

# Claude Code Cheatsheet

Created by giz.ai

## QUICK START TIPS

효과적인 요청: 모호하게 X, 구체적으로  
복잡한 작업: 번호 매긴 단계별 분리  
코드 수정 전: 탐색/분석 먼저 요청  
단축키 확인: ? 입력

## PROJECT & FILES

/init CLAUDE.md 프로젝트 초기화  
/add-dir 추가 작업 디렉토리 포함  
/memory CLAUDE.md 메모리 편집  
/export [file] 대화 내보내기

## MODEL & SESSION

/model AI 모델 전환  
/compact [inst] 대화 압축 (선택적 포커스)  
/resume 이전 세션 이어하기  
/rewind 변경/대화 되돌리기

## CODE QUALITY

/review 코드 검토 요청  
/security-review 보안 감사 수행  
/pr-comments PR 피드백 확인

## DEV TOOLS

/sandbox 격리 bash 실행 활성화  
/hooks 도구 이벤트 핸들러 설정  
/plugin 확장 프로그램 관리  
/mcp MCP 서버 연결 관리  
/agents 커스텀 서브에이전트 생성

## WORKSPACE

/clear 대화 기록 삭제  
/context 토큰 사용량 시작화  
/cost 토큰 통계 (구독별)  
/usage 플랜 제한/속도 상태  
/status 설정 열기 (Status 탭)  
/config 설정 열기 (Config 탭)  
/todos 대기 중 항목 표시

## 입력 접두사

접두사	동작
#	CLAUDE.md에 메모 저장
/	슬래시 커맨드
!	bash 직접 실행
@	파일 경로 자동완성

## MULTILINE 입력

\+Enter 모든 터미널  
Option+Enter macOS 기본  
Shift+Enter /terminal-setup 후

## CUSTOM COMMANDS 만들기

### PROJECT

```
# .claude/commands/optimize.md
설명 분석 후 최적화 제안:
→ 사용: /optimize
```

### PERSONAL

```
# ~/claude/commands/security.md
보안 취약점 검토:
→ 사용: /security
```

## ARGUMENTS & REFERENCES

기능	문법	예시
전체 인자	\$ARGUMENTS	/fix 123 high → "123 high"
위치 인자	\$1 \$2	\$1="123", \$2="high"
파일 참조	@path	@src/utils.js → 내용 삽입
Bash 실행	!`cmd`	!`git status` → 결과 삽입

## NAMESPACE (하위폴더 구성)

```
.claude/commands/frontend/component.md
→ /component (project:frontend)

.claude/commands/backend/migrate.md
→ /migrate (project:backend)
```

## FRONTMATTER 옵션

```
--- description: 커맨드 설명 (자동호출 필수) allowed-tools:
Bash/git/*, Read(*) model: claude-3-5-haiku-20241022 argument-
hint: [param1] [param2] disable-model-invocation: false ---
```

## ALLOWED-TOOLS 패턴

패턴	설명
Bash/git:*)	git으로 시작하는 모든 명령
Bash/npm test:*)	npm test 명령만
Read(*)	모든 파일 읽기 허용
Edit(src/**)	src 폴더 내 파일 편집
Write(*.md)	md 파일만 쓰기
WebFetch(*)	모든 URL fetch 허용
mcp__github_*	Github MCP 모든 도구

## SLASHCOMMAND TOOL 권한

```
# /permissions 에서 설정 SlashCommand:/commit # 정확히 일치
SlashCommand:/review-pr:* # 접두사 매칭
```

글자 제한: 15,000자 (기본값)  
환경변수: SLASH\_COMMAND\_TOOL\_CHAR\_BUDGET

## MCP COMMANDS

```
# MCP 서버에서 동적 발견 /mcp__서버명__프롭프트명 [args] # 예시
/mcp__github_create-issue bug report /mcp__slack_send-message
#general /mcp__db_query SELECT * FROM users /mcp__docker_logs
container-name
```

## PLUGIN COMMANDS

```
# 마켓플레이스에서 배포된 플러그인 /plugin-name:command-name
/command-name # 모호하지 않으면 # 예시 /docker:build/docker:push
registry/image /aws:deploy production /k8s:rollout deployment/app
```

## SKILLS VS SLASH COMMANDS

SLASH COMMANDS	SKILLS
단순, 자주 사용하는 작업	복잡한 워크플로우
단일 .md 파일	다중 파일 + 스크립트
빠른 프롬프트 실행	구조화된 지식 기반
인자 전달 (\$1, \$2)	템플릿 시스템

## MODEL 옵션

모델	특징	용도
haiku	빠른, 저비용	간단한 작업
sonnet	균형	일반 작업
opus	최고 성능	복잡한 분석

## QUICK REFERENCE

커맨드 위치:  
• 프로젝트: .claude/commands/\*.md  
• 개인: ~/claude/commands/\*.md

자동 호출 차단:  
disable-model-invocation: true

## UI & EDITOR

/vim vim 모드 터글  
/statusline 상태 UI 설정  
/terminal-setup Shift+Enter 설치  
/ide IDE 통합 관리  
/output-style 출력 포맷 설정

## SYSTEM & SETUP

/login Anthropic 계정 전환  
/logout 로그아웃  
/doctor 설치 상태 점검  
/help 도움말 문서  
/bug 이슈 보고  
/permissions 권한 관리  
/release-notes 업데이트 보기  
/exit 종료

## CLI 실행 방법

명령	설명
claude	인터랙티브 모드 시작
claude "task"	단일 작업 실행
claude -p "query"	질문 후 종료
claude -c	최근 세션 계속
claude -r	이전 대화 재개
claude commit	Git 커밋 생성

## 키보드 단축키

키	동작
Ctrl+C	현재 작업 중단
Ctrl+D	세션 종료
Ctrl+L	화면 클리어
Ctrl+R	명령 기록 검색
Ctrl+B	백그라운드로 이동
Tab	자동완성 / thinking 터글
↑ ↓	이전 명령 탐색
Shift+Tab	권한 모드 전환

## 설치 방법

```
# macOS/Linux (Homebrew) brew install --cask claude-code #
macOS/Linux/WSL (Shell) curl -fsSL https://claude.ai/install.sh |
bash # Windows (PowerShell) irm https://claude.ai/install.ps1 | iex
```

## CHECKPOINTING (되돌리기)

접근	방법
Esc+Esc	rewind 메뉴 열기
/rewind	체크포인트 옵션

복원 옵션:  
• 대화만 - 코드 유지, 메시지 복원  
• 코드만 - 파일 복원, 대화 유지  
• 둘 다 - 전체 세션 복원

제한: Bash 명령(rm, mv 등)은 추적 안됨

## BACKGROUND 실행

Ctrl+B: 현재 Bash를 백그라운드로  
npm, webpack, 테스트 등 정시간 작업에 유용  
BashOutput 도구로 나중에 출력 확인