# 데이터 크롤링 (Selenium)

## 아이스크림 정보 수집

안다윤

2024 01 11 목

# 목적

Open Source 자동화 도구 중 하나인 Selenium을 사용하여

Chrome 브라우저를 자동으로 제어하는 기술을 통해

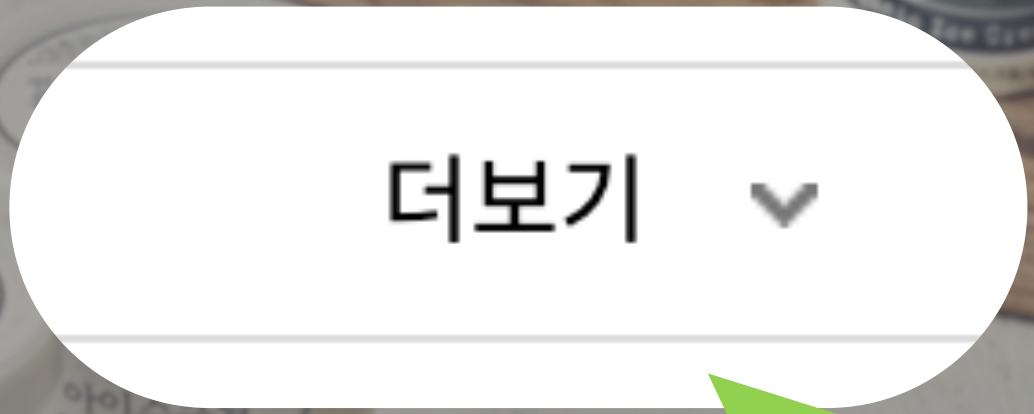
판매하는 아이스크림에 대한 정보가 있는

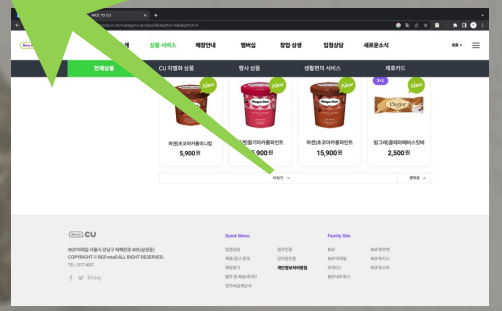
CU 웹 사이트에 접속하여 각 웹사이트에서 원하고자 하는 정보를 수집하여

Excel 파일과 MySQL 데이터베이스에 저장하기

# CU 의 특징

1. 한 화면에 전체 판매목록이 있지 않고 '더보기' 버튼이 총 7번 있음
2. 대부분 판매명에 아이스크림 브랜드와 이름이 중간에 ')'로 구분되어 있음 ('deffalo'와 'HEYROO' 브랜드 제외)
   -> 이름과 브랜드명 추출 용이



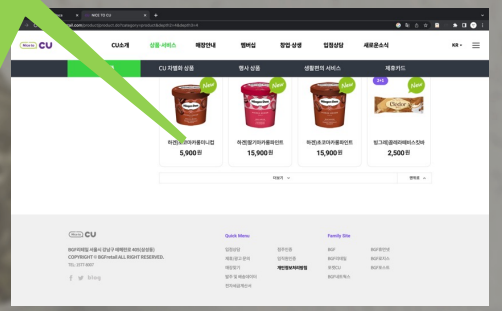
https://cu.bgfretail.com/product/product.do?category=product&depth2=4&depth3=4

# CU 의 특징

1. 한 화면에 전체 판매목록이 있지 않고 '더보기' 버튼이 총 7번 있음
2. 대부분 판매명에 아이스크림 브랜드와 이름이 중간에 ')'로 구분되어 있음 ('deffalo'와 'HEYROO' 브랜드 제외)
   -> 이름과 브랜드명 추출 용이

하겐)초코마카롱미니컵

5,900원



https://cu.bgfretail.com/product/product.do?category=product&depth2=4&depth3=4


수집 프로그램 구현

# Step 1:

# 필요한 Package 참조

```
import chromedriver_autoinstaller
from selenium import webdriver
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.common.by import By
import time

from bs4 import BeautifulSoup
from pandas import DataFrame

import pymysql
```

# Step 2:

# Chrome Browser 가동

```
chromedriver_autoinstaller.install()
driver = webdriver.Chrome()

# 크롬브라우저가 준비될 때 까지 최대 5초씩 대기
driver.implicitly_wait(5)
```

# Step 3:

# CU 아이스크림 페이지로 이동

```
driver.get("https://cu.bgfretail.com/product/product.do?category=product&depth2=4&depth3=4")
# 1초간 대기
time.sleep(1)
```

# Step 4:

# 모든 아이스크림을 조회하기 위해 '더보기' 버튼 누르고 5초간 대기 (총 7번)

```
a = 0
for a in range(7):
    if a < 7:
        more_button = WebDriverWait(driver, 3).until(
            lambda x: x.find_element(By.CSS_SELECTOR, ".prodListBtn-w")
        )
        more_button.click()
        # 5초간 대기
        time.sleep(5)
        a += 1
```

# Step 5:

# 내용 수집

## Step 5-1) 모든 정보 수집할 리스트 생성 및 아이스크림의 정보가 표시되는 html & css 추출

```
icecream_data = []

soup = BeautifulSoup(driver.page_source)
icecreamList = soup.select(".prod_wrap")
```

# Step 5:

# 내용 수집

## Step 5-2) 추출된 아이스크림 목록 수 만큼 반복하며 이름, 브랜드명, 가격 가져오기

```
for i in icecreamList:
    # 이름
    name = i.select(".name")
    if name:
        nameValue = name[0].text.strip()
        if ')' in nameValue:
            indexBeforeName = nameValue.index(')')
            nameValue = nameValue[indexBeforeName + 1:]
        elif 'delaffe' in nameValue:
            nameValue = nameValue[7:]
        elif 'HEYROO' in nameValue:
            nameValue = nameValue[6:]
        else:
            nameValue = nameValue
    else:
        nameValue = None

    # 브랜드명
    brand = i.select(".name")
    if brand:
        brandValue = brand[0].text.strip()
        if ')' in brandValue:
            indexBeforeName = brandValue.index(')')
            brandValue = brandValue[:indexBeforeName]
        elif 'delaffe' in brandValue:
            brandValue = 'delaffe'
        elif 'HEYROO' in brandValue:
            brandValue = 'HEYROO'
        else:
            brandValue = None
    else:
        brandValue = None

    # 가격
    price = i.select(".price")
    if price:
        priceValue = int(price[0].text.strip().replace('원', '').replace(',', ''))
    else:
        priceValue = None

    # 결과 병합
    resultDic = {'이름': nameValue,
                 '브랜드': brandValue,
                 '가격(원)': priceValue}
    icecream_data.append(resultDic)
```

# Step 6:

# Excel 파일로 수집한 내용 저장

```
df = DataFrame(icecream_data)
df.to_excel("icecream_list_cu.xlsx")
```

# 수집한 데이터 확인

## icecream_list_cu.xlsx

| | A | B | C | D |
|---|---|---|---|---|
| 1 | | 이름 | 브랜드 | 가격(원) |
| 2 | 0 | 생우유파인트 | 라라스윗 | 8900 |
| 3 | 1 | 끌레도르쿠키초코 | 빙그레 | 2500 |
| 4 | 2 | 찰떡아이스황치즈 | 롯데 | 2200 |
| 5 | 3 | 마카다미아파인트 | 나뚜루 | 14900 |
| 6 | 4 | 마카다미아미니컵 | 나뚜루 | 4800 |
| 7 | 5 | 엑설런트딸기소다 | 빙그레 | 11000 |
| 8 | 6 | 생초코바 | 라벨리 | 2500 |
| 9 | 7 | 서울우유밀꾸즈 | 프레시 | 2900 |
| 10 | 8 | 바닐라초코바 | 라라스윗 | 3300 |
| 11 | 9 | 초콜릿초코바 | 라라스윗 | 3300 |
| 12 | 10 | 미니니녹차우유샌드 | 서주 | 2500 |
| 13 | 11 | 커피마루샌드 | 해태 | 2200 |
| 14 | 12 | 미니니초코모나카 | 서주 | 2500 |
| 15 | 13 | 망고코코넛포멜로 | 동그린 | 14900 |
| 16 | 14 | 끌레솔티바닐파인 | 빙그레 | 14900 |
| 17 | 15 | 끌레밀크티파인트 | 빙그레 | 14900 |
| 18 | 16 | 초코비파인트 | 초코 | 12400 |
| 19 | 17 | 딸기치즈케이크도넛 | 서주 | 2000 |
| 20 | 18 | 쌀모나카 | 라라스윗 | 3300 |
| 21 | 19 | 빵또아초코쿠앤크 | 빙그레 | 2200 |
| 22 | 20 | 슈크림붕어싸만코 | 빙그레 | 2200 |
| 23 | 21 | 부라보콘피스타치오 | 해태 | 2200 |
| 24 | 22 | 체스트넛미니컵 | 하겐 | 5900 |
| 25 | 23 | 피스타치오미니컵 | 하겐 | 5900 |
| 26 | 24 | 체스트넛파인트 | 하겐 | 15900 |
| 27 | 25 | 피스타치오파인트 | 하겐 | 15900 |
| 28 | 26 | 1000샌드밀크 | 서주 | 1000 |
| 29 | 27 | 슈퍼콘초코바나나 | 빙그레 | 2200 |
| 30 | 28 | 끌레끼리치즈멀티 | 빙그레 | 14900 |
| 31 | 29 | 덴마크초코초코콘 | 제스 | 2500 |

...

| | A | B | C | D |
|---|---|---|---|---|
| 277 | 275 | 스크류바 | 롯데 | 1500 |
| 278 | 276 | 더위사냥액티브 | 빙그레 | 1800 |
| 279 | 277 | 더블비얀코 | 롯데 | 2200 |
| 280 | 278 | 설레임밀크쉐이크 | 롯데 | 2200 |
| 281 | 279 | 쿠앤크바 | 빙그레 | 1500 |
| 282 | 280 | 빅구슬복숭아키위 | 동학 | 1800 |
| 283 | 281 | 구구크러스터홈 | 롯데 | 6000 |
| 284 | 282 | 빵빠레초코 | 롯데 | 2200 |
| 285 | 283 | 우유팥빙수컵 | HEYROO | 2800 |
| 286 | 284 | 찰떡아이스 | 롯데 | 2200 |
| 287 | 285 | 돼지바 | 롯데푸드 | 1500 |
| 288 | 286 | 옥동자바 | 롯데 | 1500 |
| 289 | 287 | 와샤베트 | 롯데 | 2200 |
| 290 | 288 | 파르페플러리 | HEYROO | 2700 |
| 291 | 289 | 파르페딸기 | HEYROO | 2700 |
| 292 | 290 | 쌍쌍바 | 해태 | 1500 |
| 293 | 291 | 비비빅 | 빙그레 | 1500 |
| 294 | 292 | 수박바 | 롯데 | 1500 |
| 295 | 293 | 구구콘 | 롯데 | 2200 |
| 296 | 294 | 죠스바 | 롯데 | 1500 |
| 297 | 295 | 빠삐코 | 롯데 | 1800 |
| 298 | 296 | 메로나 | 빙그레 | 1500 |
| 299 | 297 | 부라보콘바닐라 | 해태 | 2200 |
| 300 | 298 | 본가찰옥수수 | 롯데 | 2200 |
| 301 | 299 | 초콜릿파인트 | 나뚜루 | 14900 |
| 302 | 300 | 딸기파인트 | 나뚜루 | 14900 |
| 303 | 301 | 녹차파인트 | 나뚜루 | 14900 |
| 304 | 302 | 초코미니컵 | 나뚜루 | 4800 |
| 305 | 303 | 바닐라아몬드바 | 나뚜루 | 4800 |
| 306 | 304 | 딸기미니컵 | 나뚜루 | 4800 |
| 307 | 305 | 녹차바 | 나뚜루 | 4800 |
| 308 | 306 | 녹차미니컵 | 나뚜루 | 4800 |
| 309 | | | | |

# Step 7:

# MySQL 데이터베이스에 내용 저장

## Step 7-1) 데이터베이스에 접속

```
dbcon = pymysql.connect(host="127.0.0.1",    # 서버주소 - "127.0.0.1" 또는 "localhost"
                        port = 3406,         # 포트번호
                        user = "root",       # 계정이름
                        password = "1234",   # 비밀번호
                        db = "icecream",     # 데이터베이스 이름
                        charset = "utf8"     # 인코딩
                        )
```

Step 7:

# MySQL 데이터베이스에 내용 저장

## Step 7-2) 커서 객체 생성

```
cursor = dbcon.cursor()
```

수집 프로그램 구현

# Step 7:

# MySQL 데이터베이스에 내용 저장

## Step 7-3) 데이터 입력

```
웹페이지_이름 = ['cu']

for 웹페이지 in 웹페이지_이름:
    # 테이블 생성
    create_table = "CREATE TABLE %s (id INT auto_increment, name VARCHAR(10), brand VARCHAR(10), price INT, PRIMARY KEY (id))" % 웹페이지
    cursor.execute(create_table)          # 쿼리문 실행

    # 각 테이블 (CU, EMART)에 해당 아이스크림 정보 입력
    for 아이스크림 in range(len(icecream_data)):
        이름 = icecream_data[아이스크림]['이름']
        브랜드 = icecream_data[아이스크림]['브랜드']
        가격 = icecream_data[아이스크림]['가격(원)']
        insert = "INSERT INTO %s (name, brand, price) VALUES ('%s', '%s', %d)" % (웹페이지, 이름, 브랜드, 가격)
        cursor.execute(insert)          # 쿼리문 실행
```

# 수집한 데이터 확인

## MySQL - icecream

```
for 웹페이지 in 웹페이지_이름:
    조회 = "SELECT * FROM %s" % 웹페이지
    cursor.execute(조회)          # 쿼리문 실행
    result = cursor.fetchall()   # 데이터 반환받기
    df = DataFrame(result)
    df.set_index(0, inplace= True)
    df.index.name = 'id'
    df.rename(columns = {1: '이름', 2: '브랜드', 3: '가격(원)'}, inplace=True)

print(df)
```

| id | 이름 | 브랜드 | 가격(원) |
|---|---|---|---|
| 1 | 생우유파인트 | 라라스윗 | 8900 |
| 2 | 끌레도르쿠키초코 | 빙그레 | 2500 |
| 3 | 찰떡아이스황치즈 | 롯데 | 2200 |
| 4 | 마카다미아파인트 | 나뚜루 | 14900 |
| 5 | 마카다미아미니컵 | 나뚜루 | 4800 |
| .. | ... | ... | ... |
| 303 | 초코미니컵 | 나뚜루 | 4800 |
| 304 | 바닐라아몬드바 | 나뚜루 | 4800 |
| 305 | 딸기미니컵 | 나뚜루 | 4800 |
| 306 | 녹차바 | 나뚜루 | 4800 |
| 307 | 녹차미니컵 | 나뚜루 | 4800 |

[307 rows x 3 columns]


감사합니다
데이터 수집: CU 아이스크림
안다윤
2024 01 11 목