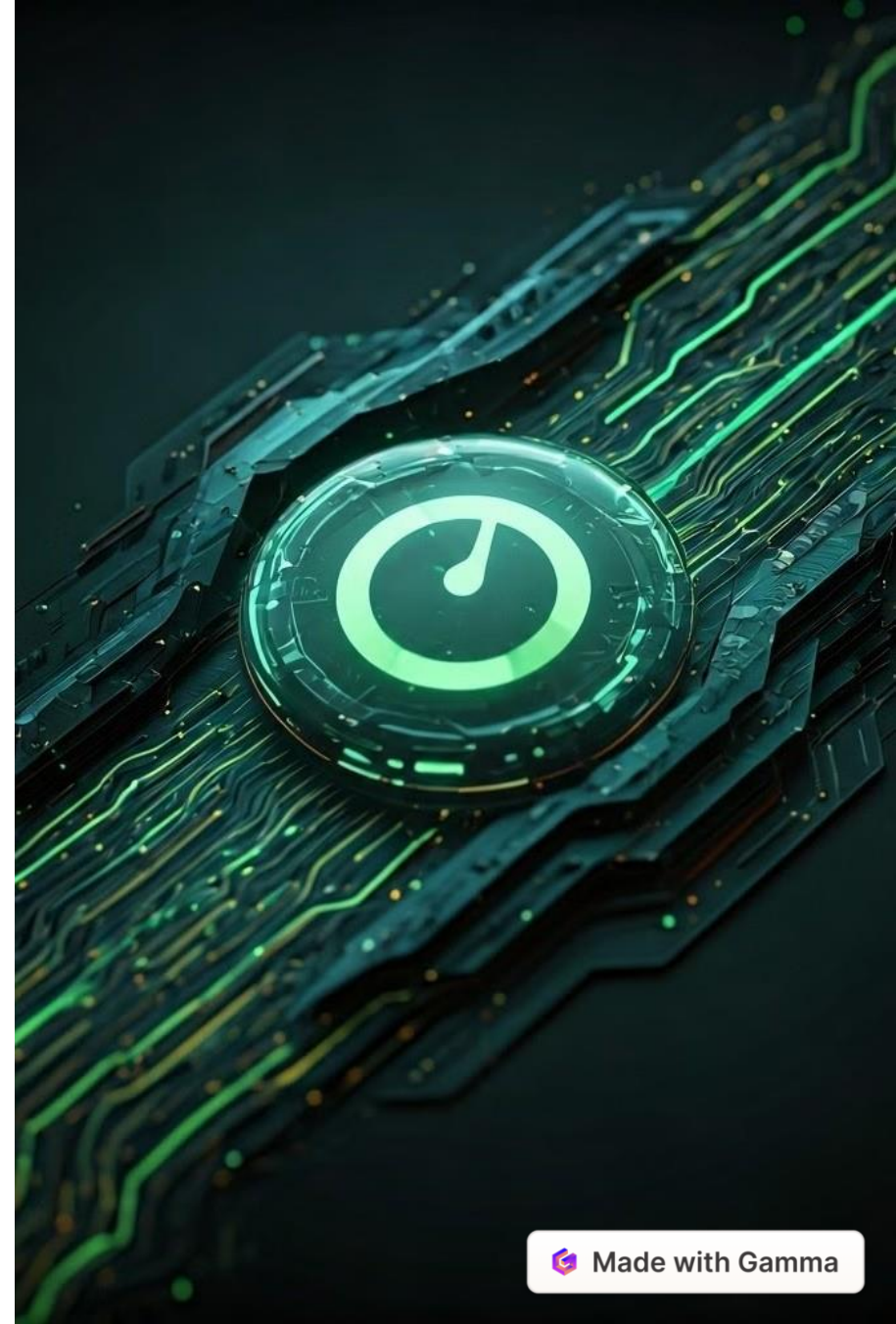


# Node.js: 강력한 서버 측 JavaScript 런타임

Node.js는 Chrome의 V8 JavaScript 엔진을 기반으로 한 오픈 소스 서버 측 JavaScript 런타임 환경입니다. 비동기 이벤트 기반 프로그래밍 모델을 사용하여 높은 성능과 확장성을 제공합니다. 이 플랫폼은 웹 서버 구축, 실시간 애플리케이션 개발, 마이크로서비스 아키텍처 구현 등 다양한 용도로 사용됩니다.



A diagram showing a blue circle with the text "Event loop" inside. A curved arrow on the left side of the circle indicates a continuous loop.

Event  
loop

## Node.js의 주요 특징

1

### 비동기 I/O

Node.js는 비동기적으로 작업을 처리하여 높은 처리량을 유지합니다. 이를 통해 서버가 요청을 처리하는 동안 다른 작업을 수행할 수 있습니다.

2

### 단일 스레드

Node.js는 단일 스레드 모델을 사용하지만, 이벤트 루프를 통해 여러 클라이언트의 요청을 동시에 처리할 수 있습니다.

3

### NPM (Node Package Manager)

방대한 패키지 생태계를 갖춘 NPM을 통해 다양한 라이브러리와 모듈을 쉽게 설치하고 관리할 수 있습니다.

# Node.js의 장점

## 빠른 성능

V8 엔진 덕분에 JavaScript 코드가 매우 빠르게 실행됩니다.

## 확장성

비동기 I/O 모델 덕분에 대규모 애플리케이션을 쉽게 확장할 수 있습니다.

## JavaScript 사용

프론트엔드와 백엔드 모두에서 JavaScript를 사용할 수 있어 개발자들이 동일한 언어로 작업할 수 있습니다.

# Node.js의 사용 사례



## 웹 서버

Express.js와 같은 프레임워크를 사용하여 RESTful API 및 웹 애플리케이션을 구축할 수 있습니다.



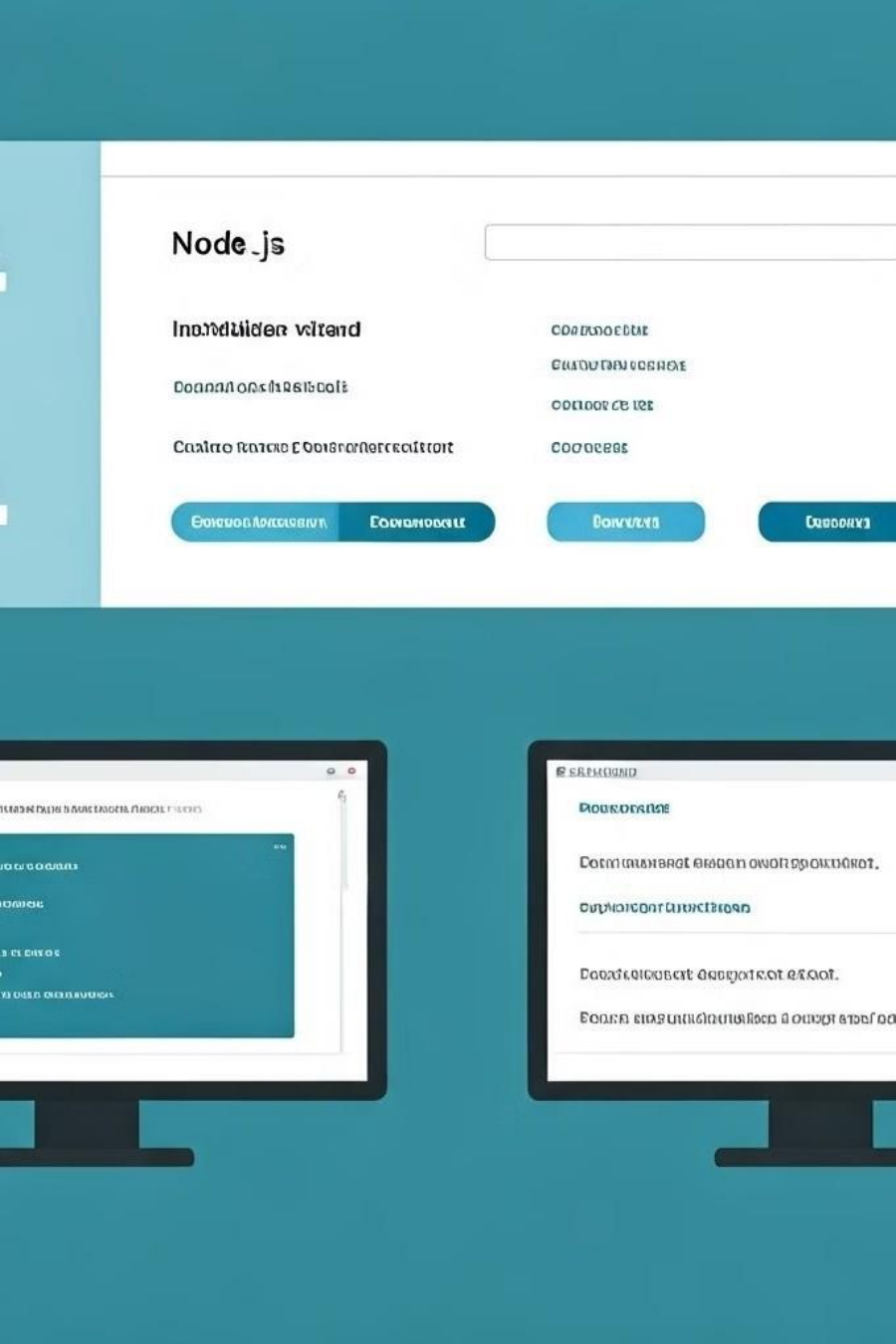
## 실시간 애플리케이션

채팅 애플리케이션, 게임 서버 등 실시간 데이터 처리가 필요한 애플리케이션에 적합합니다.



## 마이크로서비스

Node.js의 경량성과 빠른 성능 덕분에 마이크로서비스 아키텍처에 적합합니다.



# Node.js 설치 방법

1

## Node.js 다운로드

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

2

## 설치

다운로드한 파일을 실행하여 설치 과정을 완료합니다.

3

## 확인

터미널 또는 명령 프롬프트에서 `node -v`와 `npm -v` 명령어를 입력하여 설치가 제대로 되었는지 확인합니다.



# Node.js의 비동기 프로그래밍

Node.js의 핵심 특징 중 하나는 비동기 프로그래밍 모델입니다. 이 모델은 콜백 함수, 프로미스, `async/await` 구문을 통해 구현됩니다. 비동기 프로그래밍은 I/O 작업이 완료될 때까지 기다리지 않고 다른 작업을 수행할 수 있게 해주어 애플리케이션의 전반적인 성능을 향상시킵니다.

1

## 콜백 함수

전통적인 비동기 처리 방식으로, 작업 완료 후 실행될 함수를 인자로 전달합니다.

2

## 프로미스

비동기 작업의 최종 완료 또는 실패를 나타내는 객체로, 체이닝을 통해 가독성을 높입니다.

3

## Async/Await

프로미스를 기반으로 한 더 직관적인 비동기 처리 구문으로, 동기 코드처럼 작성할 수 있습니다.



# Node.js 생태계와 N P M

Node.js의 강점 중 하나는 풍부한 생태계와 NPM(Node Package Manager)입니다. NPM은 세계 최대의 소프트웨어 레지스트리로, 수많은 오픈 소스 패키지를 제공합니다. 개발자들은 NPM을 통해 필요한 라이브러리와 도구를 쉽게 설치하고 관리할 수 있습니다.

패키지 수	1,000,000+
주간 다운로드	수십억 건
사용자	전 세계 개발자



# Node.js의 미래와 발전 방향

Node.js는 지속적으로 발전하고 있으며, 성능 개선, 보안 강화, 새로운 기능 추가 등을 통해 더욱 강력한 플랫폼으로 성장하고 있습니다. 특히 마이크로서비스 아키텍처, 서버리스 컴퓨팅, IoT(사물인터넷) 등의 분야에서 Node.js의 활용도가 높아지고 있습니다.



## IoT와 Node.js

Node.js는 경량성과 효율성으로 IoT 디바이스 프로그래밍에 적합합니다.



## 서버리스 컴퓨팅

Node.js는 서버리스 환경에서 빠른 시작 시간과 효율적인 리소스 사용으로 주목받고 있습니다.