

HTTP Server

통신

서로 다른 프로그램간의 데이터 교환

A라는 프로그램이 B라는 프로그램에게 어떤 데이터를 요청하면 B가 그에 대한 결과를 응답한다.

이 때 A를 클라이언트, B를 서버라고 한다.

웹 환경에서의 통신

웹 브라우저가 웹 사이트가 구동되고 있는 웹 서버에게 URL 형식으로 특정 파일의 내용을 요청한다.

웹 서버는 자신이 요청받은 페이지의 HTML 코드를 응답 데이터로 전송한다.

- 웹 브라우저 = 클라이언트
- 웹 서버 = 서버

프론트엔드

클라이언트 안에서 구동되는 프로그램

엔드유저(최종사용자)에게 직접적으로 눈으로 보고, 조작할 수 있는 환경을 제공.

- 웹 페이지 형식: HTML+CSS+JS
- 안드로이드(JAVA), 아이폰 앱(Swift)

백엔드

서버 안에서 구동되는 프로그램

전통적인 웹 프로그래밍의 영역

- ASP(VB) -> ASP.NET (C#)
 - 점유율이 다소 희박함.
- PHP(C와 비슷한 자체 언어) -> Codeigniter, Laravel
 - 저렴한 환경에서 구동 가능. 소규모 웹사이트,쇼핑몰들이 활발히 활용중
- JSP(JAVA) -> Spring -> 전자정부표준프레임워크
 - 정부과제용~
 - 다른 웹 프로그래밍 환경과 비교해서 대규모 시스템을 구축하는데 가장 적합.

새로 주목 받는 영역

- Node.js(Javascript) -> Express.js(Javascript) -> Nest.js(Typescript)
 - 새로운 언어를 공부하는 부담은 없음
- Django(Python)
 - Python을 공부해야 함. 말도 안되게 쉽다.
- Flask(Python)
 - 완전 사기~!!

| 공통적으로 쉽고, 빠르게 개발이 가능하지만 대규모 시스템을 건디기에는 다소 무리가 있다.

| 그래서 주로 서비스 초창기에 Node.js, Django등으로 구축을 하고 이용자가 늘어나기 시작하면 Java로 재구축을 하는 추세임.

| ex) Netflix