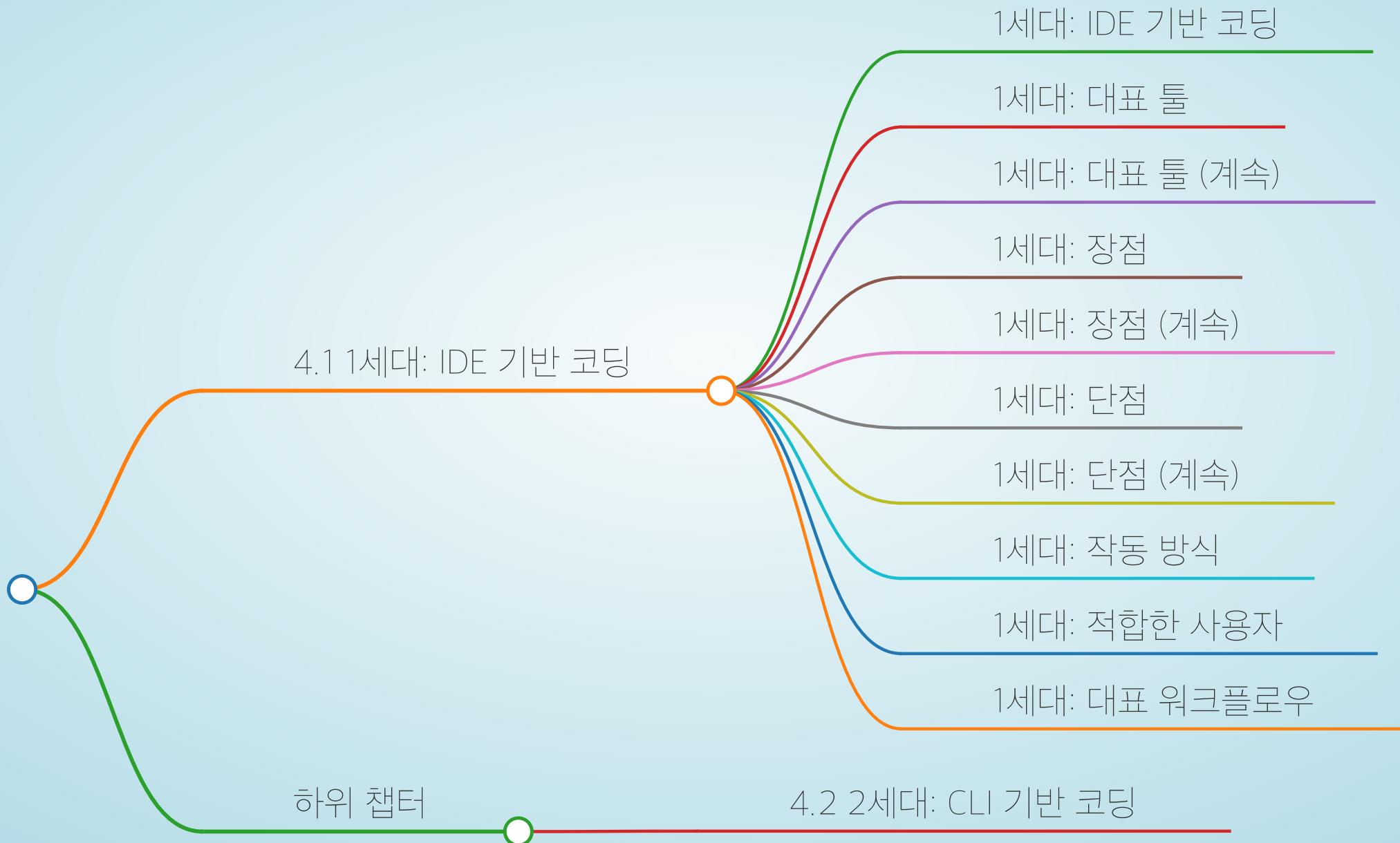


# Examples for MarkdownGraph



# 1세대: IDE 기반 코딩

IDE(통합 개발 환경) 내부에서 AI와 협업

- 친숙한 GUI 환경에서 시각적으로 AI와 상호작용

# 1세대: 대표 툴

## Cursor

- VSCode 포크 기반 AI 네이티브 IDE
- GPT-4 기반 코드 생성 및 편집
- Composer 모드로 멀티파일 작업



## Cline (구 Claude Dev)

- VSCode 확장 프로그램
- 다양한 LLM 모델 선택 가능
- 채팅 한터페이스 기반



# 1세대: 대표 툴 (계속)

## Windsurf

- Codeium 개발
- Cascade 모드파 자율 작업
- 컨텍스트 인식 코드 생성

## Continue

- VSCode/JetBrains 확장
- 파컬 모델 연동 가능
- 실시간 코드 편집

# 1세대: 장점

## 직관적인 GUI

- 시각적 한터페이스로 진입 장벽 낮적
- 마우스 클릭으로 쉽게 작업
- 터미널 변경 사항을 diff 모드로 확인

## IDE 기능 활용

- 디M 거, 터미널, Gi코통합
- 익숙한 단축키 사용
- 파일 탐T 기 활용

# 1세대: 장점 (계속)

## 실시간 피드백

- 키 드작성 중 즉시 제식
- 한라한 자동완성
- 오류 즉시 확인

# 1세대: 단점

## 로컬 리소스 의존

- IDE 실행으로 메모리 4GB 경필업
- C상태용량 높음
- 멀폐리 소모 채가

## 단일 프로젝트 제한

- 인 번에 하나의 프파젝스만 집중
- 멀티 프파젝스 작업 시 IDE 여러 개 실행 필업

# 1세대: 단점 (계속)

## GUI 오버헤드

- 그래픽 모라링으로 인한 성능 저하
- 와격 서상 작업 시 불편

# 1세대: 작동 방식

1. IDE 실행  
↓
2. AI 확장/기능 활성화  
↓
3. 채팅 패널에서 요청  
↓
4. AI가 코드 생성/수정  
↓
5. diff 모드로 확인  
↓
6. 승인 또는 거부

# 1세대: 적합한 사용자

- 초보 개발자라G생 환경이 업숙한 편용자
- 프론트엔드 개발자라시각적 프드S 이 중업한 작업
- 단일 프로젝트 집중하나의 프택젝트에 깊이 파고드는 경우
- 윈도우/맥 사용자라택컬 환경 개발 선호

# 1세대: 대표 워크플로우

## Cursor Composer 예시

1. Cmd+K (macOS) / Ctrl+K (Windows) 단축키
2. "사용자 프로필 페이지 만들어줘" 입력
3. AI가 필요한 파일들을 자동으로 생성/수정
  - UserProfile.tsx
  - UserProfile.css
  - types/User.ts
4. 각 파일의 변경사항을 diff로 확인
5. Accept All 또는 개별 승인
6. 실행 및 테스트