웹프로그래밍 기술 소개

2019-08-28

이승진

## 1980년대

|  |  |
| --- | --- |
| 프로그래밍 언어 | 기술 |
| C | CGI |

### C + CGI

장점: 최초의 웹프로그래밍 기술

단점: 웹프로그래밍을 하기에는 C 언어가 너무 불편.

C 언어는 성능은 좋지만, 단순한 작업을 구현하기에는 불편. (구현작업 생산성이 낮다)

CGI 기술의 너무 떨어지는 성능.

## 1990년대

|  |  |
| --- | --- |
| 프로그래밍 언어 | 기술 |
| VBscript | ASP |
| Java | Servlet |
| PHP | PHP |
| Java | JSP |

### VBscript + ASP

장점: VBscript 언어가 단순해서 대부분의 웹프로그래밍 작업에 편함. (구현작업 생산성이 좋다)

html 태그 사이에 VBscript 코드를 추가하는 형태의 구현이라서 직관적임.

단점: VBscript 언어가 너무 단순해서, 소수 고급 기능 구현을 못함. 그래서 C로 구현한 ActiveX가 필요함.

그래서 요즘은 사용 안함.

### Java + Servlet

장점: Java 언어. (성능도 나쁘지 않고, 단순한 작업을 구현하기에 불편하지도 않고, 고급 기능 구현 가능하고)

단점: 모든 부분을 Java로 구현해야해서, 직관적이지 않다 (어렵다. 작업 생산성이 낮다)

### PHP

충분히 강력하고, 충분히 생산성이 좋고.

html 태그 사이에 php 코드를 추가하는 형태의 구현이라서 직관적임.

요즘도 널리 사용되는 대표적인 웹프로그래밍 기술 중 하나

### Java + JSP

장점: Java 언어. (성능도 나쁘지 않고, 단순한 작업을 구현하기에 불편하지도 않고, 고급 기능 구현 가능하고)

html 태그 사이에 Java 코드를 추가하는 형태의 구현이라서 직관적임. (Java + Servlet의 단점을 해결)

## 2000년대

|  |  |
| --- | --- |
| 프로그래밍 언어 | 기술 |
| Java | Spring Web MVC |

대형 정보시스템, 대형 쇼핑몰, 인터넷 뱅킹 구현에는 Java 언어가 가장 적절하다.

이 분야에서 PHP 기술이나 JSP 기술은 매우 부족하다.

거대한 정보시스템을 구현하려면,

html 태그 사이에 코드를 끼워 넣는 형태의 구현이 아니고,

높은 수준의 객체지향 설계 및 구현이 요구되기 때문이다.

Java 언어와 Spring 기술의 조합이, 유시보수성과 재사용서이 가장 좋은 조합이다.

그래서 정보시스템 개발 언어로 자바가 압도적인 1위이다.

C#, Kotlin, Swift 등 새 객체지향 언어들이 나왔지만,

정보시스템 개발 분야에서 자바의 1등은 당분간(앞으로 10년?) 계속될 것으로 전망.

## 2010년대

|  |  |  |
| --- | --- | --- |
| 프로그래밍 언어 | 프론트엔드 기술 | 서버 기술 |
| Javascript | React.js Angular Vue.js | node.js + Expresss.js |

웹인지 앱인지 구별이 안될 정도의 화려면 UI를 구현하려면

Javascript 언어로 구현하는 프론트엔드 기술이 반드시 필요하다.

Javascript 프론트엔드 개발은 요즘 굉장히 핫하다.

프론트엔드 개발에서 Javascript 외에는 선택지가 없다.

javascript + node.js 서버 기술의 장점

현재 완벽한 비동기(asynchronous) 서버 프로그래밍은 javascript + node.js 만 가능

(java의 비동기 기술은 아직 완벽하지 않고 발전 중임)

비동기 서버가 동기 서버보다 훨씬 성능이 좋아서, 비동기 프로그래밍 기술

node.js의 서버 기술의 단점

대형 정보시스템을 구현하기에는, Java + Spring 보다 훨씬 못하다. (유지보수성이 낮다)

대형 정보시스템 개발의 정답은 Java + Spring

## 요약

웹사이트 개발 => PHP

(예: 홈페이지, 블로그, 동호회, 작은 쇼핑몰 등)

정보시스템 개발 =>

Javascript 프론트엔드 기술 + Java와 Spring 백엔드 기술

(예: 학사 시스템, 도서관 시스템, 대형 쇼핑몰, 인터넷 뱅킹)

빠르게 개발해야 하는 분야에서는,

어차피 Javascript 프론트엔드 기술은 꼭 필요하므로

서버를 Java Spring으로 따로 구현하지 않고,

javascript node.js로 개발하는 경우가 많음.

(예: 벤처 창업 -> 빨리 개발해서 오픈 해야 함)

javascript로 프론트엔드 백엔드 두 다 구현하는 것을 full stack 개발이라고 부른다.

## 결론

(1) 웹 개발자가 되고 싶다 -> PHP (웹디자인도 잘해야함)

(2) 정보시스템 서버 개발자가 되고 싶다 -> Java, Spring, Database

(3) 프론트엔드 개발자가 되고 싶다 -> Javascript, 프론트엔드 프레임웍 (웹디자인도 잘해야함)

(4) 풀스택 개발자가 되고 싶다 =>

Javascript, 프론트엔드 프레임웍

node.js 서버

(5) 정보시스템 풀스택 개발자가 되고 싶다 =>

Javascript, 프론트엔드 프레임웍

Java Spring 서버