

Day 1 - Internet and Web Service

<https://ryuzypjproject.tistory.com/51>

네트워크:

- 네트워크(network)는 여러 대의 컴퓨터와 장치들이 서로 데이터를 주고받을 수 있도록 연결된 구조를 말합니다.
- 각 장치는 고유한 주소(IP 주소)를 가지고 서로를 식별하고 통신합니다.

IP주소

- IP주소(Internet Protocol Address)는 인터넷이나 네트워크에 연결된 각 장치를 구분하기 위해 부여되는 고유한 번호입니다.
- IPv4는 예를 들어 192.168.0.1처럼 점(.)으로 구분된 네 부분으로 구성되고, IPv6는 더 많은 장치를 구분할 수 있도록 더 긴 형태를 사용합니다.
- 웹사이트에 접속할 때 실제로는 이 IP주소를 통해 해당 서버와 연결되는 것입니다

IPv4

- IPv4 주소는 숫자 4개가 점(.)으로 구분된 형태입니다. 각 숫자는 0부터 255까지 가능하며 총 4개 = 8비트 × 4 = 32비트로 구성되어 있습니다. 즉, IPv4는 32비트 주소체계이며, 이론상 약 43억 개(2^{32})의 주소를 만들 수 있습니다.

IP 주소 클래스

클래스	시작 IP	끝 IP	용도 / 설명
A	0.0.0.0	127.255.255.255	매우 큰 네트워크용 (정부, 대형 기업 등)
B	128.0.0.0	191.255.255.255	중간 규모의 네트워크용
C	192.0.0.0	223.255.255.255	소규모 네트워크용 (일반 기업, 가정 등)
D	224.0.0.0	239.255.255.255	멀티캐스트 용도
E	240.0.0.0	255.255.255.255	실험/연구용, 예약됨 (일반 사용 금지)

사설 IP 주소 (Private IP)

- 인터넷에 직접 연결되지 않고 내부 네트워크(예: 가정, 사무실)에서 사용하는 주소입니다. 이 주소들은 공유기나 사무실 LAN에서 사용되고, 외부에서는 직접 접근할 수 없습니다.

클래스	사설 IP 범위	예시
A	10.0.0.0 ~ 10.255.255.255	10.1.2.3
B	172.16.0.0 ~ 172.31.255.255	172.20.10.5
C	192.168.0.0 ~ 192.168.255.255	192.168.0.1

특수한 IP 주소

주소	의미
127.0.0.1	루프백 주소 (자기 자신을 가리킴)
0.0.0.0	모든 IP를 의미하거나 아직 주소 없음
255.255.255.255	브로드캐스트 주소 (모든 장치에게 전송)

IPv6

- IPv6는 16비트 × 8블록 = 총 128비트 주소입니다. 숫자와 영문자가 16진수로 표현되며, 콜론(:) 으로 구분됩니다. 너무 길기 때문에 0은 생략할 수 있습니다.

2001:0db8:85a3:0000:0000:8a2e:0370:7334

비트와 바이트

- 비트(bit)는 컴퓨터가 이해할 수 있는 가장 작은 정보 단위입니다. 비트는 오직 두 가지 값만 표현할 수 있습니다. 0과 1의 조합으로 모든 정보를 만들어내는 게 바로 컴퓨터입니다.
 - 1바이트(Byte) = 8비트(bit)
 - 즉, 8개의 0과 1이 모이면 1바이트가 됩니다.
 - 예: 1바이트 = 01001010 ← 이런 식의 8자리 0과 1

IP주소에서 비트

IPv4는 총 32비트 주소입니다. 8비트씩 4부분으로 나누어서 10진수로 변환한 것이 우리가 보는 IPv4 주소입니다.

예: 11000000.10101000.00000001.00000001 → 192.168.1.1

IPv6는 무려 128개의 0과 1로 구성된 주소입니다. 너무 길기 때문에, 16비트씩 8덩어리로 나누고, 16진수로 바꿔서 표현합니다.

예: 2001:0db8:85a3:0000:0000:8a2e:0370:7334

2. 인터넷

- 인터넷은 단순한 선이나 전파가 아니라, 전 세계 수많은 장치들이 규칙(프로토콜)에 따라 서로 데이터를 주고받는 방식으로 작동하며, 정보를 빠르게 주고받을 수 있게 해주는 정보 고속도로라고 할 수 있습니다.

프로토콜

- 프로토콜(protocol)은 컴퓨터나 네트워크 장치들이 서로 통신할 때 지켜야 할 약속이나 규칙입니다.
- 우리가 웹사이트를 볼 때 사용하는 HTTP, 파일을 전송할 때 사용하는 FTP, 이메일을 주고받는 SMTP 같은 것들이 모두 프로토콜입니다.

HTTP (HyperText Transfer Protocol)

- 웹사이트를 주고받을 때 사용하는 약속(프로토콜)

FTP (File Transfer Protocol)

- 파일을 주고받기 위한 규칙 (파일 업로드, 다운로드)

SMTP (Simple Mail Transfer Protocol)

- SMTP는 이메일을 보낼 때 사용하는 규칙입니다.

3. 웹

- 웹(Web)은 우리가 인터넷을 통해 정보를 보고, 읽고, 상호작용할 수 있게 해주는 서비스입니다.
- 웹은 HTML로 만들어진 문서(웹 페이지)를 브라우저(크롬, 사파리 등)가 읽어서 화면에 보여주는 방식으로 작동합니다.
- 이 웹은 인터넷 위에서 돌아가는 서비스 중 하나로, 우리가 주소창에 URL을 입력하면 HTTP 같은 프로토콜을 통해 서버에 요청하고, 그 결과를 받아와서 보여주는 구조입니다. 쉽게 말해, 웹은 인터넷 위에 펼쳐진 거대한 정보 공간입니다.

웹의 작동 방식:

1. 사용자가 웹 브라우저에 주소(URL)를 입력합니다.

- `https://www.example.com`

2. DNS 서버가 주소를 IP 주소로 바꿔줍니다.

- 컴퓨터는 사람이 쓰는 도메인(예: `www.example.com`)을 이해하지 못합니다.
- 그래서 DNS(Domain Name System)가 나서서 이 주소를 IP 주소로 바꿔줘요.
- DNS 서버는 전 세계에 퍼져있다.

3. 브라우저가 서버에 요청을 보냅니다. (HTTP 요청)

- 이제 브라우저는 해당 IP 주소를 가진 서버에게 "웹 페이지 보여줘!" 라는 메시지를 보냅니다.
- 이 메시지는 HTTP 요청(Request) 이라고 부릅니다.
- 요청은 웹서버에 도착합니다.

4. 서버가 요청을 받고, 웹 페이지 파일을 응답합니다. (HTTP 응답)

- 서버는 요청을 받으면, 내부에 저장된 HTML, CSS, JavaScript 등의 파일을 찾아서 브라우저에게 보냅니다.
- 이걸 HTTP 응답(Response) 이라고 합니다.
 - **HTML: 구조 / CSS: 디자인 / JS: 동작** 이 세 가지를 함께 보내줍니다.

5. 브라우저가 파일을 해석해서 화면에 보여준다

- 브라우저는 서버로부터 받은 HTML, CSS, JS 파일을 읽고 해석해서 웹페이지처럼 화면에 그려줍니다.
- 사용자는 웹사이트를 눈으로 보고, 클릭하고, 입력할 수 있는 상태가 됩니다.

DNS (Domain Name System)

- DNS는 우리가 웹사이트 주소를 입력할 때 사용하는 도메인 이름(예: www.example.com)을 컴퓨터가 이해할 수 있는 IP 주소(예: 93.184.216.34)로 바꿔주는 시스템입니다.

Server 와 Client

- 서버(Server)와 클라이언트(Client)는 인터넷에서 정보를 주고받는 관계를 설명하는 기본 개념입니다.
- 클라이언트는 정보를 요청하는 쪽, 예를 들어 우리가 사용하는 웹 브라우저나 스마트폰 앱이 이에 해당하고
- 서버는 그 요청을 받아 정보나 서비스를 제공하는 쪽, 즉 웹사이트의 본체나 데이터가 저장된 컴퓨터입니다.

4. 웹 서비스

- 웹 서비스란 인터넷을 통해 클라이언트(사용자)와 서버 간에 데이터를 주고받으며 기능이나 정보를 제공하는 소프트웨어 시스템입니다.
- 사용자는 웹 브라우저나 애플리케이션을 통해 웹 서버에 요청을 보내고, 서버는 요청에 맞는 데이터를 처리하여 결과를 응답합니다.
- 웹 서비스는 HTML 페이지 제공, 파일 업로드, API 서비스 등 다양한 형태로 제공되며, HTTP 프로토콜을 기반으로 작동합니다.

🐍 파이썬의 웹 서비스란?

- 파이썬에서 웹 서비스를 개발한다는 것은, 파이썬 코드로 사용자의 요청(예: 게시글 보기, 로그인 시도 등)을 처리하는 웹 애플리케이션을 만들고, 이를 웹을 통해 사용할 수 있도록 서버를 실행하는 것을 의미합니다.
- 파이썬으로 웹 서비스를 만들 때 가장 많이 사용하는 도구는 웹 프레임워크입니다. 대표적으로는 다음과 같은 것들이 있습니다:

웹 프레임워크:

- 웹을 만들기 위해 이미 만들어진 틀 (모듈)
- 모듈, 라이브러리 : 무언가를 만들기 위한 도구

1. Flask (플라스크) – 가볍고 배우기 쉬운 웹 프레임워크

- 마치 "공부용 혹은 소규모 프로젝트"에 적합한 도구입니다.
- 설치도 간단하고, 코드를 빠르게 작성해볼 수 있어 초보자에게 좋습니다.

2. Django (장고) – 대형 서비스도 가능한 풀스택 프레임워크

- 사용자 인증, 데이터베이스 연동, 관리자 페이지 등 다양한 기능이 기본 내장되어 있습니다.
- 더 구조적이고 큰 프로젝트에 적합합니다.

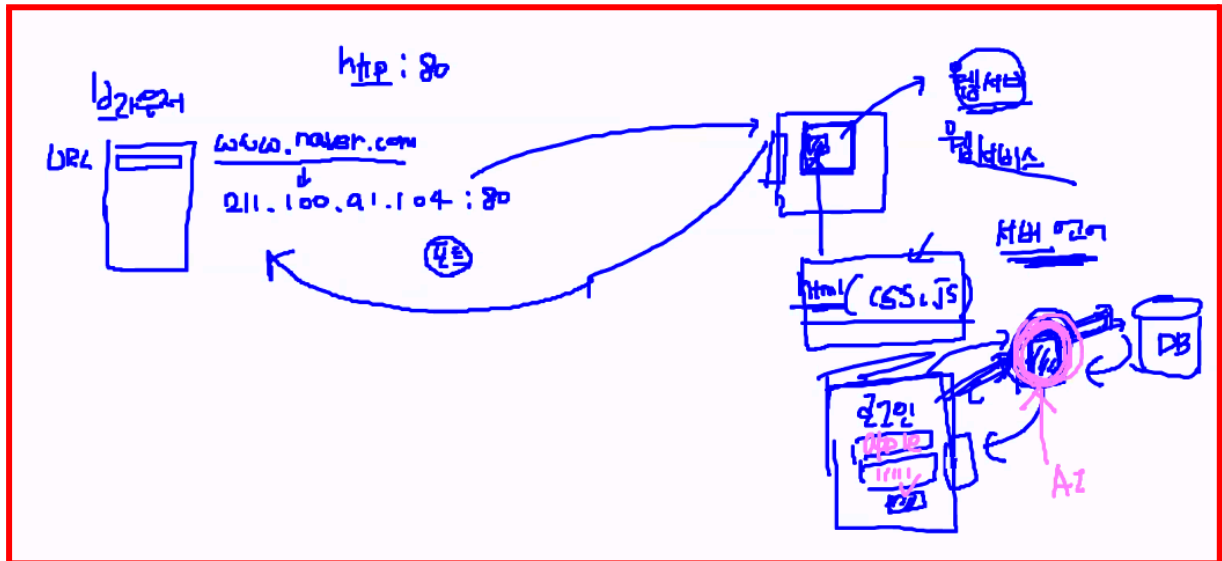
3. FastAPI – 빠르고 현대적인 비동기 웹 프레임워크

- 비동기 지원 (async/await 사용): 고성능 서버를 쉽게 작성할 수 있어, 많은 요청을 빠르게 처리할 수 있습니다.
- 자동 API 문서 생성: 개발자가 API 문서를 따로 만들 필요 없이 /docs에 접속하면 Swagger UI로 자동 생성된 문서를 확인할 수 있습니다.
- 타입 힌트를 통한 유효성 검사: Python의 타입 힌트를 이용해 입력값을 자동으로 검증해주고, 오류 메시지를 깔끔하게 제공합니다.

동기 vs 비동기

1. 동기: 프로그램이 돌아가다 많은 연산 처리가 필요할때 (루핑 등) 순서대로 연산처리를 하고 다음 task로 넘기는 처리
2. 비동기 : 많은 연산 처리가 필요할때 맨 나중에 처리

웹 서비스 작동 방식:



1. Client (request) → Server
2. Server (response) → Client

Web + LLM 프로젝트:

- 프론트엔드, 기본적인 백엔드(로그인 화면 등)는 바이트코딩 작성
- 웹 프레임워크: FastAPI 사용
- DB
- Version Control Tool : Git