

# Node.js란 무엇인가

Node.js는 Chrome V8 JavaScript 엔진 기반의 오픈 소스 JavaScript 런타임 환경입니다. 이를 통해 개발자들은 서버 측에서 JavaScript를 사용할 수 있게 되었으며, 단일 언어로 클라이언트와 서버를 모두 구축할 수 있습니다.

 작성자: 용호 김



# Node.js의 주요 특징

## 1 비동기 I/O

Node.js는 비차단적 I/O 모델을 사용하여 높은 처리량과 확장성을 제공합니다.

## 2 이벤트 기반 아키텍처

Node.js는 이벤트 루프 기반의 아키텍처를 채택하여 실시간 웹 애플리케이션 개발에 최적화되어 있습니다.

## 3 단일 언어 사용

Node.js를 사용하면 클라이언트와 서버 간에 JavaScript를 사용할 수 있어 생산성이 향상됩니다.

# Node.js의 장점

## 빠른 개발 속도

Node.js의 비동기 프로그래밍 모델을 사용하면 빠른 개발과 배포가 가능합니다.

## 높은 확장성

Node.js는 단일 프로세스로 시작하여 필요에 따라 쉽게 확장할 수 있습니다.

## 풍부한 에코시스템

Node.js의 대규모 오픈 소스 커뮤니티가 제공하는 수많은 모듈과 라이브러리를 활용할 수 있습니다.



# Node.js의 주요 사용 사례

## 실시간 웹 애플리케이션

채팅, 실시간 게임, 온라인 협업 도구 등을 구축하는 데 적합합니다.

## API 서버

Node.js의 높은 성능과 확장성을 활용하여 REST API 서버를 구축할 수 있습니다.

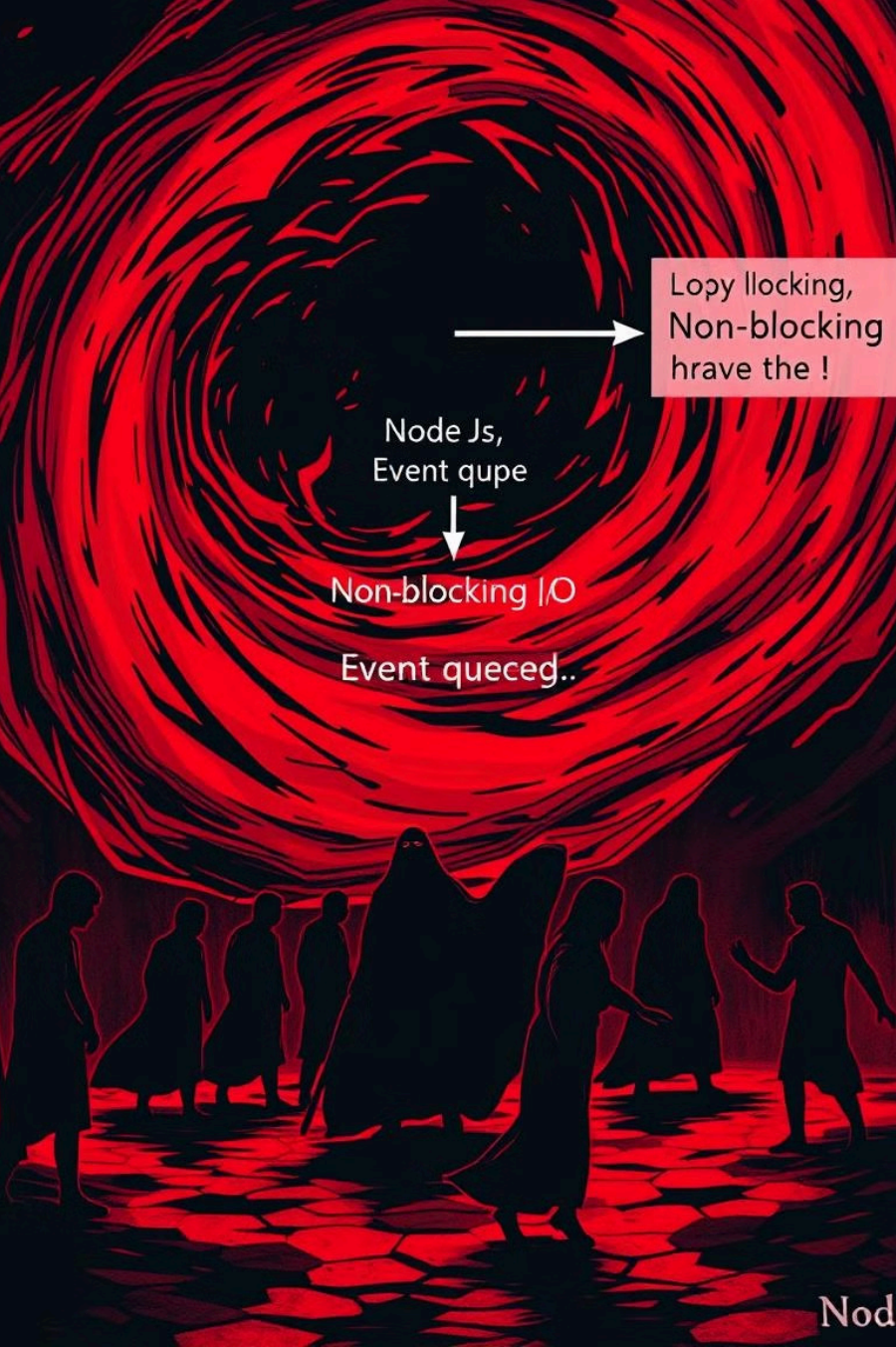
## 데이터 스트리밍

Node.js의 비동기 I/O 처리 능력을 활용하여 데이터 스트리밍 애플리케이션을 만들 수 있습니다.

## IoT 및 엣지 컴퓨팅

낮은 리소스 요구사항으로 인해 IoT 디바이스와 엣지 컴퓨팅에 적합합니다.





Copy locking,  
Non-blocking  
have the !

Node Js,  
Event qupe

Non-blocking I/O  
Event queced..

# Node.js의 비동기 I/O 처리

1

요청 수락

Node.js 서버는 들어오는 요청을 비동기적으로 수락합니다.

2

이벤트 큐

요청은 이벤트 큐에 배치되어 순차적으로 처리됩니다.

3

이벤트 루프

이벤트 루프는 이벤트 큐의 작업을 지속적으로 감시하고 실행합니다.

# Node.js의 모듈 시스템



## CommonJS

Node.js는 기본적으로 CommonJS 모듈 시스템을 사용합니다.



## 모듈 로딩

require() 함수를 사용하여 모듈을 로드하고 의존성을 관리할 수 있습니다.



## 모듈 내보내기

module.exports 객체를 통해 모듈 내의 기능을 내보낼 수 있습니다.



## npm 패키지 관리

강력한 Node.js 패키지 관리자인 npm을 통해 다양한 모듈을 설치하고 관리할 수 있습니다.

# Node.js의 이벤트 기반 아키텍처

1

## 이벤트 발생

Node.js 애플리케이션은 다양한 이벤트를 처리합니다.

2

## 이벤트 큐

발생한 이벤트는 이벤트 큐에 추가되어 순차적으로 처리됩니다.

3

## 이벤트 루프

이벤트 루프가 이벤트 큐를 지속적으로 모니터링하고 처리합니다.





# Node.js 개발 환경 설정 및 시작하기

## Node.js 설치

공식 웹사이트에서 Node.js를 다운로드하고 설치하세요.

## 코드 편집기

Visual Studio Code, Sublime Text, Atom 등의 편집기를 사용하세요.

## 첫 프로젝트 생성

간단한 Hello World 애플리케이션을 만들어 보세요.