

Claude Code 치트시트

Claude Code를 빠르게 마스터하기 위한 필수 레퍼런스

</> CLAUDECODE-LAB

1. 기본 명령어

<code>claude</code>	대화형 모드 시작
<code>claude -p "..."</code>	원샷 실행 (비대화형)
<code>/help</code>	도움말 표시
<code>/clear</code>	컨텍스트 초기화
<code>/compact</code>	컨텍스트 압축 후 계속
<code>/cost</code>	세션 비용 확인
<code>/exit</code>	Claude Code 종료

2. 키보드 단축키

<code>Ctrl + C</code>	현재 작업 취소
<code>Esc</code>	입력 취소
<code>Esc</code> ×2	이전 메시지로 돌아가기
<code>Shift + Tab</code>	다음 제안/모드 전환
<code>↑ / ↓</code>	입력 히스토리 탐색

3. CLAUDE.md 작성 포인트

- ▶ **프로젝트 개요** — 무엇을 만들고 있는지 1-2줄로
- ▶ **기술 스택** — 언어, 프레임워크, DB, 주요 라이브러리
- ▶ **코딩 규칙** — 네이밍, 포매터, lint 설정
- ▶ **자주 쓰는 명령어** — build / test / dev / deploy
- ▶ **금지 사항** — 수정하면 안 되는 파일/디렉토리

4. 효과적인 프롬프트 작성법

- ▶ **구체적인 파일명 지정** — "@src/api/user.ts 수정해줘"
- ▶ **기대하는 출력 형식 명시** — JSON / Markdown / diff 등
- ▶ **Before/After 예시 제공** — 입출력 샘플로 정확도 향상
- ▶ **단계적으로 요청** — 설계 → 구현 → 테스트 → 리팩토링
- ▶ **검증 방법 전달** — "npm test가 통과해야 함"

5. 유용한 파이프 연동

```
git diff | claude -p "이 diff를 리뷰해줘"  
즉시 코드 리뷰
```

```
cat error.log | claude -p "원인이 뭐야?"  
로그에서 원인 분석
```

```
cat package.json | claude -p "의존성 정리해줘"  
package.json 감사
```

```
git log --oneline -20 | claude -p "릴리스 노트 작성"  
커밋에서 CHANGELOG 생성
```

알아두면 좋은 팁

- ▶ **@파일명** — 파일을 직접 컨텍스트에 추가
- ▶ **/compact 정기 실행** — 긴 세션에서 정확도 유지
- ▶ **계획과 실행 분리** — 먼저 Plan, 그 다음 Act