



바이브 메이커 · 무료 자료

클로드코드 비밀 커맨드 150

공식 내장 커맨드부터 커뮤니티가 검증한 프롬프트 코드까지 —
있는 줄도 몰랐던 150개를 번호 붙여 한 권에 정리했습니다.

/goal

/effort ultracode

/ghost

/critic

ultrathink

/eli5

/swot

슈퍼파워

비밀 커맨드는 세 종류입니다

150개는 성격이 다른 세 묶음으로 나뉩니다. 어느 묶음인지는 각 페이지 상단 칩으로 표시돼 있어요.



공식 내장 커맨드 001-045

클로드코드에 이미 들어있는 슬래시 커맨드·단축키. 채팅창에 / 만 쳐도 목록이 뜹니다.



설치형 스킬 — 슈퍼파워 046-060

해외 커뮤니티 최고 인기 스킬 모음 'superpowers'. 한 번 설치하면 상황에 맞게 자동 발동됩니다.



프롬프트 코드 061-150

공식 기능이 아니라, 커뮤니티가 검증한 **약속 패턴**입니다. 프롬프트 앞이나 뒤에 붙이면 클로드가 의도를 즉시 알아들어요. 여러 개 겹쳐 써도 됩니다.

프롬프트 코드는 이렇게 씁니다

● 터미널 · 클로드코드

```
/table /brief 국내 결제 서비스 3곳 수수료 비교해줘
```



외울 필요 없어요

이 자료는 "무엇이 있는지" 보여주는 **지도**입니다. 오늘은 눈에 걸리는 **5개만** 골라 써보세요 — 손에 익으면 다음 걸 없으면 됩니다.

알아서 굴러가게 하는 기본기 15개

✅ 공식 내장 — / 만 치면 됩니다

001	/goal	끝 조건을 만족할 때까지 알아서 반복 작업
002	/loop	정해진 간격 마다 반복 감시·실행
003	/compact	긴 대화를 요약해 기억 공간 되찾기
004	/clear	대화 초기화 — 새 주제는 새 출발
005	/resume	지난 세션 목록에서 골라 이어서 작업
006	/rewind	체크포인트로 되감기 — 강력한 실행취소
007	/export	대화 내용을 파일로 내보내기
008	/context	기억 공간이 얼마나 찼는지 확인
009	/usage	내 플랜 사용량 한도 확인
010	/cost	이번 세션의 토큰·비용 확인
011	/model	모델 전환 — 깊게(opus) vs 빠르게(haiku)
012	/memory	클로드코드가 기억하는 규칙 확인·관리
013	/init	프로젝트 규칙서 CLAUDE.md 자동 생성
014	/todos	지금 작업의 할 일 목록 확인
015	/doctor	설치·설정 자동 진단 — 안 될 때 1순위

아는 사람만 쓰는 파워 커맨드 15개

✅ 공식 내장 — / 만 치면 됩니다

016	/effort ultracode	일하는 강도 최대 — 에이전트 병렬 가동
017	/deep-research	웹 교차검증 + 출처 붙은 리포트
018	/plan	고치기 전에 계획부터 세우고 승인받기
019	/agents	역할별 보조 AI 만들어 분업
020	/mcp	노션·피그마 등 외부 도구 연결 관리
021	/hooks	특정 시점마다 자동 동작 걸어두기
022	/code-review	바뀐 코드 자동 점검 — 두 번째 눈
023	/security-review	보안 구멍만 집중 점검
024	/pr-comments	깃허브 PR 코멘트 불러와 바로 반영
025	/install-github-app	깃허브에서 @claude 태그로 일 시키기
026	/permissions	도구별 허용 규칙 관리
027	/add-dir	다른 폴더도 작업 범위에 추가
028	/output-style	답변 스타일 프리셋 바꾸기
029	/statusline	하단 상태줄 내 맘대로 커스텀
030	/terminal-setup	Shift+Enter 줄바꿈 등 터미널 세팅

타자 몇 번이 체감 속도를 바꿉니다

✅ 공식 내장 — 키보드에서 바로

031	Shift + Tab	권한 모드 순환 — 기본·자동승인·플랜
032	Esc	진행 중 작업 즉시 중단
033	Esc Esc	이전 메시지로 점프해 다시 쓰기
034	@파일	파일·폴더를 콧 집어 맥락 전달
035	!명령	터미널 명령을 대화창에서 바로 실행
036	#메모	한 줄 메모를 영구 기억으로 저장
037	Ctrl + R	답변 상세 출력(전체 로그) 토글
038	Ctrl + B	실행 중 작업을 백그라운드로 넘기기
039	Ctrl + V	스크린샷 붙여넣기 — 화면 보고 디버깅
040	↑ 화살표	이전 입력 히스토리 불러오기
041	Tab	파일 이름 자동완성
042	think	요청 끝에 붙이면 조금 더 깊이 생각
043	think hard	더 깊이 — 까다로운 판단에
044	ultrathink	최대 추론 깊이 — 어려운 설계·버그에
045	이미지 드래그	시안 이미지 던져주고 "이렇게 만들어줘"

해외 커뮤니티 1위 스킵팩

슈퍼파워 15개

🔧 설치형 — 한 번 설치하면 자동 발동

046	<code>/plugin</code>	여기서 superpowers 검색해 설치 (GitHub: obra/superpowers)
047	<code>using-superpowers</code>	스킬 시스템 사용법 안내 (입문)
048	<code>brainstorming</code>	코드 짜기 전 질문으로 기획 정제
049	<code>writing-plans</code>	상세 구현 계획서 작성
050	<code>executing-plans</code>	계획을 체크포인트 단위로 실행
051	<code>test-driven-development</code>	테스트 먼저 만들고 구현 — 실수 차단
052	<code>systematic-debugging</code>	4단계 근본 원인 디버깅 절차
053	<code>verification-before-completion</code>	"끝났어요" 전에 검증 강제
054	<code>dispatching-parallel-agents</code>	서브에이전트 동시 파견 워크플로우
055	<code>subagent-driven-development</code>	에이전트 주도 빠른 반복 개발
056	<code>requesting-code-review</code>	리뷰 요청 전 체크리스트
057	<code>receiving-code-review</code>	받은 피드백 반영 절차
058	<code>using-git-worktrees</code>	본 코드 안 건드는 안전 작업본 에서 병렬 작업
059	<code>finishing-a-development-branch</code>	작업 마무리·합치기 결정 워크플로우
060	<code>writing-skills</code>	나만의 스킬 을 만드는 스킬

답변 형태를 한 단어로 지정

프롬프트 코드 — 앞뒤에 붙여 쓰는 약속 패턴

061	/ghost	최종 답만 — 부연 설명 없이
062	/minimal	가능한 가장 짧게
063	/brief	3~5줄 이내로
064	/expand	자세하고 구체적으로
065	/tldr	핵심만 한 문단 요약
066	/headline	한 줄 핵심 문장으로
067	/bullets	핵심만 불릿 리스트로
068	/numbered	번호 있는 목록으로
069	/table	표로 정리
070	/json	JSON(데이터 형식)으로 출력
071	/markdown	마크다운 문서로 작성
072	/outline	목차 구조로 작성
073	/thread	SNS 스레드 형식으로
074	/tweet	280자 이내 초압축
075	/korean-only	전문용어 풀어서 쉬운 한글로

클로드의 생각하는 방식을 골라 쓰세요

프롬프트 코드 — 앞뒤에 붙여 쓰는 약속 패턴

076	/critic	결점만 찾아내기 — 냉정한 비평가
077	/eli5	5살도 알 수 있게 설명
078	/compare	선택지 나란히 비교
079	/proscons	장단점 나열
080	/firstprinciples	기본 원리부터 다시 쌓기
081	/contrarian	통념에 도전하는 관점으로
082	/steelman	반대 의견의 최강 버전 제시
083	/devils-advocate	일부러 반박해보게 하기
084	/socratic	답 대신 질문으로 유도
085	/systems	시스템 관점에서 연결 분석
086	/second-order	2차 효과(그 다음 일)까지 고려
087	/reframe	다른 관점으로 재해석
088	/invert	반대로 뒤집어 생각하기
089	/prioritize	중요도 순서로 정렬
090	/diagnose	문제의 원인 진단부터

코드 몰라도 쓰는 실전 코드 15개

💬 프롬프트 코드 — 앞뒤에 붙여 쓰는 약속 패턴

091	<code>/explain-code</code>	이 코드가 뭐 하는지 초보 눈높이 설명
092	<code>/error-fix</code>	에러 원인 + 해결책 만 콕 집어서
093	<code>/before-coding</code>	만들기 전에 물어볼 것부터 질문받기
094	<code>/mvp-scope</code>	오늘 배포 가능한 크기로 기능 축소
095	<code>/mobile-first</code>	휴대폰 화면 우선으로 만들기
096	<code>/a11y</code>	글자 크기·색 대비 등 접근성 점검
097	<code>/seo-check</code>	검색 노출 최적화 점검
098	<code>/speed-check</code>	느린 부분 찾아 개선안 제시
099	<code>/security-basic</code>	초보가 놓치는 보안 구멍 점검
100	<code>/naming</code>	파일·변수 이름 알기 쉽게 정리
101	<code>/test-cases</code>	배포 전 꼭 확인할 시나리오 뽑기
102	<code>/edge-cases</code>	예외 상황 미리 찾아 대비
103	<code>/rollback-plan</code>	잘못되면 되돌릴 계획부터
104	<code>/stack-recommend</code>	내 아이디어에 맞는 기술 조합 추천
105	<code>/clone-analysis</code>	벤치마킹할 사이트 구조 분석

머릿속 생각을 문서로 만드는 15개

💬 프롬프트 코드 — 앞뒤에 붙여 쓰는 약속 패턴

106	/stepbystep	단계별로 명확하게 설명
107	/checklist	실행 가능한 체크리스트로
108	/framework	구조화된 틀로 정리
109	/roadmap	타임라인 기반 로드맵
110	/blueprint	실행 계획 설계도
111	/playbook	반복 가능한 운영 매뉴얼
112	/template	재사용 가능한 템플릿 제공
113	/sop	표준 운영 절차(SOP) 작성
114	/report	보고서 형식으로
115	/proposal	제안서 형태로 구성
116	/recap	지금까지 논의 한눈 정리
117	/faq	자주 묻는 질문(FAQ) 형식
118	/glossary	용어 정리 모음으로
119	/agenda	회의 안건 형식으로
120	/memo	내부 공유용 메모 형식

만든 걸 팔리게 하는 코드 15개

💬 프롬프트 코드 — 앞뒤에 붙여 쓰는 약속 패턴

121	/hook	관심을 끄는 첫 문장 생성
122	/cta	행동 유도 문구(CTA) 작성
123	/caption	SNS 캡션으로 작성
124	/hashtags	관련 해시태그 목록
125	/reel-script	릴스·쇼츠용 대본
126	/ad-copy	광고 카피 작성
127	/email-subject	열어보게 만드는 이메일 제목줄
128	/variations	다른 버전 3~5개 생성
129	/viral	바이럴 요소 극대화 버전
130	/pain-point	고객 고충에서 시작하는 구성
131	/before-after	비포/애프터 구조로
132	/case-study	사례 연구 형식으로
133	/listicle	숫자 목록형(리스트icle) 콘텐츠
134	/repurpose	다른 플랫폼용으로 재가공
135	/series	시리즈 콘텐츠로 기획

사업 결정을 돕는 마지막 15개

프롬프트 코드 — 앞뒤에 붙여 쓰는 약속 패턴

136	/swot	SWOT (강점·약점·기회·위협) 분석
137	/okr	OKR (목표·핵심결과) 프레임워크로
138	/pitch	투자·세일즈 피치 형식
139	/persona	타겟 고객 페르소나 설계
140	/funnel	마케팅 퍼널 단계로 구성
141	/kpi	성과 지표(KPI) 설계
142	/monetize	수익화 방법 제안
143	/risk	리스크 파악 + 대응 방안
144	/launch	런칭 계획 수립
145	/benchmark	업계 기준 대비 벤치마킹
146	/decision	의사결정 도우미 역할로
147	/brainstorm	자유롭게 아이디어 쏟아내기
148	/100ideas	아이디어 100개 속사포 나열
149	/todo	대화 내용을 할 일 목록 으로
150	/debrief	작업 후 회고 정리

진짜 고수는 조합으로 씁니다

커맨드 하나하나보다, **겹쳐 쓸 때** 체감이 완전히 달라집니다. 검증된 조합 5가지.

/effort ultracode + /goal

팀 단위로, 끝날 때까지, **알아서**. 강도 최대로 올리고 끝 조건을 걸면 — 직원 한 명이 아니라 팀 하나가 일합니다.

/plan → 승인 → 자동승인 모드

계획 먼저 확인하고, 승인 후엔 Shift+Tab 자동승인으로 **안전과 속도를 둘 다**.

/deep-research + /report /table

교차검증 리서치를 **표 붙은 보고서**로 바로 받기 — 회의 자료가 한 번에 나옵니다.

/critic /proscons /contrarian

아이디어 하나를 **3중 검증**. 비평 + 장단점 + 반대 관점을 한 방에 — 셀프 리뷰 위원회.

brainstorming → writing-plans → executing-plans

슈퍼파워 풀코스 — **기획 정제 → 계획서 → 실행**이 하나의 흐름으로 이어집니다.

⚡ 파워 팁

프롬프트 코드는 **2~3개까지 겹쳐도** 잘 알아듣습니다. 예: **/table /brief /korean-only** 경쟁사 비교해줘

여기까지 읽으셨다면

150개를 외우는 게 아니라, 5개를 쓰는 겁니다

오늘 눈에 걸린 것 딱 5개만 골라 이번 주에 써보세요.

다음 주에 5개를 더 얹으면 — 한 달 뒤엔 완전히 다른 사람이 됩니다.

`/goal` 오늘의 첫 자동화 — 끝날 때까지 알아서

`/effort ultracode` 큰 일은 팀 단위로 병렬 처리

`/ghost /table` 답변 형태부터 내 맘대로