바이브코딩 케이스 스터디 #6 · 개념편

AI를 쓰는 사람, 부리는 사람

카파시의 강연으로 본 세 단계
프롬프트 → 에이전트 → 그래프

출처: AI 사업일지 유튜브(2026.10.12) · 카파시 TreeHacks 2023, Stanford CS25 V2, Dwarkesh Patel 팟캐스트(2025.10), 「LLM Wiki」(2026.04)

결론부터

무엇이 남느냐로 갈리는 세 단계

1단계

프롬프트

질문 하나에 답 하나. 대화가 끝나면 끝

남는 것: 답 하나

2단계

에이전트

도구를 써서 일을 끝까지 해낸다

남는 것: 끝낸 일 하나

3단계

그래프

끝낸 일이 쌓여 다음 일의 출발점이 된다

남는 것: 계속 커지는 자산

쓰는 사람은 매번 처음부터 묻고, 부리는 사람은 어제 끝낸 일 위에서 시작한다

출처: AI 사업일지 영상 설명·챕터 · '그래프'는 채널이 카파시의 발언과 글을 근거로 내린 해석 · 하단 문장은 강의용 요약

1단계 · 프롬프트

말이 곧 코드가 됐다

카파시, 2023년 1월

“가장 핫한 새 프로그래밍 언어는 영어다”

원문: The hottest new programming language is English

할 수 있는 것

문법 없이 원하는 것을 말로 설명하면 코드와 글과 답이 나온다

한계

답 하나를 받으면 끝. 실행도, 확인도, 다음 일로 잇는 것도 사람 몫

출처: 카파시의 X 게시물(2023.01), Quote Investigator 확인 · 영상은 같은 해 스탠퍼드 강연(TreeHacks 2023, CS25 V2)을 근거로 설명

2단계 · 에이전트

말에 손발이 달렸다

사람 **목표를 준다** → 생각 **다음 할 일 결정** → 행동 **도구 실행** → 관찰 **결과 확인** → 반복 **끝날 때까지**

우리가 이미 쓰는 것

터미널의 claude, codex, agy. 파일을 읽고 명령을 실행하고 결과를 보고 다시 고친다

카파시가 그리는 목표

사람 대신 고용하고 싶을 만큼 믿고 일을 맡길 수 있는 존재

출처: AI 사업일지 영상 챕터 · 카파시, Dwarkesh Patel 팟캐스트(2025.10) 및 X 게시물 · 루프 도식과 '우리가 이미 쓰는 것'은 강의용 설명

에이전트의 빈칸

일은 끝내도, 배운 것은 남기지 않는다

“지속적인 학습이 없다. 무언가를 말해 줘도 기억할 거라 기대할 수 없다”

카파시, Dwarkesh Patel 팟캐스트(2025.10) · 원문: They don’t have continual learning. You can’t just tell them something and they’ll remember it.

‘에이전트의 10년’

올해가 에이전트의 해라기보다, 제대로 되려면 10년쯤 걸릴 거라고 봤다

데모와 제품 사이

테슬라 자율주행 경험에서 나온 말. 데모는 쉽고 제품은 어렵다

출처: Simon Willison 블로그(2025.10.18), The New Stack의 팟캐스트 정리 · 인용은 강의용 번역

3단계 · 그래프

기억은 모델이 아니라 파일에 쌓는다

1일차
일을 끝낸다
결과와 배운 점을 문서로 남긴다

→

2일차
그 문서에서 출발
AI가 어제의 기록을 읽고 이어서 한다

→

N일차
서로 연결된 지식
일이 쌓일수록 다음 일이 빨라진다

왜 '그래프'인가
끝낸 일들이 서로 링크로 이어진 지식망이 되기 때문

누가 붙인 이름인가
카파시의 용어가 아니라, 그의 발언과 글을 근거로 채널이 붙인 해석

출처: AI 사업일지 영상 설명·챕터 「그래프: 끝낸 일이 쌓이는 곳」 · 일차별 도식과 '왜 그래프인가' 카드는 강의용 설명

근거 · 카파시의 「LLM Wiki」 (2026.04)

AI가 직접 관리하는 나만의 위키

raw/

원본 자료

기사, 논문, 데이터. AI는 읽기만 하고 절대 고치지 않는다

wiki/

AI가 쓰는 위키

개념·요약·비교 페이지와 목차, 변경 기록을 AI가 쓰고 고친다

CLAUDE.md

규칙서

구조, 이름 규칙, 페이지 양식, 작업 순서를 정한다

작업	하는 일
넣기 (ingest)	새 자료를 넣으면 요약을 쓰고 관련 페이지 여러 곳을 함께 고친다
묻기 (query)	위키를 찾아 답하고, 쓸모 있는 답은 새 페이지로 저장한다
점검 (lint)	모순, 낡은 주장, 고아 페이지, 빠진 개념을 주기적으로 찾는다

출처: Karpathy, 「LLM Wiki」 GitHub gist(2026.04) · 구조 설명은 Starmorph의 해설 글 기준

1인 사업에 옮기면

같은 일, 쓰는 방식과 부리는 방식

일	쓰는 사람 (매번 처음부터)	부리는 사람 (쌓인 것 위에서)
고객 문의	문의마다 새로 답을 받는다	답변을 FAQ 문서로 쌓고 다음 답의 재료로 쓴다
콘텐츠	매번 주제부터 다시 정한다	지난 대본, 반응, 수정 사항을 기록하고 이어 간다
리서치	같은 걸 여러 번 다시 찾는다	출처와 요약을 위키에 남기고 링크로 잇는다
코딩	새 세션마다 맥락을 다시 설명한다	설계 결정과 실패 기록을 파일로 남겨 두고 읽힌다

지난 사례에서도 보였다. Profit AI의 엑셀은 수년간 쌓인 일이 그대로 제품이 된 경우였다

출처: AI 사업일지 영상 챕터 「1인 사업에 옮기면」 · 표의 예시는 강의용으로 구성

이 채널도, 이 강의 자료도

이 강의 덱들도 앞의 덱 위에서 만들어졌다

AI 사업일지

작업 파일과 지시 로그를 남기고, 이전 편을 다음 편의 출처로 링크한다

덱 #1 → #6

같은 디자인, 같은 구성, 같은 '숫자 읽기' 원칙을 이어서 썼다

쌓인 발견

#2·#3에서 찾은 숫자 차이가 #4의 팩트 체크 표가 됐다

새 영상이 올 때마다 처음부터가 아니라 어제의 덱에서 시작했다

출처: AI 사업일지 영상 챕터 「이 채널도 그렇게 합니다」 및 「실전기 #1」 설명 · 덱 사례는 이 강의 자료 제작 과정

내일부터 부리는 사람이 되는 법

나만의 그래프를 시작하는 4가지

01 끝낸 일은 파일로
대화창에 두지 말고 마크다운 문서로 남긴다

02 원본은 따로
자료는 raw 폴더에, AI가 정리한 것은 wiki 폴더에

03 좋은 답은 저장
쓸모 있는 답과 실패 기록은 새 페이지로 남긴다

04 주기적으로 점검
모순과 낡은 내용을 AI에게 찾게 하고 고친다

AI는 기억하지 못한다. 그래서 기억을 설계하는 사람이 AI를 부린다

출처: 카파시 「LLM Wiki」의 넣기·묻기·점검 구조와 AI 사업일지 영상을 강의용 실천 항목으로 정리