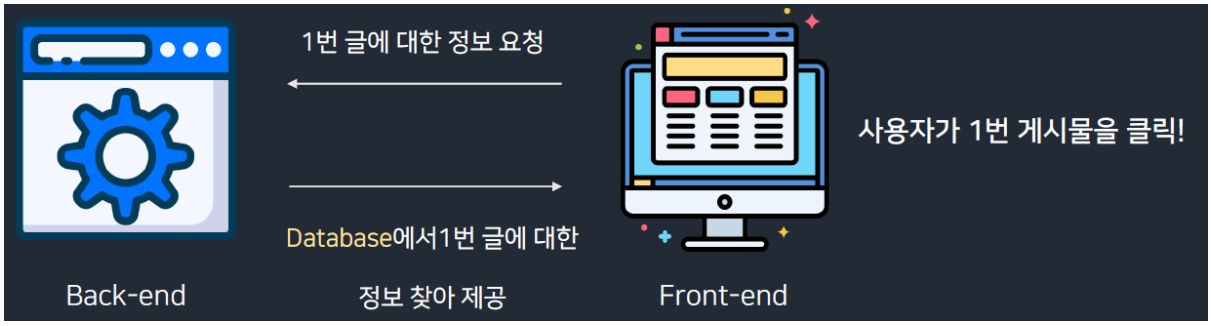
Crawling

Open API

API(Application Programming Interface)

: 애플리케이션이 요청과 응답을 주고받는 체계



Front-end가 정보요청 시 특정 규칙에 맞아야 함 = API (사용 규칙을 제공)

Crawling

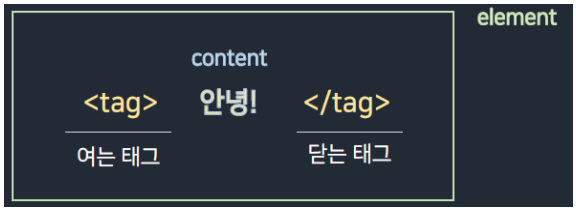
: 검색 엔진 로봇을 이용한 데이터 수집 방법

1) 파이썬의 크롤러로 웹 서버에 정보 요청

2) 서버 응답을 받은 후 웹 서버와 상호작용하며 정보 획득

3) 얻은 정보를 핸들링하여 데이터화

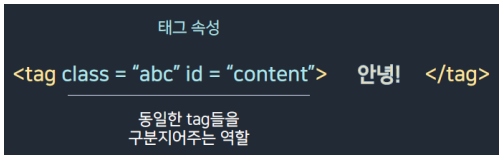
Html : 웹 브라우저에서 문서 및 웹 페이지가 표시되는 방법을 규정하는 언어



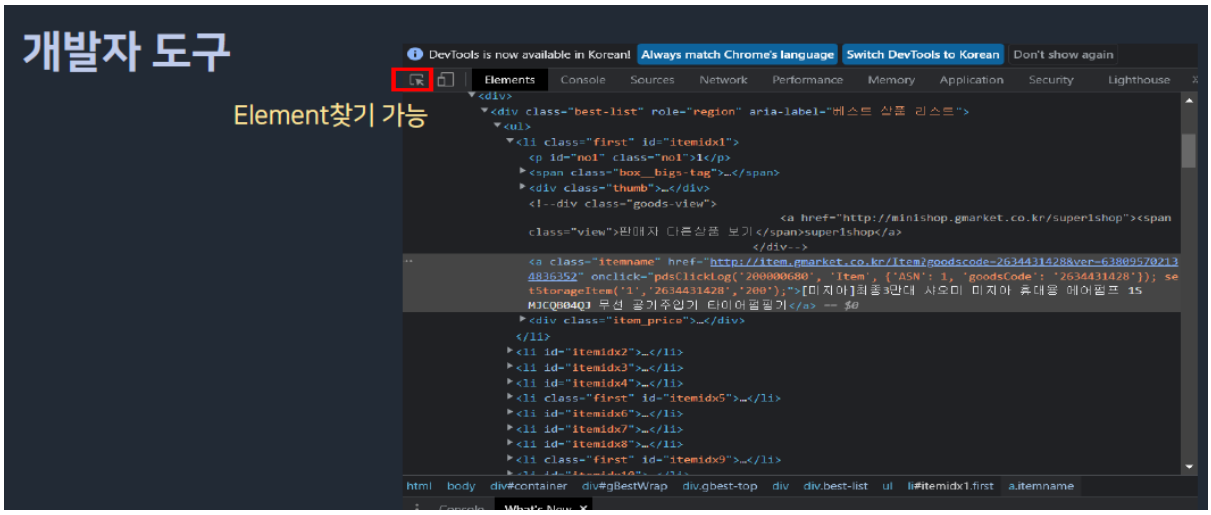
- 한 element는 여는 태그+내용+닫는 태그로 이루어져 있다
- Tag끼리 상하 관계(부모/자식/형제)가 존재한다.

CSS : Html로 만들어진 문서의 스타일을 지정하는 방식을 규정하는 스타일 시트 언어

- 바로 하위의 태그 : tag1 > tag2
- 하위의 태그(띄어쓰기) : tag1 tag2



- 클래스 이용하여 찾기(.) : tag1.abc
- ID 이용하여 찾기(#) : tag1#content



1. 정보 추출하고자 하는 element에 마우스 대고 우클릭 - [검사]
2. 화면 우측 상단에 Chrome 맞춤 설정 - [도구 더보기] - [개발자도구]
3. 키보드 F12

Beautiful Soup

Request 패키지로 텍스트 형태의 html 문서 가져온 후, 이 텍스트 형태에서 원하는 html 태그를 추출할 수 있게 해줌

```
import requests
from bs4 import BeautifulSoup

res = requests.get(웹페이지 주소)
Soup = BeautifulSoup(res.content, 'html.parser')
items = soup.find(태그 경로)
```

Find vs Select

Select : CSS Selector로 태그 객체를 찾아 반환

단일추출 : find = select_one : 가장 처음 찾은 태그 객체 반환

다중추출 : find_all = select : 해당하는 모든 태그들을 찾아 list 형태로 반환

```
items = soup.select('div.abc')
  for item in items:
    print(item.get_text)
```

Selenium

: 동적 웹 스크래핑 방법

- Javascript가 동적으로 만든 데이터를 크롤링할 수 있게 해줌
- 사용자가 실제로 브라우저를 탐색하는 것처럼 작동
- HTML 요소 클릭, 페이지 이동, 키보드 입력 자율