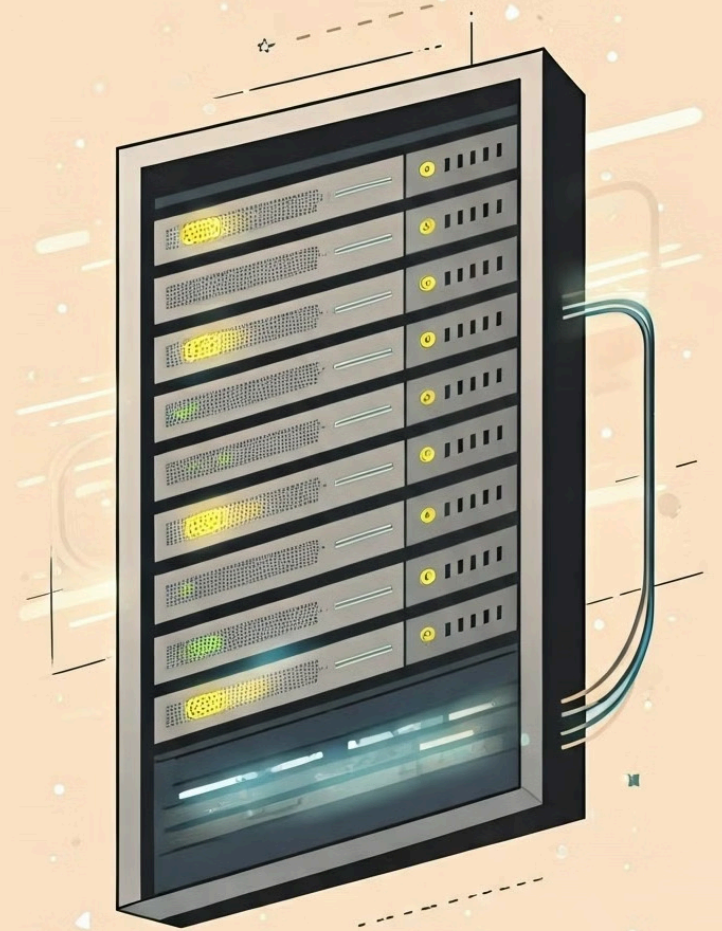


# Node.js 소개

JavaScript로 서버 사이드 프로그래밍

● 작성자: 김상훈 1





# Node.js란 무엇인가?

1. JavaScript로 서버 애플리케이션 개발
2. Chrome V8 엔진 기반, 비동기 이벤트 모델
3. 단일 스레드, 비동기 처리로 효율적 요청 처리

# Node.js의 주요 특징

- 1 비동기 I/O**  
동시 요청-응답 처리로 높은 성능
- 2 단일 스레드**  
효율적 리소스 활용
- 3 빠른 실행 속도**  
V8 엔진 기반 JavaScript
- 4 모듈화**  
NPM으로 손쉬운 모듈 설치 및 사용
- 5 크로스 플랫폼**  
Windows, macOS, Linux 지원

# Node.js의 장점

## 고성능

비동기 I/O로 높은 성능

## 빠른 개발

클라이언트-서버 동일 언어

## 스케일링 용이

이벤트 기반 아키텍처

## 넓은 생태계

NPM으로 빠른 기능 추가

# Node.js 활용 사례

## RESTful API 서버 개발

Node.js의 비동기 처리 능력을 활용하여 효율적인 API 서버를 구축할 수 있습니다.

## 실시간 애플리케이션

채팅, 알림 등의 실시간 애플리케이션을 Node.js로 구현할 수 있습니다.

## 데이터 스트리밍

비디오, 데이터 전송 등의 스트리밍 서비스를 Node.js로 구축할 수 있습니다.

## IoT 디바이스 통신 서버

Node.js로 IoT 디바이스의 통신 서버를 개발할 수 있습니다.





# Node.js 기본 사용법

Node.js를 시작하는 가장 기본적인 방법은 공식 웹사이트에서 Node.js를 설치하는 것입니다. 그런 다음 첫 번째 코드로 `console.log('Hello, Node.js!')`를 작성하여 Node.js가 정상적으로 작동하는지 확인할 수 있습니다.

더 나아가 간단한 HTTP 서버를 만들어 볼 수 있습니다. 3000번 포트에서 실행되는 HTTP 서버를 만들면 웹 브라우저에서 해당 서버에 접속할 수 있습니다.

# Express.js 소개

인기 있는 Node.js 웹 프레임워크

- 라우팅, 미들웨어, RESTful API 지원
- 간단한 HTTP 서버 예시 코드





# NPM (Node Package Manager)



## 모듈 설치

`npm install <패키지명>`



## 의존성 관리, 프로젝트 초기화

`package.json`



## Express 설치, 실행

`npm install express, node app.js`



# 성능 최적화 및 확장성

1

## 비동기 I/O

동시 요청 처리

2

## 클러스터링

멀티코어 CPU 활용

3

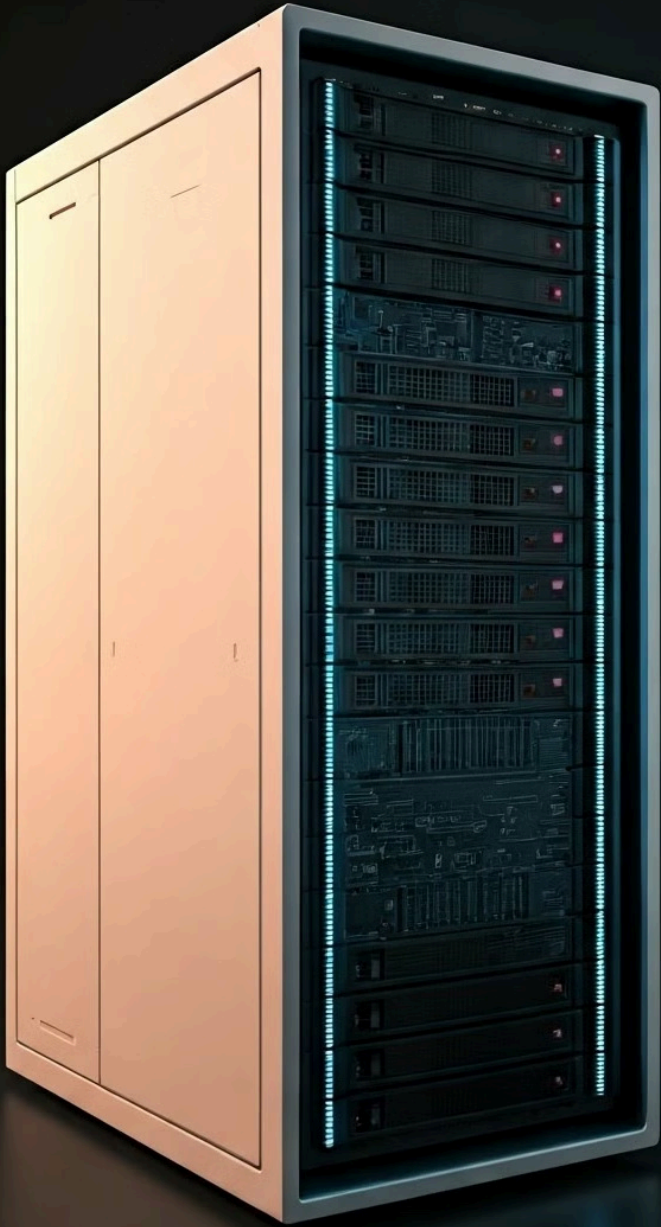
## 캐싱

데이터베이스 호출 감소

4

## 메모리 관리

대규모 애플리케이션 처리



# 결론

## 1 고성능 서버 사이드 플랫폼

Node.js는 비동기 I/O와 단일 스레드 모델로 고성능을 제공합니다.

## 2 비동기 I/O와 단일 스레드 모델

동시 요청 처리와 효율적인 리소스 활용이 가능합니다.

## 3 Express.js와 NPM으로 개발 가속화

Node.js 생태계를 통해 빠른 기능 추가와 개발이 가능합니다.

