

바이브 메이커 · 무료 자료

클로드가 멍청해지기 전에, 컨텍스트 습관 4개

클로드코드가 오래 쓰면 흐려지는 이유와, 그 전에 손쓰는 습관 네 가지. /compact 할 때 그대로 붙여 넣는 문장까지 담았습니다.

/context 로 보기 · /compact 에 남길 것 · 새 일은 /clear · 규칙은 CLAUDE.md

목차

한눈에	습관 4개, 한 장으로	03
왜	클로드는 왜 1시간 지나면 멍청해질까	04
습관 1 /context	얼마나 찼는지 본다	05
습관 2 /compact	요약은 내가 시키고, 남길 것을 같이 적는다	06
습관 3 /clear	새 일은 새 대화에서	07
습관 4 CLAUDE.md	매번 지킬 규칙은 대화 말고 파일에	08
실측	그냥 두면, 자동 요약은 언제 왔나	09
모음	붙여 넣는 문장 한 장	10
막히면	증상 → 할 일	11

한눈에

습관 4개, 한 장으로

클로드코드는 대화가 길어질수록 읽을 것이 쌓입니다. 꽉 차기 전에 내가 먼저 보고, 줄이고, 비우는 습관 네 개입니다.

습관	칠 것 · 할 것	언제	얻는 것
1 얼마나 찼나 보기	`/context`	긴 작업 중간중간, 답이 흐려질 때	무엇이 자리를 먹는지, 얼마나 남았는지
2 남길 것 적고 요약	`/compact 남길 것`	일 한 덩어리가 끝났을 때, 긴 새 작업 전	요약이 내가 고른 것을 지킨다
3 새 일은 새 대화	`/clear`	앞 대화와 관계없는 일을 시작할 때	빈 자리에서 다시 시작 (프로젝트 메모리는 유지)
4 규칙은 파일에	`CLAUDE.md`	매번 지켜야 할 규칙이 생겼을 때	요약 뒤에도 규칙이 다시 읽힌다

준비물

- 클로드코드 (명령어는 대화창 맨 앞에 `/` 로 칩니다)
- 작업 폴더 하나, 그리고 그 폴더의 `CLAUDE.md` (없으면 습관 4에서 만듭니다)

> 숫자는 2026-11-09 클로드코드 2.1.317 에서 우리 작업 폴더로 직접 잰 것입니다. 시험하지 않은 기능은 「문서 기준」이라고 적었습니다. 출처는 공식 문서 code.claude.com/docs 입니다.

왜

클로드는 왜 1시간 지나면 멍청해질까

클로드코드가 한 번에 볼 수 있는 자리를 「컨텍스트」라고 부릅니다. 이 자리가 차오르는 것이 시작입니다.

컨텍스트에 들어 있는 것 (공식 문서)

- 지금까지의 대화 전부
- 열어 본 파일 내용, 실행한 명령어의 출력
- CLAUDE.md 같은 규칙 파일, 자동 메모리, 불러온 스킬, 시스템 지시

1 **메시지 하나 보낼 때마다 전부 다시 보낸다**

클로드코드는 요청할 때마다 대화 전체를 함께 보냅니다. 대화가 길수록 매번 읽을 양이 늘어납니다.

2 **차면 저절로 요약한다**

자리가 차면 대화를 요약해 줄입니다. 요약은 요청과 핵심 코드는 남기지만, 앞쪽에서 한 자세한 지시는 빠질 수 있습니다.

3 **그래서 「아까 말했잖아」가 생긴다**

처음에 정한 규칙을 잊은 것처럼 보이는 순간이 이때입니다. 요약에 무엇이 남을지를 내가 정하지 않았기 때문입니다.

18%
새 세션, 한 글자도 안 쳤는데 이미 찬 자리 (53k / 300k)

35.1k
그중 규칙 파일(CLAUDE.md 등)이 차지한 몫

29분
우리 세션이 자동 요약에 걸리기까지 걸린 시간

> 제목의 「1시간」은 느낌입니다. 우리 기록으로는 파일을 많이 읽는 작업에서 18~29분 만에 자동 요약이 왔습니다(실측 쪽에 표로 정리).

습관 1 `/context`

얼마나 찼는지 본다

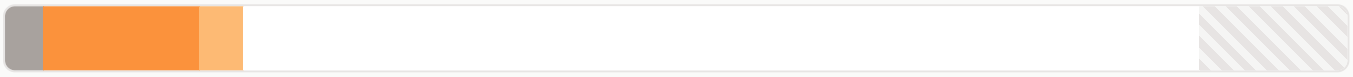
`/context` 를 치면 지금 자리를 무엇이 먹고 있는지 색깔 칸으로 보여 줍니다. 칸별 숫자를 다 보려면 `/context all` 입니다.

복사해서 그대로 쓰세요

```
/context
```

같은 폴더, 두 번 재 봤습니다

새 세션 (아무것도 안 함) 53k / 300k · 18%



구현을 한참 한 세션 166.1k / 300k · 55%



시스템 · 도구 · 규칙 파일 · 스킬 · 대화 · 남은 자리 · 자동 요약 자리

/context 칸 이름	뜻	새 세션	한참 뒤
Memory files	CLAUDE.md 같은 규칙 파일	35.1k	35.1k
Skills	불러온 스킬 설명	9.9k	9.9k
Messages	**우리 대화 · 파일 · 명령 출력**	**10**	**113.1k**
Free space	남은 자리	214k	100.9k
Autocompact buffer	자동 요약용으로 비워 둔 자리	33k	33k

> 볼 곳은 두 군데입니다. **Messages** 가 커지는 속도, 그리고 **Free space** 가 얼마 남았는지. 규칙 파일 칸이 처음부터 크면 습관 4에서 줄일 차례입니다.

습관 2 `/compact`

요약은 내가 시키고, 남길 것을 같이 적는다

`/compact` 는 지금까지의 대화를 통째로 요약하고 그 요약에서 다시 시작합니다. 뒤에 남길 것을 적으면 요약이 그것을 지킵니다.

그냥 /compact

무엇을 남길지 클로드가 고릅니다. 앞쪽의 자세한 지시는 빠질 수 있습니다.

/compact + 남길 것

내가 고른 것을 남깁니다. 공식 문서도 긴 새 작업 전에 이렇게 하라고 권합니다.

① 일 한 덩어리가 끝났을 때

복사해서 그대로 쓰세요

/compact 지금까지 확정한 결정과 그 이유, 고친 파일 경로, 남은 할 일만 남겨. 시도했다가 안 된 방법은 「안 됨」으로 한 줄씩.

② 긴 새 작업을 시작하기 전

복사해서 그대로 쓰세요

/compact 다음 작업은 (여기에 한 줄). 이 작업에 필요한 결정 · 파일 · 규칙만 남기고 나머지는 한 줄로 줄여.

③ 버그를 잡다가 길어졌을 때

복사해서 그대로 쓰세요

/compact 버그 증상, 재현 방법, 원인 후보 중 확인된 것과 아닌 것, 지금까지 고친 파일만 남겨.

> 요약도 공짜는 아닙니다. 대화 전체를 읽어서 요약하므로 긴 세션에서는 그 자체로 큰 요청입니다(문서 기준). 그래서 자리가 다 찬 뒤보다, 일이 한 번 끊기는 지점에서 하는 편이 낫습니다.

습관 3 `/clear`

새 일은 새 대화에서

앞 대화와 관계없는 일이라면 요약할 필요도 없습니다. `/clear` 는 빈 자리에서 새 대화를 시작하고, 프로젝트 메모리(CLAUDE.md)는 그대로 둡니다.

1 **넘길 것만 다섯 줄로 받아 둔다**

이어서 할 일이 있다면, 비우기 전에 클로드에게 정리를 시킵니다.

복사해서 그대로 쓰세요

새 대화에서 이어 갈 수 있게 정리해 줘. 확정한 결정, 고친 파일 경로, 남은 할 일, 지켜야 할 규칙을 다섯 줄 안으로.

2 **이름을 붙여 비운다**

`/clear` 뒤에 이름을 쓰면 방금 대화가 그 이름으로 남습니다. 나중에 `/resume` 목록에서 찾아 다시 열 수 있습니다(문서 기준).

복사해서 그대로 쓰세요

/clear 로그인-버그-1차

3 **새 대화 첫 메시지에 다섯 줄을 붙인다**

1번에서 받은 정리를 붙여 넣고 다음 할 일을 적습니다.

이럴 때는 바로 /clear

- 전혀 다른 기능, 다른 파일을 시작할 때
- 같은 문제를 두 번 고쳐 달라고 했는데도 안 될 때. 공식 문서는 이때 비우고, 배운 것을 담아 질문을 다시 쓰라고 권합니다.

습관 4 `CLAUDE.md`

매번 지킬 규칙은 대화 말고 파일에

대화에서 한 말은 요약에서 빠질 수 있지만, 프로젝트 맨 위 폴더의 CLAUDE.md 는 요약 뒤에 파일에서 다시 읽어 들입니다(공식 문서). 오래 지킬 규칙은 여기에 둡니다.

① 자꾸 다시 말하게 되는 규칙을 옮긴다

복사해서 그대로 쓰세요

이 대화에서 내가 두 번 이상 다시 말한 규칙을 찾아서, CLAUDE.md 에 한 줄씩 넣을 문장으로 정리해 줘. 넣기 전에 나에게 먼저 보여 줘.

② 자동 요약 때 남길 것을 미리 적어 둔다

CLAUDE.md 에 「Compact instructions」 칸을 만들면, 자동 요약이 올 때도 무엇을 남길지 정해 둘 수 있습니다. 공식 문서의 방법입니다. 아래를 CLAUDE.md 맨 아래에 붙입니다.

복사해서 그대로 쓰세요

```
# Compact instructions

요약할 때 다음은 꼭 남긴다.
- 확정한 결정과 그 이유
- 고친 파일 경로
- 남은 할 일
- 시도했다가 안 된 방법 (한 줄씩)
```

> ②의 한국어 문장은 문서의 영어 예시(「compact 할 때 테스트 출력과 코드 변경에 집중하라」)를 우리 습관에 맞게 바꾼 것입니다. 이 문장으로 요약 결과를 비교 시험하지는 않았습니다(문서 기준).

③ 규칙 파일은 짧게

규칙 파일도 자리를 먹습니다. 우리 폴더는 새 세션에서 규칙 파일만 35.1k 였습니다. 매번 필요한 규칙만 남기고, 가끔 쓰는 절차는 다른 파일로 빼 두는 편이 낫습니다.

실측

그냥 두면, 자동 요약은 언제 왔나

우리 세션 기록에서 자동 요약이 일어난 지점 세 곳을 꺼냈습니다. 세 번 모두 1시간이 안 돼서 왔습니다.

세션 (한국 시각)	시작 뒤	요약 전 (토큰)	요약 후 (토큰)
2026-11-07 오전	24분	272,659	21,748
2026-11-07 오후	18분	270,190	32,705
2026-11-09 새벽	**29분**	**267,319**	**23,679**

세 번 모두 약 27만 토큰에서 요약됐고, 요약 뒤에는 2~3만 토큰만 남았습니다. 마지막 세션은 「구현 진행해 줘」라고 한 지 17분 뒤, 명령을 실행하는 한가운데서 요약됐습니다. 내가 고르지 않은 순간입니다.

왜 27만 근처였나

이 맥은 자동 요약 창을 300k 로 정해 둔 설정입니다. 그중 33k 를 자동 요약 자리로 비워 두니, 300k – 33k = 약 267k 에서 요약이 시작됩니다. 내 값은 `/context` 첫 줄의 분모와 Autocompact buffer 칸에서 봅니다.

복사해서 그대로 쓰세요

```
/autocompact
```

인수 없이 치면 지금 창 크기를 보여 줍니다. `/autocompact 500k` 처럼 바꿀 수 있고, `/autocompact auto` 는 모델에 맞춘 기본값으로 되돌립니다. 클로드코드 2.1.221 이상(문서 기준).

> 창을 키우면 요약은 늦게 오지만, 그만큼 매번 보내는 대화도 길어집니다. 창 크기보다 습관 2 · 3 으로 내가 먼저 끊는 쪽이 핵심입니다.

모음

붙여 넣는 문장 한 장

이 쪽만 열어 두고 필요할 때 복사하세요.

보기 지금 얼마나 찼나

복사해서 그대로 쓰세요

/context

요약 일 한 덩어리가 끝났을 때

복사해서 그대로 쓰세요

/compact 지금까지 확정한 결정과 그 이유, 고친 파일 경로, 남은 할 일만 남겨. 시도했다가 안 된 방법은 「안 됨」으로 한 줄씩.

요약 긴 새 작업 전

복사해서 그대로 쓰세요

/compact 다음 작업은 (여기에 한 줄). 이 작업에 필요한 결정 · 파일 · 규칙만 남기고 나머지는 한 줄로 줄여.

비우기 전 넘길 것 받기

복사해서 그대로 쓰세요

새 대화에서 이어 갈 수 있게 정리해 줘. 확정한 결정, 고친 파일 경로, 남은 할 일, 지켜야 할 규칙을 다섯 줄 안으로.

비우기 이름 붙여서

복사해서 그대로 쓰세요

/clear 작업-이름

규칙 옮기기 CLAUDE.md 로

복사해서 그대로 쓰세요

이 대화에서 내가 두 번 이상 다시 말한 규칙을 찾아서, CLAUDE.md 에 한 줄씩 넣을 문장으로 정리해 줘. 넣기 전에 나에게 먼저 보여 줘.

막히면

증상 → 할 일

증상	할 일
처음에 정한 규칙을 자꾸 잊는다	습관 4. 그 규칙을 CLAUDE.md 로 옮긴다. 요약 뒤에도 파일에서 다시 읽힌다.
같은 실수를 두 번 고쳐도 또 한다	습관 3. 넘길 것 다섯 줄을 받고 `/clear`, 배운 것을 넣어 질문을 다시 쓴다.
점심 먹고 왔더니 대화가 요약돼 있다	공식 문서: 1시간 캐시를 쓸 때 20만 토큰을 넘긴 세션이 쉬고 있으면, 캐시가 끝나기 직전에 요약될 수 있다. 자리를 비우기 전에 습관 2로 직접 요약해 둔다.
중간부터 꼬였는데 처음부터 다시 하긴 아깝다	`/rewind` (Esc 두 번)로 꼬이기 전 지점을 고른다. 고른 지점부터 요약하는 선택지도 있다 (문서 기준).
옆길 질문 하나 때문에 대화가 길어진다	`/btw 질문`. 대화 기록에 더하지 않고 답만 받는다(문서 기준).
/compact 를 했더니 오래 걸린다	정상이다. 대화 전체를 읽어 요약하므로 긴 세션일수록 크다. 다 찬 뒤보다 일이 끊기는 지점에서 한다.
/context 숫자가 이 자료와 다르다	정상이다. 분모(창 크기)는 모델과 설정마다 다르다. 이 자료의 300k 는 우리 맥 설정이다.

> 더 묻고 싶으면 네이버 카페 「바이브메이커」 자료실 글 댓글로.