

# Node js

Producer: 김규태





01

## 서버 자바스크립트

서버 사이드 자바스크립트는 Node.js 를 통해 서버 측에서 자바스크립트를 실행할 수 있는 기술입니다

02

## 이벤트 기반 비동기

이벤트 기반 비동기 I/O 는 Node.js 가 이벤트 기반 비동기 방식으로 I/O 작업을 처리하는 것을 의미합니다

03

## 크로스 플랫폼 런타임

크로스 플랫폼 런타임 환경은 Node.js 가 다양한 플랫폼에서 실행될 수 있는 것을 의미합니다





## 특징

### 고성능

Node.js 는 고성능을 제공하여 빠른  
응답 속도를 제공합니다

01

02

### 싱글 스레드

Node.js 는 싱글 스레드 모델을  
사용하여 병렬 처리를 가능하게 하

03

### 확장성

Node.js 는 확장성이 높아 대규모  
애플리케이션을 쉽게 구축할 수  
있습니다

04

### 비동기 처리

Node.js 는 비동기 처리를 지원하  
데이터 처리 속도를 높입니다





## 구성 요소

### V8 자바스크립트

V8 자바스크립트 엔진은 Node.js의 핵심 구성 요소로, 코드 실행과 빠른 성능을 제공합니다

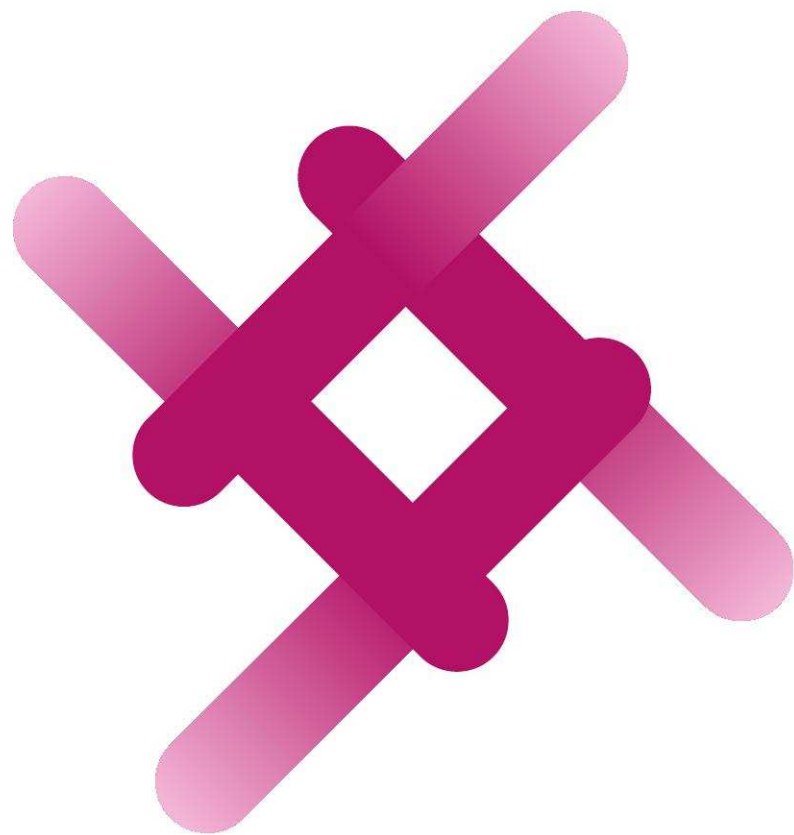
### 이벤트 루프

이벤트 루프는 Node.js의 핵심 구성 요소 중 하나입니다. 이 루프는 이벤트 처리 및 비동기 작업을 처리합니다

### C++ 추가 모듈

C++ 추가 모듈은 Node.js의 핵심 구성 요소 중 하나로, 추가 기능을 제공하고 확장성을 높입니다





## ■ 클라이언트 - 서버 모델

클라이언트 - 서버 모델 '은 컴퓨터 네트워크에 사용되는 아키텍처로, 63자로 제한된다. 추가 반환하지 않음. 모든 출력은 "한국어"로 이루어

## ■ 요청 - 응답 사이클

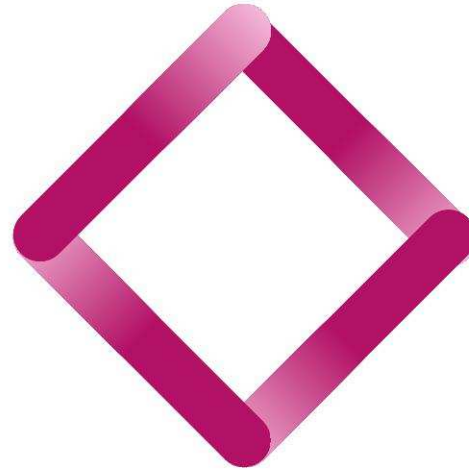
'요청 - 응답 사이클'은 정보를 주고받는 과정을 의미합니다



## 설치

### Node.js 다운로드

먼저, 브라우저에서 Node.js  
홈페이지를 열고 다운로드  
링크를 클릭합니다



### 패키지 매니저 사용

npm ; yarn







## 기본 모듈

**http**

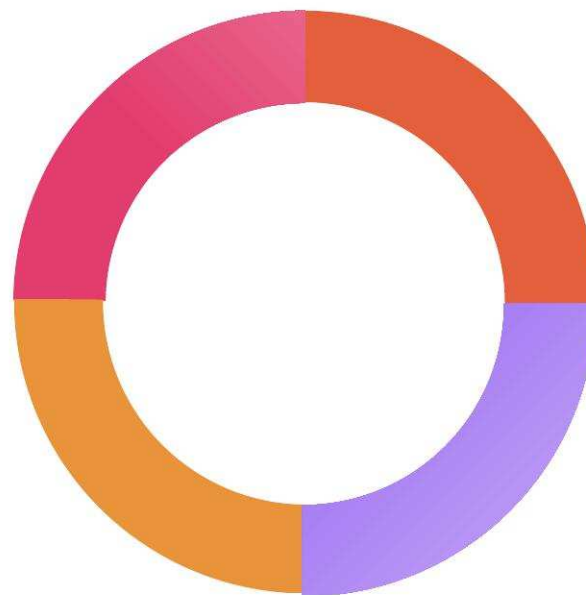


提供 HTTP 客户端和服务端功能的模块

**path**



提供处理文件路径的模块



**fs**



提供文件系统操作的模块

**os**



提供操作系统功能的模块





# 패키지 관리

## 01 npm 사용법

npm은 Node.js 패키지 관리 도구로  
패키지 설치, 업데이트, 제거 및 의존성  
관리 가능합니다

PART 01

## 02 패키지 설치

`npm install ;npm uninstall`

## 03 종속성 관리

`package.json ;package-lock.json`

PART 03







# 프레임워크



REST API 구축 ; 미들웨어 사용



轻量级的 Node.js Web 框架



基于 TypeScript 的 Node.js 框架

Express.js

Koa.js

NestJS



# 비동기 프로그래밍

## 콜백 함수

비동기  
프로그래밍에서  
콜백 함수는 함수의  
실행 결과를 받아  
처리하는 함수를  
의미합니다

PART 1

## 프라미스

프라미스는 비동기  
작업의 완료 여부를  
나타내는  
객체입니다

PART 2

## async/await

async/await 는  
비동기 작업을  
동기식으로  
처리하는  
방법입니다

PART 3



# 데이터베이스 연결

01

## MongoDB

mongoose 사용법 ; 데이터  
모델링

02

## MySQL

mysql 패키지 ; 쿼리 실행



# 테스트



## 단위 테스트

Mocha ;Chai

## 통합 테스트

통합 테스트는 모든 컴포넌트가 함께 작동하는지 확인하는 테스트입니다. 이를 통해 개발자는 애플리케이션의 모든 부분이 올바르게 작동하는지 확인할 수 있습니다

## 코드 커버리지

Istanbul





# 배포



## 클라우드 서비스

Heroku ;AWS



## 컨테이너화

Docker







# 보안

## 해킹 방지 방법

암호화 기술 사용

## 인증 및 권한 부여

JWT





## 01

### 클러스터링

클러스터링은  
Node.js  
애플리케이션의  
성능을 향상시키는  
데 도움이 됩니다

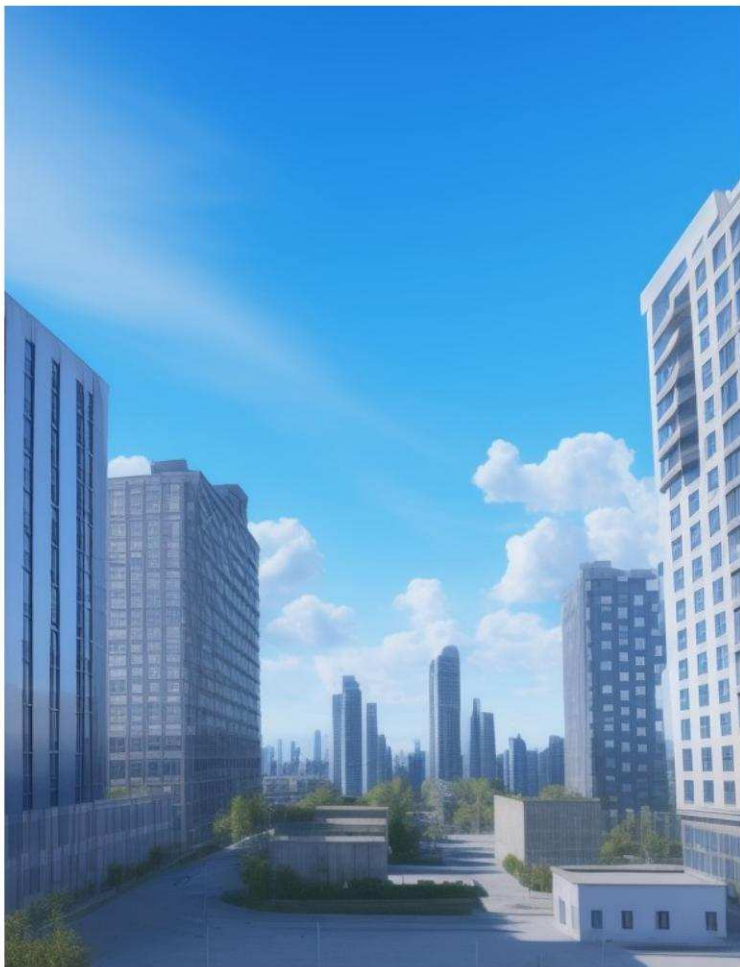


## 02

### 로드 밸런싱

로드 밸런싱은  
Node.js  
애플리케이션의  
성능을 향상시키는  
또 다른 방법입니다





## 다양한 모듈과 라이브러리

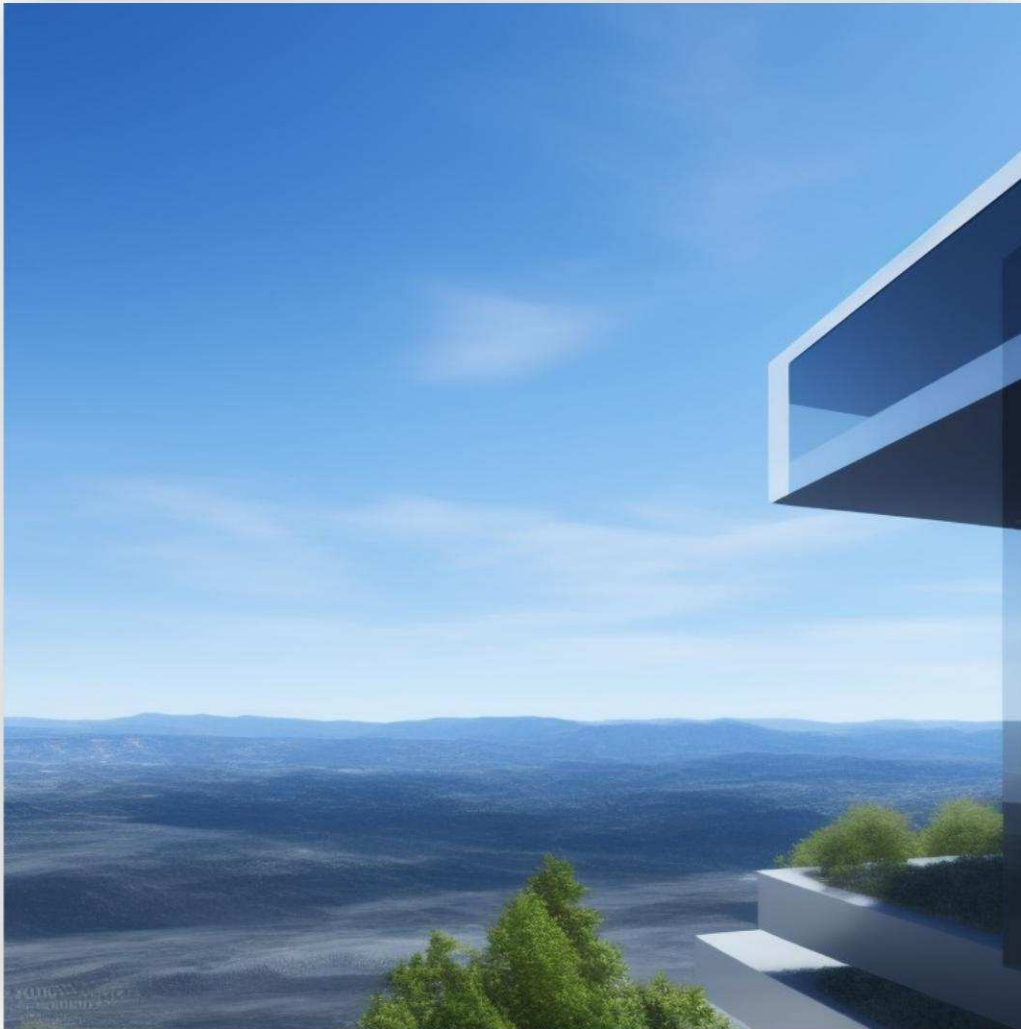
---

Socket.io ;GraphQL

---



## 커뮤니티



### GitHub 리포지토리

---

GitHub 리포지토리는 Node.js 개발자들이 프로젝트를 공유하고 협업할 수 있는 공간입니다. 다양한 프로젝트와 아이디어를 통해 개발자들은 새로운 것을 배울 수 있습니다

### 개발자 포럼

---

Stack Overflow ;Reddit





# 01

## 유명 웹사이트 사용

Netflix ;LinkedIn



# 02

## 스타트업 및 대기업 프로젝트

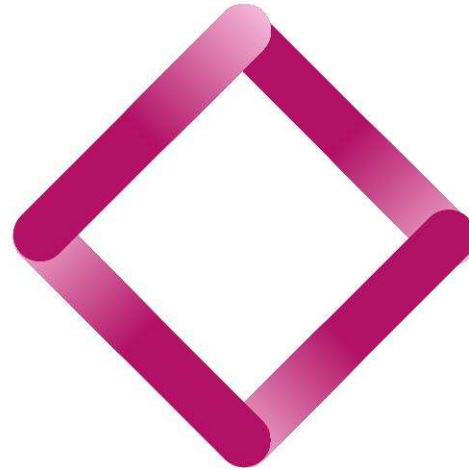
스타트업과 대기업은  
모두 Node.js 를  
사용하여 프로젝트를  
개발합니다





## 공식 문서 ■

Node.js API 문서



## ■ 커뮤니티 기여

블로그 포스트 ; 튜토리얼





## 01



### 온라인 강의

Udemy ; Coursera

## 02



### 책 추천

You Don' t Know JS ; Node.js  
Design Patterns







## 01

### 최신 트렌드

서버리스 아키텍처 ; 마이크로서비스





# 대안

1



## Django

Python 기반의 웹  
프레임워크

2



## Ruby on Rails

루비 기반의 웹  
프레임워크

3



## ASP.NET

C# 기반의 웹  
프레임워크



## 활용 사례

### RESTful API 개발

Node.js 는 RESTful API 개발에 적합하다

### 실시간 애플리케이션 개발

채팅 애플리케이션 ; 게임 서버 운영

### 스크립트 및 자동화

Node.js 는 스크립트 및 자동화 작업에 유용하다



# 비교

## Node.js vs Py

비교 언어 속성과 응용 분야

## Node.js vs Java

비교 속도, 메모리 사용량 및  
데이터 처리 능력

## Node.js vs Go

비교 속도, 메모리 사용량 및  
데이터 처리 능력





# THANKS

