

Web Scraping

웹에서 정보를 자동으로 수집하기!

도전과제 풀어보기! 



도전과제 1

앞선 “리디북스 신간 도서 스크래핑”에서 1페이지 상단 5개만 가져왔습니다

-  소스를 수정해서 1페이지에 있는 모든 도서의 제목(설명 X)을 가져와 보세요!

hint ) items[:5] → 일정 부분(?)이 이상할거에요 → 왜 이상한지, 뭐가 다른지 개발자도구로 잘 찾아보세요!

```
...<li class="fig-vj0uh1"><...></li>
▶<li class="fig-vj0uh1"><...></li>
...<li class="fig-vj0uh1"><...></li>
    <div class="b-29qxzc">Elle 2025년 4월호 (Elle 편집부, HLL중
        앙)</div>
    </li>
</ul>
```



```
▶ <li class="fig-vj0uh1">...</li>
▼<li class="fig-vj0uh1">
  ▼<div class="b-ona580"> flex == $0
    ▶ <div class="b-lyemeb">...</div> flex
    ▶ <div class="b-o96tbl">...</div> flex
      <a href="/books/4955000367? rdt.sid=newReleases_general&rdt.idx=60" class="b-1hkmplz"></a>
    </div>
  </li>
</ul>
▶ <div data-testid="pagination">...</div>
</sections>
```



도전과제 2

✓ 1페이지 말고 2,3 다른 페이지들에 있는 정보도 가져오고 싶지 않나요?

hint💡) <https://ridibooks.com/new-releases/general?order=POPULARITY&page=1>
페이지를 바꿔가면서 url 틀린그림 찾기를 해보세요!

https://~~~.com/~~/general?order=POPULARITY&page=1

URL : Uniform Resource Locator 는 인터넷 상에서 리소스(파일) 위치를 나타내는 주소

URL 의 구성요소

{포로토콜}://{호스트(아이피 주소, 도메인 네임)}:{포트번호}/{... 리소스 경로 ...}?{쿼리스트링}

ex) **http://127.0.0.1:3000/api/v1/movie?page=1&release_date=2025-03-25**

여기서 쿼리스트링은 ? 뒤에 key-value 형태로 전달하고 & 으로 구분하여 여러개 전달 가능!

위에서는 page 에는 “1”, release_date 는 “2025-03-25”라는 문자열이 들어가고

의미는 “1 페이지에 있는 2025년 3월 25일에 개봉하는 영화정보를 불러와!”

단, value 는 다 문자열 형태로 전달된다.

이번주 ...

1. **css 선택자**라는 것에 대해서 알아보겠습니다.

```
soup.select('main > section > ul.fig-1pep8jc.eis6k7i0 > li')
```

2. 다양한 BeautifulSoup 함수 사용방법을 알아보겠습니다.

3. 네이버 신작 웹툰 목록 스크래핑 해보기

CSS 선택자

'이것'의 의미를 알아보자!

👉 `soup.select('main > section > ul.fig-1pep8jc.eis6k7i0 > li')`

CSS 선택자

CSS 는 HTML 에 디자인을 입히기 위한 스타일 언어

```
p {  
    color: red;  
}
```

이름이 **p** 라는 태그를 선택하여 텍스트 색깔을 red 로 설정

선택자의 종류

type 선택자 : 태그 이름을 선택자로 사용

id 선택자 : #~~ 으로 태그에 부여된 id 를 참조할 수 있다. 중복된 id 값 부여 불가능!

class 선택자 : .~~ 으로 태그에 부여된 class 를 선택할 수 있다. 여러 개의 class 값 부여 가능!

속성 선택자 : input[checked] 대괄호 [] 안에 속성 명을 넣어서 사용 → checked 라는 속성을 가진 input 태그

CSS 선택자

관계를 나타내는 선택자

하위 선택자 : 내부에 있는 모든 요소를 대상으로 탐색

.red p → 클래스가 red 인 태그 내에 있는 모든 p 태그

자식 선택자 : 직계 자식에 있는 모든 요소를 대상으로 탐색

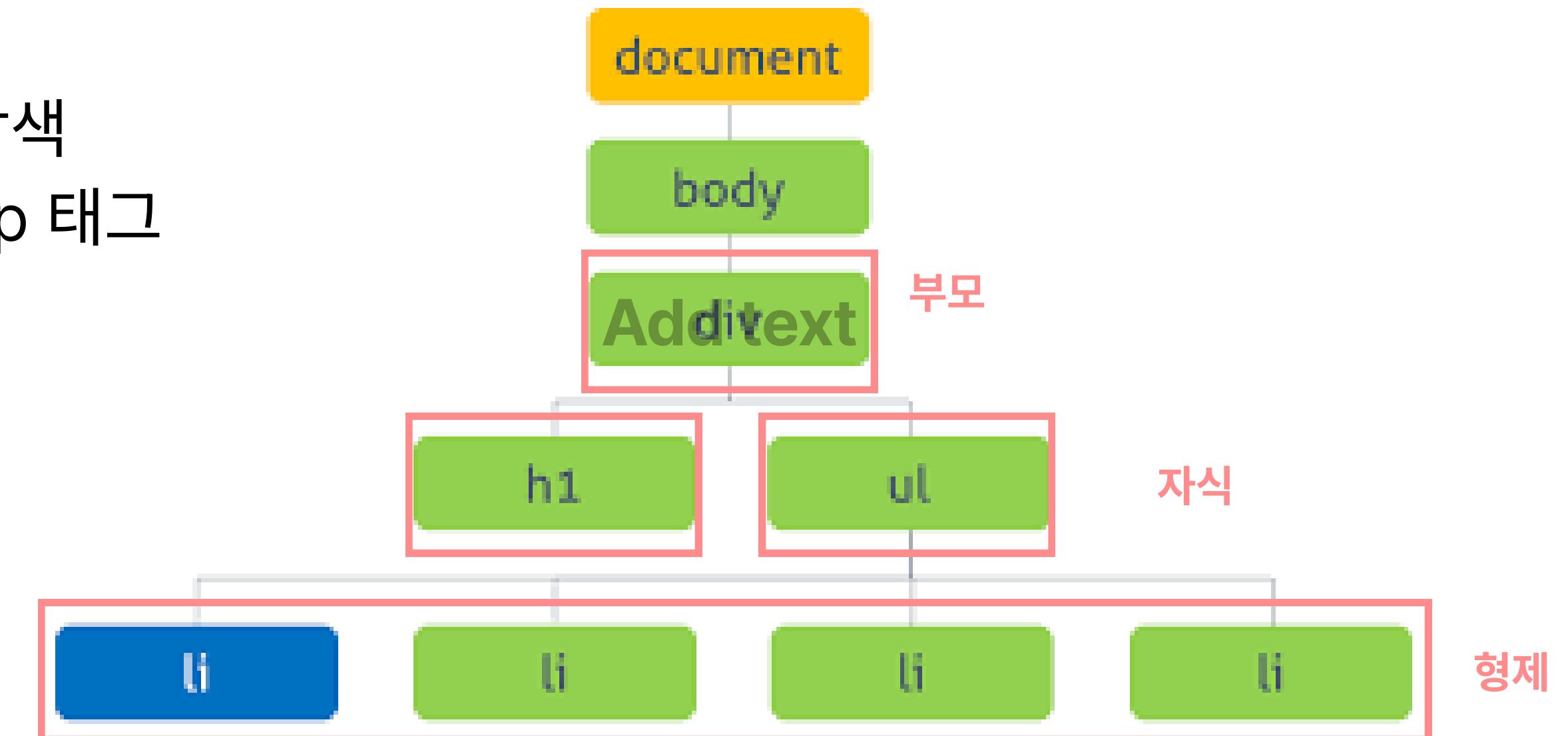
.red > p → 클래스가 red 인 태그 내에 있는 직계 자식 p 태그

인접 선택자 : 현재 요소 바로 뒤에 나오는 요소만 선택

.red + p → 클래스가 red 인 태그 바로 다음 p 태그

형제 선택자 : 현재 요소의 형제 태그를 선택

.red + p → 클래스가 red 인 태그 바로 다음 p 태그



CSS에 대한 상세한 내용

https://www.w3schools.com/css/css_selectors.asp

CSS 선택자

태그 이름이 main 에 직계 자식 중 태그 이름이 section 의 자식 중에서
태그 이름이 ul 이고 클래스가 fig-1pep8jc eis6k7i0 두 개의 값을 가지는 태그
의 자식 중에서 모든 li 태그를 가져와라!

```
soup.select('main > section > ul.fig-1pep8jc.eis6k7i0 > li')
```



flukeout.github.io



리디북스 신간 독서 코드 개선하기

CSS 선택자에 대한 이해를 적용하기 😮

리디북스 CSS 선택자를 개선해보자!

우리가 1주차에 작성한 코드는 개발자 도구에서 Copy! 하여 가져온 좀 Smart~하지 못한 선택자입니다.

```
soup.select('main > section > ul.fig-1pep8jc.eis6k7i0 > li')
```

저기 보이는 ul 뒤에 나오는 클래스를 보시면 의미있는 단어가 아니라
Web Framework 에서 스타일 적용을 위해 적용한 난수입니다.

그래서 저런 클래스를 이용해서 element 를 선택하게 된다면 웹사이트의 변경에 매우 취약하게 되어
코드를 자주 수정해야할 수 있습니다. 하지만 구조를 파악해 의미있는 선택자를 사용하게 된다면 수정을 거의 하
지 않게 되거나 하더라도 손쉽게 수정이 가능하다는 장점이 있습니다.

그럼 리디북스 신간 도서 HTML 코드의 구조를 파악해볼까요? 👉

파악한 구조를 바탕으로 다시 선택자를 작성해보자!

먼저, 신간 도서 목록 아이템 ... 를 선택하는 선택자를 아래와 같이 작성해 볼 수 있습니다.

main > section > ul:nth-of-type(2) > li

다음으로 li 태그 안에서 제목과 설명에 대한 선택자는 아래와 같이 작성해 볼 수 있습니다.

div > div:nth-of-type(2) > **div:nth-of-type(1)** > a

div > div:nth-of-type(2) > **div:nth-of-type(2)** > a > p

위처럼 가상 선택자 등을 활용하여 UI 의 배치를 근거로 자료에 접근할 수 있습니다.

위 선택자들은 예시를 위해 class 속성을 사용하지 않고 가상 선택자로 접근했지만, 경우에 따라서 다양한 선택자를 고려하여 최적의 선택자를 구성하는 것이 좋습니다.

BeautifulSoup 에 대해 더 알아보기

BeautifulSoup 이 가지고 있는 메소드, 속성을 더 알아보겠습니다. 😊

BeautifulSoup 의 더 많은 메소드 알아보기

`find_all(name, attrs, recursive, ...)`

`name` : 태그 이름

`attrs` : dict, 속성에 대한 조건 넘겨줌 ex) {"class:['title', 'content'], "data":"true"}

`recursive` : bool, 자식의 자식까지 탐색할 것인지 정해줌

`find(name, attrs, recursive, ...) == find_all(...)[0]`

`select(selector, ...)`

`selector` : 선택자

`namespaces` : dict, 속성에 대한 조건 넘겨줌 ex) {"class:['title', 'content'], "data":"true"}

`flags` : bool, 자식의 자식까지 탐색할 것인지 정해줌

`select_one(selector, ...) == select(...)[0]`

BeautifulSoup 의 더 많은 속성 알아보기

bs4.태그이름 : soup.p → 모든 p 태그 중 첫번째 태그만 가져옴

bs4(태그이름) : soup('p') → 모든 p 태그를 list로 가져옴

bs4.prettify() : HTML 구조를 보기 좋게 문자열로 반환

tag.attrs[속성명], tag[속성명] : a_tag['href'] → href 속성에 있는 값을 가져옴

```
▶<a href="/books/1546001277?rdt_sid=newReleases_general&rdt_idx=3"  
class="b-yvs9ws">...</a> flex == $0
```

tag.text, tag.get_text() : a_tag.text → ... 안에 있는 모든 텍스트 반환

tag.string : a_tag.string → 자신의 직계 텍스트만 반환, 여러개면 None 반환

tag.name : a_tag.name → “a” 태그의 이름을 반환

crummy.com/software/BeautifulSoup/bs4/doc/

리디북스에서 링크 가져오기

리디북스 예제에서 도서의 링크 정보를 가져와 보겠습니다. a 태그를 찾고 href 속성을 가져와 보자!

```
▼<li class="fig-vj0uh1">
  ▼<div class="b-ona580"> flex
    ▼<div class="b-lyemeb"> flex
      ▼<a href="/books/4698000062? rdt sid=newReleases_general& rdt idx=1" class="b-
        </span>
      </a>
    </div>
  </div>
```

근데... /books/46~~ 는 URL, 링크가 아닌거 같은데?

그 이유는 앞에 있는 base url, 즉 사이트 주소가 생략되었으므로 앞에 붙여줘야 한다.

https://ridibooks.com/books/4698000062?_rdt_sid=newReleases_general&_rdt_idx=1

“<https://ridibooks.com>” + book_link or...

import urllib.parse

urllib.parse.urljoin(스크래핑하려는 URL, 경로) → urllib.parse.urljoin('https://example.com/a/b/c', '/d')
→ 'https://example.com/d'

네이버 웹툰 스크래핑 해보기

이제 웬만한 건 다 스크래핑할 수 있을 것 같다! 😜

네이버 웹툰 HTML 가져오기

requests 를 통해서 url 으로 GET 요청을 보내서 text 하면 HTML 받아올 수 있을 것 같다!

네이버 웹툰 HTML 가져오기

requests 를 통해서 url 으로 GET 요청을 보내서 text 하면 HTML 받아올 수 있을 것 같다!

왜 안 가져와 질까?

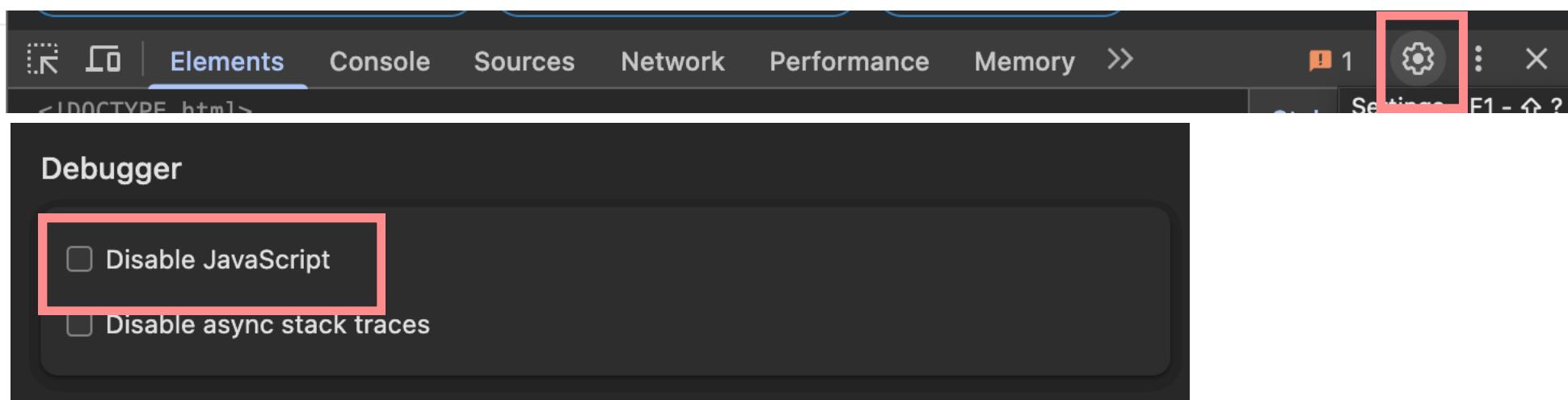
```
~/FLAG-Web-Scraping-Study > on main !2 ?1 ➔ python w2/w2-2-1.py
<body>
<div id="root"></div>
</body>
```

이유는 JavaScript 를 통해 비동기적으로 웹 페이지에 필요한 데이터를 요청해서 보여주기 때문이다.

그럼 못 가져오는 건가...?

개발자 도구로 확인해 보기

설정에서 Javascript 를 비활성화할 수 있다.



Network 웹 페이지가 요청한 네트워크 사용 내역을 모니터링 할 수 있다

The screenshot shows the Network tab in the DevTools. On the left, a list of network requests is displayed, with the 'new?order=update' request highlighted by a red box. On the right, a detailed view of this request is shown in the Headers, Payload, Preview, Response, Initiator, Timing, and Cookies sections. The 'Request URL' field is also highlighted with a red box.

Request URL: https://comic.naver.com/api/webtoon/titlelist/new?order=update

Request Method: GET

Status Code: 200 OK

Remote Address: 175.158.5.162:443

Referrer Policy: unsafe-url

Headers

Content-Type: application/json

Date: Wed, 26 Mar 2025 11:17:27 GMT

Expires: 0

Pragma: no-cache

Referrer-Policy: unsafe-url

Server: nfront

JSON 이란?

JavaScript Object Notation

: Javascript 객체 문법으로 구조화된 데이터를 표현하기 위한 포맷

JSON 구성요소

객체 : { }

배열 : []

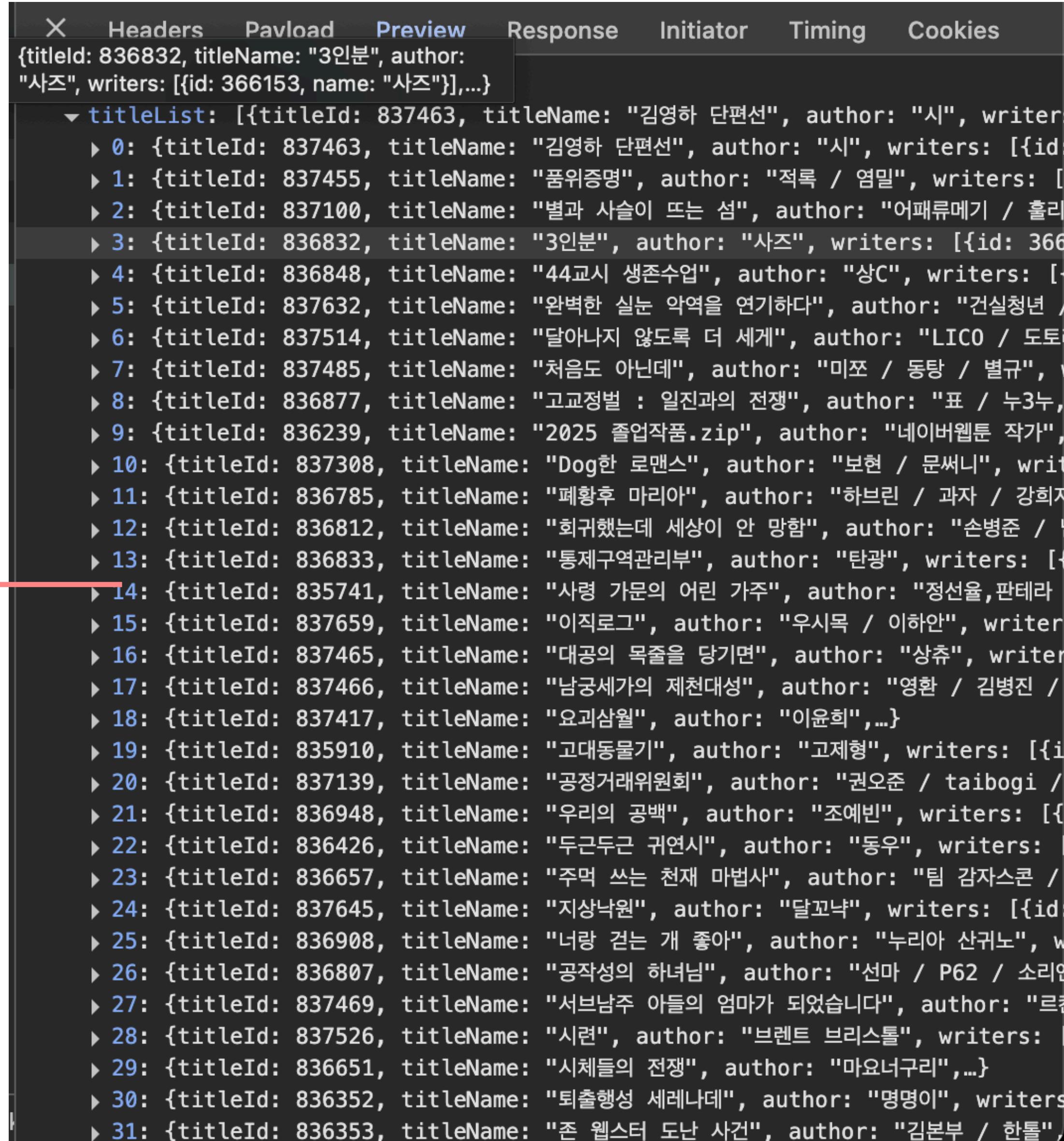
숫자 : 1

문자열 : "제목"

불 : true or false

널 : null

```
{  
    "titleList": [  
        {  
            "titleId": 837463,  
            "titleName": "김영하 단편선",  
            "author": "시",  
            "writers": [  
                {  
                    "id": 387706,  
                    "name": "시"  
                }  
            ],  
            "painters": [  
                {  
                    "id": 387706,  
                    "name": "시"  
                }  
            ],  
            "novelOriginAuthors": [  
                {  
                    "up": true,  
                    "rest": false,  
                    "bm": false,  
                    "adult": false,  
                    "starScore": 9.74687,  
                    "viewCount": 0,  
                    "openToday": true,  
                    "potenUp": false,  
                    "bestChallengeLevelUp": false,  
                    "finish": false,  
                    "new": true  
                }  
            ]  
        }  
    ]  
}
```



The screenshot shows a browser's developer tools Network tab with the Response panel selected. The response is a JSON object representing a list of books. The object has properties: titleId, titleName, author, writers, titleList, and preview. The titleList property contains an array of book objects, each with titleId, titleName, author, and writers. The first book in the list is highlighted with a gray background. The JSON code is displayed in a monospaced font.

```
{  
    "titleId": 836832, "titleName": "3인분", "author": "사즈", "writers": [{"id": 366153, "name": "사즈"}]},  
    "titleList": [  
        {  
            "titleId": 837463, "titleName": "김영하 단편선", "author": "시", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837455, "titleName": "품위증명", "author": "적록 / 엠밀", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837100, "titleName": "별과 사슬이 뜨는 섬", "author": "어페류메기 / 훌리", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836832, "titleName": "3인분", "author": "사즈", "writers": [{"id": 366153, "name": "사즈"}]},  
        {  
            "titleId": 836848, "titleName": "44교시 생존수업", "author": "상C", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837632, "titleName": "완벽한 실눈 악역을 연기하다", "author": "건실청년", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837514, "titleName": "달아나지 않도록 더 세계", "author": "LICO / 도토", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837485, "titleName": "처음도 아닌데", "author": "미쪼 / 동탕 / 별규", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836877, "titleName": "고교정벌 : 일진과의 전쟁", "author": "표 / 누3누", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836239, "titleName": "2025 졸업작품.zip", "author": "네이버웹툰 작가", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837308, "titleName": "Dog한 로맨스", "author": "보현 / 문써니", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836785, "titleName": "폐황후 마리아", "author": "하브린 / 과자 / 강희자", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836812, "titleName": "회귀했는데 세상이 안 망함", "author": "손병준 /", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836833, "titleName": "통제구역관리부", "author": "탄광", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 835741, "titleName": "사령 가문의 어린 가주", "author": "정선율, 판테라", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837659, "titleName": "이직로그", "author": "우시목 / 이하안", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837465, "titleName": "대공의 목줄을 당기면", "author": "상츄", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837466, "titleName": "남궁세가의 제천대성", "author": "영환 / 김병진 /", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837417, "titleName": "요괴삼월", "author": "이윤희", ...},  
        {  
            "titleId": 835910, "titleName": "고대동물기", "author": "고제형", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837139, "titleName": "공정거래위원회", "author": "권오준 / taibogi /", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836948, "titleName": "우리의 공백", "author": "조예빈", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836426, "titleName": "두근두근 귀연시", "author": "동우", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836657, "titleName": "주먹 쓰는 천재 마법사", "author": "팀 감자스콘 /", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837645, "titleName": "지상낙원", "author": "달꼬냑", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836908, "titleName": "너랑 걷는 게 좋아", "author": "누리아 산귀노", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836807, "titleName": "공작성의 하녀님", "author": "선마 / P62 / 소리안", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837469, "titleName": "서브남주 아들의 엄마가 되었습니다", "author": "르침", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 837526, "titleName": "시련", "author": "브렌트 브리스톨", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836651, "titleName": "시체들의 전쟁", "author": "마요너구리", ...},  
        {  
            "titleId": 836352, "titleName": "퇴출행성 세레나데", "author": "명명이", "writers": [{"id": 387706, "name": "시"}]},  
        {  
            "titleId": 836353, "titleName": "존 웨스터 도난 사건", "author": "김본부 / 한톨", "writers": [{"id": 387706, "name": "시"}]}]
```

JSON 이란?

```
import json  
data = json.loads(webtoonsResult) # data type → dict
```

data[“titleList”] == 🤔?

data[“titleList”][0][“titleName”] == 🤔?

```
for item in data[“titleList”]:  
    print(item) 🤔🤔🤔
```

JSON 구성요소

객체 : { }

배열 : []

숫자 : 1

문자열 : “제목”

불 : true or false

널 : null

```
{ ⌂  
    "titleList": [ ⌂  
        { ⌂  
            "titleId": 837463,  
            "titleName": "김영하 단편선",  
            "author": "시",  
            "writers": [ ⌂  
                { ⌂  
                    "id": 387706,  
                    "name": "시"   
                }  
            ],  
            "painters": [ ⌂  
                { ⌂  
                    "id": 387706,  
                    "name": "시"   
                }  
            ],  
            "novelOriginAuthors": [ ⌂  
                ],  
                "thumbnailUrl": "https://image-comic.pstatic.net/v  
                "up": true,  
                "rest": false,  
                "bm": false,  
                "adult": false,  
                "starScore": 9.74687,  
                "viewCount": 0,  
                "openToday": true,  
                "potenUp": false,  
                "bestChallengeLevelUp": false,  
                "finish": false,  
                "new": true  
            }  
        ]  
    }  
}
```

네이버 웹툰 제목 출력해보기

아래 Pseudo Code 를 참고하여 코드를 작성해보자! 

0. Network 모니터링으로 url 을 알아낸다.
1. requests와 json 라이브러리를 불러온다
2. 네이버 웹툰 API URL을 변수에 저장한다
3. requests.get() 함수로 API를 호출하여 결과를 텍스트로 받아온다
4. 받아온 텍스트를 json.loads()로 파싱하여 딕셔너리 형태로 변환한다
5. 파싱된 데이터(data)에서 'titleList' 키의 값을 가져온다
 - titleList는 웹툰 정보가 담긴 딕셔너리들의 리스트
 - 각 딕셔너리는 웹툰의 제목, 작가, 설명 등의 정보를 포함
6. titleList를 for 반복문으로 순회하면서:
 - 각 웹툰 정보(item)에서 'titleName' 키의 값을 가져온다
 - 가져온 titleName을 print()로 출력한다
 - 이 과정을 titleList의 모든 요소에 대해 반복한다

다음주 ...

1. 로그인 과정이 필요한 웹사이트 스크래핑하기
2. 파일(이미지) URL 을 통해 다운로드 하는 방법
3. **Selenium** 으로 Interactive 한 사이트 스크래핑하기

도전과제

앞선 “네이버 웹툰 예제”에서 제목만 가져왔습니다.

 소스를 수정해서 제목 뿐만 아니라 **작가이름**을 가져와 보세요! ex) 제목 / 작가명
hint ) 작가를 의미하는 writers 는 json 객체입니다!

만약, 위 과제를 성공하셨나요? 그럼 하나 더!!

 정렬이 “업데이트” 순으로 되어 있습니다. 인기순, 업데이트, 조회순, 별점순에 대한 데이터도 요청해 보세요!
hint ) [Network] 탭에서 JSON 요청 URL 을 잘 살펴보세요!