



Node.js란 무엇인가?

JavaScript 기반의 오픈 소스 서버 사이드 런타임 환경

웹 서버 및 네트워크 애플리케이션 개발에 사용



작성자: 최원욱

학번: 2020575068



Node.js의 등장 배경과 특징

웹 애플리케이션 개발의 비동기 I/O 처리 필요성

높은 성능과 확장성을 제공

1 단일 스레드 이벤트 루프

비동기 방식으로 다량의 요청 처리 가능

3 대규모 커뮤니티 지원

활발한 개발자 커뮤니티와 풍부한 라이브러리

2 패키지 관리 시스템

npm(Node Package Manager)을 통한 모듈 관리

4 다양한 플랫폼 지원

Windows, Linux, macOS 등 다양한 운영 체제 지원

Node.js의 비동기 I/O와 이벤트 기반 모델

비동기 I/O 방식을 통해 동시에 여러 요청 처리

이벤트 루프를 사용하여 비동기 작업 관리

1

요청 수신

클라이언트에서 요청이 들어오면 이벤트 루프에 등록

2

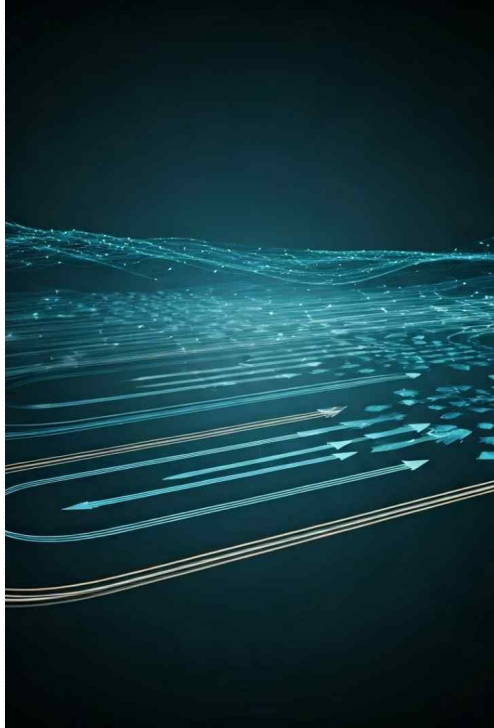
비동기 I/O 실행

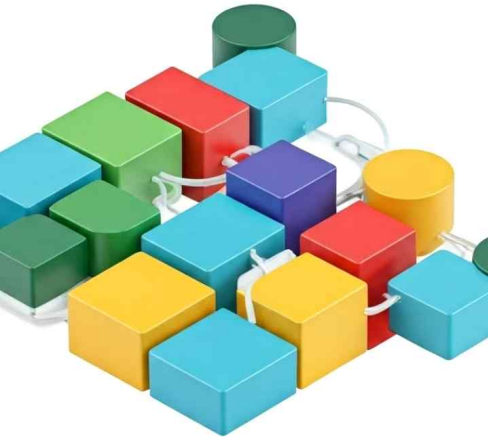
I/O 작업이 완료되면 콜백 함수 호출

3

콜백 함수 실행

이벤트 루프는 콜백 함수를 실행하여 응답 생성





Node.js의 핵심 모듈 이해하기

모듈화를 통해 코드 재사용 및 관리 용이

핵심 모듈은 Node.js의 기본 기능 제공

모듈	설명
fs	파일 시스템 조작
http	웹 서버 구축
net	네트워크 통신
url	URL 파싱 및 조작

다양한 Node.js 예제 살펴보기

웹 서버, API, CLI 애플리케이션 개발 예시

실제 프로젝트에서 활용되는 다양한 코드 예시 제공



웹 서버

웹 페이지 제공



API

데이터 교환



CLI

명령줄 인터페이스



Node.js의 개발 환경 구축하기

Node.js 설치 및 개발 도구 설정

개발 환경 설정을 통해 프로젝트 시작



1

Node.js 설치

Node.js 공식 웹사이트에서 다운로드

2

패키지 관리자 설치

npm(Node Package Manager) 설치

3

텍스트 에디터 선택

Visual Studio Code, Atom 등 사용

Node.js로 웹 어플리케이션 개발하기

웹 서버 구축 및 API 개발

동적인 웹 콘텐츠 생성 및 데이터 관리

Express.js

웹 프레임워크, 라우팅, 미들웨어 제공

MongoDB

NoSQL 데이터베이스, 문서 기반 저장

React

JavaScript 라이브러리, 사용자 인터페이스 개발

Node.js의 미래와 발전 방향

IoT, 엣지 컴퓨팅 분야 적용 확대

웹 개발 기술의 지속적인 발전과 함께 성장

Serverless Computing

서버 관리 없이 코드 실행 가능

WebAssembly

다양한 언어로 작성된 코드 실행
지원

AI 및 머신러닝

Node.js 기반 AI 애플리케이션 개발

