



# Node.js의 생태계: NPM과 주요 프레임워크

Node.js는 서버 측 JavaScript 런타임 환경입니다. 비동기적이고 이벤트 기반의 애플리케이션 개발을 가능하게 합니다. NPM과 다양한 프레임워크가 Node.js의 강력한 생태계를 형성합니다.

 작성자: 용효 김



# NPM (Node Package Manager)

## 1 패키지 관리

NPM은 Node.js의 공식 패키지 관리자입니다. 패키지 설치, 업데이트, 관리를 쉽게 할 수 있습니다.

## 2 개발 효율성

전 세계 개발자들의 패키지를 사용할 수 있어 개발 시간과 노력을 절약할 수 있습니다.

## 3 주요 기능

패키지 설치, 의존성 관리, 스크립트 실행 등의 기능을 제공합니다.

# Express.js

## 인기 프레임워크

Node.js에서 가장 많이 사용되는 웹 프레임워크 중 하나입니다.

## 간단한 API

간단하고 유연한 API를 제공합니다.

## 웹 서버 구축

웹 서버와 RESTful API를 쉽게 구축할 수 있습니다.



# Koa.js

## 개발 팀

Express를 만든 팀이 개발한 또 다른 웹 프레임워크입니다.

## 비동기 함수

미들웨어 구성 요소를 비동기 함수로 작성합니다.

## 장점

더 나은 오류 처리와 간결한 코드 작성이 가능합니다.



# Nest.js

## Nest.js

1

### 영감

Angular에서 영감을 받은 Node.js 프레임워크입니다.

2

### TypeScript

TypeScript를 기본으로 사용합니다.

3

### 모듈화

모듈화된 구조를 제공합니다.



# Node.js 설치하기

1

## 다운로드

Node.js 공식 웹사이트에서 운영 체제에 맞는 설치 프로그램을 다운로드합니다.

2

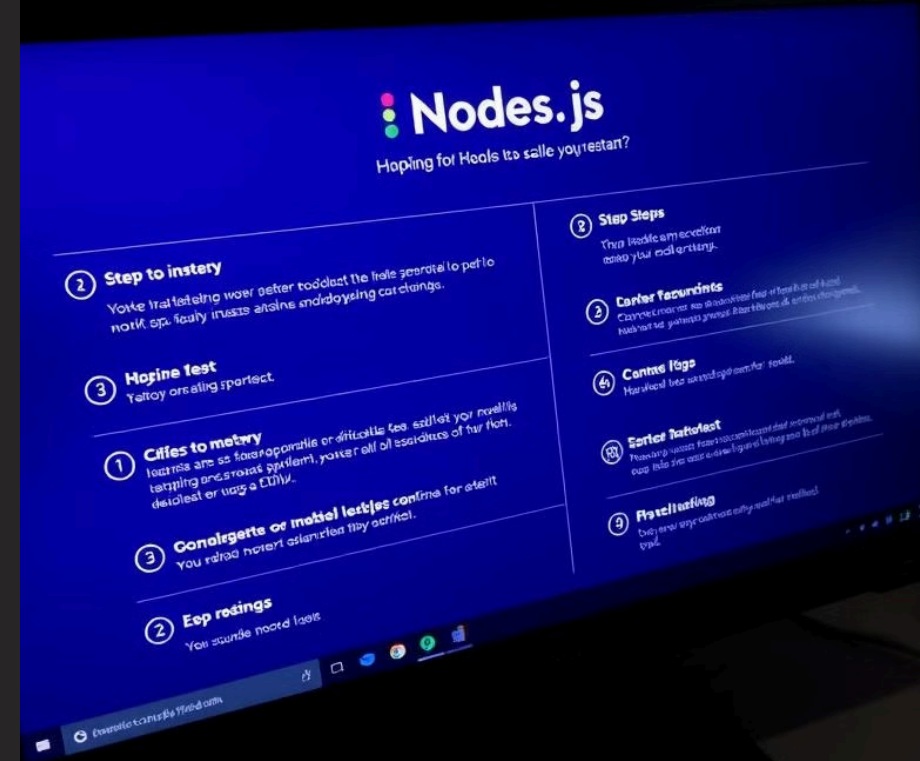
## 설치

다운로드한 설치 프로그램을 실행하여 Node.js를 설치합니다.

3

## 확인

터미널에서 `node -v`와 `npm -v` 명령어로 설치를 확인합니다.





# 간단한 애플리케이션 작성

1

## 프로젝트 생성

새 디렉토리를 만들고 npm init으로 package.json 파일을 생성합니다.

2

## Express 설치

npm install express 명령어로 Express.js를 설치합니다.

3

## 서버 코드 작성

index.js 파일에 간단한 서버 코드를 작성합니다.

4

## 서버 실행

node index.js 명령어로 서버를 실행합니다.



# Node.js 생태계의 중요성



## 생산성 향상

NPM의 방대한 패키지로 개발자의 생산성을 높입니다.



## 유연성

다양한 프레임워크로 특정 요구에 맞는 솔루션을 제공합니다.



## 혁신

Node.js 환경에서 창의적이고 혁신적인 프로젝트를 구축할 수 있습니다.