

Codex·Cowork Guide

_비스타 김인숙

채팅을 넘어 AI에게 일을 맡기는 방식을 이해하기 위한
코덱스·코워크 복습 가이드

이 자료는 단순 사용법 매뉴얼이 아니라, 강의 내용을 다시 이해하기 위한 복습용 전자책입니다.

이번 강의의 핵심은 “어떤 버튼을 누르느냐”보다, AI를 바라보는 관점이 어떻게 달라지는가에 있었습니다. 그동안 우리는 AI를 질문에 답하는 채팅창으로 많이 사용했습니다. 하지만 Codex, Cowork, Claude Code 같은 도구를 쓰기 시작하면 AI는 단순한 대화 상대가 아니라, 내 자료를 읽고, 내 폴더 안에서 작업하고, 결과물을 만들어내는 작업 파트너가 됩니다. 이 자료는 그 전환을 차근차근 복습할 수 있도록 정리한 자료입니다.

목차

1. 이번 강의의 핵심 질문: 왜 채팅을 넘어가야 할까
2. 대화형 AI와 작업형 AI는 무엇이 다른가
3. 로컬 폴더라는 개념: AI가 일하는 책상
4. 권한과 안전장치: AI에게 일을 맡길 때 먼저 알아야 할 것
5. Codex, Cowork, Claude Code, Hermes의 역할 차이
6. 모델과 토큰: 똑똑한 AI를 아껴 써야 하는 이유
7. 자료를 주는 방식이 결과를 바꾼다
8. 사례 1: 랜덤 회고 질문 HTML 도구
9. 사례 2: 대학원 리포트와 논문 매칭
10. 사례 3: 녹취록 기반 보고서 작성
11. 사례 4: 강점 34개 인터랙티브 도구
12. Markdown과 스킴: 반복 업무를 자산으로 바꾸기
13. 이번 강의에서 보완해서 이해하면 좋은 부분
14. 처음 시작하는 사람을 위한 정리
15. 후기에서 많이 나온 질문으로 다시 정리하기
16. 내 업무에 바로 대입하는 5가지 출발점
17. 초보자를 위한 안전한 첫 실습 순서

1. 이번 강의의 핵심 질문: 왜 채팅을 넘어가야 할까

이번 강의는 아주 단순한 질문에서 출발했습니다. “AI를 계속 채팅창으로만 써도 될까?”라는 질문입니다.

ChatGPT, Claude, Gemini 같은 도구를 처음 접하면 대부분 채팅창에서 시작합니다. 질문을 입력하면 답이 나오고, 마음에 들지 않으면 다시 물어봅니다. 이 방식만으로도 번역, 요약, 문장 다듬기, 아이디어 정리 같은 일은 충분히 할 수 있습니다.

하지만 실제 업무는 채팅창 안에서 끝나지 않습니다. 강의안을 만들려면 과거 강의자료를 봐야 하고, 보고서를 쓰려면 녹취록과 양식과 이전 샘플을 함께 봐야 합니다. 홈페이지를 수정하려면 파일 구조, 이미지, CSS, 기존 문구를 확인해야 합니다. 카드뉴스나 워크북을 만들 때도 내 브랜드 톤, 기존 결과물, 독자 수준, 배포 목적을 함께 고려해야 합니다.

그래서 이번 강의에서 강조한 전환은 다음 한 문장으로 정리할 수 있습니다.

AI에게 답을 받는 단계에서, AI에게 일을 맡기는 단계로 넘어가야 한다.

여기서 “일을 맡긴다”는 말은 단순히 프롬프트를 길게 쓰는 것이 아닙니다. AI가 참고할 자료를 주고, 작업할 공간을 지정하고, 결과물의 형식을 정하며, 어디까지 수정해도 되는지 권한을 정해주는 일입니다. 즉, AI를 잘 쓰는 능력은 이제 질문을 잘 쓰는 능력만이 아니라 **업무를 구조화해서 위임하는 능력**에 가까워집니다.

2. 대화형 AI와 작업형 AI는 무엇이 다른가

이번 강의에서 구분한 중요한 개념은 **대화형 AI**와 **작업형 AI**입니다. 대화형 AI는 우리가 익숙한 채팅 방식입니다. 질문을 입력하면 답변이 나오고, 그 답변을 사람이 복사해서 다른 문서나 도구에 옮깁니다. 이때 AI는 답을 주지만, 실제 업무 환경 안으로 들어와 움직이지는 않습니다.

반면 작업형 AI는 내 컴퓨터나 프로젝트 공간 안으로 들어와 움직입니다. 폴더를 읽고, 파일을 만들고, 필요한 경우 명령을 실행하고, 결과물을 저장합니다. 사용자는 “이 문장을 바꿔줘”라고 말하는 수준을 넘어 “이 폴더 안의 자료를 기준으로 강의안을 만들어줘”, “이 녹취록을 읽고 보고서 양식에 맞춰 초안을 작성해줘”, “이 HTML 도구를 전체화면용으로 수정해줘”처럼 일을 맡깁니다.

구분	대화형 AI	작업형 AI
기본 관계	질문자와 답변자	업무지시자와 작업자
자료 제공	채팅창에 붙여넣거나 파일 업로드	폴더, 프로젝트, 커넥터, 외부 도구 연결
결과 위치	채팅창 안	파일, 문서, 폴더, 코드, 웹페이지
사람의 역할	답변을 읽고 복사·수정	목표 설정, 권한 승인, 결과 검토
잘 맞는 일	짧은 질문, 번역, 아이디어	복잡한 산출물, 반복 업무, 자료 기반 작업

이 차이를 이해하면 왜 강의자가 “저는 AI와 대화하고 싶은 것이 아니라 일을 하고 싶다”고 말했는지 이해할 수 있습니다. 물론 채팅형 AI가 쓸모없다는 뜻은 아닙니다. 간단한 질문, 빠른 번역, 짧은 아이디어 정리에는 여전히 채팅이 편합니다. 다만 업무 산출물을 만들 때는 채팅창만으로는 매번 사람이 복사하고 붙여넣고 정리해야 하므로 한계가 생깁니다.

3. 로컬 폴더라는 개념: AI가 일하는 책상

이번 강의에서 반복해서 등장한 단어 중 하나가 **로컬**입니다. 로컬은 어렵게 들리지만, 쉽게 말하면 “내 컴퓨터 안”이라는 뜻입니다. 로컬 폴더는 내 컴퓨터 안에 있는 특정 폴더입니다.

작업형 AI에게 로컬 폴더를 지정한다는 것은, AI에게 “이 책상 위에 있는 자료를 보고 일해”라고 말하는 것과 비슷합니다. 책상 위에 강의자료, 논문, 녹취록, 보고서 샘플, 이미지가 잘 정리되어 있으면 AI는 훨씬 수월하게 일할 수 있습니다. 반대로 다운로드 폴더처럼 파일명이 엉망이고 자료가 섞여 있으면, AI도 무엇이 중요한 자료인지 파악하기 어렵습니다.

쉬운 비유

폴더는 AI에게 주는 업무 책상입니다. 잘 정리된 폴더는 필요한 자료가 놓인 책상이고, 정리되지 않은 폴더는 서류가 바닥에 흩어진 사무실입니다. AI가 아무리 똑똑해도, 어떤 자료를 기준으로 일해야 하는지 모르면 좋은 결과를 내기 어렵습니다.

예를 들어 창업 강의안을 새로 만들고 싶다면, “창업 강의” 폴더 안에 과거 강의 PPT, 참고 PDF, 고객 질문, 워크북 샘플이 정리되어 있어야 합니다. 그러면 AI에게 “이 폴더 안의 자료를 참고해서 새 강의안을 만들어줘”라고 말할 수 있습니다. 이때 중요한 것은 AI가 단지 일반 지식으로 답하는 것이 아니라, 내가 지정한 자료를 기반으로 작업한다는 점입니다.

이것이 채팅과 작업형 AI의 가장 큰 차이입니다. 채팅에서는 매번 자료를 올리거나 설명해야 하지만, 작업형 AI에서는 폴더 자체가 하나의 맥락이 됩니다.

4. 권한과 안전장치: AI에게 일을 맡길 때 먼저 알아야 할 것

작업형 AI가 강력한 이유는 파일을 읽고 만들고 수정할 수 있기 때문입니다. 그런데 바로 그 이유 때문에 안전장치가 필요합니다. AI에게 폴더 권한을 준다는 것은, 그 폴더 안에서 어떤 작업을 할 수 있는 권한을 주는 것입니다. 읽기만 허용할 수도 있고, 수정까지 허용할 수도 있습니다. 실행 전에 매번 확인하게 할 수도 있고, 묻지 않고 실행하게 할 수도 있습니다.

강의에서는 강사 본인이 빠른 작업을 위해 “항상 허용”을 쓰는 장면도 있었습니다. 하지만 이것은 숙련자가 자기 작업 방식과 위험을 알고 사용하는 방식입니다. 처음 시작하는 사람에게는 권장되지 않습니다. 특히 원본 문서, 고객 자료, 계약서, 보고서, 홈페이지 파일처럼 중요한 자료가 있는 폴더에서는 더 조심해야 합니다.

초보자의 기본 원칙

처음에는 원본 폴더가 아니라 복사본 폴더를 사용하세요. 그리고 AI에게 “계획만 먼저 보여줘. 내가 승인하기 전에는 이동·삭제·덮어쓰기 하지 마”라고 말하는 것이 안전합니다.

작업형 AI를 쓸 때 가장 좋은 시작 문장은 다음과 같습니다.

이 폴더를 기준으로 작업해줘. 먼저 작업 계획만 세워줘. 내가 승인하기 전에는 파일을 이동하거나 삭제하거나 덮어쓰지 마. 결과물은 새 파일로 만들어줘.

이 문장에는 세 가지 안전장치가 들어 있습니다. 첫째, 계획을 먼저 보게 합니다. 둘째, 실행 권한을 바로 주지 않습니다. 셋째, 원본을 덮어쓰지 않고 새 파일을 만들게 합니다. 이 세 가지만 지켜도 초보자가 작업형 AI를 훨씬 안전하게 시작할 수 있습니다.

5. Codex, Cowork, Claude Code, Hermes의 역할 차이

강의에서 여러 도구가 함께 언급되었기 때문에 헷갈릴 수 있습니다. 가장 쉽게 정리하면 다음과 같습니다.

도구	이해하기 쉬운 정의	주로 잘 맞는 일
Claude Cowork	Claude의 작업형 기능을 문서, 리서치, 파일 정리, 보고서 같은 지식업무에 맞춘 모드	강의안, 보고서, 폴더 정리, 자료 요약, 녹취록 정리, HTML 한 장짜리 도구
Claude Code	Claude의 코딩/개발 중심 에이전트	웹사이트 코드 수정, HTML/CSS/JS, 앱 개발, Git, 테스트, 배포 전 검토
OpenAI Codex	OpenAI의 코딩/소프트웨어 개발 에이전트	코드 수정, 웹앱 제작, 파일 수정, Git 기반 작업, 개발 프로젝트
Hermes / 파니니	사용자 맥락을 기억하고 여러 도구를 연결해 장기 운영을 돕는 AI 비서	반복 운영, 일정·문서·메일·노션·드라이브 연결, 장기 프로젝트 관리

여기서 특히 보완해서 이해하면 좋은 부분은 **Cowork와 Claude Code의 차이**입니다. Cowork를 “Claude Code의 쉬운 버전”이라고만 이해하면 조금 부족합니다. 더 정확히는 Cowork는 Claude Code의 에이전트 능력을 비개발 지식업무에 맞춘 모드입니다. 문서, 리서치, 폴더 정리, 보고서 작성처럼 비개발자가 하는 업무에 맞춰진 화면과 흐름을 제공합니다. 반면 Claude Code는 코드, 웹앱, Git, 테스트처럼 개발 작업을 중심으로 설계되어 있습니다.

Codex와 Cowork도 사용감은 비슷하게 느껴질 수 있습니다. 둘 다 폴더를 지정하고, 자료를 읽고, 결과물을 만들 수 있기 때문입니다. 하지만 공식적인 방향은 다릅니다. Codex는 개발과 코드 중심이고, Cowork는 지식업무 산출물 중심으로 이해하는 것이 좋습니다.

6. 모델과 토큰: 똑똑한 AI를 아껴 써야 하는 이유

강의에서는 Haiku, Sonnet, Opus, Fable 같은 모델 이야기도 나왔습니다. 초보자에게는 모델 이름이 낯설 수 있습니다. 모델은 쉽게 말하면 AI의 “엔진” 또는 “뇌”에 해당합니다. 다만 사람의 지능처럼 단순히 초등학생, 대학생, 박사로만 나누기보다, **속도, 비용, 추론 능력, 작업 적합성**의 차이로 이해하는 것이 좋습니다.

가벼운 모델은 빠르고 비용이 적게 듭니다. 복잡한 모델은 더 깊게 생각하고 더 좋은 결과를 낼 수 있지만, 사용량을 많이 소모합니다. 그래서 모든 일을 가장 비싼 모델에게 맡기는 것은 좋은 전략이 아닙니다.

업무 종류	적합한 접근
간단한 번역, 짧은 문장 정리	가벼운 모델 또는 일반 채팅
파일명 정리, 단순 변환	낮은 추론 수준 + 실행 전 확인
강의안 초안, 보고서 초안	중간~상위 모델
브랜드 가이드, 문체 가이드, 스킴 설계	상위 모델로 기준을 만든 뒤 반복은 낮은 모델로
복잡한 연구, 논문 기반 분석, 중요한 검토	상위 모델 + 원문 대조 + 교차검토

강의에서 말한 중요한 전략은 “비싼 모델은 생각하는 일에 쓰고, 반복 실행은 더 낮은 모델이나 다른 도구에 맡긴다”는 것입니다. 예를 들어 내 브랜드 글쓰기 가이드를 만들 때는 고성능 모델을 쓰는 것이 좋습니다. 한 번 잘 만든 가이드는 이후 Sonnet이나 Codex, Cowork에서도 반복 활용할 수 있습니다. 즉 고성능 모델은 완성품을 계속 찍어내는 기계라기보다, 앞으로의 반복 작업 품질을 높여줄 기준을 만드는 데 쓰면 좋습니다.

7. 자료를 주는 방식이 결과를 바꾼다

AI 결과물이 별로일 때 많은 사람이 “AI가 아직 부족하다”고 생각합니다. 물론 AI가 부족한 경우도 있지만, 실제로는 AI에게 준 자료와 기준이 부족한 경우가 많습니다. 이번 강의의 여러 사례는 모두 같은 원리를 보여줍니다.

좋은 결과를 얻으려면 AI에게 최소한 다음 네 가지를 알려줘야 합니다.

1. **목적:** 무엇을 만들고 싶은가
2. **대상:** 누가 사용할 결과물인가
3. **자료:** 어떤 문서와 사례를 참고해야 하는가
4. **기준:** 어떤 형식, 톤, 안전수칙을 지켜야 하는가

예를 들어 “강의안 만들어줘”라고만 하면 결과가 일반적일 수밖에 없습니다. 하지만 “이 폴더 안의 기존 강의자료 3개와 고객 질문을 참고해서, AI 초보자인 1인기업가가 이해할 수 있는 90분 강의안으로 만들어줘. 개념 설명을

먼저 하고, 뒤에는 실습 예시를 넣어줘”라고 하면 훨씬 좋은 결과가 나옵니다.

AI는 아무것도 없는 상태에서 마법처럼 내 취향을 맞추는 존재가 아닙니다. 내가 가진 자료와 기준을 잘 넘겨줄수록, 나에게 맞는 결과물이 나옵니다.

8. 사례 1: 랜덤 회고 질문 HTML 도구

첫 번째 실전 사례는 대학원 종강 파티에서 사용할 랜덤 회고 질문 도구였습니다. 처음에는 PPT로 질문을 띄우려고 했지만, 약 30명이 참여하는 상황에서는 질문을 랜덤으로 뽑는 방식이 더 자연스럽고 재미있을 수 있었습니다. 그래서 HTML 파일 하나로 열 수 있는 도구를 만들었습니다.

이 사례에서 중요한 것은 코딩 지식을 많이 아는 것이 아니라, **상황을 구체적으로 설명했다**는 점입니다. “연세대 심리과학 이노베이션 대학원 종강 파티”, “한 학기를 회고”, “참가자들이 1분 내외로 한마디씩 답변”, “PPT 대신 랜덤 뽑기처럼 만들고 싶다” 같은 정보가 들어갔기 때문에 AI가 어떤 도구를 만들어야 하는지 이해할 수 있었습니다.

HTML은 인터넷 창에서 열리는 문서 형식입니다. 서버에 올리면 웹사이트처럼 여러 사람이 접속할 수 있고, 내 컴퓨터에서 열면 인터넷 없이도 실행되는 로컬 파일이 될 수 있습니다. 강의에서 만든 랜덤 질문 도구는 후자가 가깝습니다. 파일 하나를 열면 브라우저에서 전체화면으로 보여줄 수 있고, 버튼이나 키보드 입력으로 질문을 넘길 수 있습니다.

이 사례의 핵심 교훈

코딩을 몰라도 “어떤 장면에서 어떤 경험을 만들고 싶은지”를 설명하면, 작업형 AI가 작은 도구를 만들어줄 수 있습니다. 중요한 것은 코드 지식보다 사용 장면을 구체적으로 말하는 능력입니다.

9. 사례 2: 대학원 리포트와 논문 매칭

두 번째 사례는 대학원 통계 과제였습니다. 이 과제는 단순히 글을 쓰는 것이 아니라, 수업에서 배운 통계 개념을 이해하고, 그 개념이 실제 논문에서 어떻게 쓰였는지 찾아 설명해야 했습니다. 즉 통계 개념, 과제 가이드, 논문 자료, 설명 능력이 모두 필요했습니다.

이런 작업은 채팅형 AI에게 “논문 추천해줘”라고 묻는 것만으로는 부족합니다. 왜냐하면 내가 가지고 있는 논문 폴더, 수업자료, 과제 조건을 함께 봐야 하기 때문입니다. 작업형 AI에게는 폴더를 지정하고 “이 과제 가이드를 기준으로, 이 폴더 안 논문 중 가장 적합한 것을 찾아줘”라고 말할 수 있습니다.

이 사례에서 중요한 부분은 AI가 추천한 결과를 그대로 믿지 않았다는 점입니다. 여러 AI로 교차 검토하고, 이해가 안 되는 부분은 다시 설명을 요청하고, 최종적으로 자기 언어로 다시 쓰는 과정을 거쳤습니다. 이것은 공부나 전문 보고서 작업에서 매우 중요합니다.

AI는 후보를 좁혀주고 설명을 도와줄 수 있지만, 이해와 최종 판단은 사람의 몫입니다.

특히 논문, 통계, 법률, 의료, 계약처럼 정확성이 중요한 영역에서는 AI가 만든 설명을 그대로 제출하거나 공유하면 안 됩니다. 원문과 대조하고, 숫자와 개념을 확인하고, 필요하면 다른 모델이나 전문가 검토를 거쳐야 합니다.

10. 사례 3: 녹취록 기반 보고서 작성

세 번째 사례는 멘토링이나 컨설팅 후 제출해야 하는 보고서 작성입니다. 기관 업무에서는 양식이 있고, 기존 보고서 샘플이 있고, 참여자 명단과 회기 정보가 있으며, 실제 상담이나 멘토링 녹취록이 있습니다. 사람이 하나씩 쓰면 시간이 많이 걸리지만, 구조가 명확한 반복 업무이기 때문에 AI에게 맡기기 좋습니다.

이때 핵심은 AI에게 충분한 자료를 주는 것입니다. 빈 양식만 주면 AI는 무엇을 어떻게 채워야 하는지 모릅니다. 녹취록만 주면 어떤 문체와 분량으로 써야 하는지 모릅니다. 작년 보고서 샘플만 주면 이번 참여자의 내용이 없습니다. 그래서 다음 자료를 함께 줘야 합니다.

- 이번 회기의 녹취록
- 이전 보고서 샘플
- 빈 보고서 양식
- 참여자 명단과 시간 정보
- 작성 시 주의할 표현이나 기관 제출 기준

이렇게 자료를 주면 AI는 “신입 직원이 작년 보고서를 참고해 이번 보고서를 작성하는 것”과 비슷하게 일할 수 있습니다. 다만 최종 검수는 반드시 사람이 해야 합니다. 이름, 날짜, 회기, 민감한 표현, 과장된 해석, 기관 제출 형식은 사람이 확인해야 합니다.

11. 사례 4: 강점 34개 인터랙티브 도구

네 번째 사례는 갤럽 강점 강의에서 사용할 수 있는 인터랙티브 HTML 도구였습니다. 34개 강점을 버튼으로 보여주고, 클릭하면 설명이 나타나게 하는 도구입니다. 팀 강의에서는 참가자들의 상위 강점을 체크하거나, 특정 강점이 많은지 보여주는 방식으로 활용할 수 있습니다.

이 사례에서 중요한 것은 “AI에게 인터넷에서 알아서 조사해서 넣어달라”고 하지 않았다는 점입니다. 강점 설명은 전문성과 정확성이 중요합니다. 그래서 강의자는 자신이 검증한 자료, 자신이 쓰는 설명, 자신의 해석 기준을 넣어 도구를 만들고 싶어 했습니다.

이것은 전문가가 AI를 활용할 때 매우 중요한 원칙입니다. AI에게 모든 것을 조사하게 하면 빠르기는 하지만, 오류나 환각이 들어갈 수 있습니다. 특히 내가 전문성을 가진 분야라면, 내가 검증한 자료를 먼저 주고 AI에게 그것을 구조화하게 하는 편이 더 안전합니다.

12. Markdown과 스킬: 반복 업무를 자산으로 바꾸기

강의에서는 Markdown과 스킬 이야기도 나왔습니다. Markdown은 텍스트 기반 문서 형식입니다. AI 전용 형식은 아니지만, 제목, 목록, 표 같은 구조를 단순한 텍스트로 표현하기 때문에 AI가 읽기 편한 경우가 많습니다.

PDF나 PPT를 AI가 읽을 수는 있지만, 매번 복잡한 문서를 읽히면 사용량이 많이 들 수 있습니다. 특히 반복적으로 참고할 자료라면 한 번 Markdown으로 바꿔두면 이후 작업이 쉬워집니다. 다만 Markdown은 원본을 대체하는 파일이 아닙니다. 논문, 계약서, 증빙자료, 표와 그림이 중요한 문서는 반드시 원본을 보관해야 합니다.

스킬은 반복 업무를 위한 레시피입니다. 한 번 잘 만든 작업 방식을 다음에도 재사용할 수 있도록 규칙으로 저장해두는 것입니다. 예를 들어 “녹취록을 참가자용 복습자료로 바꾸는 방식”, “강의자료를 워크북으로 바꾸는 방식”, “행사 목적을 랜덤 질문 HTML로 바꾸는 방식”은 모두 스킬이 될 수 있습니다.

Markdown과 스킬의 차이

Markdown은 AI가 읽기 쉬운 자료 형식에 가깝고, 스킬은 AI가 일을 반복해서 수행하기 위한 절차와 기준에 가깝습니다. 자료를 정리하는 것이 Markdown이라면, 일하는 방식을 정리하는 것이 스킬입니다.

13. 이번 강의에서 보완해서 이해하면 좋은 부분

강의는 실시간 시연과 개념 설명이 함께 진행되었기 때문에, 몇 가지 표현은 후속자료에서 더 정확하게 보완해두는 것이 좋습니다. 아래 내용은 참가자가 오해하지 않도록 정리한 부분입니다.

1) Cwork는 Claude Code의 “쉬운 코딩 버전”이 아니다

강의 중에는 쉽게 설명하기 위해 Cwork를 Claude Code의 쉬운 버전처럼 말한 부분이 있었습니다. 하지만 더 정확히는 Cwork는 Claude Code의 에이전트 능력을 문서, 리서치, 파일 정리, 보고서 같은 지식업무에 맞춘 모드입니다. Claude Code는 개발 업무에 더 적합합니다.

2) Codex와 Cwork는 비슷하지만 같은 도구는 아니다

둘 다 폴더를 읽고 결과물을 만들 수 있기 때문에 사용자는 비슷하게 느낄 수 있습니다. 하지만 Codex는 OpenAI의 개발/코딩 중심 에이전트이고, Cwork는 Claude의 지식업무 중심 에이전트로 이해하면 좋습니다. 문서와 자료 중심이면 Cwork, 코드와 웹앱 중심이면 Codex나 Claude Code가 더 자연스럽습니다.

3) “쓸 때마다 돈이 나간다”는 표현은 한도 소모로 이해하기

구독형 서비스에서는 보통 매번 카드 결제가 되는 것이 아니라, 구독 플랜 안의 사용량이나 한도를 소모합니다. 다만 API 키, usage credits, 별도 고성능 모델은 실제 사용량 기반 과금이 될 수 있습니다. 그래서 중요한 것은 “무조건 돈이 나간다”보다 “어떤 모델과 기능이 내 한도와 비용에 영향을 주는지 확인해야 한다”입니다.

4) “항상 허용”은 초보자 기본값이 아니다

숙련자는 실험 폴더에서 빠르게 작업하기 위해 자동 실행을 켤 수 있습니다. 하지만 처음 시작하는 사람은 실행 전 확인이 기본입니다. 특히 삭제, 이동, 덮어쓰기, 게시, 발송, 공유 권한 변경은 반드시 사람이 확인해야 합니다.

5) Markdown은 원본 대체가 아니라 작업용 사본이다

Markdown은 AI가 읽기 편한 형식이지만 원본을 완전히 대체하지는 않습니다. 표, 이미지, 수식, 쪽번호, 서식이 중요한 문서는 원본을 반드시 보관해야 합니다.

14. 처음 시작하는 사람을 위한 정리

이번 강의를 들었다고 해서 바로 모든 도구를 설치하고 자동화를 만들 필요는 없습니다. 가장 먼저 해야 할 일은 작은 실험입니다. 내 컴퓨터에 테스트 폴더를 만들고, 중요하지 않은 복사본 자료를 넣고, AI에게 “계획만 먼저 세워줘”라고 말해보는 것입니다.

처음부터 홈페이지를 만들거나 대량의 파일을 정리하거나 고객 보고서를 자동화하려고 하면 부담이 큼니다. 대신 다음 중 하나를 골라 시작하면 좋습니다.

- 작은 녹취록 하나를 복습자료로 정리하기
- 내 글 3개로 문체 특징 분석하기
- 워크숍용 랜덤 질문 HTML 만들기
- 다운로드 폴더 정리 계획만 받아보기
- 기존 보고서 샘플을 읽고 작성 기준 정리하기

AI를 잘 쓰는 사람은 프롬프트를 화려하게 쓰는 사람이 아니라, 자신의 일을 잘 나누고, 자료를 잘 준비하고, 기준을 분명히 말하는 사람입니다. 작업형 AI 시대에는 “무엇을 물어볼까?”보다 “무엇을 맡길 수 있을까?”, “어떤 자료와 기준을 줘야 할까?”, “어디까지 권한을 줄까?”가 더 중요한 질문이 됩니다.

마지막 정리

Codex, Cowork, Claude Code는 AI를 채팅창 밖으로 데리고 나오는 도구입니다. 이 도구들을 잘 쓰려면 설치법보다 먼저 개념을 이해해야 합니다. 로컬 폴더, 권한, 자료, 모델, 한도, 검수 기준을 이해하면 AI는 단순한 답변자가 아니라 실제 업무를 함께 진행하는 파트너가 될 수 있습니다.

15. 후기에서 많이 나온 질문으로 다시 정리하기

웨бина 후기에 가장 많이 나온 반응은 “실제 쓰는 모습을 보니 이해가 됐다”였습니다. 동시에 많은 분들이 “그래서 나는 어떤 도구를 먼저 써야 하지?”, “내 업무에서는 무엇을 맡기면 되지?”, “내 컴퓨터 자료에 접근하게 해도 안전한가?”를 궁금해했습니다. 이 장은 그 질문에 맞춰 다시 정리한 부분입니다.

후기에서 확인된 핵심 반응

- 사람들은 이론 설명보다 **실제 업무 흐름과 결과물을 보는 장면**에서 가장 많이 이해했습니다.
- “로컬 폴더”, “권한”, “에이전트”, “토큰” 같은 말은 이해됐지만, 혼자 시작하려면 **정리된 기준표**가 필요합니다.
- 다음으로 알고 싶은 것은 도구 자체보다 **내 업무에 무엇을 맡길 수 있는지**입니다.
- 카드뉴스, 홈페이지 카피, 콘텐츠 기획, 녹취록 정리처럼 **실제 산출물 중심 실습**에 대한 요구가 컸습니다.

1) “무엇을 어떤 AI에게 맡겨야 하나요?”

처음에는 도구 이름보다 업무 성격을 먼저 보면 쉽습니다. 내가 하려는 일이 “짧은 질문”인지, “문서와 자료를 읽어야 하는 일”인지, “코드나 웹페이지를 고쳐야 하는 일”인지, “반복 운영을 연결해야 하는 일”인지부터 나눠보세요.

하고 싶은 일	먼저 생각해볼 도구	이유
짧은 질문, 아이디어, 문장 다듬기	ChatGPT · Claude · Gemini	채팅창에서 빠르게 묻고 답을 받기 좋습니다.
녹취록, 자료 묶음, 보고서, 강의자료 정리	Claude Cowork · Claude	문서와 맥락을 읽고 지식업무 산출물로 정리하는 데 적합합니다.
HTML, 웹앱, 코드, Git, 테스트	Codex · Claude Code	파일을 수정하고 코드 실행·검증이 필요한 작업에 적합합니다.
장기 프로젝트, 메일·일정·드라이브·노션 연결	Hermes / AI 비서	여러 도구와 반복 업무를 연결해 운영하는 데 적합합니다.
브랜드 기준, 글쓰기 기준, 업무 지시 기준 만들기	상위 모델의 ChatGPT · Claude	한 번 좋은 기준을 만들어두면 이후 반복 작업의 품질이 올라갑니다.

2) “내 업무에서는 무엇을 자동화할 수 있나요?”

자동화는 거창한 시스템부터 시작하지 않아도 됩니다. 반복해서 설명해야 하는 일, 매번 비슷한 형식으로 정리하는 일, 여러 자료를 읽고 요약해야 하는 일부터 시작하면 됩니다.

강사·컨설턴트

녹취록 요약, 강의 복습자료, Q&A 정리, 워크북 초안, 다음 강의 개선점 정리

콘텐츠 제작자

글 소재 분류, 카드뉴스 초안, 제목 후보, 독자 질문 추출, 발행 전 체크리스트

1인기업·전문가

서비스 안내문, 홈페이지 카피 비교, 고객 질문 정리, 상담 후 보고서 초안

연구·리포트 작성자

논문 후보 정리, 개념 비교표, 참고문헌 요약, 과제 조건과 자료 매칭

3) “안전하게 시작하려면 무엇을 지켜야 하나요?”

작업형 AI는 강력하지만, 파일을 다룬다는 점에서 조심해야 합니다. 초보자는 빠른 실행보다 안전한 실험이 먼저입니다.

초보자 안전 기준 5가지

1. 원본 폴더가 아니라 복사본 폴더에서 시작하세요.
2. 처음 지시는 “계획만 먼저 보여줘”로 시작하세요.
3. 삭제, 이동, 덮어쓰기, 발송, 게시, 공유 권한 변경은 반드시 직접 승인하세요.
4. 고객 개인정보, 계약서, 민감한 보고서는 바로 넣지 말고 익명화하거나 요약본으로 실험하세요.
5. 결과물은 원본을 고치게 하지 말고 새 파일로 만들게 하세요.

16. 내 업무에 바로 대입하는 5가지 출발점

후기에서 “자동화할 수 있는 업무가 여전히 어렵다”는 반응이 있었습니다. 그럴 때는 아래 다섯 가지 중 하나만 골라 시작하면 됩니다.

1 녹취록 하나를 복습자료로 바꾸기

강의나 미팅 녹취록을 넣고, “내용을 시간순으로 정리하고 참가자가 다시 읽을 수 있는 복습자료로 만들어줘”라고 요청합니다.

2 내 글 3개로 문체 기준 만들기

내가 쓴 글 몇 개를 넣고, “공통 문체와 피해야 할 표현, 앞으로의 글쓰기 기준을 정리해줘”라고 요청합니다.

3 카드뉴스 주제 10개 뽑기

내 서비스 설명이나 블로그 글을 넣고, “초보 독자가 궁금해할 카드뉴스 주제 10개와 첫 장 후킹 문장을 뽑아줘”라고 요청합니다.

4 홈페이지 문구 비교하기

현재 홈페이지 문구와 고객 질문을 넣고, “어떤 문장이 어렵고 무엇을 더 쉽게 바꾸면 좋은지 비교해줘”라고 요청합니다.

5 폴더 정리 계획만 받아보기

중요하지 않은 테스트 폴더에서 "파일을 이동하지 말고, 정리 기준과 폴더 구조 계획만 먼저 제안해줘"라고 요청합니다.

17. 초보자를 위한 안전한 첫 실습 순서

이 강의를 듣고 바로 Codex, Cowork, Claude Code, Hermes를 모두 설치하고 연결할 필요는 없습니다. 오히려 첫 실습은 작고 안전해야 합니다.

1. **테스트 폴더 만들기:** 바탕화면이나 문서 폴더에 “AI_실습_복사본” 폴더를 만듭니다.
2. **중요하지 않은 복사본 넣기:** 블로그 글, 짧은 녹취록, 샘플 문서처럼 유출되어도 위험이 낮은 자료만 넣습니다.
3. **목표를 한 문장으로 쓰기:** “이 녹취록을 복습자료로 정리하고 싶다”처럼 결과물을 분명히 합니다.
4. **계획 먼저 요청하기:** “실행하지 말고 계획만 먼저 보여줘”라고 지시합니다.
5. **새 파일로만 만들게 하기:** 원본 수정 없이 새 문서, 새 HTML, 새 요약 파일로 만들게 합니다.
6. **결과를 검토한 뒤 반복하기:** 마음에 들지 않는 부분을 기준으로 다시 요청합니다. 이 반복 과정이 AI와 일하는 감각을 만듭니다.

후기 반영 최종 정리

이번 강의에서 사람들이 가장 크게 얻은 것은 “AI가 답만 주는 도구가 아니라, 내 자료와 폴더를 기준으로 함께 일할 수 있는 도구”라는 감각이었습니다. 다음 단계는 더 많은 도구를 한꺼번에 배우는 것이 아니라, 내 업무 하나를 골라 작게 맡겨보는 것입니다. 복사본 폴더, 계획 먼저, 새 파일 생성이라는 세 가지 안전장치를 지키면 초보자도 작업형 AI를 훨씬 편하게 시작할 수 있습니다.

이 자료는 2026.07.05 “AI 기초 활용 — 코워크와 코덱스” 웨비나 녹취록을 바탕으로, 참가자가 강의 내용을 정확히 복습할 수 있도록 참가자가 강의 내용을 정확히 복습할 수 있도록 전자책형으로 재구성한 배포용 자료입니다. 일부 도구 설명은 OpenAI Codex 및 Anthropic Claude Cowork/Claude Code 공개 문서를 기준으로 보완했습니다.