Selenium 활용 데이터 수집 - 요일별 네이버 웹툰 목록

개요

selenium

웹 브라우저를 제어할 수 있는 파이썬 패키지

chromedriver

selenium과 Google Chrome 브라우저를 연결하는 프로그램

파이썬 스스로 chromedriver를 내려받도록 하기 위해 `chromedriver_autoinstaller` 패키지가 필요함

설치해야 하는 패키지

```
$ pip install --upgrade chromedriver_autoinstaller
$ pip install --upgrade selenium
```

#01. 패키지 참조

```
# ChromeDriver 자동 설치 모듈
import chromedriver_autoinstaller
# Chrome을 제어하기 위한 객체
from selenium import webdriver
# Chrome이 웹 페이지 로딩을 완료 할 때까지 최대 n초간 대기하는 기능.
from selenium.webdriver.support.ui import WebDriverWait
from bs4 import BeautifulSoup
from pandas import DataFrame
# 파이썬 프로그램에 지정된 시간동안 랙을 거는 기능을 위해 사용
import time
```

#02. 크롬브라우저 가동하기

```
# 크롬드라이버 자동 설치
chromedriver_autoinstaller.install()

# 크롬드라이버를 통해 크롬을 실행시킴
# -> driver 객체는 Chrome 자체
driver = webdriver.Chrome()

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

```
# 요일별 네이버 웹툰에 접근하기 위한 변수값
# -> URL을 분석하여 얻어낸 값
```

```
params = ['mon', 'tue', 'wed', 'thu', 'fri', 'sat', 'sun', 'dailyPlus']

# 네이버 웹툰의 주소 형식
naverWebtoonUrl = "https://comic.naver.com/webtoon?tab={0}"
```

```
# 수집된 결과가 누적될 빈 리스트
naverWebtoolData = []

# 요일별 반복
for p in params:
    # 특정 요일의 네이버 웹툰 페이지
    url = naverWebtoonUrl.format(p)
    print(url)

    # 크롬브라우저로 방문시킴
    driver.get(url)
    time.sleep(1)

    # 브라우저에 표시되는 전체 코드를 추출 --> bs4 객체로 변환
    soup = BeautifulSoup(driver.page_source)
    #print(soup)

    # 웹툰이 표시되는 부분만 추출
    webtoolList = soup.select(".ContentList__content_list--q5KXY > .item")
    #print(webtoolList)

    # 추출된 웹툰 목록 수 만큼 반복
    for w in webtoolList:
        # 포스터 URL 가져오기
        poster = w.select(".Poster__image--d9XTI")

        # 가져온 이미지가 존재하고, src속성이 있다면?
        if poster and "src" in poster[0].attrs:
            posterValue = poster[0].attrs['src']
        else:
            posterValue = None

        #print(posterValue)

        # 웹툰의 URL 가져오기
        url = w.select(".Poster__link--sopnC")

        if url and "href" in url[0].attrs:
            urlValue = url[0].attrs['href']

            if urlValue.find("https://comic.naver.com") == -1:
                urlValue = "https://comic.naver.com" + urlValue
        else:
            urlValue = None

        #print(urlValue)

        # 웹툰 제목 가져오기
        title = w.select(".ContentTitle__title--e3qXt > .text")

        if title:
```

```
            titleValue = title[0].text.strip()
        else:
            titleValue: None
        #print(titleValue)

        # 작가 이름 가져오기
        author = w.select(".ContentAuthor__author--CTAAP")

        if author:
            authorValue = author[0].text.strip()
        else:
            authorValue = None
        #print(authorValue)

        # 별점 가져오기
        rating = w.select(".Rating__star_area--dFzsb > .text")

        if rating:
            ratingValue = rating[0].text.strip()
        else:
            ratingValue = None
        #print(ratingValue)

        # 결과 병합
        resultDic = {"요일": p, "제목": titleValue, "작가": authorValue, "별점": ratingV
        #print(resultDic)
        naverWebtoolData.append(resultDic)

df = DataFrame(naverWebtoolData)
df.to_excel("요일별_네이버_웹툰.xlsx")
df
```