

MBC AI 에이전트 PoC · 1차 교육 — 실습 가이드

2026. 9. 7(월) 14:00~18:00 · 인재개발강의실 · 240분 · 3개 사이클 · mbc.mzgcp.net

MBC AI 에이전트 PoC · 1차 교육 — 실습 가이드

2026. 9. 7(월) 14:00 ~ 18:00 · 인재개발강의실

이 페이지를 노트북에 띄워 놓고 실습하십시오. 필요한 것은 전부 여기 있습니다.

오늘 만드는 것	본인 업무의 SKILL.md 1개
오늘 쓰는 데이터	본인 실데이터 1건 (가상 예시로 하지 않습니다)
구성	3개 사이클 — ① 문서·서식 표준화 ② 실데이터 리포트 출력 ③ 조별 상호 비교

오늘의 기준 — 이 표만 지키면 됩니다

제약 : 입력 1종 · 출력 1종 · 처리 2단계 이내

입력 1종	파일 개수가 아니라 종류입니다. 월별 실적 엑셀이면 3개든 30개든 1종. 엑셀 + PDF 계약서 + 메일 본문은 3종이라 오늘 범위 밖
출력 1종	산출물 1가지. 표면 표, 문서면 문서
2단계 이내	판단이 일어나는 지점이 2번까지

범위를 넘는 것은 버리지 말고 과제 카드에 2차 확장 후보 로 적어 두십시오. 잘라내는 것도 설계입니다.

좁게 만드는 이유가 있습니다. 범위가 넓으면 결과가 나빠도 어디가 문제인지 알 수 없습니다.

기준선 : 재생성 3회 이내 · 수정률 30% 이하

수정률 = (수정 + 삭제 + 추가한 문단 수) ÷ 전체 문단 수 × 100

- 단위는 문단. 표는 행 단위로 셉니다
- 오타자 1건만 고쳐도 그 문단은 수정 1건
- 오늘 하루 샘플을 바꾸지 마십시오. 사람마다 다르게 세면 비교가 안 됩니다

이 숫자는 합격선이 아닙니다

넘기면 통과, 못 넘기면 실패가 아닙니다. 대화를 여는 기준점입니다.

결과	다음에 물을 것
도달했다	어떤 규칙이 그걸 만들었는가? → 그게 조 공용 자산 후보입니다
못 미쳤다	어디서 막혔는가? → 그게 한계 기록입니다

두 숫자가 서로 다른 것을 봅니다.

지표	무엇을 보는가
수정률 30%	결과물 자체가 쓸 만한가
재생성 3회	거기 도달하는 데 드는 비용. 실무에서 아무도 20번 돌리지 않습니다

어디서 온 숫자인가 — 사전 설문에서 "수정 10~30%면 쓴다"가 53.2%로 과반이었고, 절반 이상 고쳐야 하는 결과물은 쓰지 않는다는 응답이 나왔습니다. 여러분이 직접 말한 수용 한계선입니다.

내 업무에는 안 맞을 수 있습니다. 근거 인용처럼 틀리면 안 되는 항목이 있는 업무는 그 항목만 보면 0%여야 하고, 초안 생성 성격이면 40%도 이득입니다. 숫자보다 "무엇이 틀렸는지"가 중요합니다. 못 미쳤을 때 막힌 지점을 한 문장으로 적는 이유가 이것입니다.

스킬 이름 규칙 : <이니셜>-<용도>-0907

예 — jh-dept-format-0907 · jgy-weekly-report-0907

⚠ test · demo 를 쓰지 마십시오

스킬 이름(SKILL_ID)은 영구 예약입니다. 삭제해도 그 이름을 회사에서 다시 못 씁니다. gcp- 로 시작하는 이름도 금지입니다. 소문자·숫자·하이픈만, 1~63자.

반드시 넣을 규칙 4개

어떤 과제든 SKILL.md에 이 4줄이 들어가야 합니다.

#	상황	이렇게
①	값이 없을 때	없음 — 0으로 채우지 않습니다
②	확신이 없을 때	판단 보류 + 사유 한 줄
③	원문과 불일치할 때	고치지 말고 불일치: 원문 000 / 출력 000
④	범위 밖	자동 실행 · 쓰기 · 전송 · 결제 상신은 하지 않습니다

왜 이 4개인가

넷 다 "AI가 모른다는 사실을 숨기지 못하게 하는" 규칙입니다.

도구는 빈칸을 싫어해서 무엇으로든 채우려 합니다. 그 채움이 그럴듯할 때가 가장 위험합니다 — 틀린 걸 알아채지 못하면 그대로 나갑니다.

① 없음 — 0으로 채우지 않는다

0은 "없음"이 아니라 "0이라는 값"입니다. 빈칸을 0으로 채우면 합계가 맞아 보이고, 맞아 보이면 아무도 검산하지 않습니다.

상황	0으로 채우면	없음 으로 두면
어느 해에 그 항목이 아예 없었다	그 해 실적이 0원이었던 것과 구분 불가	항목 부재가 그대로 보임

상황	0으로 채우면	없음 으로 두면
지표 하나를 못 받았다	증감률이 -100%로 왜곡	그 칸만 확인하면 됨

② 판단 보류 + 사유 한 줄

이 규칙이 수정률을 가장 크게 낮춥니다.

	틀린 값이 들어오면	판단 보류 면
해야 할 일	① 틀렸다는 걸 알아채고 ② 원문 찾아 대조하고 ③ 고침	그 줄만 채움
못 알아채면	그대로 나감	—

사유를 한 줄 붙이는 이유가 있습니다. 사유가 쌓이면 그게 곧 함정(Pitfalls) 섹션이 됩니다.

③ 불일치는 고치지 말고 표기한다

검증하라고 시켰는데 도구가 조용히 맞춰 놓으면, 검증한 게 아니라 증거가 사라진 것입니다.

틀린 걸 찾는 것이 목적인 업무에서는 이게 결정적입니다. 대조·검수·검증이 과제에 들어 있다면, 맞춰 주는 도구는 그 업무를 정확히 반대로 수행합니다. 어느 쪽이 맞는지 사람이 판단합니다 — 도구는 **양쪽 값을 나란히** 보여주지만 합니다.

④ 자동 실행 · 쓰기 · 전송 · 결제 상신 금지

앞 셋과 성격이 다릅니다. 품질이 아니라 PoC 범위와 안전 문제입니다.

이번 PoC의 범위는 초안 생성까지입니다. 메시지 발송·데이터 쓰기·결제 상신은 "실제로 보내면" 사고입니다. 그래서 규칙으로 두지 말고 SKILL.md의 **하지 말 것**에 문장으로 박아 두십시오.

네 규칙의 공통점 — 전부 *****"모르면 모른다고 표시하라"*****의 변형입니다. 없으면 모름 · 애매하면 보류 · 다르면 표기 · 실행은 사람이. 이 표시가 있어야 사람이 볼 곳을 알고, 그래서 수정률이 떨어집니다.

오늘 제출하는 것 4가지

1. SKILL.md v1 — 등록까지
2. 실행 로그 — 2회 이상 실행 기록 (실패한 실행도 반드시)
3. 조별 상호 리뷰 시트 — 조별 1장
4. 한계 기록 시트 — 2건 이상 (아래 참고)

"안 됨"은 실패가 아니라 오늘의 산출물입니다

이 PoC의 목적은 "AI가 되네"를 확인하는 게 아니라 어디까지 되고 어디서 안 되는지 특정하는 것입니다. 잘 된 것만 적으면 아무것도 판정할 수 없습니다. 안 된 것을 그대로 적으십시오. 좋게 적을 필요가 전혀 없습니다.

오늘의 진행

시간	무엇을	내가 할 일
14:00	오리엔테이션	오늘의 기준 확인

시간	무엇을	내가 할 일
14:15	사이클 1 시연	보기
14:40	사이클 1 실습 (40분)	SKILL.md 작성 → 등록 → 첫 실행
15:20	사이클 1 체크포인트	구조 점검받기
15:35	휴식	
15:45	사이클 2 시연	보기
16:05	사이클 2 실습 (35분)	실데이터 투입 → 출력 고정 → 개선
16:40	사이클 2 체크포인트	지표 점검받기
16:55	휴식	
17:05	사이클 3 실습 (35분)	조원과 비교 → 개선점 적용 → 재실행
17:40	사이클 3 체크포인트	개선 결과 점검받기
17:55	랩업	제출

실습은 시간이 되면 완성 여부와 관계없이 멈춥니다. 미완성으로 체크포인트에 들어가는 것이 정상입니다.

막히면 같은 조 조원에게 먼저 물어보시오. 조마다 코딩 역량자가 있습니다. 같은 조는 과제 구조가 같아서 대개 같은 지점에서 막힙니다.

왜 세 번 하는가 — 사이클의 차이

같은 것을 세 번 고치는 게 아닙니다. 무엇이 들어오는지 바뀌면서 새로 배울 것이 생깁니다.

	사이클 1	사이클 2	사이클 3
주제	문서 및 서식 표준화	실데이터 리포트 출력·고정	조별 상호 비교와 개선
무엇이 들어오는가	내가 쓴 것 (메모·초안)	남이 준 실데이터 (양식 제각각)	내가 만든 산출물
핵심 난점	양식·표기를 지키는 것	값이 매번 다른데 형식은 고정해야 하는 것	내 눈에 안 보이는 것
새로 배우는 것	출력 규격에 실물 양식 붙여넣기	없을 때·모를 때 처리 "적절히" → 조건 출력 형태 전환	상호 검증 조 공용 기준
누가 검증하는가	강사 (구조 점검)	지표 (수정률·재생성)	조원
시연	있음	있음	없음

각 사이클이 나머지 둘로는 못 하는 것

사이클 1 — 데이터를 내가 통제합니다. 그래서 결과가 나쁘면 원인이 내 규격 하나로 좁혀집니다. 입력까지 흔들리는 상태에서 규격을 배우면 무엇이 문제인지 알 수 없습니다.

사이클 2 — 값이 통제 밖으로 나갑니다. 여기서 처음 만나는 문제가 셋입니다. 값이 비어 있을 때, 판단이 필요할 때, 그리고 같은 규격인데 실행마다 형식이 흔들릴 때. 사이클 1에서는 입력이 일정해서 이 문제가 드러나지 않습니다. 그래서 동일 입력 3회 실행이 사이클 2에 있습니다.

사이클 3 — 아무것도 새로 만들지 않습니다. 이미 만든 것을 남의 눈으로 검증합니다.

왜 앞의 둘로 대체가 안 되는가 — 자기 결과물의 지어내기는 안 보입니다. 원문에 없는 수치가 그럴듯하게 들어와 있을 때, 만든 사람은 그게 자기가 의도한 것처럼 읽습니다. 옆 사람은 원문과 대조해 바로 찾아냅니다.

그리고 같은 조는 과제 구조가 같으니 *****가장** 잘된 규격 하나를 골라 나머지가 베끼는 것***이 성립합니다. **조 공용 자산은 여기** 서만 나옵니다.

Skill이 무엇인가 (5분)

Skill은 반복 업무를 처리하는 방법을 미리 적어 둔 업무 매뉴얼입니다.

Gemini Enterprise 공식 정의로는 **“어시스턴트가 특정 작업을 수행하도록 돕는 재사용 가능한 맞춤 지침”**이며, `SKILL.md` 파일 하나가 중심입니다.

AI에게 스킬을 준다는 것은 신입에게 상세한 업무 매뉴얼을 건네는 것과 같습니다. 매뉴얼이 없으면 “뭔가 나오긴 하는데 평범한” 수준에 머물고, 잘 설계된 스킬이 있으면 **같은 도구가 매번 올바른 구조를 적용**합니다.

프롬프트와 무엇이 다른가

	프롬프트	Skill
저장	안 됨 — 매번 다시 씀	<code>SKILL.md</code> 로 남음
호출	매번 전문을 붙여넣음	설명이 맞으면 자동으로 불러 옴
일관성	쓸 때마다 조금씩 달라짐	같은 규칙이 매번 적용됨
공유	내 대화창에만 남음	조 공용 자산으로 넘겨줄 수 있음
성격	소비된다	자산으로 쌓인다

마지막 줄이 이 PoC의 이유입니다. 각자 프롬프트를 잘 쓰는 것으로는 **조직에 아무것도 남지 않습니다**.

모든 스킬의 3요소

요소	질문	SKILL.md의 어디
지식	무엇을 알아야 하는가	입력 · 배경 · 제약
절차	어떤 순서로 하는가	번호 매긴 처리 단계
도구	무엇으로 하는가	도구 · 다른 스킬

Gemini Enterprise에서 스킬 만들기

등록 경로

단계	무엇을
1	좌측 메뉴 Skills → +
2	Create skill with Gemini (대화로 만들기) 또는 Upload skill (Markdown 파일 업로드)
3	저장하면 기본 활성 상태가 됩니다
4	호출 확인 — @ 또는 / 로 직접 부르거나, 그냥 요청해서 자동으로 불러 오는지 봅니다

알아두실 것 — 검증이 비동기라 업로드 직후 즉시 반영되지 않을 수 있습니다. 잠시 기다리십시오. 자연어로 스킬을 만들 때는 대화 맥락과 커넥터를 참조하지 않습니다. 필요한 정보는 요청 안에 다 넣으십시오.

왜 description이 전부인가

Gemini Enterprise는 처음에 각 스킬의 이름과 설명만 읽습니다. 그리고 맞다고 판단했을 때만 본문 전체를 읽습니다.

본문이 아무리 좋아도 description 이 부실하면 애초에 호출되지 않습니다.
오늘 "스킬이 안 불러 와요"의 99%가 여기입니다. 그럴 때는 [P3](#)을 쓰십시오.

SKILL.md 작성법

1단계 · 이름과 설명

description 은 사람용 요약이 아니라 모델용 트리거 조건입니다

- ❌ "보고서를 만드는 스킬입니다."
- + ✅ "메모·요약을 부서 전용 양식과 표기 규칙에 맞춘 문서 초안으로 만듭니다."
- + 트리거 키워드: 기안서, 제안서, 부서 양식, 양식 맞추기, 초안 정리, 표기 규칙.
- + 입력: 메모·요약 텍스트 1건 / 출력: 부서 양식 문서 초안 1건.
- + 회사 표준 서식(HWPX·PPTX) 변환에는 사용하지 않습니다 — 사내 「MBC 문서 작성」 사용."

4원칙

#	원칙	왜
1	트리거 키워드	내가 실제로 입력창에 칠 단어를 그대로 넣습니다. "문서 표준화"라고 적고 평소에 "양식 맞춰줘"라고 치면 안 걸립니다
2	경계 조건	쓰면 안 되는 상황도 씁니다. 사내에 이미 있는 도구와의 경계를 여기에
3	입력·출력	무엇을 받아 무엇을 만드는지
4	우선순위 힌트	같은 조의 다른 분들도 비슷한 스킬을 만듭니다. 이게 없으면 서로 잘못 불러 옵니다

2단계 · 본문 — 순서가 중요합니다

출력부터 정하십시오

대부분 "입력을 어떻게 읽지"부터 시작해서 헤맬니다. **최종 산출물이 어떻게 생겨야 하는지가 정해지지 않으면 처리 단계를 정할 수 없습니다.**

그리고 출력 규격에는 **실물 양식을 파일에서 복사해 그대로 붙여넣으십시오.** 말로 서술한 규격은 매번 다른 결과가 나옵니다. **이 한 가지가 수정률을 가장 크게 낮춥니다.**

순서	무엇을	요령
①	출력 규격	실물 양식을 복사해서 붙여넣기. 표면 열 이름·순서·형식 고정, 문서면 섹션 제목 고정
②	처리 2단계	각 단계에 판단 기준을 조건으로 . "적절히"."중요한 것"."알아서"는 쓰지 않습니다
③	입력	형식 1종 · 구성(열 이름·섹션 이름 그대로) · 빈 값이 올 수 있는 항목
④	예시	실제 입력 1건 → 이상적 출력 1건. 없으면 도구가 입력 형태를 추측 합니다
⑤	하지 말 것	범위 밖 동작 명시

"적절히"를 조건으로 바꾸는 법

- ❌ 중요한 항목을 골라낸다
- + ✅ 다음 중 1개 이상 해당하는 항목만 표시한다
 - + ① 증감률 $\pm 10\%$ 이상
 - + ② 절대 증감액이 기준선 이상
 - + ③ 계약 조건과 불일치
- + - 모든 지적에 수치 근거 필수 (몇 % · 몇 원)
- + - 근거 없는 지적은 출력하지 않는다

3단계 · 템플릿 (복사해서 채우십시오)

```
---
name: <이니셜>--<용도>-0907
description: |
  <무엇을 하는 스킬인지 한 문장>
  트리거 키워드: <실제로 칠 단어들을 나열>
  입력: <형식 1종> / 출력: <형식 1종>
  <이런 요청에는 사용하지 않습니다 - 경계 조건>
metadata:
  category: Productivity
---
```

<스킬 이름>

목표

<한 문장. "~을 만든다"로 끝낸다>

입력

- 형식: (xlsx / pdf / 텍스트 - **1종만**)
- 구성: (열 이름·섹션 이름을 그대로)
- 없을 수 있는 항목: (빈 값이 올 수 있는 항목)

절차

1. <1단계> - 판단 기준: <조건을 나열. "적절히"는 쓰지 않는다>
2. <2단계> - 판단 기준: <조건을 나열>

출력 규격

<실물 양식을 여기에 그대로 붙여넣는다>

- 표인 경우 열: | 열1 | 열2 | 열3 | ← 이름·순서·형식 고정
- 문서인 경우 섹션 제목 고정
- 값이 없을 때: "없음" ← 0으로 채우지 않는다
- 확신이 없을 때: "판단 보류" + 사유 한 줄
- 원문과 불일치할 때: 고치지 말고 "불일치: 원문 000 / 출력 000"

예시

입력 예시

<실제 입력 1건을 그대로>

출력 예시

<그 입력에 대한 이상적 출력 1건>

예외 처리

- 정보가 부족하면 추측하지 않고 되묻는다
- <내 업무 특유의 예외>

함정 (Pitfalls)

- <이걸 몰랐으면 틀렸을 내용만, 일반 상식은 쓰지 않는다>

품질 체크리스트

- [] <규칙을 지켰는지 확인할 항목>

하지 말 것

- 원문에 없는 내용·수치를 만들어 넣지 않는다
- 자동 실행 · 쓰기 · 전송 · 결재 상신을 하지 않는다 (이 PoC의 범위 밖)

4단계 · 테스트와 개선

1. 등록하고 실제로 그 작업을 요청해 봅니다
2. 자동으로 불러 왔는가 → 안 되면 **description** 문제 → [P3](#)
3. 결과가 기대에 맞는가 → 안 되면 본문 문제 → [P5](#)
4. 고치고 다시 실행. 매 실행을 실행 로그에 적습니다

첫 시도에 완벽할 수 없습니다. 2~3회 수정은 완전히 정상입니다. 목표는 완성이 아니라 3회 이내에 수정률 30% 이하가 나오는지 확인하는 것입니다.

자주 하는 실수 3가지

실수	증상	해결
한 스킬에 여러 작업	5가지 일을 하면 언제 적용할지 도구가 혼란	작업 하나에 스킬 하나
모호한 설명	"보고서 만드는 스킬" → 호출 안 됨	트리거 키워드를 구체적으로
추상적 지침	"알아서 잘 만들어"	"다음 3개 조건 중 2개 이상 해당하는 항목만"

함정(Pitfalls) 섹션이 가장 중요합니다

당연한 것을 쓰지 마십시오. 도구는 이미 일반 관행을 압니다. 아는 것을 반복하면 신호가 희석됩니다. *******이걸 몰랐으면 틀렸을 내용***만 씁니다.

실습 중 실패할 때마다 한 줄씩 추가하십시오. 오늘 만난 한계가 곧 함정 섹션입니다.

함정 (Pitfalls)

- 빈 칸을 0으로 채우지 않는다 → 합계가 맞아 보여서 오류를 못 찾는다
- 종결을 `~됩니다`로 바꾸지 않는다 → 부서 표기 규칙은 `~함`·`~임`
- 불일치를 조용히 고치지 않는다 → 증거가 사라진다

사이클 1 · 문서 및 서식 표준화 (14:40~15:20 · 40분)

내가 할 일

본인 업무로 SKILL.md를 씁니다. 15:20에 완성 여부와 관계없이 멈춥니다.

순서	분	무엇을
1	5	오늘 만들 범위 확정 — 막히면 P1

순서	분	무엇을
2	15	출력 규격부터 — 실물 양식을 복사해 붙여넣기
3	10	처리 2단계에 판단 기준을 조건으로 + 예시 1쌍 — 막히면 P2
4	5	등록 → 자동 호출 확인 — 안 되면 P3
5	5	첫 실행 → 안 된 것을 한계 기록 시트에 1건

이 사이클의 산출

- ☐ SKILL.md v0 — 등록 완료
- ☐ 한계 기록 1건 이상

조별 유의

조	유의
1조 외부 리서치·모니터링	출처 열을 필수로 두십시오. "원문에 없는 내용은 쓰지 않는다"를 규칙 맨 위에 — 조사형이 지어내기가 가장 잘 나옵니다
2조 문서 생성·자료 정형화	실물 양식 파일이 있어야 진행됩니다. 없으면 아무 문서 1건이라도 구조를 그대로 옮겨 쓰십시오
3조 근거 인용·자료 검증	존재하지 않는 문서번호·판례번호를 만들어 내는 것이 알려진 실패입니다. "확인 불가 시 비워 둔다"를 맨 위에. 개인정보는 반입 전 가림
4조 정기 산출물 자동 생성	지표 통합이 본체이니 여기서는 출력 표 스키마 고정까지만. 본론은 사이클 2
5조 사내 시스템 연동	권한을 기다리지 마십시오. 시스템에서 내려받은 파일 1건을 입력으로 쓰고, 로그에 연동 미비 · 파일 대체 표기. 쓰기·전송·결재 상신은 범위 밖
6조 데이터 검수·통계	판정 업무라 "적절히"가 가장 많이 남습니다. 위반 유형을 조건으로 나열하십시오
검증 트랙	SKILL.md 대신 정답지 초안(양식 07) — "맞았다고 판정할 조건" 5개. 읽고 O/X를 바로 찍을 수 있어야 합니다

체크포인트에서 확인하는 것 (15:20)

- ☐ 입력이 1종으로 특정되었는가
- ☐ 출력 규격에 실물 양식이 들어갔는가 — 말로 서술한 것은 미충족
- ☐ 처리가 2단계 이내이고 각 단계에 **판단 기준(조건)**이 있는가
- ☐ 한계 기록이 1건 이상 있는가

사이클 2 · 실데이터 리포트 출력·고정 (16:05~16:40 · 35분)

내가 할 일

본인 실데이터 1건을 넣습니다. 부문별 반입 규칙을 적용하십시오 — 취재원·개인정보는 가립니다.

순서	분	무엇을
1	10	실데이터 투입 → 실행 → 실행 로그 기록
2	8	출력 형식이 흔들리는지 확인 — 동일 입력 3회 실행 해 같은 형식인지
3	10	개선 → 재실행 — 막히면 P4 · P5
4	5	여유 되면 출력 규격만 바꿔 이메일·기안 형태로 한 번 뽑기
5	2	안 된 것을 한계 기록 에 1건 추가

실행 로그 쓰는 법

매 실행을 적습니다. 실패한 실행도 반드시 적으십시오 — 실패 기록이 없는 로그는 검증 자료로 쓸 수 없습니다.

항목	규칙
매 실행	# · 일시 · 무엇을 바꿨는가 · 입력 규모 · 결과 · 수정률 · 판정
수정률	문단(표는 행) 단위
샘플을 썼으면	샘플 표기 → 채택 여부 지표에서 제외됩니다
연동 못 했으면	연동 미비 · 파일 대체 표기

출력 형태는 하나가 아닙니다

같은 SKILL.md에서 출력 규격 섹션만 바꾸면 다른 형태가 나옵니다.

형태	규격에 무엇을 박는가
표	열 이름 · 순서 · 데이터 형식
이메일	제목 규칙 · 첫 3줄 요약 · 본문 항목 순서 · 맺음 문구
기안 · 게시글	섹션 제목 고정 · 배경 / 현황 / 조치 순서 · 첨부 표기

바뀌는 건 출력 규격 섹션 하나뿐입니다. 목표·입력·처리·판단 기준은 그대로입니다. 정형 문서를 안 만드는 직군도, 메일이나 공지를 쓰신다면 오늘 배운 게 그대로 적용됩니다.

체크포인트에서 확인하는 것 (16:40)

- ☐ 실행 로그에 **2회 이상** 실행 기록이 있는가
- ☐ 수정률을 **문단 단위**로 썼는가
- ☐ **3회 이내** 수정률 **30% 이하**에 도달했는가
- ☐ **실데이터**로 실행했는가 (샘플이면 표기했는가)
- ☐ **한계 기록**이 누적 2건 이상인가

미도달이면 막힌 지점을 한 문장으로 적으십시오. "그냥 잘 안 된다"는 안 됩니다.

예 — “출력 표의 열 순서는 고정됐는데 금액 형식이 매 실행마다 달라진다” “입력 PDF의 표를 인식하지 못해 3단계가 필요해진다”

사이클 3 · 조별 상호 비교와 개선 (17:05~17:40 · 35분)

이 사이클에는 시연이 없습니다. 조원들이 각자 만든 산출물을 서로 놓고 비교해 개선점을 찾고 그 자리에서 적용합니다. 같은 조는 과제 구조가 같습니다. 같은 구조인데 결과물이 다르다면 그 차이가 곧 개선점입니다.

진행 (05_사이클3_조별상호리뷰_시트.md에 기록)

순서	분	무엇을
1	5	산출물 내놓기 — 결과물을 출력하거나 화면에 띄워 조에 공개
2	10	상호 비교 — 조원 전원의 것을 4개 축으로 읽고 시트에 표시
3	5	가장 잘된 것 1개 를 조에서 합의 + 이유
4	12	개선점 3개 적용 → 재실행
5	3	개선 전/후 수정률 변화 기록

비교 4개 축

축	서로 확인할 것
① 출력 규격	실물 양식이 들어갔는가 / 서술로 때웠는가. 누구 규격이 가장 안 흔들리는가
② 판단 기준	조건으로 적었는가 / “적절히”가 남아 있는가
③ 없을 때·모를 때	없음 · 판단 보류 규척이 있는가. 없으면 지어냄이 섞여 있을 것
④ 지어내기	원문에 없는 값이 출력에 있는가

④가 이 사이클의 핵심입니다

자기 결과물의 지어내기는 잘 안 보입니다. 남의 것에서는 보입니다. 서로 원본과 대조해 찾아 주십시오. 이게 오늘 가장 값진 15분입니다.

개선점 쓰는 법

- 3개까지만. 많으면 12분에 적용이 안 됩니다
- “어느 섹션을 어떻게 고칠지”로 씁니다 — “더 잘 만들자”는 개선점이 아닙니다
- 조에서 가장 잘된 규척 1개를 골라 나머지가 베끼십시오. 이것이 조 공용 자산의 첫걸음입니다

프롬프트

비교가 막히면 [P6](#) 을 쓰십시오.

한계 기록 — 오늘 가장 중요한 산출물

07_한계기록_시트.md 에 2건 이상 적습니다.

원인 4분류

분류	뜻	판별 질문
① 규칙 문제	내 SKILL.md가 부족했다	규격·조건을 고치면 해결되는가?
② 도구 한계	같은 지시로도 도구가 못 한다	같은 규칙을 사람이 받으면 할 수 있는가? 그렇다면 도구 한계
③ 데이터 문제	입력 자체가 부족·불량	데이터를 바꾸면 되는가?
④ 판단 불가	아직 모르겠다	—

②를 고르기 전에 반드시 ①을 의심하십시오. 대부분은 규칙 문제입니다. "규칙을 어떻게 고쳐도 안 된다"를 확인한 뒤에만 ②로 분류하십시오.

반대로 정말 도구 한계인 것을 규칙 문제로 적으면 2차에서 같은 벽에 다시 부딪힙니다. 솔직하게 적으십시오.

구체적으로 적는 법

✗ 이렇게 쓰지 마십시오	✓ 이렇게 쓰십시오
"결과가 이상함"	"출력 표의 열 순서는 고정됐는데 금액 형식이 매 실행마다 달라짐 (1,000 / 1000 / 1천)"
"잘 못 함"	"종결을 ~함 으로 지정했는데 3문장 중 1문장이 ~됩니다 로 나옴"
"안 읽음"	"입력 PDF의 표를 인식하지 못해 값이 전부 빠짐. 같은 표를 텍스트로 붙여넣으면 정상"
"느림"	"17건 처리에 4분 소요. 손으로는 20분이라 이득이지만, 실무 30건이면 대기가 부담"

반복 실행과 실행 로그 — 실제로 어떻게 하나

반복에는 두 종류가 있습니다

섞으면 측정이 무의미해집니다.

	A. 재생성	B. 개선 후 재실행
스킬을	안 고침	고침
입력을	갈게	갈게

	A. 재생성	B. 개선 후 재실행
무엇을 재나	일관성 — 같은 형식이 나오나	개선 효과 — 수정률이 떨어졌나
언제	사이클 2의 동일 입력 3회 실행	개선 루프

로그에 어느 쪽인지 구분해 적으십시오. "무엇을 바꿨는가" 칸이 비어 있으면 A, 채워져 있으면 B입니다.

반복하는 법

- 1. 첫 요청 문구를 메모에 복사해 둡니다. 매번 그걸 붙여넣습니다
- 2. 같은 첨부 파일을 씁니다
- 3. 매 실행을 로그에 적습니다

 매 실행을 새 대화에서 하십시오

같은 대화창에서 3번 돌리면 2·3회차가 앞 결과를 보고 있습니다. 그러면 "형식이 일관된다"는 결과가 스킬 덕분인지 앞 대화를 따라한 것인지 알 수 없습니다.

일관성을 재려면 매번 새 대화여야 합니다. 첫 실행 때 한 번 확인해 보십시오 — 같은 대화에서 두 번, 새 대화에서 두 번 돌려 차이가 있는지.

요청 문구를 조금이라도 바꾸면 비교가 깨집니다. 바뀌야 할 것은 요청이 아니라 SKILL.md입니다.

"3회 이내"의 뜻

목표에 도달하기까지 고쳐서 다시 돌린 횟수입니다. 무한히 시도하지 않는다는 뜻이고, 4회 넘게 고쳐야 하면 규격 설계 자체를 다시 봐야 한다는 신호입니다.

실행 로그 — 보는 게 아니라 쓰는 것입니다

Gemini Enterprise가 자동으로 남기는 로그를 이 실습에서 쓰지 않습니다

실행 로그는 양식 04에 본인이 손으로 적는 기록입니다. 종이·엑셀·메모 무엇이든 됩니다.

왜 자동 로그를 안 쓰는가 — 이 PoC의 지표 두 개가 자동으로는 안 남습니다.

기록할 것	자동으로 남는가
언제·무엇을 요청했는지	GE 대화 히스토리에 남습니다
무엇을 바꿨는가	✗ 본인만 압니다
수정률 몇 %인가	✗ 본인이 세야 합니다
판정 (사용가능 / 보완필요 / 실패)	✗ 본인 판단입니다

결과물 자체는 대화 히스토리에서 다시 볼 수 있습니다. 판정은 사람이 적습니다.

로그 한 줄은 이렇게 생깁니다

#	일시	무엇을 바꿨는가	입력	결과	수정률	판정
1	16:05	(최초 실행)	17건	열 순서가 규격과 다름	62%	실패
2	16:18	열 순서를 규격에 명시	17건	열 순서 맞춤, 금액 형식 3종	35%	보완필요
3	16:31	금액 형식 예시를 규격에	17건	형식 일치, 빈 지표만 확인 필요	18%	사용가능

"무엇을 바꿨는가" 칸이 로그의 심장입니다

이게 비어 있으면 개선 루프를 돌릴 수 없습니다 — [P5](#)에 넣을 입력이 없습니다. 그리고 실패한 실행도 반드시 적으십시오. 위 예시의 1회차가 없으면 62% → 18%로 떨어뜨린 것이 무엇인지 남지 않습니다.

개선 루프를 스킬로 만들면 되지 않나

기술적으로는 가능합니다. 오늘은 하지 마십시오.

이유	
원인 판별을 사람이 해야 배웁니다	자동화하면 SKILL.md는 좋아질 수 있지만 "형식이 흔들리는 건 출력 규격 문제, 값 판단이 흔들리는 건 판단 기준 문제"라는 감각이 안 납니다. 그러면 2차·3차와 보완 기간에 혼자 고칠 수 없습니다. 오늘 목표는 좋은 SKILL.md 하나가 아니라 고칠 수 있는 사람입니다
스킬이 자기 자신을 고칠 수 없습니다	웹앱 Skills는 파일을 쓰지 못하고, 자동 실행·쓰기는 이 PoC 범위 밖입니다. 결국 사람이 수정률을 세서 넣고 진단을 읽고 골라 적용해야 합니다. 그 과정이 곧 핵심 작업이라 자동화할 대상이 남지 않습니다
오늘의 제약에 안 맞습니다	입력 1종·출력 1종·2단계는 실업무 과제용입니다. 개선 루프는 메타 작업이라 오늘 만들면 정작 본인 업무 스킬을 못 만듭니다

다만 2차에는 의미가 있습니다. P5를 조 공용 "스킬 리뷰어" 스킬로 만들면 조원 전원이 같은 진단 프레임을 쓰게 됩니다. 실제로 좋은 조 공용 자산 후보입니다.

오늘은 P5를 프롬프트로 붙여 쓰고, 스킬화는 [P8](#) 논의에서 후보로 올려 두십시오.

실습 프롬프트 P1~P8

Gemini Enterprise 대화창에 그대로 붙여 사용합니다. {{ }} 부분만 교체하십시오. 05_실습_프롬프트_라이브러리.md 로도 내려받을 수 있습니다.

프롬프트	언제
P1 스킬 후보 발굴	무엇을 만들지 못 정했을 때
P2 SKILL.md 초안 생성	초안을 한 번에
P3 description 개선	자동 호출이 안 될 때 (99% 여기)
P4 적대적 리뷰	내 스킬이 실패할 방법 찾기
P5 개선 루프	수정률이 안 떨어질 때

프롬프트	언제
P6 조별 상호 비교	사이클 3
P7 합정 수집	한계 기록을 합정 섹션으로
P8 조 공용 자산화	세션 간 과제

P1. 스킬 후보 발굴

나는 {{조 번호와 조명 · 예: 2조 문서 생성·자료 정형화}}에 속하고,
내 과제는 다음과 같다.

{{과제 카드의 '만들 것'과 '페인포인트'를 그대로 붙여넣기}}

이 과제를 오늘의 제약에 맞게 잘라줘.

- 입력 1종 (파일 형식 1가지, 개수는 여러 개여도 됨)
- 출력 1종
- 처리 2단계 이내

다음은 표로 제시해줘.

1. 오늘 만들 범위 (입력 / 출력 / 2단계)
2. 잘라내서 "2차 확장 후보"로 미룰 항목
3. 이 범위에서 AI가 못 할 가능성이 높은 것 1~2개와 그 이유
(한계 기록에 미리 적어둘 후보)

주의: 내 과제 조건에 이미 적혀 있는 규칙을 놓치지 말고 처리 단계나
출력 규격으로 옮겨줘. 새 규칙을 만들지 말고 내 조건을 쓰는 것이 우선이다.

P2. SKILL.md 초안 생성

Gemini Enterprise용 SKILL.md 파일을 작성해줘.

- 대상 업무: {{업무 이름}}
- 목적(한 문장): {{만들어낼 결과물}}
- 입력: {{형식 1종}}
- 출력: {{형식 1종}}
- 처리 2단계:
 1. {{ }}
 2. {{ }}
- 반드시 지켜야 할 규칙: {{과제 카드의 '조건'을 그대로 붙여넣기}}
- 실물 양식: {{출력 양식을 여기에 그대로 붙여넣기}}

작성 규격:

- YAML 프론트매터에 name(소문자·숫자·하이픈, {{이니셜}}-{{용도}}-0907 형식)과 description(1024자 이내)을 포함
- description은 "모델용 트리거 조건"으로:
 - (1) 트리거 키워드 (2) 쓰면 안 되는 경계 조건 (3) 입력·출력 (4) 우선순위 힌트
- 본문 섹션: 목표 / 입력 / 절차(번호) / 출력 규격 / 예시 / 예외 처리 / 함정 / 품질 체크리스트 / 하지 말 것
- 출력 규격에는 위에 붙여넣은 실물 양식을 그대로 넣을 것
- 다음 규칙을 반드시 포함:
 - 값이 없을 때 "없음" (0으로 채우지 않음)
 - 확신이 없을 때 "판단 보류" + 사유 한 줄
 - 원문과 불일치할 때 고치지 말고 "불일치: 원문 000 / 출력 000"
 - "하지 말 것"에 "자동 실행·쓰기·전송·결재 상신을 하지 않는다"
- 절차는 추상 표현("알아서 잘") 금지. 각 단계가 검증 가능하게
- 일반 상식은 쓰지 말고 우리 조직·부서에만 해당하는 정보에 집중

작성 후 별도로 알려줘.

- 이 스킬이 호출되지 않을 가능성이 높은 표현 3가지
- 추가로 물어봐야 할 정보

P3. description / 트리거 개선

아래는 내가 만든 스킬의 description이다.

{{현재 description 붙여넣기}}

다음 4원칙으로 평가하고 개선안을 제시해줘.

1. 트리거 키워드 - 어떤 단어가 나오면 활성화되어야 하는가
2. 경계 조건 - 쓰면 안 되는 상황이 명시되어 있는가
3. 입력·출력 명시
4. 우선순위 힌트 - 비슷한 스킬과 어떻게 구분되는가

참고 1: 같은 조의 다른 사람들도 비슷한 스킬을 만든다.

다음과 혼동되지 않아야 한다 → {{조원들의 스킬 이름·용도}}

참고 2: 사내에 「MBC 문서 작성 (HWPX, PPTX, 발표용HTML)」 에이전트가 이미 있다.

출처를 입력하면 회사 표준 서식 문서로 만들어 주는 도구다.

내 스킬이 그것과 겹치지 않도록 경계 조건에 반영해줘.

출력 형식:

- 원칙별 진단 (충족/미흡 + 근거)
- 개선된 description 전문 (1024자 이내)
- 잘못 트리거될 위험이 있는 요청 예시 3개
- 트리거되어야 하는데 놓칠 위험이 있는 요청 예시 3개

P4. 적대적 리뷰

너는 이 스킬을 비판적으로 검토하는 리뷰어다. 작성자 입장이 아니라
"이 스킬이 실패할 방법을 찾는" 입장에서 검토해줘.

{{SKILL.md 전문 붙여넣기}}

다음 관점으로 지적해줘.

1. 모호성 - 사람마다 다르게 해석할 수 있는 지시문
2. 누락된 예외 - 데이터가 없거나 입력이 불충분할 때의 처리
3. 과잉 제약 - 재사용을 방해할 만큼 세세한 지정
4. 당연한 내용 - 모델이 이미 아는 상식을 반복해 신호를 희석하는 부분
5. 범위 과다 - 하나의 스킬이 여러 작업을 하려 하는가
6. 검증 불가 - 규칙을 지켰는지 확인할 방법이 없는 규칙
7. 지어내기 위험 - 원문에 없는 값을 만들어 낼 여지가 남은 곳

각 지적에 심각도(높음/중간/낮음)와 구체적 수정 문구를 제시하고,

마지막에 "함정(Pitfalls) 섹션에 추가할 항목" 목록을 만들어줘.

P5. 개선 루프

스킬을 {{N}}회 실행한 결과다.

- 스킬: {{이름}}
- 실행 기록: {{각 실행의 요청 문구 / 호출 여부 / 결과물 문제점}}
- 측정:
 - 자동 호출: {{되었음 / 안 되었음}}
 - 재생성 횟수: {{ }}회 (목표 3회 이내)
 - 최종 수정률: {{ }}% (기준선 30% 이하 · 문단 단위)
 - 막힌 지점 한 문장: {{ }}

관찰 → 분석 → 개선 → 검증으로 정리해줘.

1. 관찰: 반복되는 실패 패턴
2. 분석: 원인이 (a) description (b) 절차 모호성 (c) 출력 규격이 서술형 (d) 판단 기준 부재 (e) 범위 과다 중 무엇인가
3. 개선: before/after로 (우선순위 순)
4. 검증: 효과를 확인할 테스트 요청 문구 3개

가장 큰 원인 하나에 집중해서, 한 번에 고칠 수 있는 수정만 제안해줘.

P5를 돌리는 법

수정률이 안 떨어질 때 대부분 아무 데나 고칩니다. 그러면 나아져도 무엇 때문인지 모르고 다음에 또 막힙니다. P5는 원인을 다섯 개 중 하나로 좁혀 하나만 고치게 만드는 절차입니다.

쓰는 시점

상황	
2회 돌렸는데 수정률이 비슷하다	P5
체크포인트에서 미도달 판정	P5
스킬이 아예 안 불러 온다	P5 아니고 P3
1회만 돌려봤다	아직 이릅니다 — 한 번 더 돌려십시오

전제 : 실행 기록이 있어야 합니다. P5의 품질은 입력의 품질입니다.

❌ 실행 기록: 잘 안 됐음

✅ 실행 기록:

- 1회 - "8월 지표 정리해줘" → 호출됨 → 표 열 순서가 규격과 다름, 빈 지표가 0으로 들어감
- 2회 - 열 순서를 규격에 명시 후 → 열 순서는 맞음, 금액이 1,200,000 / 120만 / 1.2M 로 섞임

1단계 · 관찰 (2분) — 반복되는 것만

한 번만 일어난 것은 무시하십시오. 두 번 이상 같은 양상이면 그게 패턴입니다.

2단계 · 분석 (3분) — 증상으로 원인을 역추적

이 판별이 P5의 핵심입니다.

증상	원인	고칠 곳
스킬이 안 불러 옴	(a) description	→ P3
불러 왔는데 형식이 매번 다름	(c) 출력 규격이 서술형	규격에 실물 양식 붙여넣기
형식은 같은데 내용 판단이 매번 다름	(d) 판단 기준 부재	처리 단계에 조건 나열
단계를 건너뛰거나 순서가 뒤엎김	(b) 절차 모호성	절차를 번호로 쪼개고 각 단계 산출 명시
어떤 건 잘 되고 어떤 건 전혀 안 됨	(e) 범위 과다	스킬을 쪼개기 — 오늘은 한쪽만

(c)와 (d)를 헷갈리는 경우가 가장 많습니다. 구분은 간단합니다 — 표의 모양이 흔들리면 (c), 표는 같은데 칸에 들어간 값의 판단이 흔들리면 (d) 입니다.

3단계 · 개선 (5분) — 한 번에 하나만

이게 개선 루프의 전부입니다

세 개를 동시에 고치고 나아지면, 무엇이 효과 있었는지 알 수 없습니다. 그러면 다음에 또 세 개를 고치게 되고, SKILL.md는 계속 길어지는데 왜 되는지는 모르는 상태가 됩니다.

도구가 개선안을 여러 개 주면 **우선순위 1번만** 적용하십시오. 나머지는 다음 루프에.

4단계 · 검증 (3분) — 입력을 바꾸지 마십시오

같은 입력으로 재실행합니다. 입력이 바뀌면 나아진 게 수정 때문인지 입력 때문인지 알 수 없습니다. 그리고 수정률을 다시 셉니다. "좀 나아진 것 같다"는 기록이 안 됩니다.

한 루프의 시간 예산

단계	분
관찰 — 로그 정리	2
분석 — P5 실행, 진단 읽기	3
개선 — 1개 적용	5
검증 — 재실행 + 수정률 재측정	3
한 루프	13분

실습 2가 35분이므로 루프는 최대 2회입니다.

3회 돌려도 안 떨어지면 — 멈추십시오

루프를 무한히 돌리지 마십시오. 3회 이내에 안 떨어지면 그건 개선 대상이 아니라 **한계 기록** 대상입니다.

판별 질문은 "같은 규칙을 사람이 받으면 할 수 있는가" 입니다. 사람은 되는데 도구가 안 되면 ② 도구 한계이고, 그게 2차의 자료가 됩니다.

흔한 실패 4가지

실패	왜 문제인가
한 번에 여러 개 고침	무엇이 효과 있었는지 모름

실패	왜 문제인가
재실행 때 입력을 바꿈	나아진 원인을 귀속시킬 수 없음
description 문제인데 본문을 뜯음	호출 자체가 안 되는데 본문을 고쳐도 변화 없음
측정 없이 감으로 판단	기록이 남지 않아 2차에서 못 씀

P6. 조별 상호 비교 (사이클 3)

같은 조 조원들이 같은 구조의 과제로 각자 스킬을 만들었다.

우리 조 구조 축은 "{{조 구조 축 · 예: 양식 규칙을 정의해 문서를 만들거나 제각각인 입력을 표준 포맷으로 밀어 넣음}}}" 이다.

■ 내 SKILL.md:

{{전문 붙여넣기}}

■ 조원의 SKILL.md (또는 산출물):

{{조원 것 붙여넣기 · 여러 명이면 이어서}}

다음 4개 축으로 비교표를 만들어줘.

- ① 출력 규격 - 실물 양식이 들어갔는가 / 서술로 때웠는가. 누가 가장 안 흔들리는가
- ② 판단 기준 - 조건으로 적었는가 / "적절히"가 남아 있는가
- ③ 없을 때·모를 때 - "없음"·"판단 보류" 규칙이 있는가
- ④ 지어내기 - 원문에 없는 값이 출력에 들어갈 여지가 있는가

그리고 다음을 제시해줘.

- 이 조에서 가장 잘된 규격 1개와 그 이유
- 내 스킬에 바로 적용할 개선점 3개 (어느 섹션을 어떻게 고칠지 구체적으로)
- 조 전체가 공유하면 좋을 공통 규격 1개

주의: 개선점은 3개까지만. "더 잘 만들자" 같은 것은 개선점이 아니다.

12분 안에 적용 가능한 것만 제시해줘.

P7. 함정(Pitfalls) 수집

{{업무 이름}}을 AI에게 맡겼을 때 오늘 실제로 발생한 문제들이다.

{{한계 기록 시트에 적은 내용을 나열}}

이것을 SKILL.md의 "함정(Pitfalls)" 섹션으로 정리해줘.

- 각 항목은 "무엇을 하지 말아야 하는가"를 한 줄로
- 왜 문제인지 짧은 근거를 덧붙임
- 일반 상식 수준은 제외하고 우리 업무에 특수한 것만
- 가능하면  올바른 예 /  잘못된 예 대비

신호 밀도를 최우선으로. 분량보다 "이걸 몰랐으면 틀렸을 내용"인지가 중요하다.

그리고 이 중에서 (a) 규칙을 고치면 해결되는 것과
(b) 도구 자체의 한계로 보이는 것을 나눠줘.
(b)는 2차 노코드 한계 판정 자료로 쓸 것이다.

P8. 조 공용 자산화

우리 조는 {{조 번호와 조명}}이고 {{N}}명이 각자 스킬을 만들었다.
조 공용 산출물 목표는 "{{편성안의 공용 산출물}}" 이다.

■ 각자의 스킬: {{이름 · 용도 · 입력 · 출력을 한 줄씩}}

다음은 제안해줘.

1. 겹치는 스킬 - 통합할 수 있는 것과 이유
2. 빠진 영역 - 우리 조 업무 중 아무도 만들지 않은 것
3. 공용 스킬 후보 - 조 전원이 쓸 수 있게 일반화할 것 1~2개
4. 그때 description에 넣을 트리거 키워드 (사람마다 부르는 말이 다를 수 있음)
5. 개인 전용으로 남겨야 할 것 - 일반화하면 오히려 나빠지는 것

주의: "만들기는 쉽게, 배포는 신중하게" 원칙에 따라
중복되거나 품질이 낮은 것을 공용으로 올리지 않도록 걸러줘.

세션 간 과제 (2차 9/14 전까지)

1. 실업무 1회 사용 + 채택 여부 기록 — 사용 / 부분 사용 / 미사용. 안 썼으면 안 썼다고 적으십시오. 그것도 데이터입니다.
2. 개인 도구 대비 비교 판정 (양식 05) — 같은 입력으로 개인 도구와 나란히 놓고 4개 축(출력 형식 · 내용 정확성 · 수정률 · 소요 시간) 판정. 열위면 사유를 4개 중 하나로 분류하십시오 — 설계 문제 / 컨텍스트 부족 / 도구 성능 / 속도. "그냥 별로다"는 쓸 수 없습니다.
3. 한계 기록 이어 쓰기 — 실업무에서 만난 한계를 추가

2차 준비물

- 실데이터 규모 확대분 — 다수 파일 · 다기간 데이터

- 오늘 잘라낸 **2차 확장 후보** 항목
- **5조**는 시스템 접근 권한 상태
- 조에서 고른 **가장 잘된 규격 1개**

한 장 요약

1. Skill은 반복 업무를 처리하는 방법을 미리 적어 둔 업무 매뉴얼이다. 프롬프트는 소비되고, 스킬은 자산으로 쌓인다.
2. **description** 은 사람용 요약이 아니라 모델용 트리거 조건이다. 호출이 안 되면 99% 여기다.
3. 출력 규격에 실물 양식을 붙여넣는 것이 수정률을 가장 크게 낮춘다. 말로 서술한 규격은 미충족이다.
4. "적절히"는 조건 목록으로 바꾼다. "중요한 것"이 아니라 "다음 중 1개 이상 해당".
5. 값이 없으면 **없음** (0으로 채우지 않는다). 확신이 없으면 **판단 보류 + 사유 한 줄**.
6. 불일치는 고치지 말고 표기한다. 조용히 맞춰 놓으면 증거가 사라진다.
7. 입력 1종은 개수가 아니라 종류다. 같은 형식이면 30개도 1종.
8. 함정(Pitfalls) 섹션의 신호 밀도가 가장 높다. 당연한 것은 쓰지 않는다.
9. 오늘 만난 한계가 곧 함정 섹션이고, 2차의 입력이다. "안 됨"은 실패가 아니라 산출물이다.
10. 스킬 이름은 영구 예약이다. <이니셜>--<용도>-0907 .
11. 남의 결과물에서는 지어내기가 보인다. 그래서 조별로 비교한다.
12. 오늘 잘라낸 것은 버린 것이 아니라 2차 확장 후보다.

오늘 안에: 본인 업무로 SKILL.md 하나를 만들고, 실데이터로 두 번 이상 실행하고, 안 된 것을 두 건 이상 기록하고, 조원과 비교해 개선점 하나를 적용하십시오.