# 크롤링 세미나

20180919

정윤식

- 목차

- 크롤링이란?
- 웹 기초
- HTML 기초
- 파이썬 환경설정
- 크롤링 기초
- 크롤링 실습
- 다루지 못한 이야기들

# 크롤러란 무엇인가

**웹 크롤러**(web crawler)는 조직적, 자동화된 방법으로 월드 와이드 웹(=인터넷)을 탐색하는 컴퓨터 프로그램이다.

- 위키피디아

인터넷에서 데이터를 가져오는 프로그램

# 크롤링이란 무엇인가

크롤링을 하는 이유

웹에서 필요한 데이터를 얻기 위해!

- 기업이나 정부기관에서 엑셀/API로 데이터를 제공하지 않는 경우
- 커뮤니티의 게시글 정보가 필요할 때 (네이버카페, 디시인사이드 등)

(페이스북이나 트위터는 자신의 데이터를 활용할 수 있도록 API를 제공함)

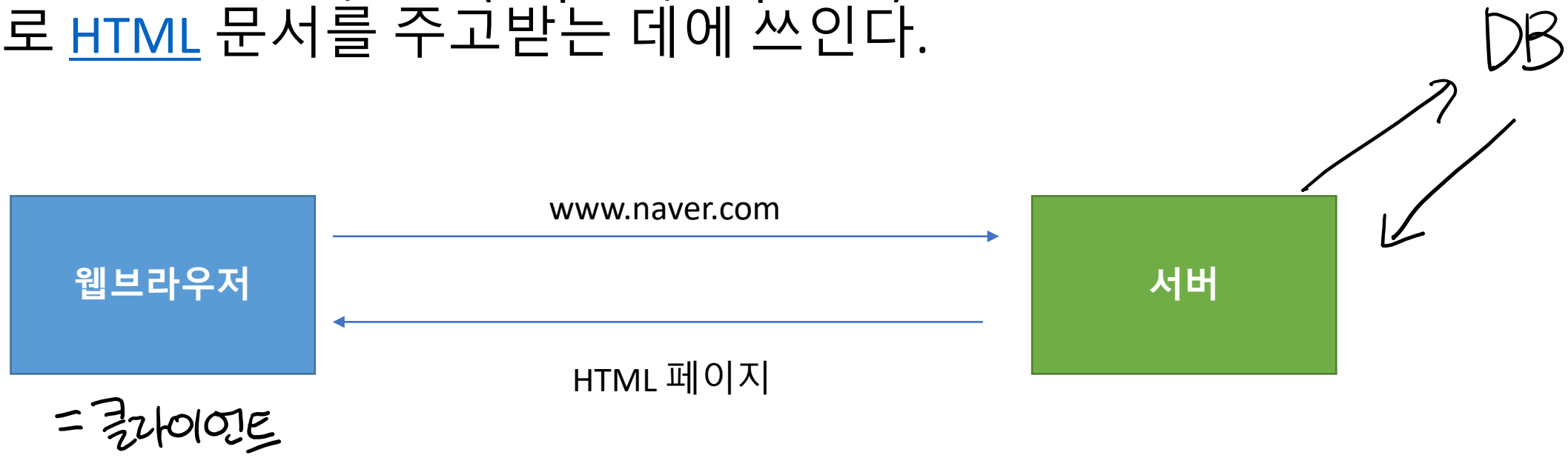
# 웹 기초

- HTTP란 무엇인가
- HTTP 요청 방식
- Header
- Status code

# 웹 기초

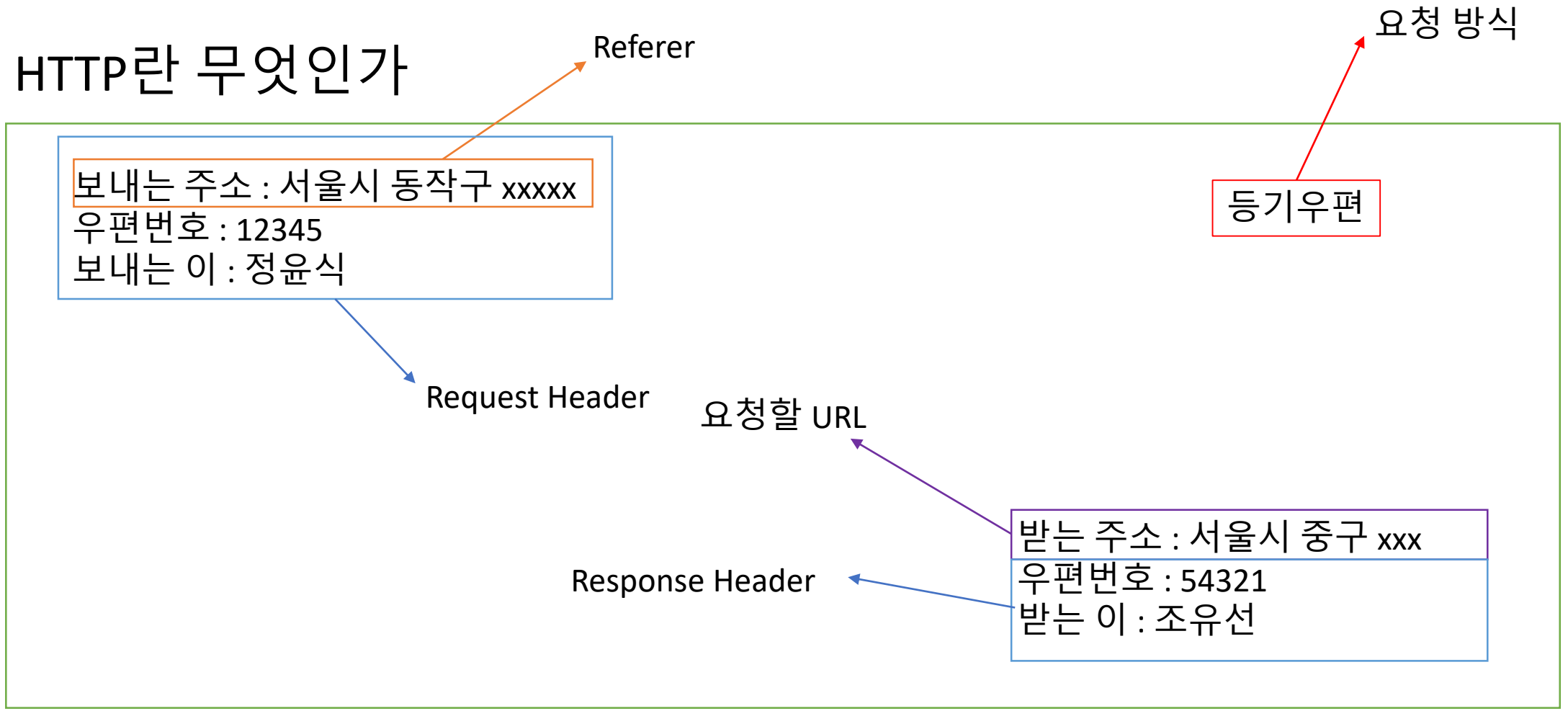
## HTTP란 무엇인가

HTTP(**H**yper**T**ext **T**ransfer **P**rotocol)는 클라이언트와 서버 사이에 이루어지는 요청/응답(request/response) 프로토콜이다.
주로 HTML 문서를 주고받는 데에 쓰인다.



# 웹 기초

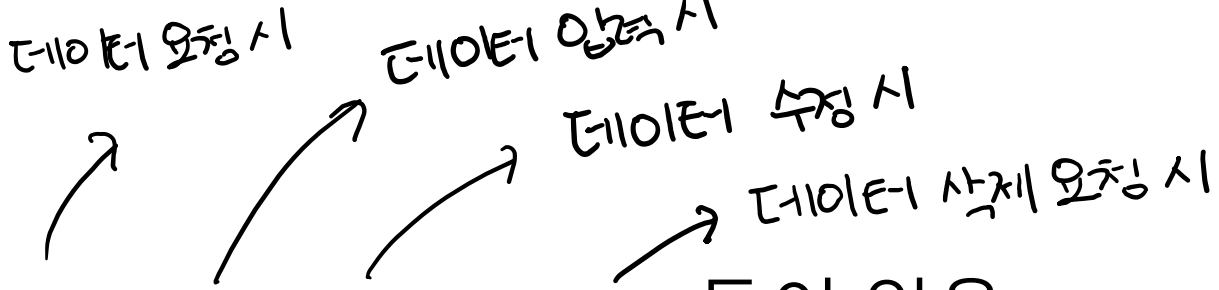
## HTTP란 무엇인가




*편지의 내용이 받는 주소에 적혀있으면 GET, 편지의 내용이 편지 봉투 안에 숨어있으면 POST

# 웹 기초

HTTP 요청 방식




요청 방식에는 GET, POST, PUT, DELETE 등이 있음

가장 많이 사용하는 요청 방식은 GET과 POST 방식

# 웹 기초

## GET :

**Request URL :**
https://search.naver.com/search.naver?where=nexearch&sm=top_hty&fbm=1&ie=utf8&query=dongguk

| # | 데이터 이름 | 데이터 내용 |
|---|---|---|
| 1 | where | nexearch |
| 2 | Sm | Top_hty |
| 3 | fbm | 1 |
| 4 | Ie | Utf8 |
| 5 | Query | dongguk |

# 웹 기초

- POST :

**Request URL:** https://nid.naver.com/nidlogin.login

**encpw:**8531ef0349084747a39412272b87dc7372126da3417daf4108aadff0f4f9d3a6b70e56962d0fafc4e787355b8159fb73d9c03dd8120166a5d4858a312b33d2ba308c28a376fcb4cbfc142e5b1655dd0397d0c440cf527473d6770e2c1d8a5782afa15182a4d746b8f9b51f2ee644842564e77fb7b9df135dbce0b490bf82e146

**encnm:** 100013584

**locale:** ko_KR

**url:** https://www.naver.com

| # | 데이터 이름 | 데이터 내용 |
|---|---|---|
| 1 | encpw | 8531ef03490… |
| 2 | encnm | 100013584 |
| 3 | locale | Ko_KR |
| 4 | url | https://www.naver.com |

# 웹 기초

GET/POST

+ 속도 빠름

+ 보안 느림

| | GET | POST |
|---|---|---|
| 정보 노출 | URL을 통해 노출 | HTTP 헤더 속으로 감춰짐 |
| 전달 가능한 정보의 양 | URL 주소의 한계 길이인 4096 바이트를 넘을 수 없음(이론적으로는 URL 주소의 길이 제한 없음) | 이론적으로는 무한대지나 현실적으로는 웹서버의 응답 지연 시간만큼 전송 가능 |
| 정보 전달 방식 | URL 뒤에 ?를 붙이거나 HTML 폼 형식에서 이용 가능 | 반드시 HTML 폼을 통해서만 사용할 수 있음 |
| 사용 범위 | 포탈 사이트의 검색어 전달, 게시판 페이지 번호 등 정보 위험도와 관계 없는 부분에서 많이 사용됨 | 회원 아이디, 비밀번호, 개인 정보 등 개인 정보 전송에 많이 사용됨 |

↳ URL에 게시판 정보, 게시글 정보가 들어있어 공유하기 용이함

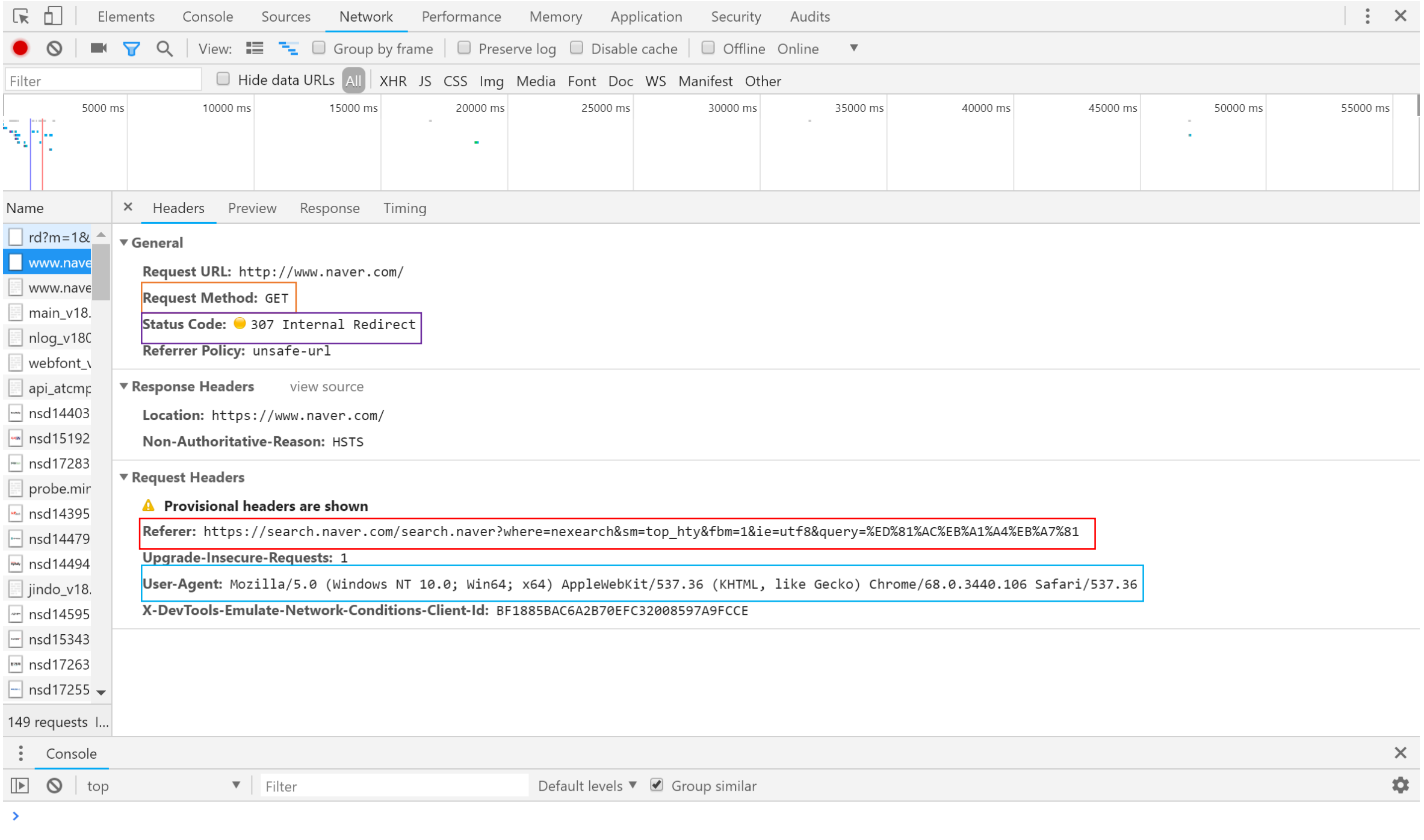
↳ URL에 정보가 드러나지 않음

# 웹 기초

Header

요청/응답 시 부수적으로 같이 보내는 정보

Referer : 해당 url 요청하기 직전에 방문한 URL (방문자 추적)
User-Agent : 요청을 보낸 사용자의 정보(운영체제, 브라우저 정보..)
Encoding : 어떤 형태로 인코딩이 되어있는지
....


Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
View:
Group by frame
Preserve log
Disable cache
Offline
Online
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
5000 ms
10000 ms
15000 ms
20000 ms
25000 ms
30000 ms
35000 ms
40000 ms
45000 ms
50000 ms
55000 ms
Name
rd?m=1&
www.nave
www.nave
main_v18.
nlog_v180
webfont_v
api_atcmp
nsd14403
nsd15192
nsd17283
probe.mir
nsd14395
nsd14479
nsd14494
jindo_v18.
nsd14595
nsd15343
nsd17263
nsd17255
149 requests
Headers
Preview
Response
Timing
General
Request URL: http://www.naver.com/
Request Method: GET
Status Code: 307 Internal Redirect
Referrer Policy: unsafe-url
Response Headers
view source
Location: https://www.naver.com/
Non-Authoritative-Reason: HSTS
Request Headers
Provisional headers are shown
Referer: https://search.naver.com/search.naver?where=nexearch&sm=top_hty&fbm=1&ie=utf8&query=%ED%81%AC%EB%A1%A4%EB%A7%81
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/68.0.3440.106 Safari/537.36
X-DevTools-Emulate-Network-Conditions-Client-Id: BF1885BAC6A2B70EFC32008597A9FCCE
Console
top
Filter
Default levels
Group similar

# 웹 기초

HTTP Status Code

웹으로 연결된 요청의 결과를 알려줌
(잘 완료되었는지 실패했는지, 그 경로에는 아무것도 없는지 등)

# 웹 기초

200 ~ 299 : 성공

400 ~ 499 : 클라이언트 오류

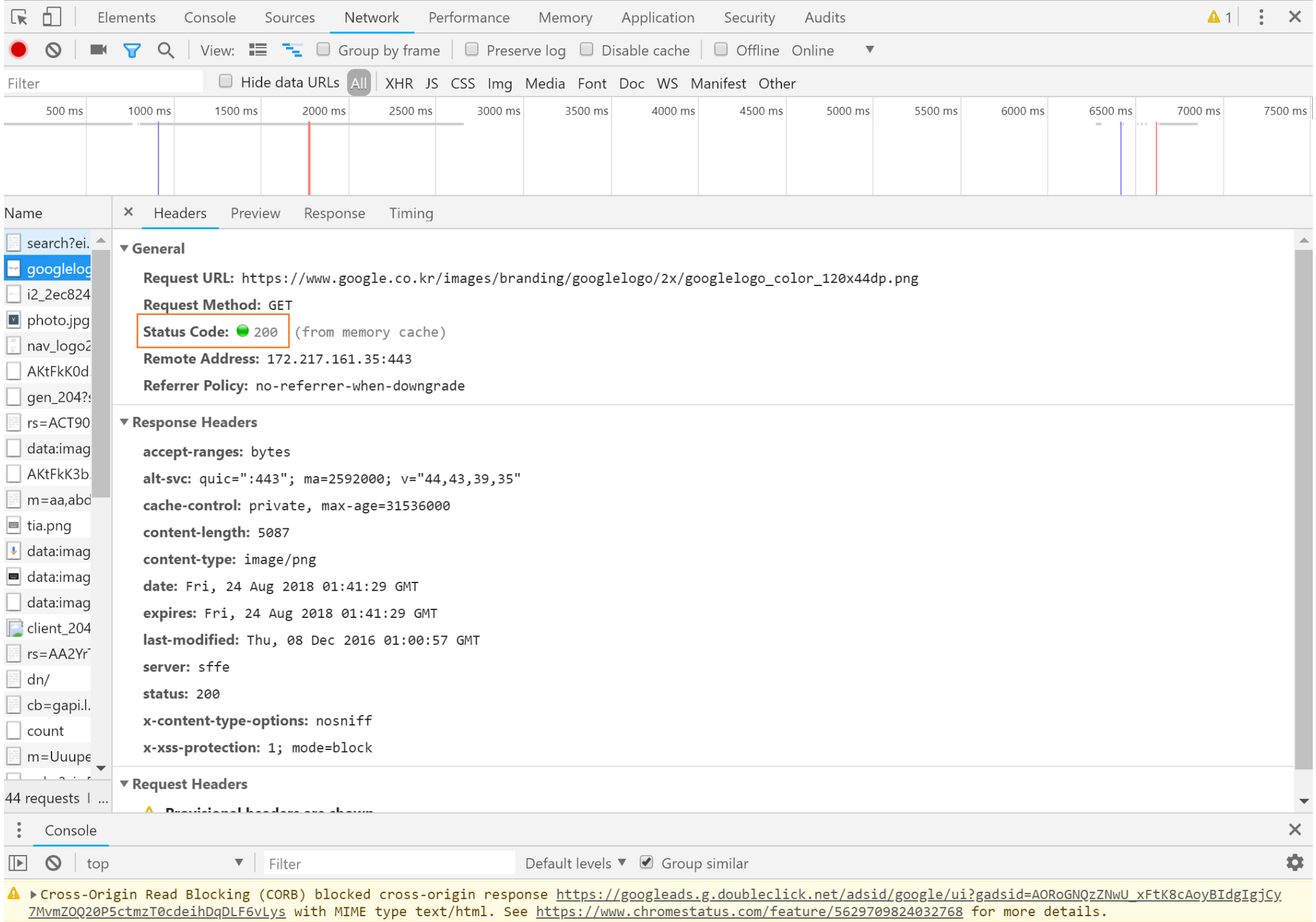
500 ~ 599 : 서버 오류

HTTP Status Code

200 : 성공

404 : Not found (요청한 url에 아무것도 없음)

500 : Internal Server Error(서버 문제로 접속할 수 없음)


Elements
Console
Sources
Network
Performance
Memory
Application
Security
Audits
View:
Group by frame
Preserve log
Disable cache
Offline
Online
Filter
Hide data URLs
All
XHR
JS
CSS
Img
Media
Font
Doc
WS
Manifest
Other
500 ms
1000 ms
1500 ms
2000 ms
2500 ms
3000 ms
3500 ms
4000 ms
4500 ms
5000 ms
5500 ms
6000 ms
6500 ms
7000 ms
7500 ms
Name
search?ei.
googlelog
i2_2ec824
photo.jpg
nav_logo2
AKtFkK0d
gen_204?s
rs=ACT90
data:imag
AKtFkK3b
m=aa,abd
tia.png
data:imag
data:imag
data:imag
client_204
rs=AA2Yr
dn/
cb=gapi.l.
count
m=Uuupe
44 requests
Headers
Preview
Response
Timing
General
Request URL: https://www.google.co.kr/images/branding/googlelogo/2x/googlelogo_color_120x44dp.png
Request Method: GET
Status Code: 200 (from memory cache)
Remote Address: 172.217.161.35:443
Referrer Policy: no-referrer-when-downgrade
Response Headers
accept-ranges: bytes
alt-svc: quic=":443"; ma=2592000; v="44,43,39,35"
cache-control: private, max-age=31536000
content-length: 5087
content-type: image/png
date: Fri, 24 Aug 2018 01:41:29 GMT
expires: Fri, 24 Aug 2018 01:41:29 GMT
last-modified: Thu, 08 Dec 2016 01:00:57 GMT
server: sffe
status: 200
x-content-type-options: nosniff
x-xss-protection: 1; mode=block
Request Headers
Console
top
Filter
Default levels
Group similar
Cross-Origin Read Blocking (CORB) blocked cross-origin response https://googleads.g.doubleclick.net/adsid/google/ui?gadsid=AORoGNQzZNwU_xFtK8cAoyBIdgIgjCy7MvmZOQ20P5ctmzT0cdeihDqDLF6vLys with MIME type text/html. See https://www.chromestatus.com/feature/5629709824032768 for more details.

# HTML 기초

- HTML이란?
- 태그
- 속성

# HTML 기초

HTML : 웹 페이지를 만드는 언어

매우 쉽고, 우리가 가장 많이 보는 언어

웹에서 데이터를 크롤링 한다 = HTML에서 데이터를 가져온다

# HTML 기초

태그

- HTML은 태그로 이루어져 있음 <html></html>
- 태그는 열림 태그와 닫힘 태그가 존재하고, 내용은 그 두 태그 안에 위치함
- 각각의 태그는 고유의 의미와 동작을 가짐(의미만 가지고 있는 경우도 있음)

# HTML 기초

## 태그

- ALL HTML documents must start with a document type declaration: `<!DOCTYPE html>`.
- The HTML document itself begins with `<html>` and ends with `</html>`.
- The visible part of the HTML document is between `<body>` and `</body>`.

# HTML 기초

자주 쓰이는 태그(1)

- <h1> (Heading) : 제목을 작성할 때 사용. h1~h6까지 있음
- <p> (paragraph) : 본문을 작성할 때 사용
- <a href="www.naver.com"> : 클릭했을 때 해당 페이지로 이동(하이퍼링크)
- <ul> (Unordered List) : 순서가 없는 목록
- <ol> (ordered list) : 순서가 있는 목록
- <li> (list) : 목록의 내용이 들어가는 부분

# HTML 기초

HTML 태그 실습 (1)

https://www.w3schools.com/html/

https://www.w3schools.com/html/html_links.asp

https://www.w3schools.com/html/html_lists.asp

# HTML 기초

자주 쓰이는 태그(2)

- <table> : 표를 만들때 사용하는 태그. 표에 관련된 태그는 <table>태그 안에 위치해야 함
- <th> (table head) : 테이블의 맨 위 제목에 사용되는 태그. 강조가 들어간다.
- <tr> (table row) : 테이블의 행을 지정하는 태그
- <td> : 테이블의 행에 들어가는 내용이 들어가는 태그

# HTML 기초

HTML 태그 실습 (2)

https://www.w3schools.com/html/html_tables.asp

# HTML 기초

속성

- All HTML elements can have **attributes**
- Attributes provide **additional information** about an element
- Attributes are always specified in **the start tag**
- Attributes usually come in name/value pairs like: **name="value“**

**Ex ) <a> 태그가 기능하기 위해서는 반드시 href=“xxx” 속성 필요**

# HTML 기초

자주 쓰이는 속성

href, class, name, id, value, …..

# HTML 기초

생활코딩 WEB 강의 : https://opentutorials.org/course/3084

# 파이썬 세팅

- 파이참 설치
- 프로젝트 생성
- virtualenv 생성 및 Project Interpreter 설정
- Github에서 예제 코드 가져오기

# 크롤링 이론

- 어떤 방식으로 크롤링은 동작하는가
- Requests
- Beautiful Soup
- 개발자도구

# 크롤링 이론

웹에서 데이터를 크롤링 한다 = HTML에서 데이터를 가져온다

-> 서버에 요청해서 HTML을 받아온다 : **Requests**

-> 필요한 정보가 어디있는지 알아본다 : **개발자 도구**

-> 받아온 HTML에서 필요한 정보를 찾아 뽑아낸다 : **Beautiful Soup**

# 크롤링 이론

```
…
<h3 class="blind">실시간 급상승 검색어</h3>
<div class="ah_roll_area PM_CL_realtimeKeyword_rolling">
<ul class="ah_l">
<li class="ah_item">
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r">1</span>
<span class="ah_k">구하라 남자친구 상처</span>
</a>
…
```

# 크롤링 이론

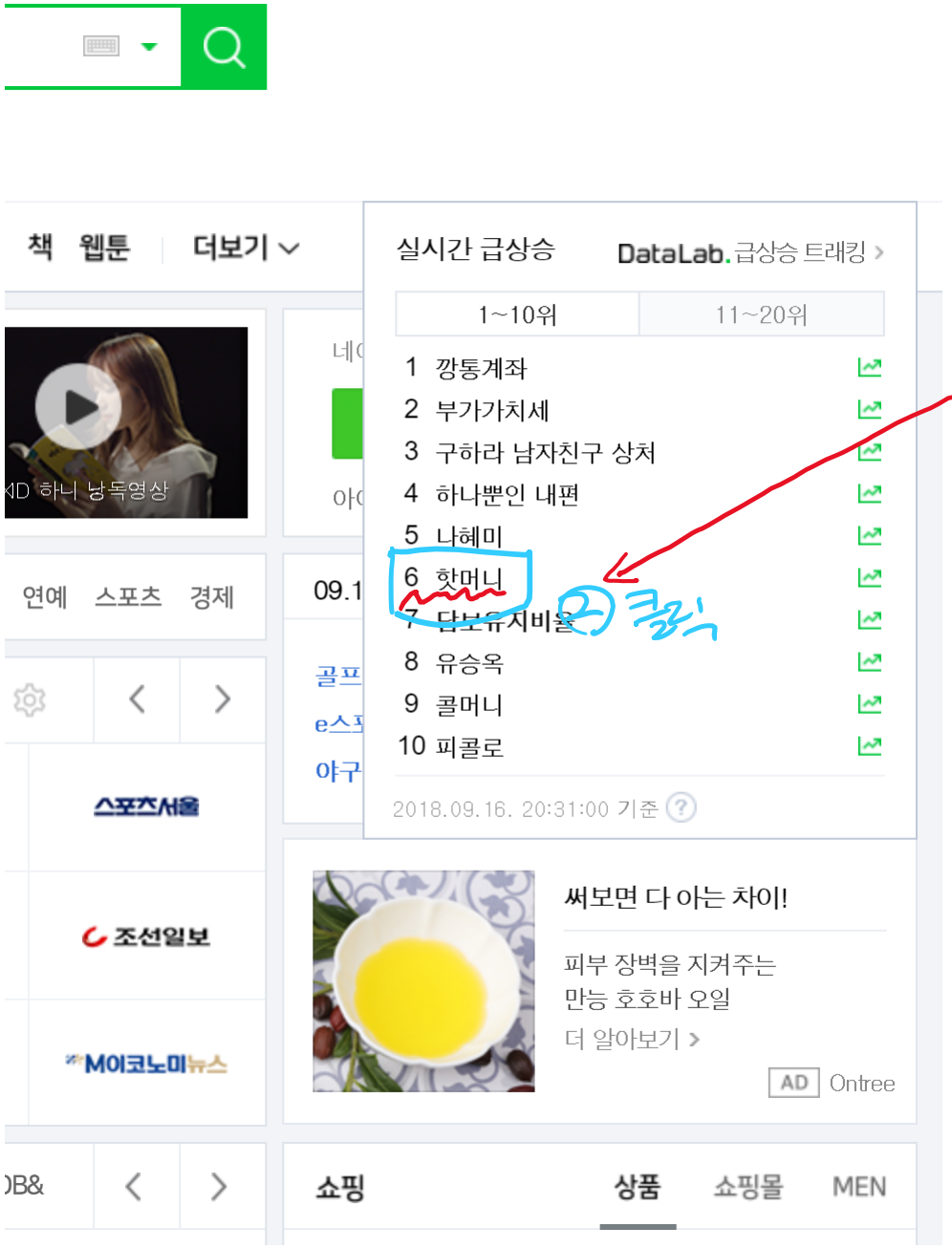
Requests 실습

GET 요청 보내기 : response = requests.get(‘www.naver.com’)
POST 요청 보내기 : response = requests.post(‘www.naver.com’)
Status Code 확인 : response.status_code
응답 텍스트(=HTML) : html = response.text

# 크롤링 이론

개발자 도구

필요한 정보가 어디에 어떤 HTML코드로 위치하고 있는지 찾을 때 사용한다.




```
Elements Console Sources Network Performance Memory Application Security Audi
<div class="section_navbar">
  <div class="area_navigation" role="navigation">…</div>
  <div class="area_hotkeyword PM_CL_realtimeKeyword_base" queryid="C1537097477531971428">
    <div class="ah_roll PM_CL_realtimeKeyword_rolling_base" aria-hidden="false">
      <h3 class="blind">실시간 급상승 검색어</h3>
      <div class="ah_roll_area PM_CL_realtimeKeyword_rolling" queryid="C153709747753080351">
        <ul class="ah_l" queryid="C153709767337143076">
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">
            <a href="#" class="ah_a" data-clk="lve.keyword">
              <span class="ah_r">6</span>
              <span class="ah_k">핫머니</span> == $0
            </a>
          </li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">
            <a href="#" class="ah_a" data-clk="lve.keyword">
              <span class="ah_r">16</span>
              <span class="ah_k">유이</span>
            </a>
          </li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
          <li class="ah_item">…</li>
html body #PM_ID_ct div div div div div ul.ah_l li.ah_item a.ah_a span.ah_k
Console
top Filter Default levels Group similar
>
```

# 크롤링 이론

Beautiful Soup 실습

Soup = BeautifulSoup(html, ‘html.parser’) : HTML을 자체적으로 분해하여 select, find와 같은 함수로 필요한 값을 가져올 수 있도록 하는 BeautifulSoup 객체를 만든다.

Soup.select(‘span.ah_k’) : CSS Selector(선택자)를 이용하여 CSS Selector에 해당하는 HTML을 가져온다.

Soup.find(‘’, attrs={‘class’:’ah_k’}) : 첫 번째 파라미터로는 태그명을, attrs라는 파라미터로는 태그의 속정을 지정하여 해당하는 html을 가져온다.

# 크롤링 이론

CSS Selector

HTML에 디자인을 입힐 때 원하는 HTML에 태그에만 디자인을 입힐 필요가 있음

Ex) 실시간 검색어 1위에게만 빨간색 강조를 주고 싶음

-> 실시간 검색어 1위만 선택하는 방법 필요

# 크롤링 이론

CSS Selector

CSS 선택자 = 태그 + 속성 기호 + 속성값

태그 : 태그 명 (a, h1, p, …)
Class 속성 기호 : .
Id 속성 기호 : #

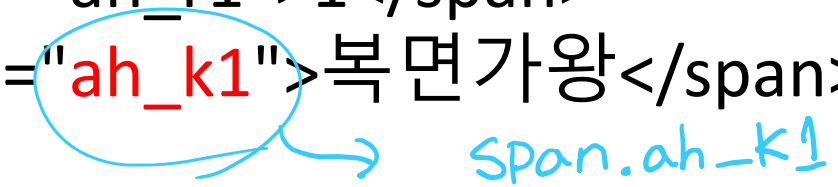
Ex) <h1 class=“title”>제목</h1> 의 CSS 선택자는 h1.title
Ex2) 내용 의 CSS 선택자는 p#contents

# 크롤링 이론

CSS Selector

…

```
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r1">1</span>
<span class="ah_k1">복면가왕</span>
</a>
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r2">2</span>
<span class="ah_k2">구하라</span>
</a>
```





…

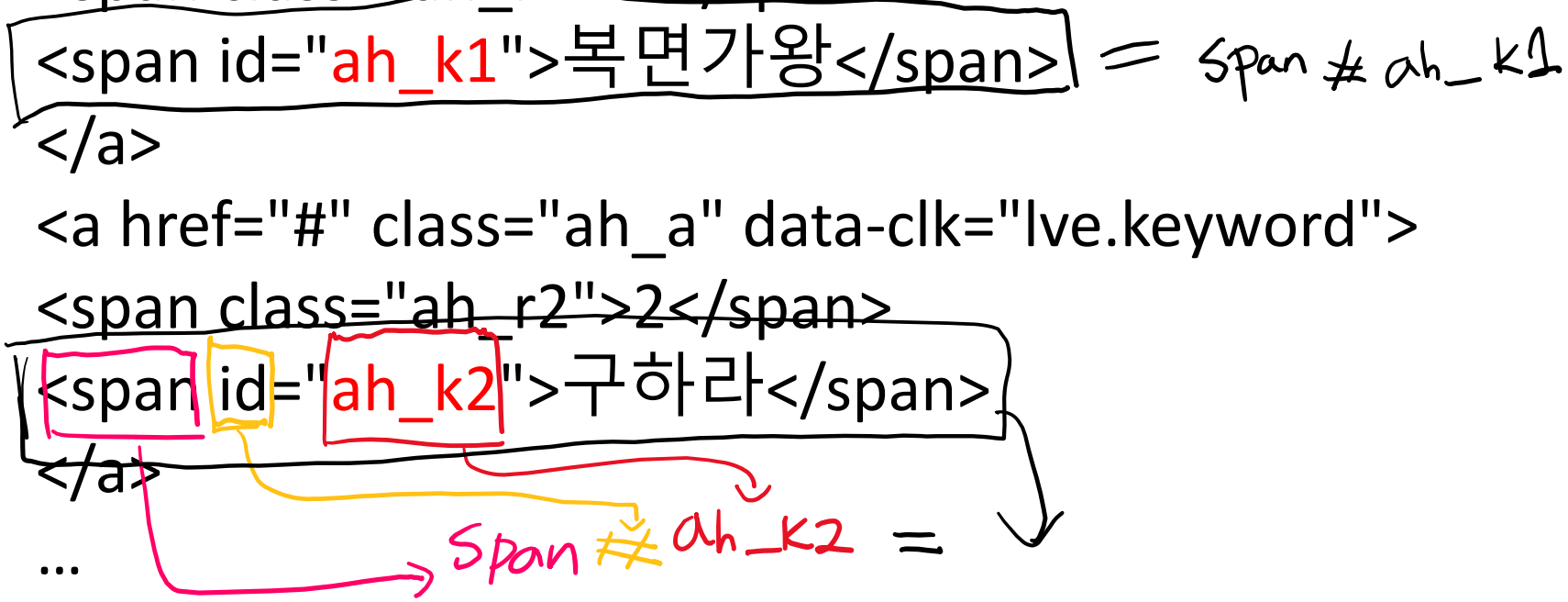
# 크롤링 이론

CSS Selector

…

```
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r1">1</span>
<span id="ah_k1">복면가왕</span>
</a>
<a href="#" class="ah_a" data-clk="lve.keyword">
<span class="ah_r2">2</span>
<span id="ah_k2">구하라</span>
</a>
```

…



# 크롤링 실습

크롤링 실습 1 : 네이버 실시간 검색어 크롤링하기

크롤링 실습 2 : DC인사이드 식물갤러리 게시글 크롤링 하기

# 다루지 못한 이야기

Selenium

API

DataBase(pymysql)