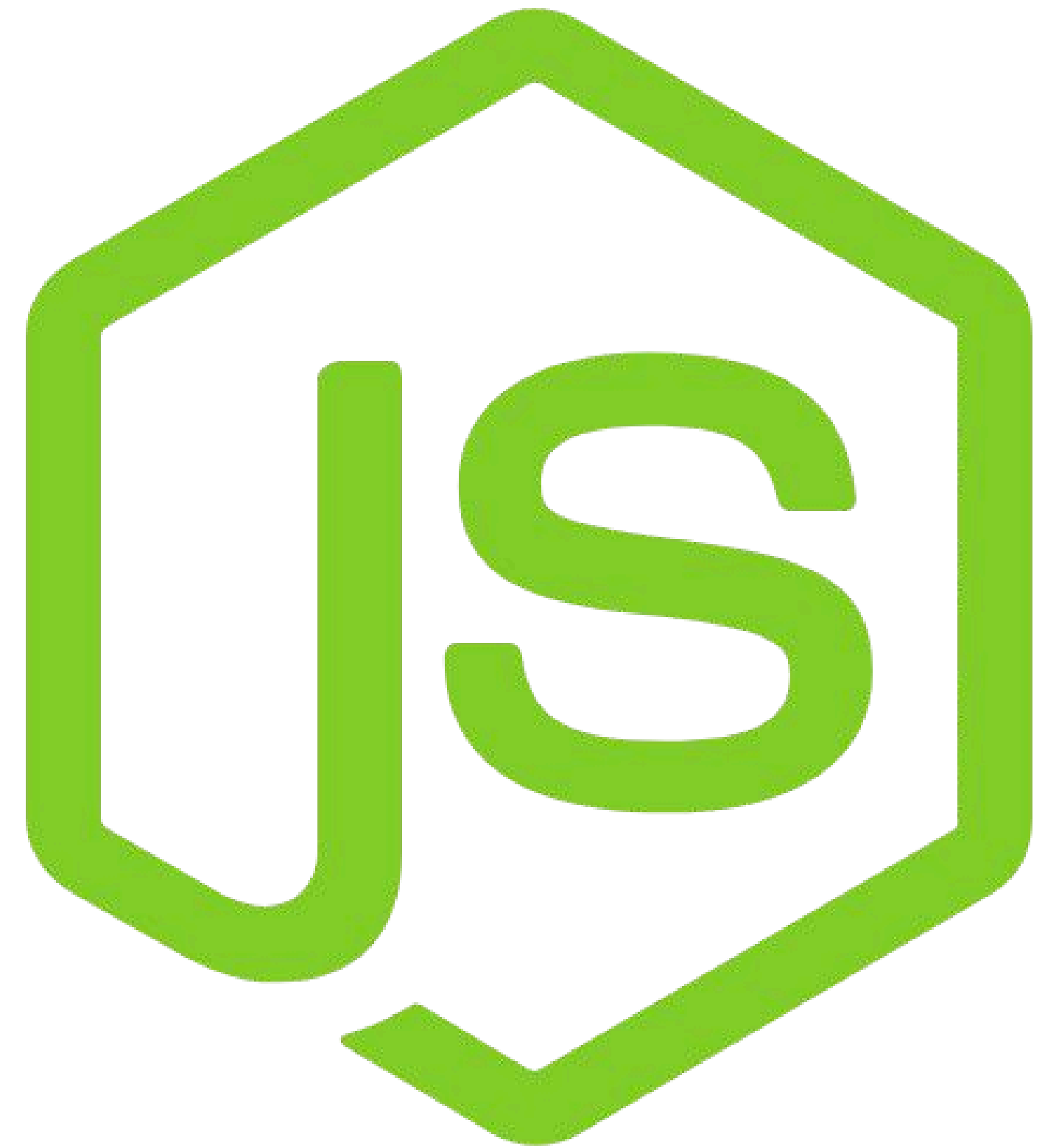


[1주차] Node JS to TS

1장 / 이윤슬



Node란?

- Node.js®는 Chrome V8 엔진으로 빌드된 자바스크립트 런타임

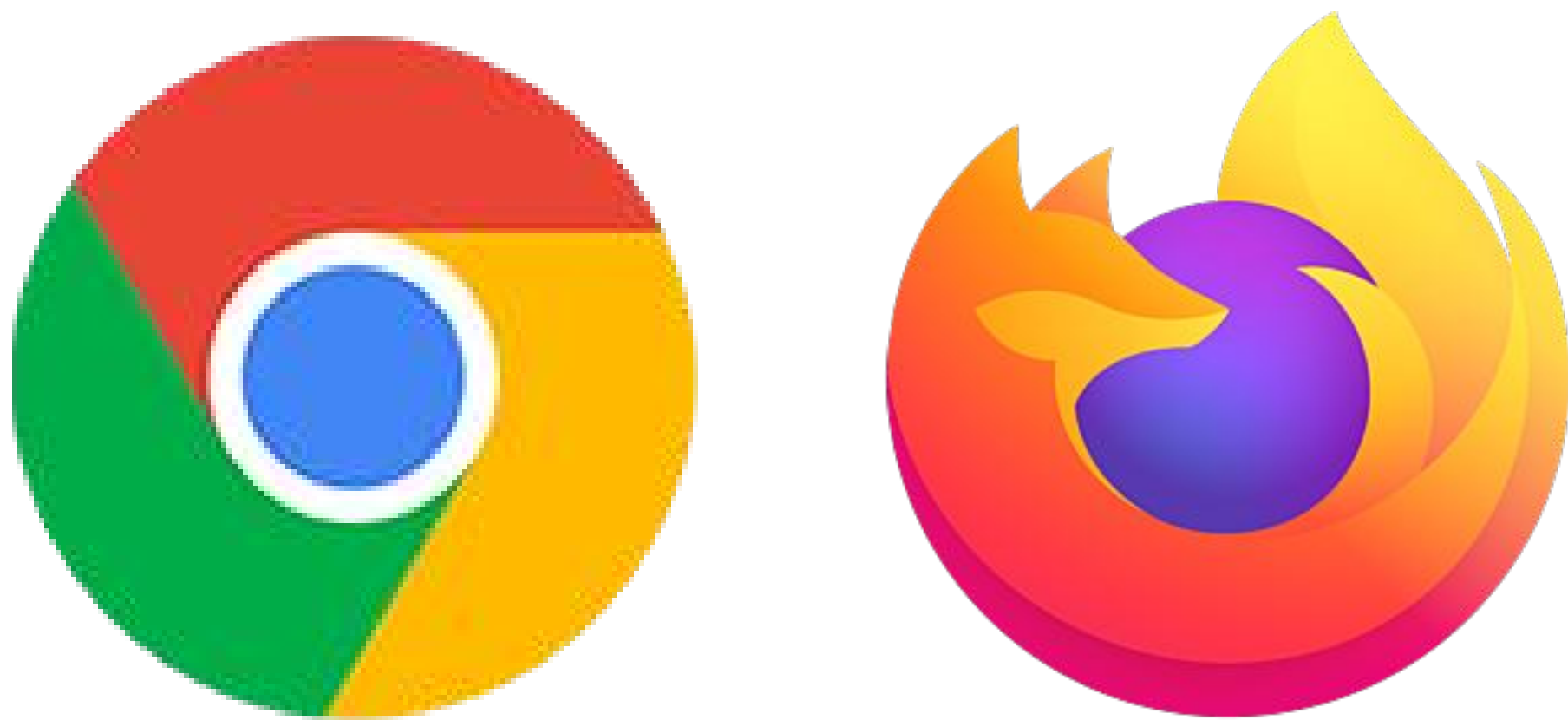
Node.js®에 대하여

Node.js는 비동기 이벤트 기반의 JavaScript 런타임으로, 확장 가능한 네트워크 애플리케이션을 구축하도록 설계되었습니다. 다음의 "hello world" 예제에서는 많은 연결을 동시에 처리할 수 있습니다. 각 연결마다 콜백이 호출되지만, 할 일이 없으면 Node.js는 대기 상태가 됩니다.

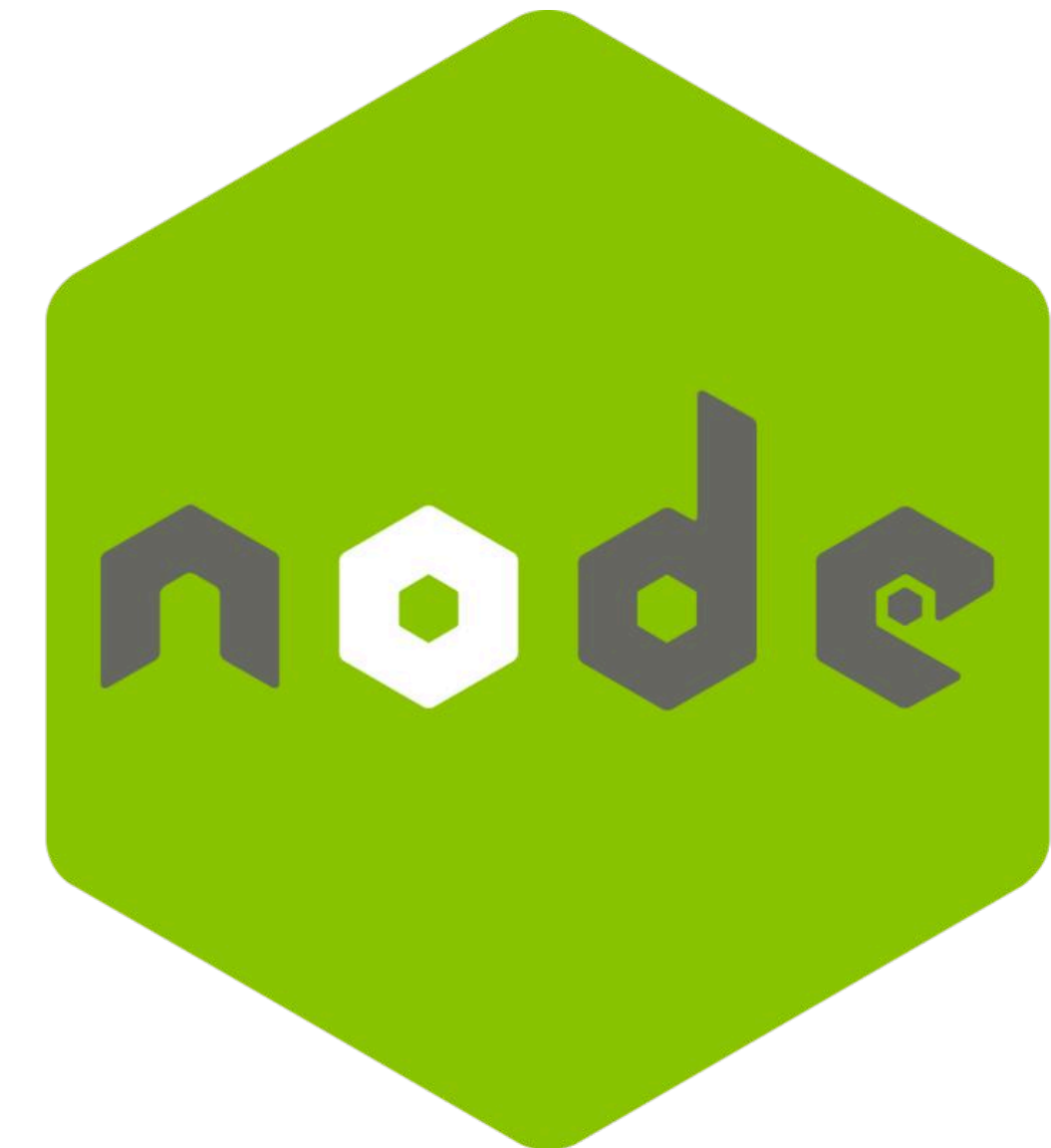
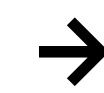
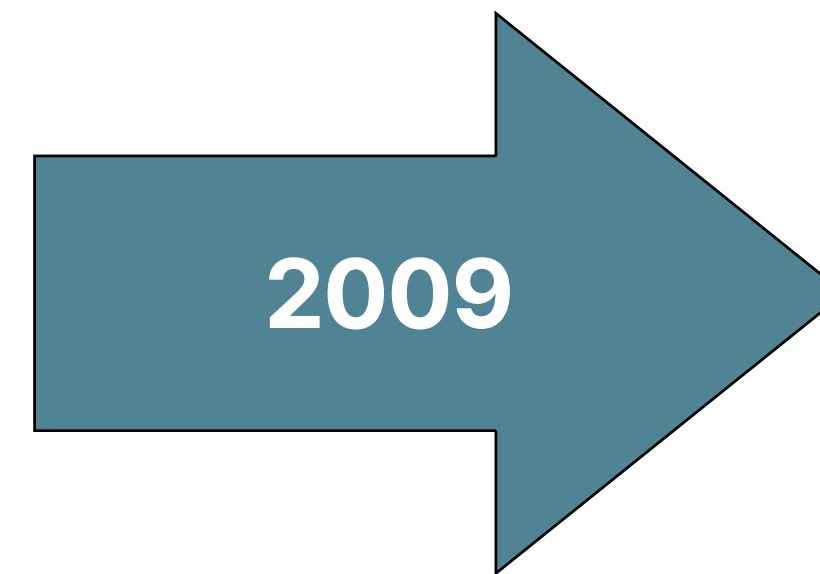
- Node는 “서버를 만들 수 있는” 도구이자 환경이지 “서버 그 자체”는 아니다.
- Node.js는 “자바스크립트를 어디서든 실행할 수 있는 런타임”

JS 런타임

- 런타임은 쉽게 말해 실행 환경



크롬, 파이어폭스 같은 브라우저들이 JS 런타임을 내장



JS를 브라우저 밖에서도 사용

V8 엔진

- V8은 구글이 만든 JavaScript 엔진
- 크롬 브라우저에 들어가 있는 바로 그 엔진!
- V8의 장점 ? 빠르다 ⇒ Node.js도 빠르다.



서버와 Node.js

- 서버가 뭘까?
요청에 응답하는 컴퓨터 혹은 프로그램
- 네이버에 접속한다 → 네이버 서버에 웹 페이지 요청 → 서버가 HTML CSS 이미지를 보내줌 → 응답
→ 브라우저가 이걸 화면에 표시
⇒ 서버는 요청과 응답을 처리한다.

그렇다면 왜 Node를 서버로 쓸까?

논블로킹 I/O

싱글스레드

프로세스와 스레드

- 프로세스 = “실행 중인 프로그램 하나”
 - 크롬 브라우저 창 하나 → 프로세스 1개
 - VS Code 실행 → 프로세스 1개
 - Node.js 서버 실행 → 프로세스 1개

⇒ 각 프로세스는 독립적인 메모리 공간을 가지겠죠?

- 스레드 = “프로세스 내에서 실제 일을 하는 작업자”
 - 프로세스 = 회사 건물 (독립적인 공간)
 - 스레드 = 그 안에서 일하는 직원들

싱글스레드 VS 멀티스레드

- 싱글스레드:
 - 프로세스 안에 작업자가 1명
 - 한 번에 한 가지 일만 처리
 - 장점: 단순하고, 관리 쉬움, 메모리 적게 사용
 - 단점: 한 가지 일에 막히면 다른 일 못 함
 - JS
- 멀티스레드:
 - 프로세스 안에 작업자가 여러 명
 - 동시에 여러 가지 일 처리
 - 장점: 병렬 처리, CPU 코어 여러 개 활용
 - 단점: 관리 복잡, 메모리 많이 사용, Race Condition 문제
 - Java, Python

논블로킹 I/O?



전통적인 멀티스레드 블로킹 방식 (예: 옛날 Apache 서버)

손님(요청) 1명당 점원(스레드) 1명 배정
점원은 그 손님의 음식이 나올 때까지 옆에서 계속 대기
손님이 100명이면? 점원도 100명 필요 → 인건비(메모리) 폭발!

살려줘



Node.js의 싱글스레드 논블로킹 방식

점원(메인 스레드)이 딱 1명
하지만 이 점원은 주문만 받고 바로 다음 손님한테 감
주방(libuv 라이브러리)에서 음식이 완성되면 벨이 울림 → 그때 가서 서빙
점원 1명이 100명 손님을 동시에 처리 가능!

노드도 사실 멀티스레드를 쓴다?

점원이 한명이면 무거운 작업은 어떻게 돌릴까??

사실 Node.js는 내부적으로 멀티스레드를 사용

정확히 말하면:

- 이벤트 루프(점원)는 싱글스레드
 - 뒷단의 주방(Thread Pool)은 멀티스레드
- 이걸 담당하는 게 libuv라는 라이브러리!

CPU 많이 쓰는 작업 ⇒ Thread Pool로 넘김

libuv가 하는 일:

1. 이벤트 루프 제공
2. 비동기 I/O 처리
3. Thread Pool 관리 (기본 4개)
4. 파일 시스템, 네트워크, DNS 등 처리

Thread Pool로 가는 작업들:

- 파일 시스템 작업 (fs 모듈)
- DNS 조회 (dns.lookup)
- Crypto 작업 (암호화)
- Zlib (압축)

Kernel API로 가는 작업들 (더 빠름):

- HTTP 요청 (http 모듈)
- TCP/UDP 통신

노드 선택의 이유

- 실시간 채팅, 알림 (카카오톡, 슬랙)
- REST API 서버 (수많은 작은 요청)
- 스트리밍 서비스
- I/O 작업이 많은 서비스

JavaScript 하나로 풀스택 개발
JSON 처리 최적화
npm 생태계 (수백만 개 패키지)
React, Vue도 Node.js 기반

But

- 이미지/영상 처리
 - 대규모 데이터 분석
 - 머신러닝/AI
 - CPU 집약적 연산
- 이런 건 Python, Java가 더 적합함

QnA

자유롭게 질문해주세요.

Node.js

조현영 지음