[Python] requests와 BeautifulSoup을 활용한 디시인사이드 크롤링

2020. 1. 9. 09:50 · Python 데이터 분석/Python_Crawling

읽기전에

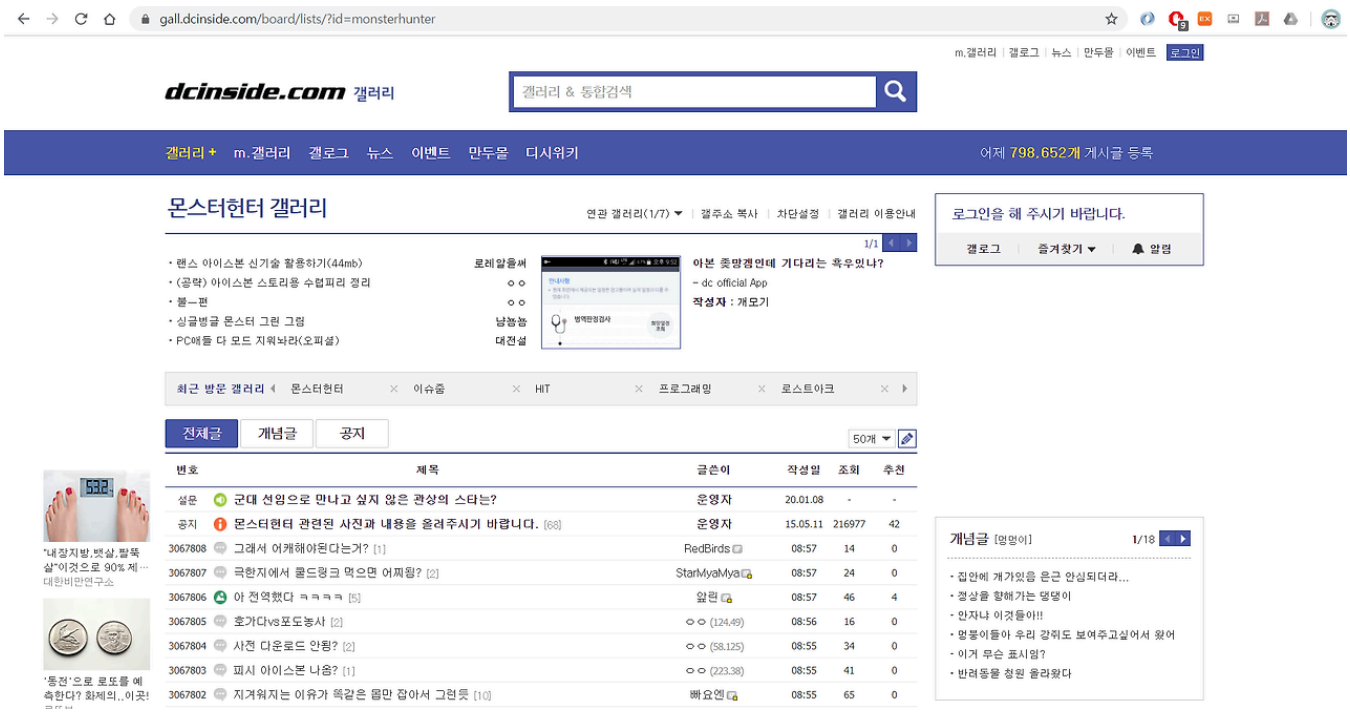
실습환경은 Jupyter notebook에 최적화 되어 있습니다.

Python 크롤링에 자주 사용하는 requests 모듈과 bs4 패키지 내 BeautifulSoup을 활용하여 크롤링을 해보겠습니다.

저는 디시인사이드(DC) 사이트를 크롤링 해볼 것입니다.

```
# moduel import
import requests
from bs4 import BeautifulSoup
```

먼저 수집할 갤러리를 결정하고, 사이트를 들어가 봅시다.



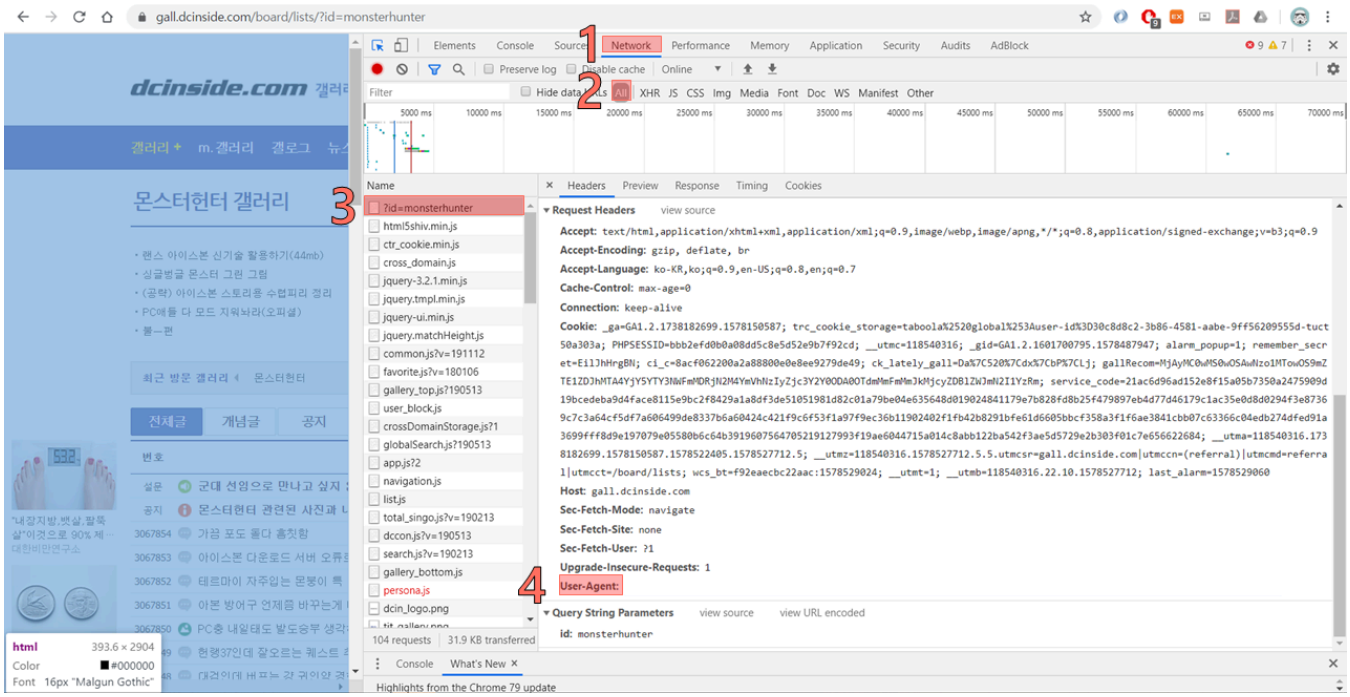
코드를 작성하지 전에 먼저 수집할 대상을 정해봅시다. 보통 수집할 대상은 목적을 위해 어떤 데이터들이 필요한지를 고려하여 결정합니다. 저는 **글 번호, 글 제목, 글쓴이, 본문, 작성일, 조회 수, 추천 수**를 수집하기로 결정했습니다.

크롤링의 기초는 URL을 파악하는 것에서부터 시작합니다. 수집할 사이트의 URL을 확인해봅시다.
https://gall.dcinside.com/board/lists?id=monsterhunter

url에서 ? 뒤에 나타나는 것은 파라미터를 의미합니다.
requests모듈 사용을 위해 아래와 같이 기본이 되는 url과 파라미터를 설정해 줍니다.

headers에는 내가 누구인지를 알려줍니다. requests 모듈에서 요청을 보내는 것은 robot이 보내는 것입니다. 따라서 몇몇 웹사이트의 경우에는 robot이 접근할 시에 올바른 정보를 전달하지 않습니다. 따라서 robot이 아닌 특정한 사용자가 요청을 보내는 것으로 인식하도록 header를 설정해주어야 합니다.

User-Agent는 크롬기준 사이트에서 F12를 누르고 다음 사진대로 클릭하면 User-Agent에 대한 정보를 알 수 있습니다.
현재 저는 가려 놓았지만, 실제로는 4번 위치에 User-Agent를 나타내는 여러 정보들이 쓰여있습니다. 전부 복사합니다.



이제 아래 서식에 맞게 User-Agent를 채워주고 요청을 보냅니다.

headers = {'User-Agent' : 복사한 User-Agent 정보 붙여넣기 }

```
BASE_URL = "https://gall.dcinside.com/board/lists"

# 파라미터 설정
params = {
            'id': 'monsterhunter',}

# 헤더 설정
headers = {
    'User-Agent' : }



resp = requests.get(BASE_URL, params=params, headers=headers)
```

requests내에 있는 get 함수는 보내면 BASE_URL과 파라미터를 조합하여 "https://gall.dcinside.com/board/lists/?id=monsterhunter" 라는 URL로 요청을 보내 크롤링을 수행합니다.

먼저 잘 수집했는지 확인해봅시다.

```
resp

# <Response [200]>
```

숫자 200은 요청을 정상적으로 보내고 정상적인 답변을 받아왔다는 것을 의미합니다.

```
resp.url

# 'https://gall.dcinside.com/board/lists?id=monsterhunter'
```

url도 정상적으로 조합되었습니다.

```
resp.content

# b'<!DOCTYPE html>\r ... \r\n</html>\r\n'
```

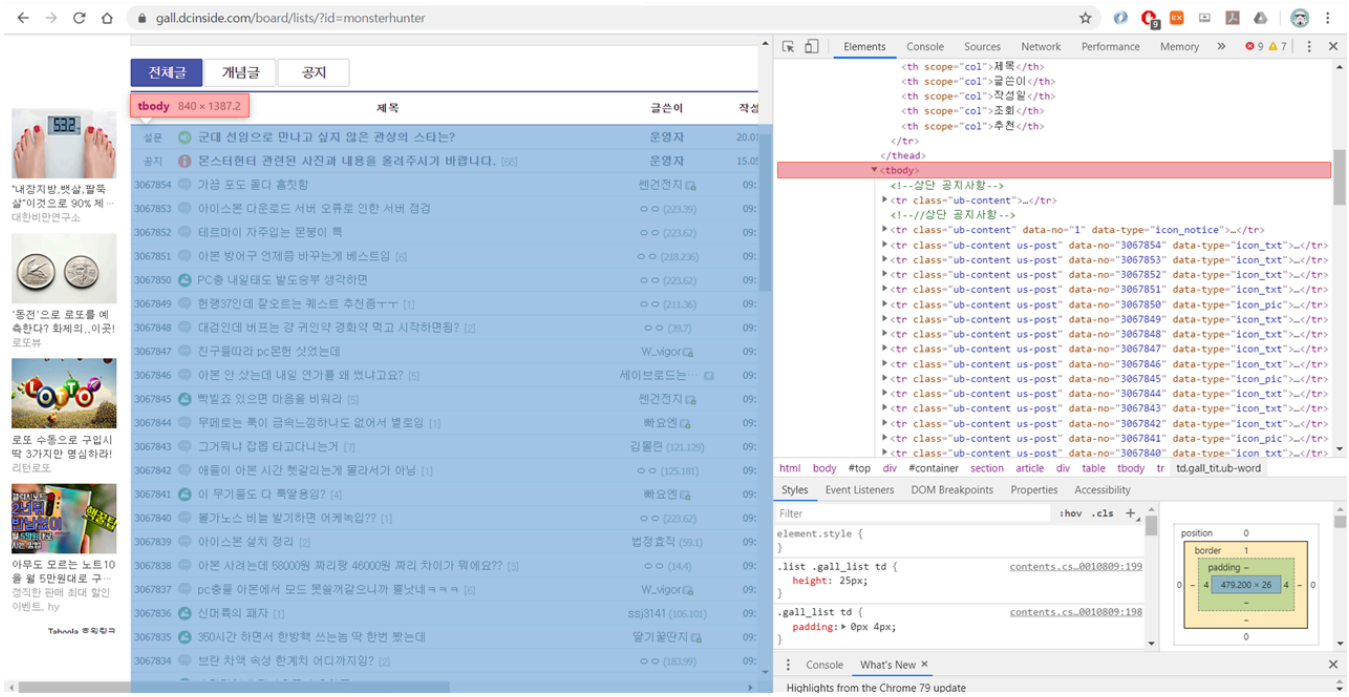
content를 확인해보면 잘 byte형식의 html을 잘 긁어온 것을 확인할 수 있습니다. 이제 BeautifulSoup를 사용해서 이 html을 파싱해봅시다.

```
soup = BeautifulSoup(resp.content, 'html.parser')
soup

# <!DOCTYPE html>

# <html lang="ko">
# <head>
# <meta charset="utf-8"/>
# <meta content="IE=edge" http-equiv="X-UA-Compatible"/>
# <meta content="no" http-equiv="imagetoolbar"/>
# <meta content="kr" name="content-language"/>
# <meta content="8_SyZg2Wg3LNnCmFXzETp7ld4yjZB8ny17m8QsYsLwk" name="google-site-
verification"/>
# <meta content="디시인사이드" name="author"/>
# <meta content="몬스터헌터 갤러리" name="title"/>
# <meta content="온라인게임,헌팅 액션,캡콤,플레이스테이션" name="description"/>
# <meta content="website" property="og:type"/>
# <meta content="몬스터헌터 갤러리" property="og:title"/>
...
```

깔끔하게 파싱한 것을 확인할 수 있습니다.
모든 사이트의 내용을 긁어올 필요는 없으니, 수집하고자하는 대상이 있는 부분만을 찾아봅니다. 크롬기준 f12를 눌러 커서를 가져다 대보면서 어느 부분을 수집해야하는지를 알아봅니다. 디시인사이드의 경우, <tbody>라는 태그 안에 제가 수집하고자하는 모든 정보들이 있습니다.




```
contents = soup.find('tbody').find_all('tr')
```

soup.find 함수를 사용하여 <tbody> 부분만을 가져온 다음, 그 안에 속하는 모든 <tr> 태그를 find_all 함수를 사용하여 찾아냅니다.

이제 본격적으로 크롤링된 데이터에서 우리가 필요한 것을 추출할 코드를 짜봅시다.
저는 항상 다음과 같은 순서를 따릅니다.

> 1. 원하는 내용이 있는 태그찾기
> 2. 태그에서 내용을 text로 추출할 수 있는가 없는가를 판별
> 3. text인 경우 그대로 내용 추출
> 4. text가 아닌 경우 태그를 딕셔너리 형태로 변환후 추출
> 5. None 값이 있는 경우 조건문을 통해 pass

이를 바탕으로 코드를 짜면 다음과 같습니다.

```
# 한 페이지에 있는 모든 게시물을 긁어오는 코드
for i in contents:
    print('-'*15)

    # 제목 추출
    title_tag = i.find('a')
    title = title_tag.text
    print("제목: ", title)
```

```
    # 글쓴이 추출
    writer_tag = i.find('td', class_='gall_writer ub-writer').find('span',
class_='nickname')
    if writer_tag is not None: # None 값이 있으므로 조건문을 통해 회피
        writer = writer_tag.text
        print("글쓴이: ", writer)

    else:
        print("글쓴이: ", "없음")

    # 유동이나 고닉이 아닌 글쓴이 옆에 있는 ip 추출
    ip_tag = i.find('td', class_='gall_writer ub-writer').find('span', class_='ip')
    if ip_tag is not None:  # None 값이 있으므로 조건문을 통해 회피
        ip = ip_tag.text
        print("ip: ", ip)

    # 날짜 추출
    date_tag = i.find('td', class_='gall_date')
    date_dict = date_tag.attrs

    if len(date_dict) is 2:
        print("날짜: ", date_dict['title'])

    else:
        print("날짜: ", date_tag.text)
        pass

    # 조회 수 추출
    views_tag = i.find('td', class_='gall_count')
    views = views_tag.text
    print("조회수: ", views)

    # 추천 수 추출
    recommend_tag = i.find('td', class_='gall_recommend')
    recommend = recommend_tag.text
    print("추천수: ", recommend)
```

코드를 실행하면 한 페이지에 있는 모든 게시물의 제목, 글쓴이, 날짜, 조회 수, 추천 수를 추출하게 됩니다.

실행결과 중 일부는 다음과 같습니다.

```
---------------
제목:  어떤 위기도 유쾌하게 극복할 듯한 스타는?
글쓴이:  없음
날짜:  20.04.28
조회수:  -
추천수:  -
---------------
제목:  월간디시 5월호 : 아싸는 이제 제겁니다.
글쓴이:  없음
날짜:  20.04.29
조회수:  -
추천수:  -
---------------
제목:  몬스터헌터 관련된 사진과 내용을 올려주시기 바랍니다.
글쓴이:  없음
날짜:  2015-05-11 15:01:00
조회수:  255687
추천수:  71
---------------
제목:  모넌 질문좀
```

```
글쓴이:  ㅇㅇ
ip:  (211.43)
날짜:  2020-04-29 22:39:16
조회수:  4
추천수:  0
---------------
제목:  맘타 피갤집 당장 와라 n6nhkZN4WyHE
글쓴이:  ㅇㅇ
ip:  (14.55)
날짜:  2020-04-29 22:39:06
조회수:  3
추천수:  0
```


TAG BeautifulSoup, Python, requests, 디시인사이드, 크롤링, 파이썬

관련글

[Python] 윈도우 작업 스케줄러를 활용한 웹…
2020.04.30

댓글 16

셰일 · 2020.04.19 21:44 신고

굉장히 도움되는 글이었습니다. 파이썬 암것도 모르는데 이해하기 쉽네요 ㅎㅎ
다음 글들도 기다려지네요!!

수정/삭제|답글

11 · 2020.09.24 12:24 신고

유저 에이전트에서 신텍스 에러 뜨는데 어캐해야하나요,, 쥬피터 노트북 사용하고 있고 뷰티풀 숲 리퀘스트 다했습니다

수정/삭제|답글

ㄴ **닭강정호** · 2020.09.24 12:48 신고

신택스 에러는 오타가 있다는 뜻입니다. 한번 찬찬이 살펴보세용

수정/삭제

Ladun · 2020.09.29 19:53 신고

디시인사이드로 여러번 request.get을 보내면 차단당하기도 하나요?

수정/삭제|답글

ㄴ **닭강정호** · 2020.09.29 19:54 신고

맞습니다. 단 시간에 너무 많이 요청을 보내면 일시적으로 차단하더라구요

수정/삭제

ㅇㅇ · 2020.10.27 00:58 신고

많은 도움 되었습니다
정말 감사합니다.

수정/삭제|답글

ㄴ **닭강정호** · 2020.11.01 09:35 신고

도움이 되었다니 다행입니다 ㅎㅎ

수정/삭제

dorudoru · 2020.11.19 10:38 신고

많은 도움이 되었습니다. 혹시 이상황에서 여러 페이지를 크롤링하려면 어떻게 수정해야할까요?

수정/삭제|답글

익명 · 2021.01.20 13:04

비밀댓글입니다.

수정/삭제

무명씨 · 2021.01.20 12:41 신고

구글링해서 나오는 디씨 크롤링 게시물 중 정확하게 작동하는 최고의 게시글이네요. 큰 도움 받았습니다. 감사합니다.

수정/삭제|답글

닭강정호 · 2021.01.20 13:04 신고

도움이 되었다니 다행입니다 ㅎㅎ

수정/삭제

채치치 · 2021.01.24 17:57 신고

다음 페이지도 크롤링하려면 어떻게 해야 하나요?!?!?!

수정/삭제|답글

익명 · 2021.01.25 10:25

비밀댓글입니다.

수정/삭제

익명 · 2021.03.04 00:19 신고

저도 다음 페이지를 크롤링하려고 하는데, 1페이지 내용만 계속 반복됩니다. 어떻게 하면 되나요?

수정/삭제|답글

놀라움의연속 · 2021.05.07 01:58 신고

저도 다음 페이지 크롤링하는 방법 알고 싶습니다!! ㅜㅜ 정말 유익한 페이지에요! 감사해요

수정/삭제|답글

크하하 · 2021.05.26 13:51 신고

soup.find 에서 자꾸 none값이 반환되는데 다른방법을 써야할까요? request에선 정상적으로 응답이 되고 url도 정상적으로 가져오는데 tbody를 찾는 코드에선 none값이 반환되네요 ㅠ

수정/삭제|답글

이름 비밀번호

댓글을 입력해주세요.

비공개 댓글 남기기