



# Node.js: JavaScript을 위한 강력한 런타임 환경

Node.js는 웹 브라우저 외부에서 JavaScript를 실행할 수 있게 해주는 런타임 환경입니다. 이벤트 기반, 논블로킹 I/O 모델을 사용하여 확장 가능한 네트워크 애플리케이션을 구축할 수 있는 플랫폼을 제공합니다.

# Node.js의 핵심 구조

## 이벤트 루프

이벤트 루프는 Node.js의 핵심으로, 비동기 I/O 작업을 효율적으로 처리합니다.

## 논블로킹 I/O

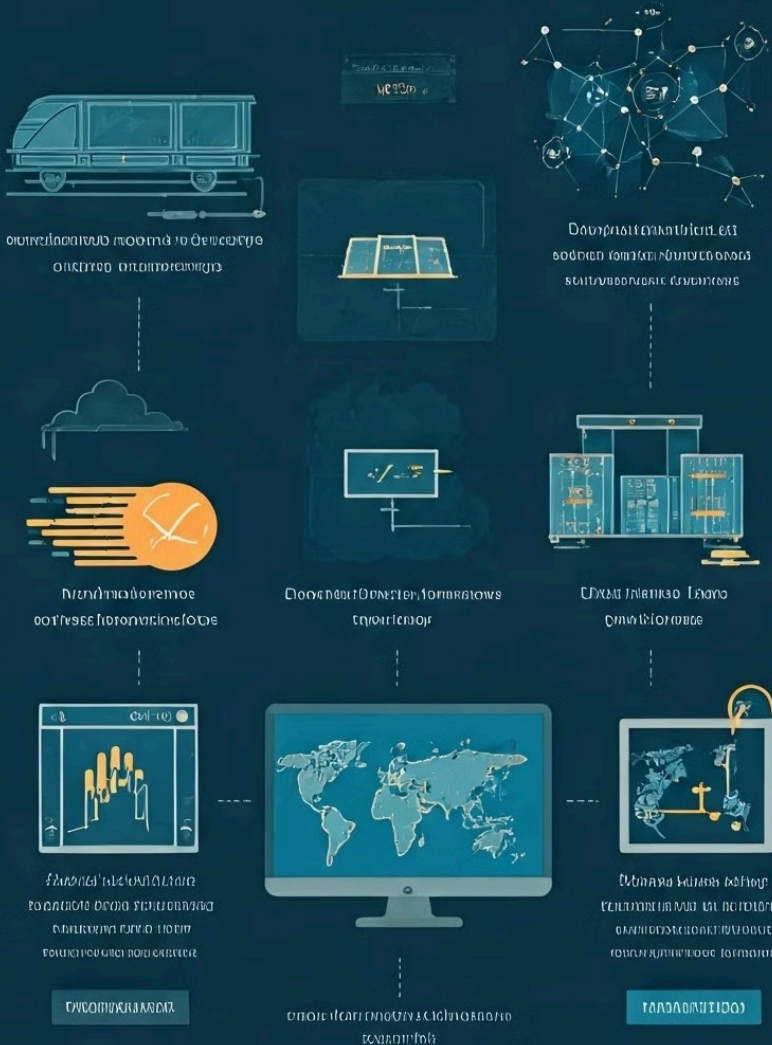
Node.js는 논블로킹 I/O 모델을 사용하여 여러 클라이언트 요청을 동시에 처리할 수 있습니다.

## 모듈 시스템

Node.js는 모듈 방식으로 설계되어 코드를 쉽게 구성하고 재사용할 수 있습니다.

## Node.js Features

Informational



# Node.js의 주요 기능

1

## 비동기식 및 이벤트 기반

Node.js는 고도로 확장 가능하고 비차단적인 네트워크 애플리케이션 구축에 탁월합니다.

2

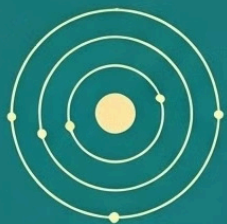
## 크로스 플랫폼 호환성

Node.js는 Windows, macOS, Linux 등 다양한 운영 체제에서 실행됩니다.

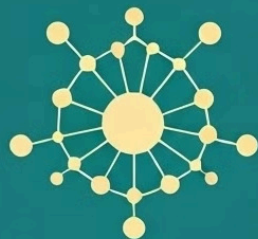
3

## 풍부한 생태계

Node.js는 활발한 커뮤니티와 다양한 모듈 및 라이브러리를 제공합니다.



SCALABILITY  
The ability of a system to handle a growing amount of work by adding resources to the system.



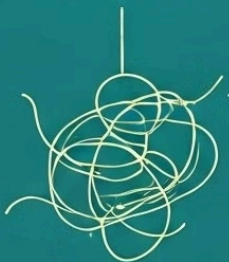
HIGH PERFORMANCE  
LARGE COMMUNITY



CALLBACK HELL  
A state of affairs in which a program's execution is blocked by a series of nested callbacks, leading to a complex and difficult-to-maintain code structure.



SECURITY VULNERABILITIES  
Weaknesses or flaws in a system's security that can be exploited by attackers to gain unauthorized access or cause damage.



CALLBACK HELL



COMPLEX INVESTIGATION

# Node.js의 장단점

## 장점

가벼우며 빠르고 효율적임. 실시간 및 I/O 집약적 애플리케이션에 탁월함.

## 단점

JavaScript에 익숙하지 않은 개발자에게는 가파른 학습 곡선. CPU 집약적 작업에 대한 지원이 제한적.

# Node.js를 사용한 웹 애플리케이션 구축

1

## Express.js

Node.js를 위한 인기 있는 웹 애플리케이션 프레임워크로, 웹 및 모바일 애플리케이션을 위한 강력한 기능 세트를 제공합니다.

2

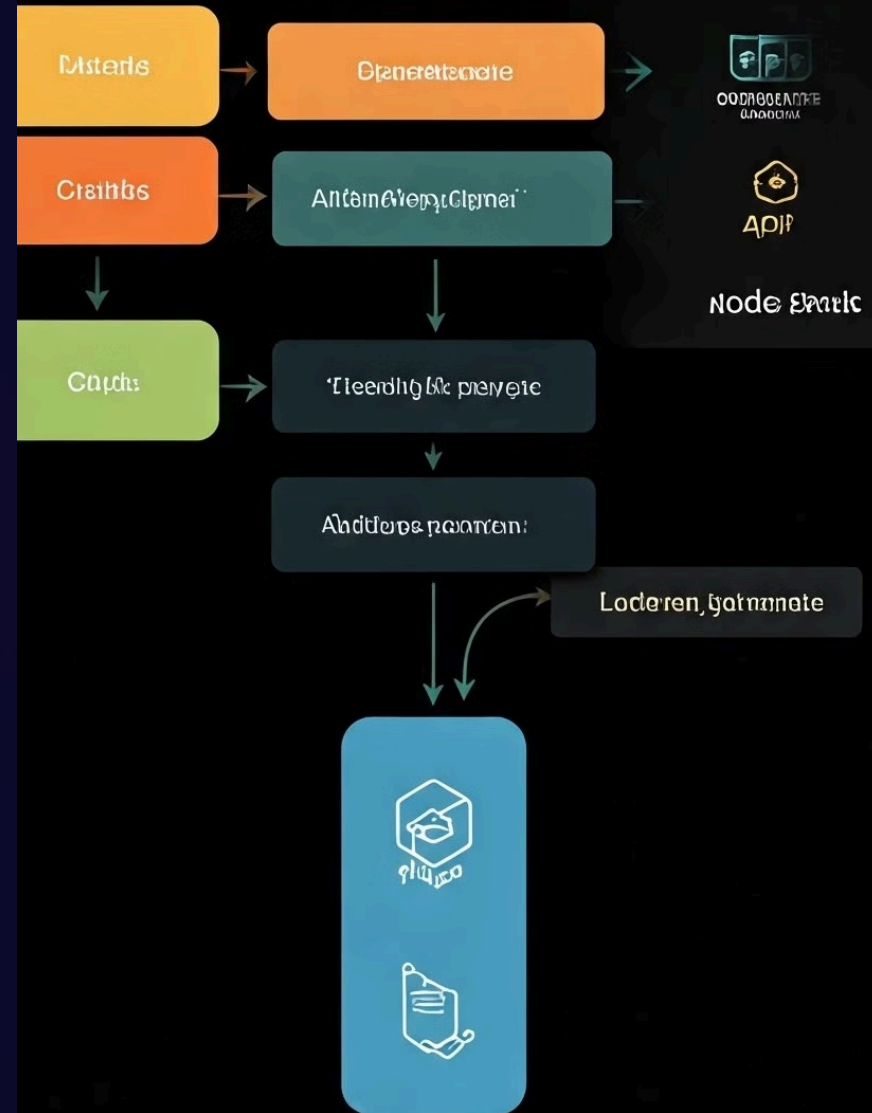
## 실시간 통신

Node.js는 WebSockets와 Socket.IO를 사용하여 실시간 애플리케이션을 구축하는 데 뛰어납니다.

3

## 확장성

Node.js의 비차단 I/O 모델을 통해 매우 확장 가능하고 효율적인 웹 애플리케이션을 만들 수 있습니다.





# Node.js의 실제 사용 사례



## 채팅 애플리케이션

Node.js의 실시간 통신 기능은 채팅 앱 구축에 적합합니다.



## 백엔드 서비스

Node.js를 사용하여 확장 가능하고 효율적인 백엔드 서비스와 API를 구축할 수 있습니다.



## 데이터 스트리밍

Node.js의 이벤트 기반 특성은 데이터 스트리밍 애플리케이션 구축에 적합합니다.



# Node.js를 위한 필수 모듈 및 프레임워크

## Express.js

웹 및 모바일 애플리케이션 구축을 위한 인기 있는 웹 애플리케이션 프레임워크.

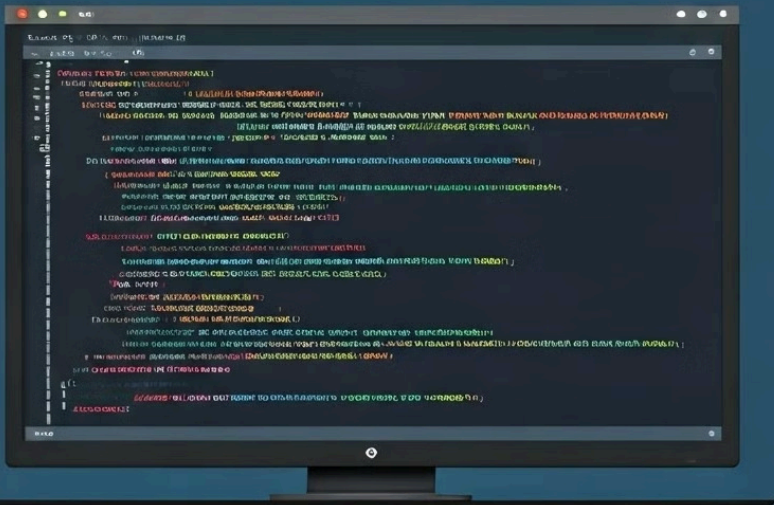
## Socket.IO

실시간, 양방향, 이벤트 기반 통신을 위한 라이브러리.

## Mongoose

MongoDB를 위한 Object Data Modeling (ODM) 라이브러리로, 더 높은 수준의 추상화를 제공합니다.

# Node.js 개발 환경 설정



1

## Node.js 설치

공식 웹사이트에서 최신 버전의 Node.js를 다운로드하여 설치하세요.

2

## 편집기 선택

Visual Studio Code, Sublime Text, Atom 등의 코드 편집기를 사용하여 Node.js 코드를 작성하세요.

3

## 프로젝트 설정

새 디렉토리를 만들고, Node.js 프로젝트를 초기화한 후 필요한 모듈을 설치하세요.