

# **Web Scraping**

웹에서 정보를 자동으로 수집하기!

# **Scraping vs Crawling**

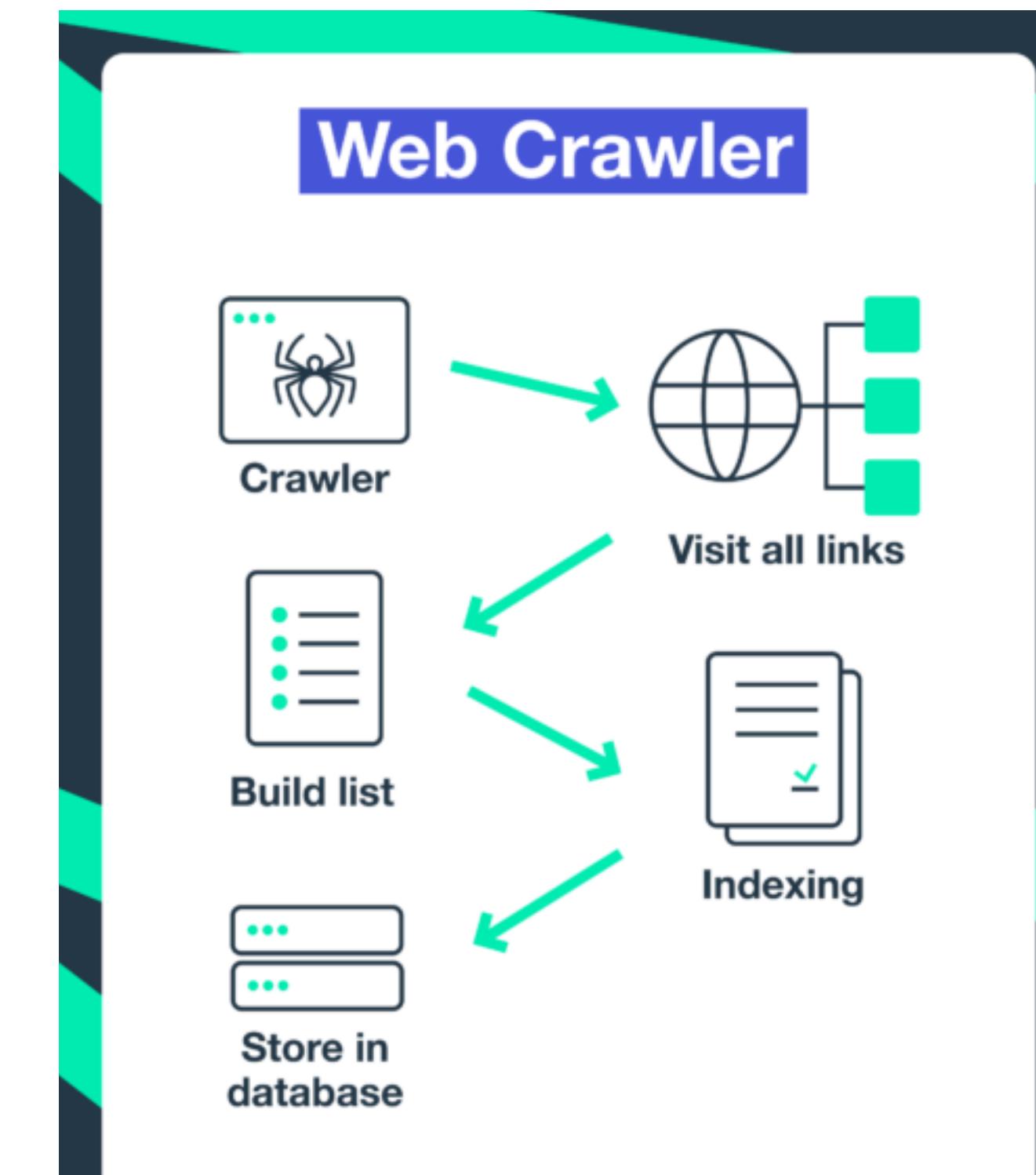
# Scraping

필요한 데이터를 추출하는 것  
텍스트, 이미지 등 ...  
원하는 정보를 자동으로 추출하기 위해 진행



# Crawling

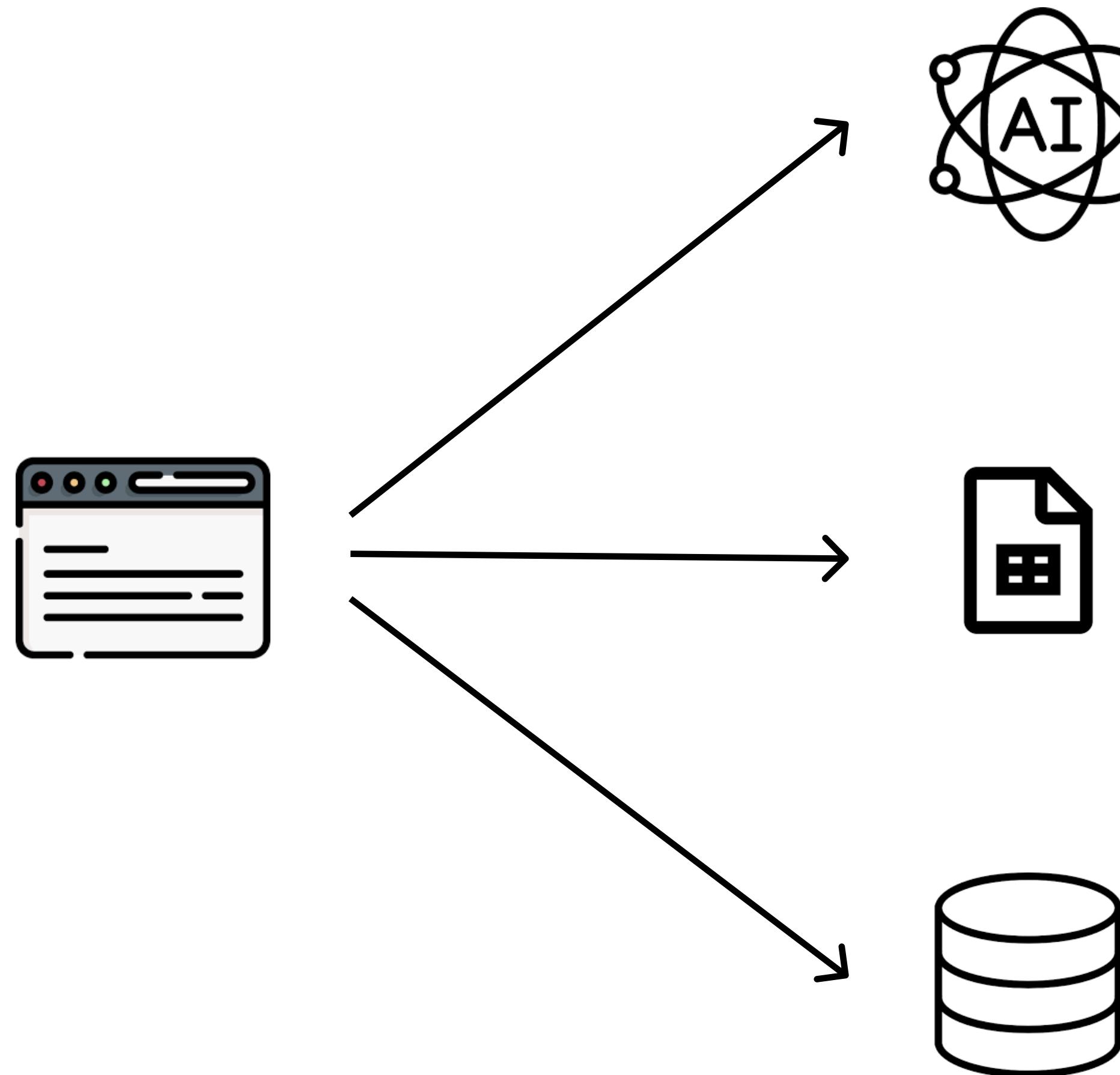
여러 웹사이트의 페이지를 탐색하는 것  
사이트 전체, 사이트의 특정 부분  
보통 검색엔진의 인덱싱 작업을 위해 진행



# **Why : Scraping 은 왜 필요할까?**

Scraping 을 어디에 활용할 수 있을까요?

# Scraping 은 ...



## 빅데이터 수집

데이터가 **돈**이 되는 요즘 많은 양의 데이터를 무료로 수집하여 부족한 데이터셋을 보충하기 위해 Scraping 을 활용할 수 있습니다.

## 자동화 및 생산성 향상

자신의 업무 중에 웹에서 주기적으로 정보를 가져오는 **반복적인 작업**이 있다면 이를 소프트웨어로 대체할 수 있습니다.

## 기존에 없던 데이터베이스 구축 가능

API 로 제공되지/공개되지 않은 데이터를 수집하여 **데이터베이스**를 구축하고 RESTful API 등으로 사용이 가능합니다.

# **How : Scraping 을 어떻게 하지?**

Python 을 이용한 Scraping 방법을 배워봅시다!

# Python 설치 확인

Windows

> CMD 실행

> python --version →

> pip --version

```
~/FLAG-Web-Scraping-Study ➔ python --version
Python 3.10.16

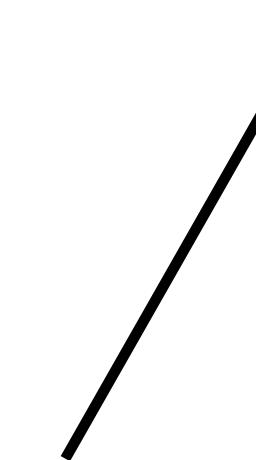
~/FLAG-Web-Scraping-Study ➔ pip --version
pip 24.3.1 from /opt/homebrew/lib/python3.10/site-packages/pip (python 3.10)
```

Mac or Linux

> Terminal 실행

> python --version

> pip --version



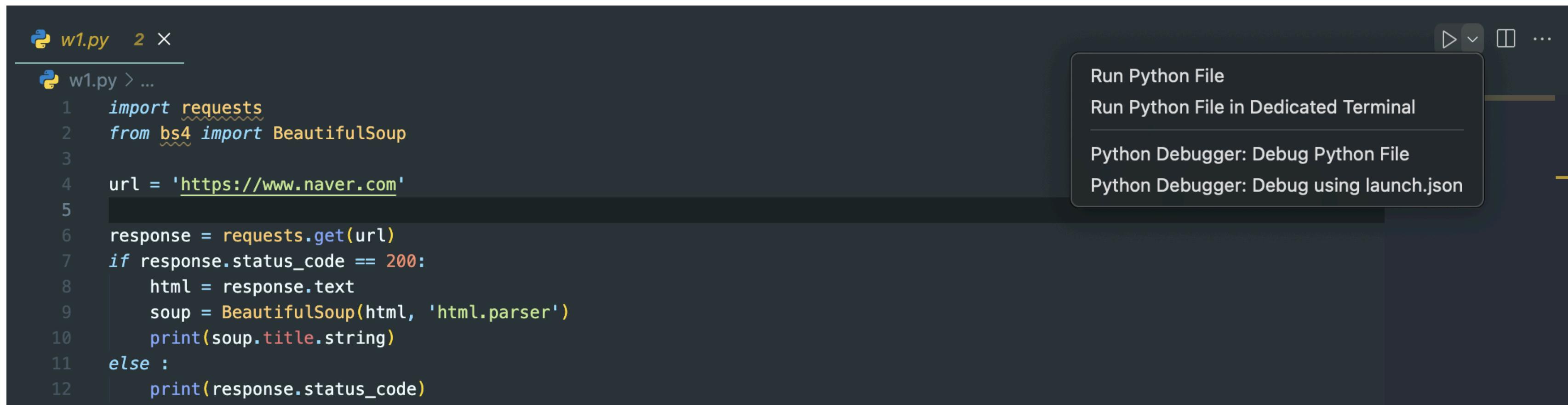
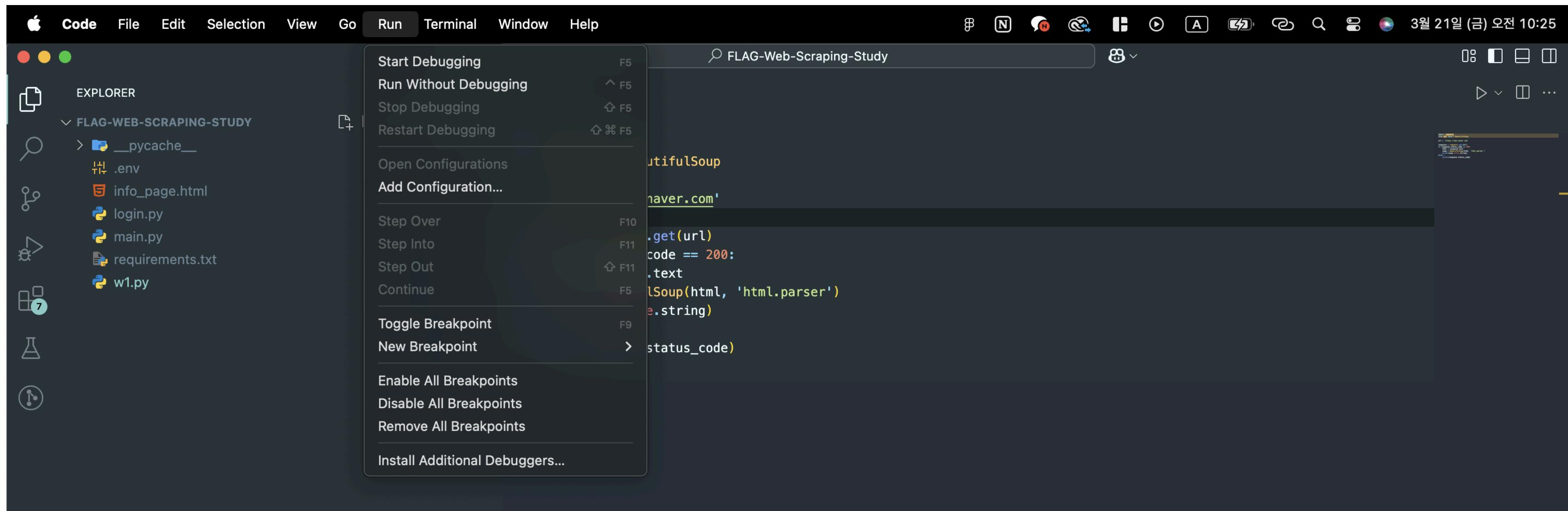
```
w1.py
import requests
from bs4 import BeautifulSoup
url = 'https://www.naver.com'
response = requests.get(url)
if response.status_code == 200:
    html = response.text
    soup = BeautifulSoup(html, 'html.parser')
    print(soup.title.string)
else :
    print(response.status_code)
```

에디터는 자유롭게 사용하시면 되고, VS Code 권장합니다.

단, Colab, Jupyter Notebook은 Selenium 실행을 할 수 없다는 점 참고바랍니다.



# VS Code에서 Python 실행하기



# Scraping 을 위한 도구

## BeautifulSoup

소규모 또는 중간 규모의 스크래핑 작업에 적합합니다.  
사용이 간편하며, 속도가 큰 문제가 되지 않는 프로젝트에 이상적입니다.



## Scrapy

대규모이고 복잡한 웹 스크래핑 작업에 가장 적합하며,  
여러 요청을 처리하고 데이터를 체계적으로 처리해야 할 때 유용합니다.  
웹 크롤링 프레임워크에 가깝기 때문에 기능이 더 많고 학습 곡선이 더 가파릅니다.



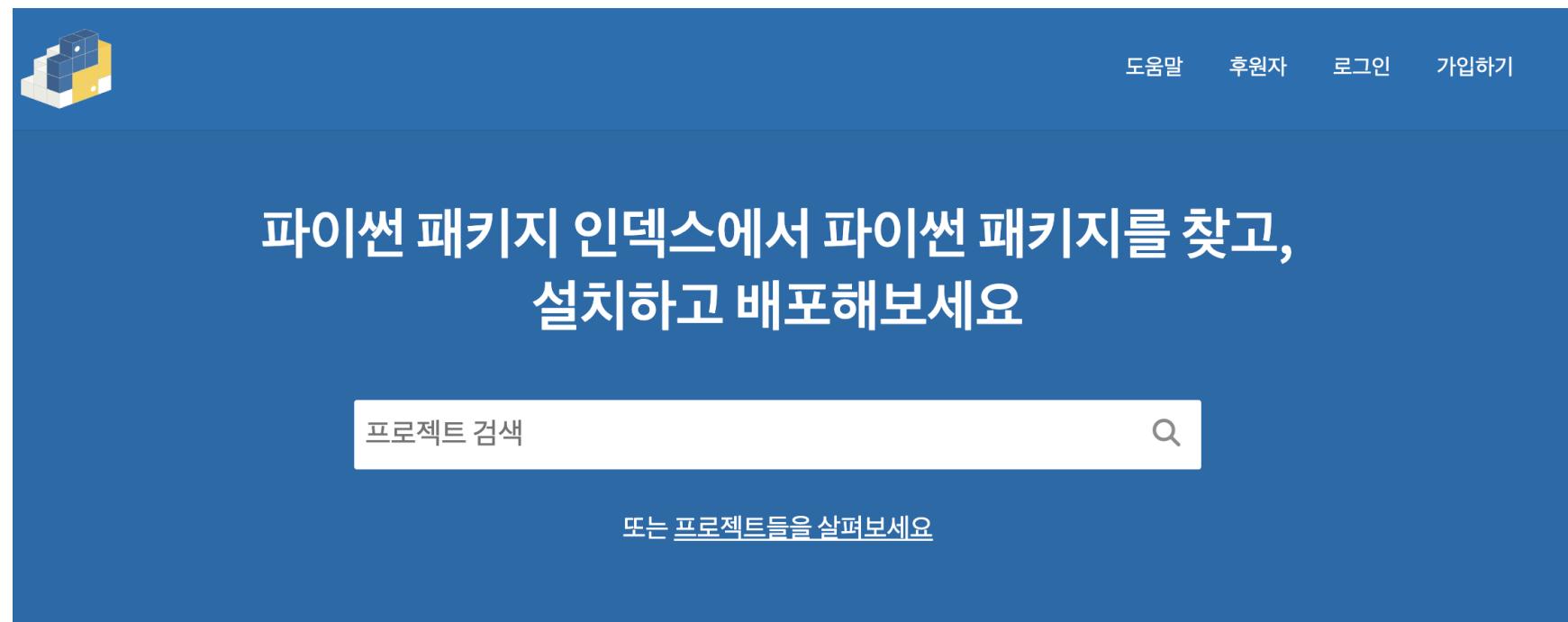
## Selenium

실제 웹 브라우저를 띠워서 진행하기 때문에 프로그램으로서의 제약사항이 거의 없  
다. 대신에 안정성이나 속도 측면에서 단점이 있다.



# BeautifulSoup 설치하기 및 간단한 예제 실행

> pip install **beautifulsoup4** # pip로 pypi.org에 있는 패키지를 다운로드 할 수 있음



Github > 'FLAG-Web-Scraping-Study' 검색 > Woochang4862/FLAG-Web-Scraping-Study 레포지토리

A screenshot of a GitHub repository page. The top card displays the repository information: owner 'Woochang4862', name 'FLAG-Web-Scraping-Study', language 'Python', stars '0', and last update 'Updated 2 minutes ago'. There is also a 'Star' button. Below this, the commit history is shown with one entry: 'initial commit' by 'Woochang4862' at '6638ff4 · 3 minutes ago'. The commit details show five files: 'w1', 'w4', '.DS\_Store', '.gitignore', and 'LICENSE', all with an 'initial commit' status and timestamped '3 minutes ago'.

File	Commit Type	Timestamp
w1	initial commit	3 minutes ago
w4	initial commit	3 minutes ago
.DS_Store	initial commit	3 minutes ago
.gitignore	Initial commit	5 minutes ago
LICENSE	Initial commit	5 minutes ago

# 어떻게 웹 페이지에 접속할 수 있을까?

Python 이 웹 페이지에 접속한 방법

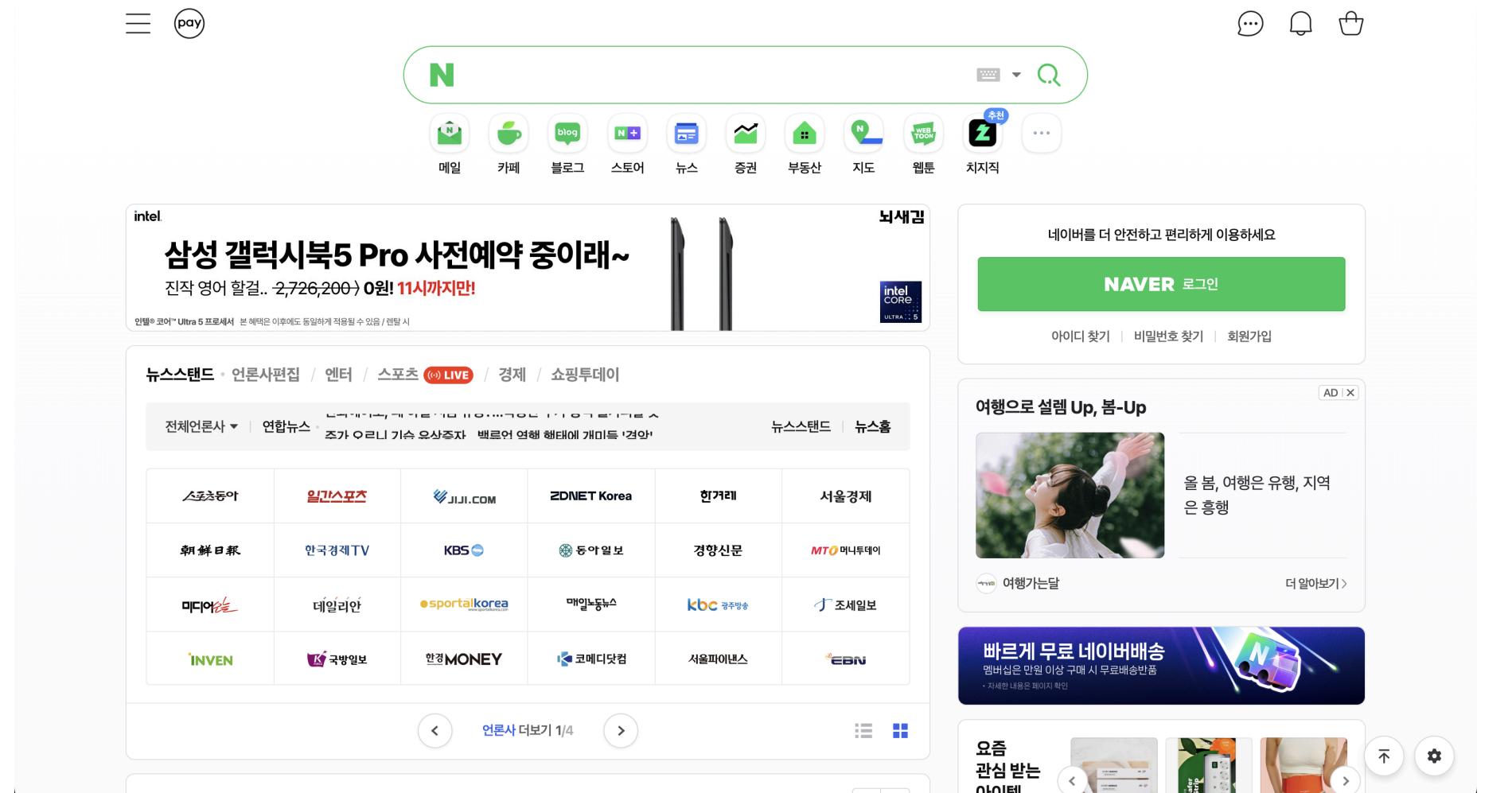
# 웹 페이지의 실체?!

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    ...
  </head>
  <body>
    ...
  </body>
</html>
```

웹 브라우저 개발자도구  
또는  
우클릭 > 페이지 소스 보기

← →

렌더링  
+CSS (디자인)  
+JavaScript (동적인 액션)



HTML (HyperText Mark-up Language)  
웹 페이지의 내용을 담고 있는 문서

보기 좋게 꾸며진 웹 사이트

# HTTP 요청과 requests 모듈



목적      **GET(조회)**  
          **POST(데이터 처리/생성)**  
          DELETE(삭제)  
          PUT(수정)  
          ...

**POST** <https://nid.naver.com/nidlogin.login>

Response  
Success or Fail!

# requests 를 이용해서 실제 “웹”에 접속하기

> pip install **requests**

```
import requests  
from bs4 import BeautifulSoup
```

```
url = 'https://www.naver.com'
```

```
response = requests.get(url)  
if response.status_code == 200:  
    html = response.text  
    soup = BeautifulSoup(html, 'lxml')  
    print(soup.title.string) # NAVER  
else :  
    print(response.status_code)
```



직접 손으로 적어 봅시다!



# requests 모듈의 역할 : 웹 페이지 접속

```
<!DOCTYPE html>
<html lang="ko" class="fzoom" data-dark="false">
... ▼<head> == $0
  <script async src="https://ntm.pstatic.net/ex...
  <script async type="text/javascript" src="htt...
dk/prod/ndp-core.js"></script>
  <meta charset="utf-8">
  <meta name="Referrer" content="origin">
  <meta http-equiv="X-UA-Compatible" content="I...
  <meta name="viewport" content="width=1190">
  <title>NAVER</title>
  <meta name="apple-mobile-web-app-title" conte...
  <meta name="robots" content="index,nofollow">
  <meta name="description" content="네이버 메인에서...
  <meta property="og:title" content="네이버">
  <meta property="og:url" content="https://www...
  <meta property="og:image" content="https://s...
016/0705/mobile_212852414260.png">
  <meta property="og:description" content="네이...
  세요">
  <meta name="twitter:card" content="summary">
  <meta name="twitter:title" content>
  <meta name="twitter:url" content="https://ww...
  <meta name="twitter:image" content="https://s...
```



url = 'https://www.naver.com'

response = requests.get(url)

response.text #HTML 소스

# BeautifulSoup 모듈의 역할 : 웹 페이지 접속

```
<!DOCTYPE html>
<html lang="ko" class="fzoom" data-dark="false">
... ▶ <head> == $0
    <script async src="https://ntm.pstatic.net/ex...
    <script async type="text/javascript" src="htt...
    <meta charset="utf-8">
    <meta name="Referrer" content="origin">
    <meta http-equiv="X-UA-Compatible" content="I...
    <meta name="viewport" content="width=1190">
    <title>NAVER</title>
    <meta name="apple-mobile-web-app-title" conte...
    <meta name="robots" content="index,nofollow">
    <meta name="description" content="네이버 메인에서...
    <meta property="og:title" content="네이버">
    <meta property="og:url" content="https://www...
    <meta property="og:image" content="https://s...
    <meta property="og:description" content="네이...
    <meta name="twitter:card" content="summary">
    <meta name="twitter:title" content>
    <meta name="twitter:url" content="https://ww...
    <meta name="twitter:image" content="https://s...
```

```
soup = BeautifulSoup(html, 'lxml')
print(soup.title.string) # NAVER
```

## Parser 란?

일정한 규칙이나 위계질서로 정의된 문서로부터 구조를 파악하는 것을 Parsing 이라고 하는데 이런 parsing 을 수행하는 역할을 하는 것을 말한다.

## HTML Parser 란?

HTML 문서로부터 구조를 파악하는 역할을 한다.  
종류 : lxml, html.parser, html5lib

## lxml 이 가장 빠르고 단순

# 웹 페이지에서 원하는 내용을 찾아보자!

BeautifulSoup 을 이용해서 Element 를 찾는 방법

# 리디북스 신간 도서 스크래핑

RIDI



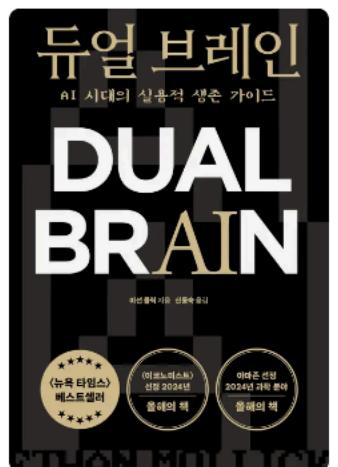
회원가입 · 로그인

웹툰 · 만화 · 웹소설 · 도서 · 셀렉트

☰ 전체 카테고리 · Ⓜ 캐시충전

## 일반도서 신간

인기순 · 최신순



### 듀얼 브레인

이선 몰릭 외 1명 · 상상스퀘어 · 성공/삶의자세

★ 5.0 (2)

★ <뉴욕 타임스> 베스트셀러 ★ <이코노미스트> 선정 2024년 올해의 책 ★ 아마존 선정 2024년 과학 분야 올해의 책 『듀얼 브레인』은 AI 시대를 살아가기 위해 꼭 읽어야 할 필독서다. 저자 이선 몰릭은 <타임>에서 선정한 '인공지능 분야에서 가장 영향력 있는 인물' 중 한 명이다. 여러 AI 기업에 자문을 제공하고, 와튼 스쿨에서 교육에 AI 활용을 접목하는 등 다양한...

소장 14,700원



### 수진의 함바식당

종려나무숲 · 제우미디어 · 한국소설

★ 4.7 (18)

"점심시간이네요. 자, 다들 밥 먹고 합시다." 거칠고 힘든 공사 현장의 유일한 낙, 함바식당에서 먹는 따뜻한 한 끼! 부경건설 수원지구 아파트 공사 현장에는 조금 특별한 함바식당이 있다. 땀내 나는 거친 공사 현장에는 조금 이질적인 30대의 여자 사장 '수진'! 언뜻 보기에 대학을 갓 졸업한 풋내기 같아 보이지만, 이전에 인부들의 취향에 맞지 않는 메뉴와 맛으로 적자를...

대여 5,400원

소장 9,720원 (10%) 10,800원



### 배틀그라운드, 새로운 전장으로

이기문 · 김영사 · 경영일반

★ 0 (0)

# 웹 사이트 스크래핑 과정

RIDI

회원가입 · 로그인

#1

웹툰 만화 웹소설 도서 셀렉트

전체 카테고리 캐시충전

## 일반도서 신간

인기순 · 최신순

#2

듀얼 브레인

이선 몰릭 외 1명 · 상상스퀘어 · 성공/삶의자세

★ 5.0 (2)

★ <뉴욕 타임스> 베스트셀러 ★ <이코노미스트> 선정 2024년 올해의 책 ★ 아마존 선정 2024년 과학 분야 올해의 책  
『듀얼 브레인』은 AI 시대를 살아가기 위해 꼭 읽어야 할 필독서다. 저자 이선 몰릭은 <타임>에서 선정한 '인공지능 분야에...'

소장 14,700원

#3

수진의 함바식당

종려나무숲 · 제우미디어 · 한국소설

★ 4.7 (18)

"점심시간이네요. 자, 다들 밥 먹고 합시다." 거칠고 힘든 공사 현장의 유일한 낙, 함바식당에서 먹는 따뜻한 한 끼! 부경건설 수원지구 아파트 공사 현장에는 조금 특별한 함바식당이 있다. 땀내 나는 거친 공사 현장에는 조금 이질적인 30대의 여자...

대여 5,400원

소장 9,720원 (10%) 10,800원

10% 대여

KRAFTON WAY

배틀그라운드, 새로운 전장으로

이기문 · 김영사 · 경영일반

★ 0 (0)

배틀그라운드 출시 이후 게임 소 저작보다 더 치열하 혁신 저작에서 부트하던 그녀가 드녀이 기로 아전되 조지으 엉고

DevTools is now available in Korean!

Always match Chrome's language Switch DevTools to Korean Don't show again

Elements Console Sources Network

<!DOCTYPE html>

<html lang="ko"> scroll

><head prefix="og: http://ogp.me/ns# fb: http://ogp.me/ns/fb# books: http://ogp.me/ns/books #">@@</head>

... <body data-new-gr-c-s-check-loaded="14.1226.0" data-gr-ext-installed cz-shortcut-listen="true"> == \$0

> <div id="\_\_next">@@</div>

> <script id="\_\_NEXT\_DATA\_\_" type="application/json">@@</script>

> <div>@@</div>

> <script type="text/javascript" id="charset">@@</script>

> <noscript>@@</noscript>

> <script id="G-YB9VX70336" type="text/javascript" src="https://www.googletagmanager.com/gtag/j...js?id=G-YB9VX70336"></script>

> <script type="text/javascript" id="charset">@@</script>

> <iframe height="0" width="0" style="display: none; visibility: hidden;">@@</iframe>

> <next-route-announcer>@@</next-route-announcer>







html body

Styles Computed Layout Event Listeners DOM Breakpoints Properties Accessibility

Filter

element.style { }

body { margin: 0; padding: 0; overflow-x: hidden; }

body { margin: 0px; padding: 0px; overflow-x: hidden; }

body { margin: 0; }

\*:::after, \*:::before { box-sizing: border-box; }

# 웹 사이트 스크래핑 과정

```
import requests as rq
from bs4 import BeautifulSoup

url = 'https://ridibooks.com/new-releases/general?order=POPULARITY&page=1'

html = rq.get(url).content
soup = BeautifulSoup(html, 'lxml')
items = soup.select('main > section > ul.fig-1pep8jc.eis6k7i0 > li')
```

위 소스는 url에 HTML 코드를 요청하는 코드이므로 웹 페이지를 불러오려면 위 코드를 쓸 수 밖에 없다.

앞에서 알아낸 css 선택자를 bs4.select( css 선택자 )에 넣어서 정보를 담고 있는 태그를 찾아낸다.

이게 Scraping, BeautifulSoup의 핵심이다.

# 다음주 ...

1. **css 선택자**라는 것에 대해서 알아보겠습니다.

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

2. 좀 더 복잡한 Element 선택을 해보겠습니다!

3. **네이버 신작 웹툰 목록** 스크래핑 해보기

## 도전과제

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!

만약, 위 과제를 성공하셨나요? 그럼 하나 더!!

 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!