, (document.body.scrollHeight)*0.15);") # 리뷰 메뉴 확인을 위한 스크롤 이동 (전체의 15% 지점)
    time.sleep(0.5) # 페이지 로드 대기
    # 리뷰 메뉴 지정 (해당 CSS 선택자 검색시 총 4개의 요소가 잡히며 상세정보-리뷰-Q&A-반품/교환정보순, 1번째인 리뷰를 선택)
    self.review_btn = self.driver.find_elements(By.CSS_SELECTOR, 'a._11xFby3Le')[1]
    self.review_btn.click() # 리뷰 메뉴 선택
    print('url : {0} 에 성공적으로 접속했습니다.'.format(url)) # 페이지 접속 성공 메시지 출력
    time.sleep(0.5) # 페이지 로드 대기
```

1. CSS 선택자로 a.\_11xFby3Le를 선택할 경우 총 4개의 요소가 선택됨

- 상세정보, 리뷰, Q&A, 반품/교환정보의 4개로 인덱스가 1번째인 리뷰 번호를 인덱싱으로 선택하여 클릭하여 리뷰 페이지로 진입



connect\_url 메서드로 리뷰 페이지 접속

# 과제 수행 코드 및 결과

Step 2 : 쇼핑 사이트 데이터 수집

```
def rip_review(self, x): # 단일 리뷰 페이지의 리뷰 데이터를 수집하는 메서드, x는 크롬드라이버의 .page_source 메서드로 생성된 소스 데이터
    self.soup = BeautifulSoup(x) # 현재 크롬드라이버의 소스 코드를 수집하여 soup 객체 생성

    self.reviews = self.soup.select('div._1McWUwk15j') # 페이지당 20개의 리뷰가 존재하며, 해당 20개의 리뷰 영역을 선택하는 CSS 선택자
    self.review_list = [] # 리뷰 데이터를 담은 리스트 생성

    for i in self.reviews: # 20개의 리뷰 영역에 대하여 반복문 수행
        self.review_temp = {} # 1개 분량의 리뷰 데이터를 담은 딕셔너리 생성

        try: # 선택한 상품 옵션을 추출
            self.review_temp['선택상품'] = i.select('div._2FXNMst_ak')[0].text.strip()
        except:
            self.review_temp['선택상품'] = None

        try: # 리뷰 작성일 추출
            self.review_temp['리뷰작성일'] = i.select('div.iWGqB6S4Lq span._2L3vDiadT9')[0].text.strip()
        except:
            self.review_temp['리뷰작성일'] = None

        try: # 별점을 정수형 데이터로 추출
            self.review_temp['별점'] = int(i.select('em._15NU42F3kT')[0].text.strip())
        except:
            self.review_temp['별점'] = None

        # 추출한 정수형 별점 데이터가 3 이상일 경우 긍정 판정, 별점이 2 이하일 경우 부정, 별점을 추출하지 못했을 경우 결측치로 처리
        if self.review_temp['별점'] >= 3:
            self.review_temp['긍정/부정'] = '긍정'
        elif self.review_temp['별점'] == None:
            self.review_temp['긍정/부정'] = None
        else:
            self.review_temp['긍정/부정'] = '부정'

        try: # 리뷰의 내용을 추출
            self.review_temp['리뷰내용'] = i.select('div._1kMfD5ErZ6 span._2L3vDiadT9')[0].text.replace("\n", " ").strip()
        except:
            self.review_temp['리뷰내용'] = None

        self.review_list.append(self.review_temp) # 추출한 리뷰 정보를 담은 딕셔너리를 리뷰 리스트에 추가

    return self.review_list # 최종적으로 20개의 리뷰를 반환
```

상세정보	리뷰 829	Q&A 63	반품/교환정보
<p>★★★★★ 5 p_**** · 23.12.09.   신고 베즈2프로(색상): 화이트</p> <p>가성비 대만족합니다. 음질도 좋네요. 타제품 비교 노캔이 약간부족한거같긴한데 그렇게 나쁘지도 않아요. 이제품 나름대로의 노캔 영역이라고 해야할까요? 기본적인 소음은 정말 깔끔한편인것 같아요. 아무튼 좋겠! 가격까지 비교하면 정말 대만족! = 제가 가진 다른 이어폰들하고 크기비교샷도 찍었습니다.</p>			
<p>★★★★★ 5 gemo**** · 23.11.28.   신고 베즈2프로(색상): 화이트</p> <p>이전에 보스체를 코에서 구입하고 노캔베고는 가격대비 별로라 반납하고 베즈 고민하다 구입했는데 제귀가 막이인지 음질이 잘모르겠네요. 다음 기종나올때 까지 기다릴까도 생각했지만 나와도 20후반대 가격이라면 베즈2프로가 현시점 가격이나 성능에서 현명한 선택이라는 생각이드네요</p>			
<p>★★★★★ 5 베즈2프로(색상): 화이트</p> <p>잘 받았어요~! 원래 베즈 프로 보라색 썼는데 신랑이 가격가서 새로 주문했네요~! 소리도 잘 들리고 기존 베즈 프로보다는 살짝 작은거같아요~! 귀가 작아서 전에꺼 끼면 귀가 늘어나면서 따끔하고 간지러웠는데 이걸 그런 느낌에 견어 없네요~^^ 알록달록한걸 좋아해서 흰색은 처음으로 써보는데 흰색도 깔끔하고 예뻐요^^ 사진은 케이스 싸우기 전에 찍어놨는데 첨부가 안되서 케이스 끼운뒤 찍은 사진으로 올렸어요~</p>			
<p>판매회사 23.12.18.   신고</p> <p>허여적 제품케이스에 김나귀어온 토끼머리를놓으니까 너무귀여워영&gt; _&lt; 너무잘어울린다.. 게는제거한생각을못했을까용... 아이디어 너무센스좋은시내여!! 예쁘게잘사용해주세요~</p>			
<p>★★★★★ 5 b**** · 23.12.12.   신고 베즈2프로(색상): 화이트</p> <p>잘 받았어요! 배송이 확실히 빠르고 반자마자 충전 인식 다 잘되서 좋았어요 다음날 오전에 2시간 정도 연속으로 써봤는데 중간에 좀 끊기는 부분이 있었지만 잘 작동되고 있어요 변형하세요!</p>			

## 20개의 리뷰를 원소로 갖는 리스트를 추출

# 과제 수행 코드 및 결과

Step 2 : 쇼핑 사이트 데이터 수집

```
def rip_review(self, x): # 단일 리뷰 페이지의 리뷰 데이터를 수집하는 메서드, x는 크롬드라이버의 .page_source 메서드로 생성된 소스 데이터
    self.soup = BeautifulSoup(x) # 현재 크롬드라이버의 소스 코드를 수집하여 soup 객체 생성
    self.reviews = self.soup.select('div._1McWUwk15j') # 페이지당 20개의 리뷰가 존재하며, 해당 20개
    self.review_list = [] # 리뷰 데이터를 담은 리스트 생성

    for i in self.reviews: # 20개의 리뷰 영역에 대하여 반복문 수행
        self.review_temp = {} # 1개 분량의 리뷰 데이터를 담은 딕셔너리 생성

        try: # 선택한 상품 옵션을 추출
            self.review_temp['선택상품'] = i.select('div._2FXNMst_ak')[0].text.strip()
        except:
            self.review_temp['선택상품'] = None

        try: # 리뷰 작성일 추출
            self.review_temp['리뷰작성일'] = i.select('div.iWGqB6S4Lq span._2L3vDiadT9')[0].text.strip()
        except:
            self.review_temp['리뷰작성일'] = None

        try: # 별점을 정수형 데이터로 추출
            self.review_temp['별점'] = int(i.select('em._15NU42F3kT')[0].text.strip())
        except:
            self.review_temp['별점'] = None

        # 추출한 정수형 별점 데이터가 3 이상일 경우 긍정 판정, 별점이 2 이하일 경우 부정, 별점을 추출하지 못했을 경우 결측치로 처리
        if self.review_temp['별점'] >= 3:
            self.review_temp['긍정/부정'] = '긍정'
        elif self.review_temp['별점'] == None:
            self.review_temp['긍정/부정'] = None
        else:
            self.review_temp['긍정/부정'] = '부정'

        try: # 리뷰의 내용을 추출
            self.review_temp['리뷰내용'] = i.select('div._1kMfD5ErZ6 span._2L3vDiadT9')[0].text.replace("\n", " ").strip()
        except:
            self.review_temp['리뷰내용'] = None

        self.review_list.append(self.review_temp) # 추출한 리뷰 정보를 담은 딕셔너리를 리뷰 리스트에 추가

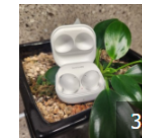
    return self.review_list # 최종적으로 20개의 리뷰를 반환
```



★★★★★ 5  
gems\*\*\* 23.11.28. 14:40

버즈2프로(색상): 화이트

이전에 보스제품 코코에서 구입하고 노캔빼고는 가격대비 별로라 반납하고 버즈 고민하다 구입했는데 제귀가 막귀인지 음  
질차이 잘모르겠네요. 다음 기종나올때 까지 기다릴까도 생각했지만 나와도 20후반대 가격이라면 버즈2프로가 현시점 가격  
이나 성능에서 현명한 선택이라는 생각이드네요



3



# 과제 수행 코드 및 결과

Step 2 : 쇼핑 사이트 데이터 수집

```
def rip_all(self): # 현재 조회중인 단일 url의 전체 리뷰를 수집하는 메서드
    self.all_reviews = [] # 모든 리뷰를 담을 리스트 생성
    self.idx = 1 # 조회하는 페이지를 판별하는 index 변수 생성

    while True: # 반복을 수행할 횟수가 불분명하므로 무한루프 설정
        self.all_reviews += self.rip_review(self.driver.page_source) # rip_review 메서드를 이용하여 모든 리뷰를 담을 리스트에 데이터를 추가
        self.idx += 1 # 현재 페이지의 리뷰를 모두 수집했으므로 수집할 페이지 변수를 1 증가시킴

        self.button_list = self.driver.find_elements(By.CSS_SELECTOR, 'div._1HJarNZHiI._2UJrM31-Ry a') # 모든 리뷰 페이지 버튼을 탐색하는 CSS 선택
        self.button_text = [] # 리뷰 페이지 버튼의 실제 텍스트를 얻을 검색어 리스트 생성
        for i in self.button_list: # 리뷰 페이지 버튼을 담은 리스트에 반복문을 적용하여 버튼의 텍스트를 추출
            try:
                self.button_text.append(i.text)
            except:
                self.button_text.append(None)

        if self.idx > 100: # 테스트용 리뷰 수집 갯수 제약 (100페이지 x 20개 = 2000개)
            break

        try:
            # 탐색하고자 하는 리뷰 페이지가 text의 몇 번째 인덱스인지 확인하여 해당 인덱스 번호를 기존 리뷰 페이지 버튼에 적용하여 다음 페이지 버튼을 지정
            self.btn = self.button_list[self.button_text.index(str(self.idx))]
            self.btn.click() # 다음 페이지 버튼 클릭
            time.sleep(1) # 페이지 로드 대기
        except:
            try:
                # 탐색하고자 하는 리뷰 페이지가 11, 21 페이지 등에 의하여 다음 페이지로 넘어가야 하는 경우 '다음'으로 되어 있는 버튼을 탐색
                # 기존 idx로는 에러가 발생하므로 예외처리
                self.btn = self.button_list[self.button_text.index('다음')]
                self.btn.click() # 다음 페이지 버튼 클릭
                time.sleep(1) # 페이지 로드 대기
            except:
                break # 다음 idx 페이지도, '다음' 페이지도 모두 없을 경우 페이지의 끝에 도달한 것으로 판단하여 break 문으로 무한루프 해제

    return self.all_reviews # 현재까지 수집한 모든 리뷰 데이터를 반환
```



# 과제 수행 코드 및 결과

Step 2 : 쇼핑 사이트 데이터 수집

```
self.button_text = [] # 리뷰 페이지 버튼의 실제 텍스트를 담은 임시 리스트 정의
for i in self.button_list: # 리뷰 페이지 버튼을 담은 리스트에 반복문을 적용하여 버튼의 텍스트를 추출
    try:
        self.button_text.append(i.text)
    except:
        self.button_text.append(None)
```



1
2
3
4
5
6
다음 >

- button\_list = [객체(1), 객체(2), 객체(3), 객체(4), 객체(5), 객체(6), 객체(다음)]
- button\_text = [ '1', '2', '3', '4', '5', '6', '다음' ]

```
self.btn = self.button_list[self.button_text.index(str(self.idx))]
```

- 텍스트로 변환한 리스트에서 idx 값을 기준으로 필요한 인덱스 값을 추적하여 해당 인덱스 값을 find\_elements 리스트에 적용하여 필요한 객체를 지정

# 과제 수행 코드 및 결과

Step 2 : 쇼핑 사이트 데이터 수집

```
def page_summary(self): # 현재 조회중인 url에서 리뷰 외 추가적인 상품명, 상품 가격을 추출하는 메서드
    summary = {} # 데이터를 담은 딕셔너리 생성
    summary['product_name'] = self.driver.find_element(By.CSS_SELECTOR, 'h3._22kNQuEXmb').text # 상품명 수집
    summary['product_price'] = int(self.driver.find_element(By.CSS_SELECTOR, 'span._1LY7DqCnwR').text.replace(",","")) # 상품가격 수집
    summary['reviews'] = self.rip_all() # rip_all도 현재 조회중인 url의 리뷰 데이터 수집

    print("상품 페이지에서 총 {0}개의 리뷰를 수집했습니다.".format(len(summary['reviews']))) # 수집한 데이터의 개수 출력
    return summary # 데이터를 담은 딕셔너리 반환
```

1. 수집한 정보는 아래와 같은 딕셔너리 구조로 저장

- 'product\_name' 키 : 텍스트 형식의 상품명을 value로 저장
- 'product\_price' 키 : 숫자 형식의 상품가격을 value로 저장
- 'reviews' 키 : rip\_all 메서드를 통해 반환받은 리스트로 된 리뷰 데이터를 value로 저장

갤럭시 버즈2프로 SM-R510

오늘출발

138,960원

page\_summary 메서드에서 상품의 기본 정보와 리뷰를 딕셔너리 형태로 반환

# 과제 수행 코드 및 결과

Step 2 : 쇼핑 사이트 데이터 수집

```
def rip_list(self, x): # 여러개의 url에서 데이터를 수집하는 메서드, x : url을 원소로 하는 리스트
    self.all_list = [] # 모든 데이터를 수집할 리스트 정의

    for i in x: # connect_url 메서드로 접속하고, page_summary 메서드로 url의 정보를 수집하는 메서드를 반복문으로 수행하여 리스트에 데이터 수집
        self.connect_url(i)
        self.all_list.append(self.page_summary())

    return self.all_list # 수집한 모든 데이터 반환
```

1. rip\_list는 page\_summary 메서드를 반복수행하여 반환값을 리스트에 담은 메서드
2. 모든 작업이 완료되면 close 메서드로 크롬드라이버 종료

```
def close(self): # 크롬드라이버를 닫는 메서드
    try:
        self.driver.close()
    except:
        pass
```

## naver\_review 클래스 사용 예시 및 수집 데이터 출력 결과

```

1 url_list = ['https://smartstore.naver.com/o-ma/products/7363123499',
2             'https://brand.naver.com/applestore/products/9360093290',
3             'https://smartstore.naver.com/uniyuni/products/6071556962',
4             'https://smartstore.naver.com/o-ma/products/7363123499',
5             'https://brand.naver.com/sonystore/products/8932776097',
6             'https://smartstore.naver.com/cotini/products/5357757813']

[56] ✓ 0.0s

1 test = naver_review()
2 all_list = test.rip_list(url_list)
3 test.close()

[57] ✓ 7m 45.2s

... url : https://smartstore.naver.com/o-ma/products/7363123499 에 성공적으로 접속했습니다.
    상품 페이지에서 총 828개의 리뷰를 수집했습니다.
    url : https://brand.naver.com/applestore/products/9360093290 에 성공적으로 접속했습니다.
    상품 페이지에서 총 1006개의 리뷰를 수집했습니다.
    url : https://smartstore.naver.com/uniyuni/products/6071556962 에 성공적으로 접속했습니다.
    상품 페이지에서 총 780개의 리뷰를 수집했습니다.
    url : https://smartstore.naver.com/o-ma/products/7363123499 에 성공적으로 접속했습니다.
    상품 페이지에서 총 828개의 리뷰를 수집했습니다.
    url : https://brand.naver.com/sonystore/products/8932776097 에 성공적으로 접속했습니다.
    상품 페이지에서 총 1477개의 리뷰를 수집했습니다.
    url : https://smartstore.naver.com/cotini/products/5357757813 에 성공적으로 접속했습니다.
    상품 페이지에서 총 2000개의 리뷰를 수집했습니다.

```

```

1 all_list

[60] ✓ 0.4s

... [{ 'product_name': '갤럭시 버즈2프로 SM-R510',
      'product_price': 137480,
      'reviews': [{ '선택상품': '버즈2프로(색상): 화이트',
                    '리뷰작성일': '23.12.08.',
                    '별점': 5,
                    '긍정/부정': '긍정',
                    '리뷰내용': '일단 같은 삼성 브랜드 제품과의 호환성이 좋았습니다. 이어팁은 대/중/소 사이즈가
                    { '선택상품': '버즈2프로(색상): 화이트',
                      '리뷰작성일': '23.12.09.',
                      '별점': 5,
                      '긍정/부정': '긍정',
                      '리뷰내용': '가성비 대만족합니다. 음질도 좋네요. 타제품 비교 노캔이 약간부족한거같긴한데 그릴
                    { '선택상품': '버즈2프로(색상): 화이트',
                      '리뷰작성일': '23.11.28.',
                      '별점': 5,
                      '긍정/부정': '긍정',
                      '리뷰내용': '이전에 보스제품 코코에서 구입하고 노캔빼고는 가격대비 별로라 반납하고 버즈 고민하
                    { '선택상품': '버즈2프로(색상): 화이트',
                      '리뷰작성일': '23.12.16.',

```

## 수집한 리뷰 데이터 형태

	선택상품	리뷰작성일	별점	긍정/부정	리뷰내용
0	버즈2프로(색상): 화이트	23.12.08.	5	긍정	일단 같은 삼성 브랜드 제품과의 호환성이 좋았습니다. 이어팁은 대/중/소 사이즈가...
1	버즈2프로(색상): 화이트	23.12.09.	5	긍정	가성비 대만족합니다. 음질도 좋네요. 타제품 비교 노캔이 약간부족한거같긴한데 그렇게...
2	버즈2프로(색상): 화이트	23.11.28.	5	긍정	이전에 보스제품 코코에서 구입하고 노캔빼고는 가격대비 별로라 반납하고 버즈 고민하다...
3	버즈2프로(색상): 화이트	23.12.16.	5	긍정	잘 받았어요~! 원래 버즈 프로 보라색 썼는데 신랑이 가져가서 새로 주문했네요~!...
4	버즈2프로(색상): 화이트	23.12.12.	5	긍정	잘 받았어요! 배송이 확실히 빠르고 받자마자 충전 인식 다 잘되서 좋았어요 다음날 ...

## 파이썬 – 오라클 연동 클래스 (input\_sql)

메서드	입력데이터	내용	출력데이터
생성자 (__init__)	없음	cx_oracle을 통하여 DB에 접속	없음
create_table	없음	데이터를 입력할 테이블들을 생성하는 메서드	없음
Input_data	1. naver_review의 rip_list 반환값 2. 텍스트 url을 원소로 하는 리스트	전달받은 데이터를 create_table에서 생성한 테이블에 입력	없음
del_all	없음	입력된 데이터가 잘못되었을 경우 생성되었던 테이블을 삭제하는 메서드	없음
dose	없음	DB 접속 종료	없음

```
class input_sql(): # CX_ORACLE을 통하여 SQL 데이터에 접근 및 수정하는 클래스 객체 정의
    def __init__(self): # 생성자에서 DB에 접속
        self.dbcon = cx.connect("hr",
                                "hr",
                                "localhost:1521/xe")
```

클래스 생성 시 생성자에 의하여 오라클 데이터베이스 접속



## create\_table 메서드로 데이터를 입력할 테이블 및 시퀀스 생성

```
def create_table(self): # 객체 생성시 입력받은 리스트로 SQL 테이블을 생성하는 메서드 정의
    self.cursor = self.dbcon.cursor() # SQL문을 입력할 cursor 객체
    self.num_table = 0 # 생성한 테이블 개수를 기록할 변수
    # 상품 정보를 담은 포맷을 가진 테이블 생성
    self.sql_base = "create table PRODUCT_LIST (product_id number not null, product varchar2(255) not null, price number not null, url varchar2(255) not null, primary key(product_id))"
    self.sql_base2 = "create sequence SEQ_PRODUCT start with 1 increment by 1"
    self.cursor.execute(self.sql_base)
    self.cursor.execute(self.sql_base2)
    print("테이블 'PRODUCT_LIST'를 생성했습니다.")
    self.sql = "create table REVIEW_LIST (id number not null, product_id number not null, product_option varchar2(255), reg_date date not null, score number not null, positive_negative varchar2(25) not null, review clob not null, primary key(id), foreign key(product_id) references PRODUCT_LIST(product_id))"
    self.sql2 = "create sequence SEQ_REVIEW start with 1 increment by 1"
    self.cursor.execute(self.sql)
    self.cursor.execute(self.sql2)
    print("테이블 'REVIEW_LIST'를 생성했습니다.")
    self.cursor.close() # cursor 객체 접속 종료
```

```
create table PRODUCT_LIST (product_id number not null, product varchar2(255) not null, price number not null, url varchar2(255) not null, primary key(product_id))
```

```
create table REVIEW_LIST (id number not null, product_id number not null, product_option varchar2(255), reg_date date not null, score number not null, positive_negative varchar2(25) not null, review clob not null, primary key(id), foreign key(product_id) references PRODUCT_LIST(product_id))
```

## 생성되는 테이블 구조

PRODUCT_LIST			
	COLUMN_NAME	DATA_TYPE	NULLABLE
1	PRODUCT_ID	NUMBER	No
2	PRODUCT	VARCHAR2 (255 BYTE)	No
3	PRICE	NUMBER	No
4	URL	VARCHAR2 (255 BYTE)	No

PRODUCT\_LIST (상품리스트 테이블)

- PRODUCT\_ID : 상품 ID, NUMBER, Primary Key, SEQ\_PRODUCT로 자동생성
- PRODUCT : 상품명, VARCHAR2(255)
- PRICE : 상품가격, NUMBER
- URL : 상품 페이지 URL, VARCHAR2(255)

REVIEW_LIST			
	COLUMN_NAME	DATA_TYPE	NULLABLE
1	ID	NUMBER	No
2	PRODUCT_ID	NUMBER	No
3	PRODUCT_OPTION	VARCHAR2 (255 BYTE)	Yes
4	REG_DATE	DATE	No
5	SCORE	NUMBER	No
6	POSITIVE_NEGATIVE	VARCHAR2 (25 BYTE)	No
7	REVIEW	CLOB	No

REVIEW\_LIST (리뷰 데이터 테이블)

- ID : 리뷰 ID, NUMBER, Primary Key, SEQ\_REVIEW로 자동생성
- PRODUCT\_ID : 상품 ID, NUMBER, PRODUCT\_LIST의 PRODUCT\_ID를 Foreign Key로 참조
- PRODUCT\_OPTION : 구매 옵션, VARCHAR2(255), 결측값 허용
- REG\_DATE : 리뷰 날짜, DATE
- SCORE : 별점, NUMBER
- POSITIVE\_NEGATIVE : 긍정/부정, VARCHAR2(25)
- REVIEW : 리뷰 본문, CLOB

## input\_data 메서드로 테이블에 데이터 순차 입력

```
def input_data(self, x, y): # 생성한 테이블에 데이터를 삽입하는 메서드 정의
    self.table = x.copy() # 클래스 객체를 생성할 때 naver_review 클래스의 rip_list로 반환받은 리스트를 입력
    self.url = y.copy() # 상품 정보 테이블에 입력할 url 목록을 리스트로 입력
    self.cursor = self.dbcon.cursor() # SQL문을 입력할 cursor 객체
    self.num_data = 0 # 입력한 데이터 개수를 기록할 변수

    # PRODUCT_LIST에 상품 목록을 기입하는 반복문 수행
    for i, v in enumerate(self.table):
        self.sql_base = "insert into PRODUCT_LIST values({0}, '{1}', {2}, '{3}')"
        self.cursor.execute(self.sql_base.format('SEQ_PRODUCT.nextval', v['product_name'], v['product_price'], self.url[i]))

    # REVIEW_LIST에 리뷰 데이터를 기입하는 반복문 수행
    for i, v in enumerate(self.table): # url 개수만큼의 요소를 갖는 리스트 반복문 수행
        for j, w in enumerate(v['reviews']): # 단일 url 요소인 딕셔너리에서 'reviews'인 key의 리스트를 반복문으로 수행
            # 데이터를 입력할 SQL문 포맷 생성
            self.sql = "insert into REVIEW_LIST values({0}, '{1}', '{2}', TO_DATE('{3}', 'YY.MM.DD.'), {4}, '{5}', '{6}')"
            # 포맷에 맞게 데이터를 입력(리뷰 내용은 너무 긴 것은 LONG 타입의 컬럼에도 담을 수 없어 내용을 1000자까지 슬라이싱하여 기입) 및 SQL 문 수행
            self.cursor.execute(self.sql.format('SEQ_REVIEW.nextval', i+1, w['선택상품'],
                                                w['리뷰작성일'], w['별점'], w['긍정/부정'], w['리뷰내용'].replace("'", "'")[:1000]))
            self.num_data += 1 # 데이터를 입력했을 경우 입력한 데이터 수를 1 증가시킴

    print("'REVIEW_LIST' 테이블에 {0} 상품에 대한 {1}개의 리뷰 데이터를 입력했습니다.".format(v['product_name'], self.num_data)) # 입력한 데이터
    self.cursor.execute('commit') # COMMIT SQL문으로 입력한 데이터를 실제로 적용
    self.num_data = 0 # 다음 테이블에서 사용할 입력한 데이터 수를 위하여 0으로 초기화
    self.cursor.close() # cursor 객체 접속 종료
```

PRODUCT\_LIST 테이블

REVIEW\_LIST 테이블

## 생성했던 테이블을 삭제하는 del\_all 메서드와 접속을 종료하는 close 메서드

```
def del_all(self): # 데이터 입력이 잘못되었거나 완전히 새로운 데이터를 입력할 경우 테이블과 Sequence를 삭제하는 메서드 정의
    self.cursor = self.dbcon.cursor() # SQL문을 실행할 cursor 객체

    # 리뷰 리스트 테이블과 Sequence 삭제
    # REVIEW_LIST가 외래 키로 PRODUCT_LIST의 product_id를 참조하고 있으므로 정상적인 삭제 프로세스로 REVIEW_LIST를 먼저 삭제해야함
    try:
        self.cursor.execute("drop table REVIEW_LIST")
        self.cursor.execute("drop sequence SEQ_REVIEW")
        print("리뷰 테이블과 Sequence를 삭제했습니다.")
    except:
        pass

    try: # PRODUCT_LIST 테이블과 Sequence 삭제
        self.cursor.execute("drop table PRODUCT_LIST")
        self.cursor.execute("drop sequence SEQ_PRODUCT")
        print("상품 테이블과 Sequence를 삭제했습니다.")
    except:
        pass

    self.cursor.close() # cursor 객체 접속 종료
```

```
def close(self): # DB에 접속한 dbcon을 종료하는 메서드
    try:
        self.dbcon.close()
    except:
        pass
```

REVIEW\_LIST의 product\_id가 PRODUCT\_ID의 product\_id를 외래키로 참조하고 있으므로 REVIEW\_LIST를 먼저 삭제해야 PRODUCT\_LIST를 삭제할 수 있음

## input\_sql 메서드 사용 예시

```
1 testing = input_sql()
```

[50] ✓ 0.0s

```
1 testing.create_table()
```

[51] ✓ 0.0s

... 테이블 'PRODUCT\_LIST'를 생성했습니다.  
테이블 'REVIEW\_LIST'를 생성했습니다.

```
1 testing.input_data(all_list, url_list)
```

[52] ✓ 3.4s

... 'REVIEW\_LIST' 테이블에 갤럭시 버즈2프로 SM-R510 상품에 대한 828개의 리뷰 데이터를 입력했습니다.  
'REVIEW\_LIST' 테이블에 Apple 2023 에어팟 프로 2세대 USB-C 충전 케이스 모델 (MTJV3KH/A) 상품에 대한 1005개의 리뷰 데이터를 입력했습니다.  
'REVIEW\_LIST' 테이블에 삼성전자 갤럭시 버즈2 SM-R177 상품에 대한 780개의 리뷰 데이터를 입력했습니다.  
'REVIEW\_LIST' 테이블에 갤럭시 버즈2프로 SM-R510 상품에 대한 828개의 리뷰 데이터를 입력했습니다.  
'REVIEW\_LIST' 테이블에 소니 WF-1000XM5 상품에 대한 1456개의 리뷰 데이터를 입력했습니다.  
'REVIEW\_LIST' 테이블에 T13 블루투스 이어폰 무선 노이즈 캔슬링 국내AS QCY 상품에 대한 2000개의 리뷰 데이터를 입력했습니다.

## 데이터베이스 최종 입력 결과 확인

## PRODUCT\_LIST 테이블

	PRODUCT_ID	PRODUCT	PRICE	URL
1	1	갤럭시 버즈2프로 SM-R510	137480	https://smartstore.naver.com/o-ma/products/7363123499
2	2	Apple 2023 에어팟 프로 2세대 USB-C 충전 케이스 모델 (MTJV3KH/A)	359000	https://brand.naver.com/applestore/products/9360093290
3	3	삼성전자 갤럭시 버즈2 SM-R177	99900	https://smartstore.naver.com/uniyuni/products/6071556962
4	4	갤럭시 버즈2프로 SM-R510	137480	https://smartstore.naver.com/o-ma/products/7363123499
5	5	소니 WF-1000XM5	359000	https://brand.naver.com/sonystore/products/8932776097
6	6	T13 블루투스 이어폰 무선 노이즈 캔슬링 국내AS QCY	42000	https://smartstore.naver.com/cotini/products/5357757813

## REVIEW\_LIST 테이블

질의 결과 x

SQL | 1,000개의 행이 호출됨(0.016초)

ID	PRODUCT_ID	PRODUCT_OPTION	REG_DATE	SCORE	POSITIVE_NEGATIVE	REVIEW
821	821	1 버즈2프로 (색상): 화이트	22/11/19	5	긍정	노캔 잘 됩니다. 좋은 제품 빠른배송 감사합니다
822	822	1 버즈2프로 (색상): 그라파이트	22/11/17	4	긍정	대체적으로 만족함 화이트로 살걸 그랬나 후회중
823	823	1 버즈2프로 (색상): 화이트	22/11/22	5	긍정	딸아이 생일선물로 구매했는데 무척 좋아하네ㅠ
824	824	1 버즈2프로 (색상): 화이트	22/11/20	5	긍정	잘받았습니다 사용은 안해봤지만 삼성이니까...
825	825	1 버즈2프로 (색상): 라벤더	22/11/26	5	긍정	너무 좋아요 ㅎㅎㅎ
826	826	1 버즈2프로 (색상): 그라파이트	22/12/06	5	긍정	가격도 좋고 상품도 좋고 만족스럽습니다!!
827	827	1 버즈2프로 (색상): 그라파이트	22/11/28	5	긍정	너무 너무 감사 드립니다.
828	828	1 버즈2프로 (색상): 화이트	22/11/28	5	긍정	너무 너무 감사 드립니다.
829	829	2 모델 선택: 에어팟 프로 2세대 MTJV... 23/12/16	23/12/16	4	긍정	고딩아들 선물로 사줬는데 너무 잘 사용하고 있네요. 소음차단이 잘 돼서 좋다고 하네요. 2
830	830	2 모델 선택: 에어팟 프로 2세대 MTJV... 23/12/14	23/12/14	5	긍정	프로1쓰다가 바꿨습니다 음질이 뭔가 좋아진거같습니다!!! 너무 영통해요 배송도 바로오고
831	831	2 모델 선택: 에어팟 프로 2세대 MTJV... 23/12/15	23/12/15	5	긍정	프로1세대보다 모든게 좋아졌네요 노이즈캔슬링도 그렇고 매우 만족합니다.
832	832	2 모델 선택: 에어팟 프로 2세대 MTJV... 23/12/07	23/12/07	5	긍정	와이프가 기존에 쓰던 에어팟이 망가져서 올해 마지막 기념으로 선물해줬는데 아주 만족스럽
833	833	2 모델 선택: 에어팟 프로 2세대 MTJV... 23/12/12	23/12/12	5	긍정	언니 생일선물로 장만해줬습니다ㅎㅎ 엄청 좋아하네요 배송 빨랐고 포장상태도 좋았습니다!
834	834	2 모델 선택: 에어팟 프로 2세대 MTJV... 23/12/14	23/12/14	5	긍정	에어팟 프로 2세대 이슈없이 잘 사용하고 있습니다. 제조일도 한달 밖에 안되어서 좋았어요
835	835	2 모델 선택: 에어팟 프로 2세대 MTJV... 23/12/05	23/12/05	5	긍정	에어팟프로 3년정도쓰니까 꺾꺾소리나고 배터리도 금방 닳더라구요 ㅠ 이번에 c타입으로 새!
836	836	2 모델 선택: 에어팟 프로 2세대 MTJV... 23/12/17	23/12/17	4	긍정	제가 귀가 많이 작은 타입이라 이게 제대로 안 들어가서 제 기능을 못해요ㅠㅠ 저처럼 귀가 큰 사람도 귀가 많이 작은 타입이라 이게 제대로 안 들어가서 제 기능을 못해요ㅠㅠ 저처럼 귀가 큰 사람도

Saebyeol's PowerPoint



---

Part 3, 과제 수행 중 오류 해결 프로세스

---



## 리뷰 내용 내 작은 따옴표(') 존재 시 에러

- 리뷰 내용에 홑따옴표 (') 가 포함되어 있을 경우 SQL문 포맷 파괴로 에러가 발생
- 이스케이프 문자 (\)로도 해결이 안되는 경우가 있었음
- 홑따옴표를 아스키 코드 명령어로 치환하여 해결함

```
[94] 1 testing.input_data(all_list, url_list)
... 'REVIEW_LIST' 테이블에 갤럭시 버즈2프로 SM-R510 상품에 대한 8287
'REVIEW_LIST' 테이블에 Apple 2023 에어팟 프로 2세대 USB-C 충전
...
DatabaseError                                Traceback (most recent call last)
Cell In[94], line 1
----> 1 testing.input_data(all_list, url_list)

Cell In[91], line 40
    38 self.sql = "insert into REVIEW_LIST values({0}, '{1}', {2})"
    39 # 포맷에 맞게 데이터를 입력(리뷰 내용은 너무 긴 것은
--> 40 self.cursor.execute(self.sql.format('SEQ_REVIEW.n
    41                                     w[ '리뷰작성일'
    42 self.num_data += 1 # 데이터를 입력했을 경우 입력한
    44 print("'REVIEW_LIST' 테이블에 {0} 상품에 대한 {1}개의 리

DatabaseError: ORA-00917: 누락된 콤마
```

```
.replace("'", "''||CHR(039)||'")
```

REG_DATE	SCORE	POSITIVE_NEGATIVE	REVIEW
23/02/04	5	긍정	배송빠르고 가격대비좋아요
23/01/19	3	긍정	무선이어폰 6개째이고 노이즈캔슬링 기
23/02/09	5	긍정	배송 빠르고 좋아요
23/01/21	5	긍정	모두 만족한 상품입니다
23/01/15	5	긍정	아버지 선물로샀어요~ '잘끔'합니다
23/02/27	5	긍정	감사합니다감사합니다



## 지나치게 긴 리뷰 본문에 의한 에러

```

1 testing.input_data(all_list, url_list)
[120] 1.7s
... 'REVIEW_LIST' 테이블에 갤럭시 버즈2프로 SM-R510 상품
'REVIEW_LIST' 테이블에 Apple 2023 에어팟 프로 2세대
'REVIEW_LIST' 테이블에 삼성전자 갤럭시 버즈2 SM-R177
'REVIEW_LIST' 테이블에 갤럭시 버즈2프로 SM-R510 상품
...
-----
DatabaseError                                Traceback
Cell In[120], line 1
----> 1 testing.input_data(all_list, url_list)

Cell In[116], line 40
    38 self.sql = "insert into REVIEW_LIST va
    39 # 포맷에 맞게 데이터를 입력(리뷰 내용은
----> 40 self.cursor.execute(self.sql.format('S
    41                                     w[
    42 self.num_data += 1 # 데이터를 입력했을
    44 print("'REVIEW_LIST' 테이블에 {0} 상품에 대

DatabaseError: ORA-01704: 문자열이 너무 길니다

```

- 리뷰 내용이 너무 길 경우 문자열이 길다는 에러 발생
- 리뷰 내용을 담는 데이터형은 CLOB이지만 에러가 발생
- 확인 결과 컬럼 데이터형이 CLOB이라도 한번에 4000Byte (1000자) 이상의 문자열은 한번에 넣을 수 없었음.

# Part 3, 과제 수행 중 오류 해결 프로세스

## 지나치게 긴 리뷰 본문에 의한 에러

```
except:
    self.sql2 = "insert into REVIEW_LIST values({0}, '{1}', '{2}', TO_DATE('{3}', 'YY.MM.DD.'), {4}, '{5}', {6})"
    self.temp_txt = w['리뷰내용'].replace("'", "''||CHR(039)||'")
    self.n = len(self.temp_txt) // 1000
    self.li = []

    for z in range(0, self.n + 1):
        self.li.append(self.temp_txt[1000*z:1000*(z+1)])

    self.txt = ""
    for a, b in enumerate(self.li):
        self.txt += "TO_CLOB('{0}').format(b)
        if a < len(self.li) - 1:
            self.txt += "||"

    self.cursor.execute(self.sql2.format('SEQ_REVIEW.nextval', i+1, w['선택상품'],
                                         w['리뷰작성일'], w['별점'], w['긍정/부정'], self.txt))
```

- 해법은 문자열을 TO\_CLOB('문자열')의 형태로 4000Byte 단위로 쪼개서 각 TO\_CLOB 사이를 ||로 연결하여 데이터를 입력하는 것
- 기존의 방식대로 문자열을 추가하고 해당 제약으로 인해 에러 발생시 예외 처리를 통해 지나치게 긴 문자열을 1000자 단위로 쪼갠 후 TO\_CLOB을 적용시켜 하나의 문자열로 다시 합쳐서 insert 문 실행

## 문법 적용 예시

```
"TO_CLOB('asdfjk')||TO_CLOB('asdfkj')||TO_CLOB('hasdf')||TO_CLOB('askljd')||TO_CLOB('hfaskj')||TO_CLOB('fdf')"
```

REVIEW	LENGTH(REVIEW)
1 학생시절부터 함께하며 주인의 취직까지 옆을 지키다가 얼마 전 장렬하게 사망한 XM3을 눈물로 떠나보내고 슬픔만이 남은 그 빈자리에 XM...	2268
2 이전 모델인 wf-1000xm4 제품을 사용해본 사람으로서 전작과 달라진 점 위주로하여 장단점 및 제품의 특징 설명 드리겠습니다. 구매하실...	2202
3 장점 : 노캔 너무좋은 차소리 정말 많이 줄어들며(노래를경우 거의안들림), 바람부딪히는소리는 아예 없다고말하는게 맞을듯. 주변소리듣기...	1999
4 소니 블루투스이어폰은 10가지 이상 사용해 본 유저입니다 기존에 코드리스 제품중에는 최초로 LDAC코덱이 지원되는 wf-1000xm4가 가장...	1623
5 1. 박스구성 박스 구성은 라방 혜택 파우치가 동봉되어있네요. 파우치 재질은 사진 참고. 박스 구성은 저게 다입니다. 심플 또는 조출.....	1422
6 1.음질 : xm5>xm4> 에어팟 프로 2 2.주변 음 허용 : 에어팟프로2>xm5=xm4 3.착용감 : 에어팟프로2>xm4>xm5 4.연결성 : 에어팟프로2>xm...	1373
7 wf-1000xm4, WH-1000xm4 사용 중입니다. wf-1000xm4 배터리 오류 문제로 교체 받은 터라, 좀 더 사용할까 고민이 많았는데, 경...	1225
8 원래 전작 wf-1000xm4 사용했다가 유닛 크기와 생태계 연동성에 음질을 포기하고 버즈2 프로로 이동했었습니다. 이번에 xm5에서 멀티 ...	1218
9 에어팟 프로 1세대를 쓰고 있었는데, 여행 중 분실을 하게 되어 부득이하게 이어폰 구매하려고 알아보고 있었는데, 마침 wf-1000xm5가...	1217

- 1000자 이상의 리뷰가 정상 기입됨

# 과제 수행 중 오류 해결 프로세스

## 과제 수행 중 느낀 점 + 개선해야 할 점

### 느낀 점

- 수집하고자 하는 데이터 및 수집 페이지에 따라 API, 정적/동적 크롤링 등을 적절하게 사용해야 하는 점
- 파이썬으로 수집한 정보와 오라클 데이터베이스 연결 패키지로 생각보다 편리하고 빠르게 파이썬 - DB 연동을 수행할 수 있었음

### 개선할 점

- 데이터베이스 파트에서 테이블 구조, 테이블간 참조를 적절하게 사용하여 더 효율적인 데이터베이스 구축이 필요했음
- 데이터베이스 입력 시 SQL문에 대한 이해도 부족으로 다소 비효율적인 코드를 작성한 부분에 대한 개선이 필요했음



# Q&A

