

01

Web Application Overview

PNU Mini Bootcamp – Front End

software language ranking

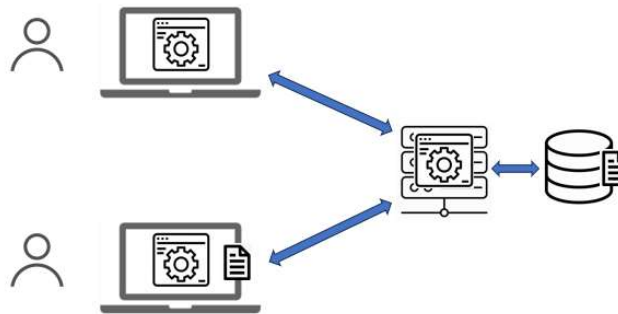
Worldwide, Sept 2023 :

Rank	Change	Language	Share	1-year trend
1		Python	27.99 %	+0.1 %
2		Java	15.9 %	-1.1 %
3		JavaScript	9.36 %	-0.1 %
4		C#	6.67 %	-0.4 %
5		C/C++	6.54 %	+0.3 %
6		PHP	4.91 %	-0.4 %
7		R	4.4 %	+0.2 %
8		TypeScript	3.04 %	+0.2 %
9	↑↑	Swift	2.64 %	+0.6 %
10		Objective-C	2.15 %	+0.1 %
11	↑↑↑	Rust	2.12 %	+0.5 %
12	↓↓↓	Go	2.0 %	-0.1 %
13	↓	Kotlin	1.78 %	-0.0 %
14		Matlab	1.58 %	+0.1 %
15		Ruby	1.05 %	-0.1 %
16	↑↑↑↑	Ada	1.02 %	+0.3 %
17	↑	Dart	0.95 %	+0.1 %
18	↓	Powershell	0.94 %	+0.0 %

웹 개발이란

서버 vs 클라이언트

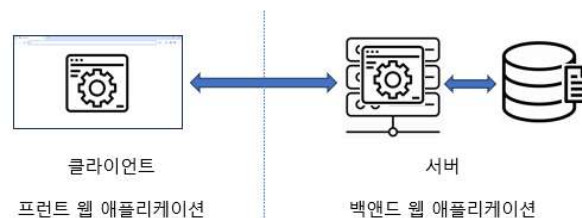
- 클라이언트 - 서버 구조란 클라이언트 애플리케이션과 서버 애플리케이션이 상호 네트워킹을 통해 업무를 진행하는 구조
- 네트워킹을 위한 커넥션을 요청하는 애플리케이션이 클라이언트 애플리케이션
- 대기하고 있다가 커넥션 요청이 들어오면 커넥션을 완성하는 애플리케이션이 서버 애플리케이션
- 사용자끼리 데이터를 공유해야 하는 경우
- 서버 데이터베이스 데이터를 사용자가 이용해야 하는 경우



웹 개발이란

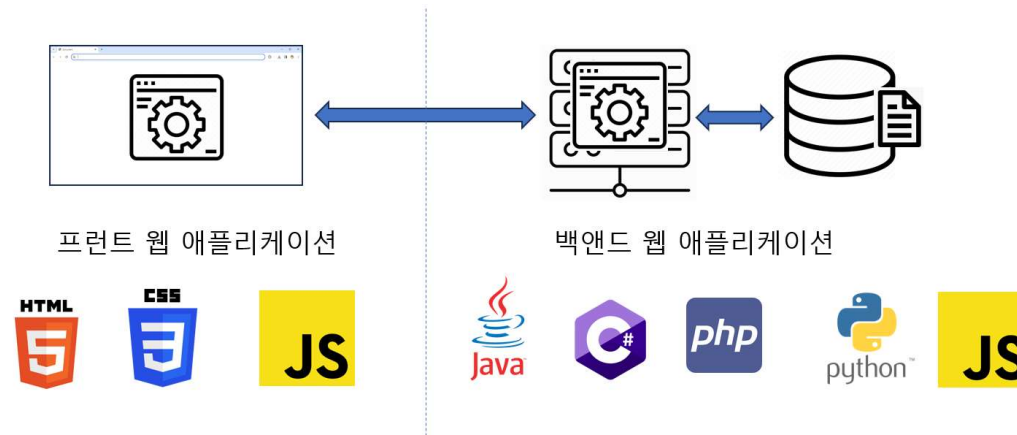
프론트엔드 웹 애플리케이션 vs 백엔드 웹 애플리케이션

- 클라이언트 - 서버 애플리케이션은 범용적인 용어
- 카카오톡, 게임 애플리케이션등
- 클라이언트 - 서버 구조에서 애플리케이션들이 웹의 프로토콜인 HTTP 프로토콜을 이용해 네트워킹을 한다면 이를 웹 애플리케이션
- 웹 애플리케이션 중 서버에서 실행되는 애플리케이션을 백엔드 웹 애플리케이션
- 사용자 브라우저에서 실행되는 애플리케이션을 프론트 웹 애플리케이션



웹 개발, 어디서부터 시작할까

웹 개발을 위해 공부해야 할 기술



웹 개발, 어디서부터 시작할까

프론트 웹 애플리케이션 개발 기술 – HTML, CSS, JavaScript



웹 문서의 뼈대를 만드는 **HTML**

- 웹 브라우저 창에 웹 문서의 내용을 보여주기 위한 약속
- HTML에서 약속한 표기법을 사용해서 문서 작성해야 함



웹 문서를 꾸미는 **CSS**

- 웹 문서를 꾸미거나 웹 요소를 적절하게 배치하는 방법
- 다양한 디바이스에 맞는 반응형 웹 디자인을 만들기 위해 필수적으로 학습해야 함



사용자 동작에 반응하는 **자바스크립트**

- 사용자 동작에 반응해서 동적인 효과를 만들기 위한 기술

웹 개발, 어디서부터 시작할까

프론트 웹 애플리케이션 개발 기술 - 자바스크립트 프레임워크

Angular



- 구글에서 만든 프레임워크로 현재의 자바스크립트 프레임워크의 기본 사상인 SPA 개념을 만든 프레임워크

React.js



- 페이스북에서 프론트 웹 애플리케이션을 개발하기 위해 만든 프레임워크
- 선언형 UI 프로그래밍, 가상 DOM, 상태 관리등의 많은 개념이 React.js 부터 시작

Vue.js



- 구글의 개발자였던 Evan You 에 의해 만들어진 프레임워크로 Angular, React.js 와 거의 유사한 개념 및 구조를 지원하는 프레임워크

웹 개발, 어디서부터 시작할까

백엔드 웹 애플리케이션 개발 기술

자바 기반의 백엔드 웹 애플리케이션 개발



자바스크립트 기반의 백엔드 웹 애플리케이션 개발



C# 기반의 백엔드 웹 애플리케이션 개발



파이썬 기반의 백엔드 웹 애플리케이션 개발

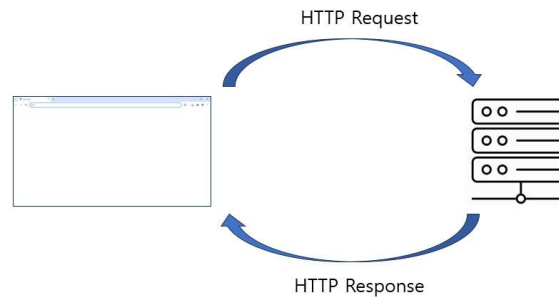


PHP 기반의 백엔드 웹 애플리케이션 개발



웹 기본 지식 - HTTP 프로토콜

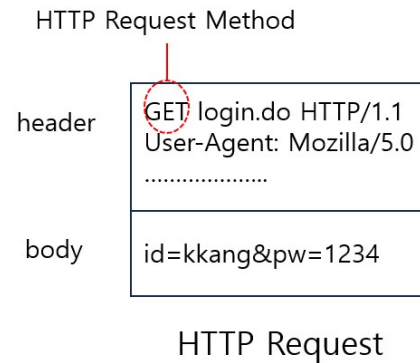
- 클라이언트 - 서버 구조에서 상호 네트워킹을 하며 데이터를 주고 받는데 이때 HTTP 프로토콜을 이용하는 애플리케이션을 웹 애플리케이션
- 프로토콜은 통신 규약으로 HTTP 프로토콜은 HTTP Request 규약과 HTTP Response 규약으로 나누어 집니다.



웹 기본 지식 - HTTP 프로토콜

HTTP Request

- HTTP Request 는 프론트 애플리케이션이 백엔드에 전달하는 데이터에 대한 규약



- HTTP Request 의 header 에 다양한 정보가 설정되어 백엔드에 전달되는데 URL 정보와 User-Agent 정보등이 담겨 있습니다.
- User-Agent 는 요청하는 프론트에 대한 정보로 브라우저명, 브라우저 버전등에 대한 정보가 담겨 있습니다.

웹 기본 지식 – HTTP 프로토콜

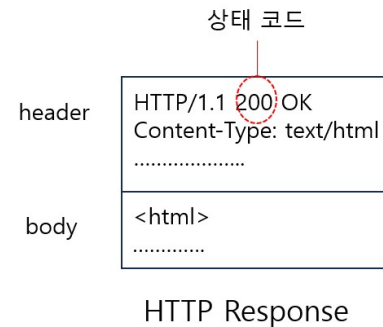
HTTP Request

- HTTP Request header 에 설정되는 정보 중 HTTP Request Method 는 흔히 요청 방식이라고 하며 GET, POST, PUT, DELETE, HEAD, PATCH 가 있습니다.
- GET : 서버의 데이터를 얻기 위한 요청
- POST : 서버에 데이터를 전달하여 저장하기 위한 요청
- PUT : 서버에 데이터를 전달하여 수정하기 위한 요청
- DELETE : 서버의 데이터를 삭제하기 위한 요청
- HEAD : 서버의 각종 정보를 확인하기 위한 요청
- PATCH : 서버의 데이터를 수정하기 위한 요청
- put – 전체, patch – 일부
- head – response header 만 요청, 서버의 상태 파악 목적

웹 기본 지식 - HTTP 프로토콜

HTTP Response

- 백엔드에서 프론트에 전달되는 데이터를 HTTP Response



- body 가 실제 백엔드에서 프론트에 전달하는 데이터입니다. 웹임으로 HTML 문서가 body 에 담겨서 전달 될 수도 있고 JSON 같은 데이터 일 수도 있습니다.

웹 기본 지식 – HTTP 프로토콜

HTTP Response

- 상태 코드는 숫자로 표현되며 프런트의 요청을 어떻게 처리한 것인지에 대한 상태를 표현하는 숫자입니다.
- 2xx : 요청을 정상적으로 처리 했음
- 3xx : 요청 완료를 위해 추가적인 작업이 필요함
- 4xx : 요청 잘못으로 요청을 처리하지 못함
- 5xx : 서버 잘못으로 요청을 처리하지 못함

웹 기본 지식 - URL 분석

- 프론트가 백엔드에 요청하는 업무 혹은 데이터를 통칭해서 리소스라고 합니다.
- 프론트에서 원하는 리소스가 식별될 수 있게 URL 을 구성해서 백엔드에 요청을 하고, 백엔드는 요청 정보의 URL 을 분석해서 어떤 리소스를 프론트에게 서비스 할것인지를 판단하게 됩니다.
- URL 은 Uniform Resource Locator 의 약어

<프로토콜>://<호스트>:<포트>/<경로>?<쿼리>#<해시>



웹 기본 지식 – URL 분석

프로토콜

- 프로토콜은 네트워킹에 사용할 통신 규약으로 웹에서는 일반적으로 HTTP 혹은 HTTPS 가 사용됩니다

호스트

- 호스트는 흔히 도메인이라 불리우는 부분으로 요청하는 서버의 네트워크 상에서의 주소
- 포트는 하나의 서버에서 네트워킹하는 애플리케이션을 식별하기 위한 숫자

경로

- 경로는 서버 애플리케이션의 리소스 식별

웹 기본 지식 – URL 분석

쿼리 문자열

- 쿼리 문자열은 프론트에서 백엔드에 요청을 하면서 전달하는 데이터

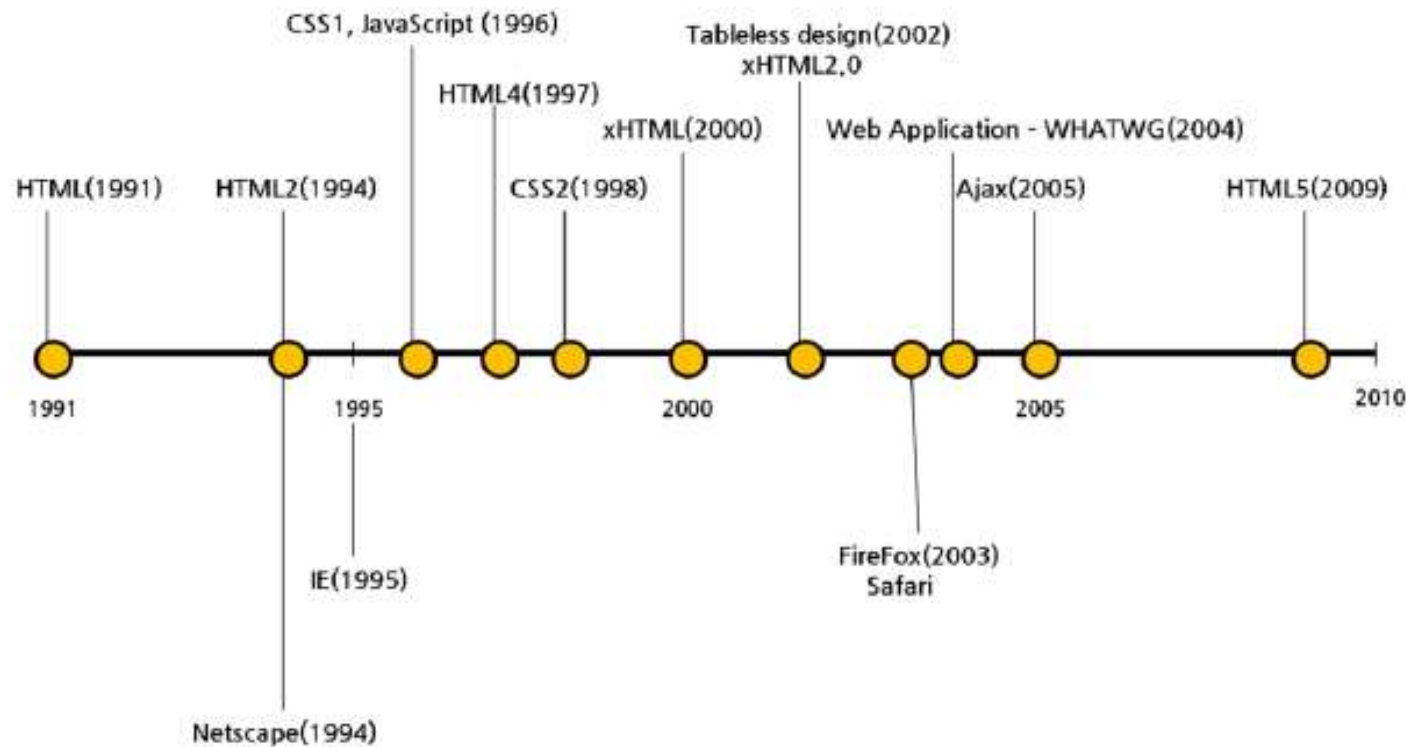


해시문자열

- 해시문자열은 # 뒤에 추가되는 문자열
- 프론트 웹 애플리케이션에서 현재의 애플리케이션의 상태를 URL 에 추가하기 위해서 사용

HTML Timeline

► HTML Timeline



HTML Timeline

HTML의 한계

•HTML4

- 1990년대 초부터 소개되어 진화를 거듭하던 표준안은 1997년 HTML4를 마지막으로 더 이상의 버전업이 없었다.
- 1990년대의 웹 환경에 초점을 맞추어 지다보니 브라우저 환경에서의 UI 표현에만 치중, 기능적 구현 한계

•Vendor 중심의 기술발전

- Active X Control
- Script Language
- Flash, Flex, SilverLight

• Web2.0 과 Web Application

- Web의 UI 표현 중심에 한계, Application 급의 다양한 기술에 대한 요구
- Ajax의 등장
- Vendor 중심의 기술 등장 – Flex, SilverLight
- 다양한 브라우저의 등장 – Chrome, Safari, Opera, FireFox

HTML Timeline

- **새로운 웹 표준에 대한 필요성**

- WHATWG 주도로 새로운 Web 표준 책정 작업 시작
- WHATWG 의 작업을 W3C 에서 이어받아 HTML5 로 명명
- Mobile 환경의 급격한 변화로 인한 HTML5 필요성 급부상

- **HTML5 의 목적**

- 특정 Vendor 독점 기술에 의존하지 않는 공개되고 표준화된 웹 환경을 만들기 위한 기술
- add-on 또는 plug-in 없이도 웹을 사용 가능
- Application 수준의 다양한 기능구현 기능

HTML Timeline

- **Web Application 을 위한 Javascript API**

- 비디오/오디오 재생을 위한 요소(video/audio)
- 2D, 3D 그래픽 처리(canvas)
- 오프라인에서도 작동되는 애플리케이션(애플리케이션 캐시)
- 클라이언트 측에 데이터 저장(Web Storage, Web SQL Database)
- 백그라운드 처리 수행(Web Workers)
- 서버로부터의 데이터 푸시나 서버와의 양방향 통신 수행(Server-Sent Events, Web Sockets)
- 로컬 파일의 내용을 읽어들이(File API)

- **Semantic Markup**

- 문서 구조의 의미나 문서 안에 삽입된 데이터의 의미 등을 명확히 하기 위한 사양이 다수 포함되어 Semantic 웹의 구현
- <header>, <section>, <footer>, <article>, <nav> 등

- **CSS3**

- 다양한 CSS 기법
- CSS3 Media Query, CSS3 Animation



감사합니다

단단히 마음먹고 떠난 사람은
산꼭대기에 도착할 수 있다.
산은 올라가는 사람에게만 정복된다.



윌리엄 셰익스피어
William Shakespeare