Node.js 소개

Node.js는 자바스크립트 실행 환경으로, 웹 애플리케이션 개발을 위한 강력한 플랫폼입니다. 비동기 처리와 이벤트 기반 아키텍처를 통해 빠르고 효율적인 서버 애플리케이션을 구축할 수 있습니다.

Node.js의 특징과 장점

빠른 속도

비동기 I/O와 단일 스레드 이벤트 루프로 인해 높은 처리량과 낮은 대기 시간을 자랑합니다.

확장성

CPU 집약적인 작업을 별도의 프로세스로 처리하여 확장성과 복원력을 높일 수 있습니다.

생산성

JavaScript 생태계의 풍부한 라이브러리와 도구를 활용할 수 있어 개발 생산성이 높습니다.

Node.js의 기본 구조와 동작 원리

1 단일 스레드 이벤트 루프

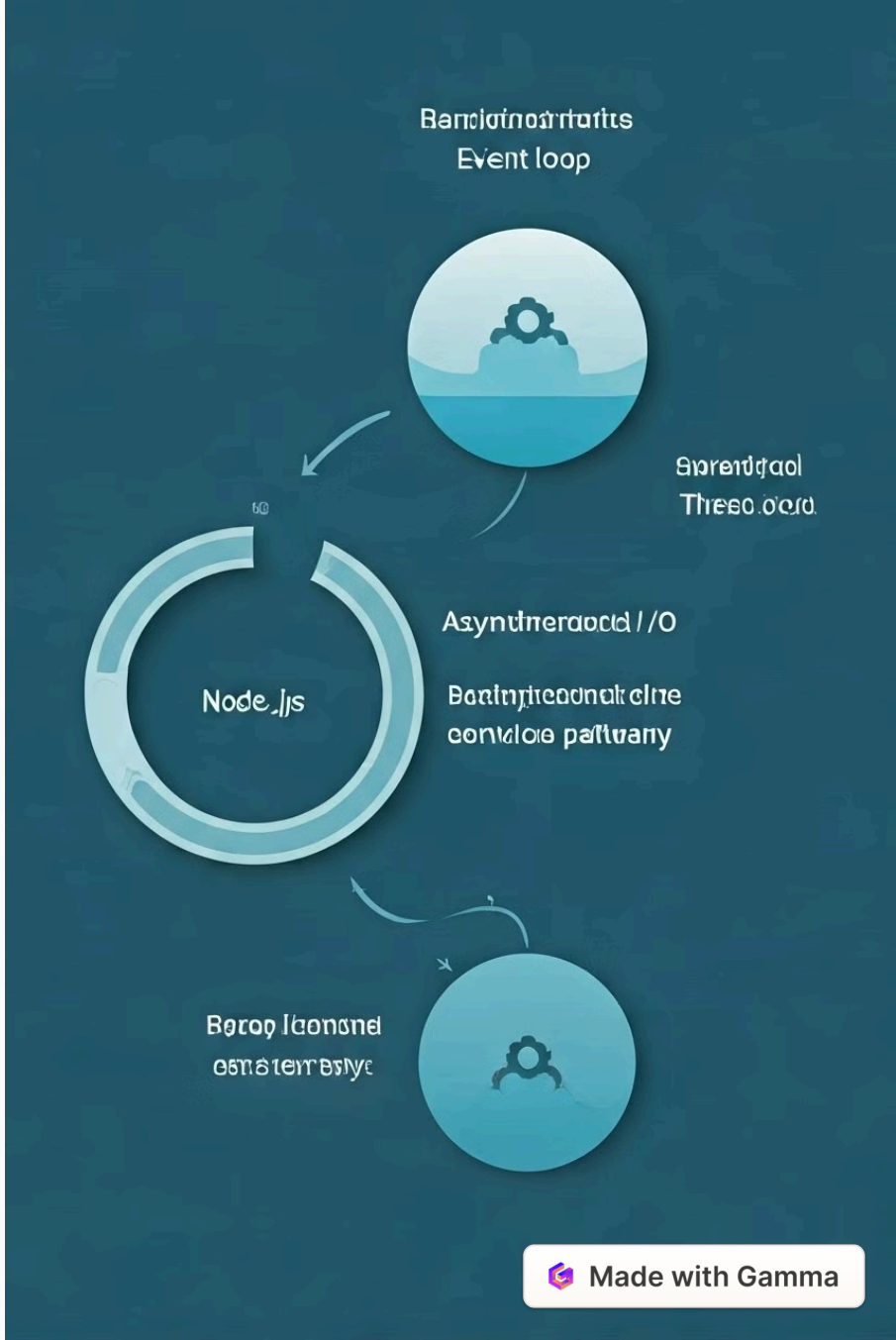
이벤트 기반으로 동작하며, 비동기 I/O를 통해 CPU 부하를 최소화합니다.

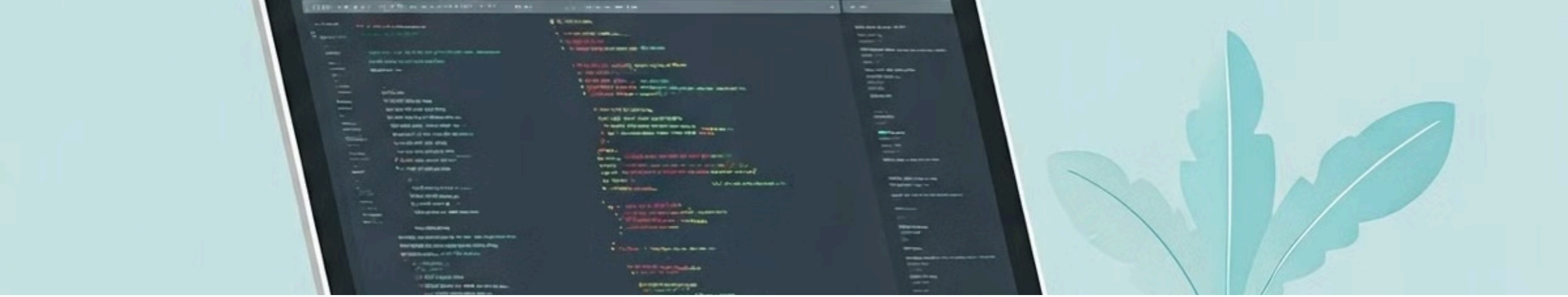
2 스레드 풀

CPU 집약적인 작업은 별도의 스레드 풀에서 처리하여 성능을 높입니다.

3 모듈 시스템

Node.js는 CommonJS 모듈 시스템을 사용하여 코드의 재사용성과 확장성을 지원합니다.





Node.js 코드 이해하기

서버 구축

Node.js로 간단한 웹 서버를 만들 수 있습니다.

파일 처리

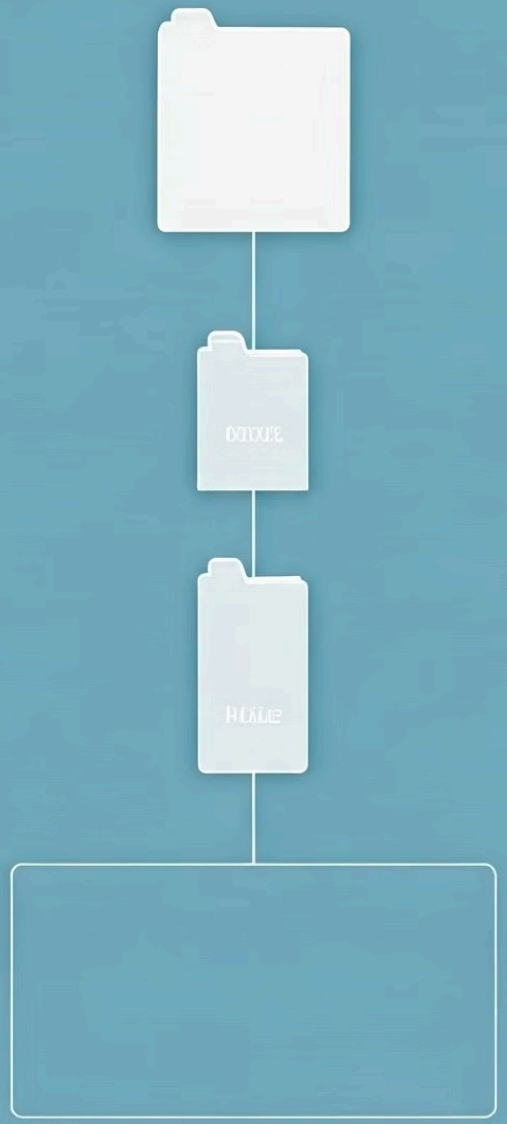
Node.js의 내장 fs 모듈로 파일 읽기/쓰기 작업을 수행할 수 있습니다.

HTTP 통신

Node.js의 내장 http 모듈로 HTTP 서버와 클라이언트를 구현할 수 있습니다.

이벤트 처리

Node.js의 내장 events 모듈로 이벤트 기반 프로그래밍을 할 수 있습니다.



Node.js 프로젝트 구조 및 파일 관리

1

프로젝트 구조

일반적으로 src, tests, config, routes 등의 폴더로 구성됩니다.

2

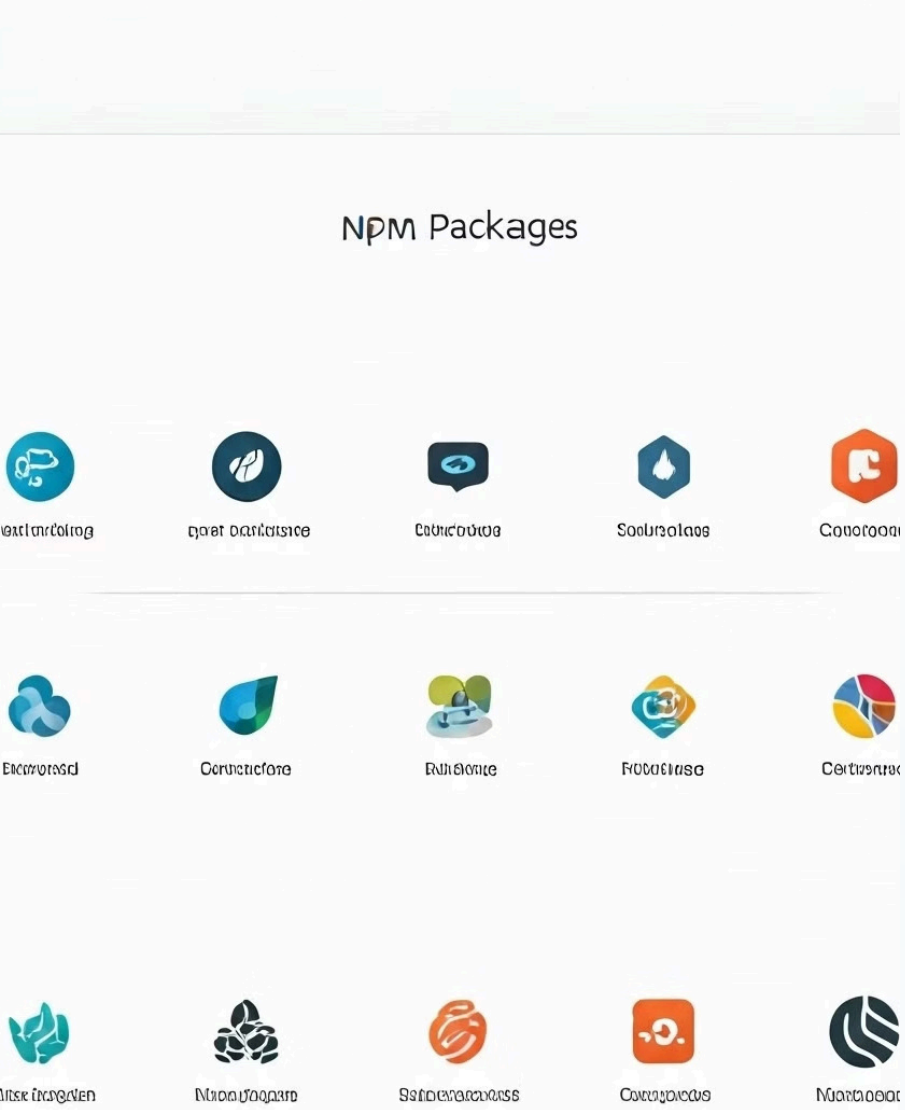
파일 관리

모듈화된 구조로 관리하여 코드의 가독성과 유지보수성을 높일 수 있습니다.

3

의존성 관리

package.json 파일로 프로젝트의 종속성을 관리하고 배포할 수 있습니다.



Node.js 모듈과 NPM 패키지

내장 모듈

Node.js에는 다양한 내장 모듈이 있어 기본 기능을 제공합니다.

NPM 패키지

Node.js 생태계의 방대한 오픈소스 라이브러리를 활용할 수 있습니다.

모듈 생성

필요에 따라 직접 모듈을 만들어 프로젝트에 통합할 수 있습니다.

Node.js 비동기 프로그래밍

1

콜백 함수

비동기 작업의 완료를 알리기 위해 콜백 함수를 사용합니다.

2

프로미스

콜백 지옥을 해결하기 위해 프로미스 객체를 사용할 수 있습니다.

3

async/await

비동기 코드를 더욱 깔끔하게 작성할 수 있습니다.

Node.js 프로젝트 배포 및 관리

1

배포 방식

Docker, AWS Lambda, Heroku 등 다양한 방식으로 배포할 수 있습니다.

2

모니터링

PM2, StrongLoop 등의 도구로 프로세스 관리와 모니터링이 가능합니다.

3

로깅

Winston, Morgan 등의 로깅 라이브러리를 사용하여 로그를 관리할 수 있습니다.

4

보안

helmet, express-validator 등으로 Node.js 애플리케이션의 보안을 강화할 수 있습니다.