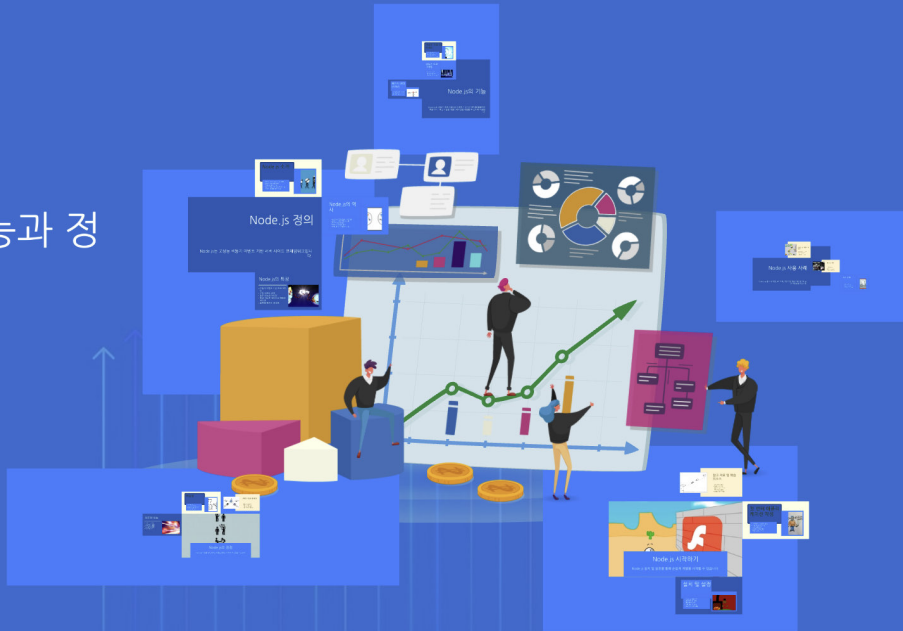


# Node.js의 기능과 정의

Node.js 개요 및 활용법

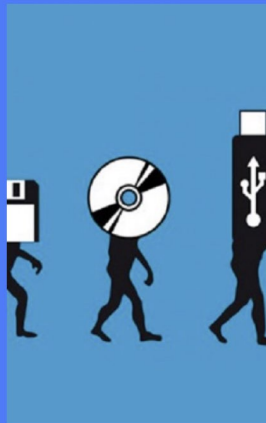


# Node.js 정의

Node.js는 고성능 비동기 이벤트 기반 서버 사이드 프레임워크입니다.

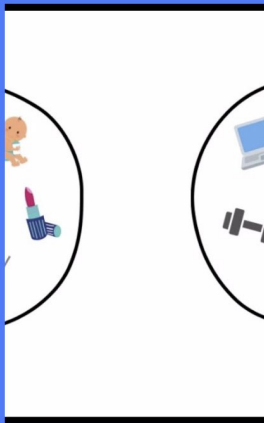
# Node.js 소개

- 서버 사이드 JavaScript 실행
- 비동기 I/O 모델 사용
- 구글 V8 엔진 기반
- 자바스크립트로 서버 구축 가능
- 크로스 플랫폼 개발 지원



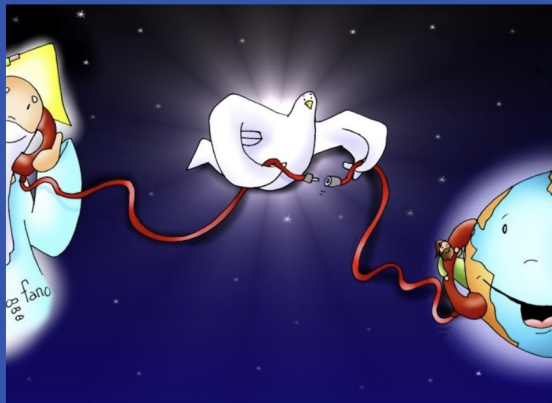
# Node.js의 역사

- 2009년 라이언 달에 의해 개발
- 오픈 소스 프로젝트로 전환
- 2011년 NPM 출시
- 커뮤니티 참여로 성장
- 현재 다수의 기업에서 사용



# Node.js의 특징

- 비동기 이벤트 기반 프로그래밍
- 단일 스레드 모델
- 높은 성능과 처리량
- 확장 가능한 네트워크 애플리케이션
- 풍부한 패키지 생태계

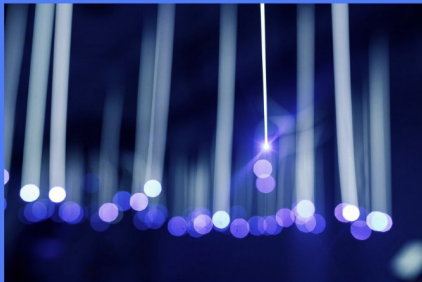


# Node.js의 기능

Node.js는 비동기 프로그래밍과 이벤트 기반 아키텍처를 활용하여 효율적이고 확장 가능한 애플리케이션을 개발할 수 있도록 지원한다.

# 비동기 프로그래밍

- 요청 처리 및 응답 대기
- 블로킹 없는 코드 실행
- 성능 향상 및 응답 시간 단축
- 콜백, 프로미스 및 `async/await` 사용
- 대규모 데이터베이스 작업 지원



# 이벤트 기반 아키텍처

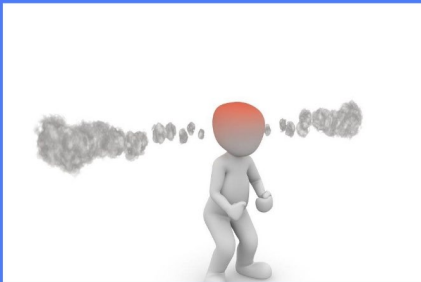
- 이벤트 루프를 통한 처리
- 동시성과 효율적 리소스 사용
- 실시간 데이터 전송 용이
- 다양한 이벤트 핸들링 지원
- 대규모 애플리케이션에서 유용함





# 패키지 관리 (NPM)

- Node.js 패키지 관리 도구
- 다양한 서드파티 패키지 이용 가능
- 쉽게 의존성 관리 가능
- 커뮤니티에 의해 관리되고 업데이트
- 모듈 시스템을 통한 코드 재사용





# Node.js의 장점

Node.js는 탁월한 속도, 뛰어난 확장성, 강력한 커뮤니티로 인정받고 있습니다.

# 속도와 성능

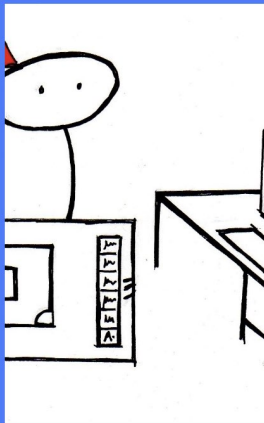
---

- 비동기 I/O 처리
- 단일 스레드 모델
- 이벤트 루프 활용
- 높은 처리량 제공
- 빠른 개발 주기



# 확장성

- 수평 확장 지원
- 클러스터링 기능 제공
- 마이크로서비스 아키텍처 적용
- 다양한 플랫폼에서 실행 가능
- 높은 동시 사용자 처리





# 커뮤니티와 생태계

- 활발한 개발자 커뮤니티
- 풍부한 모듈과 패키지
- 오픈 소스 프로젝트 지원
- 다양한 학습 자료 제공
- 협업과 스프린트 활동 활발

# Node.js 사용 사례

Node.js는 웹 서버 개발, API 구축, 리얼타임 애플리케이션으로 높은 유용성을 보입니다.

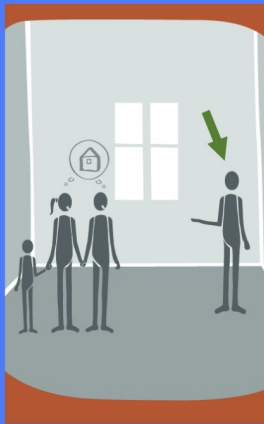


# 웹 서버 개발

- 고성능 웹 서버 지원
- 비동기 처리로 빠름
- 많은 동시 연결 처리 가능
- RESTful API 통합 용이
- 미들웨어 사용으로 유연함

# API 구축

- CRUD 기능 지원
- JSON 데이터 처리 용이
- 다양한 클라이언트 통신 가능
- Swagger로 문서화 간편
- 마이크로서비스 아키텍처 활용

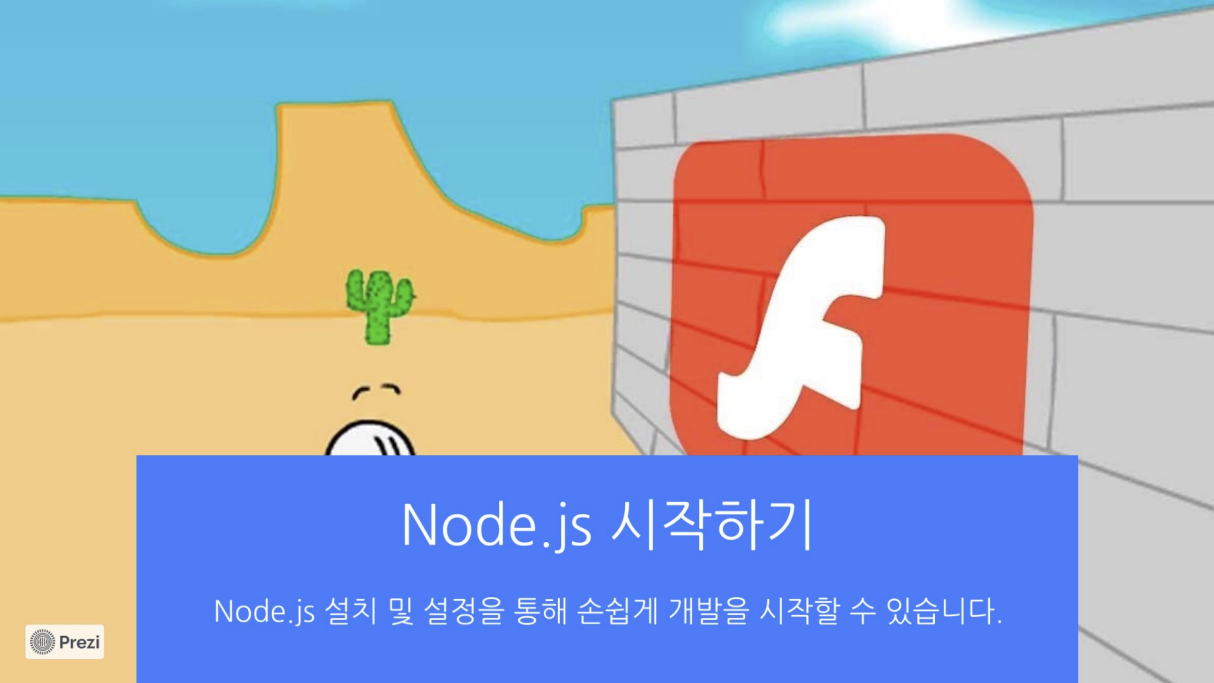






# 리얼타임 애플리케이션

- 웹소켓 지원으로 실시간 통신
- 순차적 데이터 처리 가능
- 게임, 채팅 앱에 적합
- 사용자 경험 향상
- 이벤트 기반 데이터 전송



# Node.js 시작하기

Node.js 설치 및 설정을 통해 손쉽게 개발을 시작할 수 있습니다.

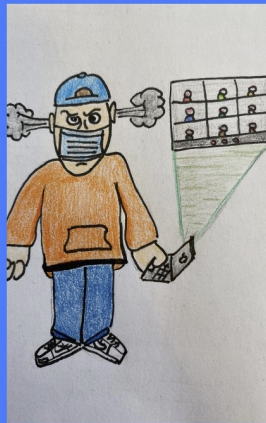
# 설치 및 설정

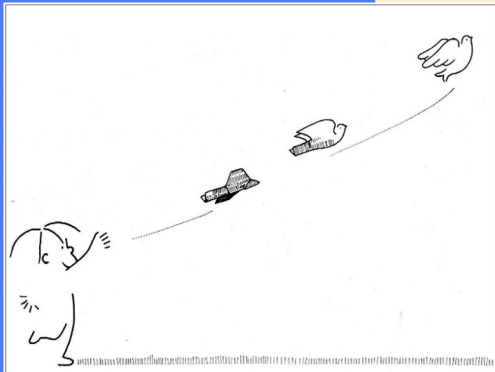
- Node.js 다운로드
- 운영체제 별 설치
- 환경 변수 설정
- 기본 패키지 설치
- 일반적인 오류 해결



# 첫 번째 애플리케이션 작성

- 프로젝트 디렉토리 생성
- 기본 서버 코드 작성
- 서버 실행 방법
- 브라우저에서 확인
- 에러 디버깅 안내





## 참고 자료 및 학습 리소스

- 공식 문서 링크
- 커뮤니티 포럼
- 온라인 강의 소개
- 추천 도서 목록
- 유용한 튜토리얼