



Node.js 소개: JavaScript를 활용한 서버 사이드 개발

Node.js는 JavaScript를 사용하여 서버 사이드 애플리케이션을 개발할 수 있는 런타임 환경입니다. Node.js는 빠르고 확장 가능하며 다양한 애플리케이션에 사용됩니다.

 **작성자: 은지 이**

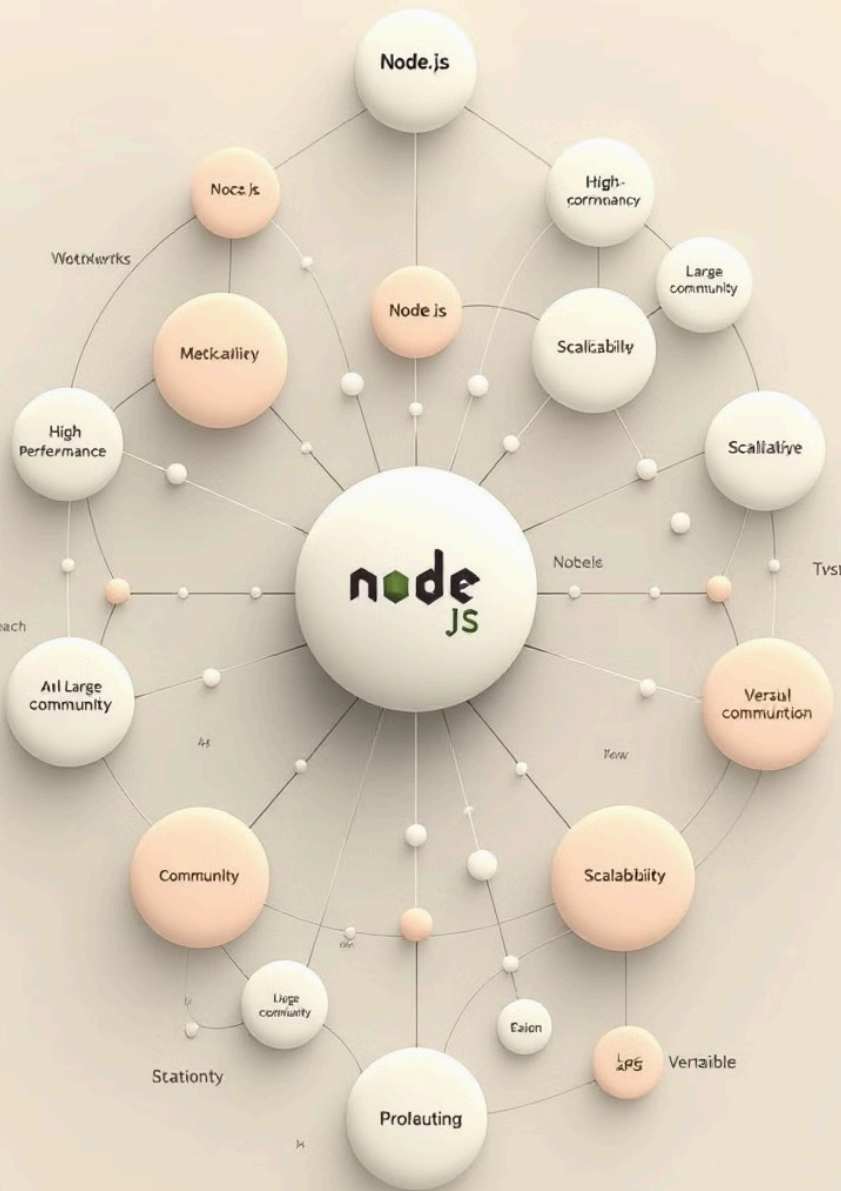
Node.js의 주요 특징

비동기식 이벤트 루프

Node.js는 비동기식 이벤트 루프를 사용하여 효율적으로 여러 요청을 처리합니다. 이를 통해 빠르고 확장 가능한 서버 애플리케이션을 구축할 수 있습니다.

단일 스레드 모델

Node.js는 단일 스레드 모델을 사용하며 이벤트 루프를 통해 다중 작업을 관리합니다. 이는 뛰어난 성능과 경량화된 프로세스를 제공합니다.



Node.js의 장점

1 고성능

Node.js는 비동기식 이벤트 루프와 단일 스레드 모델을 통해 높은 성능과 효율성을 제공합니다.

2 확장성

Node.js는 웹 스케일 애플리케이션을 위한 확장 가능한 아키텍처를 제공합니다.

3 활발한 커뮤니티

Node.js는 활발한 커뮤니티와 다양한 패키지 및 라이브러리를 제공합니다.

Node.js의 활용 분야

웹 애플리케이션

Node.js는 빠르고 확장 가능한 웹 애플리케이션을 구축하는 데 적합합니다. Express.js와 같은 프레임워크를 사용하면 쉽게 웹 애플리케이션을 개발할 수 있습니다.

API 개발

Node.js는 RESTful API와 같은 다양한 API를 개발하는 데 사용됩니다. 다양한 라이브러리를 통해 API 개발을 간소화할 수 있습니다.

실시간 애플리케이션

Node.js는 웹소켓을 통해 실시간 애플리케이션을 구축하는 데 적합합니다. 실시간 채팅, 게임, 데이터 스트리밍 애플리케이션을 개발할 수 있습니다.

Node.js 설치 및 실행



다운로드

Node.js 웹사이트에서 Node.js를 다운로드합니다.



설치

다운로드한 설치 파일을 실행하여 Node.js를 설치합니다.



실행

터미널에서 `node -v` 명령을 실행하여 Node.js가 설치되었는지 확인합니다.

Node.js로 웹 애플리케이션 구축

1

프로젝트 생성

새로운 Node.js 프로젝트를 생성합니다.

2

의존성 설치

필요한 패키지 및 라이브러리를 설치합니다.

3

서버 구축

Express.js와 같은 프레임워크를 사용하여 웹 서버를 구축합니다.

4

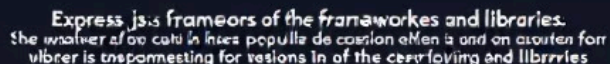
라우팅 설정

URL 경로에 따라 다른 페이지를 렌더링하는 라우팅 설정을 합니다.

5

템플릿 엔진 사용

EJS, Pug와 같은 템플릿 엔진을 사용하여 HTML 콘텐츠를 렌더링합니다.



1

Express.js는 Node.js 웹 애플리케이션 개발을 위한 가장 인기 있는 프레임워크입니다.

2

NestJS는 타입스크립트를 기반으로 한 Node.js 프레임워크로, 엔터프라이즈급 애플리케이션을 개발하는 데 적합합니다.

3

Mongoose는 MongoDB와 같은 NoSQL 데이터베이스와 상호 작용하기 위한 ODM(Object Document Mapper)입니다.

4

Socket.IO는 실시간 양방향 통신을 위한 라이브러리로, 웹소켓을 통해 실시간 애플리케이션을 개발하는 데 사용됩니다.

Node.js의 미래와 전망

Node.js는 빠른 속도, 확장성, 다양한 활용 분야를 통해 앞으로도 계속 성장할 것으로 예상됩니다. 특히 서버리스 컴퓨팅과 클라우드 기반 개발 환경에서 Node.js는 중요한 역할을 할 것으로 전망됩니다.

