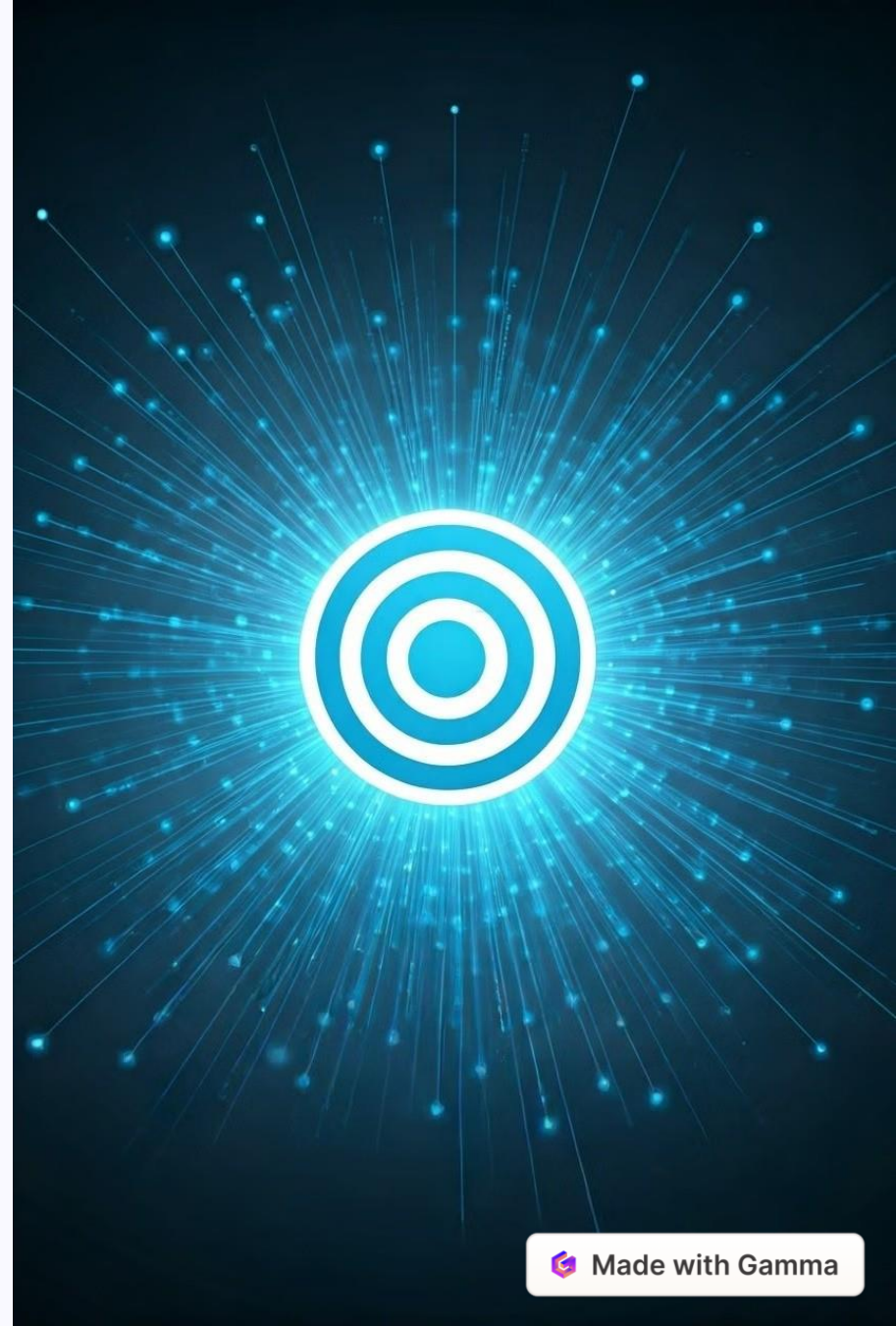
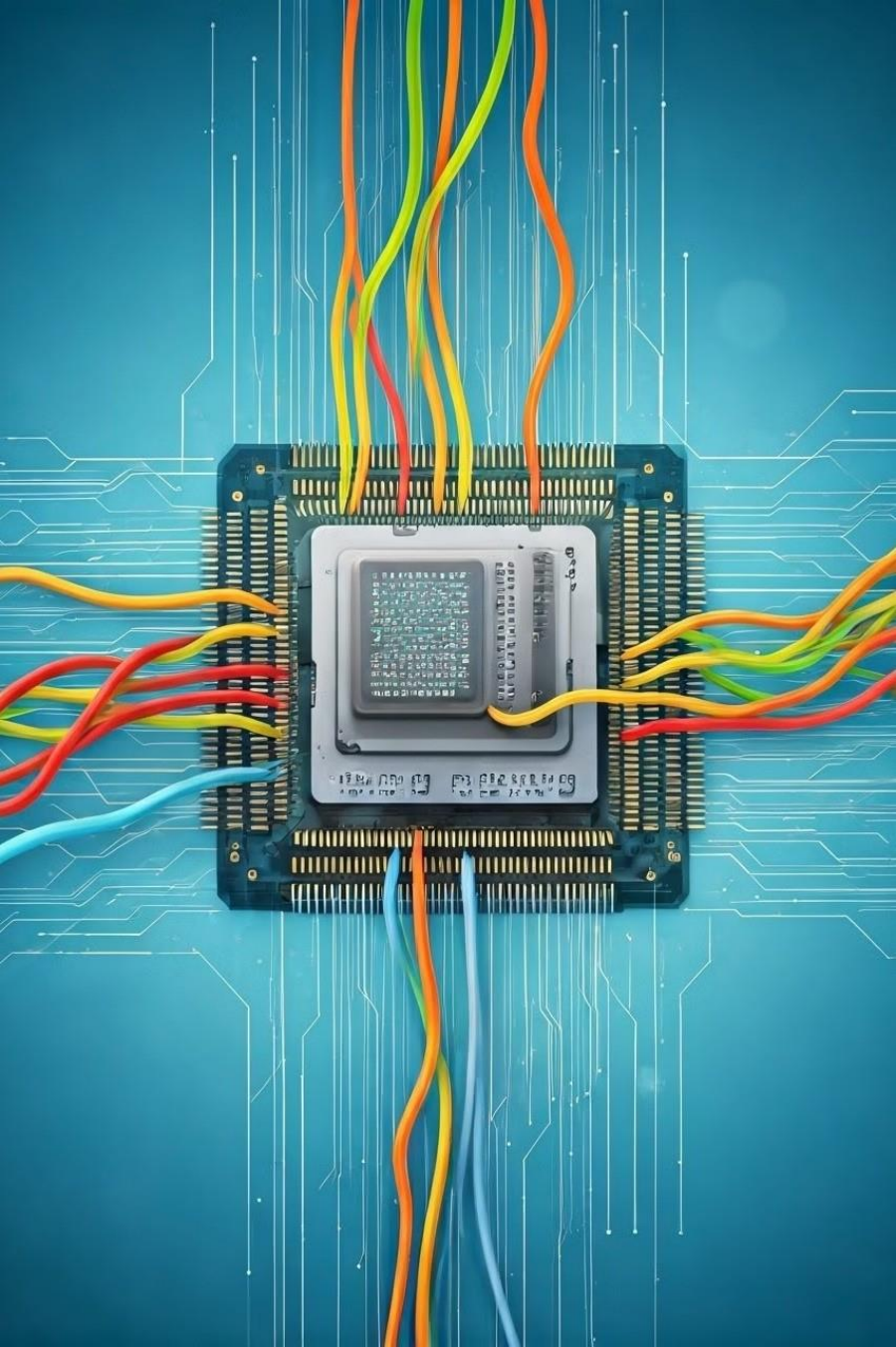


Node.js: 현대 웹 개발 의 핵심

Node.js는 서버 측 JavaScript 실행 환경입니다. 비동기 이벤트 기반 프로그래밍을 지원하며, 웹 애플리케이션 개발에 혁신을 가져왔습니다. 높은 성능과 확장성으로 현대 웹 개발의 중심에 있습니다.





비동기 I/O: Node.js의 핵심

1

요청 접수

서버가 클라이언트로부터 요청을 받습니다. 비동기 처리로 즉시 다음 요청을 받을 수 있습니다.

2

비차단 처리

I/O 작업 중에도 다른 작업을 수행합니다. 리소스를 효율적으로 사용하여 성능을 향상시킵니다.

3

콜백 실행

I/O 작업이 완료되면 콜백 함수가 실행됩니다. 이벤트 루프가 이를 관리합니다.



이벤트 기반 아키텍처

1

이벤트 루프

Node.js의 핵심메커니즘입니다.
비동기 작업을 효율적으로 관리하
고 처리합니다.

2

이벤트 에미터

커스텀 이벤트를 생성하고 처리할
수 있습니다. 모듈 간 통신을 용이
하게 합니다.

3

콜백 큐

완료된 작업의 콜백을 저장합니다.
이벤트 루프가 순차적으로 처리합
니다.

단일 스레드 모델의 장점

효율적인 메모리 사용

단일 스레드로 메모리 오버헤드를 줄입니다. 리소스를 효율적으로 관리할 수 있습니다.

간단한 프로그래밍 모델

멀티스레딩의 복잡성을 피할 수 있습니다. 동시성 문제를 쉽게 해결할 수 있습니다.

높은 확장성

비동기 I/O로 많은 연결을 처리합니다. 클러스터 모듈로 멀티코어 활용이 가능합니다.



npm: 방대한 생태계



패키지 설치

npm install 명령어로 쉽게 패키지를 설치할 수 있습니다. 의존성 관리가 자동화됩니다.



버전 관리

package.json 파일로 버전을 관리합니다. 프로젝트의 일관성을 유지할 수 있습니다.



패키지 배포

자신의 패키지를 npm에 배포할 수 있습니다. 커뮤니티에 기여할 수 있습니다.

Node.js의 주요장점

빠른 성능

V8 엔진의 최적화된 성능으로 빠른 실행 속도를 제공합니다. 실시간 애플리케이션에 적합합니다.

뛰어난 확장성

비동기 처리로 많은 동시 연결을 처리합니다. 클러스터링으로 수평적 확장이 가능합니다.

풀스택 JavaScript

프론트엔드와 백엔드에서 동일한 언어를 사용합니다. 개발 효율성이 크게 향상됩니다.

활발한 커뮤니티

광범위한 지원과 풍부한 자료를 제공합니다. 문제 해결과 학습이 용이합니다.



Node.js의 주요 사용 사례

1

실시간 애플리케이션

채팅 앱, 온라인 게임 등에 적합합니다. 웹소켓을 통한 실시간 통신을 지원합니다.

2

RESTful API

Express.js 등의 프레임워크로 API를 쉽게 구축합니다. JSON 처리에 효율적입니다.

3

마이크로서비스

독립적인 서비스 구축에 적합합니다. 각 서비스의 독립적인 배포와 확장이 가능합니다.

Node.js 개발 시작하기



단계

설명

1. 설치

공식 웹사이트에서 Node.js를 다운로드하고 설치합니다.

2. 프로젝트 초기화

npm init 명령어로 새 프로젝트를 시작합니다.

3. 의존성 설치

필요한 패키지를 npm install로 설치합니다.

4. 코딩 시작

app.js 파일을 만들고 서버 코드를 작성합니다.