

Module 1 — 프롬프트가 정하는 것은 결과물이 아니라 '완료의 정의'다

이 저장소의 4-Product 시스템은 전부 AI 협업으로 만들어졌습니다. 프롬프트는 매번 그럴듯한 산출물을 냈고 검토도 통과했는데, **비어 있는 것이** 네 번 반복해서 발견됐습니다. 네 건 다 코드 버그가 아니라 프롬프트가 "무엇을 완료로 볼지"를 안 적어서 생긴 일입니다.

모듈	난이도	학습 시간	선행 학습
Module 1	입문	90분	없음

학습 목표 (Learning Objectives)

이 모듈을 마치면 수강자는 다음을 할 수 있습니다.

- ✓ R-C-T-C-F 다섯 요소로 프롬프트를 구성하고, 어느 칸이 부실해서 사고가 났는지 역추적할 수 있다
- ✓ Constraints 를 "항상 지켜라"가 아니라 적용 대상이 명시된 규칙으로 쓸 수 있다
- ✓ Format 에 성공 출력뿐 아니라 미정·해당 없음·실패의 표현 방식까지 규정할 수 있다
- ✓ "만들어라" 대신 관측 가능한 완료 조건으로 Task 를 적을 수 있다
- ✓ AI 산출물의 숫자가 어디서 왔는지 대조할 수 없으면 지워야 한다고 판단할 수 있다

실무 배경 — 어떤 문제에서 출발했나

- **상황:** IT PM/PO 업무 자동화 4-Product 시스템(위키 공간 2개 · Jira 프로젝트 7개 · Next.js 앱 2개)을 AI 협업으로 구축했습니다. 매 단계 프롬프트는 R-C-T-C-F 형태를 갖췄고, 나온 산출물은 읽기에 문제가 없었습니다.
- **그런데 반복해서 나온 증상:** 사고가 전부 같은 모양이었습니다. 실행은 성공했다고 보고되고, 형식은 스펙대로이고, **내용만 없습니다.**

#	무엇이 비었나	언제 드러났나
1	<code>vercel.json</code> 에 넣은 설명 주석이 배포를 거부시킴	배포 시도 시 (즉시)
2	교안 산출물 표 16행 중 10행이 화면에서 사라짐	화면을 사람이 열어 보고
3	위키 페이지 22개가 <code>{{...}}</code> 자리표시자만 담은 채 발행됨	교안 연동을 붙이려다
4	교안의 실측 지표에 대조할 로그가 없음	정식 교안을 쓰면서 (지금)

실무 배경 — 어떤 문제에서 출발했나 (계속)

- **기존 방식의 비용:** 네 건 모두 "검토를 한 번 더 했으면" 막을 수 있는 종류가 아닙니다. ①은 로컬에서 안 보이고, ②③은 형식이 완벽해서 사람 눈을 통과하며, ④는 숫자가 문서 → 데이터 파일 → 슬라이드로 복제되면서 출처가 사라집니다.



이 모듈의 핵심 교훈: 프롬프트에 완료 조건을 안 적으면, 모델이 아니라

사람 눈이 최종 판정자가 됩니다. 그리고 사람 눈은 '형식이 갖춰진 빈 것'을 통과시킵니다.

3.1 프롬프팅 프레임워크 적용 — R-C-T-C-F

다섯 칸의 정의는 어디서나 같습니다. 여기서는 이 저장소에서 그 칸이 부실했을 때 실제로 난 사고를 붙여 둡니다. 일반론이 아니라 사고 기록으로 읽으세요.

요소	이 프로젝트에서 실제로 쓴 값	이 칸이 부실해서 난 사고
Role	Full-Stack Software Engineer & AI Automation Architect	(없음 — 이 칸은 톤만 바꿉니다)
Context	IT PM/PO 자동화 4-Product. Confluence·Jira 를 단일 원본으로 관리	④ 근거 없는 지표가 채워짐 (§4.4)
Task	강의 웹 구축 · 교안 파싱 · 위키 세팅	③ 빈 템플릿 22페이지도 "완료" (§4.3)
Constraints	먹통 실행 · 비밀값 커밋 금지 · 설계 근거를 주석으로 남길 것	① <code>vercel.json</code> 배포 거부 (§4.1)
Format	Markdown 교안 · 링크 목록 · JSON 스키마	② 산출물 10행 소실 (§4.2)

Role 칸이 사고를 안 낸 것이 우연은 아닙니다. Role 은 톤과 어휘 수준을 바꾸고, Constraints·Format 은 산출물이 통과할지 말지를 바꿉니다. 재작업은 거의 전부 뒤쪽 두 칸에서 나옵니다.

3.2 아키텍처 — 완료 판정을 사람 눈에서 코드로 옮기기

[R-C-T-C-F 프롬프트]

|

▼

[AI 산출물] → 사람 검토(형식만 봄) → "완료" → 조용한 실패

|

▼

[완료 판정기] ← 여기가 이 모듈이 추가한 단계

| · 자리표시자 탐지 isPlaceholder()

| · 미정/해당없음 분류 classifyStatus()

| · 출처 표기 source: REPO | LOCAL | NONE

|

|─ 통과 → 완료

|─ 위반 → 무엇이 왜 미완인지 화면에 그대로 노출

핵심은 **"AI 를 더 잘 시키는 것"**이 아닙니다. 시킨 결과가 비었는지를 기계가 **판정**하게 만드는 것입니다. 프롬프트를 아무리 정교하게 써도 판정기가 없으면 같은 사고가 반복됩니다.

§3

3.2 아키텍처 — 완료 판정을 사람 눈에서 코드로 옮기기 (계속)

§4

구현 과정 (Step by Step)

단계	작업	소요	산출물
1	사고 4건을 R-C-T-C-F 칸으로 역추적	30분	§3.1 표
2	Constraints 에 적용 대상 명시	20분	§4.1 개정 문장
3	Format 에 비정상 값 표현 규정	40분	<code>classifyStatus()</code>
4	완료 조건을 관측 가능한 형태로 재작성	30분	<code>isPlaceholder()</code> · <code>isTemplateOnly</code>
5	근거 없는 지표 제거 및 대조 가능한 수치로 교체	40분	이 교안 §5

4.1 사례 A — 착한 제약이 배포를 죽였다 (Constraints)

상시 제약에 이 문장이 있었습니다.

왜 이렇게 했는지 설계 근거를 주석으로 남긴다.

마크다운·파이썬·TypeScript 에서는 옳은 규칙입니다. AI 는 이 규칙을 성실하게 `vercel.json` 에도 적용해 `$comment` 키를 넣었고, 배포가 이렇게 떨어졌습니다.

```
Invalid request: should NOT have additional property `comment`
```

Vercel 은 `vercel.json` 에 `$schema` 외의 커스텀 키를 허용하지 않습니다. 스키마가 닫힌 설정 파일에는 주석을 넣을 자리가 없습니다. 커밋 `d8abad2` 가 키를 지우고 설명을 README 로 옮겼고, 이어서 `1b6d80f` 가 `vercel.json` 자체를 없앴습니다 — 남아 있던 건 `functions` 블록 하나뿐이었는데 App Router 에서는 라우트 파일의 `export const maxDuration` 이 정식 방법이고 4개 중 3개에 이미 들어 있었기 때문입니다.

4.1 사례 A — 착한 제약이 배포를 죽였다 (Constraints) (계속)

개정한 제약 문장:

설계 판단의 근거를 주석으로 남긴다.

단, 스키마가 달한 설정 파일(vercel.json · package.json · tsconfig.json)에는

키를 추가하지 않고 같은 내용을 README 나 커밋 메시지에 적는다.

 **일반화:** 제약에는 적용 대상을 같이 적으세요. "항상 X 하라"는 X 가 불가능한 산출물을 만났을 때 모델을 멈추게 하지 않습니다 — 그냥 X 를 합니다.

4.2 사례 B — 링크가 아닌 행 10개가 조용히 사라졌다 (Format)

교안 §6 산출물 표를 강의 웹 화면에 렌더링하는 작업이었습니다. Format 칸에는 이렇게 적혀 있었습니다 — "산출물을 열 수 있는 링크 목록으로 보여라."

구현은 절대 URL 만 링크로 인식하고, 나머지 행은 **행째로 버렸습니다**. Module 2 교안 두 챕터의 산출물 16행 중 10행이 화면에서 사라졌고, 화면은 "산출물 6개"라고 말했습니다. 그건 거짓말입니다.

실제 교안 §6 이 쓰는 표기는 네 가지가 넘습니다.

4.2 사례 B — 링크가 아닌 행 10개가 조용히 사라졌다 (Format) (계속)

표기	예	처리
절대 URL	<code>https://github.com/ursulax/action/pull/20</code>	그대로
앱 내부 경로	<code>/downloads/module-1-prompting.pdf</code>	그대로
저장소 파일	<code>`docs/4-product-system/templates/case-study.md`</code>	GitHub <code>blob</code> 링크
저장소 디렉토리·그룹	<code>`pm-po-automation-tools/tests/`</code>	GitHub <code>tree</code> 링크
미제작	<code>(제작 예정)</code>	<code>PLANNED</code> 배지
비해당	<code>(해당 없음)</code>	<code>NOT_APPLICABLE</code> 배지

고친 방식이 중요합니다. 스키마(`templates/case-study.md`)는 손대지 않았습니다 — 스키마는 `| 구분 | URL |` 두 열만 요구하고 값의 형식을 제한한 적이 없습니다. 좁았던 것은 파서입니다.

4.2 사례 B — 링크가 아닌 행 10개가 조용히 사라졌다 (Format) (계속)

```
export function classifyStatus(cell: string, href: string | null): DeliverableStatus {  
  if (href) return 'READY';  
  const text = unwrapCode(cell);  
  if (PLANNED.test(text)) return 'PLANNED';  
  if (NOT_APPLICABLE.test(text)) return 'NOT_APPLICABLE';  
  return 'UNLINKED';  
}
```

⚠ **일반화:** Format 에 성공 출력의 모양만 적으면, 모델은 비정상 값을 버립니다.

버리는 쪽이 제일 깔끔해 보이기 때문입니다. "미정·해당 없음·실패를 어떤 모양으로 표현할지"까지 적으세요.

4.3 사례 C — 형식은 완벽하고 내용이 없는 위키 22페이지 (Task)

`setup confluence` 가 `PMPOWIKI` · `PMPOTOOLS` 에 페이지 22개를 발행했습니다. 계층·제목·라벨이 전부 스펙대로였고 실행 로그는 성공이었습니다. 그런데 Module 1~4 페이지의 본문은 템플릿 자리표시자(`{{...}}`)뿐이었습니다.

Task 는 달성됐습니다. "위키에 교안 페이지를 만들어라"는 빈 껍데기로도 달성됩니다.

이걸 모르고 강의 웹이 "위키 우선"으로 교안을 읽었다면, 내용 있는 교안을 자리표시자가 밀어내고 화면이 **오히려** 비었을 겁니다. 그래서 두 가지를 넣었습니다.

```
/** `{{...}}` 는 템플릿 자리표시자입니다 — 아직 안 채워진 칸으로 봅니다. */
export function isPlaceholder(value: string): boolean {
  return /\{\{[\s\S]*?\}\}/.test(value) || value.trim() === '';
}
```

4.3 사례 C — 형식은 완벽하고 내용이 없는 위키 22페이지 (Task) (계속)

그리고 학습 목표도 산출물도 하나도 안 채워졌으면 `isTemplateOnly` 로 표시해 '교안 작성 전'으로 내립니다. 결국 본문 출처의 우선순위를 저장소 우선으로 뒤집었고 (커밋 `ef31b5c`, PR #39), 그 순서를 테스트로 못 박았습니다 — 누군가 "위키가 최신일 테니 먼저 보자"로 되돌리면 `lecture-web/tests/moduleGuide.test.ts` 에서 깨집니다.

 **일반화:** "만들어라"는 관측 가능한 조건이 아닙니다.

"페이지를 만들어라" → "자리표시자가 0개인 페이지를 만들어라"로 바꾸세요.

4.4 사례 D — 대조할 수 없는 숫자 (Context)

이 정식 교안을 쓰기 전, 자리 채우기용 임시 교안([lecture-web/content/module-1.md](#))에는 이런 표가 있었습니다.

프롬프트 전략	1회 통과율	평균 재작업
역할 부여만	41%	2.1
출력 스키마 고정	78%	0.6
둘 다	82%	0.5
← "동일 과제 60건 A/B"		

정식 교안을 쓰면서 그 60건의 실행 로그를 찾았습니다. **저장소 어디에도 없습니다.** 남아 있는 것은 같은 숫자를 담은 실습 데이터 파일 ([module-1-rctcf-template.json](#) 의 [measuredComparison](#))뿐이고, 그건 근거가 아니라 **같은 주장의 사본**입니다.

그래서 이 교안은 그 표를 옮기지 않습니다. 임시 교안은 화면이 비지 않게 하는 자리 채우기였고, 자리 채우기의 숫자가 정식 교안으로 승격 되면 그때부터는 거짓말이 됩니다.

§5 의 지표는 전부 커밋 해시·PR 번호·테스트 출력으로 대조할 수 있는 것만 남겼습니다.

4.4 사례 D — 대조할 수 없는 숫자 (Context) (계속)

⚠ **일반화:** Context 를 안 주면 모델은 그럴듯한 수치를 채웁니다. 그 수치는 문서 → 데이터 파일 → 슬라이드로 복제되면서 출처가 사라집니다. **숫자에는 측정 방법과 날짜를 같이 적고, 못 적으면 숫자를 빼세요.**

4.5 다섯 칸을 '검사 가능한 문장'으로 고쳐 쓰기

칸	검사 불가능한 문장	검사 가능한 문장
Constraints	"설계 근거를 주석으로 남겨라"	"...남겨라. 단 <code>vercel.json</code> 등 스키마가 달한 파일은 제외"
Format	"열 수 있는 링크 목록으로"	"링크·앱 경로·저장소 경로·(제작 예정)·(해당 없음) 5종을 각각 다르게 표시"
Task	"위키에 교안 페이지를 만들어라"	"자리표시자가 0개인 교안 페이지를 만들어라"
Context	"PM 문서 작업이 오래 걸린다"	"PRD 12건 실측, 2026-05~07, <code>module-1-painpoints.csv</code> "

결과 & 검증 (Result & Validation)

지표	Before	After	측정 근거
교안 산출물 표 렌더링	16행 중 6행만 표시	16행 전부 (상태 4종 구분)	<code>tests/caseStudy.test.ts</code> · 04 §4
<code>vercel.json</code> 배포	<code>\$comment</code> 로 거부	파일 제거 후 정상 배포	커밋 <code>d8abad2</code> · <code>1b6d80f</code>
교안 미완성 판정	사람이 눈으로	<code>isPlaceholder()</code> · <code>isTemplateOnly</code>	<code>src/lib/case-study.ts</code>
교안 본문 출처	위키 우선 (빈 템플릿에 밀림)	저장소 우선 + 출처 배지	커밋 <code>ef31b5c</code> · PR #39
교안 지표의 출처	미상 (60건 A/B)	커밋·PR·테스트 출력으로 대조 가능	이 교안 §4.4
<code>lecture-web</code> 테스트	110건	122건	<code>npm test</code>

배운 것 / 다시 한다면 바꿀 것

-  **잘한 것**: 완료 판정을 사람 눈에서 코드로 옮긴 것. 같은 사고가 다시 나면 화면이 먼저 말합니다.
-  **잘한 것**: 스키마가 아니라 파서를 고친 것. 계약([templates/case-study.md](#))을 흔들지 않고 해결했습니다.
-  **다시 한다면**: 제약을 "항상 지켜라"로 쓰지 않겠습니다. 적용 대상이 없는 제약은 적용 불가능한 곳에서 조용히 사고를 냅니다.
-  **다시 한다면**: 임시 교안에 숫자를 넣지 않겠습니다. 자리 채우기라고 적어 뒀도 숫자는 혼자 걸어 나갑니다 — 실습 데이터 파일에 이미 복제돼 있었습니다.

산출물 링크 (Deliverables)

구분	URL
교안 PDF/슬라이드	/downloads/module-1-prompting.pdf
슬라이드 (HTML)	/downloads/module-1-prompting.html
실습용 데이터 세트	/downloads/module-1-painpoints.csv
R-C-T-C-F 프롬프트 골격	/downloads/module-1-rctcf-template.json
시뮬레이터 실행	(해당 없음)
GitHub 저장소	https://github.com/ursulax/action
완료 판정기 구현	lecture-web/src/lib/case-study.ts
파서 계약 테스트	lecture-web/tests/caseStudy.test.ts
교안 스키마	docs/4-product-system/templates/case-study.md
관련 PR	https://github.com/ursulax/action/pull/39

§6

산출물 링크 (**Deliverables**) (계속)

실습 과제 (Action Item)

1. 본인이 최근 **AI** 로 만든 산출물 하나를 골라, 아래 세 가지를 실제로 세어 보세요. - 자리표시자(`{{}}` , `TODO` , `TBD`)가 몇 개 남아 있는가
 - 표·목록의 행 중 화면·다음 단계로 넘어가지 **못한** 행이 몇 개인가 - 문서에 적힌 숫자 중 출처를 댈 수 있는 것이 몇 개인가

1. 그때 쓴 프롬프트의 다섯 칸을 **\$4.5** 형식으로 고쳐 쓰세요. Constraints 에는 적용 대상을, Format 에는 비정상 값의 표현을 반드시 넣습니다. `` `[Constraints] ... 단, 스키마가 달한 파일에는 적용하지 않는다. [Format] 값이 없으면 행을 버리지 말고 (미정) / (해당 없음)` 으로 표기한다. ``

1. 완료 판정기를 한 개 만드세요. 함수 하나면 충분합니다 — 자리표시자를 세거나, 필수 칸이 빈 행을 찾아내는 정도로 시작합니다.

1. 개정 전/후 프롬프트로 같은 과제를 한 번씩 돌리고, **1번의 세 숫자를 다시 세세요.**

제출물: 개정 전/후 프롬프트 + 완료 판정기 함수 1개 + 세 숫자 비교표 **자가 점검 기준:**

실습 과제 (Action Item) (계속)

- ✓ 모든 Constraints 에 적용 대상이 적혀 있는가
- ✓ Format 이 비정상 값(미정·해당 없음·실패)의 표현을 규정하는가
- ✓ 완료 여부를 사람 눈이 아니라 코드가 판정하는가
- ✓ 교안·문서에 남은 숫자가 전부 출처를 댈 수 있는가

Q&A / 자주 막히는 지점

증상	원인	해결
답이 매번 다른 형식으로 나온다	Format 칸이 비었거나 서술형이다	JSON 스키마·표 헤더를 그대로 붙여 준다
그렇듯한데 그대로 쓸 수가 없다	Context 가 일반적이다	도메인·유저·막힌 지점을 한 문장씩, 숫자에는 출처를 함께
실행은 성공인데 결과가 비어 있다	완료 조건이 "만들어라" 수준이다	자리표시자 0개 같은 관측 가능한 조건으로
표의 일부 행이 화면에서 사라진다	파서가 비정상 값을 버린다	상태를 달아 전부 표시. 행을 버리면 개수가 거짓말을 한다
설정 파일에 주석을 넣었더니 배포가 거부된다	스키마가 달한 JSON 이다	주석을 README·커밋 메시지로. 제약에 적용 대상을 명시
문서의 숫자를 아무도 설명하지 못한다	근거 없이 생성된 값이 복제됐다	측정 방법·날짜를 못 적으면 숫자를 지운다
AI 가 제약을 어긴 걸 나중에 안다	검사 주체가 사람 눈이다	제약마다 판정 함수를 하나씩 붙인다