

2019

웹 프로그래밍 기초

1. 인터넷과 웹 브라우저

인터넷을 상상하다 : 정보가 연결되어 있는 사회

노버트 위너(Norbert Wiener, 1894~1964)

- 사람의 신경 작용에서 영감을 받아 인간과 기계 그리고 사회 전반에서 서로 통신해야 한다.(사이버네틱스 이론)

버니바 부시(Vannevar Bush, 1890~1974)

- 연결된 데이터들이 들어 있는 거대한 도서관을 예측

J.C.R. 리클라이더(J.C.R. Licklider, 1915~1990)

- 커뮤니케이션 창시자로서 컴퓨터를 고안하여 온라인 공동체 개발을 논의

컴퓨터 광역 네트워크를 예측

인터넷을 상상하다 : 정보가 연결되어 있는 사회

패킷 교환(packet switching)

- 작은 블록의 패킷으로 쪼개서 데이터를 전송하는 방식.
- 데이터를 작게 쪼개서 먼 곳에 보냈다가 다시 합침.
- 1960년대 초 영국과 미국의 연구자들이 발명

인터넷의 시작 : ARPANET

- 냉전시대인 1960년대, 미국 국방부 주도 하에 계획.
- 정보를 교환하여 연구 역량을 극대화하려는 목적.
- 분산 네트워크로 하나가 붕괴하더라도 나머지에 영향이 없도록 하려는 목적

인터넷의 시작 : ARPANET 그 이후

- 군대에서 시작되어 대학과 기업으로 퍼져나갔으나 소수 엘리트만 사용.
- 초기 인터넷은 파일과 자원 공유, 원거리에 있는 컴퓨터에 로그인하려는 목적.
- 일반인에게도 PC(Personal Computer)가 보급되기 시작.

인터넷의 시작 : 월드와이드웹(WWW)의 발명

유럽입자물리연구소(CERN)의 소속 연구원 팀 버너스 리가 고안.

- 한 문서에서 다른 문서로 접근한다는 개념인 하이퍼텍스트(Hypertext)를 적용한 HTML(HyperText Mark-up Language)
- 문서에 주소를 붙이는 URL(Uniform/Universal Resource Locator)
- 컴퓨터와 컴퓨터간의 메시지를 주고받을 수 있는 통신 규약 HTTP(HyperText Transfer Protocol)

전세계 인터넷 콘텐츠를 연결해 공유할 수 있는 WWW(World Wide Web)의 발명

인터넷의 시작 : 월드와이드웹(WWW)의 발명

- 최초의 사이트 : <http://info.cern.ch/>

인터넷의 시작 : 웹의 진화

하이퍼텍스트 문서

- 기본적인 서식과 하이퍼링크를 내재한 텍스트 문서

웹 페이지

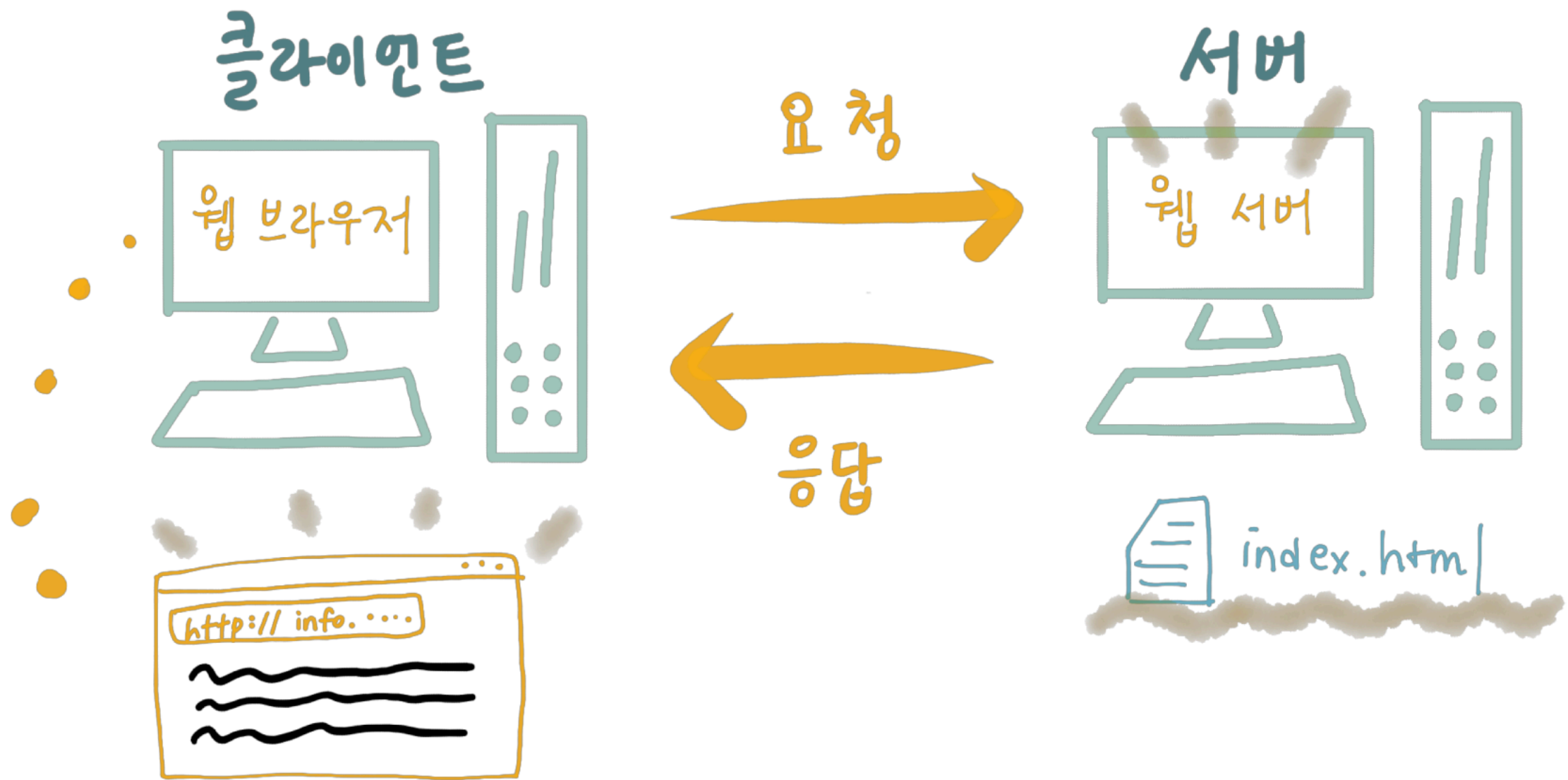
- 이미지, 오디오 등 리소스를 제공하고 풍부하고 화려한 레이아웃 내재화
- 시각적으로는 아름답지만 사용자와의 상호작용이 불가능해서 마치 종이에 인쇄된 페이지 같음

웹 애플리케이션

- 사용자와 상호작용이 가능하고 복잡한 처리도 가능한 웹 브라우저 프로그램

2. 서버와 클라이언트

서버와 클라이언트



서버와 클라이언트

클라이언트 : 요청을 보내는 주체

- 크롬, 사파리, 엣지, 파이어폭스 등 웹 브라우저
- 데스크탑 앱 또는 모바일 앱
- 다른 서버에 요청을 보내는 서버
- 개발자가 개발한 별도의 프로그램

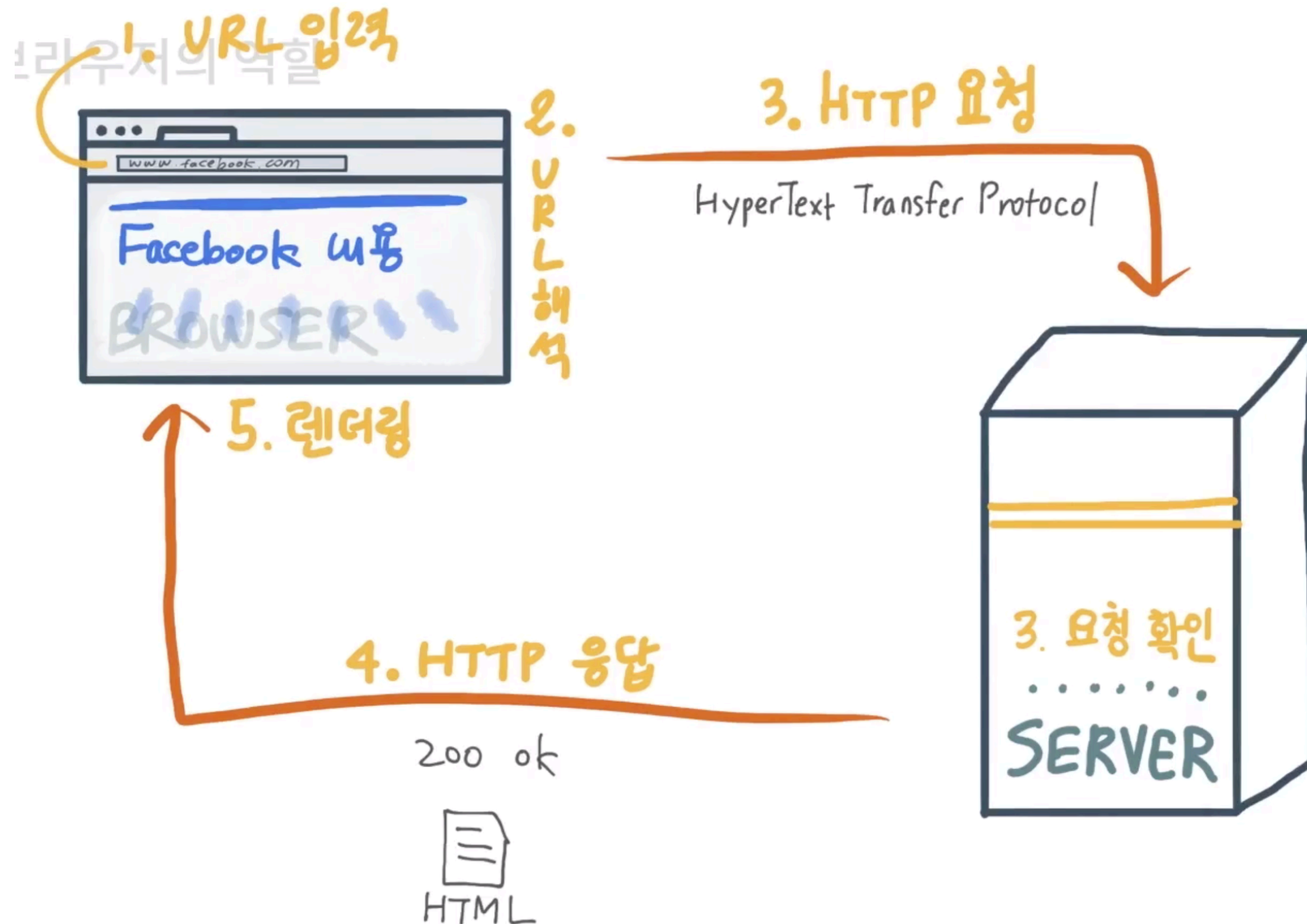
서버 : 네트워크를 통해 정보 혹은 서비스를 제공하는 컴퓨터 또는 프로그램

- 웹 사이트를 저장한 컴퓨터
- 웹 서버 애플리케이션을 통해 만들어진 애플리케이션

웹 프로그래밍이란

- HTTP로 통신하는 클라이언트와 서버를 개발하는 것

HTTP와 웹 브라우저의 역할



HTTP와 웹 브라우저의 역할

HTTP 요청

메서드 Method

서버가 수행해야 할 동작.
GET, POST, DELETE - - - -

URL 주소

헤더 Header

브라우저 정보, 언어 등
여러 정보가 포함.

본문 Body

id/PW, 새 글 내용 등. 없어도 됨.

HTTP 응답

상태 코드

요청의 성공 여부.
200 OK, 404 Not Found - -

헤더 Header

본문 Body

HTML 코드 등의 파일.
예러가 났다면 생략될 수도 있다.

HTTP와 웹 브라우저의 역할

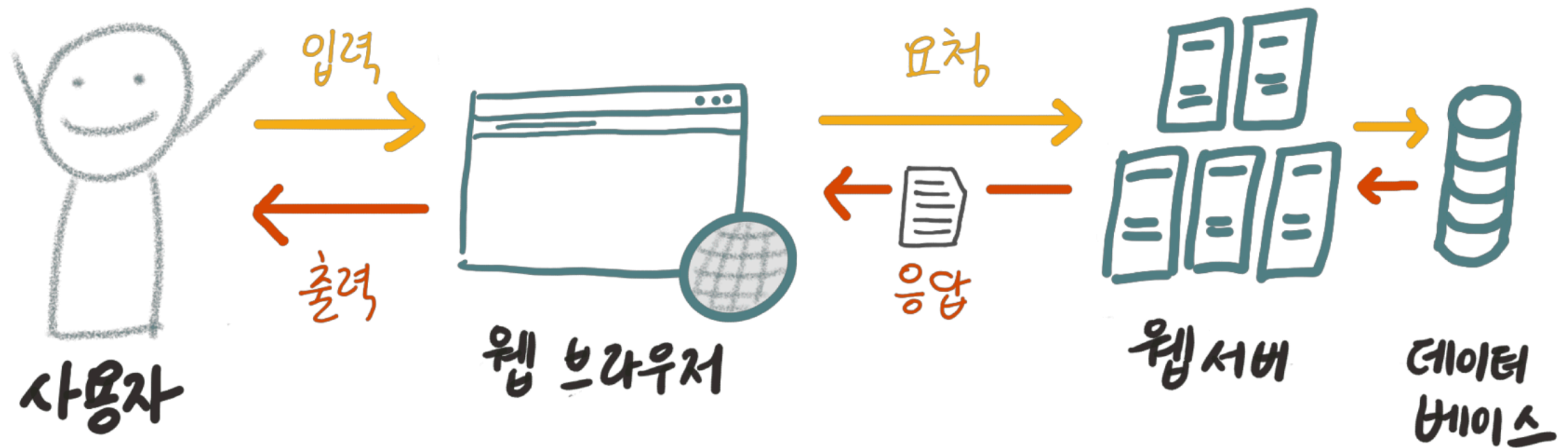
- 웹 브라우저와 서버는 리소스를 주고 받는 형식을 미리 약속(프로토콜)으로 정해두었고 이에 따라 동작한다.

개발자 도구 사용하기

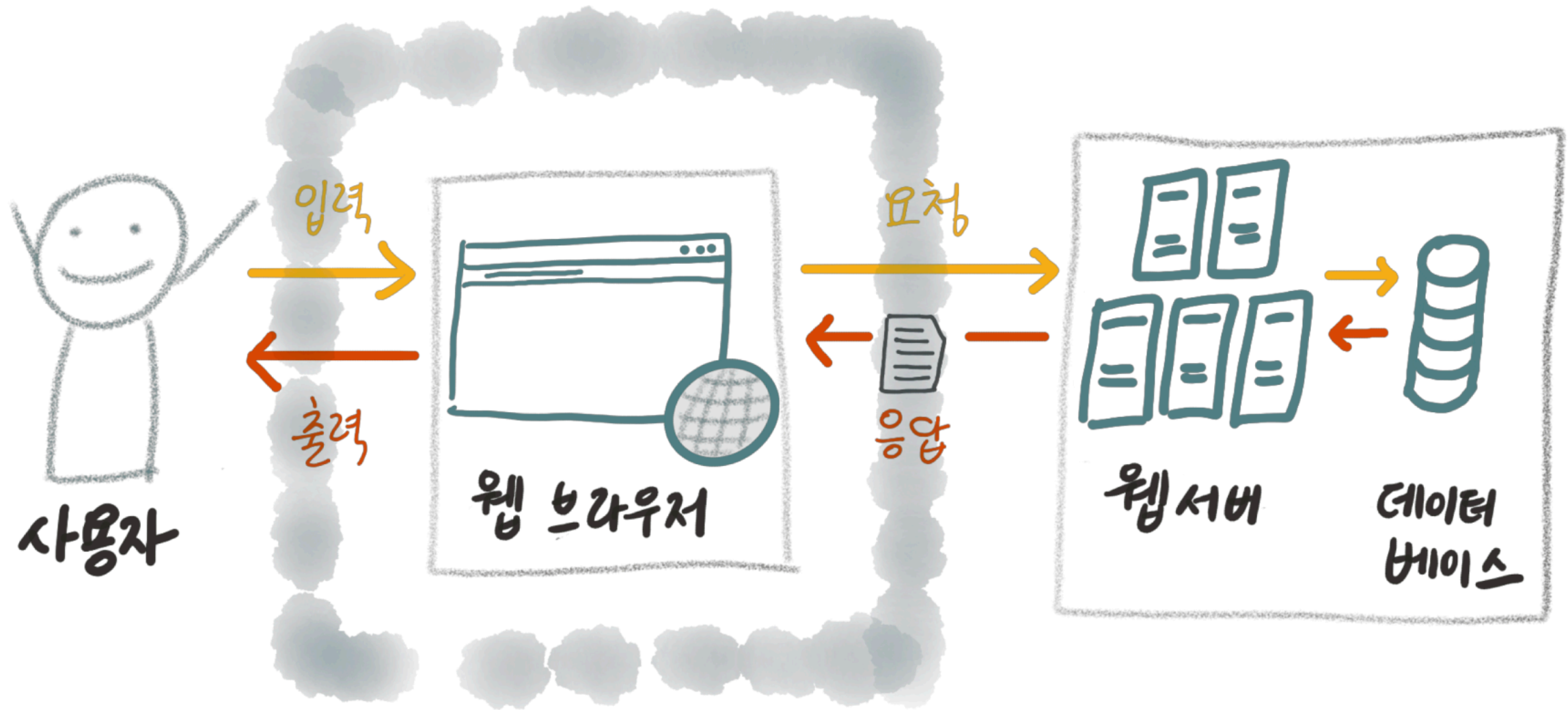
- <https://developers.google.com/web/tools/chrome-devtools/>

3. 프론트엔드와 백엔드

프론트엔드와 백엔드



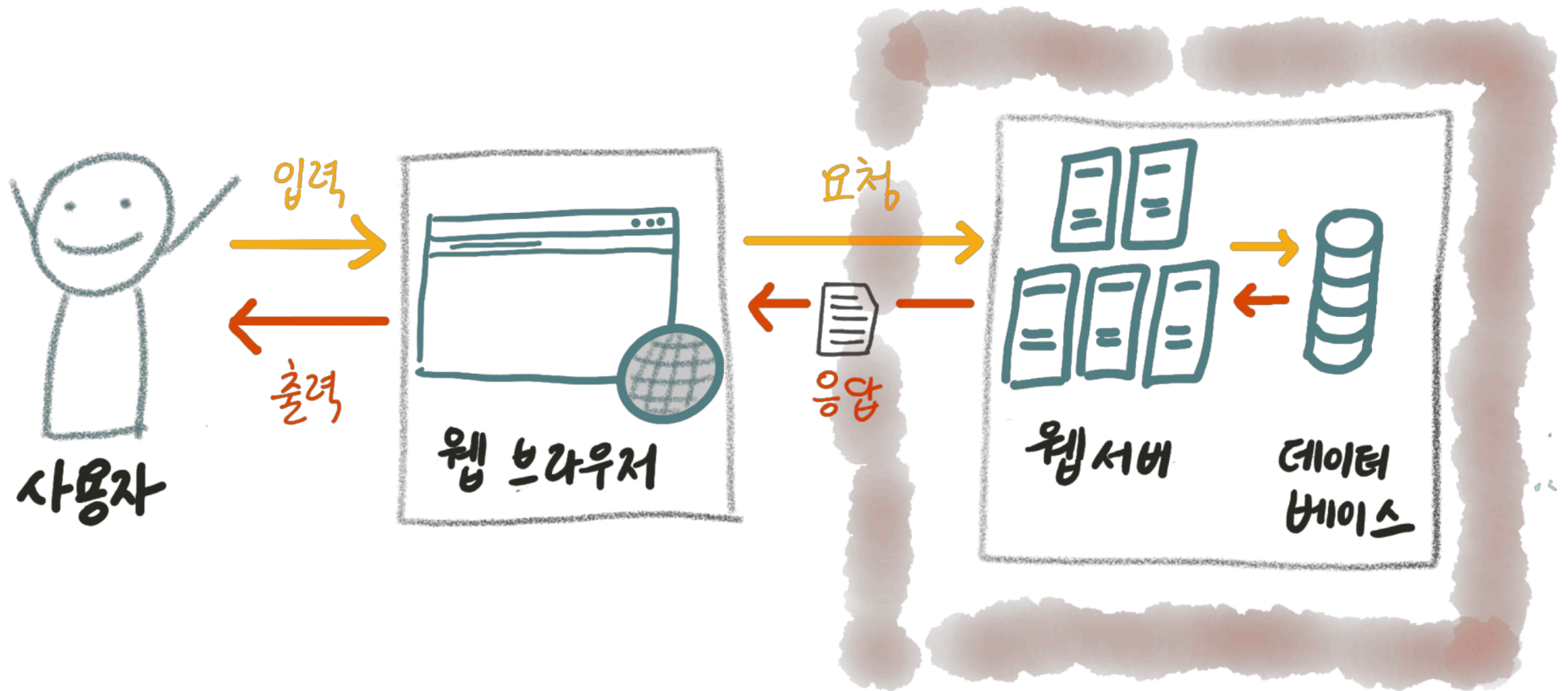
프론트엔드와 백엔드



사용자의 요청을 잘 반영
적절한 레이아웃과 보아줄 디자인
웹 콘텐츠를 구조화

JavaScript
CSS
HTML

프론트엔드와 백엔드



운영체제 (Linux ...), 네트워크

데이터베이스

Spring, Flask/Django 등 프레임워크

Java, PHP, Python 등 프로그래밍 언어

4. 웹 프레임워크

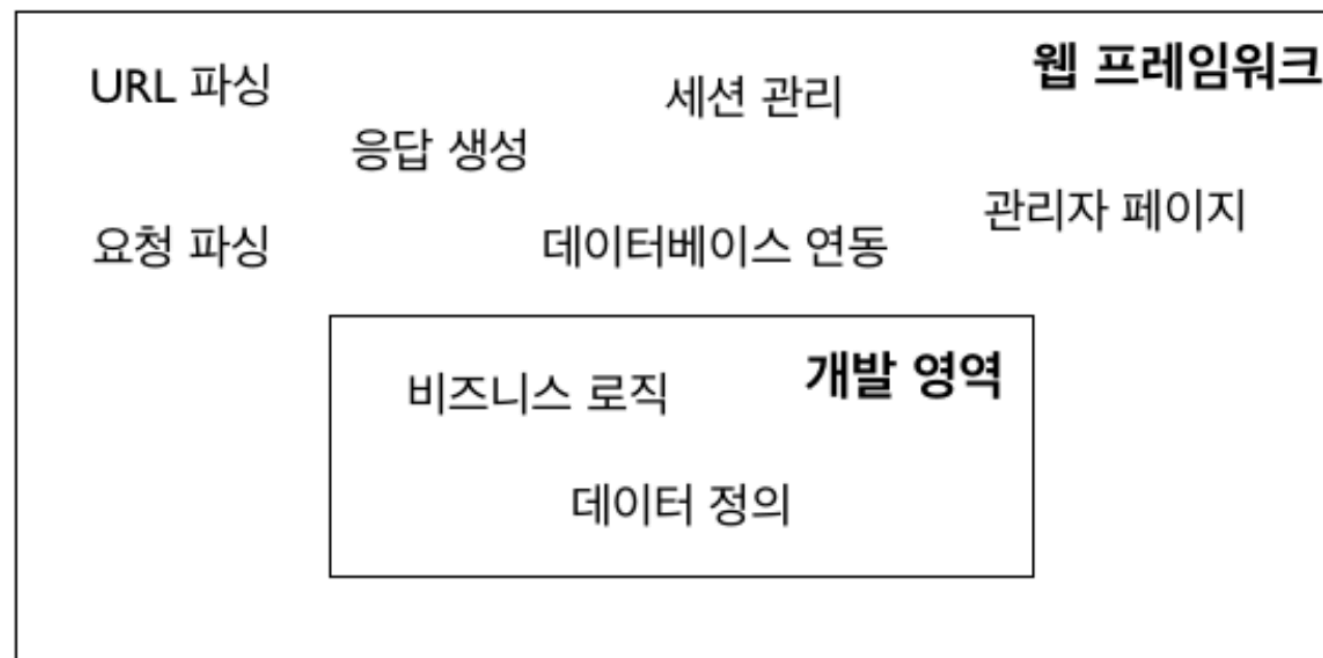
프레임워크

- 자주 사용되는 코드를 체계화하여 쉽게 사용할 수 있도록 도와주는 코드 집합.
- 라이브러리와 혼동될 수 있지만 좀 더 규모가 크고 프로젝트의 기반이 됨.
- 건축에 비유하면 구조를 만드는 골조가 프레임워크라면 그 외 자재들이 라이브러리가 됨.

웹 프레임워크

- 웹 개발에 필요한 기본적인 구조와 코드(클래스, 함수 등)가 만들어져 있음.

웹 개발



Django

- <https://wiki.python.org/moin/WebFrameworks/>
- <https://www.djangoproject.com/>

MTV 계층

모델(Model)

- 테이블을 정의
- 데이터베이스 처리

템플릿(Template)

- 사용자 인터페이스 정의 및 처리

뷰(View)

- 애플리케이션의 제어 흐름 및 처리 로직 정의
- HTTP 요청 처리

MTV 계층

- 모델, 템플릿, 뷰 모듈 간에 독립성 유지.
- 느슨한 결합(Loose Coupling) 설계 원칙 부합.
- 디자이너, 응용 개발자, DB 설계자 간에 협업 쉬워짐.