

---

# 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)

## 파이썬 기반 빅데이터 처리 및 분석 기술

학습하기

안녕하세요 대구가톨릭대학교 교수학습지원팀입니다.

본교 재학생의 경우 해당 강좌 이수증 제출시 비교과 활동이 인정됩니다.

따라서 2021년 3월 12일까지 본관(A1) 306호로 이수증을 제출하시길 바랍니다.

-대구가톨릭대학교 교수학습지원팀 드림-

모두 펼치기

### 1. 데이터분석 방법의 이해와 Colab환경구축

1-1 데이터분석 방법의 이해  
퀴즈 날짜 2021년 3월 1일 00:00 KST

강의영상

퀴즈

1-2 Colab 활용법  
퀴즈 날짜 2021년 3월 1일 00:00 KST

### 2. 파이썬 데이터 타입과 Numpy 이해하기

### 3. Pandas 기초 및 실습

### 4. Pandas 심화 실습

### 5. 빅데이터 시각화(Matplotlib)

### 6. 빅데이터 회귀 분석

### 7. 빅데이터 분류분석

### 8. 빅데이터 군집분석

### 기말고사

### ebook

강좌 도구

공지사향

즐거찾기

중요 강좌 날짜

강좌 종료  
4주 전 -2021년 3월 1일

이 강좌는 보관되었으며 이는 강좌 콘텐츠를 볼 수 있지만 더이상 활성화되지 않는다는 것을 뜻합니다.

오늘은 2021년 3월 26일 11:27 KST 일입니다

학습 자료

안녕하세요. '파이썬 기반 빅데이터 처리 및 분석 기술' 조교 이영찬 입니다.

4강에서 '일별평균대기오염도\_2018\_중구.csv'파일을 개인적으로 보내드리겠습니다.

강의실습에 많은 도움이 되기를 바랍니다.

더 궁금한 사항은 조교 (dludcks1779@naver.com)에게 메일을 주시면 성심껏 도와드리겠습니다.

감사합니다.

실습데이터

\* 구글링에서 '일별평균대기오염도'로 찾아보시면 '서울시 일별 평균 대기오염도 정보' 데이터셋으로 연결되어 '서울 열린데이터광장'과 연결됩니다. 여기서 여러가지 공공데이터를 가져와서 공부하시는 것도 많은 도움이 되실 것입니다. (공공데이터는 엑셀로 내려받을 수 있으니, csv로 파일을 변형해서 사용하시면 됩니다.)

# 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

파이썬 기반 빅데이터 처리 및 분석 기술

학습하기

안녕하세요 대구가톨릭대학교 교수학습지원팀입니다.

본교 재학생의 경우 해당 강좌 이수증 제출시 비교과 활동이 인정됩니다.

따라서 2021년 3월 12일까지 본관(A1) 306호로 이수증을 제출하시길 바랍니다.

-대구가톨릭대학교 교수학습지원팀 드림-

모두 펼치기

1. 데이터분석 방법의 이해와 Colab환경구축

1-1 데이터분석 방법의 이해  
퀴즈 날짜 2021년 3월 1일 00:00 KST

강의영상

퀴즈

1-2 Colab 활용법  
퀴즈 날짜 2021년 3월 1일 00:00 KST

2. 파이썬 데이터 타입과 Numpy 이해하기 4/3까지 과제

3. Pandas 기초 및 실습 4/10까지 과제

4. Pandas 심화 실습 4/10까지 과제

5. 빅데이터 시각화(Matplotlib) 5/1까지 과제

6. 빅데이터 회귀 분석 5/15까지 과제

7. 빅데이터 분류분석 5/22까지 과제

8. 빅데이터 군집분석 5/29까지 과제

기말고사

ebook

강좌 도구

공지사항

즐거찾기

중요 강좌 날짜

강좌 종료

4주 전 -2021년 3월 1일

이 강좌는 보관되었으며 이는 강좌 콘텐츠를 볼 수 있지만 더이상 활성화되지 않는다는 것을 뜻합니다.

오늘은 2021년 3월 26일 11:27 KST 일입니다

학습 자료

안녕하세요. '파이썬 기반 빅데이터 처리 및 분석 기술' 조교 이영찬 입니다.

4강에서 '일별평균대기오염도\_2018\_중구.csv'파일을 개인적으로 보내드리겠습니다.

강의실습에 많은 도움이 되기를 바랍니다.

더 궁금한 사항은 조교 (dldudcks1779@naver.com)에게 메일을 주시면 성심껏 도와드리겠습니다.

감사합니다.

\* 구글링에서 '일별평균대기오염도'로 찾아보시면 '서울시 일별 평균 대기오염도 정보' 데이터셋으로 연결되어 '서울 열린데이터광장'과 연결됩니다. 여기서 여러가지 공공데이터를 가져와서 공부하시는 것도 많은 도움이 되실 것입니다. (공공데이터는 엑셀로 내려받을 수 있으나, csv로 파일을 변형해서 사용하시면 됩니다.)

# 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](#)

 통합 데이터 지도

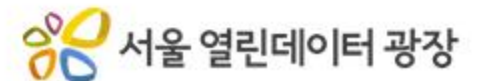
 AI Hub

 DATA 공공데이터포털  
GO KR

 KOSIS 국가통계포털  
Korean Statistical Information Service

 MDIS  
MicroData Integrated Service

 LOCALDATA

 서울 열린데이터 광장



# API란?

## Application Programming Interface

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

### 응용 예시:

도서관 관련 서비스 만들고 싶다!

->알라딘 api사용

증권 관련 데이터로 주식주문 자동화를 해보고 싶다!

->대신증권 api, 키움증권api등

카카오톡의 기능을 내가 만든 서비스에 적용하고 싶다!

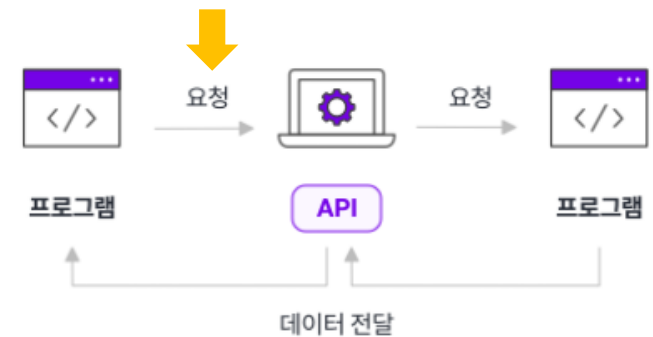
->카카오 api

#### • 점원의 역할



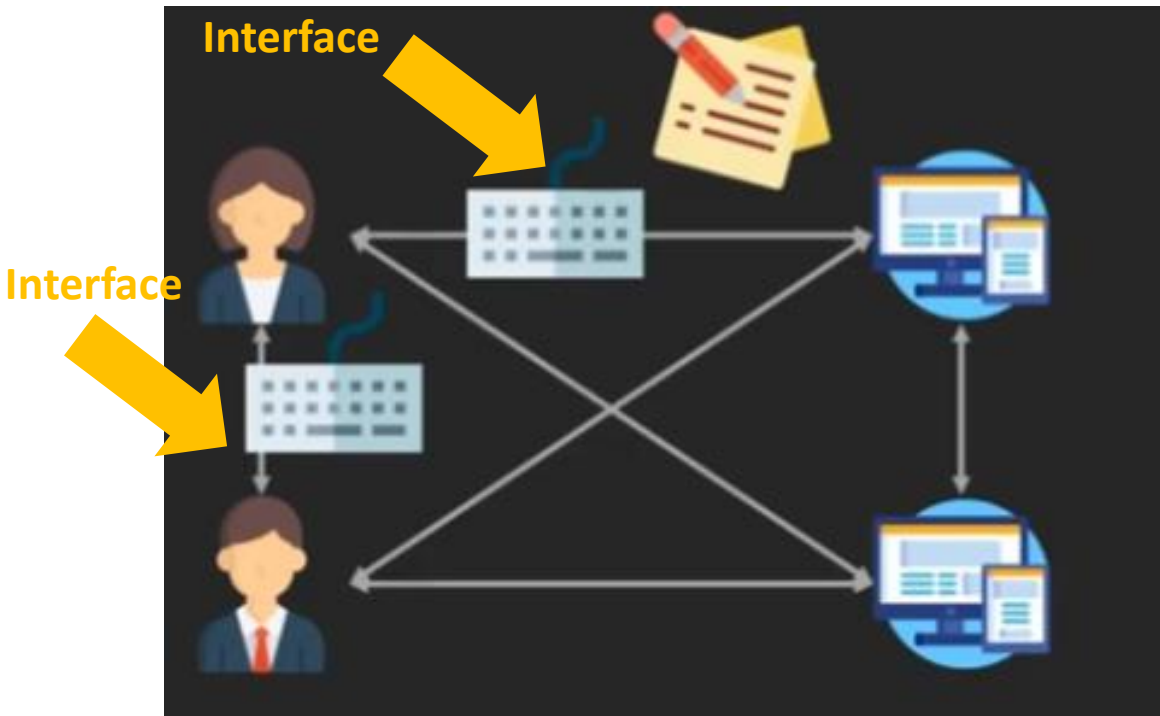
#### • API의 역할

### Request

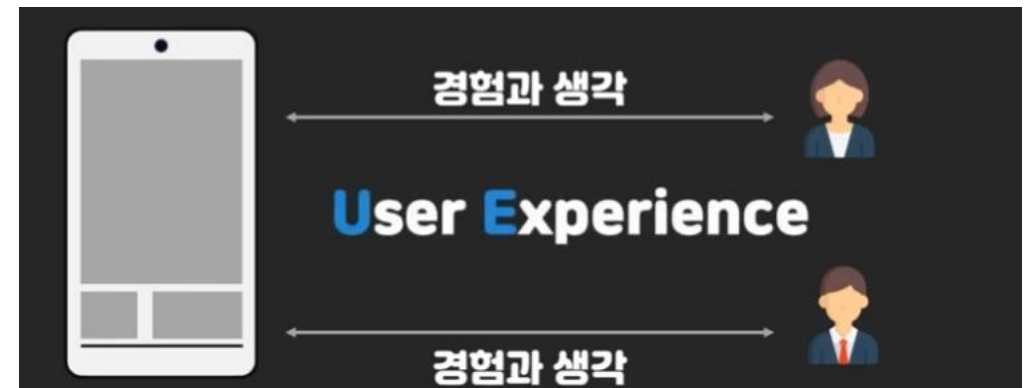


# Interface

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](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 파싱하고, 필요한 데이터만 추출하는 기법
2. Open API(Rest API)를 제공하는 서비스에 Open API를 호출해서, 받은 데이터 중 필요한 데이터만 추출하는 기법
3. Selenium등 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.



# Crawling 개념

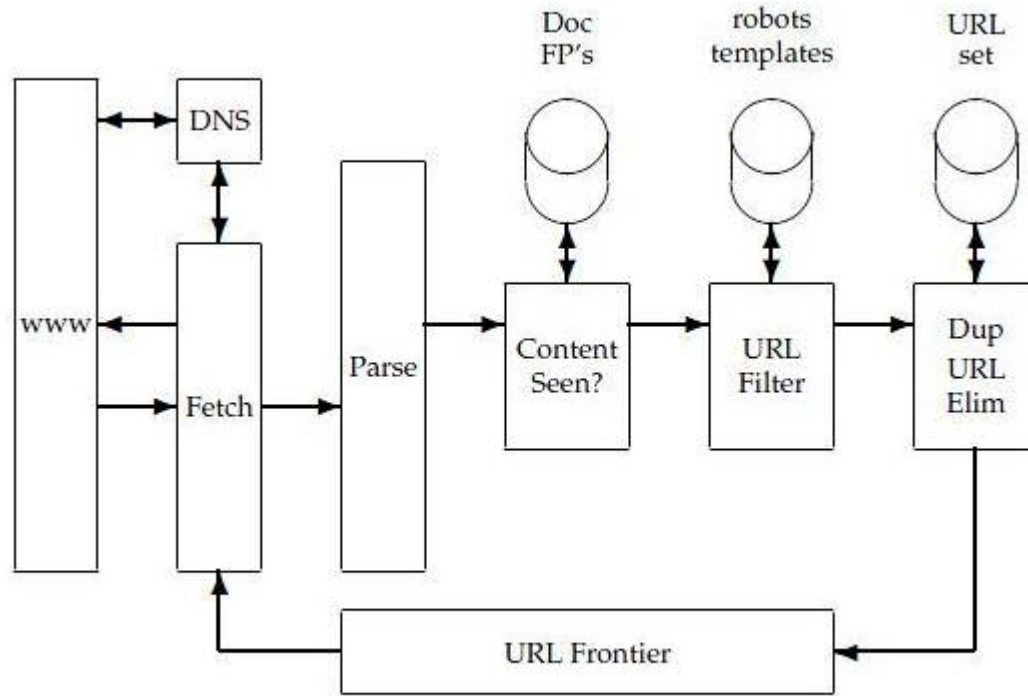


Figure 20.1 Basic crawler architecture.

## 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

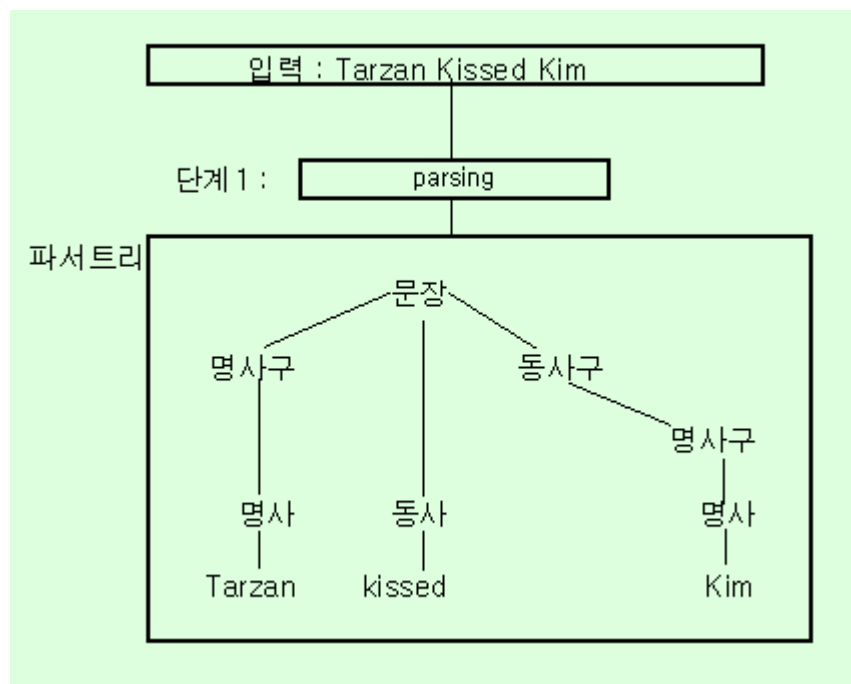
- 프로그램을 만들어 웹서버에 쿼리를 보내 데이터를 요청(request)
- 이를 **파싱(Parsing)**해 필요한 정보를 추출하는 작업



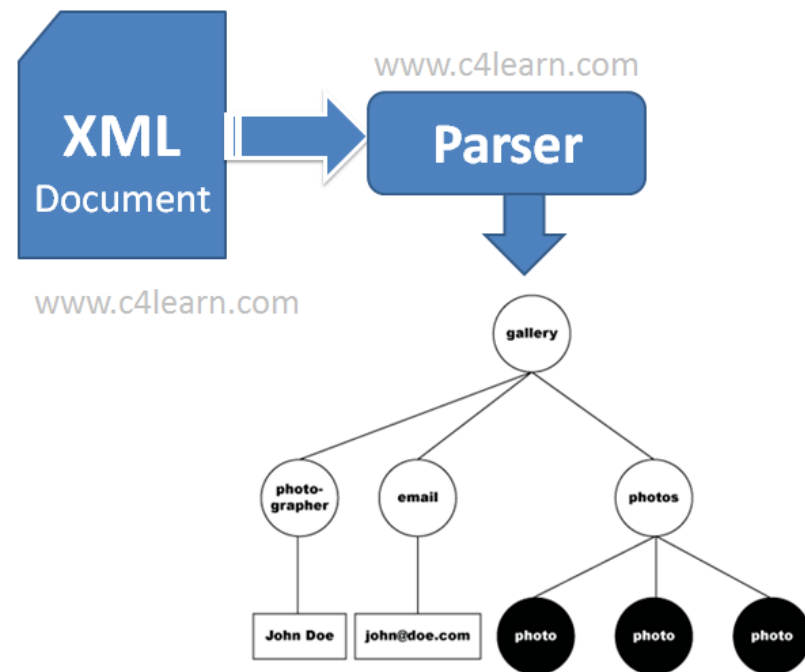


# Parsing

## 언어학



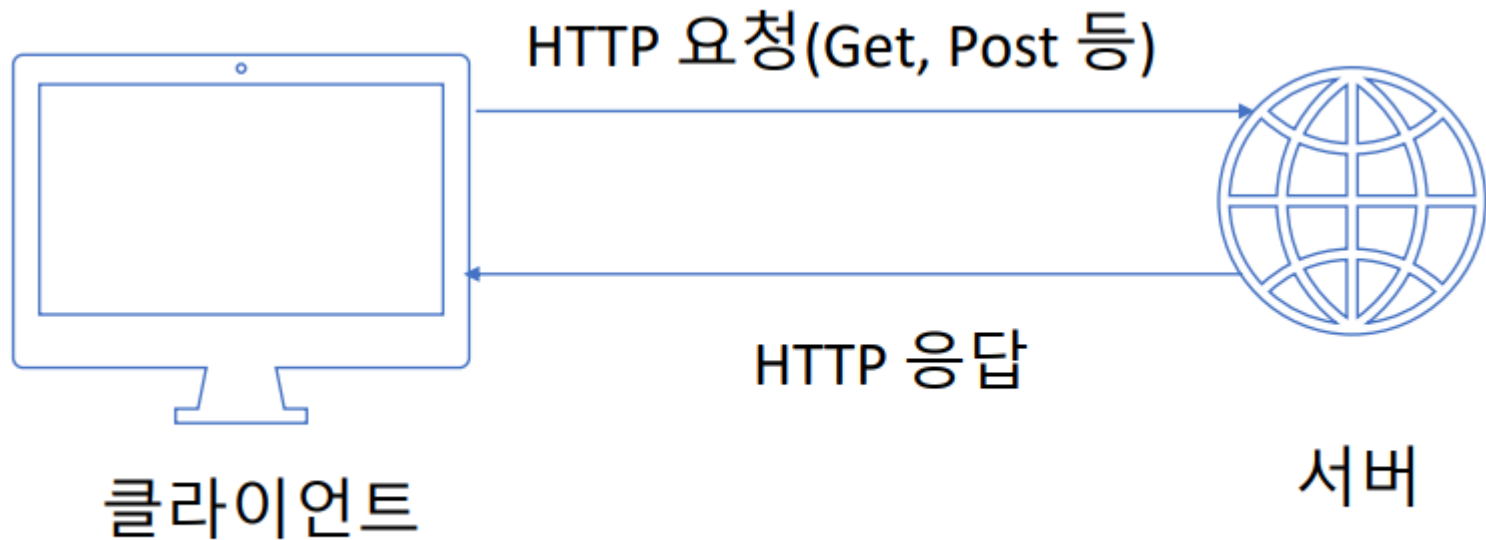
## 컴퓨터 과학



# 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 \(naver.com\)](https://www.naver.com)

Post 요청: 데이터를 Form data에 포함하여 전달(주로 로그인에 사용)

[https://www.kangcom.com/member/member\\_check.asp](https://www.kangcom.com/member/member_check.asp)

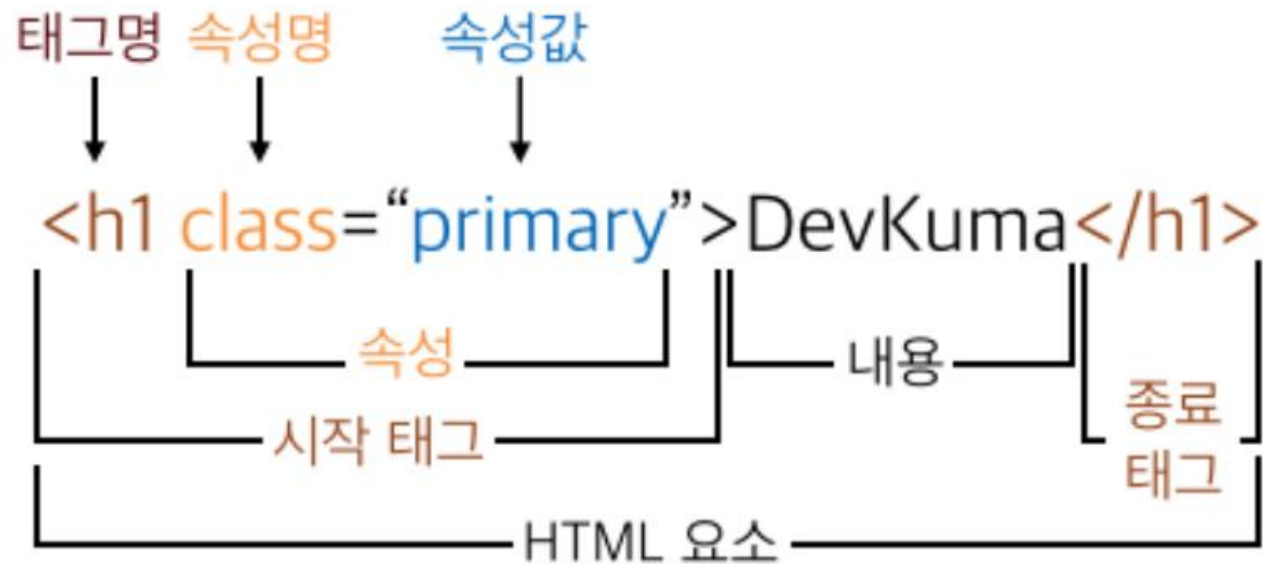
GET	POST
정보를 얻기 위함 (GET)	정보를 쓰기 위함 (POST)
Query String 활용	Request Body 활용
*Query String: 주소 뒤에 "?"를 붙인 뒤 정보 전송	보안 우수



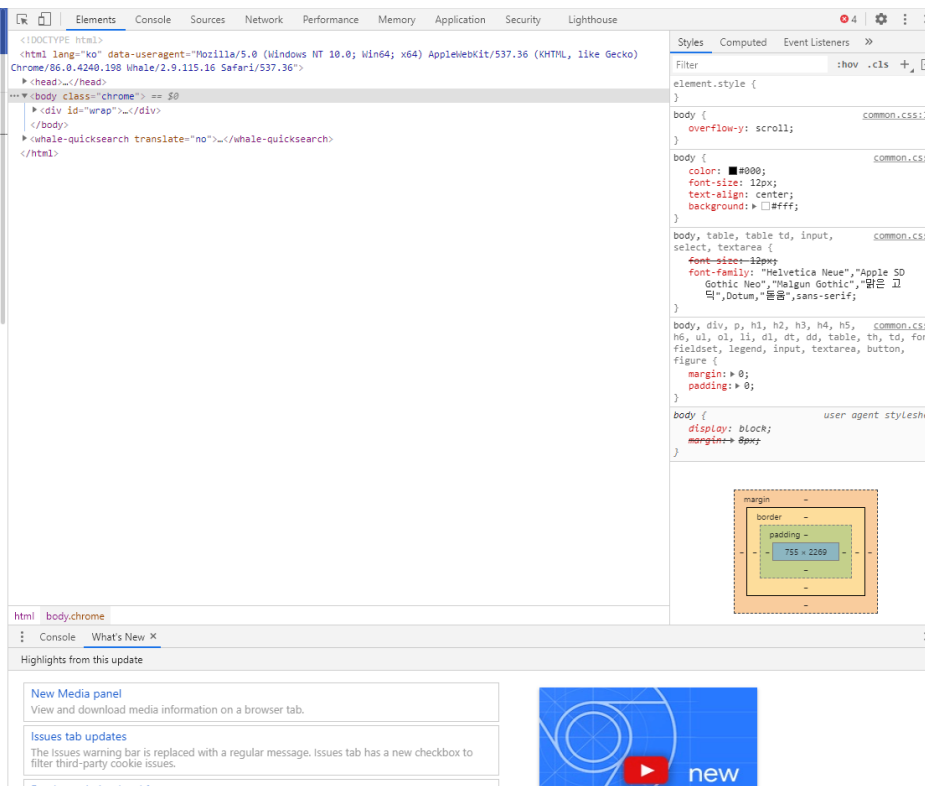
# HTML

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

The screenshot shows the NAVER news homepage. The Chrome DevTools Accessibility Inspector is open, displaying the 'a.Ink\_hdline\_article.nclicks' element. The inspector shows the following properties:

- Color: #000000
- Font: 16px "Helvetica Neue", "Apple SD Gothic..."
- ACCESSIBILITY: Aa 21
- Name: 뱃길 막힌 수에즈, 발류인 배 185척...하...
- Role: link
- Keyboard-focusable: true

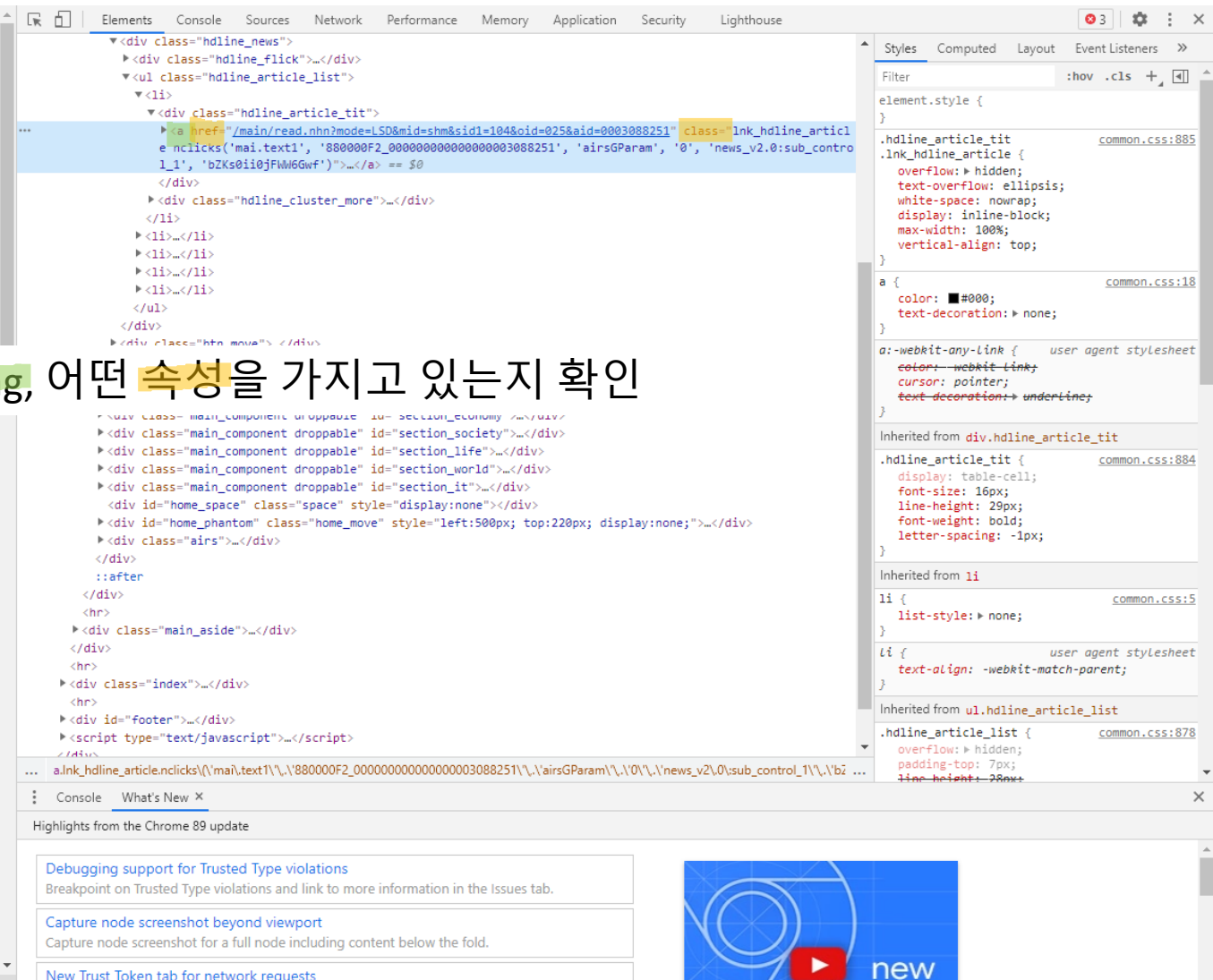
The browser's address bar shows the URL 'a.Ink\_hdline\_article.nclicks' with a red box highlighting the 'a' link. The page content includes news headlines about North Korea's missile tests and the Suez Canal closure.

2) Ctrl + shift + c  
또는  클릭

3) 마우스를 제목 등에 올려놓고 클릭



# 크롬 개발자 도구



# 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 콘텐츠 텍스트와 일치하는 텍스트 찾기

bs.findAll(text = 'the prince')



# 태그 찾는 방법

Html

-----body

-----div

-----h1

-----div

-----table.#giftList

----tr

-----th

-----th

-----th

-----th

----tr.gift#gift1

-----td

-----td

----span.excitingNote

-----td

-----td

-----img

----- ...

이 부분을 찾다고 가정하자

이 부분을 찾다고 가정하자



# 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
```





```

11
12 data_rows = soup.find("table", attrs={"class": "type_2"}).find("tbody").find_all("tr")
13 for row in data_rows:
14     columns = row.find_all("td")
15     data = [column.get_text() for column in columns]
16     print(data)

```

```

<!-- [D] 활성화된 메뉴에 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="title">삼성전자</a> == $0
      </td>
      <td class="number">81,500</td>
      <td class="number">
        
        <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)" style="background-color: rgb(255, 255, 255);">...</tr>

```

# Tag 주의!

The image shows a screenshot of a web browser displaying the Naver homepage. The browser's developer tools are open, showing the HTML structure and CSS styles. The HTML structure is visible on the left, and the CSS styles are visible on the right. The HTML structure shows a navigation bar with links to various services, including Naver, Mail, and News. The CSS styles show the layout and styling of the page, including the navigation bar and the main content area. The browser's address bar shows the URL "http://www.naver.com". The browser's title bar shows "Kyeongrok". The browser's status bar shows the date and time "5월 16일 (수) 00:31".

div.speci  
al\_bg 1190 x 161

연합뉴스 > 김

뉴스스탠드

메일경제

코리아헤럴드

헤럴드경제

파이낸셜뉴

오늘 읽을만한 글

엔터

Find by string, selector, or XPath

Console What's New

Styles

element.  
style {  
}  
nmain.2...  
#gnb  
.list\_nav.t  
type\_fix  
.nav {  
color:  
#03c  
}  
nmain.2...  
#gnb  
.list\_nav  
.nav {  
text-  
decora  
:  
none;  
}  
nmain.2...  
a {  
color:  
inheri  
text-  
decora  
++  
none;  
}  
user ag...  
a:-webkit-  
any-link {  
color:  
-  
webkit  
link;  
cursor  
:  
pointe  
text-  
decora  
++  
underl

element.  
style {  
}  
nmain.2...  
#gnb  
.list\_nav.t  
type\_fix  
.nav {  
color:  
#03c  
}  
nmain.2...  
#gnb  
.list\_nav  
.nav {  
text-  
decora  
:  
none;  
}  
nmain.2...  
a {  
color:  
inheri  
text-  
decora  
++  
none;  
}  
user ag...  
a:-webkit-  
any-link {  
color:  
-  
webkit  
link;  
cursor  
:  
pointe  
text-  
decora  
++  
underl

body #PM\_ID\_ct div div div ul li a span.an\_txt

Filter :hov .cls +

element.style {  
}  
.area\_navigation .an\_txt {  
position: absolute;  
top: 0;  
right: 0;  
bottom: 0;  
left: 0;  
overflow: hidden;  
text-align: center;  
white-space: nowrap;  
}



# 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