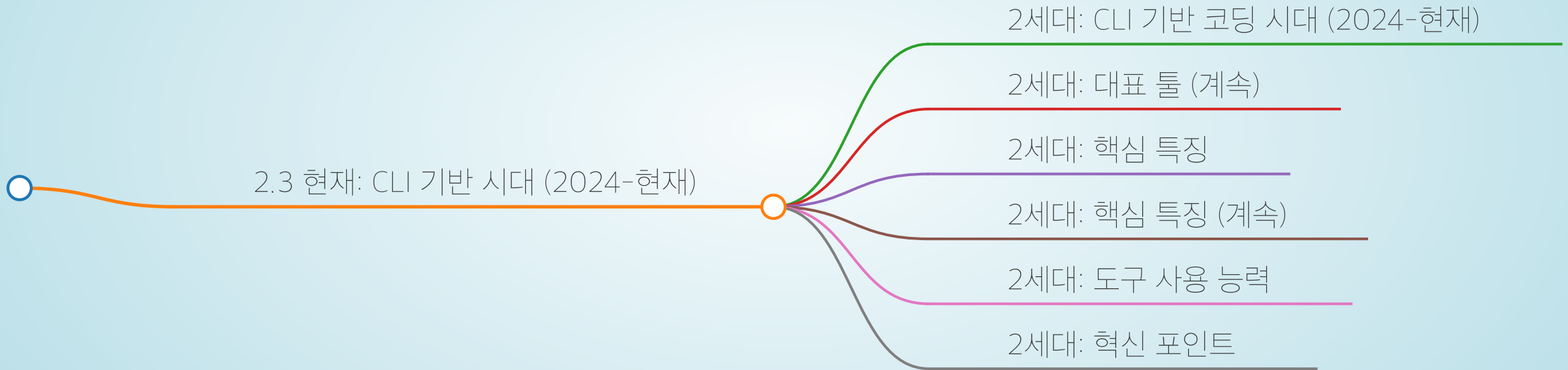


Examples for MarkdownGraph



2세대: CLI 기반 코딩 시대 (2024-현재)

2.3 현재: CLI 기반 시대 (2024-현재)

터미널에서 실행되는 AI 도구

- 에이전트처럼 자율적으로 복잡한 작업 수행

대표 툴

Claude Code

- Anthropic 공식 CLI 도구
- 프로젝트 전체 컨텍스트 이해
- 파일 생성, 수정, 삭제 자동화
- 명령어 실행 및 결과 확인

2세대: 대표 툴 (계속)

2.3 현재: CLI 기반 시대 (2024-현재)

Gemini CLI

- Google의 GPT-4 기반
- 빠른 응답 속도
- 클라우드 기반 실행

Codex CLI

- OpenAI 기반 CLI 도구
- 터미널 기반 코드 생성
- Git 통합 및 자동화

2세대: 핵심 특징

2.3 현재: CLI 기반 시대 (2024-현재)

CLI 기반 실행 환경

- 터미널에서 i 점 실행되널 독립 이로그램
- IDE 없이도 완전인 개발 작업 수행
- 서버 환경에서도 동작 (수수 접속 환경)
- 로컬 리소스 거의 사용 안 함

2세대: 핵심 특징 (계속)

2.3 현재: CLI 기반 시대 (2024-현재)

에이전트형 자율 작업

- (발자 i i 도를 파악하여 다단계 작업 수행
- 필요한 파일 g 스스로 찾아 수정
- 오류 발생 시 자동으로 수정 시도

멀티파일 작업

- 프로젝트 전체를 대상으로 작업
- 파일 간 i 존성 자동 해결
- 일관성 1 는 코드 스타일 유지

2세대: 도구 사용 능력

2.3 현재: CLI 기반 시대 (2024-현재)

Tool Use (Function Calling)

- 사미널 명령어 직의 실행
- 파일 시스템 직의 조작
- Git 명령어 자동 실행
- 패키지 설치 성 관리

대화형 협업

- 자연어파 복잡한 요구사항 전달
- AI가 불명확한 부분 질문
- 반복적인 피드백 미이

Intent-First Programming

개발자: "무엇을(What)" 만들지만 설명

AI: "어떻게(How)" 구현할지 결정

→ 구현 세부사항은 AI에게 위임

Human-in-the-Loop Verification

AI 코드 생성 → 개발자 검토 → 승인/수정 → 다음 단계

End-to-End Automation

- 요구사항 분석부서 배시까함 자동화
- 테스트 코드 자동 생성
- 문서화 자동 작성