

# 응용 계층-HTTP의 기초

전제윤

# 도메인 네임과 DNS

## 도메인 네임

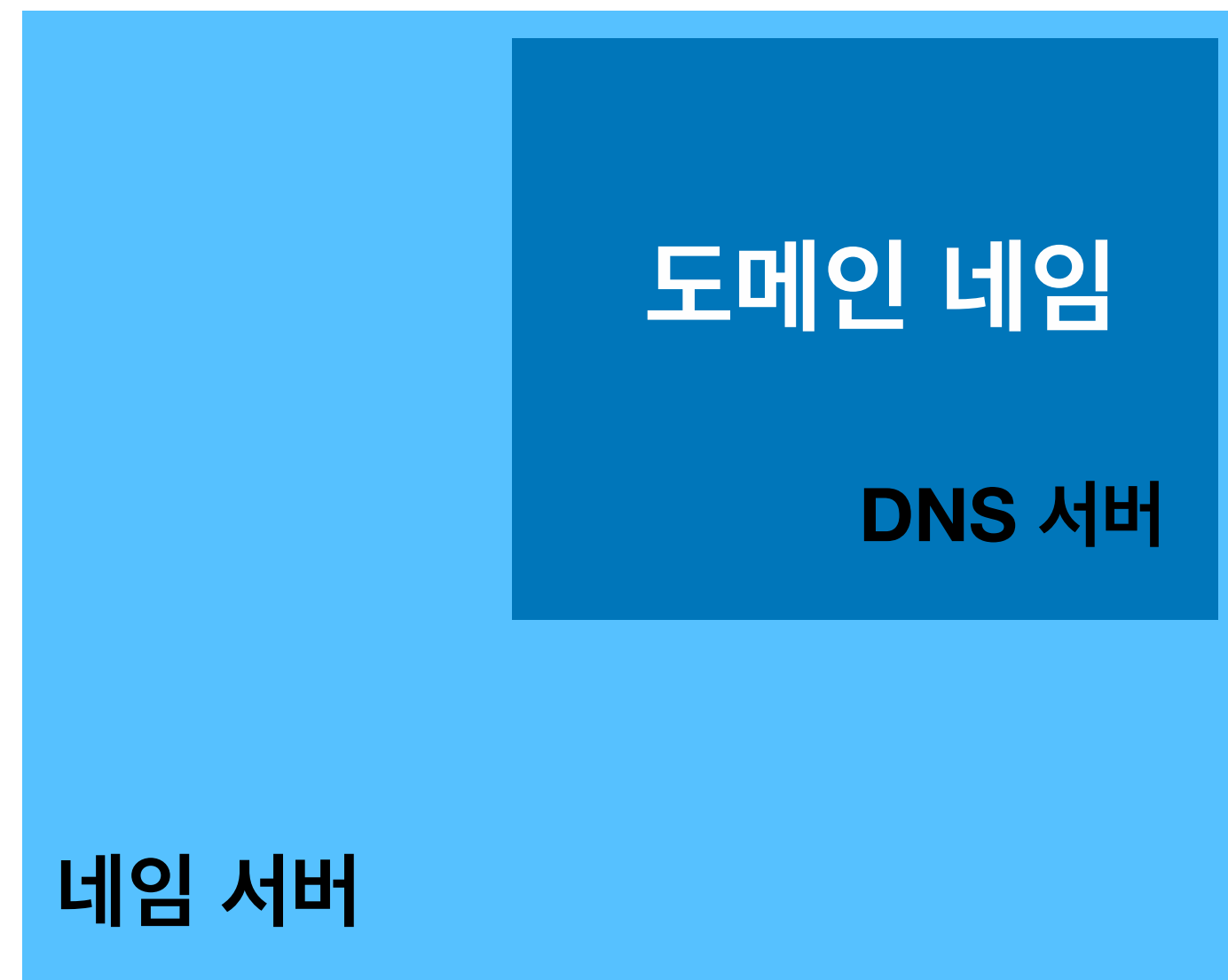
- 예시
  - 1) www.example.com
  - 2) www.naver.com

# 도메인 네임과 DNS

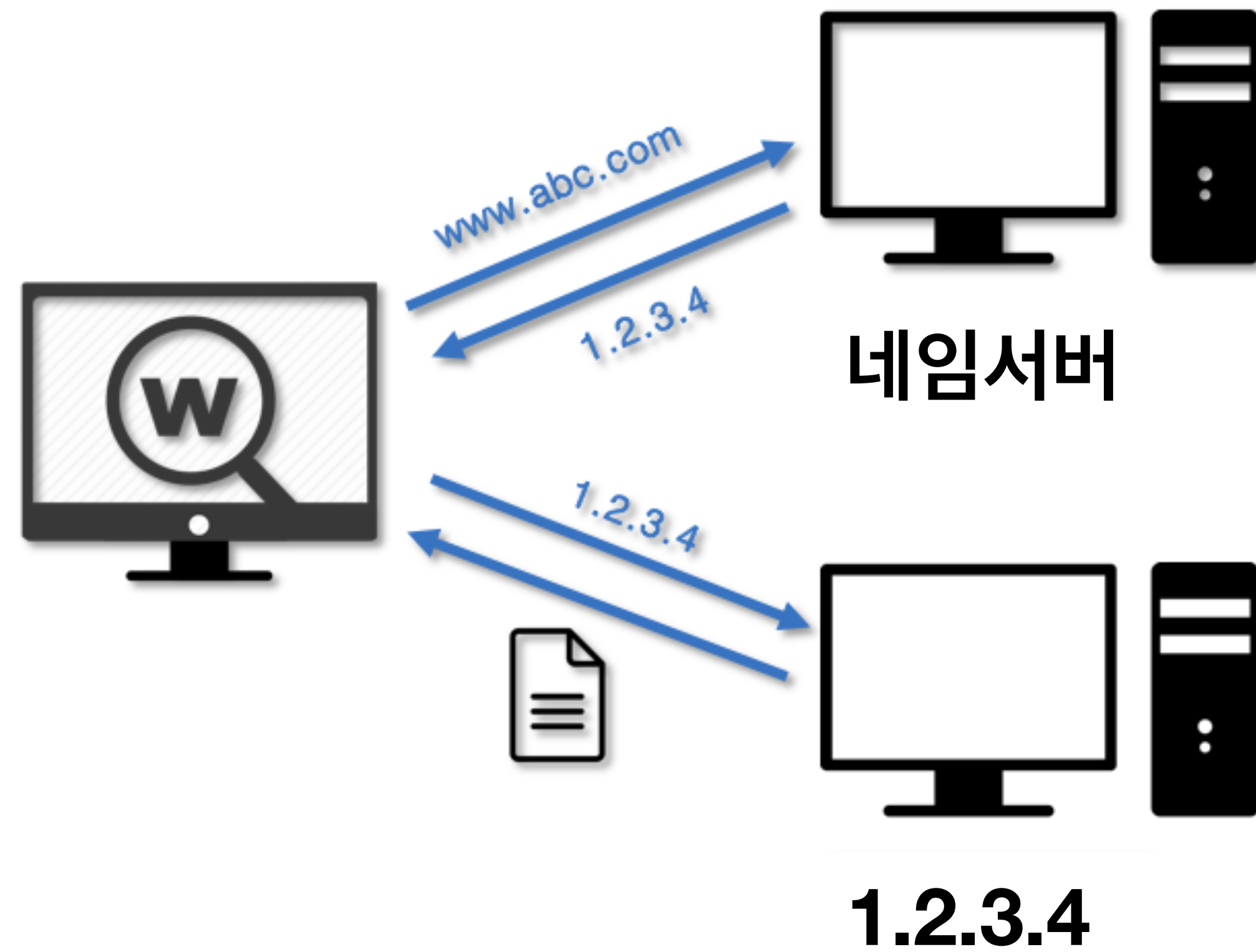
도메인 네임 + IP 주소

네임 서버

# 도메인 네임과 DNS



# 리졸빙(resolving)



# 루트 도메인 과 최상위 도메인(TLD)



---

**FQDN(Fully-Qualified Domain Name)**

# 도메인 네임 시스템

naver.com



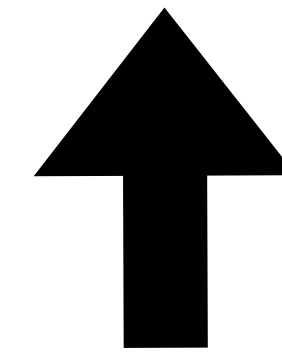
로컬 네임 서버

# 도메인 네임 시스템

naver.com



로컬 네임 서버



ISP

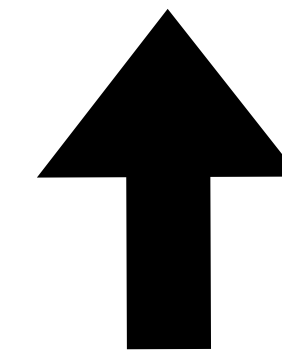


# 도메인 네임 시스템

n timer.com



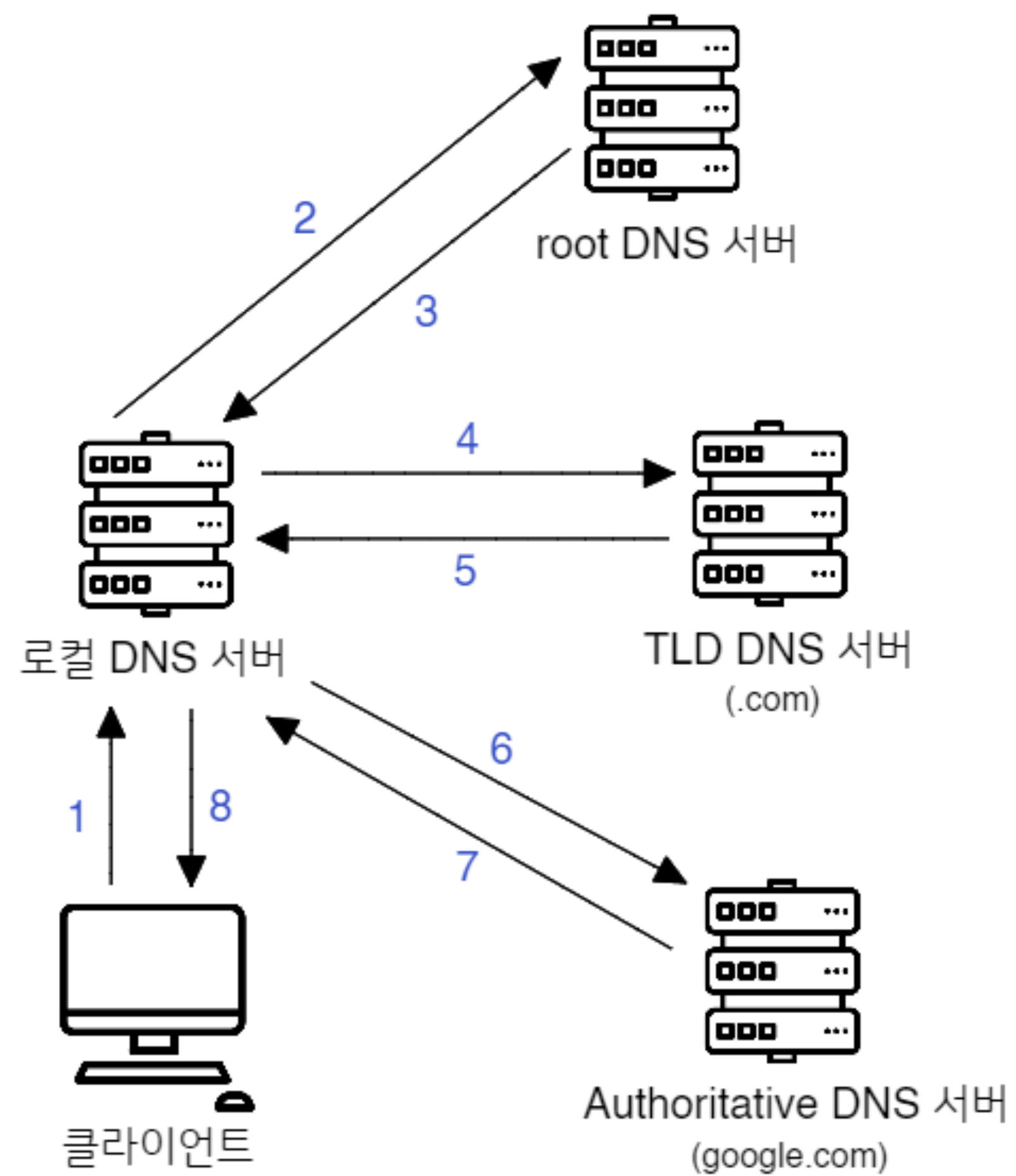
로컬 네임 서버



공개 DNS 서버

Best Free & Public DNS Servers		
Provider	Primary DNS	Secondary DNS
<a href="#">Google</a>	8.8.8.8	8.8.4.4
<a href="#">Quad9</a>	9.9.9.9	149.112.112.112
<a href="#">OpenDNS Home</a>	208.67.222.222	208.67.220.220
<a href="#">Cloudflare</a>	1.1.1.1	1.0.0.1
<a href="#">CleanBrowsing</a>	185.228.168.9	185.228.169.9
<a href="#">Verisign</a>	64.6.64.6	64.6.65.6
<a href="#">Alternate DNS</a>	198.101.242.72	23.253.163.53

# IP 주소 찾기



# 자원과 URI/URL

- 자원
  - 네트워크상에서 메시지를 통해 주고받는 최종 대상
- URI
  - 웹 상에서 자원을 식별하기 위한 정보
  - 자원(Resource)을 식별(Identifier)하는 통일된 방식(Uniform)
  - 이때 ‘이름’을 기반으로 식별하기도, ‘위치’로 자원을 식별하기도 한다.

# 위치로 자원을 식별한다! URL!

**foo://www.example.com:8042/over/there?name=ferret#nose**

**Scheme = 자원에 접근하는 방법**

# 위치로 자원을 식별한다! URL!

**foo://www.example.com:8042/over/there?name=ferret#nose**

**authority** = 호스트를 특정할 수 있는 IP주소나 도메인 네임이 명시

# 위치로 자원을 식별한다! URL!

**foo://www.example.com:8042/over/there?name=ferret#nose**

**Path = 자원이 위치하고 있는 경로**

# 위치로 자원을 식별한다! URL!

**foo://www.example.com:8042/over/there?name=ferret#nose**

**Query = 매개변수 역할을 하는 문자열**



# 위치로 자원을 식별한다! URL!

**foo://www.example.com:8042/over/there?name=ferret#nose**

**Fragment** = 자원의 일부분, 자원의 한 조각을 가리키기 위한 정보

**[Https://datatracher.ietf.org/doc/html/rfc3986](https://datatracher.ietf.org/doc/html/rfc3986)**

**<https://datatracher.ietf.org/doc/html/rfc3986#section-1.1.2>**

# HTTP의 특징

- 프로그래밍에 큰 영향을 미치므로 중요하다!

# HTTP의 특징

요청과 응답을 기반으로 동작한다(요청 응답 기반 프로토콜)

# HTTP의 특징

## 미디어 독립적 프로토콜

- HTML, JPEG, PNG, JSON, XML, PDF 와 같이 다양한 종류의 자원을 주고받는다
- HTTP에서 주고받는 자원의 종류를 미디어타입이라고 부른다.

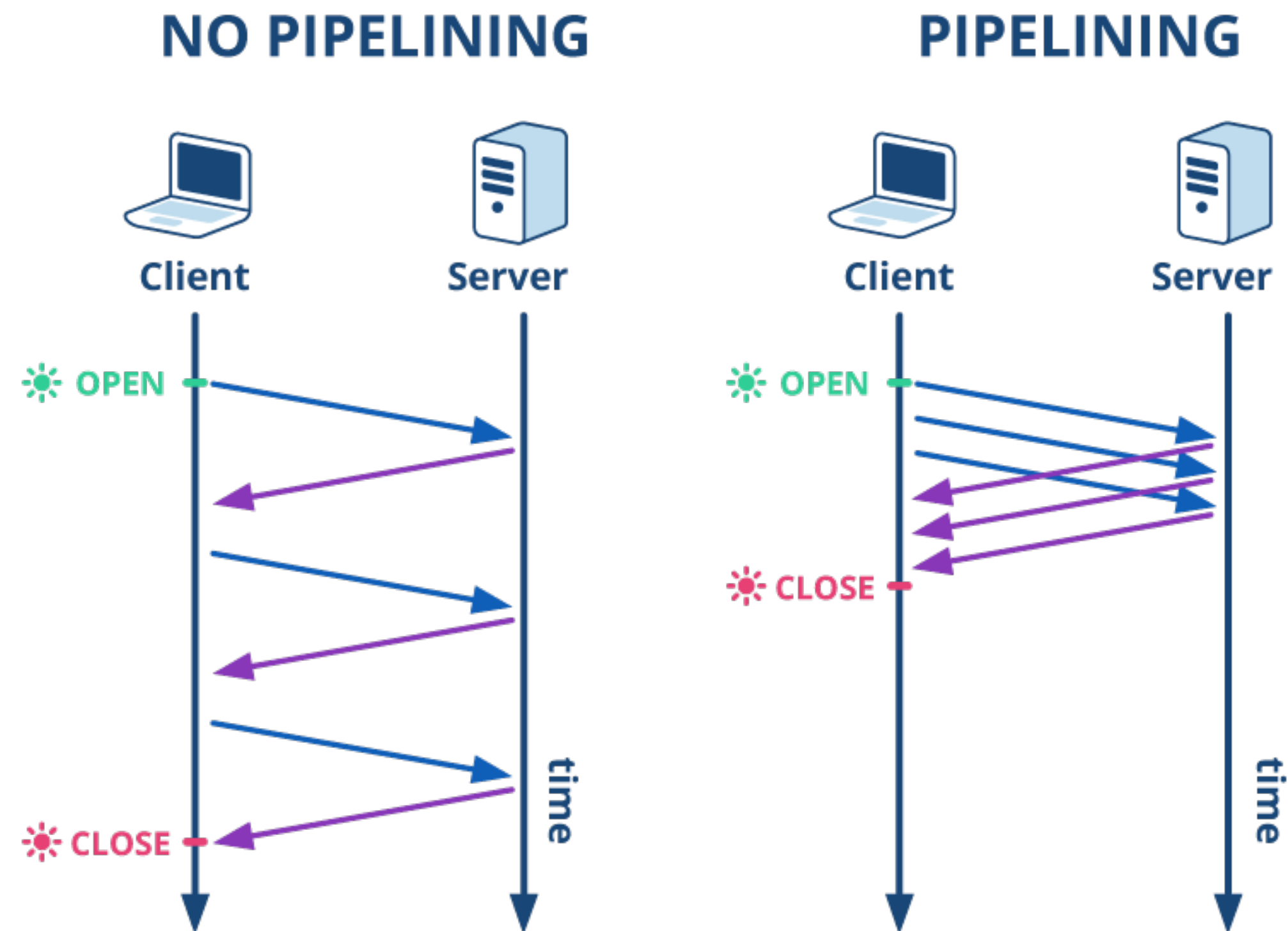
# HTTP의 특징

## 스태이트리스 프로토콜

- 클라이언트의 상태정보를 서버에 저장하지 않는다.

# HTTP의 특징

## 지속 연결 프로토콜



# HTTP 메서드

HTTP 메서드	설명
GET	자원을 습득하기 위한 메서드
HEAD	GET과 동일하나, 헤더만을 응답받는 메서드
POST	서버로 하여금 특정 작업을 처리하게끔 하는 메서드
PUT	자원을 대체하기 위한 메서드
PATCH	자원에 대한 부분적 수정을 위한 메서드
DELETE	자원을 삭제하기 위한 메서드



# HTTP 상태코드

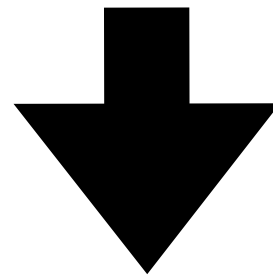
상태코드	의미	범위 별 의미
200	클라이언트의 요청이 성공적으로 끝남	클라이언트 요청 성공
400	요청한 method를 지원하지 않음	클라이언트 요청에 대한 방향 재정의, 추가 동작 필요
401	인증오류	
403	사용자 허가 모드 오류	
404	요청한 파일이 존재하지 않음	
405	요청한 method를 지원하지 않음	
500	Internal Server Error(실행오류)	서버오류

# HTTP 주요 헤더

## Host

**GET /hypertet/WWW/TheProject.html HTTP/1.1**

**Host: info.cern.ch**



**<http://info.cern.ch/hypertet/WWW/TheProject.html>**

# HTTP 주요 헤더

## User-Agent

웹 브라우저의 Mozilla 호환 여부\*

렌더링 엔진\*\* 관련 정보

User-Agent:Mozilla/5.0(Windows NT 10.0; Win64;rv:109.0) Geclo/20100101 Firefox/109.0

운영체제 및 아키텍처 정보

브라우저와 버전 정보

\* 오늘날 웹 브라우저를 통한 요청 메시지에 대부분 포함된 정보

\*\* 브라우저에 시각적 요소를 구현하는 구성 요소

# HTTP 주요 헤더

## Referer

Referer: <http://minchul.net>

# 응답메시지에서 활용되는 HTTP 주요 헤더!

## Server

**Server : Apache/2.4.1 (Unix)**

# 응답메시지에서 활용되는 HTTP 주요 헤더!

## **Allow**

**HTTP/1.1 405 Method Not Allowed**

- 
- 
- 

**Allow : POST, OPTIONS**

# 응답메시지에서 활용되는 HTTP 주요 헤더!

## Location

- 클라이언트에게 자원의 위치를 알려 주기 위해 사용됩니다.

▼ General	
Request URL:	http://localhost:7742/api/maps
Request Method:	POST
Status Code:	● 201 Created
Remote Address:	:::1:7742
Referrer Policy:	strict-origin-when-cross-origin
▼ Response Headers <input type="checkbox"/> Raw	
Access-Control-Allow-Headers:	Content-Type, Origin, Accept, Authorization, Content-Length, X-Requested-With
Access-Control-Allow-Methods:	GET,POST,PUT,PATCH,DELETE,HEAD,OPTIONS
Access-Control-Allow-Origin:	*
Connection:	keep-alive
Content-Length:	0
Content-Type:	application/json; charset=utf-8
Date:	Wed, 25 Oct 2023 06:24:28 GMT
Keep-Alive:	timeout=5
Location:	/api/maps/23

**감사합니다**