

# 하네스 엔지니어링 내 입맛에 맞게 AI가 일하게 만들기

경기북과학고등학교 교사 홍창욱

# 이론은 공유드린 입문 강의 자료 2, 3차시를 확인해주세요

3차시까지

하네스란 무엇인가  
프롬프트 → 컨텍스트 → 하네스  
그리고 6가지 축



오늘

창욱쌤의 하네스,  
실제로 돌아가는 걸  
통째로 열어보기

2, 3차시에서 우리는 하네스가 무엇인지 배웠습니다. 질문을 다듬는  
프롬프트, 배경을 주는 컨텍스트를 지나, 환경 자체를 설계하는  
하네스까지. 구조, 맥락, 계획, 실행, 검증, 개선이라는 여섯 개의  
축도요.

오늘은 그 이론이 제 컴퓨터에서 실제로 어떤 모습으로 살아  
움직이는지 보여드립니다. 개념 설명이 아니라 실물 구경입니다.

“남의 하네스 실물이 내 하네스의 가장 좋은 출발점”



입문 강의 자료 (1~3차시) ·  
도름스에 공유해 뒀어요

[dorms.school](https://dorms.school) ↗

# 하네스의 핵심은 '내 취향, 내 마음대로'

오픈소스 스킴이나 플러그인 같은 게 사실 대단한 건 없습니다. 처음엔 잘 만들어진 것(oh-my-claudecode, grill-me 같은)을 가져다 쓰며 시작하지만, 쓰다 보면 필요 없어지고 결국 내 입맛에 맞게 직접 구축하게 됩니다.

정답은 없습니다. 그냥 원하는 걸 만들기 시작하세요. 그러다 AI가 내 취향에 안 맞게 일한다 싶을 때, 그 자리를 하나씩 건드리는 겁니다. 경험적으로 개선해 나가는 것, 그게 하네스의 전부입니다.

“도구도 앱도, 모든 것이 개인화되는 시대”

01

# 어디를 건드느냐

하네스를 이루는 요소들, 그리고 각각을 언제  
어떻게 건드리는지

# ~/.claude 폴더 안, 건드는 곳 아홉 곳 소개

역할이 셋으로 갈립니다. 항상 읽는 것, 필요할 때만 펼치는 것, 알아서 발동하는 것.

|                         |                                                                                                                       |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------|
| <b>CLAUDE.md</b>        | 항상 읽음 · 얇은 목차. 핵심은 '이 상황이면 이걸 읽으세요(룰, 스킬 등)'라는 길 안내. 예: 대화는 한국어로, 긴 줄표 금지, 위험 작업 전 백업.                                |
| <b>rules/ (규칙 32개)</b>  | 항상 읽음 · 목차가 가리키는 본문. 예: 배포는 푸시로 끝이 아니라 라이브 확인까지 / UI 한 곳 고치면 같은 패턴 전부 스왑.                                             |
| <b>skills/ (스킬 17개)</b> | 필요할 때만 · 전문 매뉴얼. 예: PDF 통째로 마크다운 변환(읽기는 Codex가 분담), 발표자료 만들기(지금 이 텍스트), '저장소에 넣어' 하면 구글 드라이브 정리.                      |
| <b>hooks/ (훅 8개)</b>    | 자동 발동 · 예: '똑딱'을 알아듣는 귀, 작업 중 '++'로 할 일 끼워넣기, 위험 명령 직전 자동 백업. 잊으면 안 되는 건 여기로.                                         |
| <b>agents/ 역할 14개</b>   | 만드는 AI와 검사하는 AI를 분리해 역할을 정의한다. 검증봇, 보안봇 등을 구축함. 예: 완성된 결과물을 검증봇과 보안봇이 따로 뜯어본다.                                        |
| <b>memory/</b>          | AI가 세션을 넘어 기억하는 사실들. 예: 프로젝트마다 어느 서버, 어느 계정인지 같은 함정 메모. 낡은 기억은 지우고 고친다.                                               |
| <b>settings.json</b>    | 권한, 모델, 훅 연결 스위치. 예: 어떤 순간에 어떤 훅이 돌지 연결해 두는 곳.                                                                        |
| <b>harness-map.html</b> | 전체를 한눈에 보는 지도. 예: 다음 장에 나오는 지도 캡처. 하네스를 바꾸면 같이 갱신한다.                                                                  |
| <b>MCP (외부 도구 4개)</b>   | AI에 쏘는 외부 도구 연결선. 실제 쓰는 것: Playwright(브라우저 조작·화면 검사), supabase(서비스 데이터베이스), korean-law(법령 검색), context7(최신 라이브러리 문서). |

시행 횟수 Top

| Task                  | Count | Percentage |
|-----------------------|-------|------------|
| keyword-extract       | 312   | 41.6%      |
| agent:general-purpose | 137   | 18.3%      |
| parallel-session      | 135   | 18.0%      |
| evaluate              | 128   | 17.0%      |
| security-bug          | 88    | 11.7%      |
| agent:Explore         | 69    | 9.2%       |
| verification-bug      | 46    | 6.1%       |
| humanize-pipeline     | 44    | 5.9%       |
| deferred-task         | 36    | 4.8%       |
| agent:fork            | 15    | 2.0%       |
| email-automation      | 9     | 1.2%       |
| best-practices        | 7     | 0.9%       |
| codes                 | 7     | 0.9%       |
| agent:Plan            | 6     | 0.8%       |

02

# 실전 장치들

제 하네스에서 매일 일하는 것들을 사례로

# "똑딱" 한마디면, 자고 일어나도 돼 있다

제가 제일 아끼는 장치입니다. "똑딱"이라고 말하면 AI가 자율 모드로 들어갑니다. 요청을 목록으로 만들고, 만들고, 검증하고, 배포까지 멈추지 않고 완주합니다(배포는 Vercel·GitHub 명령줄 도구를 직접 다룹니다). 중간에 "할까요?" 하고 묻느라 서지 않습니다.

대신 안전 바닥은 자동 장치가 지킵니다. 위험한 작업 직전엔 자동으로 복원 지점을 남기고, 되돌릴 수 없는 비밀 정보 유출만은 기계가 차단합니다. 지난 넉 달 제 세션의 절반 가까이가 똑딱 모드였습니다.

**참고** 지금 보시는 이 슬라이드도 '로컬똑딱' 한마디로 초안이 만들어졌습니다. PPT의 경우 초안을 개발서버로 띄우고, 제가 텍스트를 직접 편집하면 바로 반영되게 하는건 스킬로 발동됩니다.



## 만든 AI와 검사하는 AI를 나눈다

3차시 강의자료의 Generator와 Evaluator를 실제로 굴리는 모습입니다. 기능 하나를 구현할 때마다 처음부터 끝까지 실제로 눌러보는 검사를 시키고, 코드 검사와 화면 검사를 따로 둡니다. 코드가 통과해도 화면에서는 버튼 상자보다 글자가 커서 빠져나올 수 있으니깐요.(AI가 코드만 작성하면 시각적으로 확인하지 않기 때문에 이러한 실수를 합니다)

화면 검사는 AI가 브라우저를 직접 열어 스스로 확인합니다(Playwright MCP). 그리고 검사를 빼먹은 채 "다 됐다"고 보고하려 하면, 마무리 직전에 흑이 붙잡아 세웁니다.

참고 AI는 화면을 눈으로 못 봅니다. 그래서 넘침과 겹침을 숫자로 재계 시킵니다(3차시 사례 ③).

## "독창적으로!"는 통하지 않는다

AI는 안전한 선택만 하는 확률 기계라, 예술적 감각을 기대하면 실망합니다. "AI 티 안나게 독창적으로 디자인해"는 최악의 지시입니다. 가장 좋은 방법은 레퍼런스를 주는 겁니다.

북각스토어를 만들 땐 앱스토어 화면 캡처를 주고 감상평부터 쓰게 한 뒤 그걸 토대로 코딩하게 했습니다(모방). 다만 여기엔 한계가 있었습니다. 모방은 결국 화면을 본 다음 흉내 내는 것이니까요. 그래서 AI가 가장 잘 알아듣는 언어인 코드로 레퍼런스를 주기 시작했고, 그 첫 결과물이 북광원입니다. 유지할 것(구조, 색, 간격, 애니메이션)과 바꿀 것(데이터 연결)을 가이드라인으로 갈라서요.

참고 레퍼런스 코드는 21st.dev, shaders.paper.design 같은 곳에서 가져옵니다. 이 슬라이드의 디자인도 도름스 매거진을 레퍼런스로 이식한 것입니다.

# AI 티 나는 문구는 자동으로 걸러낸다

사용자에게 보이는 안내 문구를 AI가 쓰면 티가 납니다. 긴 줄표, 번역투, 기계적인 나열. 그래서 사람이 쓴 것처럼 다듬는 스킬(공개 스킬 im-not-ai를 가져와 제 입맛대로 고쳐 쓰는 중)을 두고, 화면에 보이는 문구를 쓰고 나면 반드시 한 번 발동하도록 훅으로 걸어뒀습니다.

다듬는 동안 기다리지 않게 병렬로 돌립니다. 문구 하나 고치자고 작업 전체가 늦어지면 안 되니까요.

참고 긴 줄표 금지는 '사용자에게 보이는 한글'에만 적용합니다. 내부 문서는 예외. 규칙에도 적용 범위가 필요합니다.

# 필수 하네스 엔지니어링 추천!

## 같은 실수를 반복하지 않도록 점진적 개선: 분리수거

OpenAI 엔지니어 Ryan Lopopolo가 공식 블로그 '하네스 엔지니어링'(2026년 2월)에서 밝힌 이야기입니다. 팀이 매주 금요일을 통째로 AI가 남긴 잔해 청소에 쓰다가, 이 청소를 '가비지 컬렉션'이라 부르며 아예 상시 자동 프로세스로 만들었다고 합니다. 기술 부채는 고금리 대출 같아서 조금씩 계속 갚는 게 낫다고요.

저는 이것 '분리수거'라고 부르고, 아예 스킬로 만들어 뒀습니다. "분리수거하자" 한마디면 AI가 세션의 모든 대화 기록을 처음부터 다시 읽으면서, 제가 같은 말을 반복한 곳과 떼어야 할 규칙이 안 뜬 곳을 찾아 하네스 개선안을 내놓습니다. 더하기만 하면 살찌니 버릴 것과 합칠 것도 매번 하나 이상 고르고, 끝나면 하네스 지도까지 갱신합니다. 그리고 일주일이면 하네스가 먼저 분리수거를 제안해 옵니다.

03

# 교사의 하네스

그래서 우리는 무엇을 함께 고민해야 하나

## 교사의 하네스에는 교육이 담겨야 한다

하네스는 결국 개인화입니다. 그렇다면 교사의 하네스는 교사의 특성과 교육적 맥락을 반영하는 게 바람직하겠지요. 제 하네스에도 그런 규칙들이 자라고 있습니다. 학생 개인정보가 담긴 화면은 저장소에 올리지 않기, 학생과 학부모가 읽을 문구는 개발 용어 없이 쉬운 말로 쓰기, 기록의 맥락을 지키기.

이건 개발자의 하네스에는 없는 층입니다. 교실을 아는 사람만 만들 수 있는 규칙들입니다.

참고 여러분의 하네스에는 여러분의 교실이 담기게 됩니다.

# 교사에게, 학교에 적합한 하네스란 무엇인가

유튜브에도 어디에도 바이브코딩 강좌는 이미 넘쳐납니다. 그런데 이 질문만은 어디에서도 대신 고민해 주지 않습니다. 교실을 아는 우리만 해볼 수 있는 고민입니다. 저는 이 고민이야말로 우리가 도름스에서 함께 활동하고, 이렇게 모여 연수를 하는 본 목적이라고 생각합니다.

“도구 사용법은 배우면 됩니다. 이 질문은 우리 몫입니다.”

# 넘어지는 자리에만 기둥을 박으세요

3차시 로드맵 그대로입니다. 첫 주에는 빈 CLAUDE.md에 프로젝트 이름과 기술 스택 한 줄. AI가 틀리면 그때 한 줄 추가. 세 번 틀린 패턴은 규칙으로, 세 번 반복한 작업은 스킬로, 위험한 명령은 흑으로.

그리고 일주일 뒤, 첫 분리수거를 해보세요. 처음부터 완벽하려 하지 않는 것이 결국 가장 빠른 길입니다.

참고 어디부터 건드릴지 모르겠다면 3차시의 harness-selfcheck.md 진단부터 시작하세요.



# 여러분의 하네스가 궁금합니다

하네스에 정답은 없습니다. 그래서 서로의 하네스가 서로에게 가장 좋은 레퍼런스가 됩니다. 여러분이 만든 규칙 하나, 혹은 하나를 도름스에 공유해 주세요. 그게 다음 선생님의 출발점이 됩니다.



도름스 · 모든 링크

[dorms.school](https://dorms.school) /