크롤링의 이해 및 활용

송기태 (kitae040522@gmail.com)
Soongsil Univ. (Computer Science and Engineering)

Content

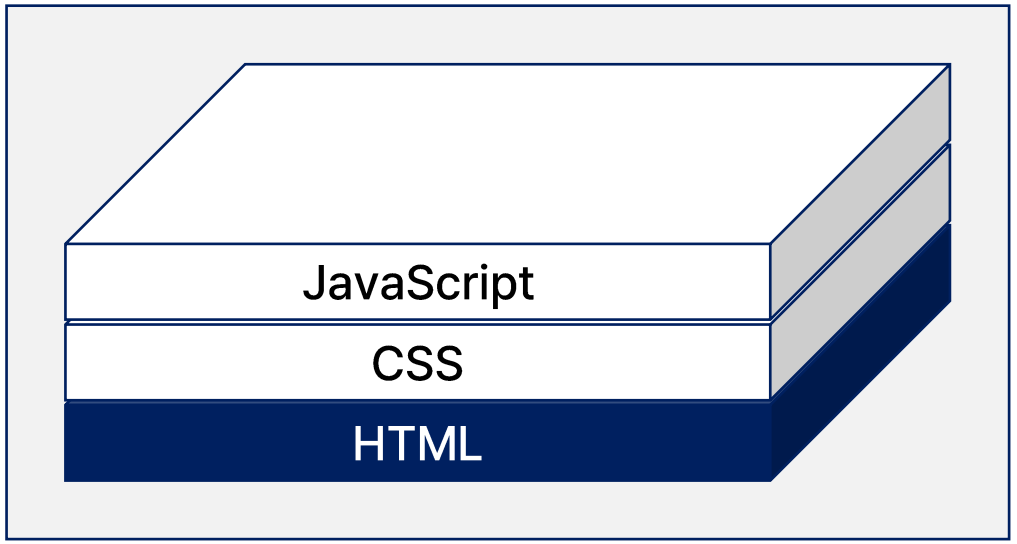
크롤링(crawling) 이란?

- 웹상에 존재하는 컨텐츠를 수집하는 작업 (프로그래밍으로 자동화 가능)
  1. HTML 페이지를 가져와서, HTML/CSS등을 파싱하고, 필요한 데이터만 추출하는 기법
  2. Open API를 제공하는 서비스에 Open API를 호출해서, 받은 데이터 중 필요한 데이터만 추출하는 기법
  3. Selenium등 브라우저를 프로그래밍으로 조작해서, 필요한 데이터만 추출하는 기법

쉽게 이해하자면 웹페이지상에서 데이터를 긁어와서 가져오는 것이다.

크롤링의 원리

웹 사이트의 구조

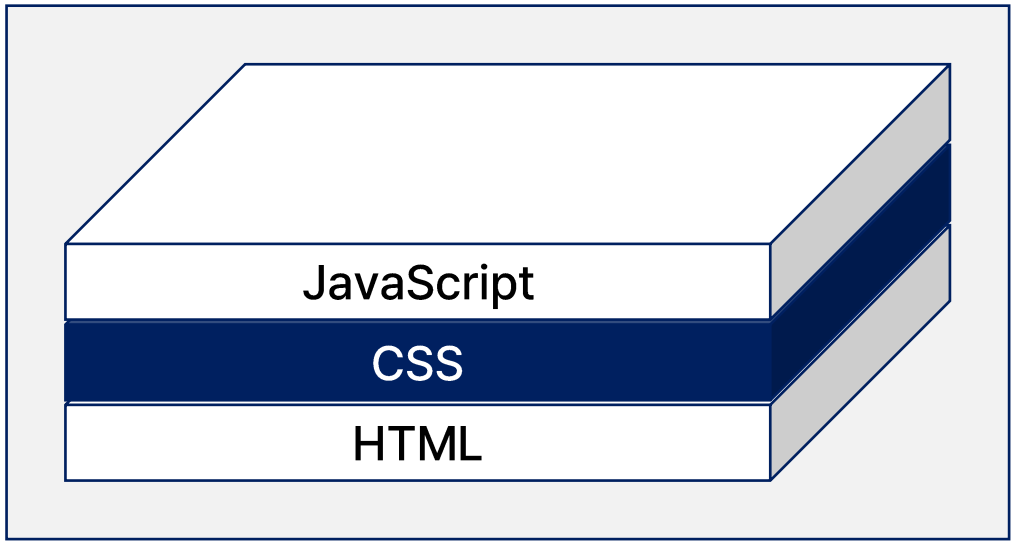


HTML

웹 페이지의 구조를 정의하는 언어. 웹의 내용과 구조를 만든다. 뼈대라고 생각하면 된다.

크롤링의 원리

웹 사이트의 구조

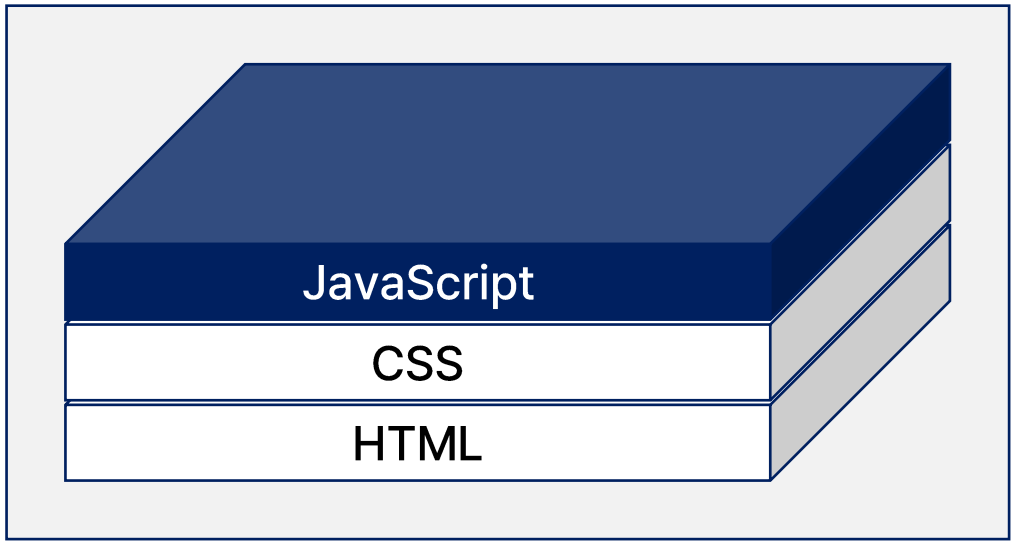


CSS

웹 페이지의 디자인과 스타일을 정의하는 언어. 뼈대 구조에 살을 입히는 거라고 생각하면 된다.

크롤링의 원리

웹 사이트의 구조

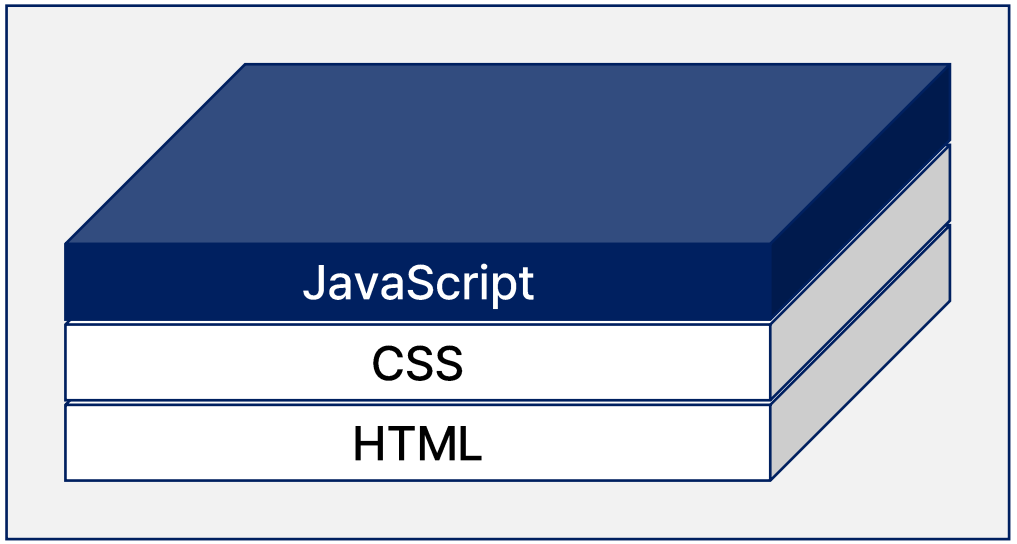


JavaScript

웹 페이지를 동적으로 만들고 상호작용을 추가하는 언어. 동적 기능을 부여하는 데 사용된다.

크롤링의 원리

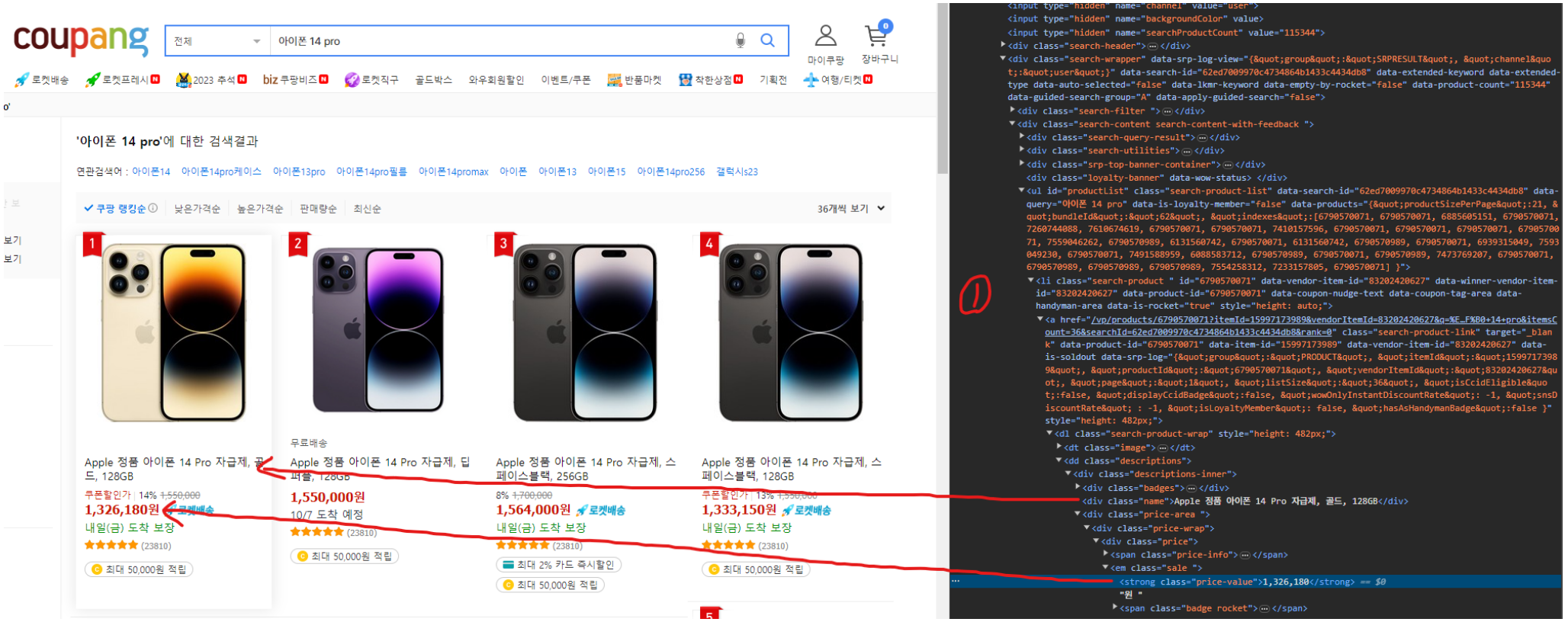
웹 사이트의 구조



JavaScript

웹 페이지를 동적으로 만들고 상호작용을 추가하는 언어. 동적 기능을 부여하는 데 사용된다.

크롤링의 원리



HTML로 이루어진 구조 데이터를 갖고 와서, 필요한 값들을 추출하는 과정

크롤링의 원리

웹 서버에
정보 요청하기

서버 응답을 받은 후
데이터 핸들링

크롤링 시작하기

1. 웹 서버에 정보 요청하기

```
import requests as req

if __name__ == '__main__':
    def main():
        url = 'http://printwiki.org/Dummy'
        html = req.get(url)
        print(html.content) # 출력 결과: 웹사이트의 HTML 코드

    main()
```

2. 서버 응답을 받은 후 데이터 핸들링

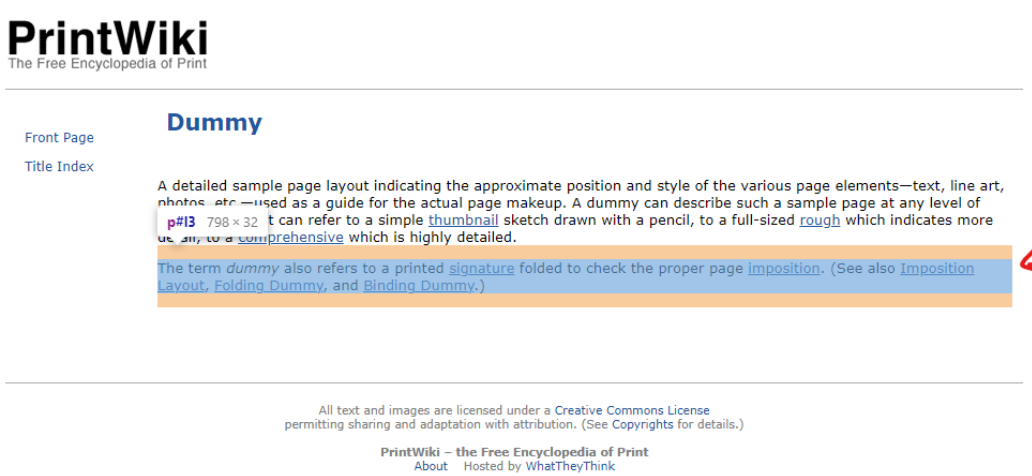
```
# 사전에 pip install을 통해 모듈을 설치해야함!

from bs4 import BeautifulSoup
import requests as req

if __name__ == '__main__':
    def main():
        url = 'http://printwiki.org/Dummy'
        html = req.get(url)
        soup = BeautifulSoup(html.content, 'html.parser')
        print(soup)


    main()
```

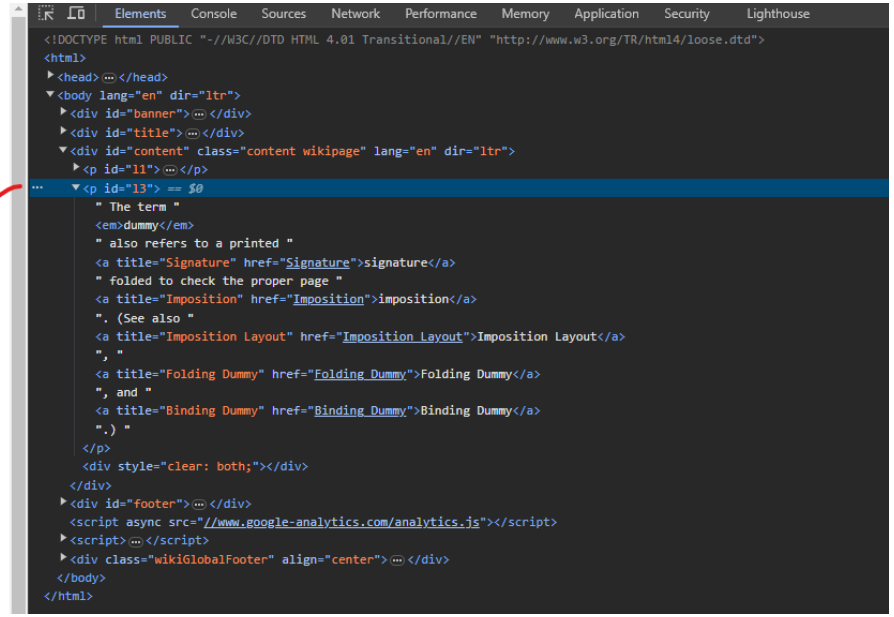
2. 서버 응답을 받은 후 데이터 핸들링 (BeautifulSoup의 사용 이유)

- Requests를 통해서 HTML를 받아올 수는 있지만, Python이 이해하는 객체 구조로 직렬화시켜주지는 못한다.
- 받은 HTML를 의미있는 형태로 만들어주기 위해서 사용한다.
- 해당 라이브러리는 객체 구조를 변환시켜주는 Parsing 역할을 맡고있다.

크롤링 시작하기

2. 서버 응답을 받은 후 데이터 핸들링

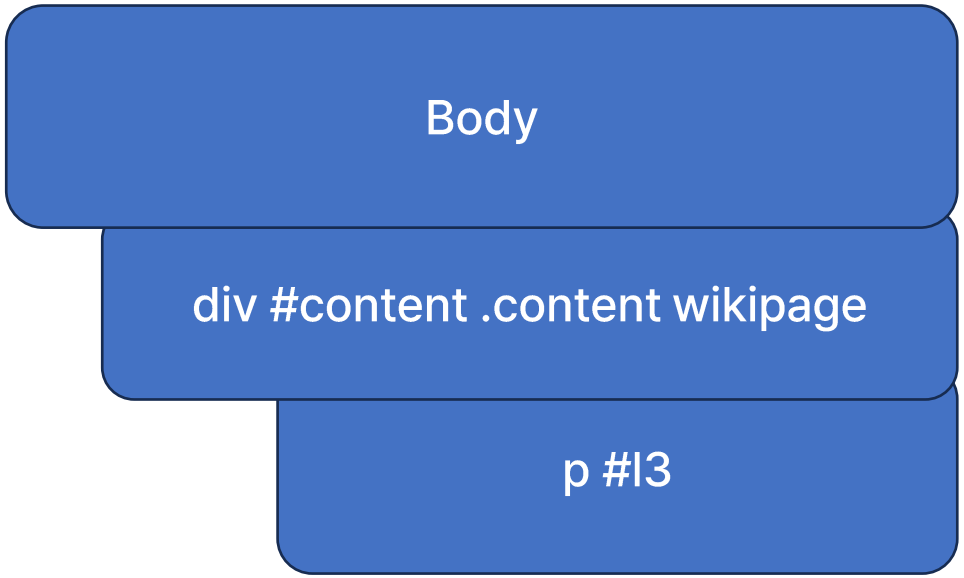




2. 서버 응답을 받은 후 데이터 핸들링

<계층 구조>

- Body
  - div (id: content, class: content wikipage)
    - p (id: l3)



2. 서버 응답을 받은 후 데이터 핸들링

```
# 사전에 pip install을 통해 모듈을 설치해야함!

from bs4 import BeautifulSoup
import requests as req

if __name__ == '__main__':
    def main():
        url = 'http://printwiki.org/Dummy'
        html = req.get(url)
        soup = BeautifulSoup(html.content, 'html.parser')
        content_box = soup.find('div', {'id': 'content', 'class': 'content wikipage'})
        extract_text = content_box.find('p', {'id': 'l3'}).text
        print(extract_text)


    main()
```

Thank You!

송기태 (kitae040522@gmail.com)
Soongsil Univ. (Computer Science and Engineering)