

# Python 크롤링 / 웹서버 개발

## 2. 기본기 다지기

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## Crawling

## Crawling

자동화된 방법으로 웹에 제공된 정보를 수집하여  
사용하기 쉬운 형태로 가공하는 작업

## Crawling

1. 자동화된 방법으로
2. 웹에 제공된 정보를 수집하여
3. 사용하기 쉬운 형태로 가공하는 작업

# 기본기 다지기

Python 크롤링 / 웹서버 개발

```
{
  "menu": [
    {
      "date": "5",
      "breakfast": ["chap쌀밥", "honghamiyeokguk5.6.", "dalgabi5.6.13.", "kimgugui13.", "baechuKimchi9.13.", "bangultomato12."],
      "lunch": ["hukmiBap", "kongnamulguk5.13.", "dwajedungbeekimchijjim9.10.13.", "dotorimokmuchim5.6.13.", "songgakKimchi9.13.", "cheongpodo"],
      "dinner": ["chap쌀밥", "ojingeojjambongguk5.6.", "baechugeoljeolli13.", "podosjuus5.13.", "manduoggonomiayagie1.5.6.10.12.13."]
    },
    ...
  ]
}
```

JSON

# 기본기 다지기

Python 크롤링 / 웹서버 개발

**Python**  
**URL**  
**HTTP**  
**JSON**

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## URL

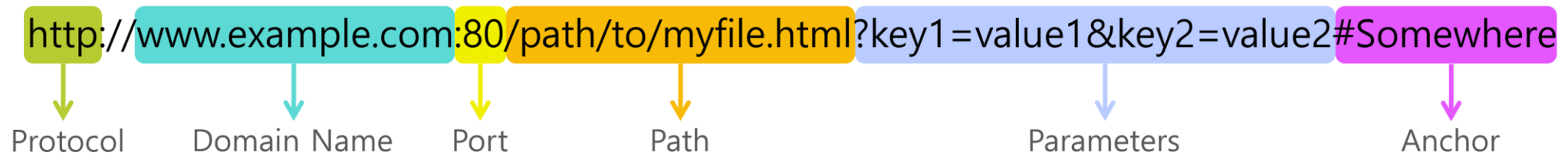
## URL

Uniform Resource Locator



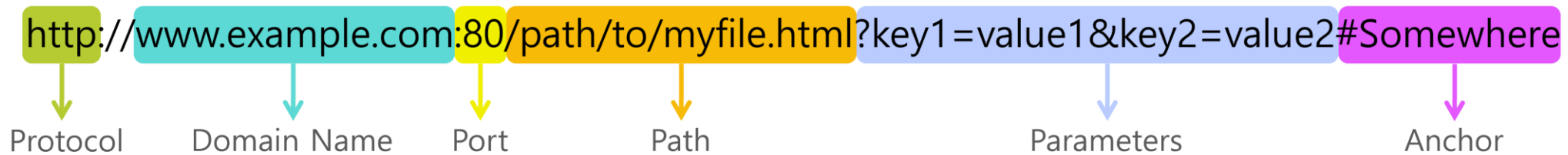
# 기본기 다지기

Python 크롤링 / 웹서버 개발



# 기본기 다지기

Python 크롤링 / 웹서버 개발



- Protocol - 브라우저가 어떤 프로토콜을 사용하는가
- Domain Name - IP 주소를 사용자가 쉽게 찾을 수 있도록 만든 서비스 (도메인 대신 IP가 올 수 있음)
- Port - 포트 번호 (HTTP = 80, HTTPS = 443)
- Parameters - 요청에 필요한 각종 추가 정보들

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## URL

최대길이?

## URL

최대길이 = 약 2,000자

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## URL

한글?

## URL

인터넷에서 통신이 가능한 문자 = ASCII  
ASCII 문자가 아닌 문자는 ASCII 변환 필요

## URL

가나다 = %EA%B0%80%EB%82%98%EB%8B%A4

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## HTTP



## HTTP

Hyper Text Transfer Protocol

## HTTP

Hyper Text Transfer Protocol

인터넷 상에서 데이터를 주고 받을 수 있는 프로토콜(통신규약)

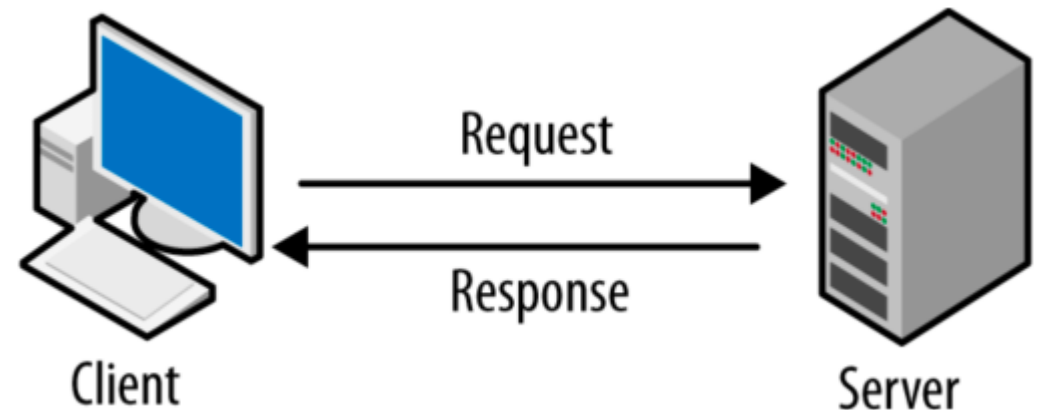
# 기본기 다지기

Python 크롤링 / 웹서버 개발

## HTTP 특징

### 1. 클라이언트 - 서버 구조

클라이언트는 서버에 요청, 서버는 요청에 대한 응답을 반환



# 기본기 다지기

Python 크롤링 / 웹서버 개발

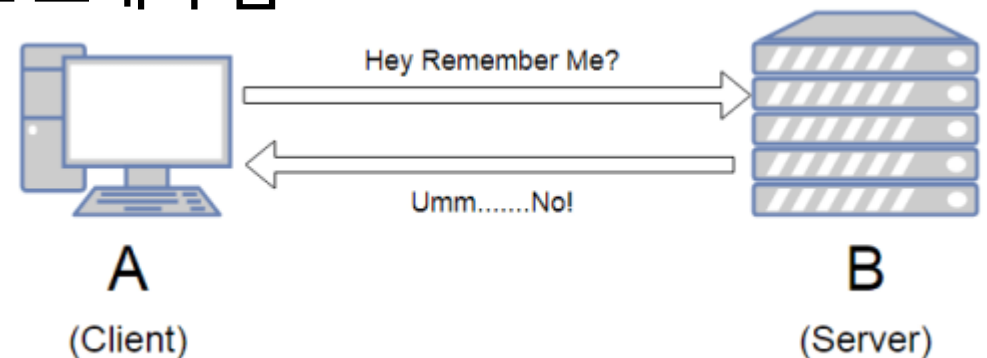
## HTTP 특징

### 2. 무상태 프로토콜 (Stateless)

서버는 클라이언트의 상태를 보존하지 않음

장점: 서버 확장성이 높아짐

단점: 클라이언트는 자신의 상태를 추가적으로 보내야 됨



## HTTP 특징

### 3. 비 연결성 (Connectionless)

서버는 클라이언트의 요청에 대한 처리(응답)를 완료하면 연결을 종료

장점: 리소스 소비를 줄임

단점: 매번 새로운 연결을 필요하기에 요청 시마다 연결-해제 오버헤드 발생

## HTTP Method

클라이언트가 서버에 요청 시,  
해당 요청이 어떤 목적인지를 나타내기 위해 사용

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## HTTP Method

- GET
- POST
- PUT
- PATCH
- DELETE

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## HTTP Method

- GET - 리소스 조회
- POST - 리소스 생성
- PUT - 리소스 수정 (전체 삽입)
- PATCH - 리소스 수정 (일부 삽입)
- DELETE - 리소스 삭제



# 기본기 다지기

Python 크롤링 / 웹서버 개발

## GET

- 서버에서 조회 역할만을 함  
(서버 상태 변경 없음)
- 요청 내용을 URL에 Query로 전달  
(Body 사용 불가, 글자 수 제한)
- URL에 요청 내용이 담기기에  
프라이빗한 내용은 GET으로 요청해서는 안됨

## POST

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## GET

- 서버에서 조회 역할만을 함  
(서버 상태 변경 없음)
- 요청 내용을 URL에 Query로 전달  
(Body 사용 불가, 글자 수 제한)
- URL에 요청 내용이 담기기에  
프라이빗한 내용은 GET으로 요청해서는 안됨

## POST

- 서버에서 생성 역할을 함  
(보통은 서버의 상태 변경을 위해 요청)
- 요청 내용을 Body로 전달  
(글자 수 제한 없음)
- Body에 데이터가 담기기에  
보안 측면에서 GET보다 우수함

## HTTP Status Code

요청에 대한 응답이 어떤 상태인지 유추할 수 있도록 숫자 코드를 반환

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## HTTP Status Code

- 1xx
- 2xx
- 3xx
- 4xx
- 5xx

## HTTP Status Code

- 1xx (정보) - 요청을 받았으며, 프로세스를 계속한다
- 2xx (성공) - 요청을 성공적으로 받았으며, 인식하였고, 수용하였다
- 3xx (리다이렉션) - 요청 완료를 위해 추가 작업 조치가 필요하다
- 4xx (클라이언트 오류) - 요청의 문법이 잘못되었거나 요청을 처리할 수 없다
- 5xx (서버 오류) - 서버가 명백히 유효한 요청에 대해 충족을 실패했다

# 기본기 다지기

Python 크롤링 / 웹서버 개발

<https://developer.mozilla.org/ko/docs/Web/HTTP/Status>

## HTTP Header

클라이언트와 서버가 요청, 응답에서 추가적인 정보 전달을 위한  
목적으로 주고받는 내용

## HTTP Message

서버와 클라이언트 간에 데이터가 교환되는 방식



## HTTP Message

각각 Head와 Body를 가지고 있음

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## Request

- Head (시작 줄 + Header)
- Body (요청을 위한 내용)

## Response

- Head (상태 줄 + Header)
- Body (응답을 위한 내용)

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## Request

- Head (시작 줄 + Header)
- Body (요청을 위한 내용)

시작 줄 = [메서드] + [경로] + [HTTP 버전]  
EX) GET /background HTTP/1.1

## Response

- Head (상태 줄 + Header)
- Body (응답을 위한 내용)

상태 줄 = [HTTP 버전] + [상태코드] + [간략한 상태내용]  
EX) HTTP/1.1 404 Not Found.

# 기본기 다지기

Python 크롤링 / 웹서버 개발

```
× 헤더 미리보기 응답 시작점 타이밍 쿠키
▼ 일반
요청 URL: https://www.naver.com/
요청 메서드: GET
상태 코드: 200
원격 주소: 23.196.172.185:443
리퍼리 정책: strict-origin-when-cross-origin
▼ 응답 헤더
cache-control: no-cache, no-store, must-revalidate
content-encoding: gzip
content-length: 32863
content-type: text/html; charset=UTF-8
date: Thu, 01 Sep 2022 05:30:58 GMT
p3p: CP="CAO DSP CURa ADMa TAIA PSAa OUR LAW STP PHY ONL UNI PUR FIN COM NAV INT DEM STA PRE"
pragma: no-cache
referrer-policy: unsafe-url
server: NWS
strict-transport-security: max-age=63072000; includeSubdomains
vary: Accept-Encoding
x-frame-options: DENY
x-xss-protection: 1; mode=block
▼ 요청 헤더
:authority: www.naver.com
:method: GET
:path: /
:scheme: https
accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
accept-encoding: gzip, deflate, br
```

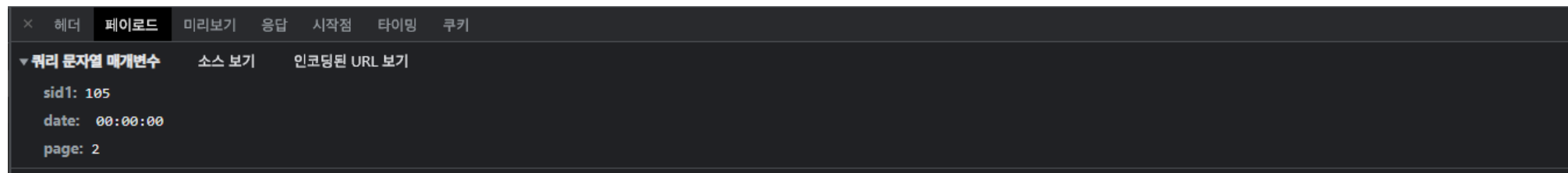
# 기본기 다지기

Python 크롤링 / 웹서버 개발

```
× 헤더 미리보기 응답 시작점 타이밍 쿠키
1
2 <!doctype html> <html lang="ko" data-dark="true"> <head> <meta charset="utf-8"> <title>NAVER</title> <meta http-equiv="X-UA-Compatible" content="IE=edge
3 useId: null, daInfo: {"BEAUTY":{"menu":"BEAUTY","childMenu":"","adType":"singleDom","multiDomAdUrl":"","multiDomUnit":"","infoList":[{"adposId":"1000163","singleDomAdUrl":"ht
4 svt: 20220901154303,
5 }}; </script> <script> window.nmain.newsstand = {
6 rcode: '09680103',
7 newsCastSubsInfo: '',
8 newsStandSubsInfo: ''
9 };
10 window.etc = { };
11 window.svr = "<!--cvweb01-->"; </script> <script src="https://ssl.pstatic.net/tveta/libs/assets/js/pc/main/min/pc.veta.core.min.js" defer="defer"></script> <script src="https
12 <style>._1syGnXOL{padding-right:18px;font-size:14px;line-height:0;letter-spacing:-.25px;color:#000}._1syGnXOL span,._1syGnXOL strong{line-height:49px}._1syGnXOL:before{display:
13 <div
14 id="NM_TOP_BANNER"
15 data-clk-prefix="top"
16 class="_1hiMwemA"
17 style="background-color: #fff7e1"
18 >
19 <div class="tY_u8r23">
20 <a
21 class="_3h-N8T9V"
22 href="https://whale.naver.com/banner/details/security/?=main&wpid=RydDy7"
23 data-clk="dropbanner1b"
24 ></a>
25 ><span
26 class="_1syGnXOL _3VkgqBX8"
27 data-clk="dropbanner1b"
28 style="padding-right: 20px; font-size: 17px; color: black"
29 ><span>매일 쓰는 브라우저 보안이 걱정된다면, </span>
30 ><strong
31 >안전하고 빠른 최신 브라우저 웨일로 업데이트 하세요.</strong>
32 ></span>
33 ><a
34 href="https://installer-whale.pstatic.net/downloads/banner/RydDy7/WhaleSetup.exe"
35 class="_2aeXMlrB BMgpjddw"
36 id="NM_whale_download_btn"
37 data-clk="dropdownload1b"
38 ><span style="background-color: #0436c7">다운로드</span></a>
39 ><button
40 type="button"
```

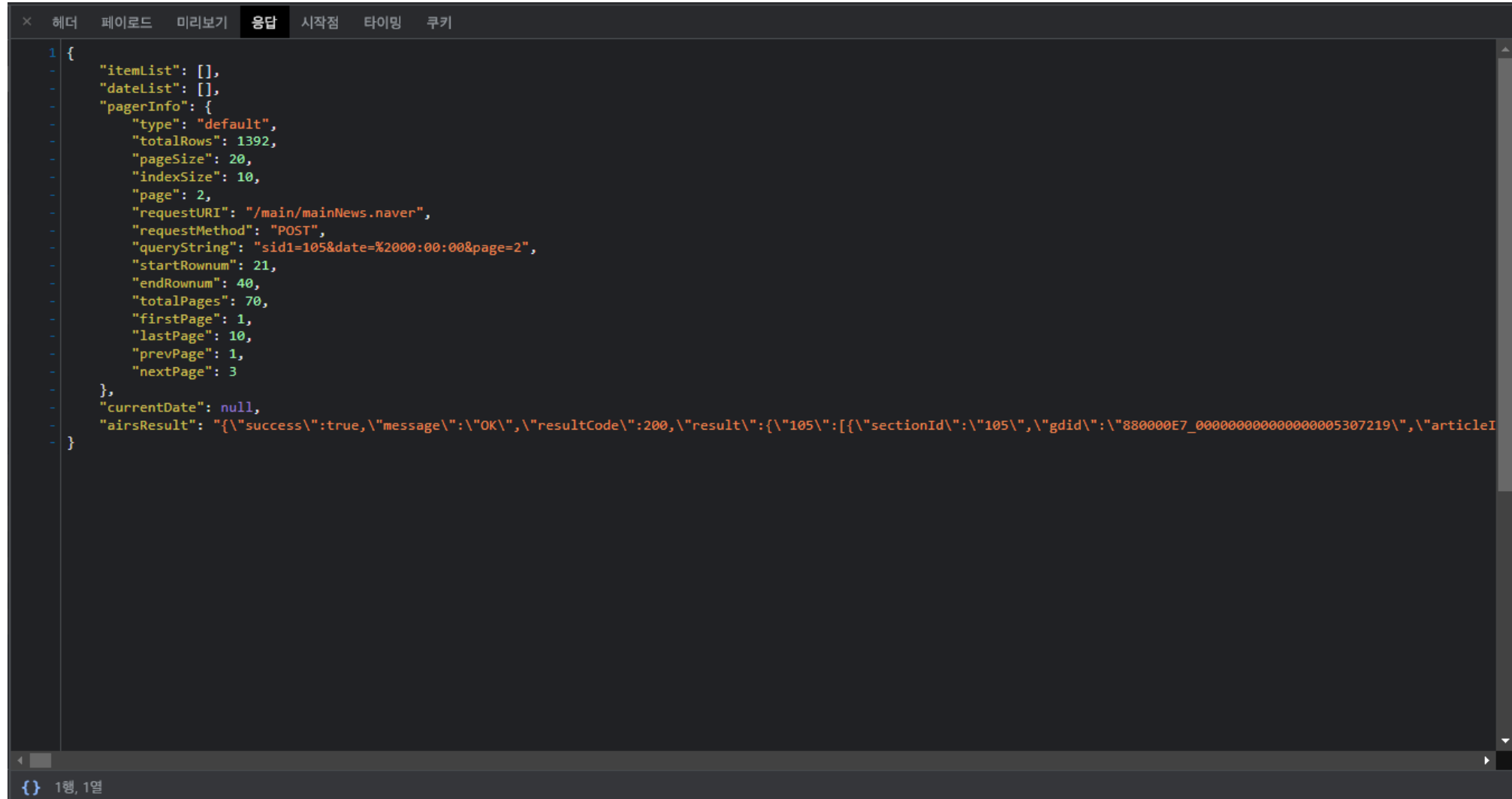
# 기본기 다지기

Python 크롤링 / 웹서버 개발



# 기본기 다지기

Python 크롤링 / 웹서버 개발



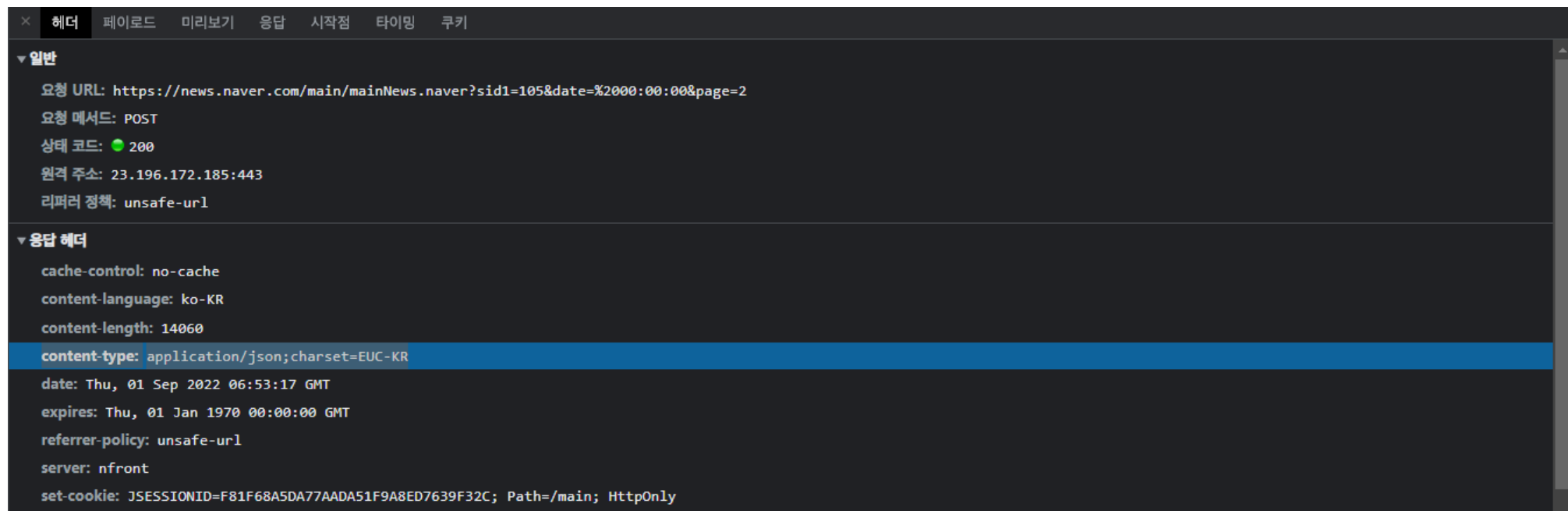
The image shows a web browser's developer console with the '응답' (Response) tab selected. The console displays a JSON object representing a page of search results. The JSON structure includes fields for item and date lists, pager information (type, total rows, page size, index size, current page), request details (URI, method, query string), and pagination details (start/end row numbers, total pages, first/last/prev/next page numbers). The 'currentDate' is null, and the 'airsResult' contains a success message and a result object with a section ID and GUID.

```
1 {  
  - "itemList": [],  
  - "dateList": [],  
  - "pagerInfo": {  
    - "type": "default",  
    - "totalRows": 1392,  
    - "pageSize": 20,  
    - "indexSize": 10,  
    - "page": 2,  
    - "requestURI": "/main/mainNews.naver",  
    - "requestMethod": "POST",  
    - "queryString": "sid1=105&date=%2000:00:00&page=2",  
    - "startRownum": 21,  
    - "endRownum": 40,  
    - "totalPages": 70,  
    - "firstPage": 1,  
    - "lastPage": 10,  
    - "prevPage": 1,  
    - "nextPage": 3  
  },  
  - "currentDate": null,  
  - "airsResult": "{\\"success\\":true,\\"message\\":\\"OK\\",\\"resultCode\\":200,\\"result\\":{\\"105\\":[{\\"sectionId\\":\\"105\\",\\"gdid\\":\\"88000E7_000000000000000005307219\\",\\"articleI  
- }  
}
```

{} 1행, 1열

# 기본기 다지기

Python 크롤링 / 웹서버 개발





# 기본기 다지기

Python 크롤링 / 웹서버 개발

## JSON

## JSON

JavaScript Object Notation

## JSON

Javascript 객체 문법을 따르는 문자 기반의 데이터 포맷

## JSON

현재 Javascript에 국한되지 않고  
대부분의 프로그래밍 환경에서 JSON을 다룰 수 있도록 제공됨

# 기본기 다지기

Python 크롤링 / 웹서버 개발

```
1  {
2    "name": "가나다",
3    "bool": true,
4    "num": 123,
5    "null": null,
6    "memos": ["메모1", "메모2", "메모3"],
7    "posts": [
8      {
9        "title": "제목",
10       "content": "내용"
11     },
12     {
13       "title": "제목",
14       "content": "내용"
15     },
16     {
17       "title": "제목",
18       "content": "내용"
19     }
20   ]
21 }
```

# 기본기 다지기

Python 크롤링 / 웹서버 개발

Key: Value

```
1  {
2  "name": "가나다"
3  "bool": true,
4  "num": 123,
5  "null": null,
6  "memos": ["메모1", "메모2", "메모3"],
7  "posts": [
8    {
9      "title": "제목",
10     "content": "내용"
11   },
12   {
13     "title": "제목",
14     "content": "내용"
15   },
16   {
17     "title": "제목",
18     "content": "내용"
19   }
20 ]
21 }
```

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## Data Type

- 문자열 (String)
- 숫자 (Number)
- 참/거짓 (Boolean)
- 널 (Null)
- 객체 (Object)
- 배열 (Array)

```
1  {
2    "name": "가나다",
3    "bool": true,
4    "num": 123,
5    "null": null,
6    "memos": ["메모1", "메모2", "메모3"],
7    "posts": [
8      {
9        "title": "제목",
10       "content": "내용"
11     },
12     {
13       "title": "제목",
14       "content": "내용"
15     },
16     {
17       "title": "제목",
18       "content": "내용"
19     }
20   ]
21 }
```

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## Object

{ }로 묶여있는 부분

```
1 {  
2     "name": "가나다",  
3     "bool": true,  
4     "num": 123,  
5     "null": null,  
6     "memos": ["메모1", "메모2", "메모3"],  
7     "posts": [  
8         {  
9             "title": "제목",  
10            "content": "내용"  
11        },  
12        {  
13            "title": "제목",  
14            "content": "내용"  
15        },  
16        {  
17            "title": "제목",  
18            "content": "내용"  
19        }  
20     ]  
21 }
```



# 기본기 다지기

Python 크롤링 / 웹서버 개발

## Object

{ }로 묶여있는 부분

```
1  {
2    "name": "가나다",
3    "bool": true,
4    "num": 123,
5    "null": null,
6    "memos": ["메모1", "메모2", "메모3"],
7    "posts": [
8      {
9        "title": "제목",
10       "content": "내용"
11     },
12     {
13       "title": "제목",
14       "content": "내용"
15     },
16     {
17       "title": "제목",
18       "content": "내용"
19     }
20   ]
21 }
```

# 기본기 다지기

Python 크롤링 / 웹서버 개발

## Array

[ ]로 묶여있는 부분

```
1  {
2    "name": "가나다",
3    "bool": true,
4    "num": 123,
5    "null": null,
6    "memos": ["메모1", "메모2", "메모3"],
7    "posts": [
8      {
9        "title": "제목",
10       "content": "내용"
11      },
12      {
13        "title": "제목",
14        "content": "내용"
15      },
16      {
17        "title": "제목",
18        "content": "내용"
19      }
20    ]
21  }
```