

# 올리브영 립메이크업 페이지 리뷰 크롤링

| ChatGPT 생성 답변.

좋아! 올리브영 사이트에서 립 제품 리뷰 데이터를 수집하는 크롤링 코드를 작성해줄게. 일반적으로 이런 사이트는 JavaScript로 동적으로 콘텐츠를 불러오기에, Selenium을 사용하는 것이 효과적이야. 아래는 필요한 정보를 크롤링하는 예제 코드와 상세한 설명이야.

## 설치해야 할 라이브러리

```
pip install selenium
pip install webdriver-manager
```

## 크롤링 코드 (Selenium 기반)

```
from selenium import webdriver
from selenium.webdriver.common.by import By
from selenium.webdriver.chrome.service import Service
from webdriver_manager.chrome import ChromeDriverManager
import time
import pandas as pd

# 1. 크롬 브라우저 자동 설정
options = webdriver.ChromeOptions()
options.add_argument('--start-maximized')
driver = webdriver.Chrome(service=Service(ChromeDriverManager().install()), options=options)

# 2. 수집할 URL로 이동
url = "https://www.oliveyoung.co.kr/store/display/getMCategoryList.do?dispCatNo=100000100020006"
driver.get(url)
time.sleep(5) # 페이지 로딩 대기

# 3. 제품 페이지로 이동 (예시로 첫 번째 제품 클릭)
product_links = driver.find_elements(By.CSS_SELECTOR, '.prd_info .tx_name')
product_links[0].click()
time.sleep(3)

# 4. 리뷰 탭으로 이동
driver.switch_to.window(driver.window_handles[-1])
review_tab = driver.find_element(By.CSS_SELECTOR, 'a[href="#reviewInfo"]')
```

```

review_tab.click()
time.sleep(3)

# 5. 리뷰 수집
data = []
for i in range(3): # 리뷰 페이지 수만큼 반복
    reviews = driver.find_elements(By.CSS_SELECTOR, '.review_list div.review_cont')

    for review in reviews:
        try:
            user_id = review.find_element(By.CLASS_NAME, 'id').text
            skin_type = review.find_element(By.CLASS_NAME, 'user_info').text
            rating = len(review.find_elements(By.CSS_SELECTOR, '.review_rating .point.full')) # 별 개
            수

            criteria = [e.text for e in review.find_elements(By.CSS_SELECTOR, '.review_evaluation span')]
            review_text = review.find_element(By.CLASS_NAME, 'txt_inner').text

            data.append({
                'ID': user_id,
                'Skin_Type': skin_type,
                'Rating': rating,
                'Criteria': ' '.join(criteria),
                'Review_Text': review_text
            })
        except Exception as e:
            print("Error:", e)
            continue

# 다음 페이지 클릭
try:
    next_btn = driver.find_element(By.CSS_SELECTOR, '.pageing a.next')
    next_btn.click()
    time.sleep(2)
except:
    break

# 6. 저장
df = pd.DataFrame(data)
df.to_csv("oliveyoung_reviews.csv", index=False, encoding='utf-8-sig')

driver.quit()

```

## 💡 각 코드 설명

코드	설명
----	----

<code>webdriver.ChromeOptions()</code>	크롬 드라이버 실행 옵션 설정 (창 크기 조절 등)
<code>ChromeDriverManager().install()</code>	자동으로 최신 드라이버 설치
<code>driver.get(url)</code>	해당 URL로 브라우저 이동
<code>find_element</code> / <code>find_elements</code>	HTML 요소 탐색 (하나/여러 개)
<code>review.find_element(By.CLASS_NAME, 'id')</code>	작성자 ID 추출
<code>skin_type = review.find_element(By.CLASS_NAME, 'user_info').text</code>	피부타입 등 사용자 프로필 정보
<code>rating = len(...)</code>	별점은 꼭 찬 별의 개수로 판단
<code>review.find_element(By.CLASS_NAME, 'txt_inner').text</code>	실제 리뷰 텍스트
<code>driver.quit()</code>	크롬 드라이버 종료
<code>df.to_csv(...)</code>	데이터를 CSV 파일로 저장

## 참고 사항

- 리뷰 구조나 클래스명이 달라질 수 있으니, 개발자 도구(F12)로 구조를 확인하고 조정해야 해.
- 리뷰가 많은 제품을 수집하려면 반복 루프( `for i in range(3)` ) 범위를 늘려야 해.
- 동적 로딩이 많아서 `time.sleep()` 으로 로딩을 기다리는 방식이 필요하지만, 추후 `WebDriverWait` 으로 개선하면 더 안정적이야.

필요하다면 **BeautifulSoup 기반의 정적 크롤링**이나 **Ajax 요청 분석을 통한 API 접근 방식**도 도와줄 수 있어!

이 코드로 테스트해보고 잘 작동하는지 알려줘 :)