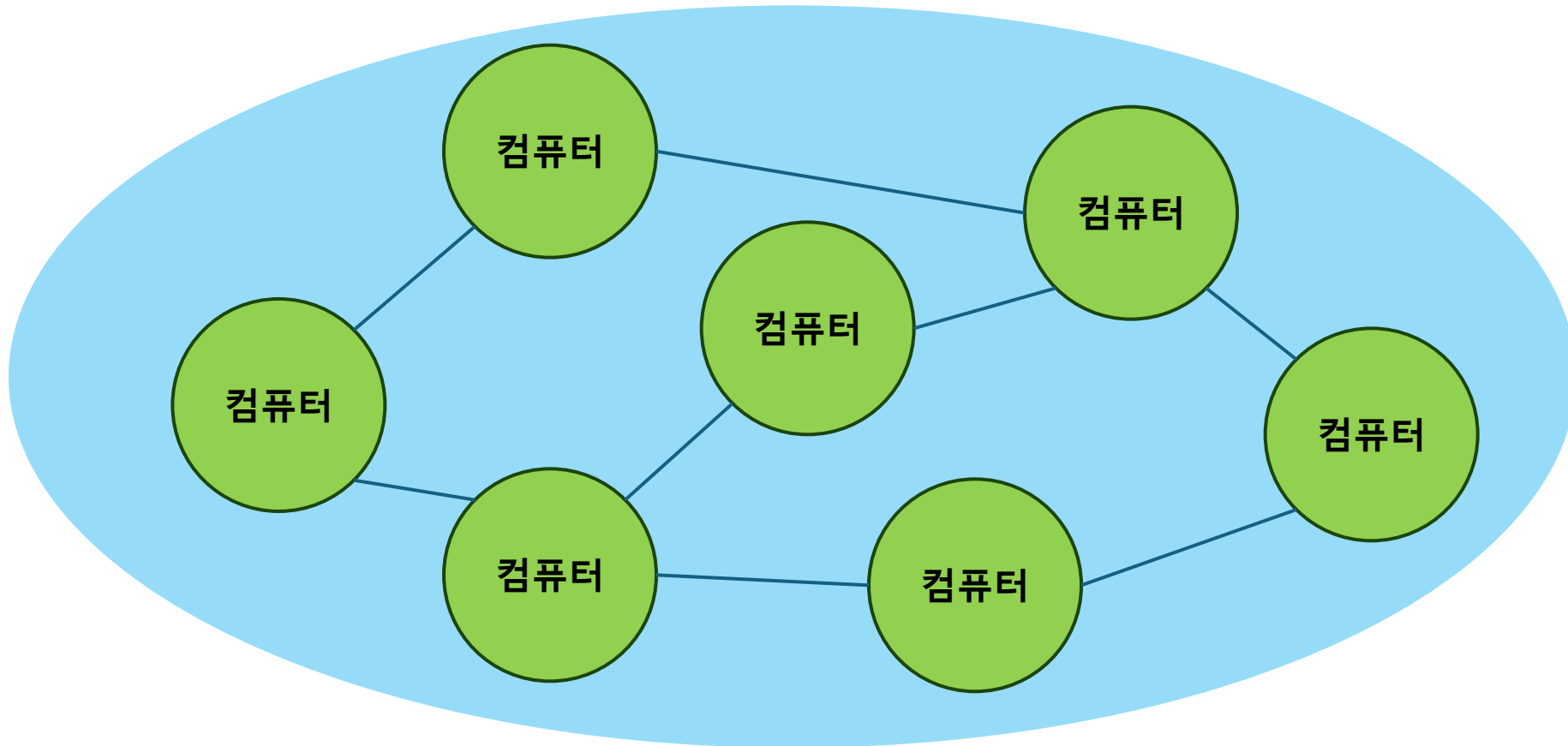


초급 백엔드 스터디

1주차 - 웹과 백엔드

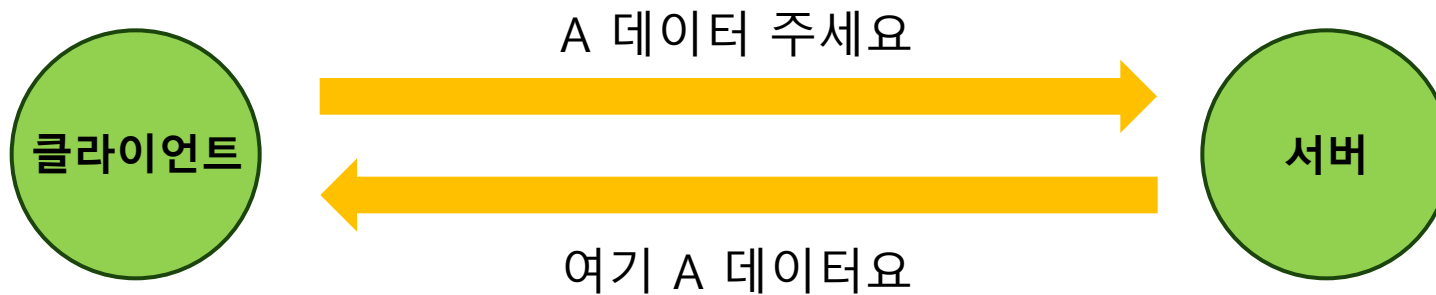
네트워크

- 여러 컴퓨터가 서로 연결되어 정보를 공유하는 공간



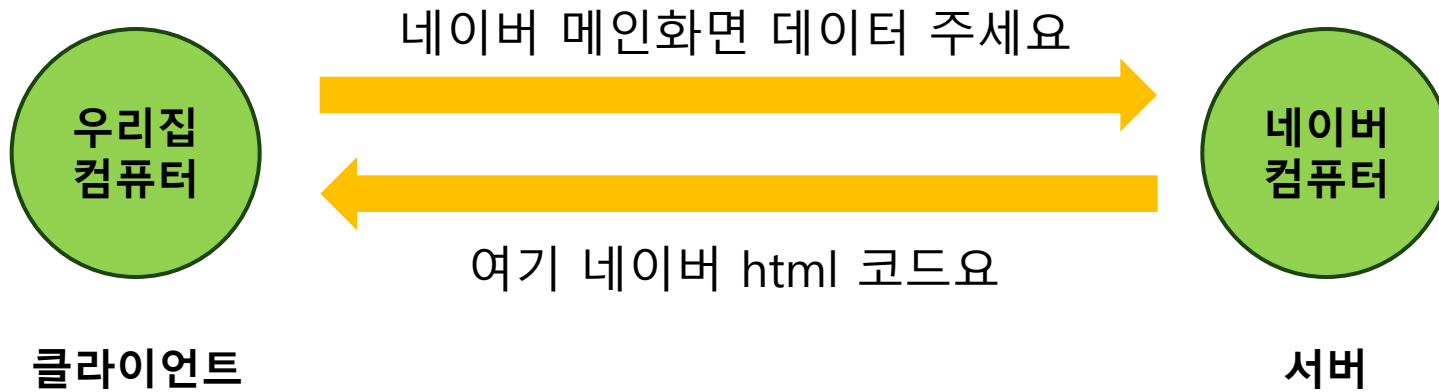
웹

- 웹에서 컴퓨터가 서로 정보를 주고 받는 일반적인 형태는 **클라이언트-서버** 패러다임
- **클라이언트** : 데이터의 **생성/조회/수정/삭제** 요청을 전송
- **서버** : 요청대로 동작을 수행하고 **응답**을 전송



예시

- 브라우저로 네이버에 접속하는 것

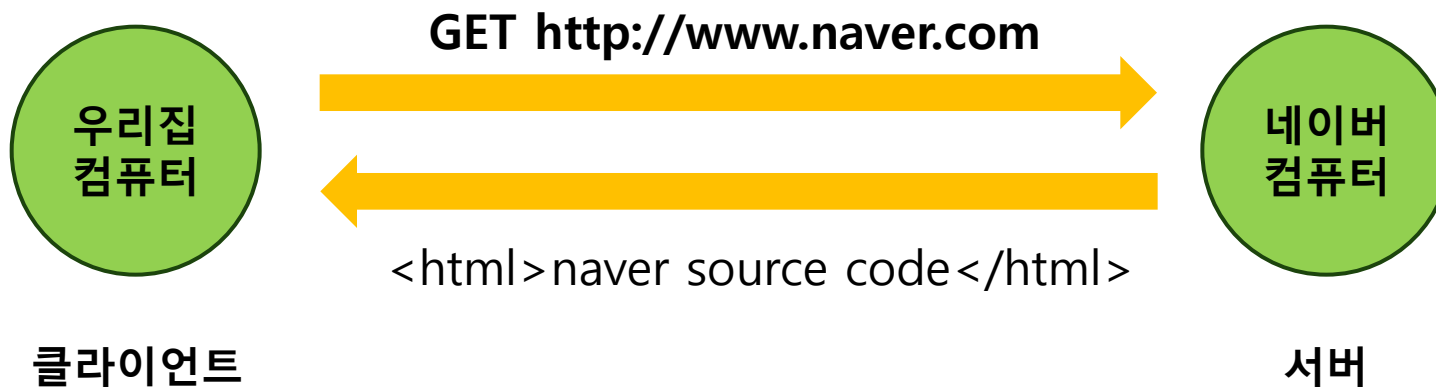


프로토콜과 HTTP

- 컴퓨터는 정해진 동작만 수행할 수 있는 기계
- 컴퓨터가 서로 정보를 공유할 때, 정해진 **규칙이 필요하다**.

프로토콜과 HTTP

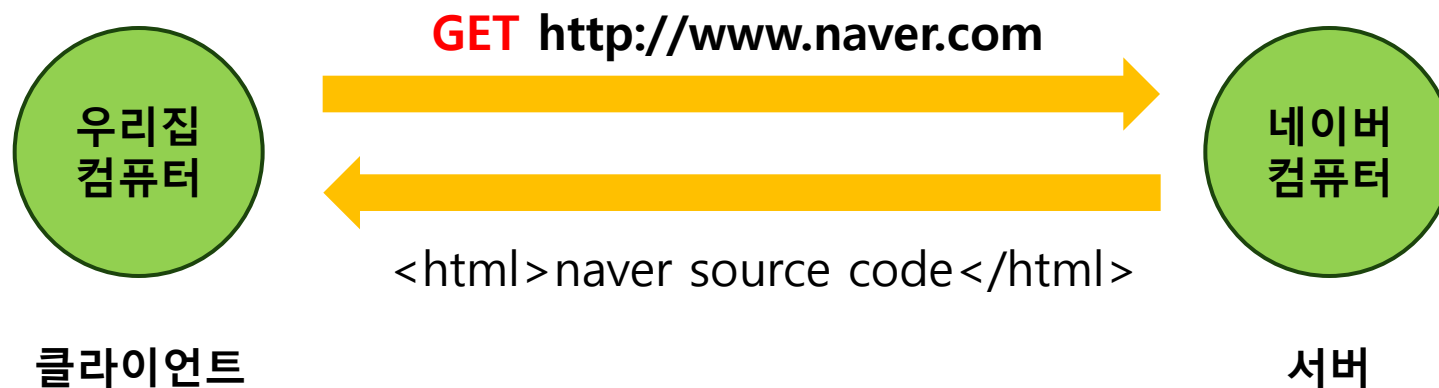
- **프로토콜** : 네트워크 안에서 요청과 응답을 보내는 **규칙**
- 웹에서는 **HTTP** 라는 프로토콜(규칙)을 사용한다.



프로토콜과 HTTP

HTTP를 통해 요청을 보낼 땐 **HTTP Method, URL**이 필요하다.

- **HTTP Method** : 데이터를 다루는 방법
- **URL**(Uniform Resource Location) : 다룰 데이터의 위치



프로토콜과 HTTP

자주 사용하는 HTTP Method

- **GET** : 데이터를 가져온다. (조회)
- **POST** : 데이터를 게시한다. (생성)
- **PUT** : 데이터를 교체한다. (수정)
- **PATCH** : 데이터를 수정한다. (수정)
- **DELETE** : 데이터를 삭제한다. (삭제)

프로토콜과 HTTP

- URL 구조

http://www.example.com/user/1/nickname

프로토콜 (scheme)

서버 주소 (domain)

서버 내 데이터 위치 (path)

프로토콜과 HTTP

- Path Parameter (URL의 일반화된 표현 방법)

`http://www.example.com/user/{user_id}/nickname`

프로토콜 (scheme)

서버 주소 (domain)

서버 내 데이터 위치 (path)

프로토콜과 HTTP

- Query String

`.com/post/search?page=1&keyword=hello`

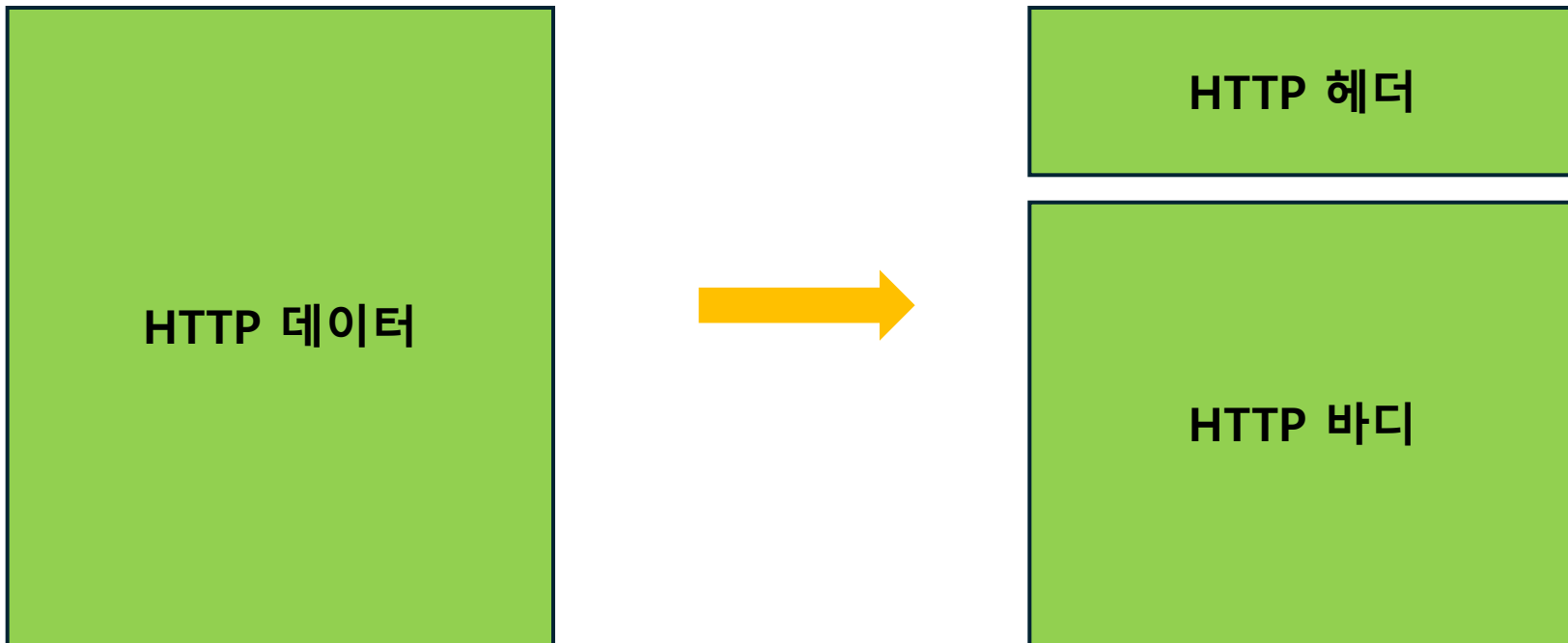
서버 주소 (domain)

데이터 위치 (path)

Query String

프로토콜과 HTTP

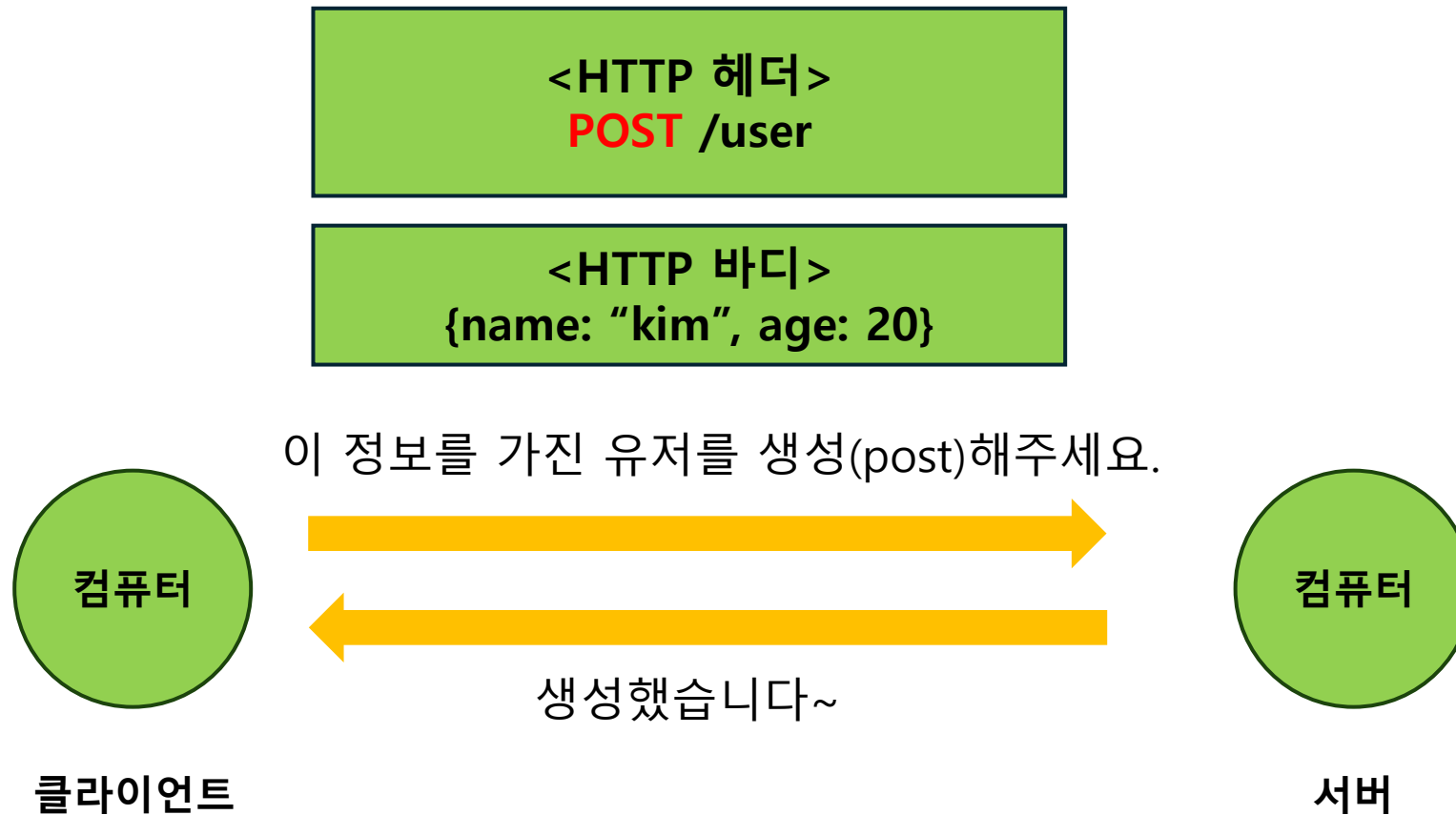
- HTTP로 주고받는 데이터의 구조



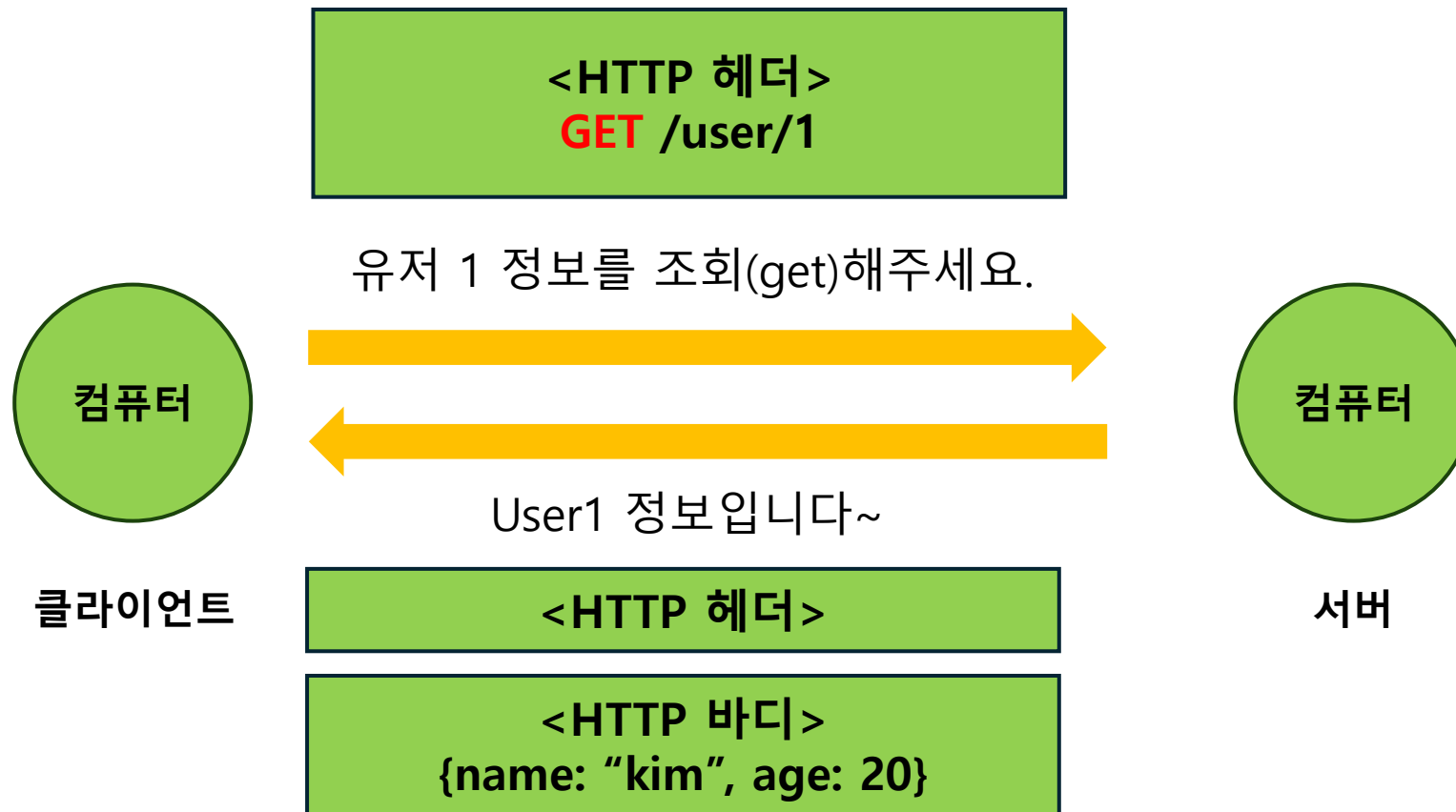
프로토콜과 HTTP

- HTTP 헤더
 - 통신에 대한 정보
(언제 보냈는지, 누가 보내는지, HTTP method 종류, 요청 경로 등)
- HTTP 바디
 - 주고 받으려는 데이터
(보통 json 형식)

프로토콜과 HTTP



프로토콜과 HTTP



프로토콜과 HTTP

- 요청에 대한 처리 결과는 HTTP가 정의하는 **상태 코드**로 나타낸다.
- 상태코드는 응답 데이터의 HTTP 헤더에 들어간다.

프로토콜과 HTTP


대표적인 상태 코드

- 200 → 처리 성공 (ok)
- 201 → 데이터 생성 성공 (created)
- 400 → 클라이언트 요청 오류 (bad request)
- 404 → 요청 데이터 없음 (not found)
- 500 → 서버 에러 (internal server error)


프론트엔드와 백엔드



프론트엔드와 백엔드




유아등 베스트셀러 100
사운드북부터 인기 그림책까지
우리 아이 감성 지수 UP
교보문고




무료 배송
배송비 부담없이 즐기는 도서
1만원 이상 무료 배송
웅진북센


분야별 신간 도서




시니어 힐링 컬러링북 : 만다...
베이직콘텐츠랩 ...




세계사를 움직이는 다섯 가지 힘
사이토 다카시 ...




에도가와 란포 기담집
에도가와 란포 ...




나는 얼마짜리입니까
6411의 목소리 ...




이서윤쌤의 초등 글쓰기 처방전
이서윤 · 메가스...




스펙타클 (spectacle) 1(창...
스펙타클 편집부...




입구 매거진(2호): 고독과 공...
입구매거진 편집...




지도로 보아야 보인다
에밀리 오브리, ...




프리세일즈, 고객과 기업을 잇는...
최종일 · 책만




세프의 가벼운 레스토랑
송사열 · 웅감한...




2025 이기적 정보처리기사 ...
영진정보연구소 ...



사이트 다카시의 홈치는 글쓰기
사이트 다카시 ...



34.1°
구름많음
26° / 35°
미세 보통 · 초미세 보통



33° 31° 29° 28° 27°
17시 19 21 23 1

증시 ① 08.14. 16:18 ↻

독일 17,812.05 ▲85.58 +0.48%

삼성전자 ▲1.45% 77,200

시프트업 ▼11.11% 69,600

HD현대에... ▲30.00% 31,850

인기종목 더보기 >

위젯 보드




캘린더

8.14 수

로그인하기

일	월	화	수	목	금	토
28	29	30	31	1	2	3
4	5	6	7	8	9	10
11	12	13	14	15	16	17
18	19	20	21	22	23	24
25	26	27	28	29	30	31

VIBE 오후의클래식 다른 추천 보기 ↻

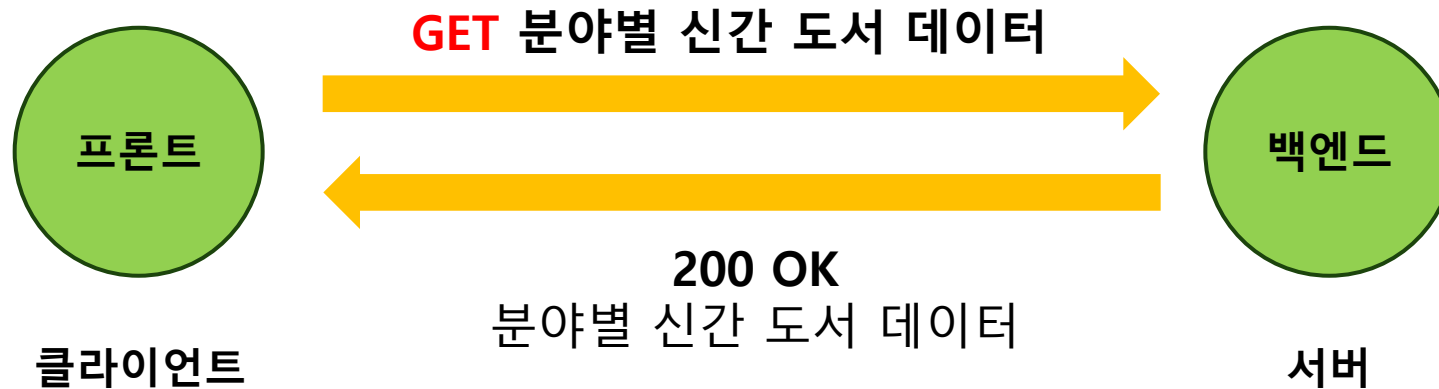




프론트엔드와 백엔드

- 응답받은 화면에는 여러가지 종류의 컨텐츠가 있다.
그런데 이 컨텐츠는 매일 바뀔 수 있는 컨텐츠이다.
- 그렇다면 html 코드를 매일 수정해야 하는 걸까?
 - 자주 변하지 않는 **화면 UI**와, 자주 변하는 **컨텐츠**를 분리
 - **프론트엔드**와 **백엔드**의 분리

프론트엔드와 백엔드

- **프론트**는 화면에 채울 콘텐츠 데이터를 백엔드에게 **요청**
- **백엔드**는 DB에서 가져온 콘텐츠 데이터를 프론트에게 **응답**



프론트엔드와 백엔드

- 백엔드가 주고받는 데이터는 보통 **JSON**(JavaScript Object Notation) 형태

```
[
  {
    "name": "Molecule Man",
    "age": 29,
    "secretIdentity": "Dan Jukes",
    "powers": ["Radiation resistance", "Turning tiny", "Radiation blast"]
  },
  {
    "name": "Madame Uppercut",
    "age": 39,
    "secretIdentity": "Jane Wilson",
    "powers": [
      "Million tonne punch",
      "Damage resistance",
      "Superhuman reflexes"
    ]
  }
]
```

API

- 프론트와 백엔드 역시 각각 웹에서 동작하는 컴퓨터 어플리케이션
→ **HTTP**를 사용하여 통신한다.
- **HTTP**는 웹에서 데이터를 주고받는 단순한 **규칙**
→ 구체적인 통신 방법은 규칙 안에서 **직접 정의**한다.

API

- Application Programming Interface
- 어플리케이션에서 원하는 기능을 수행하기 위해
어플리케이션과 소통하는 구체적인 방법을 정의한 것
- 쉽게 말하면 어플리케이션의 사용 설명서

백엔드 API

프론트가 백엔드에 요청을 보낼 때

- 어떤 http method, url을 사용해야 하는지 정의한 것
- 각 요청에 대해 어떤 응답을 보내는지 정의한 것

REST API

- REST 아키텍처를 따르도록 설계한 API

간단하게 정리하면..

- **URL**은 조작할 **데이터**를 나타낸다.
- **HTTP method**는 데이터에 대한 **행위**를 나타낸다.
- 그 외 자잘자잘한 규칙들..

REST API

- 세부적인 규칙은 잘 정리된 글들을 참고하자.
- <https://velog.io/@somday/RESTful-API-%EC%9D%B4%EB%9E%80>

프로젝트

Todo mate API 서버 클론 코딩 (<https://www.todomate.net/>)

<핵심 기능>

- 각 유저마다 자신의 todo list를 갖고 있다.
- 각 유저는 자신의 todo list를 친구와 공유할 수 있다.

프로젝트

Todo mate API 서버 클론 코딩 (<https://www.todomate.net/>)

<세부 기능>

- 유저 회원가입 / 로그인
- 로그인한 유저의 할 일 생성 / 조회 / 수정 / 삭제
- 로그인한 유저의 할 일 체크 / 체크 해제
- 친구 추가 / 친구 조회 / 친구 삭제
- 특정 친구의 할 일 조회

프로젝트

프로젝트 진행 과정

1. **API 설계**
2. DB, ERD 설계
3. API 서버 프로그램 작성
4. 테스트
5. 배포

프로젝트 - 과제

API 명세 작성하기

- API 명세를 작성할 때는 request body, request header, response body, status code 등도 함께 정의해야 한다.
- 이번 과제에서는 **HTTP Method, URL**만 정의해보자.

프로젝트 - 과제

- **URL**을 정의할 때는 공통적인 서버 정보를 빼고, **path**만 정의
- Path는 **대분류** → **소분류** 단계로 점점 구체화되도록 작성하자.

프로젝트 - 과제

- `/member/{member_id}/nickname`

→ `member` 그룹 안에서,
`{member_id}` 를 가진 member의,
`nickname` 데이터

프로젝트 - 과제

- 로그인, 좋아요와 같은 기능은 URL에 동사를 그대로 쓰고 **POST**로 호출하기도 한다.
- POST /login
- POST /logout
- POST /register
- POST /like

프로젝트

Todo mate API 서버 클론 코딩

<세부 기능>

- 유저 회원가입 / 로그인
- 로그인한 유저의 할 일 생성 / 조회 / 수정 / 삭제
- 로그인한 유저의 할 일 체크 / 체크 해제
- 친구 추가 / 친구 조회 / 친구 삭제
- 특정 친구의 할 일 조회

프로젝트 - 예시

- 할 일 전체 조회 : GET /todo/list
- 할 일 생성 : POST /todo
- 할 일 수정 : PATCH /todo/{todo_id}
- 할 일 삭제 : DELETE /todo/{todo_id}
- 할 일 체크 : POST /todo/{todo_id}/check
- 할 일 체크해제 : POST /todo/{todo_id}/uncheck

- 로그인한 유저의 할 일 생성 / 조회 / 수정 / 삭제
- 로그인한 유저의 할 일 체크 / 체크 해제

프로젝트 - 예시

- API 명세에 정답은 없으니 자유롭게 설계해주세요 😊

1주차 과제

- WIL 작성하기
- WIL 안에 API 명세서 작성하기
- 프로젝트 생성하기 (노션 과제 명세 참고)

제출할 파일

- wil.md 파일
- 생성한 프로젝트 소스코드

수고하셨습니다 😊

- 즐거운 추석 보내세요~!