

클로드 코드, 왜 벌써 한도가 찼을까?

별로 쓰지도 않았는데 사용량이 확 줄었다면 — 원인부터 찾는 한 장

1장 · 진단

2026년 9월 21일 기준

Claude Code v2.1.2xx

미국주부 (Momprenuer Diary)

왜 이게 중요한가

한도는 **늘려서** 푸는 게 아니라 **어디서 새는지 찾으면** 대부분 풀립니다. 클로드는 요청할 때마다 **그 창의 대화 전체**를 다시 보냅니다. 그래서 **질문을 길게 썼느냐보다, 대화창을 언제 비웠느냐**가 사용량을 정합니다.

언제 꺼내 보나

작업 중에 "한도 다 됐다"가 떴다	2번 부터
이상하게 빨리 닳는 것 같다	1번 부터
평소에 덜 쓰고 싶다	2장 6번
뭘 눌러야 할지 모르겠다	2장 4번

1 내가 안 썼는데 사용량이 느는 6가지

가장 흔한 순서

한 시간 쉬었다 왔다 제일 많이 겪는 경우	클로드는 이전 대화 전체 를 매번 다시 보냅니다. 평소엔 캐시가 있어서 싸게 처리되는데, 한 시간 넘게 쉬면 캐시가 식어서 전부 다시 처리 합니다. 돌아와서 한 줄만 물어봐도 그날 대화 전체값이 나갑니다. 추가 크레딧을 쓰는 중이면 5분 으로 줄어듭니다.
창을 안 닫고 계속 썼다 긴 세션	대화가 길수록 매 요청이 무거워집니다. 아침에 열어둔 창에서 오후에 한 줄 물으면 아침부터의 전부 가 같이 갑니다. 주제가 바뀌면 새 창 으로.
/compact를 자주 눌렀다	요약하려면 대화를 전부 읽어야 합니다. 즉 압축 자체가 큰 요청 입니다. 이어서 할 일이 없다면 /compact 말고 /clear(공짜) .
예약 작업이 돌고 있다	예약해둔 작업은 내가 자리에 없어도 정해진 시간마다 실행되고, 그때마다 전체 대화를 실어 보냅니다. /usage 에서 어떤 예약 작업이 얼마나 먹었는지 볼 수 있습니다.
오피스를 기본으로 두고 있다	공식 문서가 "예상보다 많이 나오는 가장 흔한 원인 둘" 중 하나로 꼽습니다(다른 하나는 위의 긴 세션). 소네트로 대부분의 작업이 됩니다 . 복잡한 설계나 여러 단계 판단이 필요할 때만 오피스로.
MCP 서버를 많이 붙였다	기본값에선 도구 이름과 안내문만 먼저 들어가지만, 서버가 많으면 이것도 쌓입니다. /mcp 로 안 쓰는 건 고세요. 명령어 도구 (gh-aws 같은)가 목록을 아예 안 만들어서 더 가볍습니다.
먼저 이것부터: <code>/usage</code> → d (24시간) / w (7일). 최근 사용량의 10% 이상을 차지한 원인 을 클로드가 짚어주고, 스킴·MCP·예약작업별 비중도 나옵니다.	

2 한도에 걸렸을 때 — 이 네 줄을 구분하세요

화면에는 영어로 뜹니다

화면에 뜨는 문장	무슨 뜻인가	모델을 바꾸면?
You've hit your session limit	지금 사용 구간을 다 씀	✗ 소용없습니다 . 모든 모델이 이 한도를 같이 씁니다. 메시지에 언제 풀리는지가 같이 나옵니다 — 그 시각을 보세요.
You've hit your weekly limit	주간 사용량을 다 씀	✗ 소용없습니다 . 위와 같습니다. 기다리거나 추가 크레딧을 커야 합니다.
You've hit your Opus limit	오피스 한 모델만 다 씀	○ 계속할 수 있습니다 . <code>/model</code> 로 다른 계열(소네트 등)로 바꾸면 그대로 작업을 이어갑니다.
You've hit your Sonnet limit	소네트 한 모델만 다 씀	○ 계속할 수 있습니다 . 위와 같습니다. 다른 계열로 바꾸세요.

가장 많이 하는 착각: 한도에 걸리면 무조건 모델부터 바꾸는데, 위의 두 줄(session-weekly)일 땐 **아무 소용이 없습니다**. 아래 두 줄일 때만 통합니다. 먼저 **어느 문장인지 읽고** 움직이세요.

기다리는 것도 방법입니다: `/rate-limit-options` 에서 **한도가 풀리면 하던 일을 자동으로 이어서 하도록** 맞춰둘 수 있습니다. 당장 더 써야 하면 `/usage-credits`.

유료 구독자 주의 — /usage의 달러 금액은 무시하세요. 그건 API로 쓰는 사람용 표시입니다. Max-Pro는 같은 화면의 「**사용량 막대**」를 보세요. 이 컴퓨터 기록으로 계산한 대략값이라 **다른 기기·claude.ai에서 쓴 건 안 들어갑니다**.

그래서 뭘 누르면 되나

상황에 맞는 명령어 하나만 고르면 됩니다

4 지금 상황이 넷 중 어느 것인가

이것만 외우면 됩니다

전혀 다른 일을
새로 시작한다

/clear

공짜 · 대화를 비움

같은 일인데
대화가 너무 길어졌다

/compact

요약 · 비용이 듭니다

방금 한 작업을
되돌리고 싶다

/rewind

코드·대화를 같이 되돌림

잠깐 딴 걸
물어보고 싶다

/btw

대화 기록에 안 남음

/clear는 공짜, **/compact**는 비쌉니다. 요약하려면 대화를 다 읽어야 하거든요. 이어서 할 일이 없으면 무조건 **/clear**. 비우기 전에 **/rename**으로 이름을 붙여두면 **/resume**으로 다시 꺼낼 수 있습니다.

5 명령어 한 장

자주 쓰는 순서

/usage	지금 얼마나 썼나. d =24시간 / w =7일. 무엇이 먹었는지(스킬·MCP·예약작업)까지 나옵니다.
/context	무엇이 자리를 차지하나. 색깔 격자 + 줄일 방법. /context all 은 항목별로 펼칩니다.
/clear	대화를 비우고 새로 시작. 비용이 들지 않습니다 .
/compact	대화를 요약해 자리를 만듭니다. /compact 뒤에 「코드 예시 위주로」처럼 남길 것을 지정 가능.
/rewind	코드와 대화를 이전 시점으로 되돌립니다. Esc 두 번으로도 됩니다.
/btw	지금 세션에 대해 옆으로 잠깐 질문 . 대화 기록에 안 쌓입니다. 질문 없이 /btw 만 치면 지난 질문을 다시 봅니다.
/model	쉬운 일엔 가볍게, 어려운 일엔 강하게. 복잡한 설계가 아니면 소네트로 충분합니다.
/effort	얼마나 깊게 생각할지. 생각하는 양도 사용량입니다. 단순 작업은 낮춰 두세요.
/mcp	붙여둔 연결 목록. 안 쓰는 건 끄세요 .
/rename /resume	대화에 이름 붙이기 / 다시 꺼내기. /clear 전에 /rename 하는 습관.
/insights	내가 어떻게 일하는지 분석 리포트 . 어디서 막히는지, 뭘 고치면 좋은지.
/usage-credits	한도를 넘겨서 더 쓰기. 월 지출 한도 도 같이 정합니다.

6 습관으로 줄이기

명령어보다 효과가 큼니다

짧게 쓰지 말고
정확하게 쓰기

두루뭉술하면 다 뒤집힙니다. "이 코드 개선해줘"는 전체를 훑게 만들고, "auth.ts의 로그인 함수에 입력값 검사 넣어줘"는 필요한 파일만 읽습니다. 어디를 볼지 짚어주는 게 핵심입니다.

CLAUDE.md는
200줄 이하로

시작할 때마다 통째로 들어갑니다. 특정 작업에만 필요한 건 **스킬**로 옮기세요 — 부를 때만 들어옵니다.

복잡한 일은
계획 모드부터

Shift+Tab. 방향을 먼저 잡고 승인받아 실행 — 엉뚱한 길로 갔다 다시 하는 낭비를 막습니다.

시끄러운 작업은
서브에이전트에게

테스트·로그 읽기처럼 결과가 긴 일은 따로 시키면 **요약만 돌아옵니다**.

7 다른 도구를 쓰신다면

이름이 다릅니다

도구	대화 요약	사용량 확인	대화 비우기
클로드 코드	/compact	/usage	/clear
Codex CLI (OpenAI)	/compact	/status	—
Gemini CLI (Google)	/compress	/stats	/clear

※ Gemini CLI는 2026년 6월 18일 무료·Google One 사용자 대상으로 **Antigravity CLI**로 대체되었습니다(공식 문서). 도구 이름은 계속 바뀝니다.