DALC Scalar Crawling

2021.03.27

DALC 공통 기초 스터디 변경 계획 안내

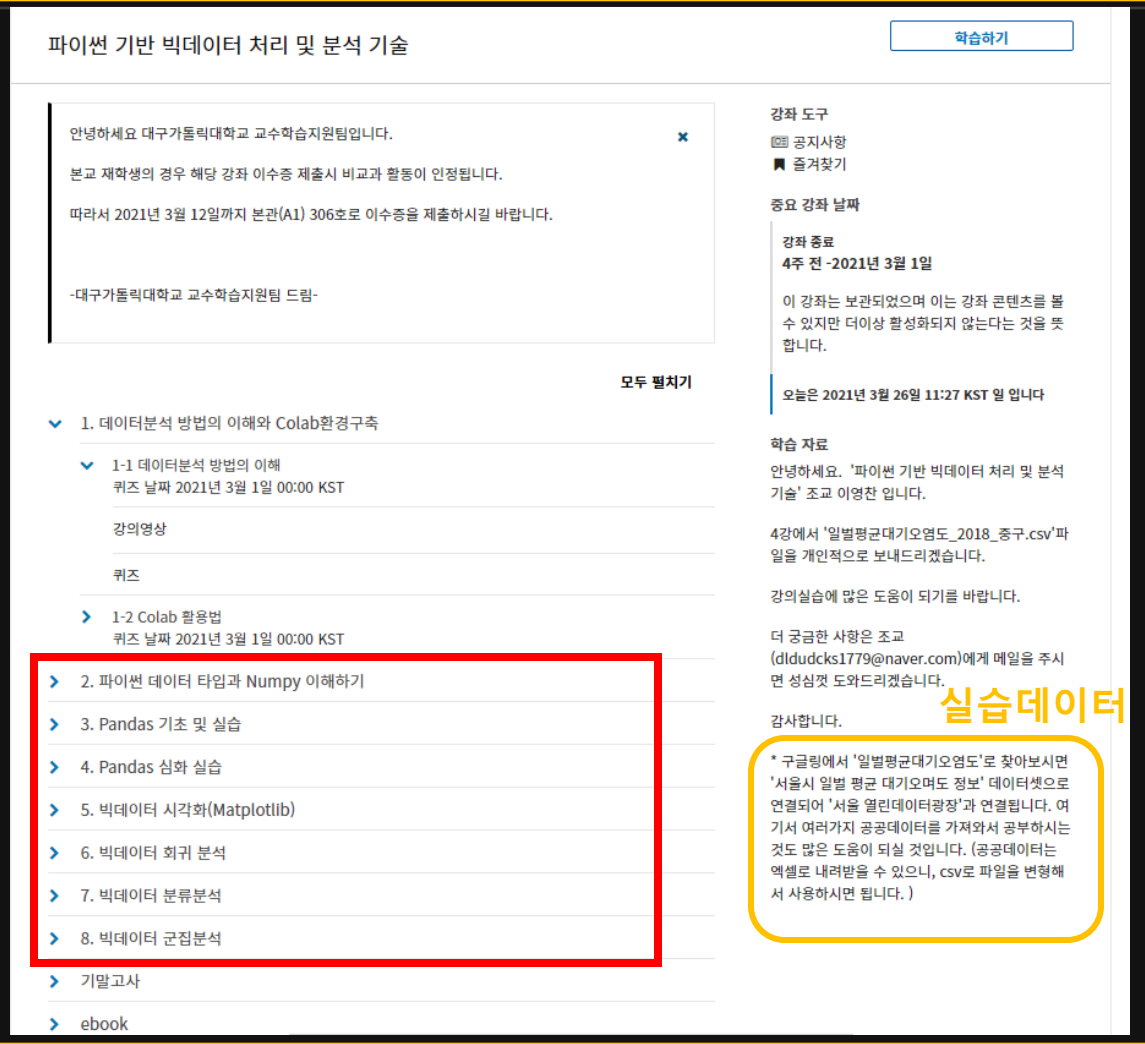
WHAT? 파이썬 기반 빅데이터 처리 및 분석 기술 | K-MOOC

스터디 시작 전(과제): 각 주차에 해당하는 강의를 미리 듣고 내용을 정리하여 팀즈 과제 란에 회신 (**내용 정리 방법은 자유**)

스터디 시간: 실습위주(데이터를 가지고 짝꿍이랑 같이 해당 내용 실습해보기 , 데이터는 '일별평균대기오염도')

공지톡 또는 팀즈 [study_math]-[파일]-[Scalar]

과제 내용: K-mooc강의를 들으시고 배운 내용을 블로그나 깃허브 노션 등 자신이 편한 곳에 정리를 한 후 그 링크를 팀즈 과제 란에 회신 해주세요. (실습코드는 제출 자유)



DALC 공통 기초 스터디 변경 계획 안내

WHEN? 일정:

0주차 3/13: OT

1주차 3/20: 깃헙과 코랩

2주차 3/27: 크롤링

3주차 4/3: 파이썬 데이터 타입과 Numpy 이해하기

4주차 4/10: pandas 기초 및 실습 + pandas 심화 실습

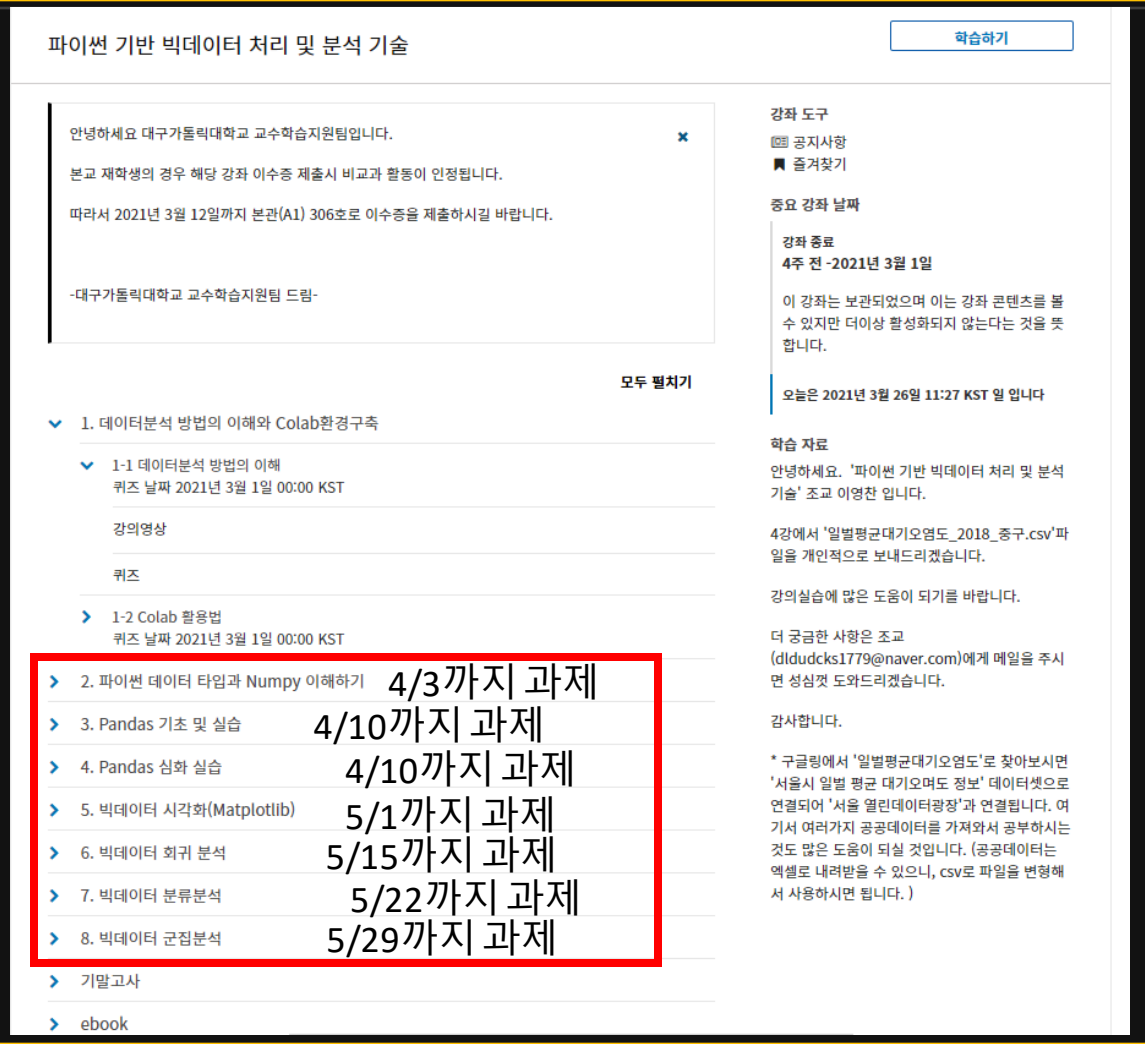
-중간고사-

5주차 5/1: 빅데이터 시각화

6주차 5/15: 빅데이터 회귀 분석

7주차 5/22: 빅데이터 분류 분석

8주차 5/29: 빅데이터 군집분석



데이터 수집 유용한 사이트

Kaggle
Imagenet (이미지데이터)
Quandl (해외 금융, 경제 관련)
KDnuggets
Data Science Central
UCI Machine Learning Repository
OECD Health Data (의료)
Google Trends
WHO (의료)
NASA EarthData
AMAZON Web Service open data
Pew Research Center Data (소셜트렌드)

통합데이터지도
AI Hub
공공데이터포털
국가통계포털
마이크로데이터
지역데이터개방
서울열린데이터광장
서울시연구데이터서비스
K-ICT 빅데이터센터-형태소사전
SK 빅데이터 허브 (통신)
한국소비자원참가격
네이버데이터랩
카카오데이터트렌드
오디피아 (기업데이터관련)
한국형 질문답변 데이터셋

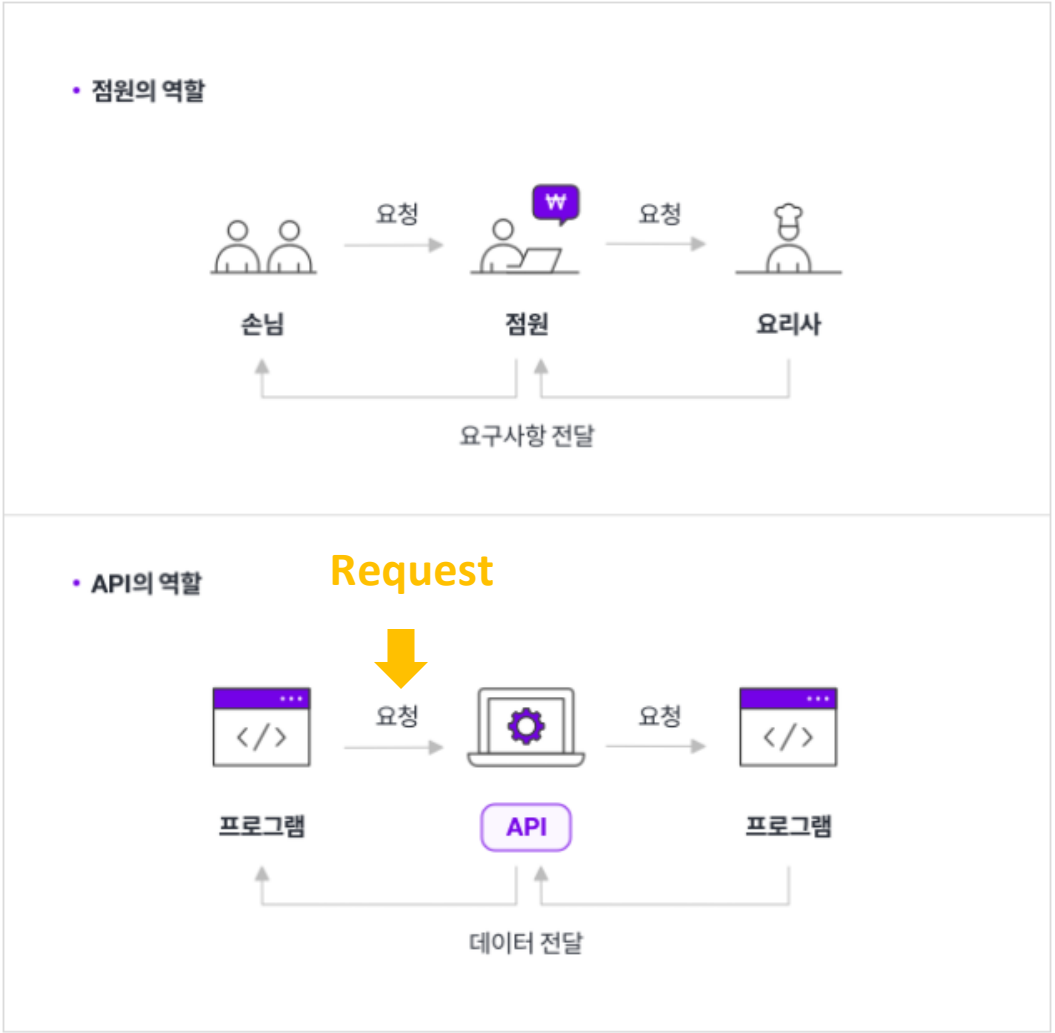
API란?

Application Programming Interface

- API는 응용 프로그램에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 뜻한다.

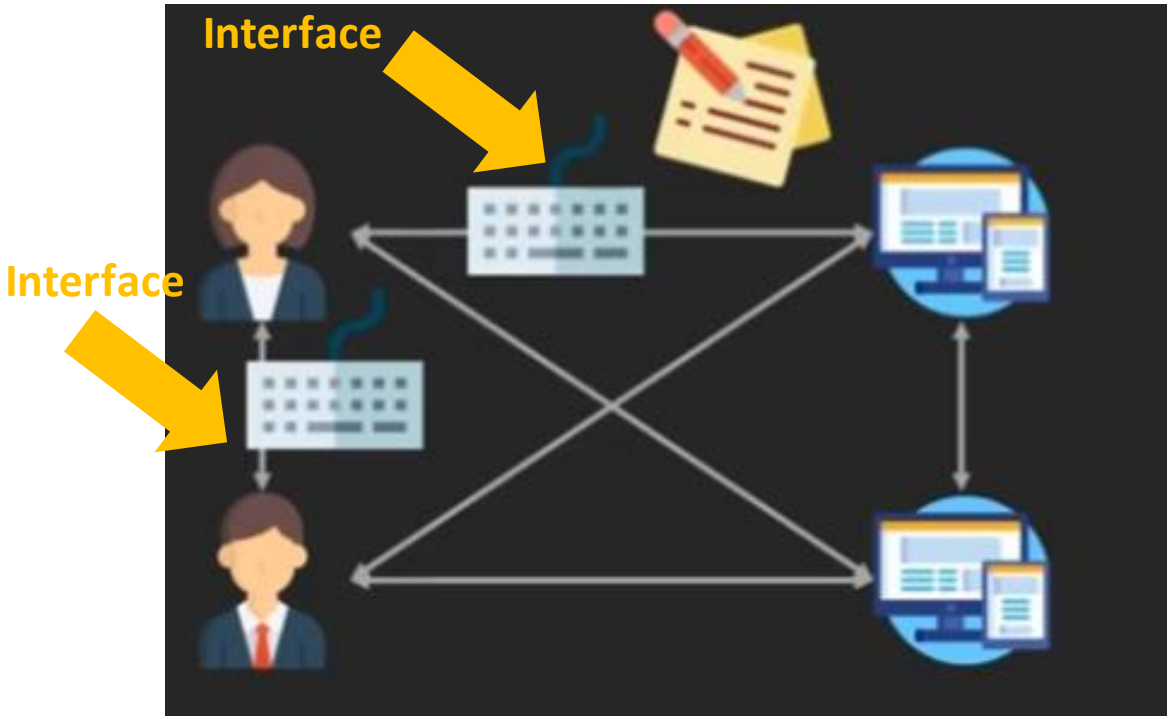
응용 예시:

도서관련 서비스 만들고 싶다!
->알라딘 api사용
증권관련 데이터로 주식주문 자동화를 해보고 싶다!
->대신증권 api, 키움증권api등
카카오톡의 기능을 내가 만든 서비스에 적용하고 싶다!
->카카오 api

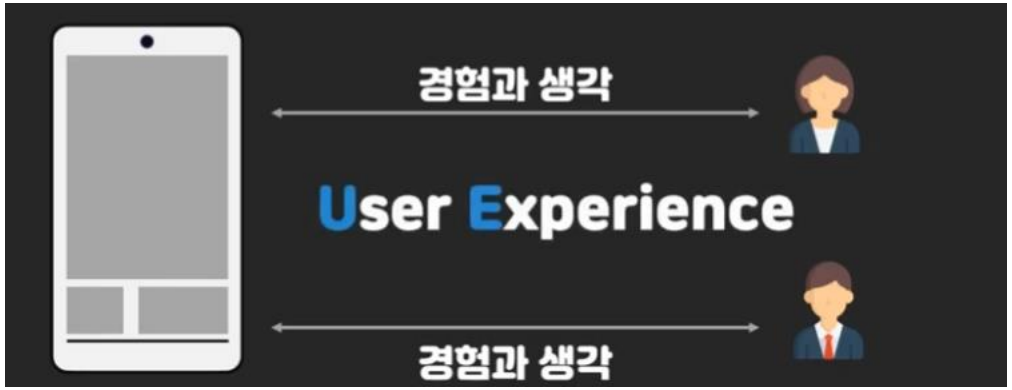


Interface

어떠한 두가지가 서로 연결되고 영향을 미칠 수 있는 장소/방법/상황



UI: (User Interface의 약자)
- 디지털 기기에 명령을 내리는 방법
- 요즘에는 UE라는 말을 많이 쓰기도 한다.



공공데이터 api 호출하는 방법:
https://www.youtube.com/watch?v=kA_46xWOhqY&t=1s

Crawler?

엄밀히 말하면..정확한 정의x

- Web상에 존재하는 Contents를 수집하는 작업 (파이썬을 통해 자료수집의 자동화를 의미한다.)
  1. HTML 페이지를 **가져와서**, HTML/CSS등을 **파싱**하고, 필요한 데이터만 추출하는 기법
  2. **Open API(Rest API)**를 제공하는 서비스에 Open API를 호출해서, 받은 데이터 중 필요한 데이터만 추출하는 기법
  3. **Selenium**등 브라우저를 프로그래밍으로 조작해서, 필요한 데이터만 추출하는 기법

- web crawling : 자동적으로 화면에 있는 data를 가져오는 것 **(실시간 연동, 자동으로 업데이트 됨)**
- web scrapping : **자동화 X** / scrapping 하는 시점에서의 데이터만 갖고오기!

=> 두 가지 모두 `웹 사이트를 분석해 원하는 데이터를 추출하는 과정` 이다.

사진출처:https://velog.io/@mowinckel/%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%A7%81-I

Crawling 개념

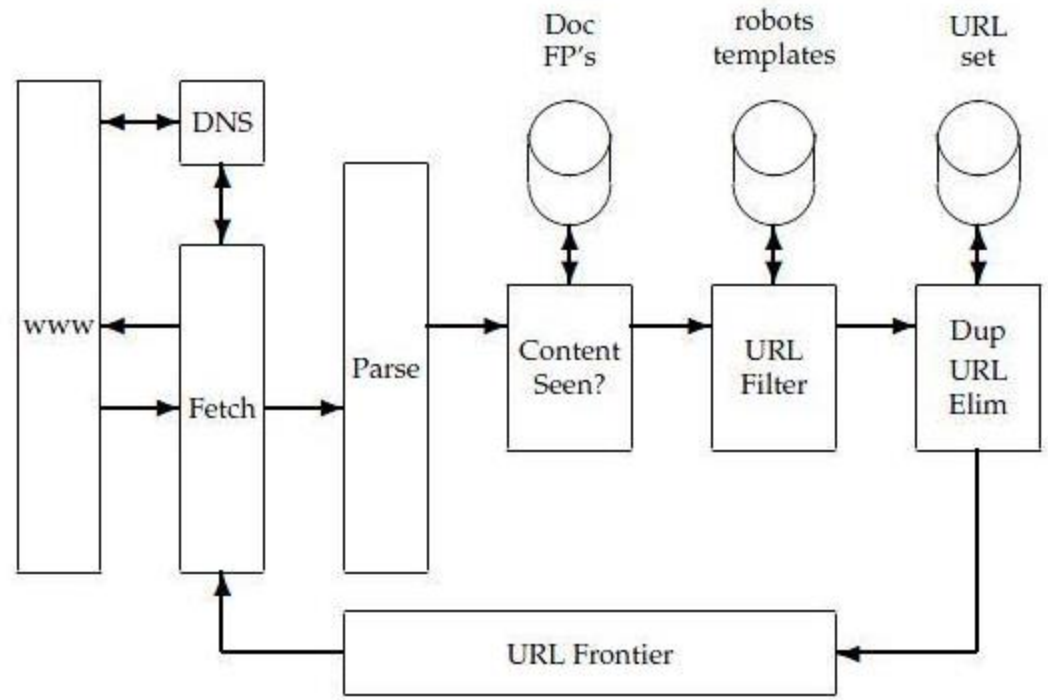


Figure 20.1 Basic crawler architecture.

Web Crawler?

"seed URL을 주면 관련된 URL을 찾아 내고, 그 URL들에서 또 다른 하이퍼 링크를 찾아내고 계속해서 이 과정을 반복하며 하이퍼 링크들을 다운로드하는 프로그램이다."
- 관련논문 MS 연구소 Marc Najork

Web Scraping

데이터 수집하는 작업 전체

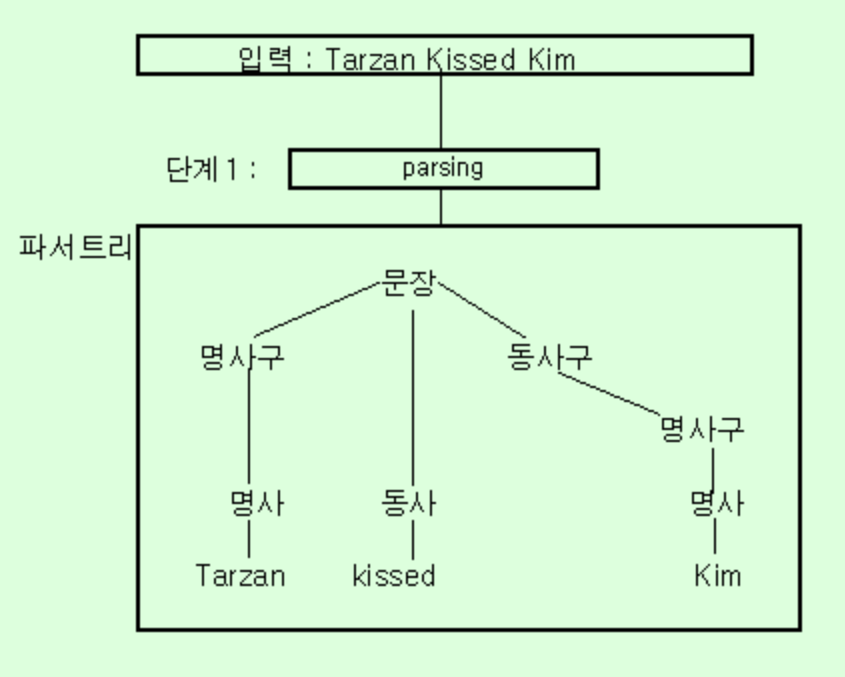
데이터분석에서 가장 널리 쓰이는 방법

- 프로그램을 만들어 웹서버에 쿼리를 보내 데이터를 요청(request)
- 이를 **파싱(Parsing)**해 필요한 정보를 추출하는 작업

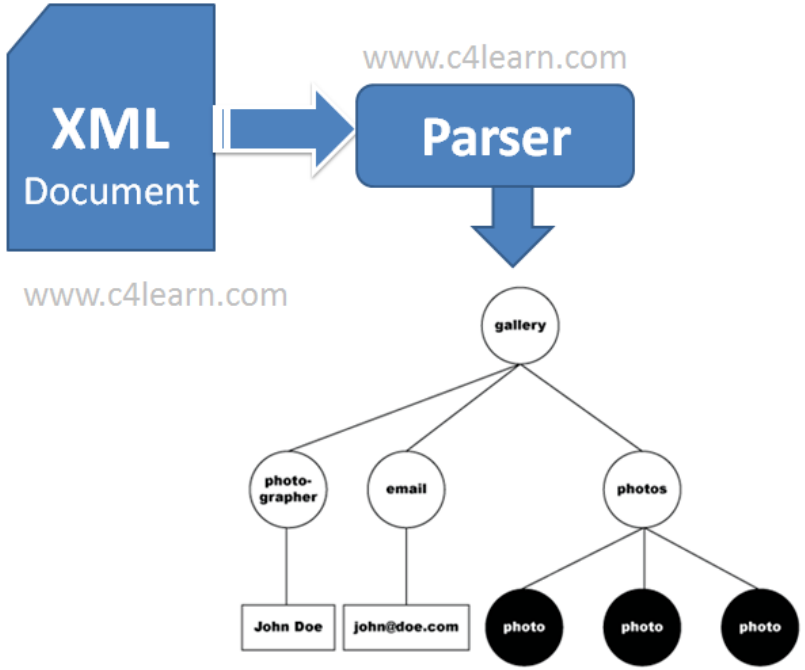
사진출처:https://velog.io/@mowinckel/%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%A7%81-I

Parsing

언어학



컴퓨터 과학

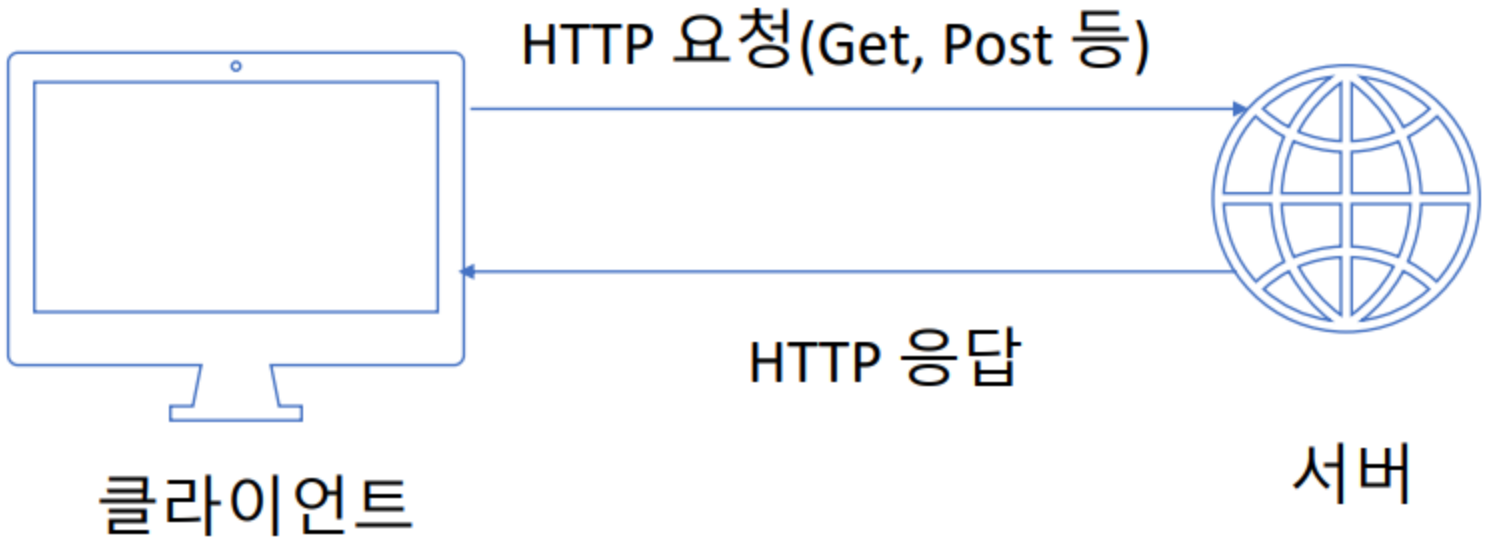


사진출처:https://velog.io/@mowinckel/%EC%9B%B9-%ED%81%AC%EB%A1%A4%EB%A7%81-I

HTTP

HyperText Transfer Protocol (HTTP): HTML 문서 등의 리소스를 전송하는 **프로토콜**

통신을 위한 약속, 통신의 유형, 내용 등을 미리 정의



HTTP요청 종류

Get 요청:데이터를 URL에 포함하여 전달(주로 리소스 요청에 사용)
동덕여대 : 네이버 통합검색 (naver.com)

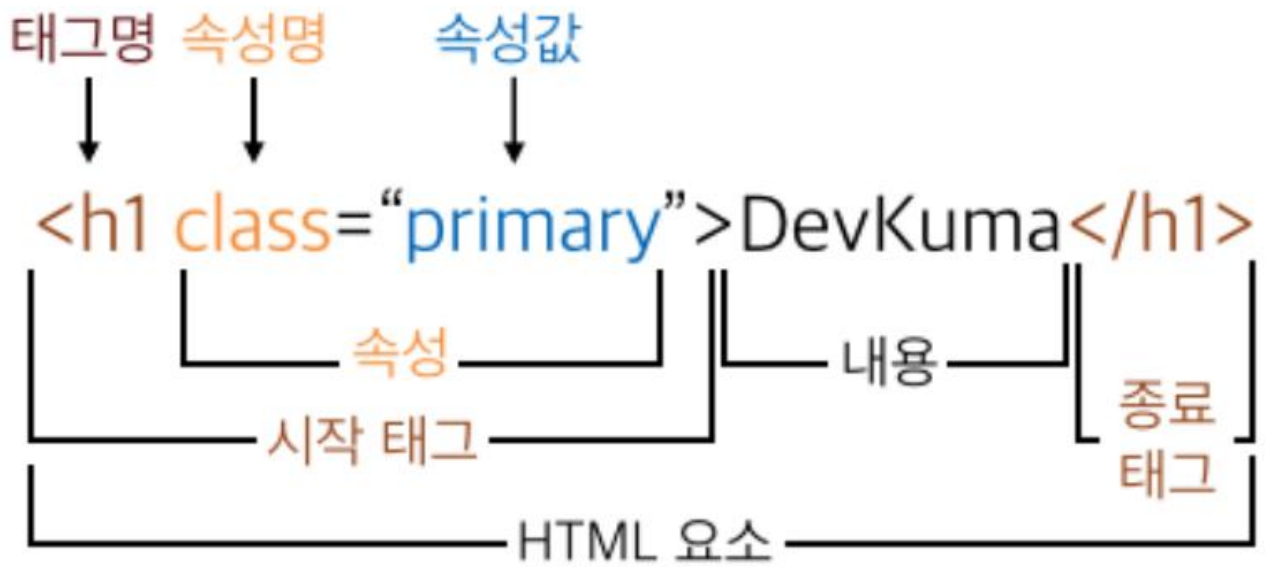
Post 요청: 데이터를 Form data에 포함하여 전달(주로 로그인에 사용
https://www.kangcom.com/member/member_check.asp

GET	POST
정보를 얻기 위함 (GET)	정보를 쓰기 위함 (POST)
Query String 활용	Request Body 활용
*Query String: 주소 뒤에 "?"를 붙인 뒤 정보 전송	보안 우수

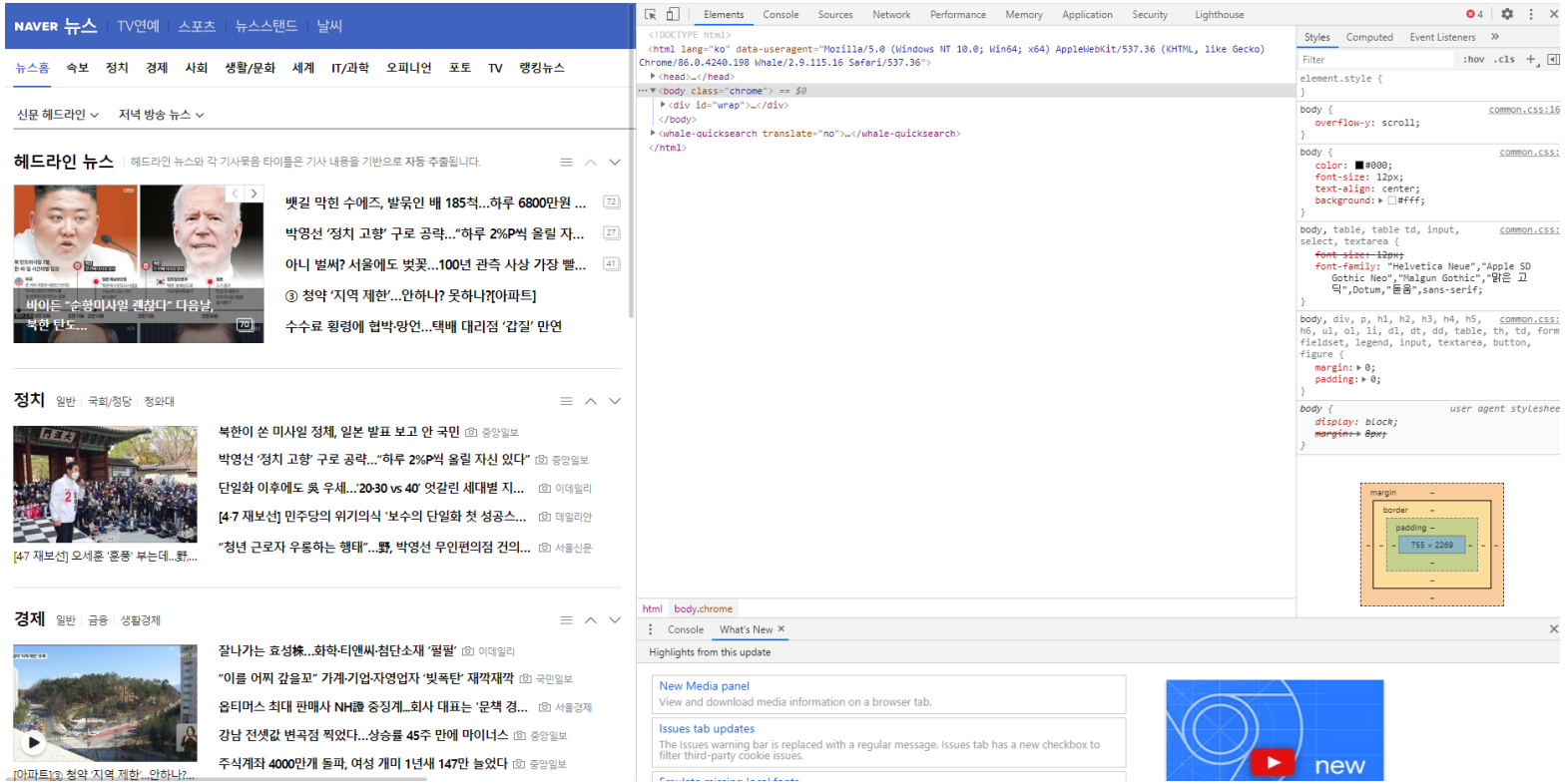
HTML

웹사이트를 생성하기 위한 언어로 문서와 문서가 링크로
연결되어 있고, 태그로 사용하는 언어

```
<div class="greenbox">
    <input type="text" id="nx_query" name="query" class="box_window"
    maxlength="255" accesskey="s" value="동덕여대" autocomplete="off"
    placeholder="검색어를 입력해 주세요." data-atcmp-element> == $0
</div>
```



크롬 개발자 도구



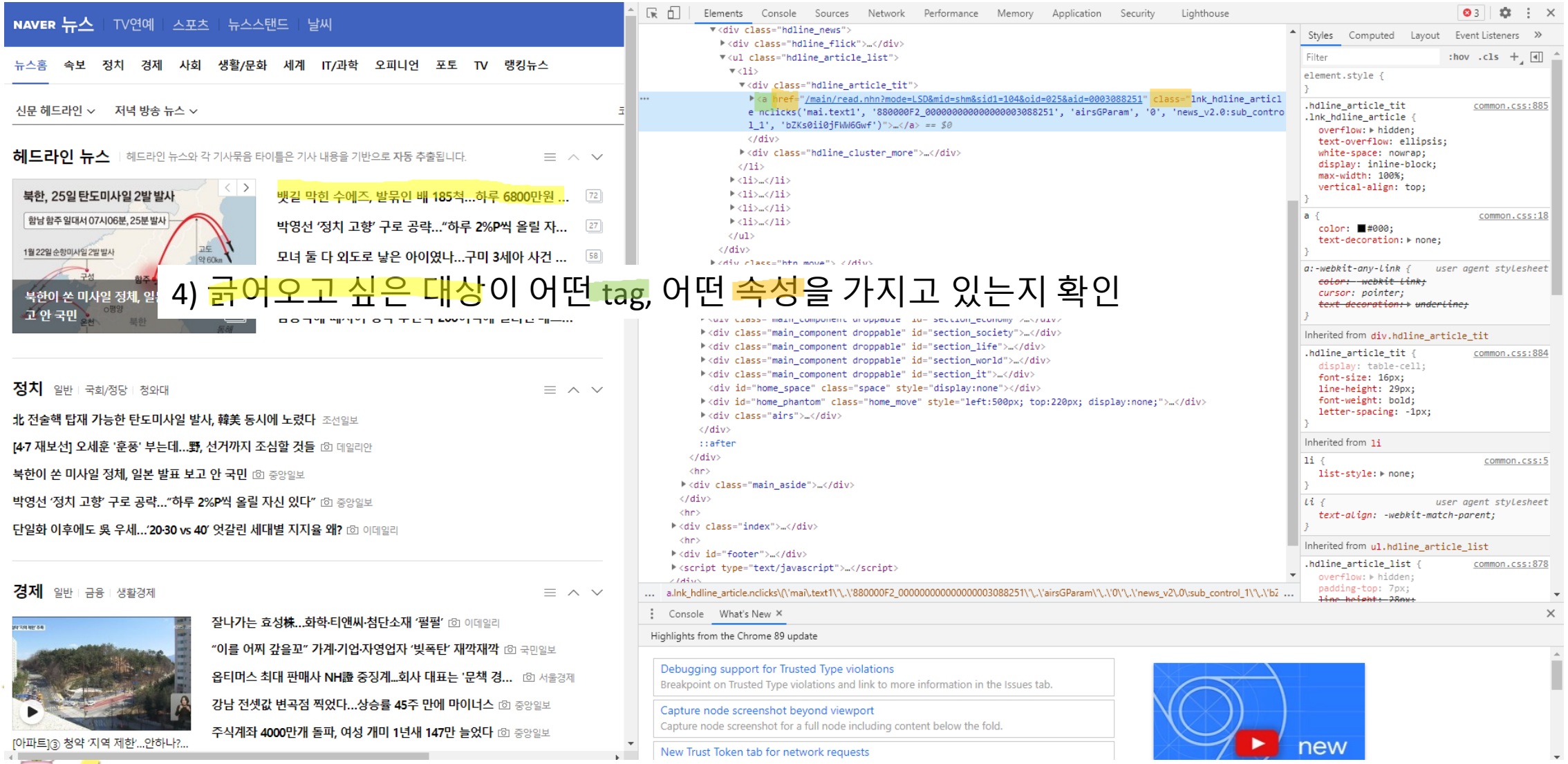
1) 웹페이지에서 F12 키 누르기

크롬 개발자 도구

2) Crtl + shift + c
또는 □ 클릭

3) 마우스를 제목 등에 올려놓고 클릭

크롬 개발자 도구



4) 긁어오고 싶은 대상이 어떤 tag, 어떤 속성을 가지고 있는지 확인

기초 크롤링 방법 코드 설명

BeautifulSoup Usage

1. pip install beautifulsoup4

```
1  from urllib.request import urlopen
2  from bs4 import BeautifulSoup
3
4  html=urlopen("http://www.abc.co.kr")
5
6  bsObj = BeautifulSoup(html.read(), "html.parser")
7
8  bsObj.h1
```

BeautifulSoup가 필요한 이유는 태그 관리들을 하기 위해 집어넣는것.

태그 찾는 방법

2. Find() / findAll()

findAll(tag, attributes, recursive, text, limit, keywords)
find(tag, attributes, recursive, text, keywords)

```
bs.findAll({‘h1’, ‘h2’, h3’, ‘h4’, ‘h5’, ‘h6’})
bs.findAll('span', {'class': 'green'})
bs.findAll(class=“green”)
```

태그 찾는 방법

2. find() / findAll()

findAll(tag, attributes, recursive, text, limit, keywords)
find(tag, attributes, recursive, text, keywords)

recursive: 문서의 깊이 지정
True: 일치하는 태그의 하위 문서들… => default
False: 문서의 최상위 태그만 검색

text: 지정한 컨텐츠 텍스트와 일치하는 텍스트 찾기
bs.findAll(text = 'the prince')

태그 찾는 방법

Html

----------body

------div

-----h1 이 부분을 찾는다고 가정하자

-----div

-----table.#giftList

----tr

-------th

-------th

-------th

-------th 이 부분을 찾는다고 가정하자

----tr.gift#gift1

-------td

-------td

----span.excitingNote

------td

------td

-----img

----- …

HTML Tag

Children

항상 부모보다 한 단계 아래 태그

bs.find('table').children

Descendants

조상보다 몇 단계든 아래에 있을 수 있음

bs.find('table').descendants

Sibling

같은 위치에 있는 태그

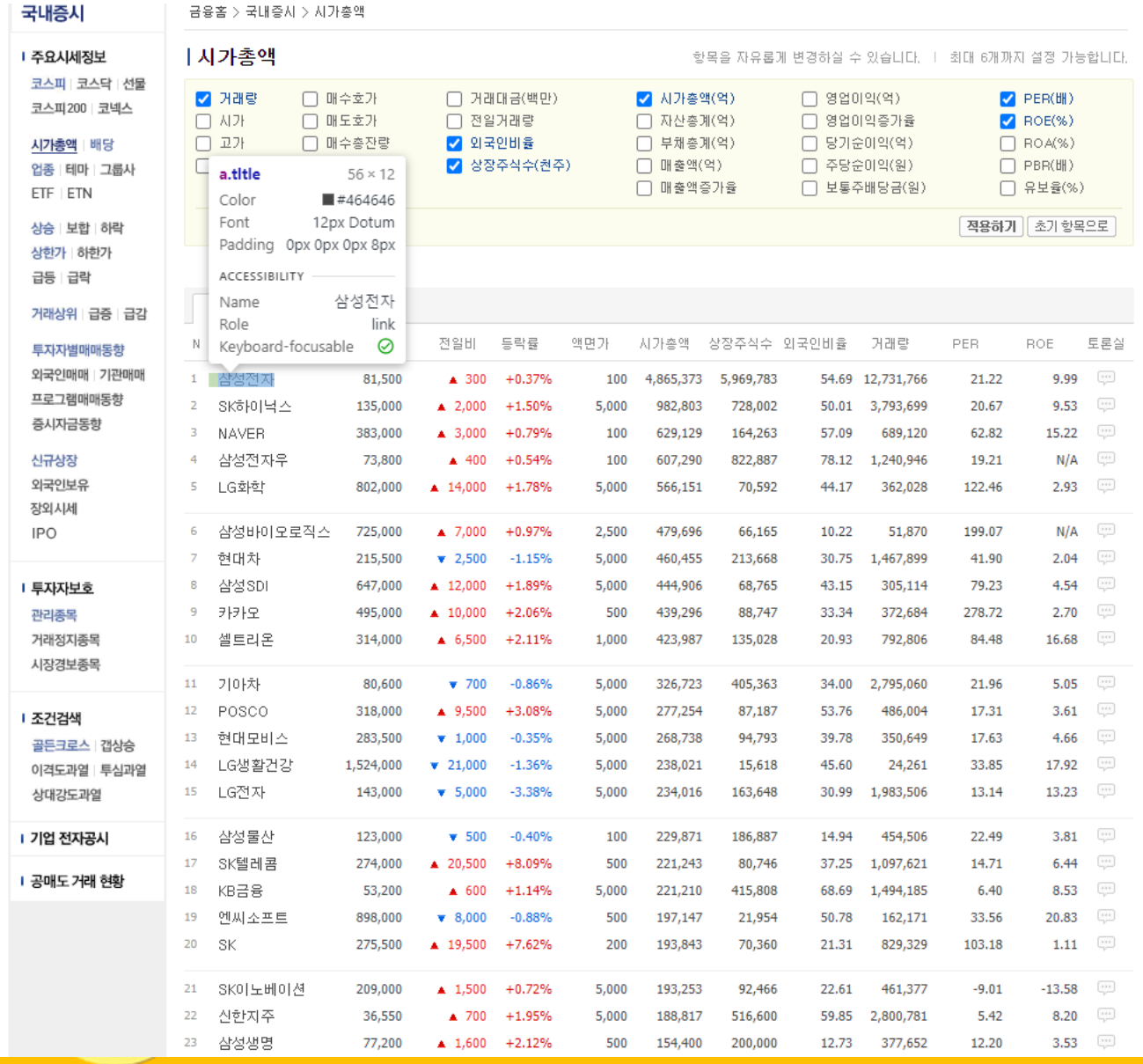
Bs.find('table').tr.next_siblings
previous_siblings, next_sibling, previous_sibling

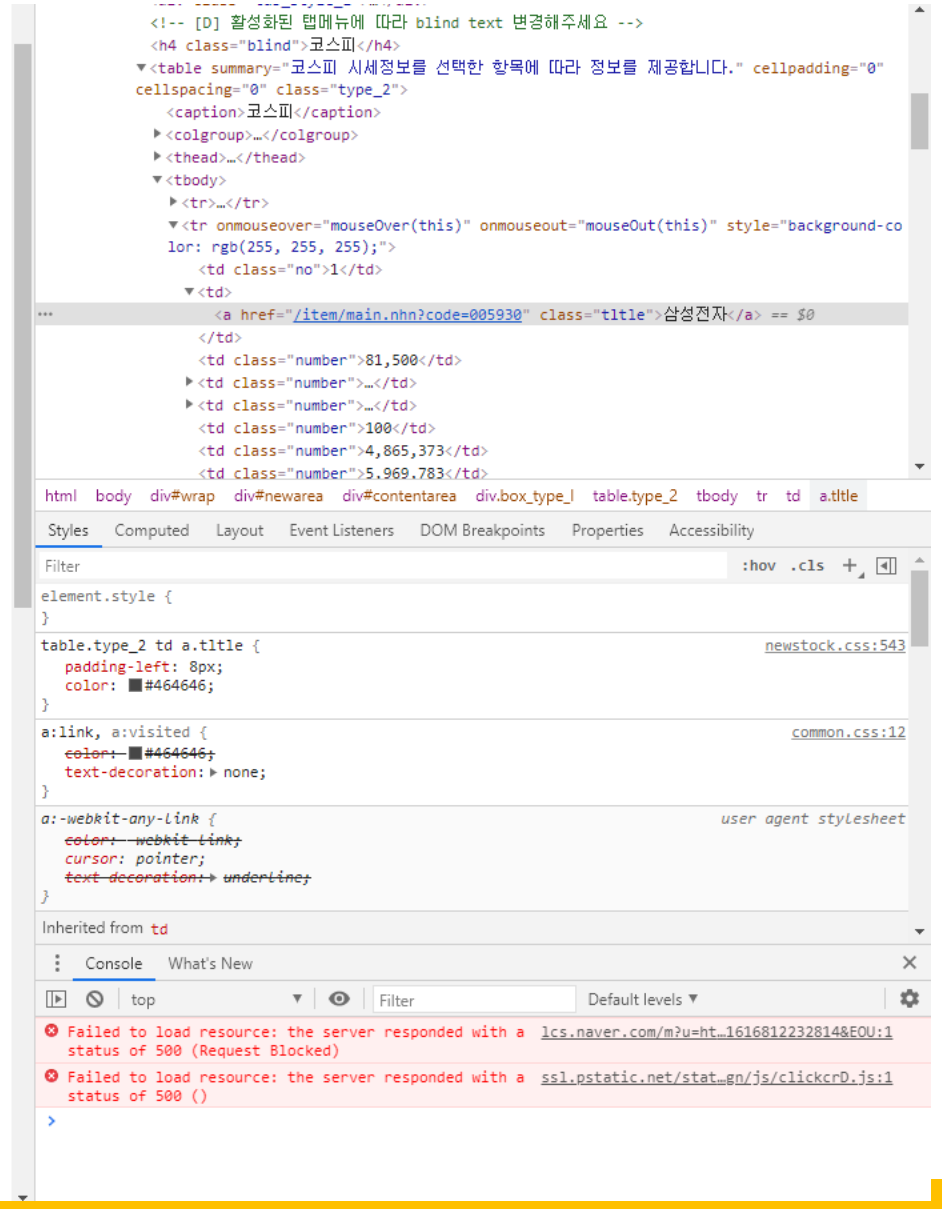
Parent

현 태그에서 한 단계 위의 태그

bs.find('img').parent

Tag 확인해서 코드에 넣을 때



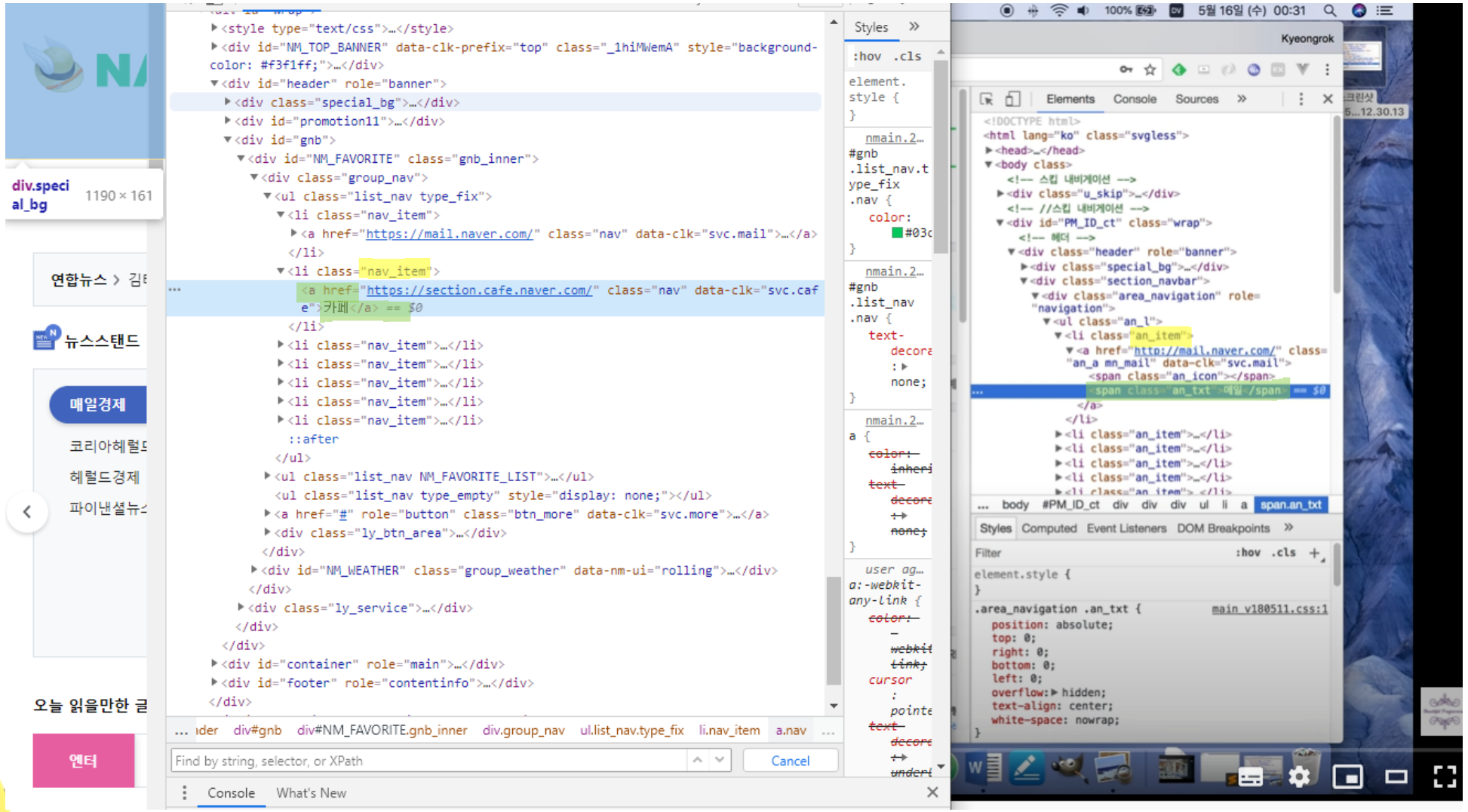


```python

data_rows = soup.find("table", attrs={"class":"type_2"}).find("tbody").find_all("tr")
for row in data_rows:
  columns = row.find_all("td")
  data = [column.get_text() for column in columns]
  print(data)
```

```
<!-- [D] 활성화된 탭메뉴에 따라 blind text 변경해주세요 -->
<h4 class="blind">코스피</h4>
<table summary="코스피 시세정보를 선택한 항목에 따라 정보를 제공합니다." cellpadding="0" cellspacing="0" class="type_2">
  <caption>코스피</caption>
  <colgroup>…</colgroup>
  <thead>…</thead>
  <tbody>
    <tr>…</tr>
    <tr onmouseover="mouseOver(this)" onmouseout="mouseOut(this)" style="background-color: rgb(255, 255, 255);">
      <td class="no">1</td>
      <td>
        <a href="/item/main.nhn?code=005930" class="tltle">삼성전자</a> == $0
      </td>
      <td class="number">81,500</td>
      <td class="number">
        <img src="https://ssl.pstatic.net/imgstock/images/images4/ico_up.gif" width="7" height="6" style="margin-right:4px;" alt="상승">
        <span class="tah p11 red02">
                    300
                    </span>
      </td>
      <td class="number">…</td>
      <td class="number">100</td>
      <td class="number">4,865,373</td>
      <td class="number">5,969,783</td>
      <td class="number">54.69</td>
      <td class="number">12,731,766</td>
      <td class="number">21.22</td>
      <td class="number">9.99</td>
      <td class="center">…</td>
    </tr>
    <tr onmouseover="mouseOver(this)" onmouseout="mouseOut(this)" style="background-color: rgb(255, 255, 255);">…</tr>
    <tr onmouseover="mouseOver(this)" onmouseout="mouseOut(this)"> </tr>
```

Tag 주의!


div.special_bg
1190 × 161
연합뉴스 › 김
뉴스스탠드
메일경제
코리아헤럴드
헤럴드경제
파이낸셜뉴스
오늘 읽을만한 글
엔터
카페 == $0
Styles
Elements
Console
Sources
Computed
Event Listeners
DOM Breakpoints
Find by string, selector, or XPath
Cancel
Console
What's New

참고자료 및 강의

책: Web Scraping with Python (파이썬으로 웹 크롤러 만들기)

영상:
파이썬 크롤러 만들기 (Kyeongrok Kim)

나도코딩 웹스크레이핑 강의

기타 참고하면 좋은 사이트:
Requests 모듈 설명
네이버 헤드라인 뉴스 가져오기

하드웨어

강의: 시스템 프로그래밍 기초

PC하드웨어

정리된 글:

컴퓨터 구성 (안경잡이 개발자 블로그)

실습시간