

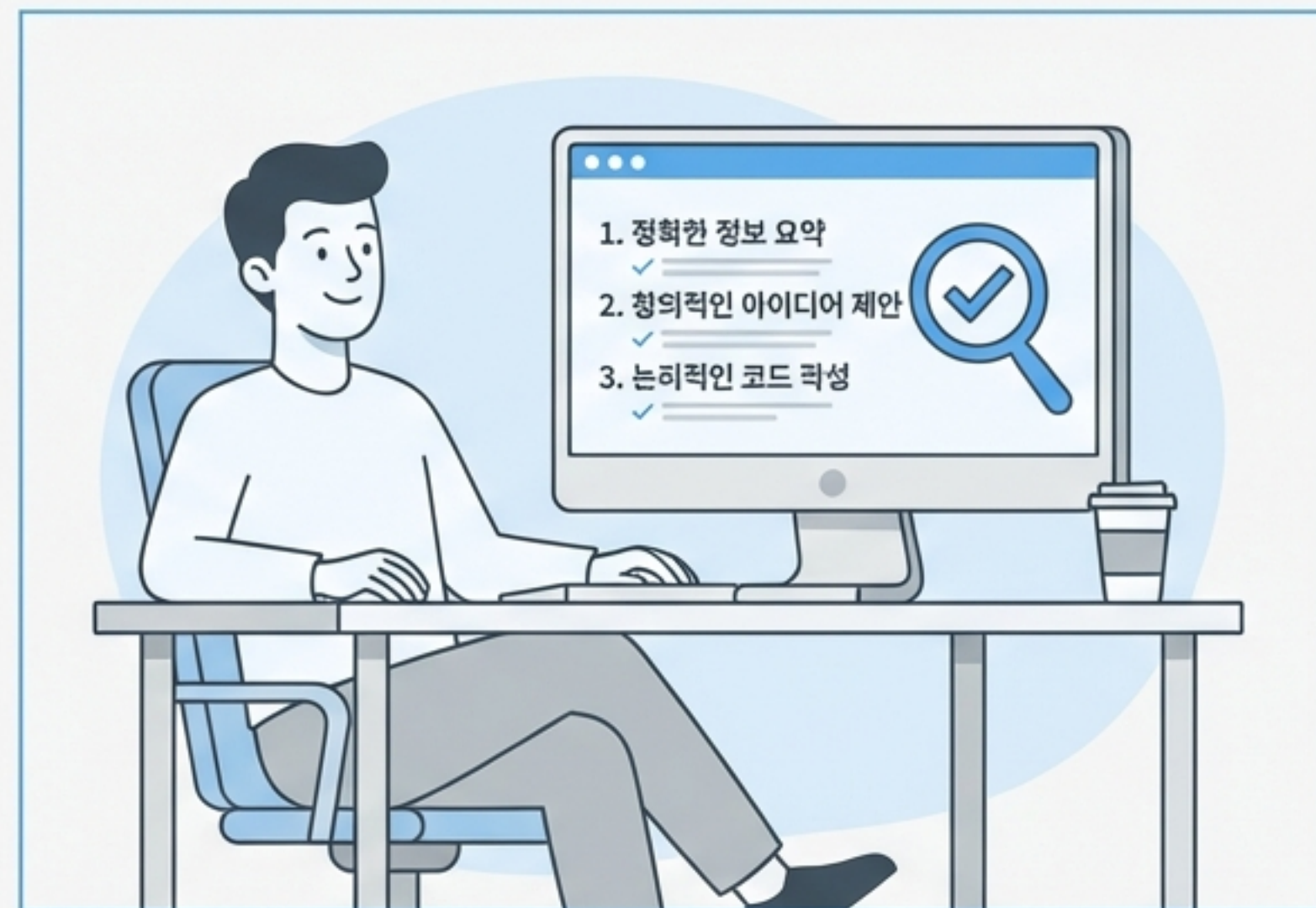
내 AI는 왜 엉뚱한 대답만 할까?

초보자를 위한 AI 프롬프트 설계 가이드북: 질문의 기술이 결과의 차이를 만듭니다.

BEFORE



AFTER

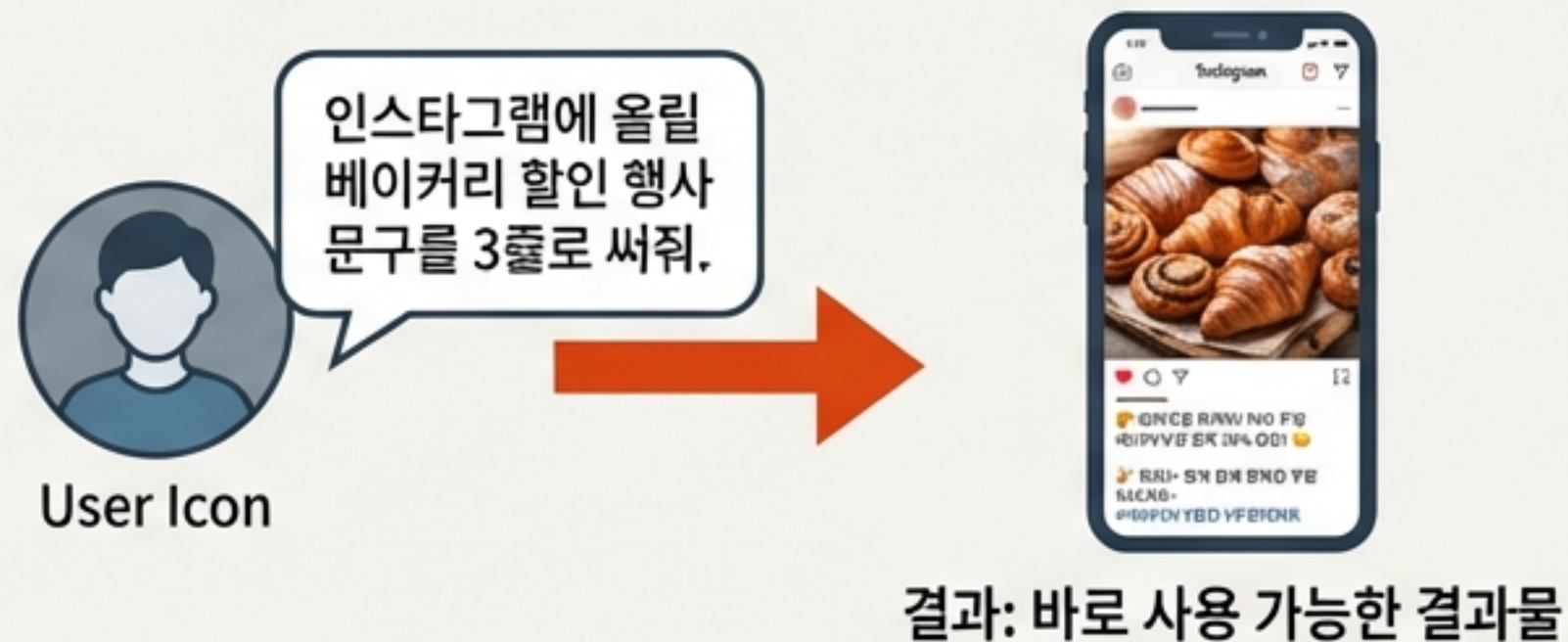


본 가이드는 AI SaaS 사용을 시작하는 분들을 위한 '언어 소통' 매뉴얼입니다.

AI는 ‘눈치 없는 서울대생 인턴’입니다.



- AI는 독심술사가 아닙니다.
- 당신이 알려주기 전까지는 백지상태입니다.
- AI는 확률적으로 다음에 올 단어를 예측하는 기계입니다.



Key Insight: 입력의 품질이 출력의 품질을 결정합니다 (Garbage In, Garbage Out)

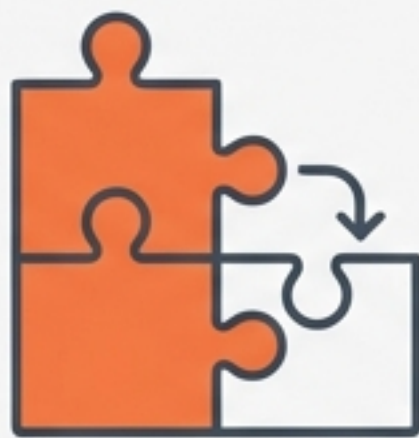
프롬프트 엔지니어링: AI와 대화하는 ‘언어’ 배우기

프롬프트 엔지니어링이란 AI로부터 최상의 결과를 얻기 위해 지시문(Prompt)을 구조화하고 다듬는 기술입니다. 코딩이 아니라 "다" ‘소통’입니다.



1. 명확성 (Clarity)

모호한 표현 대신 구체적인 지시어를 사용하세요.



2. 맥락 (Context)

배경 정보와 의도를 설명하세요.



3. 구체성 (Specificity)

원하는 형식과 제약 조건을 정해주세요.

초보자를 위한 만능 공식: RTF 프레임워크



Before & After

Bad Prompt

~~재택근무에 대해 써줘.~~



Good Prompt (RTF 적용)

인사 담당자로서(Role), 재택근무의 장점 3가지를 **보고서**
형식으로 작성해줘(Task). **글머리 기호를 사용해(Format)**

RTF 요소를 명확히 적용하여 AI의 이해를 높입니다.

RTF를 넘어 전문가처럼: 맥락(Context)과 대상(Audience) 더하기

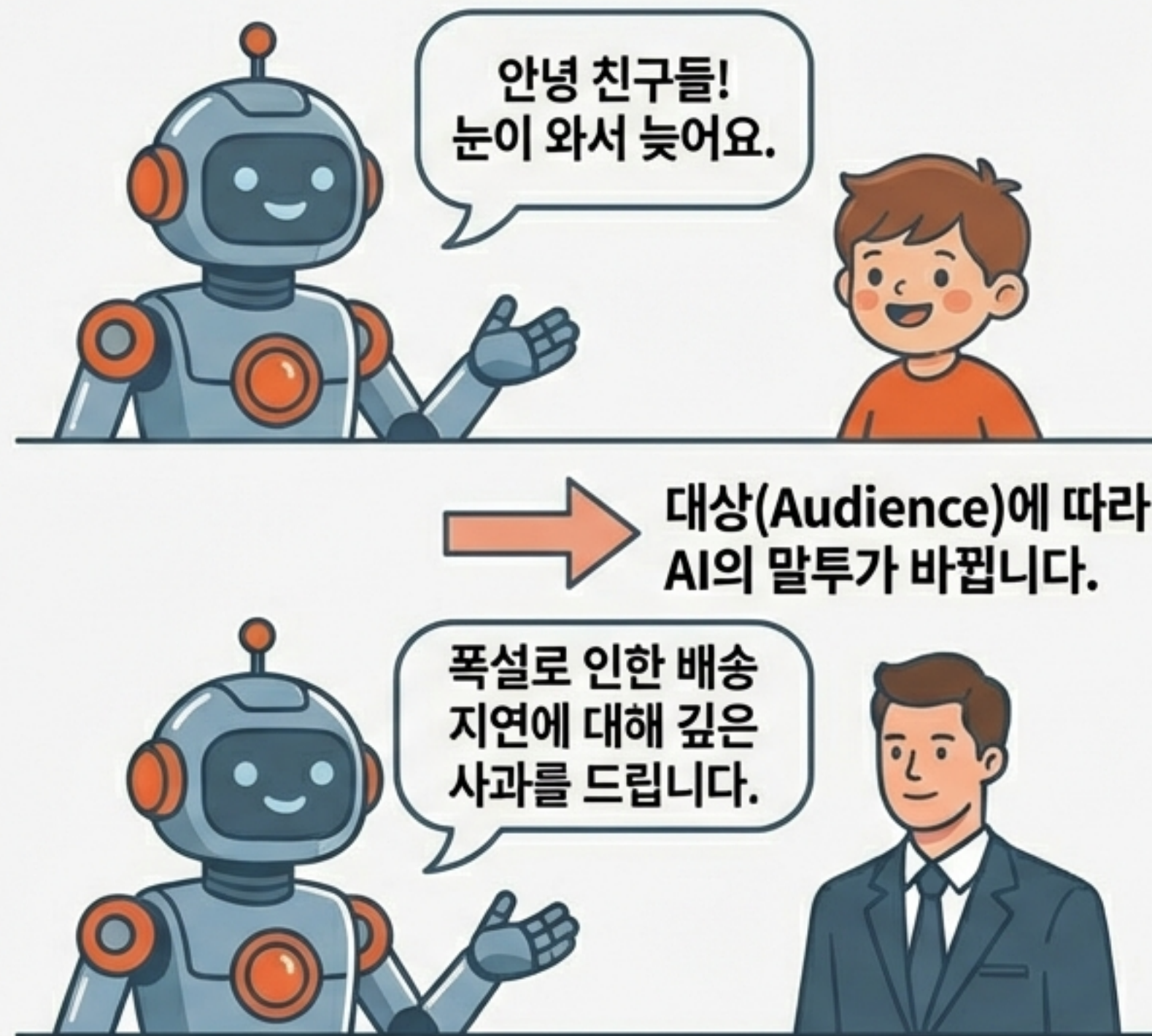
The Upgrade

Context (맥락): 왜 이 글을 쓰는가?

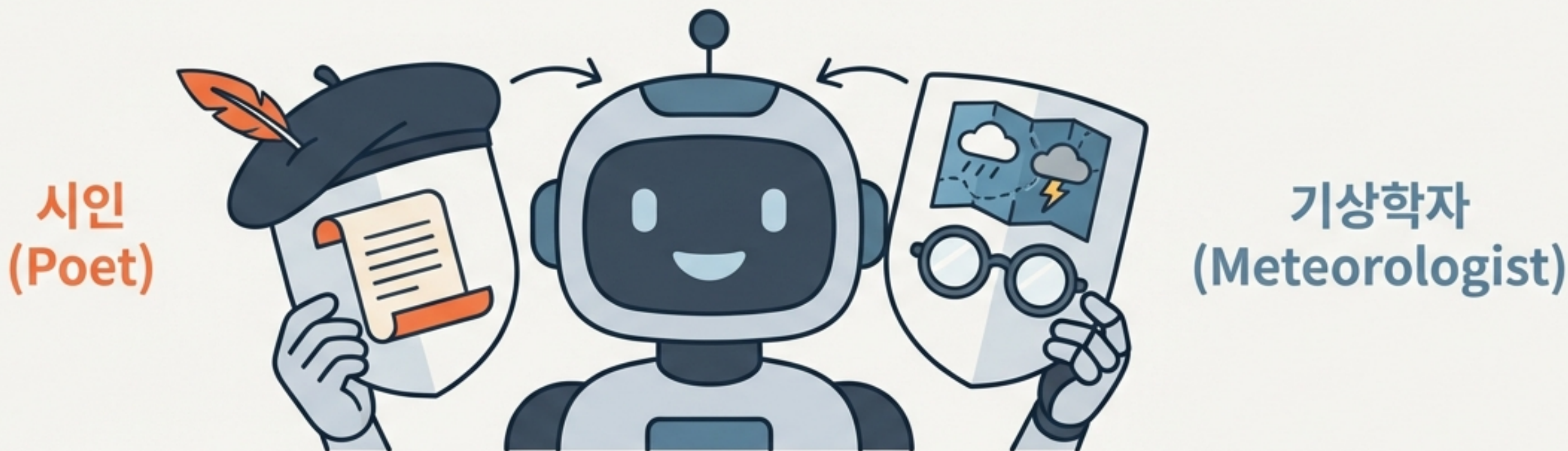
예시: "눈 때문에 배송이 지연된
상황이야."

**Audience (대상): 누가 이 글을
읽는가?**

예시: "화가 난 고객들에게 보낼 거야."



기법 1: 페르소나 설정 (Role Play) - AI에게 가면 씌우기



"비에 대해 설명해줘"
하늘에서 내리는 슬픈 눈물...

"비에 대해 설명해줘"
대기 중의 수증기가 응결하여...

Tip: 구체적일수록 좋습니다. '그냥 작가'보다는 '20년 경력의 스릴러 소설 작가'가 낫습니다.

기법 2: 예시 보여주기 (Few-Shot Prompting)

백 마디 설명보다 하나의 예시가 강력합니다.

Zero-Shot (초보)

Instruction
이 문장들을 긍정적으로 바꿔줘.

Question
회의가 너무 길어.

? (예측 불가)

Few-Shot (고수)

Instruction
원하는 패턴을 예제로 제공합니다.

입력: 음식이 너무 늦게 나왔어. ->
출력: 음식을 기다리는 시간이
길었지만, 그만큼 기대가 됩니다.

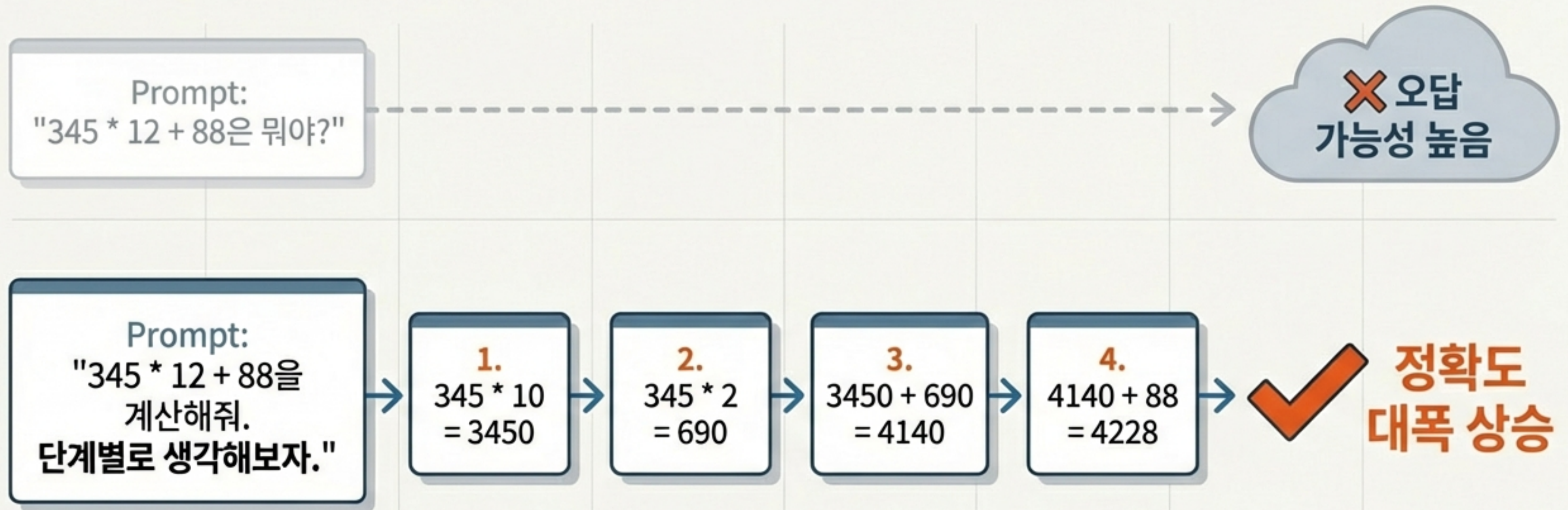
입력: 비가 와서 짜증 나. ->
출력: 비가 오니 운치 있어서 좋네요.

Question
회의가 너무 길어.

Result
회의 시간이 길어서 내용을 깊이 있게 다룰 수 있겠네요.

기법 3: 단계별로 생각하기 (Chain of Thought)

복잡한 문제를 한 번에 물어보면 AI는 답을 ‘찍어서’ 틀릴 수 있습니다.



Magic Phrase: 단계별로 생각해보자 (Let's think step by step)

기법 4: 제약 조건과 출력 형식

하지 말아야 할 것을 알려주면 할 일에 더 집중합니다.

길이 제약 (Length)



- 500자 이내로
- 3문장으로

형식 제약 (Format)



- 표(Table)로 만들어줘
- JSON 형식으로
불렛 포인트 사용



부정 프롬프트 (Negative Prompt)

- 전문 용어 쓰지 마(No jargon)
- 이모지 사용 금지

기능	Product A	Product B
가격	\$100	\$120
색상	Blue	Red
크기	Small	Large
격석	Small	Large

단순히 '표로 만들어줘' 한 마디면
텍스트 뭉치가 표로 변합니다.

초보자가 자주 하는 실수 4가지



1. 모호한 지시

'도와줘', '글 좀 써줘'는 금물.
구체적으로 말하세요.



2. 과부하 (Overloading)

한 번에 10가지를 시키지 마
세요. 작업을 쪼개세요.



3. 환각(Hallucination) 무시

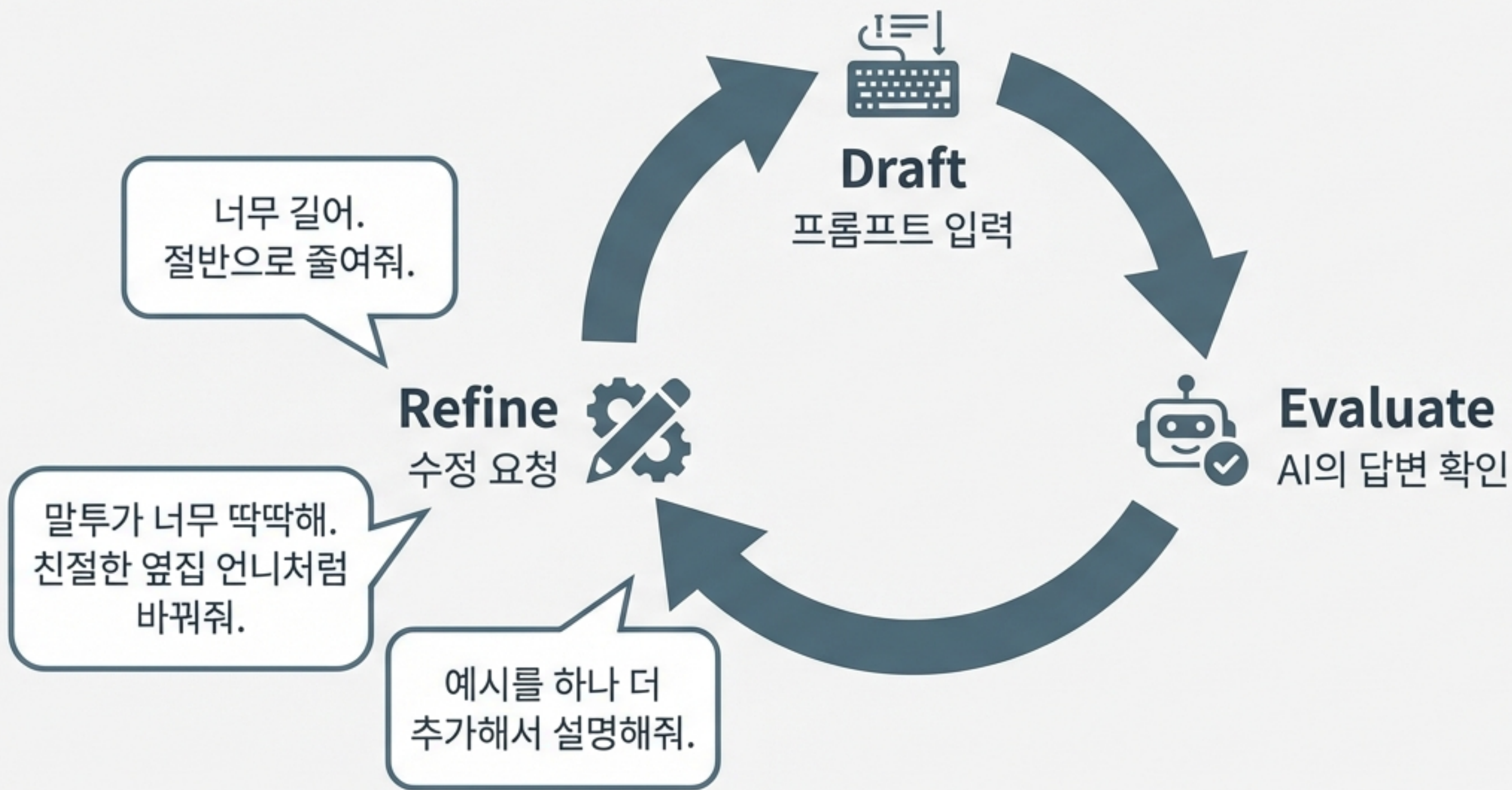
AI는 모르는 것도 아는 척할 수
있습니다. 팩트 체크 필수.



4. 반복(Iteration) 생략

첫 결과가 끝이 아닙니다.
수정 요청을 하세요.

완벽한 결과는 ‘수정(Refine)’에서 나옵니다



실전 예시 1: 정중한 고객 사과 메일 쓰기

입력 프롬프트 (Recipe)



[Role]: 고객 만족팀 팀장



[Context]: 폭설로 인해 배송이 3일 지연됨



[Task]: 정중한 사과 이메일 작성

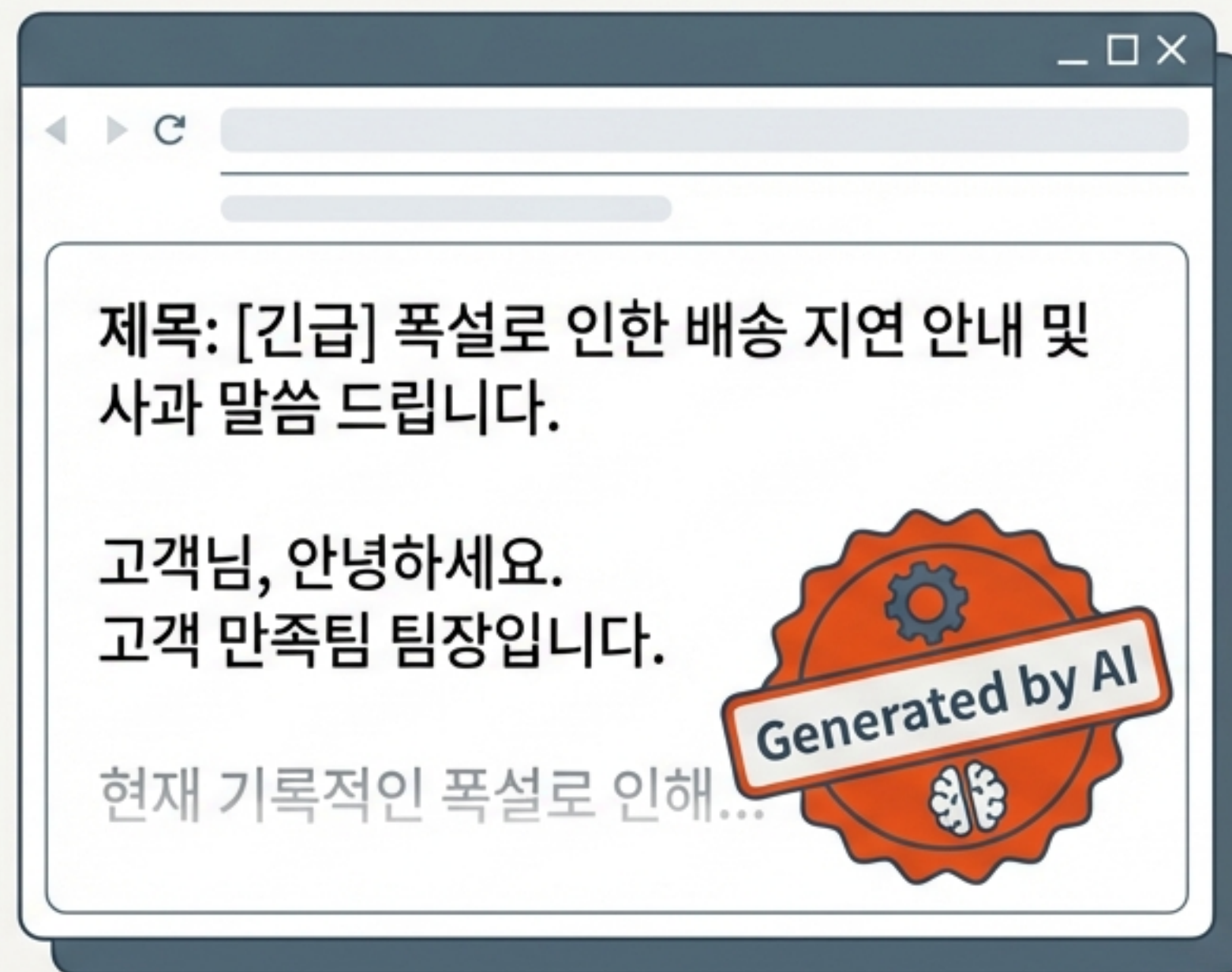


[Audience]: 배송을 기다리다 지친 고객



[Tone]: 공감하면서도 전문적인 톤

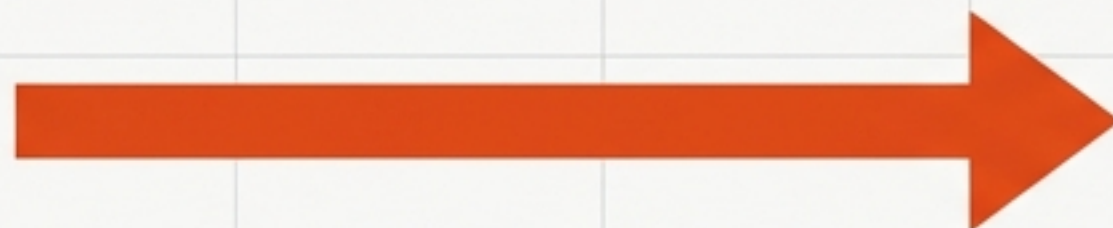
The Dish/Output



실전 예시 2: 복잡한 보고서 요약하기



복잡한 기술 보고서



[Role] 전문 기술 블로거 +
[Task] 핵심 요약 + [Target] 초보자



핵심 요약 3가지

- 기술 A는 데이터 처리를 2배 가속화합니다.
- 비용 절감 효과가 30% 발생합니다.
- 기존 시스템과 100% 호환됩니다.

한 장으로 끝내는 프롬프트 치트키

The Formula: **RTF + C** (Role, Task, Format + Context)

Magic Phrases (마법의 문장)

- 🧙 • ~의 역할을 맡아줘 (Act as...)
- 🧠 • 단계별로 생각해보자 (Let's think step by
- 💬 • 예를 들어 설명해줘 (Give me an example)
- 📄 • ~형식으로 출력해 (Format as table/list)

Checklist (체크리스트)

- ☒ 🔍 명확한가? (Clarity)
- ☐ 🌐 맥락이 있는가? (Context)
- ☐ 📋 형식을 정했는가? (Format/Constraint)
- ☐ 📄 예시를 주었는가? (Few-Shot)

지금 바로 대화를 시작하세요.

오늘 배운 RTF 공식을 사용하여 AI에게 첫 질문을 던져보세요.



프롬프트 엔지니어링은 더 나은 답변을 얻는 기술이 아니라,
더 좋은 질문을 하는 기술입니다.

감사합니다.