

MOD - WEB STUDY

0 WEEKS

[CONTENTS]

0. 컴퓨터 공학 기초 용어 정리

1. 인터넷, 웹, 브라우저의 정의

2. 웹의 동작 원리 : 서버와 클라이언트

3. 소프트웨어 제품 개발 과정

4. 웹 개발 포지션 종류 (프론트엔드와 백엔드)

0. 컴퓨터 공학 기초 용어 정리

- **컴퓨터의 구성 요소** : CPU + 기억장치[RAM(메모리) / HDD, SSD] + 입출력 장치 + 시스템 버스
- **프로그램** : 어떤 문제를 해결하기 위해 컴퓨터에게 주어지는 처리 방법과 순서를 기술한 일련의 **명령문의 집합체**
- **OS** : 어떤 프로그램을 사용했을 때 그것을 작동시키는 환경
데스크탑 : Mac OS, Window OS, Ubuntu OS / 모바일 : IOS, 안드로이드
- **Process (프로세스)** : OS가 CPU와 메모리 자원을 활용해서 실행시킨 프로그램
- **Thread (스레드)** : Process 내부에서 작동하게 되는 실행 단위.
- **Kernel (커널)** : OS의 핵심 부분으로 자원 관리와 메모리 제어 등을 담당.

프로그래밍 언어 (저급 언어와 고급 언어)

컴퓨터는 0과 1로 표현된 **기계어**라는 특별한 언어로 이해한다.

컴퓨터가 이해할 수 있는 언어인 기계어와 어셈블리어를 “**저급 언어**”라고 한다.

또한, 사람이 쉽게 칠 수 있는 프로그래밍 언어를 “**고급 언어**”라고 한다.

하지만 결국 프로그램이 돌아가기 위해선 기계어로 번역되어야 컴퓨터가 읽을 수 있다.

이렇게 고급언어에서 저급 언어로 변환되는 것을 “**컴파일(compile)**”한다고 한다.

비슷한 용어로 “**빌드(build)**”한다라고도 한다.

컴파일 언어와 인터프리터 언어

컴파일 방식에 따라 크게 “**컴파일(compile) 언어**”와 “**인터프리터(interpreter) 언어**”로 나뉜다.

컴파일 언어로 작성한 코드들은 한 번에 몽땅 기계어로 컴파일한 후에 실행 파일을 만든다.
대표적인 언어로 C 언어, Go 언어, Rust 등이 있다.

반면 **인터프리터 언어**로 작성된 코드들은 모두 한 번에 컴파일하는 것이 아니라 실행하고 나서 코드 **한 줄씩** 컴파일을 시키면서 실시간으로 돌아가게 된다.

한 줄씩 컴파일이 되면서 Bytecode(바이트 코드)라는 형태로 나오게 되고 그 Bytecode를 기계어로 번역하거나 가상 환경에서 직접 돌리게 된다.
대표적인 언어로 Python, Javascript 등이 있다.

개발 관련 기초 용어 정리

- **레거시 코드** : 실제로 에러가 나는 코드는 아니지만 트렌드를 따라가지 못하는 코드
- **리팩토링** : 레거시 코드를 현재 트렌드와 조직의 바뀐 개발 스타일에 맞게 수정하는 것
- **디버깅** : 코드에서 오류를 찾는 과정
- **라이브러리** : 특정 목적을 위해 개발자들이 미리 정리해 놓은 코드 뭉치.
예를 들어 파이썬의 numpy, 텐서플로우, JS의 React 등이 있다.
- **프레임워크** : 특정 프로그램을 개발하기 위한 여러 요소들과 메뉴얼을 제공하는 코드 뭉치.
예를 들어 파이썬의 웹 개발을 위한 프레임워크인 Flask, Django 등이 있다.
- **애플리케이션 (Application = APP)** : 기능이 있는 프로그램 (응용 프로그램)
흔히 말하는 앱 스토어에서 다운 받은 어플, 앱은 정확하게 모바일 애플리케이션이다. 넷플릭스, 인스타그램과 같은 프로그램을 웹 애플리케이션이라고 한다.
- **API (Application Programming Interface)** : 애플리케이션에서 사용할 수 있도록, 운영 체제나 프로그래밍 언어가 제공하는 기능을 제어할 수 있게 만든 인터페이스를 뜻한다.

소프트웨어란?? (Software??)

- 컴퓨터 = 하드웨어 + 소프트웨어
- 하드웨어란 ??? -> 딱딱한 것
- 하드웨어의 종류 : 데스크탑, 노트북, 패드, 스마트폰, 스마트 워치 등
- 하드웨어만 있는 컴퓨터는 고철덩어리에 불과!
- 소프트웨어란 ??? -> 하드웨어에게 명령을 하는 역할을 한다.
- 소프트웨어의 종류 : 웹, 어플(모바일), 게임, 버스 정류장 안내판 (임베디드) 등

1. 인터넷, 웹, 브라우저의 정의

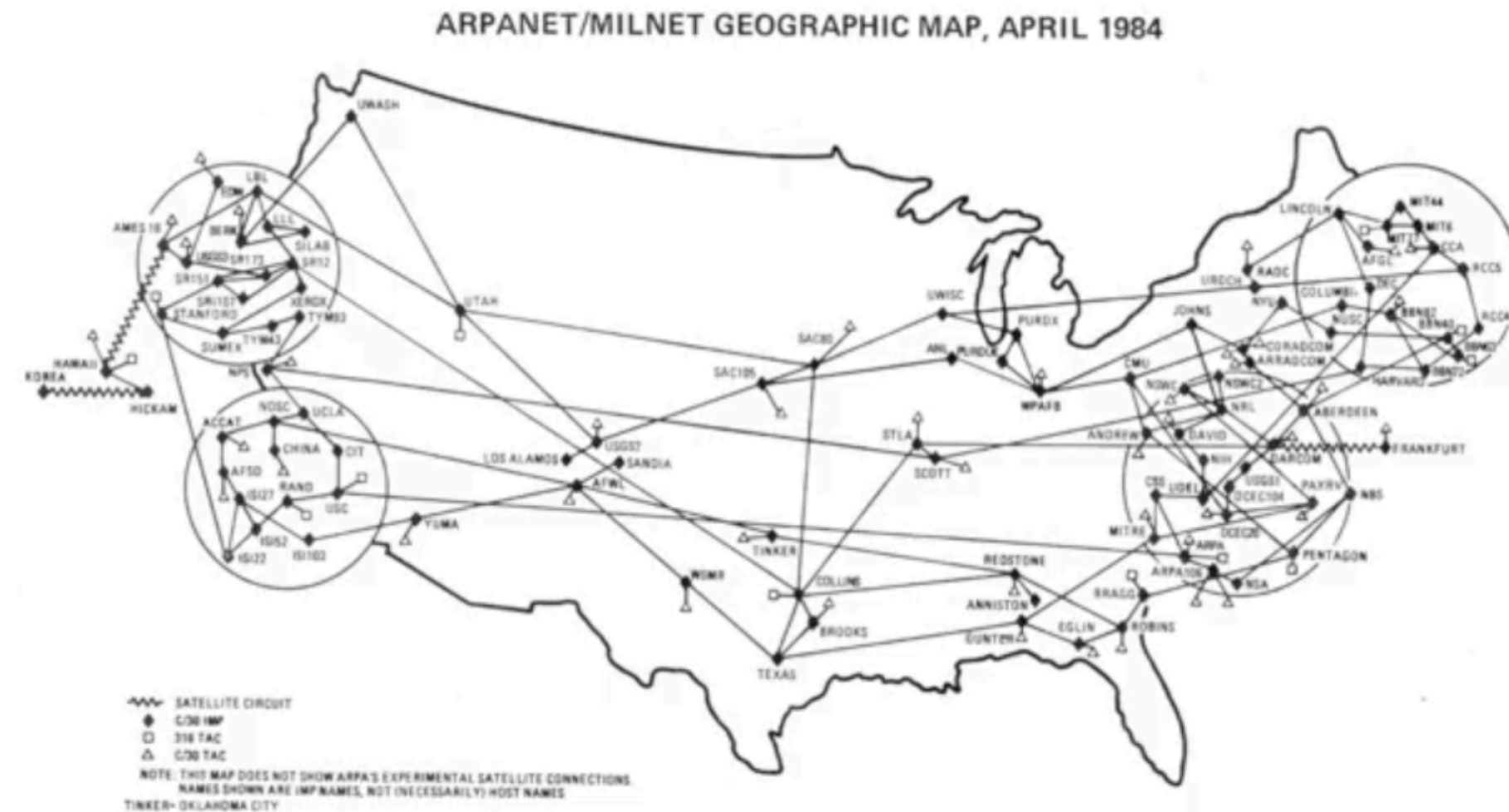
인터넷(Internet)의 정의

인터넷이란 전 세계 컴퓨터들을 하나로 연결하는 거대한 컴퓨터 통신망을 의미한다.

[인터넷의 역사] 인터넷의 시작 ARPANET

오래 전부터 많은 과학자들이 정보가 연결되어 있는 세계를 상상하였다.

냉전 시대인 1960년대, 미국 국방부 주도 하에 계획된 ARPANET 프로젝트에서 ARPANET이라는 분산 네트워크 개발.



[인터넷의 역사] ARPANET, 그 이후...

군대에서 시작되어 대학, 기업으로 퍼져나갔으나 소수 엘리트만 사용할 뿐이었다.

초기 인터넷은 파일과 자원 공유, 원거리에 있는 컴퓨터에 로그인하려는 목적 밖에 없었다.

30년이 지난 후...

일반인에게도 PC(Personal Computer)가 보급되기 시작.

[웹의 역사] 월드 와이드 웹 (www) 발명

유럽입자물리연구소 소속 연구원 팀 버너스 리가 연구를 효과적으로 공유하기 위해 고안하였다.

1. 한 문서에서 다른 문서로 접근한다는 개념인 하이퍼텍스트를 적용한 **HTML** 발명
2. 문서에 주소를 붙이는 **URL**(Uniform/Universal Resource Locator) 발명
3. 컴퓨터와 컴퓨터 간에 메시지를 주고 받을 수 있는 통신 규약인 **HTTP** 발명

-> 전 세계 인터넷 콘텐츠를 연결해 공유할 수 있도록 하는 소프트웨어 WWW (World Wide Web) 발명

웹(web)의 정의

월드 와이드 웹(World Wide Web)이란

인터넷에 연결된 사용자들이 서로의 정보를 공유할 수 있는 공간을 의미한다.

간단히 줄여서 WWW나 W3라고도 부르며, 일반적으로 웹(Web)이라고 한다.

웹은 인터넷상의 인기 있는 하나의 서비스일 뿐이다.

하지만 현재에는 인터넷과 웹이라는 단어가 혼용될 만큼 인터넷의 가장 큰 부분을 차지하고 있다.

웹 브라우저의 정의와 종류

웹 브라우저란 인터넷상에서 웹에 **연결**시켜 주는 **소프트웨어**이다.

사파리, 크롬, 인터넷 익스플로어, 엣지, 파이어 폭스, 오페라, ...

최초의 웹

<http://info.cern.ch/>

웹의 진화

1. 웹 하이퍼텍스트 문서

기본적인 서식과 하이퍼링크를 내재한 텍스트 문서. (HTML)

2. 웹 페이지

이미지, 오디오 등 리소스를 제공하고, 풍부하고 화려한 레이아웃 내재화 시각적으로는 아름답지만 사용자와의 상호작용이 불가능해서, 마치 종이에 인쇄된 페이지와 같음. (HTML + CSS + JS)

3. 웹 애플리케이션 (넷플릭스, 인스타그램, 쇼핑몰, ...)

사용자와 상호작용이 가능하고 복잡한 처리 또한 가능한 웹 브라우저에서의 프로그램.

(HTML + CSS + JS + REACT + PYTHON + ...)

[POINT]

- 1. 인터넷의 정의와 이해**
- 2. 웹 브라우저의 정의와 종류**
- 3. 웹 (WWW)의 정의와 이해**

네트워크와 인터넷 기초 이론 관련 책 추천

도서 : 손으로 익히며 배우는 네트워크 첫걸음 (Gene 지음, 한빛미디어) **강추

2. 웹의 동작 원리 : 서버와 클라이언트

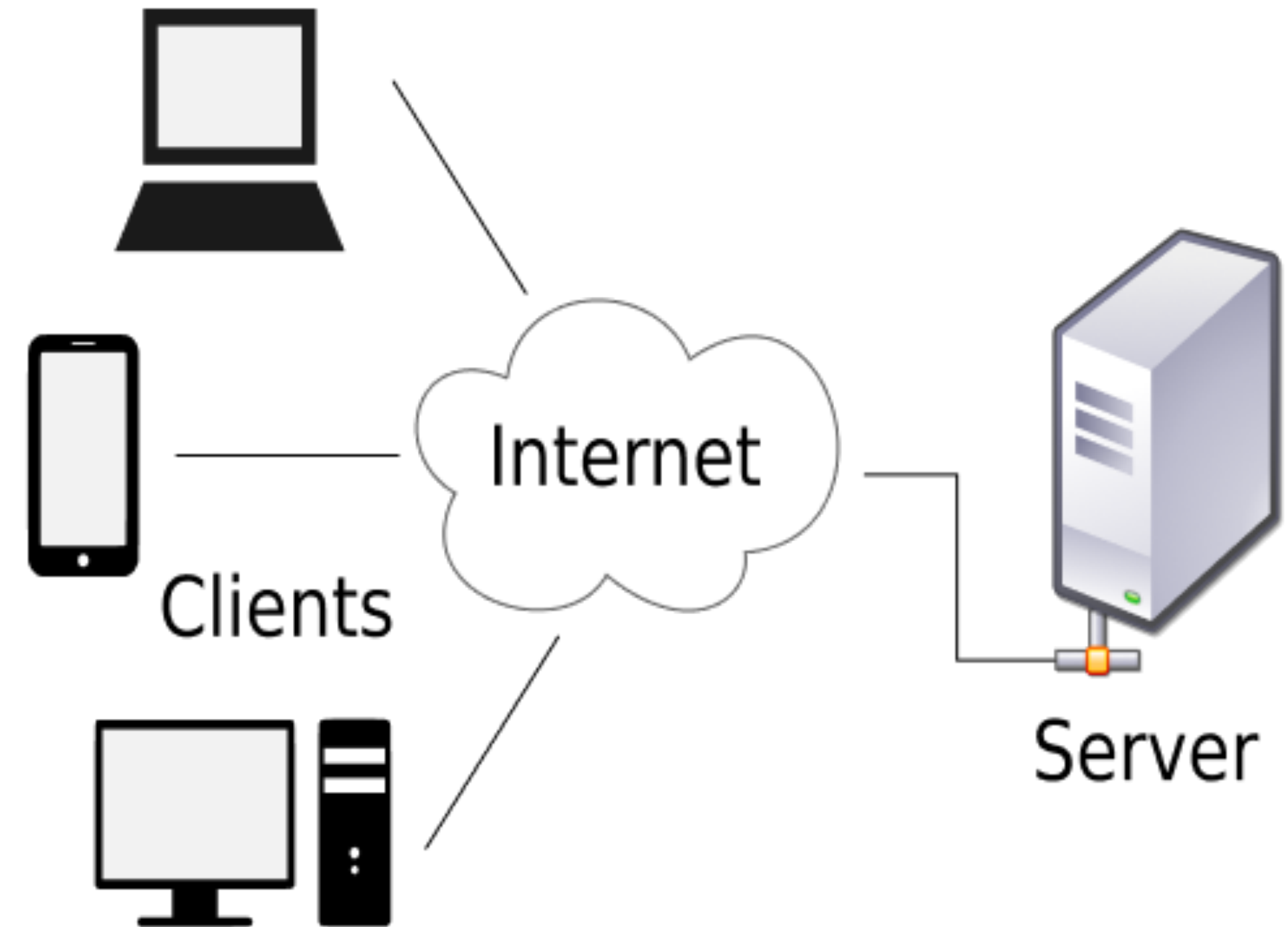
클라이언트 <-> 서버

[클라이언트]

인터넷으로 연결된 서버로부터 정보를 제공받는 컴퓨터

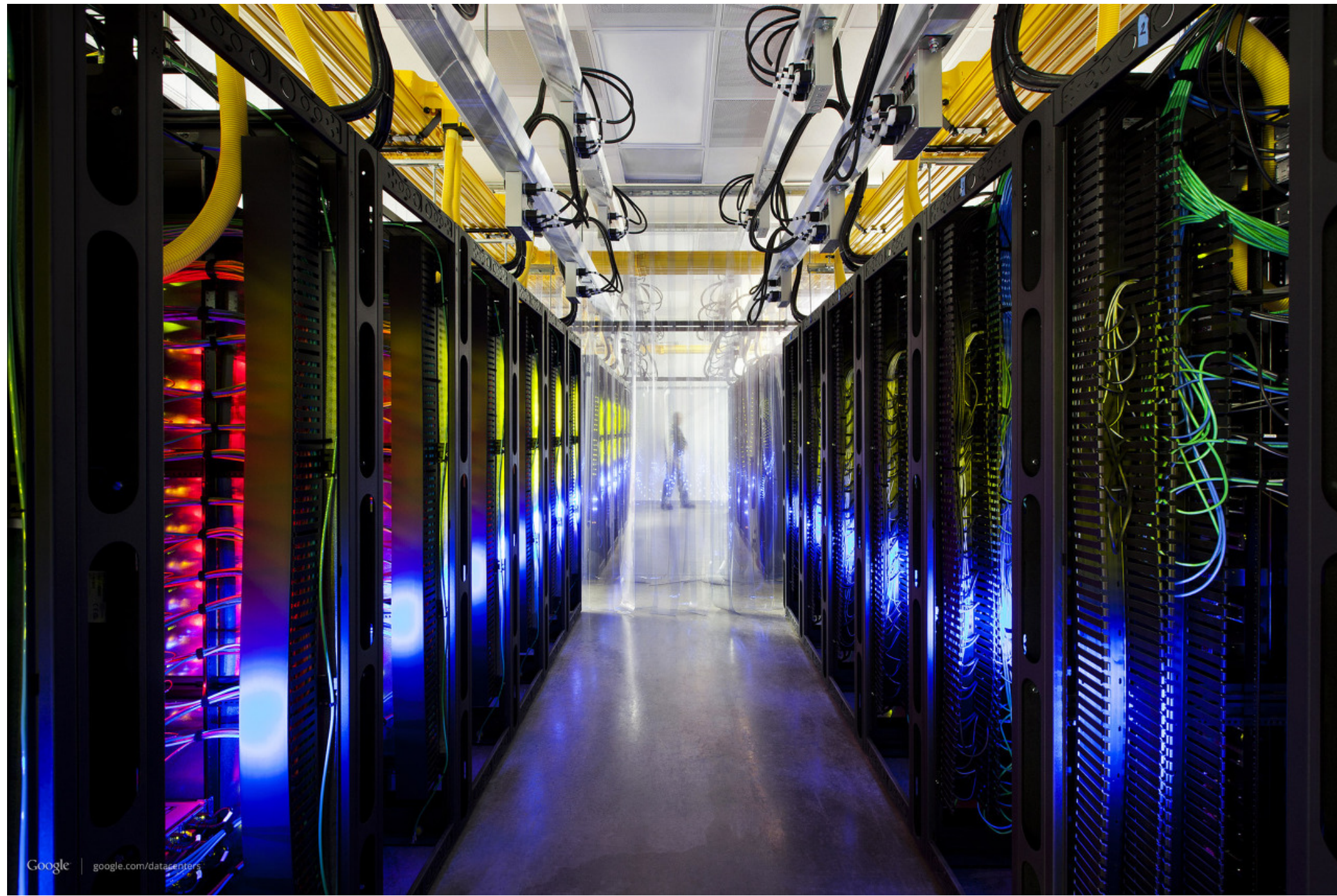
[서버]

인터넷으로 연결된 클라이언트에게 정보를 제공하는 컴퓨터



출처 : 위키백과

서버 컴퓨터 예시

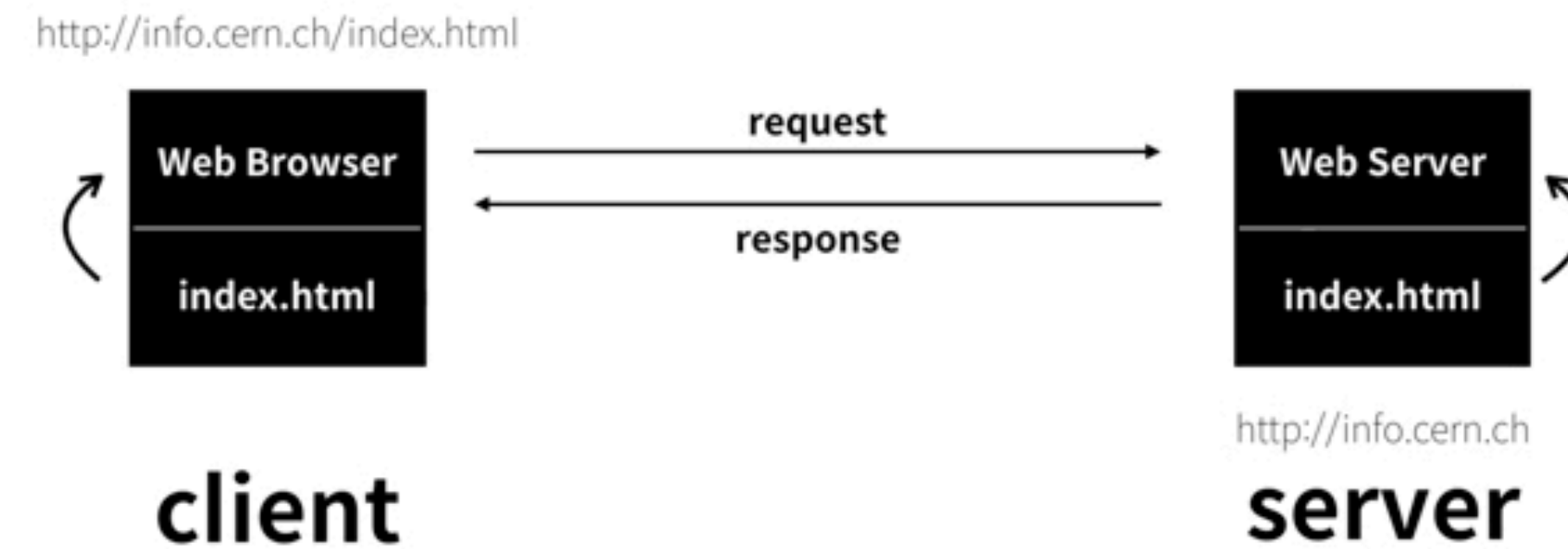


구글 서버 컴퓨터



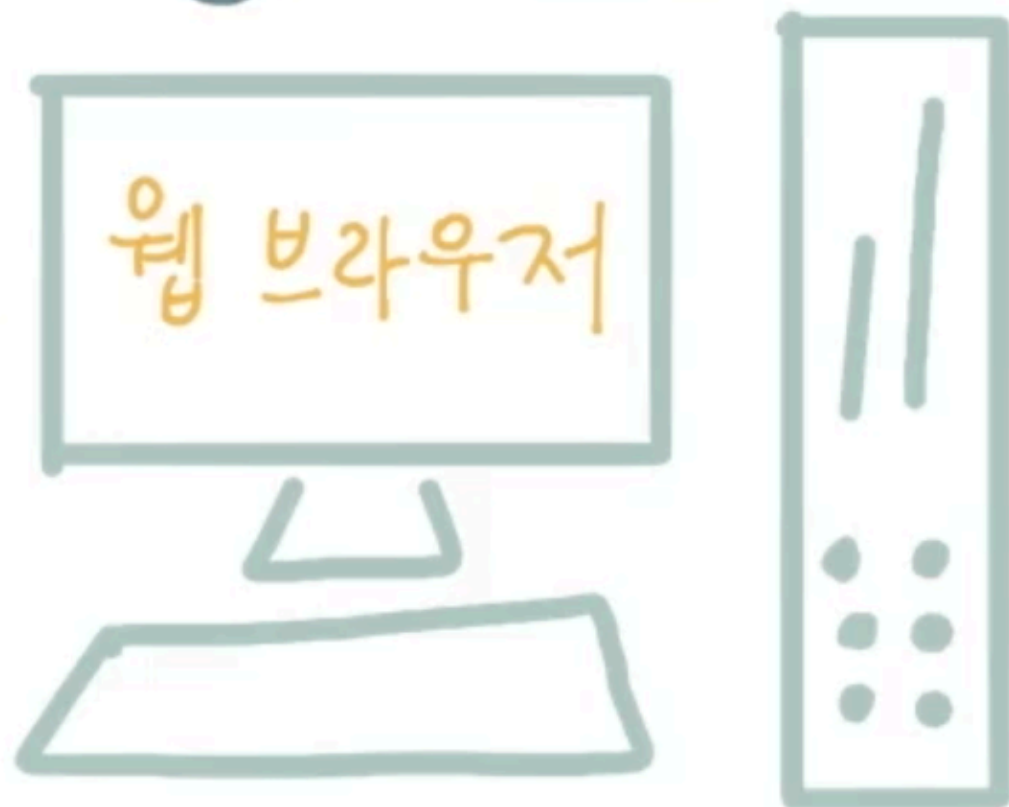
네이버 서버 컴퓨터

[서버 <-> 클라이언트] 보충 외부 온라인 강의



<https://youtu.be/yBPyzaccbkc>

클라이언트



요청



응답



서버

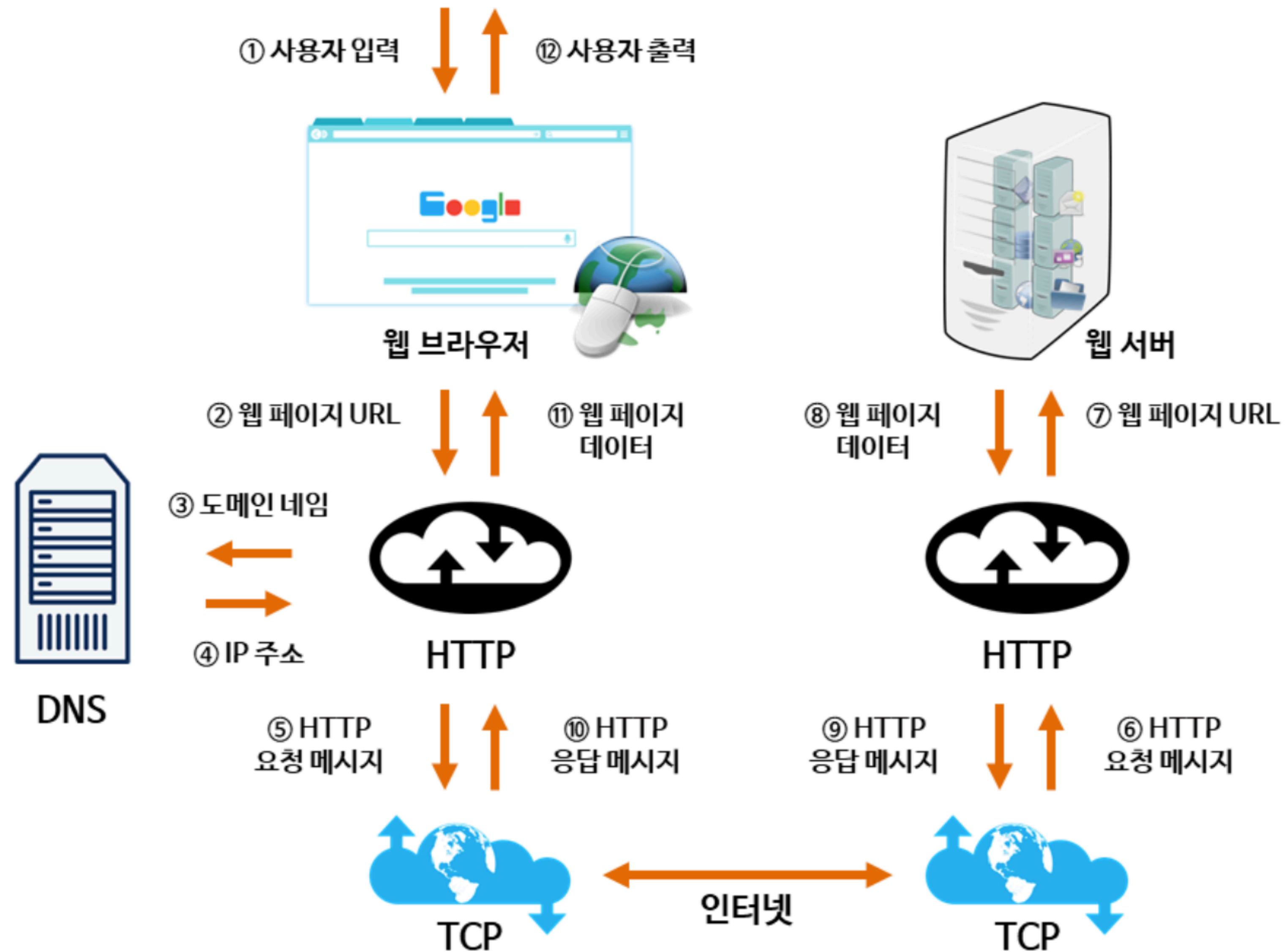


index.html

정보 통신 기초 용어 정리

- **호스팅** : 서버 컴퓨터를 임대해주는 서비스.
- **IP 주소** : 인터넷상에 있는 컴퓨터의 고유한 주소. ex) 122.13.193.617 (<https://ip.pe.kr>)
IP 주소를 통해 인터넷상의 한 컴퓨터에서 다른 컴퓨터로 데이터를 주고 받을 수 있다.
- **도메인** : 인터넷 사이트 주소.
숫자로만 구성된 아이피(IP) 주소의 단점을 보완하기 위해 사용한다.
- **DNS** : 인터넷 도메인 이름들의 위치를 알아내기 위한 IP 주소로 바꾸어주는 시스템.
- **URL** : 인터넷에서 어느 사이트에 접속하기 위해서 입력해야 하는 주소를 포함한 일련의 문자.
맨 앞에 `http://`를 입력하고 다음에 해당 사이트의 주소를 표시한다.
- **프로토콜** : 복수의 컴퓨터 사이에서 데이터 통신을 원활하게 하기 위해 필요한 **통신 규약**.
- **HTTP (HyperText Transfer Protocol)** : 웹 상에서 정보를 주고받을 수 있는 프로토콜.
주로 HTML 문서를 주고받는 데에 쓰인다. HTTPS는 HTTP에서 보안이 강화된 것이다.
- **TCP** : 데이터 전송을 제어하는 프로토콜

💻 인터넷과 웹의 전체적인 동작 원리 💻



①② 사용자가 웹 브라우저를 통해 찾고 싶은 웹 페이지의 URL 주소를 입력함.

③ 사용자가 입력한 URL 주소 중에서 도메인 네임(domain name) 부분을 DNS 서버에서 검색함.

④ DNS 서버에서 해당 도메인 네임에 해당하는 IP 주소를 찾아 사용자가 입력한 URL 정보와 함께 전달함.

⑤⑥ 웹 페이지 URL 정보와 전달받은 IP 주소는 HTTP 프로토콜을 사용하여 HTTP 요청 메시지를 생성함.

이렇게 생성된 HTTP 요청 메시지는 TCP 프로토콜을 사용하여 인터넷을 거쳐 해당 IP 주소의 컴퓨터로 전송됨.

⑦ 이렇게 도착한 HTTP 요청 메시지는 HTTP 프로토콜을 사용하여 웹 페이지 URL 정보로 변환됨.

⑧ 웹 서버는 도착한 웹 페이지 URL 정보에 해당하는 데이터를 검색함.

⑨⑩ 검색된 웹 페이지 데이터는 또 다시 HTTP 프로토콜을 사용하여 HTTP 응답 메시지를 생성함.

이렇게 생성된 HTTP 응답 메시지는 TCP 프로토콜을 사용하여 인터넷을 거쳐 원래 컴퓨터로 전송됨.

⑪ 도착한 HTTP 응답 메시지는 HTTP 프로토콜을 사용하여 웹 페이지 데이터로 변환됨.

⑫ 변환된 웹 페이지 데이터는 웹 브라우저에 의해 출력되어 사용자가 볼 수 있게 됨.

3. 소프트웨어 제품 개발 과정

소프트웨어 개발

개발자

기획자

디자이너

소프트웨어 설계 순서

1. 기획 요구사항 정의(기획자)
2. 요구사항에 따른 개발 견적 완성(개발자)
3. 요구사항 명세서 & 스토리보드 작성(디자이너, 기획자)
4. 클래스 & 함수 설계(개발자)
5. 스토리보드 최종 수정 & 디자인 파일 첨부(기획자, 디자이너, 개발자)
6. 개발 1차 완성(개발자)
7. 무한 검토(개발자, 기획자, 디자이너)
8. 배포

웹 개발

1. 기획자의 요구사항 명세서와 디자이너의 스토리보드를 기반으로 필요한 기술 스택 정의
2. 일반적으로 FrontEnd BackEnd, DevOps로 나누어서 개발 진행
 - FrontEnd 개발자 - 디자인을 화면 프론트에 구현 + API 핸들링
 - BackEnd 개발자 - API 개발 + DB 핸들링
 - DevOps 개발자 - 소프트웨어의 개발과 운영 : 배포 & 유지보수 & 업데이트

웹 개발 - FrontEnd Developer

- 웹 디자인과 UI / UX 디자인을 실제 웹에 구현하는 역할이다.
- HTML, CSS, JS 를 기본으로 추가적인 라이브러리 또는 프레임워크를 사용하여 웹을 구현한다.
- CSS 라이브러리 또는 프레임워크 : TailwindCSS, SASS, ...
- JS 라이브러리 또는 프레임워크 : React, Vue, Angular, ...

웹 개발 - BackEnd Developer

- 클라이언트에 필요한 데이터를 핸들링하며 필요한 API를 개발한다.
- 개발 언어 & 프레임워크 - nodeJS(express), Java(spring), Python(Django, Flask), Go(echo), ...
- API - Restfull API, Server API (GraphQL), ...
- 데이터베이스 - MySQL, MongoDB, Firebase, MariaDB, ...

웹 개발 - 배포

- Heroku 배포
- AWS 배포
- Firebase 배포
- Github Io 배포

[참고] Python(Django) + React 웹 풀스택 개발 로드맵

1. 인터넷과 웹에 대한 이해, HTML, CSS 이해
2. Python, Django 이해
3. HTML + CSS + Django 웹 풀스택 개발
4. JS, React 이해
5. React + Django Restfull API 풀스택 웹 SPA 애플리케이션 개발
6. Docker 이해, 인프라 관리

웹 개발 공부 추천 사이트

- <http://www.tcpschool.com/webbasic/intro> : 인터넷과 웹의 기초 이론 정리
- http://it.chosun.com/site/data/html_dir/2020/09/11/2020091100487.html :
프론트엔드와 백엔드 용어 정리
- <https://opentutorials.org/course/2039> : 무료 HTML, CSS 강의

스터디 진행 계획 (4주 + 2주 + 4주 + 2주 + a = 3개월 + a) 주 1회 2 ~ 3 시간

[0주] : OT & 소프트웨어의 이해와 웹, 어플 개발의 이해

[4주] : 웹 프론트엔드 개발 입문 HTML, CSS 이해

[2주] : 자신만의 간단한 웹 사이트 만들기 실습 + 코드 리뷰

[4주] : 웹 백엔드 서버 개발 입문 Python 활용

[2주] : 서버 구축하기 (로그인 기능, 머신러닝, 지도 기능, 등등) 실습 + 코드 리뷰

[a] : 팀으로 협업 경험하기 (실제 서비스 만들고 출시 및 배포 하기)

[~] : 현업에서 실제 서비스 되는 인스타그램, 핀터레스트, 망고 플레이트 등 개발 맛보기, 어플 개발 맛보기

[0주차] : OT & 스터디 진행 방향 정하기 & 소프트웨어의 이해와 웹, 어플 개발의 이해

[1주차] : 개발 환경 구축 + Git + HTML 완성 (프론트엔드 개발 시작)

[2주차] : HTML + CSS

[3주차] : HTML + CSS 완성

[4, 5주차] : 정적 웹 사이트 개발 (개인) 코드 리뷰

[6주차] : 서버 사이드 언어 Python 복습 & 서버 개발 맛보기 (백엔드 개발 시작)

[7주차] : 데이터 CRUD 기능 만들기

[8주차] : 로그인 기능 만들기 (쿠키와 세션에 대한 이해)

[9주차] : 투두리스트 할일 목록 웹 애플리케이션 만들기 (HTML + CSS + PYTHON)

[10, 11주차] : Full Stack 웹 사이트 개발 (개인) 코드 리뷰

[12주차 ~] : 팀 프로젝트로 실제 웹 서비스 개발 후 제품 출시 및 배포

[~] : 어플 개발 맛보기 + 실제 현업에서 요구하는 개발 기술 맛보기 (인스타그램, 넷플릭스, 핀터레스트, 망고 플레이트 등 개발)

[Javascript (JS)] + [React 기초] 스터디 추가 개설

- JS의 기초 문법을 정리하고 외부 라이브러리를 사용하지 않고 순수 JS를 이용하여 간단한 웹 게임을 개발합니다.

1. 공식 문서 기반으로 **가장 최신 문법 정리**
2. **기념일 디데이** 웹 애플리케이션 개발
3. **숫자 야구** 웹 게임 만들기
4. **할일 목록** 웹 애플리케이션 개발 (DB 핸들링)
5. **영화 정보** 웹 애플리케이션 만들기 (API 핸들링)

- 현재 프론트엔드 개발 산업에서 가장 많이 쓰이고 유명한 JS의 라이브러리인 React의 기초를 공부합니다.

1. 공식 문서 기반으로 **React의 기초 정리**
2. React로 **영화 정보 웹 SPA** 개발
3. React-Native로 **할일 목록 어플** 만들기

- **HTML**에 대하여 기초 수준으로 알고 있어야 합니다.
- **CSS**에 대하여 입문 수준으로 알고 있어야 합니다.
- **다른 프로그래밍 언어 경험**이 조금이라도 있어야 합니다.
- 개발을 진로로 정하신 분들을 환영합니다.

사용할 IDE (코드 편집기)

[0주차] : <https://www.repl.it>

[1주차 ~] : VS Code

과제

1. 이론 복습

2. 파이썬 객체 지향 프로그래밍 기본 문법 공부 (2월초까지)

3. 다음주에 이론 관련해서 간단한 시험이 있습니다.
(구글 폼으로 진행합니다.)

4. VS Code 설치하기

