

NODE.JS 소개

비동기 이벤트 기반 JavaScript 런타임

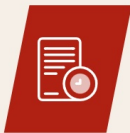
김상훈

```
Run | Debug
public static void main(String[] args) throws Exception {
    System.out.println("Hello, World!");
}
```

NODE.JS 소개: 비동기 이벤트 기반 JAVASCRIPT 런타임

Node.js의 정의와 주요 특징

비동기 I/O 처리



I/O 작업을 비동기로 수행하여 성능을 극대화합니다.

단일 스레드 모델



이벤트 루프를 통해 수많은 연결을 관리합니다.

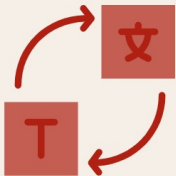
NPM



풍부한 패키지 생태계를 통해 재사용 가능한 모듈을 쉽게 설치하고 관리할 수 있습니다.

NODE.JS의 역사와 발전

Node.js의 탄생 배경과 발전 과정



Node.js는 2009년 라이언 달 (Ryan Dahl)에 의해 처음 발표됨

Node.js는 JavaScript를 서버 사이드에서 실행할 수 있도록 해주는 혁신적인 런타임입니다.



Node.js는 여러 주요 버전을 거치며 발전해옴

Node.js는 다양한 버전 업데이트를 통해 성능과 안정성을 지속적으로 개선해 왔습니다.



현재 Node.js는 많은 기업과 개발자들의 선택을 받음

Node.js는 대규모 기업부터 소규모 스타트업까지 다양한 개발자들에 의해 채택되고 있습니다.



웹 개발의 중요한 도구로 자리 잡음

Node.js는 특히 실시간 웹 애플리케이션에서 높은 성능을 발휘하여 필수 도구로 자리잡았습니다.

NODE.JS의 주요 특징

비동기 이벤트 기반 JavaScript 런타임



비동기 I/O

요청이 들어올 때마다 작업을 대기하지 않고 다음 요청을 처리하여 효율성을 극대화합니다.



이벤트 루프

단일 스레드에서 이벤트 기반으로 작업을 처리하여 높은 성능을 유지합니다.



NPM (Node Package Manager)

전 세계의 개발자들이 만든 수많은 패키지를 손쉽게 사용하고 관리할 수 있게 해줍니다.



NODE.JS의 장점과 혜택

비동기 처리로 인한 높은 성능과 확장
성

높은 성능

비동기 처리로 인해 빠른 응답 속도를 자랑합니다.

확장성

필요에 따라 쉽게 확장할 수 있는 구조를 가지고 있습니다.

커뮤니티와 생태계

활발한 커뮤니티와 다양한 라이브러리, 프레임워크가 존재합니다.

크로스 플랫폼 지원

다양한 운영 체제에서 동일한 코드를 사용할 수 있습니다.

NODE.JS의 주요 사용 사례

웹 서버, 실시간 애플리케이션, API
서버 및 마이크로서비스



1

웹 서버

RESTful API를 제공하는 웹 서버 구축에 적합합니다.

2

실시간 애플리케이션

채팅 애플리케이션이나 온라인 게임과 같은 실시간 기능을 요구하는 서비스에 효과적입니다.

3

API 서버

데이터베이스와의 효율적인 통신을 통해 API를 쉽게 구축할 수 있습니다.

4

마이크로서비스

모듈화된 아키텍처를 통해 마이크로서비스를 쉽게 개발하고 배포할 수 있습니다.

NODE.JS의 활용을 시작하세요!

지금 바로 Node.js를 배우고 활용해보세요.

