

바이브뉴스 전용 자료 · 무료

AI가 내 말을 흘려듣지 않게

바이브뉴스 실전 팁 vol.2

AI가 지시를 놓치는 이유 + 컨텍스트 엔지니어링 3원칙 — 복불 프롬프트
포함

목차

PART 1	AI가 내 말을 흘려듣는 3가지 이유	1
1.1	AI는 순서대로 읽지 않는다	1
1.2	한가운데 넣은 지시는 묻힌다 (Lost in the middle)	1
1.3	AI는 칭찬하고 싶어 한다 (아침 성향)	1
PART 2	컨텍스트 엔지니어링 3원칙	3
2.1	원칙 ① 관련된 것만 넣기	3
2.2	원칙 ② 핵심은 맨 앞이나 맨 끝에	3
2.3	원칙 ③ 길이는 비용이자 노이즈	4
PART 3	AI가 딴짓 안 하게 만드는 프롬프트 법칙	5
3.1	"어떻게"가 아니라 "무엇이 성공인지"를 줘라	5
3.2	금지 나열 대신 원하는 걸 긍정형으로	5
3.3	"괜찮지?" 대신 "약점부터 찾아줘"	5
3.4	그대로 복붙하는 Do / Don't 체크리스트	6

이 자료의 모든 팁은 원문 딥리딩·실제 사례 검증을 거쳐 비전공자 눈높이로 정리했습니다. 복붙 박스(회색 코드 상자)는 그대로 복사해 프롬프트에 붙여넣으면 됩니다.

PART 1 전체 3장

AI가 내 말을 흘려듣는 3가지 이유

"분명 시켰는데 왜 자꾸 딴짓하지?"

AI에게 규칙을 줬는데 어떤 날은 잘 지키고 어떤 날은 통째로 무시한 경험, 다들 있으실 거예요. 배달의민족 개발팀도 같은 문제를 겪었습니다. 똑같은 지시를 줬는데 어떤 날은 완벽하고 어떤 날은 규칙을 전혀 안 지키는 코드가 나왔대요.

원인을 파고들면 AI가 게을러서가 아니라, **AI가 글을 읽는 방식** 때문입니다. 세 가지만 알면 됩니다.

1.1 AI는 순서대로 읽지 않는다

우리는 글을 위에서 아래로 읽지만, AI는 다릅니다. 모든 단어를 **동시에** 놓고 **서로 얼마나 관련 있는지를** 계산해요.

문제는 여기서 생깁니다. 중요한 지시 사이에 관련 없는 내용이 잔뜩 끼면, 정작 중요한 지시의 "관련도"가 희석돼서 묻혀버려요. 불필요한 문장은 그냥 무시되는 게 아니라 — **AI를 엉뚱한 답으로 끌고 가는 방해물**이 됩니다.

1.2 한가운데 넣은 지시는 묻힌다 (Lost in the middle)

AI가 한 번에 다룰 수 있는 작업 공간(컨텍스트)은 무한하지 않습니다. 그리고 재밌는 특성이 하나 있어요.

같은 지시라도 글의 정확히 한가운데 있으면 활용도가 뚝 떨어지고, 맨 앞이나 맨 끝에 있으면 훨씬 잘 지켜집니다.

시스템 설정이나 우리가 마지막에 친 명령이 잘 먹히는 이유가 이거예요 — 위치가 양 끝이라서. 그러니 진짜 중요한 지시를 대화 중간에 슬쩍 끼워 넣으면 흘려듣기 딱 좋습니다.

1.3 AI는 칭찬하고 싶어 한다 (아침 성향)

AI는 사용자에게 동의하고 칭찬하려는 경향이 있습니다. 학습 과정에서 생긴 버릇이에요.

그래서 "이 방식 괜찮지?"라고 물으면 웬만하면 "좋습니다"가 나옵니다. 진짜 문제점이 있어도요. 검증받으려고 물었는데 칭찬만 돌아오는 거죠.



이 세 가지가 겹치면 "어제는 잘하더니 오늘은 왜 이래?"가 됩니다. AI가 번덕스러운 게 아니라, 우리가 준 글의 **구성**이 매번 달랐던 거예요. 다음 장부터 이걸 고치는 법을 봅니다.

◆ 이 PART의 핵심 정리

- AI는 순서가 아니라 관련도로 읽는다 — 잡음이 끼면 중요 지시가 희석된다
- 한가운데 지시는 묻힌다 — 핵심은 맨 앞이나 맨 끝에
- AI는 칭찬하려 한다 — "괜찮지?"로 물으면 검증이 안 된다

PART 2 전체 3장

컨텍스트 엔지니어링 3원칙

앞 장에서 본 AI의 습성을 역이용하는 게 **컨텍스트 엔지니어링**입니다. 어려운 말 같지만 뜻은 단순해요.



프롬프트 한 줄을 영리하게 쓰는 걸 넘어서, **AI가 볼 글 전체를 무엇으로·얼마나·어떤 순서로 채울지 설계하는 일.**

원칙은 딱 세 개입니다.

2.1 원칙 ① 관련된 것만 넣기

"혹시 모르니까 다 붙여두자"가 가장 흔한 실수예요. 정보가 많을수록 AI의 집중이 분산됩니다. 앞 장에서 봤듯이 불필요한 텍스트는 노이즈를 넘어 **오답을 유도**해요.

지금 이 작업에 진짜 필요한 것만 골라 넣으세요. 지난 대화, 안 쓸 자료, "참고용"이라며 붙인 긴 문서 — 관련 없으면 뺍니다.

2.2 원칙 ② 핵심은 맨 앞이나 맨 끝에

절대 놓치면 안 되는 지시는 프롬프트의 **맨 앞이나 맨 끝**에 두세요. 한가운데는 묻힙니다.

그리고 관련 있는 정보끼리는 관계가 보이게 묶으세요. 제목(##)이나 구분선으로 나눠주면, AI가 "여기가 규칙 구역이구나" 하고 필요한 대목을 더 잘 찾아냅니다.



복불해서 쓰세요. 지시가 여러 개일 때 이 뼈대로 배치하면 됩니다.

TEXT

가장 중요한 목표 (맨 앞)

{이번 작업에서 반드시 지킬 핵심 1~2개}

참고 정보

{필요한 자료만 — 관련 없는 건 빼기}

다시 강조 (맨 끝)

{맨 앞의 핵심을 한 번 더 — 끝 위치라 잘 지켜짐}

2.3 원칙 ③ 길이는 비용이자 노이즈

프롬프트가 길수록 정확해진다는 건 착각입니다. 길이는 두 가지 대가를 치러요 — 작업 공간을 잡아먹고, AI가 계산할 양을 폭발적으로 늘립니다(글이 2배 길어지면 계산은 4배).

"충분한 만큼만." 짧게 쓰라는 게 아니라, 목적에 필요한 만큼만 담으라는 거예요.

⚡ 한 단계 더 — 배달의민족 팀은 AI에게 일을 시킬 때, 지금까지의 복잡한 논의 기록을 다 넘기지 않고 **정제된 목표만** 따로 전달했더니 결과가 훨씬 깨끗해졌다고 합니다. "깨끗한 새 책상"에서 일하게 해준 거예요. 우리도 새 대화를 열어 목표만 딱 주면 같은 효과를 봅니다.

◆ 이 PART의 핵심 정리

- 관련된 것만 — 불필요한 정보는 노이즈가 아니라 오답 유도
- 핵심은 맨 앞·맨 끝, 정보는 제목으로 묶기
- 길이는 비용이자 노이즈 — 충분한 만큼만

PART 3 전체 3장

AI가 딴짓 안 하게 만드는 프롬프트 법칙

원리를 알았으니, 이제 실제로 프롬프트를 어떻게 쓰는지입니다. 최신 AI에 맞는 세 가지 법칙과, 그대로 복붙할 체크리스트를 드릴게요.

3.1 "어떻게"가 아니라 "무엇이 성공인지"를 줘라

예전엔 절차를 하나하나 떠먹였습니다. "먼저 이거 하고, 다음 저거 하고..." 그런데 요즘 똑똑한 AI에겐 절차를 과하게 지정하면 오히려 발목을 잡아요. AI가 스스로 더 나은 길을 찾는 걸 막거든요.

대신 ****무엇이 성공인지(성공 기준)****를 주세요.



복붙 비교. 왼쪽처럼 말고 오른쪽처럼.

TEXT

약함: "코드 정리해줘"

강함: "변수 이름은 전부 소문자로, 주석은 한국어로 통일해줘"

약함: "버그 고쳐줘"

강함: "지금 실패하는 이 테스트가 통과되게 고쳐줘"

약함: "먼저 검증하고, 정렬하고, 형식 바꿔"

강함: "결과는 날짜순 정렬 + 표 형식이면 돼 (방법은 알아서)"

3.2 금지 나열 대신 원하는 걸 긍정형으로

"이것도 하지 마, 저것도 하지 마"를 잔뜩 나열하면 AI가 헷갈립니다. 원하는 행동을 **긍정형**으로 적으세요.

그리고 규칙은 특정 사례로 못 박지 말고 **원리**로 적으세요. "이미지 크기는 반드시 정해진 값으로"처럼 좁게 못 박으면 AI가 엉뚱하게 조합합니다. "간격·여백은 정해진 값으로, 이미지 크기는 실제 픽셀로"처럼 경계를 원리로 주면 오해가 줄어요.

3.3 "괜찮지?" 대신 "약점부터 찾아줘"

AI는 칭찬하고 싶어 한다고 했죠. 그러니 검증을 받고 싶으면 질문을 바꿔야 합니다.



복붙. 리뷰나 점검이 필요할 때 이렇게.

TEXT

약함: "이 방식 괜찮지?"

강함: "이 방식의 한계와 반례부터 먼저 찾아줘."

칭찬은 빼고 문제점 위주로."

3.4 그대로 복붙하는 Do / Don't 체크리스트

프롬프트 쓰기 전에 이 표를 한 번 훑으세요. 왼쪽을 하고 있으면 오른쪽으로 바꾸면 됩니다.

TEXT

❌ 관련 있어 보이면 일단 다 넣는다

✅ 필요한 것만 골라 넣는다

❌ 지시를 중요도 상관없이 늘어놓는다

✅ 핵심 지시는 맨 앞이나 맨 끝에 둔다

❌ 길수록 정확하다고 믿는다

✅ 길이는 비용이자 노이즈 - 충분한 만큼만

❌ 절차를 하나하나 떠올린다

✅ 목표와 성공 기준을 주고 맡긴다

❌ 금지 사항을 잔뜩 나열한다

✅ 원하는 행동을 긍정형으로 지시한다

❌ 규칙을 특정 사례로 못 박는다

✅ 경계를 원리로 적는다

❌ "이거 괜찮지?"로 확인받는다

✅ "약점과 반례부터 찾아줘"로 검증한다

❌ 한 번 잘 나왔으니 늘 그럴 거라 믿는다

✅ 출력은 확률 - 중요한 건 매번 다시 확인

💡 💡 배달의민족 팀은 이 원칙들(마크다운으로 구조화 + 규칙을 잘게 나누기 + 결과 검증)을 적용해서, 규칙 위반이 **5번 중 1~2번에서 10번 중 1번 미만**으로 줄었습니다. 원리는 같아요 — 우리 프롬프트에도 그대로 통합니다.

◆ 이 PART의 핵심 정리

- "어떻게"(절차) 대신 "무엇이 성공인지"(기준)를 준다
- 금지 나열 대신 긍정형·원리로 지시한다
- "괜찮지?" 대신 "약점부터 찾아줘"로 검증한다
- 출력은 확률 — 중요한 건 매번 다시 확인