

목표: 신입 기술면접을 위한 CS 기본 지식 답변 준비

- 단골 CS 기본 지식 문항에 대해 관련 지식을 압축적으로 이해하고, 답변 준비

금일 목표

- : 다음 단골 CS 네트워크 지식 문항에 대해 구체적으로 이해하고, 답변 준비하기
1. HTTP와 HTTPS의 차이점에 대해 간략히 설명해주세요
 2. 쿠키와 세션의 차이점에 대해 간략히 설명해주세요
 3. HTTP GET, POST 방식에 대해 간략히 설명해주세요
 4. 웹브라우저를 실행시켜서 주소창에 특정 URL 값을 입력시킨 후, 엔터를 눌렀을 때, 페이지가 렌더링되는 과정을 웹통신 흐름에 중점을 두어 가능한 구체적으로 설명해주세요.

과제: 다음 수업까지, 위 문항에 대해 답변을 작성하여, 제출하기 (jhleeroot@gmail.com)

TCP/IP 모델

- 인터넷 통신을 위한 모델
- OSI 7 Layer는 표준 모델, 현실에서 인터넷을 위해 사용하는 모델은 TCP/IP 모델
- TCP/IP 모델의 응용 계층은 OSI 모델의 세션+표현+응용 계층 통합

| | | |
|----------------------------|-------------------------------------|-----------------------------------|
| 응용 계층 (Application Layer) | HTTP, FTP, DNS, POP3, SMTP 등등 | 응용 계층 (Application Layer) |
| 표현 계층 (Presentation Layer) | | |
| 세션 계층 (Session Layer) | | |
| 전송 계층 (Transport Layer) | TCP, UDP | 전송 계층 (Transport Layer) |
| 네트워크 계층 (Network Layer) | IPv4, IPv6 | 인터넷 계층 (Internet Layer) |
| 데이터링크 계층 (Data Link Layer) | Ethernet | 네트워크 접속 계층 (Network Access Layer) |
| 물리 계층 (Physical Layer) | | |

OSI 7 Layer (OSI 7 레이어)

| 계층 | 이름 | 설명 |
|-----|----------------------------|--|
| 7계층 | 응용 계층 (Application Layer) | 이메일, 파일 전송, 웹사이트 열람등 애플리케이션에 대한 서비스 제공 |
| 6계층 | 표현 계층 (Presentation Layer) | 문자 코드, 압축, 암호화등의 데이터 변환 담당 |
| 5계층 | 세션 계층 (Session Layer) | 세션 채결, 통신 방식등을 결정 |
| 4계층 | 전송 계층 (Transport Layer) | 신뢰할 수 있는 통신 구현 |
| 3계층 | 네트워크 계층 (Network Layer) | 다른 네트워크와 통신을 위한 경로 설정 및 논리 주소 결정 |
| 2계층 | 데이터링크 계층 (Data Link Layer) | 네트워크 기기간 데이터 전송 및 물리 주소 결정 |
| 1계층 | 물리 계층 (Physical Layer) | 시스템간 물리적 연결과 전기신호 변환 및 제어 |

다양한 응용 계층 프로토콜

이 중에서도 다음 프로토콜에 대해서는 개괄적으로 이해 필요

- HTTP/HTTPS (HyperText Transfer Protocol): 웹사이트 접속
- FTP (File Transfer Protocol): 파일 송수신
- SMTP/POP3 (Simple Mail Transfer Protocol): 메일 송신/수신
- DNS (Domain Name System): 이름 해석

다양한 응용 계층 프로토콜

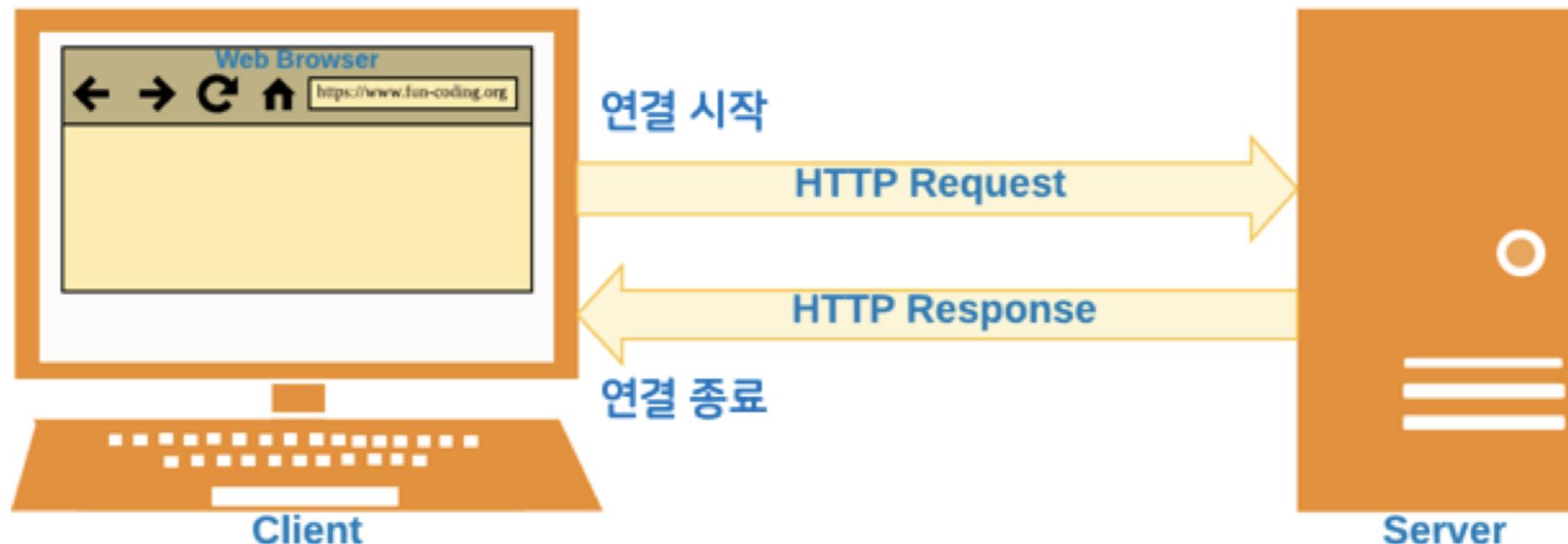
- HTTP (HyperText Transfer Protocol)
- 인터넷(WWW)상에서 정보를 주고받는 프로토콜
- HTTP method: GET, POST, PUT, DELETE

WWW 이란?

- WWW 이란?
 - W3 또는 웹(Web)
 - 주요 요소: HTML, URL, HTTP
- HTML: markup 언어, hypertext 와 hyperlink(또는 링크)로 구성

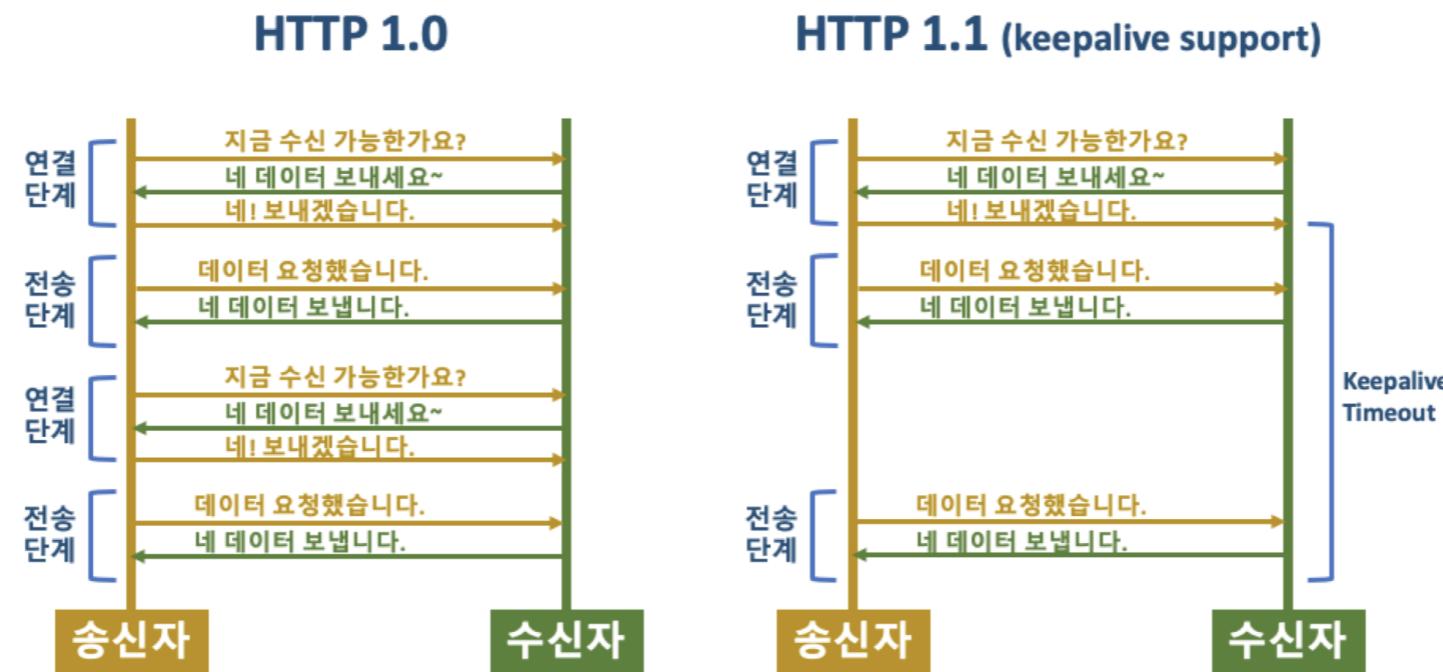
HTTP(Hypertext Transfer Protocol)

- Server/Client 모델로 Request/Response 사용
 - Client에서 요청(Request)을 보내면, Server에서 응답(Response)을 준다.
 - HTTP는 Connectionless 한 프로토콜임 - 1회성 Request 및 Response!
 - TCP/IP socket을 이용해서 연결됨



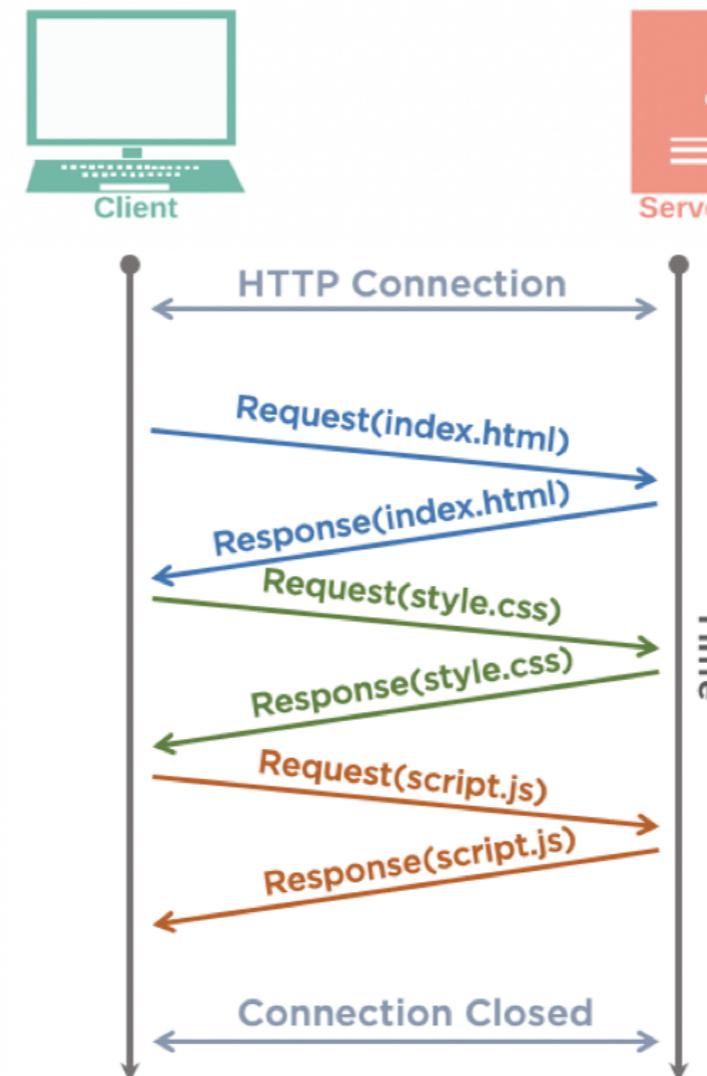
HTTP 1.1

- HTTP는 Connectionless 방식
- HTTP 1.1은 keepalive 기능을 통해, 서버에서 설정된 keepalive timeout 까지는 연결과정 없이 데이터 송수신 가능
 - 내부적으로 결국 매번 TCP 3-way handshake 과정을 거칠 필요가 없어짐



확인: vi /etc/apache2/apache2.conf

일반적인 HTTP(Hypertext Transfer Protocol) 송수신



HTTP(Hypertext Transfer Protocol) Request/Response

HTTP 헤더는 읽을 수 있음 (1.1 버전 기준)

- Request

GET / HTTP/1.1

Host: www.fun-coding.org

Connection: keep-alive

- Response

HTTP/1.1 200 OK

Server: Apache – 웹 서버 정보

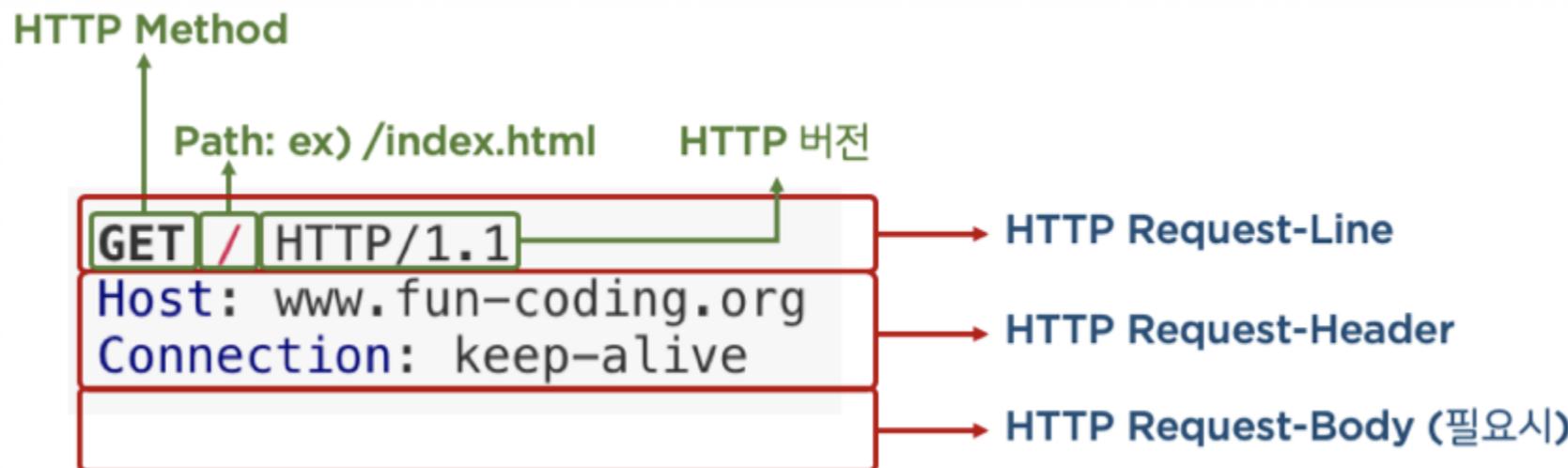
Content-type: text/html – MIME 타입

Content-length: 107 – HTTP Body 길이

<html><head></head> – HTML 데이터

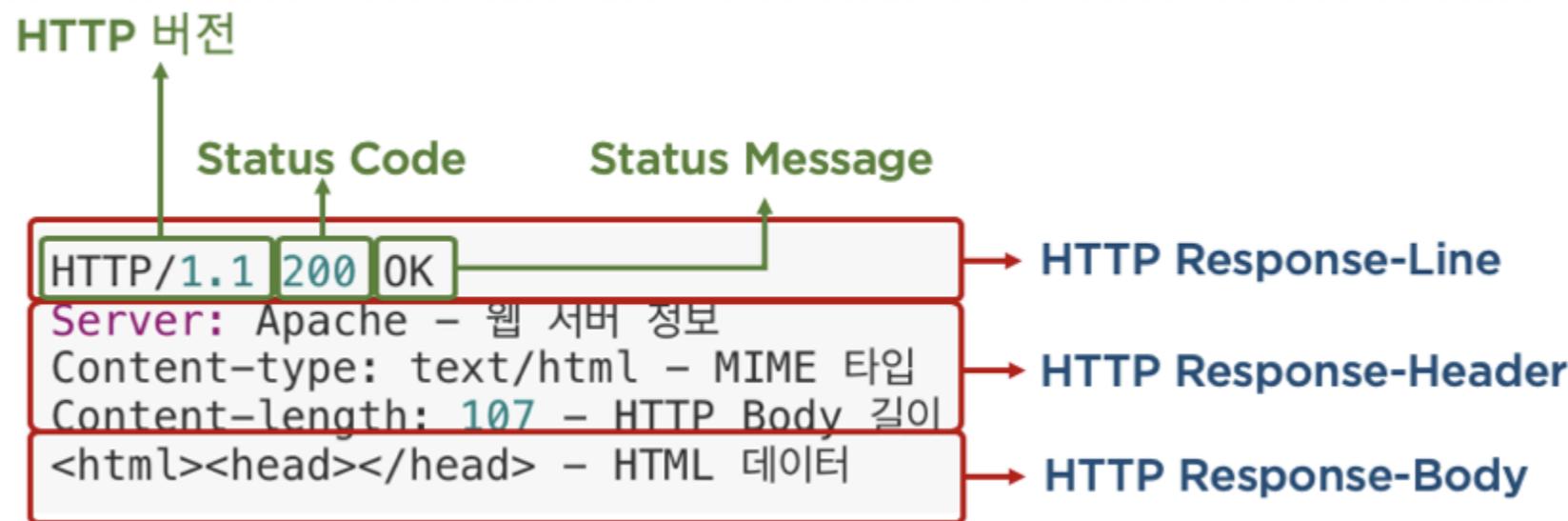
HTTP(Hypertext Transfer Protocol) Request/Response

- Request



HTTP(Hypertext Transfer Protocol) Request/Response

- Response

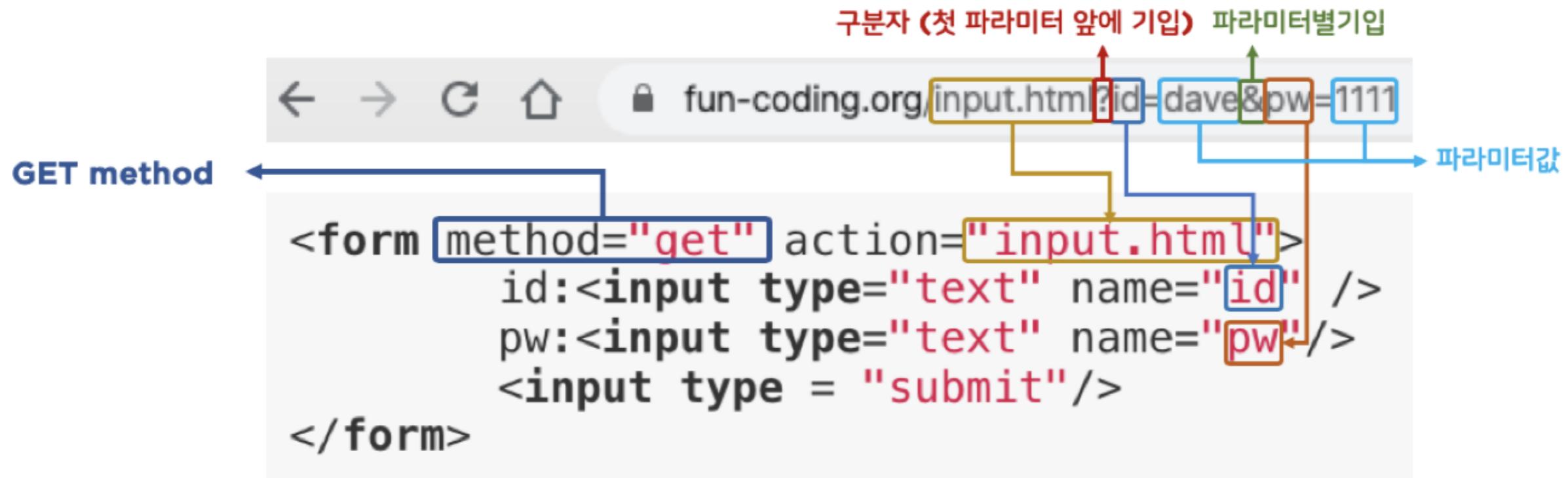


HTTP(Hypertext Transfer Protocol) 실습

| socket 프로그래밍으로 확인해보기!

주요 Request Method

- GET: 정보 읽기(SELECT)
 - 전달이 필요한 파라미터들은 URL을 통해 전달



주요 Request Method

- POST: 정보 입력하기(INSERT)
 - 전달이 필요한 파라미터들은 HTTP body에 포함되어 전달되므로, 사용자는 직접적인 확인 불가

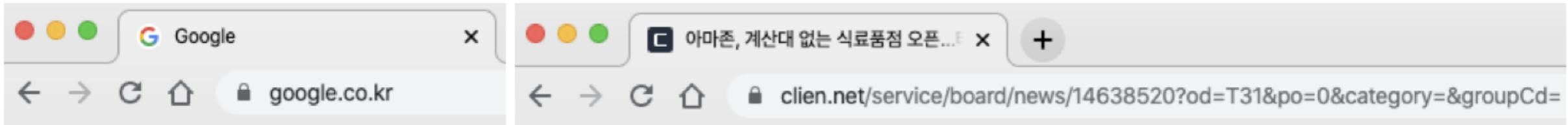
```
POST method ←  
<form method="post" action="input.htm">  
    id: <input type="text" name="id" />  
    pw: <input type="text" name="pw"/>  
    <input type = "submit"/>  
</form>
```

주요 Request Method

- PUT: 정보 수정하기(UPDATE), DELETE: 정보 삭제하기(DELETE)

주요 Request Method

- 웹사이트 접속시는 일반적으로 GET을 통해 HTML을 가져옴



주요 Request Method

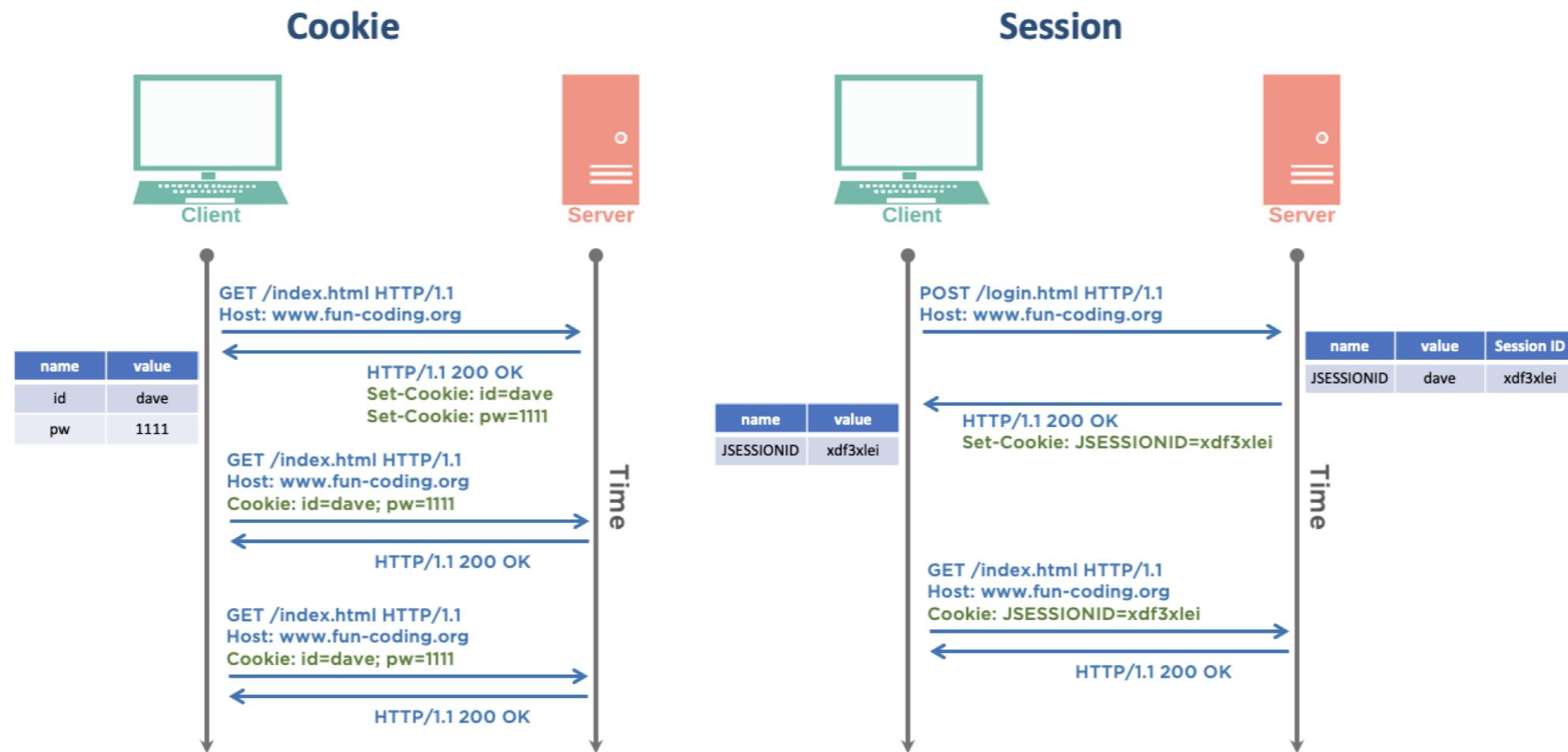
- 주요 HTTP 응답 코드

| 상태 코드 | 의미 |
|------------|------------------------|
| 200 | 정상 |
| 400 | 유효하지 않은 파라미터 또는 잘못된 요청 |
| 401 | 승인되지 않은 액세스 |
| 403 | 액세스 금지 |
| 404 | 리소스를 찾을 수 없음 |
| 500 | 내부 서버 오류 |

쿠키(Cookie)와 세션(Session)

- HTTP는 Stateless: 통신이 끝나면 상태를 유지하지 않음
- 이를 보완하기 위한 기법이 쿠키와 세션

쿠키(Cookie)와 세션(Session)



한빛미디어 마이페이지 정보 크롤링해보기

참고: 쿠키(Cookie)와 세션(Session)

- 쿠키/세션 유효기간
 - expires 설정이 있으면, 로컬 디스크에 저장 및 유효기간 경과시 삭제
 - expires 설정이 없으면, 메모리에 저장 및 브라우저 종료시 삭제

URL

- Uniform Resource Locator
- 인터넷 상의 자원 위치 표기를 위한 규약
- WWW 주요 요소 중 하나: HTML, URL, HTTP

참고: URL vs URI

- URI(Uniform Resource Identifier): 통합 자원 식별자
- URI의 하위 개념이 URL
 - <https://www.fun-coding.org> 주소
 - <https://www.fun-coding.org> 라는 서버를 나타내는 URL이면서 URI
 - <https://www.fun-coding.org/input.html?id=dave&pw=1111> 주소
 - <https://www.fun-coding.org/input.html> 은 URL
 - <https://www.fun-coding.org/input.html?id=dave&pw=1111> 은 URI
 - 내가 원하는 정보를 얻기 위해서는 **?id=dave&pw=1111** 라는 식별자가 필요하기 때문

두 용어는 혼용하고 있으므로, 참고로만 이해

HTTPS

- HyperText Transfer Protocol over Secure Socket Layer
- HTTP 통신시 사용하는 TCP/IP 소켓 통신에서, 일반 텍스트 대신, SSL 또는 TLS 프로토콜을 통해 데이터 암호화하여 송수신
- HTTPS 기본 포트는 443
- http:// 대신 https:// 로 시작

REST

- REST(Representational State Transfer)
 - 자원(resource)의 표현(representation)에 의한 상태 전달
 - HTTP URI를 통해 자원을 명시하고, HTTP Method를 통해 자원에 대한 CRUD Operation 적용
 - CRUD Operation와 HTTP Method
 - Create: 생성 (POST)
 - Read: 조회 (GET)
 - Update: 수정 (PUT)
 - Delete: 삭제 (DELETE)

REST API

- REST 기반으로 서비스 API를 구현한 것
- 마이크로 서비스, OpenAPI(누구나 사용하도록 공개된 API) 등에서 많이 사용됨

참고: Restful

- 개발자들 사이에서 비공식적으로 가이드한 REST 구현 가이드

| CRUD | HTTP verbs | Route | Example |
|----------------------------|------------|---------------|-------------------------------|
| resource 목록 표시 | GET | /resource | 전체 user 목록 표시: [GET] /users |
| resource 중 하나의 resource 표시 | GET | /resource/:id | 1번 user 표시: [GET] /users/1 |
| resource 생성 | POST | /resource | 신규 user 등록: [POST] /users |
| resource 수정 | PUT | /resource/:id | 1번 user 정보 수정: [PUT] /users/1 |
| resource 삭제 | DELETE | /resource/:id | 1번 user 삭제: [DELETE] /users/1 |

참고: Restful

- 기능에 적합한 URL Method 사용
- 정의된 상태 코드에 맞게 상태 전송
- URL Rules
 - 마지막에 / 포함하지 않는다.
 - _(underbar) 대신 -(dash)를 사용한다.
 - 소문자를 사용한다.
 - 행위(method)는 URL에 포함하지 않는다.
 - /users/insert/1 --> [POST] /users/1
 - 컨트롤 자원을 의미하는 URL 예외적으로 동사를 허용한다.
 - /posts/duplicating --> /posts/duplicate

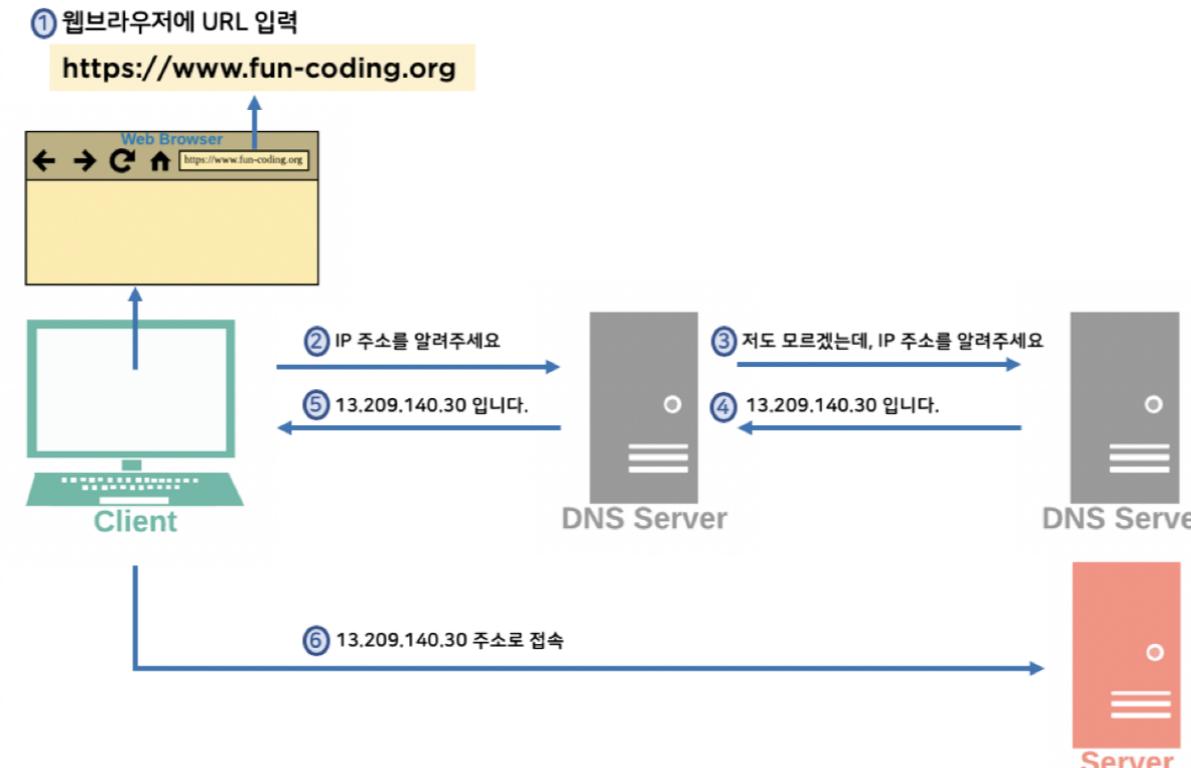
참고: SSL과 TLS

SSL 프로토콜은 넷스케이프사에서 개발

이를 국제 표준화하는 과정에서 SSL이 모기업 상용제품 이름과 동일하여, TLS로 이름을 변경
따라서, SSL = TLS

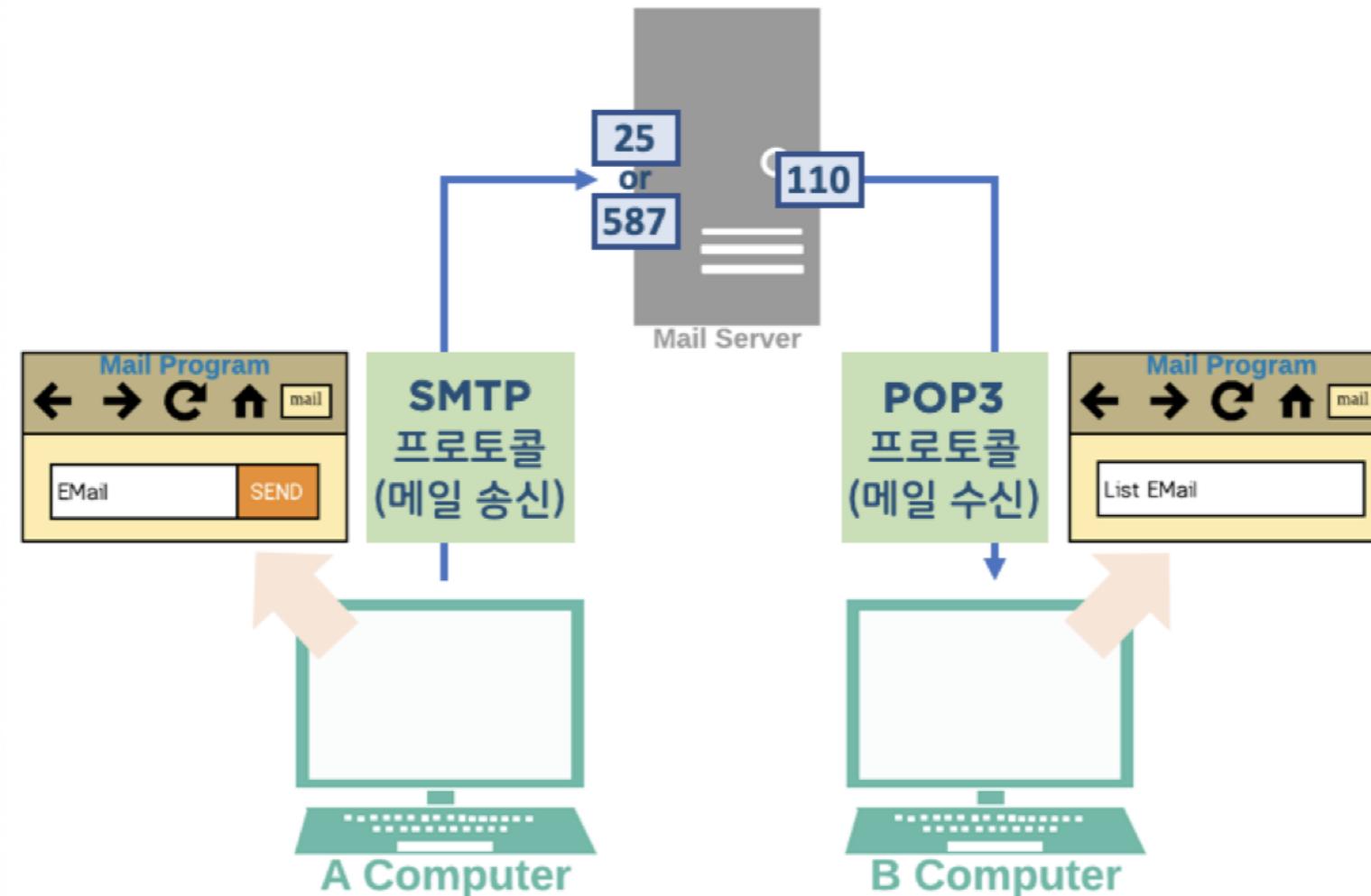
DNS(Domain Name System)

- 도메인 이름을 IP 주소로 변환, 또는 그 반대를 수행하기 위해 개발
 - 전세계 흩어진 DNS 서버를 통해, 도메인 이름:IP 주소 정보 확인 가능
 - DNS는 Network Protocol로 UDP 사용



메일 서버 (SMTP와 POP3 프로토콜)

- SMTP(Simple Mail Transfer Protocol): 메일 송신
- POP3(Post Office Protocol): 메일 수신



참고: POP3와 IMAP

- 둘다 메일 수신 프로토콜
 - IMAP (143 포트): 중앙 서버에서 메일 동기화
 - 모든 장치에서 동일한 이메일 폴더 확인 가능
 - 중앙 서버 메일 용량에 따라, 메일 수신 불가 가능
 - POP3 (110 포트): 중앙 서버에서 로컬 장치로 이메일을 내려 받음
 - 내려받은 이메일은 중앙 서버에서 지우는 것이 디폴트 동작

참고: MIME 프로토콜

- MIME(Multi-purpose Internet Mail Extensions)
 - SMTP 프로토콜은 텍스트(ASCII 코드)만 전송가능
 - SMTP 본문 내용에 MIME 헤더와 함께, 다양한 포맷의 데이터(첨부파일등)를 ASCII 코드로 변환해서 전송
 - 클라이언트에서 SMTP 본문 데이터를 MIME 헤더를 확인해서 디코딩하면 됨!
- MIME 실제 예

MIME-Version: 1.0

From: Dave Lee <jhleeroot@gmail.com>

Date: Thu, 24 Nov 2016 19:35:27 +0900

Subject: =?UTF-8?B?7J207KSA7Z2sI0yeheuLi0uLpC4gK0ydv0ygh0yXkCDtl

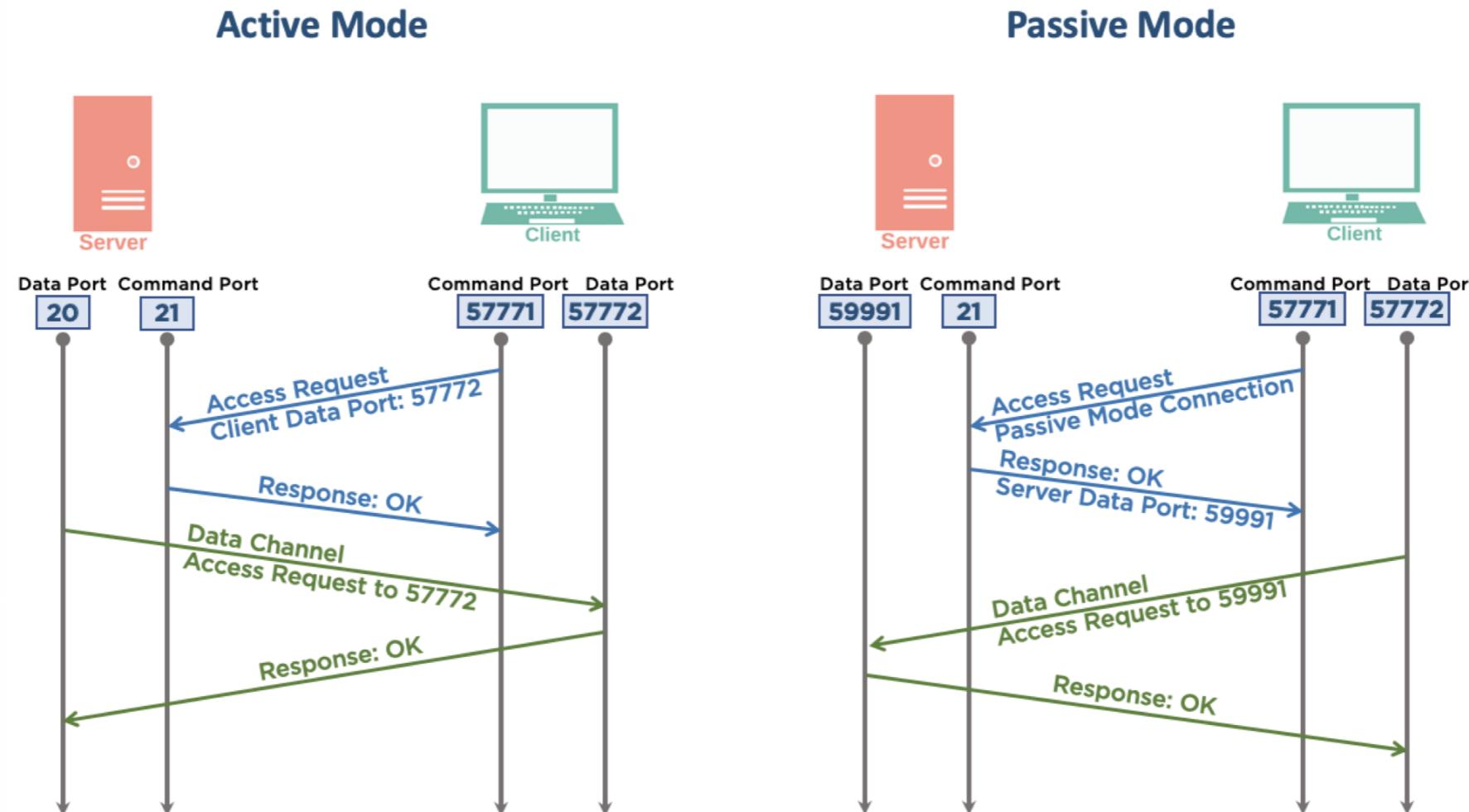
Content-Type: multipart/alternative; boundary=001a113944905f64d4054209900f

FTP

- FTP(File Transfer Protocol): 서버와 클라이언트 사이의 파일 전달을 위한 프로토콜
- 일반 서비스와 달리 2개 포트 사용
 - 21번: 접속/제어를 위한 포트
 - 보통 20번: 데이터 전송을 위한 포트

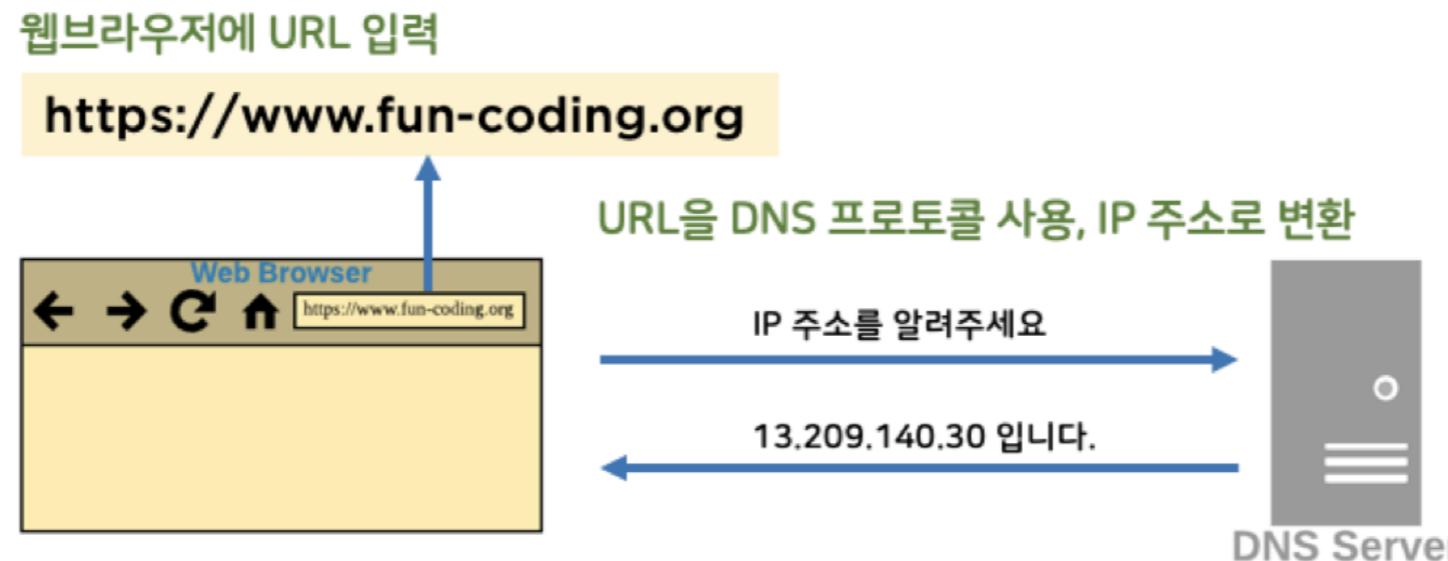
참고: Passive Mode vs Active Mode

- TCP/IP 모델: 클라이언트가 서버 접속
 - Active Mode는 서버가 클라이언트에 접속하므로, 클라이언트 방화벽에 따라 에러 발생 가능



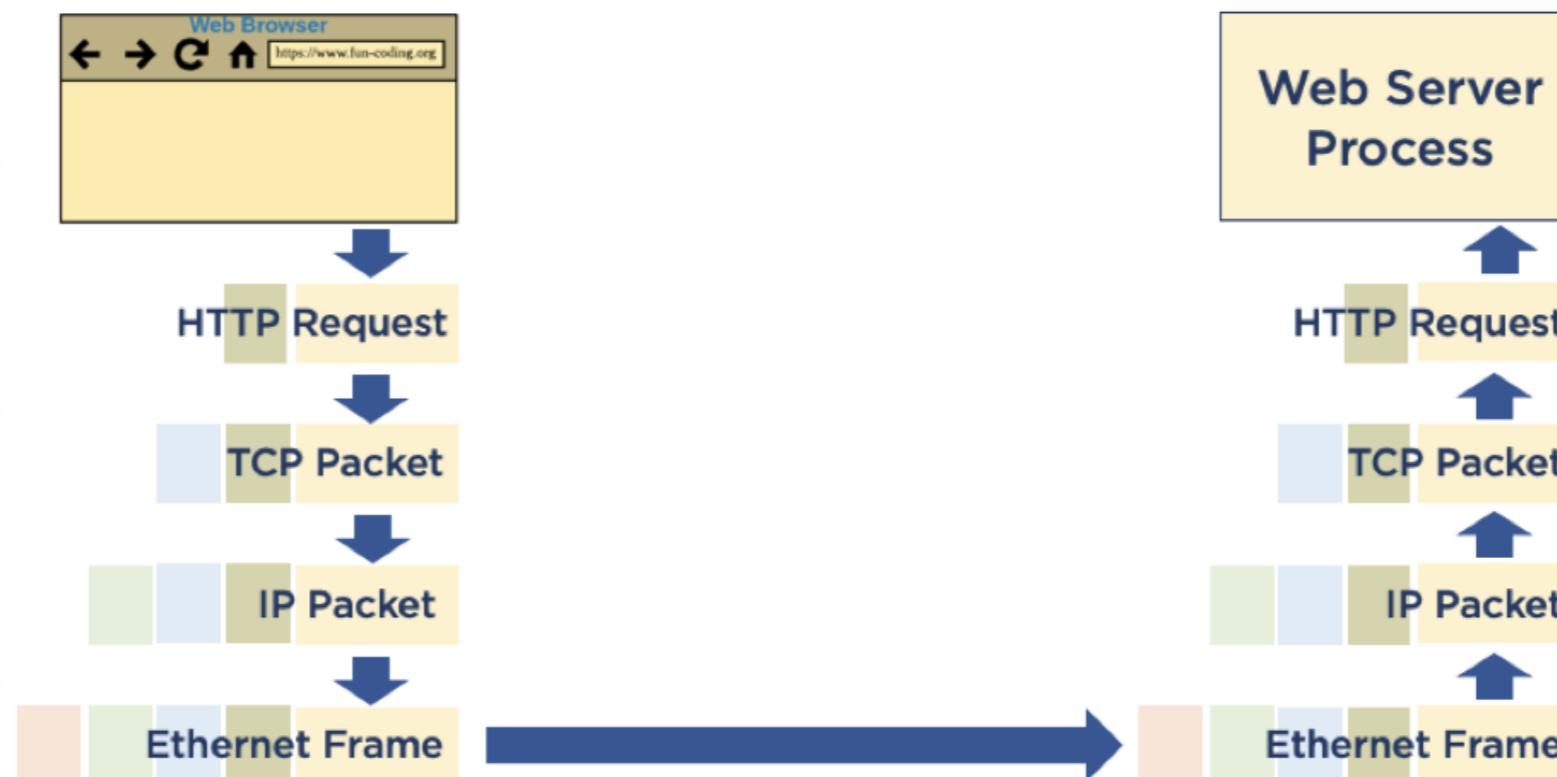
웹브라우저 동작 과정1

- 웹브라우저에 입력된 URL은 DNS 프로토콜을 사용, IP 주소로 변환



웹브라우저 동작 과정2

- HTTP Request 는 운영체제 네트워크 스택을 따라
 - HTTP -> TCP -> IP -> Ethernet 을 전기신호로 변환, 전송



웹서버에서 HTTP Response 를 작성, 브라우저로 송신

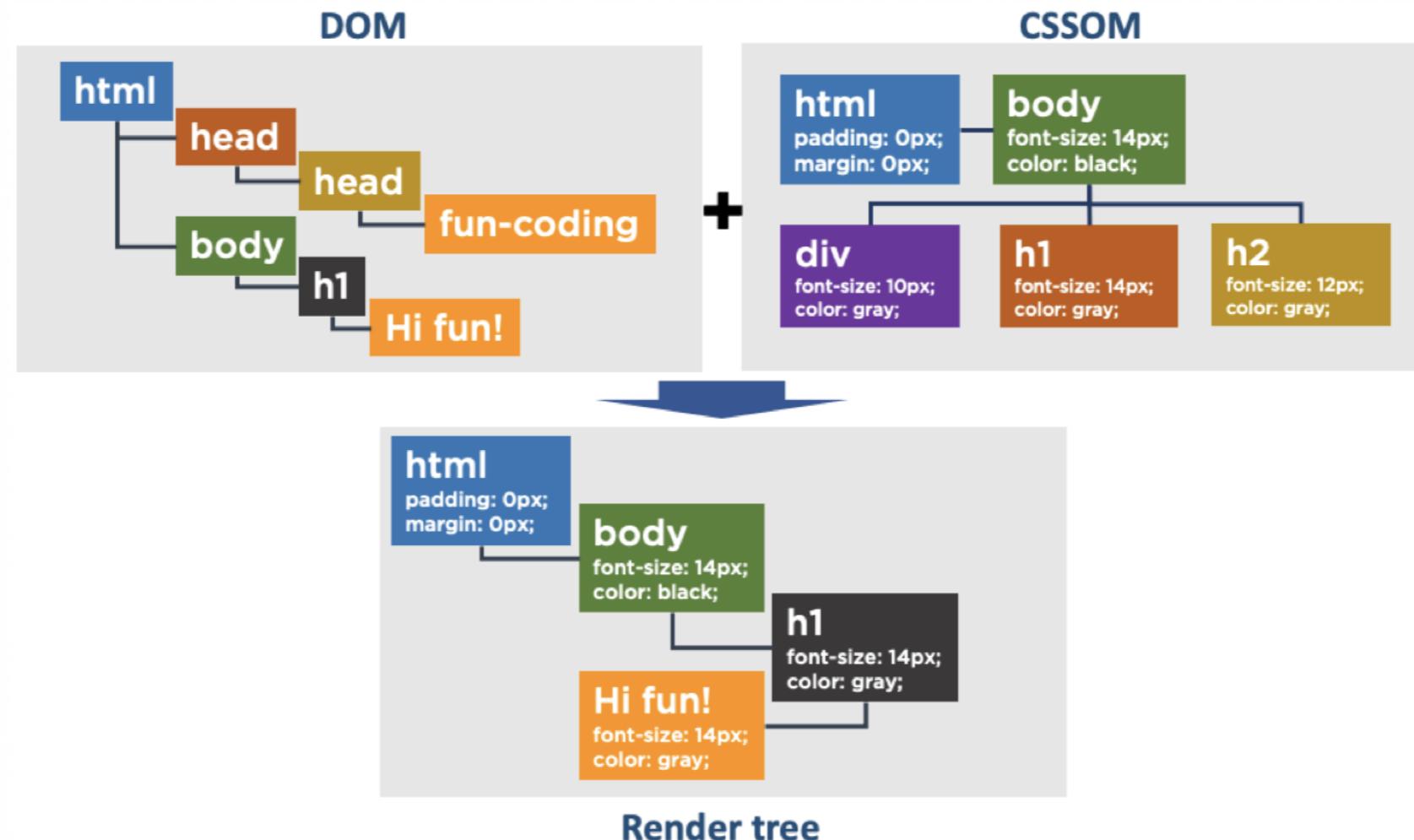
참고: DOM(Document Object Model)

- Document: 구조화된 데이터
- DOM: 구조화된 문서를 객체로 표현하는 형식



웹브라우저 동작 과정3

- HTML 파일 파싱하여 DOM tree 생성
- DOM/CSSOM tree 기반 Render tree 생성



웹브라우저 동작 과정4

- Render tree 기반, Rendering



참고: DOM 과 Virtual DOM

- javascript등으로 웹페이지 변경시, 매번 DOM tree 변경 ~ Rendering
- DOM의 가벼운 형태(json)를 메모리에 올려놓고, 필요한 부분만 변경 후, 한번에 변경된 부분만 DOM에 반영하여 조정
 - React, Vue.js 등 프론트엔드 기술에서 활용 중



