

## □ Episode 작성 가이드 (학생·초급자용)

Git에서는 “과거를 고치는 것보다,  
다음 판단을 추가하는 것이 훨씬 안전하고 올바르다.”

### 이 가이드를 읽기 전에 꼭 알아야 할 사실

- Git은 문서 편집기가 아닙니다.
- Git은 시간이 쌓이는 기록 시스템입니다.
- 한 번 커밋된 기록은
  - \***“지우거나 고치는 것”**\*보다
  - \***“다음 기록으로 보완하는 것”**\*이 원칙입니다.

☞ 그래서 Episode는 일기처럼 ‘덧어쓰기’하면 안 됩니다.

### ■. Episode 작성의 기본 원칙 (초급자 필수)

(원칙 1). Episode 번호는 “시간 순서”다

- Episode 번호는 정답 순서 ✕
- Episode 번호는 완성도 순서 ✕
- Episode 번호는 생각이 실제로 진행된 순서 ○  
Episode 002가 틀렸어도  
→ Episode 001을 수정하지 않습니다.

(원칙 2). 이미 커밋한 Episode는 고치지 않는다

- 오타 정도는 수정 가능
- 의미·가설·결론은 수정 금지

왜냐하면:

- Git에서 과거를 고치면
    - 충돌(conflict)
    - 히스토리 꼬임
    - 초급자 멘붕
- 이 발생합니다.

☞ 과거 판단은 “증거”로 남겨두는 게 목적입니다.

(원칙 3). 판단이 바뀌면 “새 Episode”를 연다

아래 상황이면 무조건 새 Episode입니다.

- 가설이 바뀌었다
- 실험 조건이 바뀌었다
- 해석이 달라졌다
- “아, 이건 아닌 것 같다”라고 느꼈다

※ 이때의 정답:

Episode003 — 이전 가설 검증 결과 반영

## 2. Episode 작성 템플릿 (초급자 권장)

Episode 00X — [문제 요약 제목]

- 1) Assumption (가정)
  - 지금 내가 사실이라고 믿고 있는 것은 무엇인가?
- 2) Observation (관측)
  - 실제로 눈으로 확인한 결과는 무엇인가?
- 3) Interpretation (해석)
  - 이 결과를 어떻게 해석할 수 있는가?
  - 가능한 가설은 2~3개까지만
- 4) Next Test (다음 실험)
  - 다음에 바꿀 단 하나의 변수는 무엇인가?

☞ 맨 마지막은 항상 “다음 실험”으로 끝나야 합니다.

## 3. Episode 수정이 필요할 때의 올바른 선택지

상황	하면 안 되는 것	올바른 행동
이전 판단이 틀림	Episode 수정	새 Episode 추가
구조가 마음에 안 듦	번호 재정렬	README에 규칙 추가
설명이 부족함	과거 문장 삭제	다음 Episode에서 보완
기록 순서가 헷갈림	히스토리 되돌리기	그대로 두고 이어서 작성

## 4. 초급자가 가장 많이 하는 실수 TOP 3

- ✖ 실수 1. “깔끔하게 정리하고 싶다”  
→ Git은 깔끔함보다 연속성을 봅니다.
- ✖ 실수 2. “이건 틀렸으니 지워야지”  
→ 틀린 판단은 가장 중요한 학습 데이터입니다.
- ✖ 실수 3. “Episode 번호를 다시 맞춰야 할 것 같다”  
→ 번호는 정리용이 아니라 시간 기록용입니다.

## 5. 한 문장으로 요약

Git 기반 Episode 기록은

‘완성된 답’을 남기는 게 아니라

‘생각이 움직인 흔적’을 남기는 일이다.

--- Tandem Robotics ---