스프링과 노드 차이

개요

- 스프링과 노드에 대한 특징과 차이점 비교
- 노드와 스프링과 관련된 자바와 JS 부터 알아가 보자

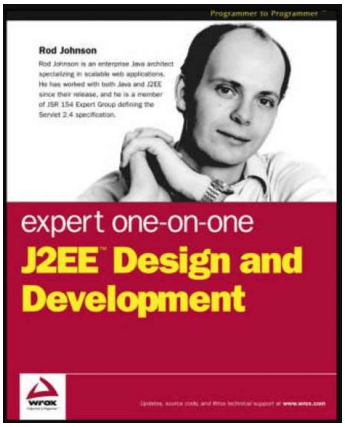
자바

자바의 등장

- 가전제품 내에 탑재해 동작하는 프로그램을 위해 제임스 고슬링 등이 만든 객체지향적 프로그래밍 언어
- 플랫폼에 독립적
    - 바이트 코드를 **JVM**을 통해 **어느 플랫폼**에서나 동일하게 실행 가능
- 그렇다면 플랫폼에 종속적인 특징은 무엇일까?
    - 기계어가 CPU마다 다름
    - OS마다 API가 다름
    - OS마다 실행파일의 형식이 다름
- C++로 작성한 프로그램을 돌리기 위해선 C++ 컴파일러가 CPU마다 필요하다.

- 월드 와이드 웹(WWW) 출현으로 자바 급부상
    - 웹의 등장으로 인터넷이 다양한 컴퓨터 시스템에 동작해야 할 필요성의 증가
    - 자바의 특징 중 하나인 어느 곳이든 이식 가능하다는 점이 큰 강점으로 부각
    - 즉, 인터넷을 통한 소프트웨어 배포에 적합한 언어
    - 가전 제품을 위한 언어에서 벗어나 웹 어플리케이션 개발에 주요한 언어로 자리 잡게 되는 전환점

스프링



- 2002년 로드 존슨이 집필한 책이 출간
- EJB가 없어도 고품질의 확장 가능한 애플리케이션 개발이 가능하다라는 것을 증명
- 스프링의 뼈대가 된 소스 코드
- 2003년 6월에 Apache 라이센스 2.0 공개
  - 공짜고 이걸로 돈을 벌어도 되지만, 위험성에 대한 책임은 사용자가 져라
- 스프링의 주요 특징
  - POJO
  - AOP
  - 의존성 주입
  - IoC
  - PSA

JS

JS의 등장

- 웹 브라우저에서 정적인 컨텐츠를 움직이게 만들어주는 목적으로 만들어진 프로그래밍 언어

JS의 단점

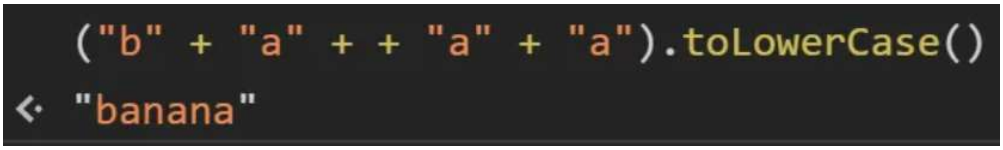
```
1 + 2
// 3

"1" + "2"
//12

102 - 2
// 100

"102" - "2"
// 100 ('+'는 문자열 연결인데, '-'는 숫자가 계산이된다..?)
```

- 일관성이 없는 암시적 형 변환



https://medium.com/naverfinancial/node-js-vs-java-spring-c4699565918e

- 애초에 이상한 언어지 않나..

```
function plus(number1, number2)
{
    return number1 + number2;
}

var result = plus('1', '2') // 3...! 다행이 이건 의도대로 작동합니다.
var result2 = plus('2', 'number') // '2number'? 우리가 원했던 결과는 아니네요
var result3 = plus(1, '2') // 12, 여기부터 심각하게 잘못된 결과가 나오기 시작합니다.
```

- 타입에 너무 자유로워서 문제다

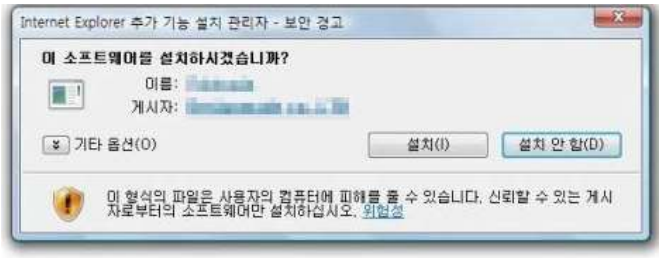
- 엉망 진창인 이유는 10일만에 급조한 프로그래밍 언어라서 그렇다.

브라우저 전쟁

- 넷스케이프 회사의 '모자이크' 브라우저가 처음으로 등장
- 이 회사에서 자바스크립트를 만듬
- 그 다음 출시된 IE는 기술력은 떨어졌으나 공격적인 마케팅으로 시장 점유율 완승
  - 전면 무료화
  - 윈도우 끼워 팔기
  - JScript라는 이름으로 그대로 뺏긴 후 IE에 집어넣음
- IE의 단점
  - 모든 브라우저에서 JS 사용법을 통일하자는 웹 표준에 참여하지 않음
  - 브라우저 호환 무시
  - 자동 업데이트 미지원

- 2008년 크롬의 등장
  - 구글에 들어가면 자연스럽게 크롬을 다운받을 수 있도록 유도
  - 압도적인 기술력(V8 엔진)
  - 구글 닥스, 구글 스프레드시트와 같은 클라우드 서비스 지원
  - IE의 액티브 X를 사용하지 않음

- 액티브 X란?



◇액티브X 방식으로 배포되는 프로그램

https://cdragon.tistory.com/entry/자바스프링JavaSpring와-Nodejs-대기업은-자바-스타트업은-Nodejs-Spring과-Nodejs-중에-고민이신가요#자바의 급부상 | 발전-1

- MS가 만든 기술
- 브라우저에서 접근한 사람의 컴퓨터의 HDD, 프린터, 웹캠 등에 접근할 수 있는 권한
- 단점
  - 해킹에 취약
  - 웹사이트 마다 다운로드 필요

- JQuery란?
  - 브라우저 마다 다르게 동작한 JS 문법을 하나로 통일시킨 라이브러리
  - jQuery로 코드를 짜면 어떤 브라우저에서 접속하든 잘 돌아간다.

- 크롬의 V8엔진
  - 엔진이란 JS를 해석해서 컴퓨터에 전달하는 역할
  - 일반인이 봐도 다른 브라우저와는 압도적인 퍼포먼스

Node.js 등장

- 크롬에서 JS를 실행할 수 있는 V8엔진을 크롬 내부가 아닌 외부에서도 사용할 수 있도록 별도의 실행 환경을 구축 한 것

장점

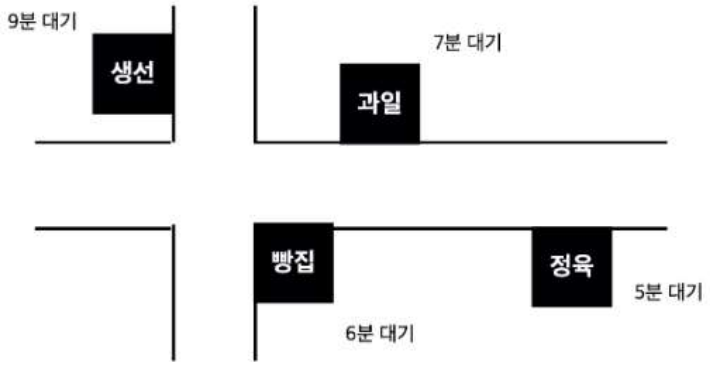
- I/O 작업을 **이벤트 기반 비동기 방식**으로 처리해 성능이 빠르다
- v8 엔진을 사용해서 성능이 빠르다
- 싱글 스레드기반이라 동시성 지옥에서 자유롭다

단점

- 고사양 스펙(CPU 코어수)에 설치되어있다 하더라도 싱글 스레드 이기 때문에 해당 서버의 성능을 활용할 수 없다.
- CPU 집약적인 작업이 발생하면 단일 스레드에서 순차적으로 처리되므로 성능에 영향이 간다.

이벤트 기반 비동기 방식이란?

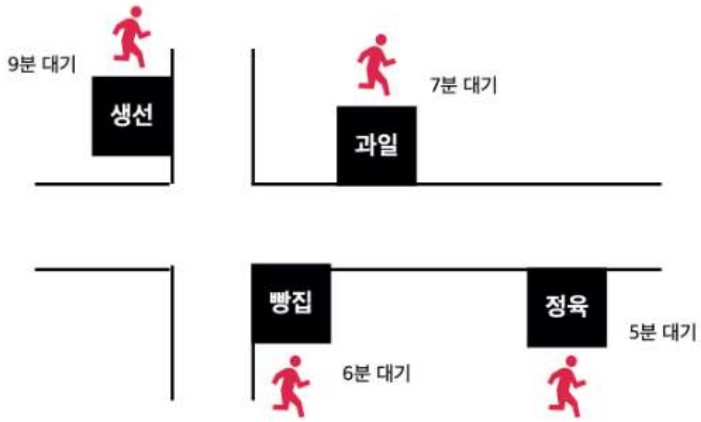
- 각 가게에 들려서 물건을 사야 한다.



https://joie-kim.github.io/Event-Driven-Non-blocking/

- 각 가게마다 기다렸다가 물건을 구입하면 9 + 7 + 6 + 5 = 27분 걸림

1. 몸을 복제해서 동시에 여러 가게에 간다.



https://joie-kim.github.io/Event-Driven-Non-blocking/

- 몸을 4개로 복제해서 각 가가에 가면 9분 안에 물건을 모두 구입 가능
- 톰캣의 스레드 기반 동기 방식

2. 시장에 대기 시스템을 도입해 대기표를 받고 기다린다.

- 각 가게에 돌면서 대기표를 받고, 대기 번호를 부르면 가게에서 가서 물건을 구입한다.
- 9분안에 모든 물건을 구입할 수 있다.
- NodeJs의 이벤트 기반 비동기 방식
- 스레드가 하나이지만 이벤트를 사용하기 때문에 빠른 속도로 일 처리 가능

Node.js와 Express.js

- Node.js는 Spring과 같은 프레임워크 라기보다는 js를 실행하는 환경이고 Express.js가 웹 프레임 워크이다.

Node.js vs Spring

스레드

- Nodejs는 모든 요청을 하나의 스레드로 처리
  - 이벤트 루프에 의해 모든 요청을 하나의 스레드로 받음
  - 그다음, 내부 아키텍처인 libuv 스레드풀 공간에 논 블로킹 방식으로 요청을 던짐
- Spring은 멀티 스레드 구조로 되어 있어 많은 양의 연산을 효율적으로 처리 가능

메모리

- Nodejs는 메모리를 덜 잡아 먹음
  - 아무것도 하지 않았을때 25MB
  - 하나의 스레드만 이용해서 구현하기 때문.
  - 가동시간이 1초 정도 걸림
- Spring은 메모리를 많이 잡아 먹음
  - 아무것도 하지 않아도 400MB
  - 가동시간이 10 ~ 20초 정도 걸립니다.

위험성

- nodej는 js라는 위험한 언어를 사용해야 함
  - 요즘은 ts가 나와서 좋아짐
- 스프링에서 사용하는 자바는 검증된 안정적인 언어

시장 검증

- nodejs는 검증된지 얼마 안됐다.
- 스프링은 20년 이상 사용되면서 검증 되었다.

msa

- msa에서는 서비스 별로 서버를 나누고 트래픽도 나눠서 받음
- 이벤트에 맞춰 서버 수가 트래픽에 유동적으로 바뀌는 상황이 많음
- 따라서, 크고 강력한 한대의 서버 보다는 빠르고 가벼운 서버가 여러 대 있는 쪽이 유리한 상황이 많음
- 빠른 서버 가동 시간과 적은 memory 사용량을 가진 node.js의 장점이 빛을 발하고 있음
- 하나의 스레드만 사용해서 CPU 애플리케이션 성격의 응답성이 떨어지는 단점들도 서버를 여러대 두면 무마 가능한 단점이 됨

변화의 바람

- nest js가 나오면서 ts + nest js 조합이 조금식 각광받고 있음
- nest.js란?
  - java spring과 같이 어노테이션을 통해 편리하게 di, ioc 기반의 개발 가능

    - typeorm 이라는 orm 프레임워크도 적극 지원
    - 또한, jest나 mocha 같은 좋은 테스트 툴 나옴 → 유닛 테스트 가능
- 하지만 아직 nodejs로 갈아탈 시점은 아님
- 자바 스프링은 실무에서 안정성이 검증된 언어
- 하지만 php → java 처럼 언젠간 nodejs 날이 올꺼다.