

01

AI 핵심 용어

생성형 AI · LLM · 에이전트 · 컨텍스트 · 프롬프트 · 코드 · 토큰

AI 활용을 위한 핵심 개념

AI 활용이 가장 큰 장벽은 용어 - 뜻풀이는 AI에게 물으면 되고, 여기서는 그 용어로 AI를 어떻게 쓰는지를 본다

01 기본 개념

- 1 AI
- 2 생성형 AI
- 3 LLM



02 LLM의 활용

- 4 컨텍스트
- 5 에이전트
- 6 프롬프트
- 7 코드



03 그 외에 알아야 할 것

- 8 모델
- 9 컨텍스트 윈도우
- 10 토큰
- 11 마크다운
- 12 할루시네이션

01

기본 개념

AI · 생성형 AI · LLM - 매일 말하는 AI가 정확히 무엇인지

AI - 데이터를 미리 학습해서 판단 · 예측

AI = 사전에 데이터를 학습해서, 그걸 토대로 판단 · 분류 · 예측하는 프로그램

우리가 처음 들은 AI

알파고

학습한 데이터
수많은 바둑 기보



하는 일
다음에 둘 수를 판단한다

메일함 속 AI

스팸메일 필터

학습한 데이터
메일 + 정답지(스팸이다 / 아니다)



하는 일
새 메일이 스팸인지 판단한다

추천 속 AI

유튜브 알고리즘

학습한 데이터
내가 본 영상 · 시청 기록



하는 일
다음에 볼 영상을 추천한다

공통점 - 판단 · 분류 · 예측만 한다. 새로운 걸 만들지는 않는다

생성형 AI - 새로운 결과물을 만드는 AI

생성형 AI = 학습한 데이터로 새로운 결과물을 생성한다

파일 공유함, 명령 4개 실행함 >

외상매출금(10800) 월별 합계표를 엑셀로 만들었습니다. 기 때문에 머리글 색, 글꼴, 열 너비, 천 단위 쉼표 서식은 넣었습니다. 월말 잔액(=전월잔액+차변-대변)과 합계(=3월은 차변 1,987,155,325원, 대변 1,908,724,051원, 월말 첨부하신 2월 파일이 3월만이 아니라 1~12월 전체 표였다고, 3월은 그 안의 한 행입니다. 3월 한 달만 필요하셨다면 (1,599,434,373원)으로 두고 3월 한 행만 남기겠습니다.

글

ChatGPT · Claude · Gemini

글을 학습했고 글만 쓴다

월말 결산 절차 4단계
매달 반복되는 결산, 네 단계로 정리합니다

1. 전표 마감
모든 거래 내역을 기록하고 확정합니다

2. 계정 대사
은행 잔고와 장부 금액을 일치시킵니다

3. 결산 분개
감가상각, 총당금 등 필요한 조정을 합니다

4. 재무제표 작성
최종적인 회계 보고서를 완성합니다

이미지

나노바나나 · GPT-image · Midjourney

이미지를 학습했고 이미지만 만든다

예시 이미지

영상

Veو · Sora · Kling

영상을 학습했고 영상만 만든다

생성형 AI 중에서도 글만 읽고 쓰는 도구가 LLM(Large Language Model)이다.

LLM(Large Language Model) - 이름에 담긴 원리와 한계

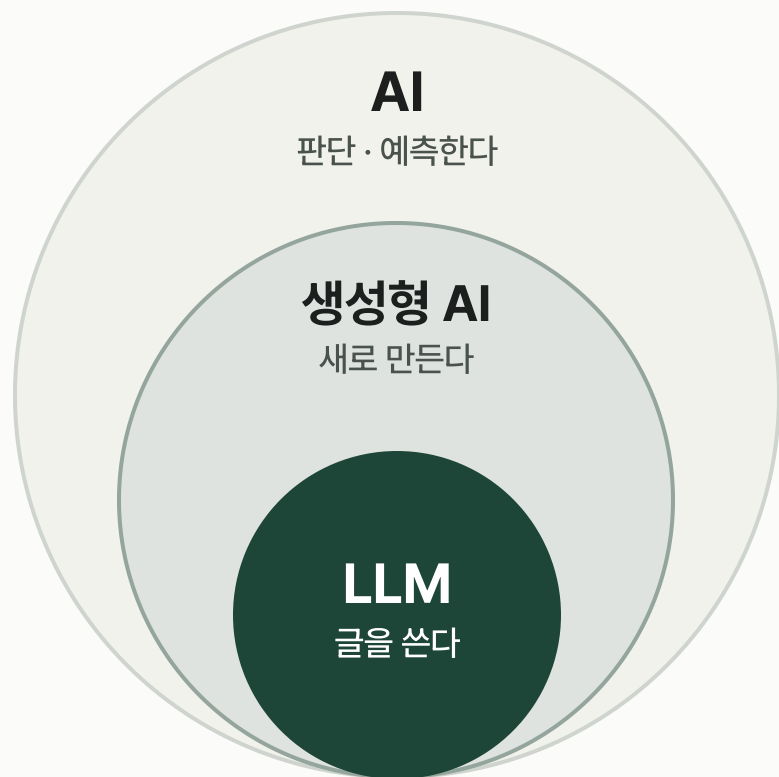
LLM = 글을 읽고 쓰는 생성형 AI

이름	무슨 뜻	그래서 생기는 한계
Large 대규모	방대한(Large) 데이터를 사전 학습했다 웹 문서 · 책 · 뉴스 · 코드	① 학습 이후의 정보는 모른다 최신 개정 · 우리 회사 자료는 알지 못한다
Language 언어	글(Language)을 읽고 쓰는 도구다 입력도 글, 출력도 글	② 글만 쓸 줄 안다 컴퓨터를 직접 조작하지 못한다
Model 모델	확률 모델(Model) - 다음에 올 가장 그럴듯한 단어를 고른다 예) "대손충당금을 ____" → 설정 62% · 계상 25% · 환입 9%	③ 같은 말에도 매번 결과가 다르다 그럴듯한 오답도 나온다

ChatGPT의 GPT도 같은 이야기 - Generative(생성하는) · Pre-trained(사전학습된) · Transformer(구조 이름)

LLM의 3가지 한계를 어떻게 극복할 것인지가 **AI 활용 전략의 핵심**

AI · 생성형 AI · LLM - 한눈에 보기



AI
판단 · 예측한다

알파고 · 스팸 필터 · 유튜브 추천

생성형 AI
새로 만든다

글 · 이미지 · 영상을 만든다

LLM
글만 쓴다

ChatGPT · Claude · Gemini
클로드 코드의 두뇌도 LLM

AI 안에 생성형 AI, 그 안에 LLM - 오늘의 주인공은 글만 쓰는 LLM

02

LLM의 활용

컨텍스트 · 에이전트 · 프롬프트 · 코드 - LLM의 한계 세 가지를 극복하는 법

LLM의 활용 - 곧 한계 세 가지를 극복하는 것

LLM을 잘 활용한다 = 이 세 가지 한계를 극복하는 것 - 02의 개념 네 가지가 하나씩 맞는다

LLM의 한계

① 사전학습

학습 이후 정보 · 우리 회사 자료는 모른다



극복 방법

모르는 걸 알려 준다

개념

④ 컨텍스트

② 글만 쓴다

컴퓨터를 직접 움직이지 못한다



글(코드)을 써서 컴퓨터가 실행하게 한다

⑤ 에이전트

③ 확률 모델

같은 요청에도 매번 결과가 다르다(흔들림)



원하는 걸 분명히 적는다
늘 같아야 하는 일은 코드로 고정한다

⑥ 프롬프트

⑦ 코드

③ 확률 모델은 두 단계로 - **프롬프트**로 흔들림을 줄이고, 매번 같아야 하는 일은 **코드**로 고정한다

컨텍스트 - AI에게 알려 주는 배경 정보(맥락)

컨텍스트 = AI가 학습하지 못한 정보를 알려 주는 것 - 전달 방법이 이렇게 넓어져 왔다



매번 프롬프트로 입력하거나 파일로 첨부하는 게 아니라 - 필요할 때 꺼내 쓰고, 내 컴퓨터에서 찾아 읽어 컨텍스트로 쓰게 한다



멀티모달 - 스크린샷 · PDF를 첨부해도 읽는 이유



"LLM은 글만 읽을 수 있다면서요?"

챗봇에 PDF 파일이나 스크린샷을 첨부해도 읽어 내는 이유

원리 · 멀티모달

조각으로 나눠 숫자로 바꿔 읽는다

- 스크린샷 · 사진

이미지를 작은 조각(시각 토큰)으로 나눠 숫자로 바꿔 읽는다

- PDF 파일

글자 문서는 텍스트를 바로 추출하고, 스캔본은 이미지로 변환

실무 활용 팁

숫자까지 정확하다는 보장은 없다

- 비용 절약

이미지를 인식하면 일반 텍스트보다 비용이 훨씬 많이 든다

- 숫자 왜곡 방지

해상도가 낮으면 숫자가 뭉개져 부정확

가장 효율적인 형식은 텍스트 - 이미지 · 스캔본은 차선택.

에이전트 - '답장'만 하던 AI에서 내가 할 '일 자체'를 해주는 AI로

예전에는 답장만 해 주던 AI가 어느 순간부터 내가 해야 되는 일 자체를 수행(엑셀·워드 파일 생성 등)하도록 진화한 것 = AI 에이전트

프롬프트 증감 분석 보고서 만들어 줘

예전의 AI (챗봇)

질문에 글로 답장만 했다

AI

매출채권 분석 방법은 다음과 같습니다. 전기 대비 15% 증가하였으며...

답장·설명만 해 줄 뿐, 실제 파일 작업은 내 몫

지금의 AI (에이전트)

내가 해야 할 일 자체를 해준다

03_산출

XLSX

증감분석_집계표.xlsx

엑셀

DOCX

증감분석_보고서.docx

워드

PDF

증감분석_제출본.pdf

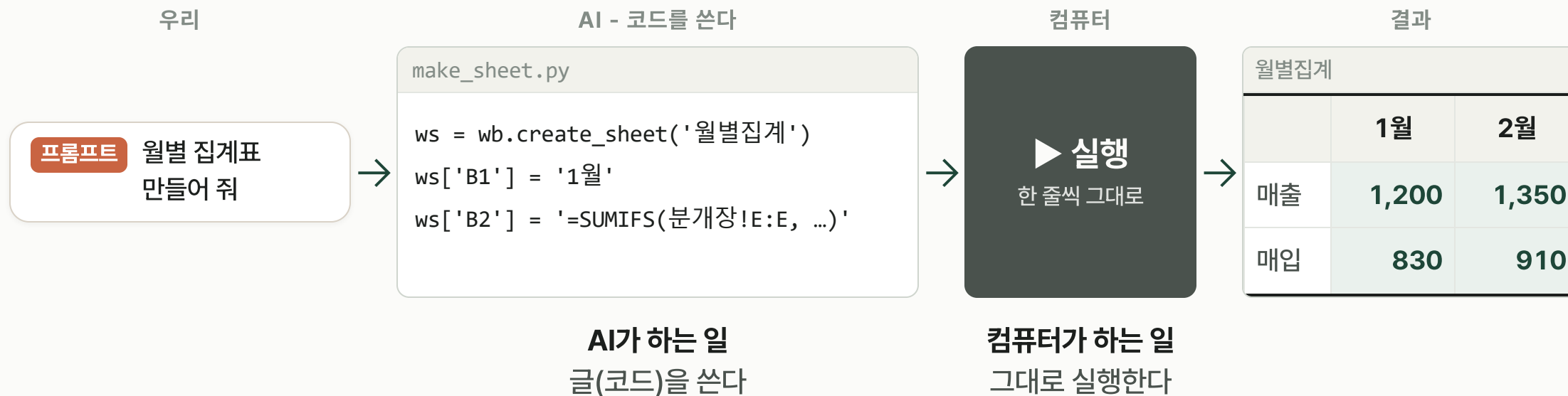
PDF

엑셀도 만들고 워드도 만들어 일을 직접 끝낸다

어떻게 진화했나? 단순 답장을 넘어, 엑셀·워드 파일을 직접 만들며 내가 할 일을 대신 끝내는 AI = 에이전트 (다음 장)

에이전트 - 글만 쓰는 AI가 어떻게 엑셀 파일을 만들까?

LLM은 글만 쓸 줄 안다 - 그런데 엑셀 · 워드 파일을 만들어 준다



~~AI가 엑셀 셀에 직접 입력하는 것이 아니라~~ → AI는 글(코드)을 쓰고, 실행은 컴퓨터가 한다



참고 · LLM의 활용

AI 에이전트 - 이미지는 어떻게 만들까?

LLM이 하는 일

실제로 만드는 쪽

결과

프롬프트 월말 결산 절차
4단계 인포그래픽
만들어 줘



클로드
LLM

① 이미지 생성 모델에 넘긴다

클로드 → 나노바나나



나노바나나가 만든다
픽셀을 직접



② 직접 코드를 쓴다

```
<h1>월말 결산 절차 4단계</h1>
<div class="card">
  <h3>전표 마감</h3>
  <p>모든 거래 내역을 기록...</p>
```



▶ 컴퓨터가 실행한다
브라우저가 코드대로 만든다





참고 · LLM의 활용

이미지 - 생성 모델이 그린 것 vs 코드로 만든 것

이미지 생성 모델 나노바나나가 그린 그림

월말 결산 절차 4단계

매달 반복되는 결산, 네 단계로 정리합니다



+ 그림이 풍부하다

- 글자가 틀릴 수 있다

- 고칠 때마다 달라진다

LLM 클로드가 코드로 만든 그림

MONTH-END CLOSE

월말 결산 절차 4단계



• HTML 코드로 그린 인포그래픽 · 글자와 숫자는 코드에 적힌 그대로

+ 글자가 정확하다

+ 코드만 바꾸면 반영

+ 같은 코드면 같은 결과

가장 큰 차이는 **고칠 때** - 생성 모델은 다시 생성해 다른 곳도 바뀔 수 있고, LLM은 코드 글자만 바꿔 다시 실행한다

프롬프트 - AI에게 일을 요청하는 글

프롬프트 = AI에게 원하는 일을 요청하려고 입력하는 글 - 무엇을, 어떤 순서로, 어떤 맥락에서 할지 전한다

① 프롬프트 엔지니어링

잘 쓰면 원하는 방향으로 나온다

[역할] 너는 20년 차 회계 전문가다

[맥락] 25년 말 결산, 채권명세 첨부

[지시] 1년 초과 50%로 계산해 줘

[형식] 결과는 엑셀 표로

최신 모델은 프롬프트 엔지니어링을 내재화하고 있다 - 공식 없이 구체적으로 말하면 된다

② 지침

매번 하는 말은 지침으로 넣어 둔다

챗봇

ChatGPT 맞춤형 지침 · Claude 프로젝트 지침 칸

클로드 코드

CLAUDE.md 파일에 적어 둔다

CLAUDE.md

- 01_받은자료는 원본이다. 고치지 않는다
- 합계 · 차이 칸은 수식으로 넣는다

일을 시작할 때마다 읽는다 - CLAUDE.md는 3장에서

아무리 완벽한 프롬프트엔지니어링이나 지침도 **확률 모델이라 흔들린다** → 매번 같아야 하는 영역은 **코드**로 고정한다

코드 - 컴퓨터에게 일을 요청하는 글(프로그래밍 언어)

코드 = 컴퓨터에게 원하는 일을 요청하려고 쓰는 글 - 엑셀 함수, VBA도 코드 중 하나다



프롬프트로 시키면 - AI가 읽는 글
입력이 같아도 매번 출력이 달라진다

프롬프트 1월 ~ 12월 매출 합계 알려 줘

1번째 표로 정리해 드렸습니다 ...

2번째 월별 합계는 다음과 같습니다 ...

3번째 연간 매출은 약 1,200 ...

프롬프트로 **줄일 수**는 있어도 **없앨 수**는 없다



코드로 돌리면 - 컴퓨터가 실행하는 글
입력이 같으면 몇 번을 실행해도 같은 결과

엑셀 수식

B14	<i>fx</i>	=SUM(B2:B13)
13	12월	110
14	합계	1,200

코드(파이썬)

=

```
합계 = df["매출"].sum()  
→ 1,200
```

백 번 계산해도 1,200 · 저장해 둔 코드 파일 = 스크립트

순서 · 맥락 · 판단은 **프롬프트**로, 반복되는 계산은 **코드**로 고정한다

03

그 외에 알아야 할 것

모델 · 컨텍스트 윈도우 · 토큰 · 마크다운 · 할루시네이션

그 외에 알아야 할 것

모델 - 학습이 완료된 AI의 두뇌, 앱은 모델을 가져와 쓴다

모델 = AI 회사가 학습시켜 내놓은 두뇌(Fable · Opus · Sonnet · GPT · Gemini) - 앱은 그 모델을 가져와 쓰는 서비스



데이터

웹 문서 · 책 · 뉴스 · 코드



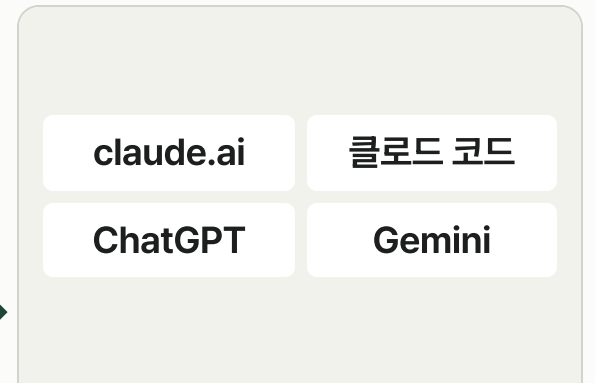
학습

데이터센터에서 몇 달 동안



모델

AI 회사가 학습시켜 내놓은
두뇌



앱

모델을 가져와 쓰는 서비스

앱은 모델을 불러 쓰는 서비스, 모델은 그 안에서 실제로 답을 만드는 AI - 앱이 같아도 모델을 바꾸면 답이 달라진다

그 외에 알아야 할 것

모델 - 무조건 높은 게 정답은 아니다(등급과 추론 수준)

① 어떤 등급의 모델을 쓸까

모델 등급 · 클로드 예시

●	상위 - 정밀 검토 · 복잡한 판단 일반적으로 더 크고 오래 학습 - 느리고 비싸다 Fable 5.1 · Opus 5.5	속도 비용
●	표준 - 일상 실무 대부분의 작업은 여기서 충분하다 Sonnet 5.5	속도 비용
●	경량 - 대량 단순 작업 작게 만들어 빠르고 싸다 Haiku 4.5	속도 비용

② 얼마나 생각하게 할까

추론 수준



낮게 - 답이 뻔한 일

요약 · 분류 · 서식 정리 · 변환

높게 - 따져 봐야 하는 일

회계 판단 · 이슈 검토 ·
복잡한 수식 설계

일의 난이도에 맞춰 고른다 - 요약 · 분류는 낮게, 회계 판단은 높게



참고 · 그 외에 알아야 할 것

모델 - 대표적인 AI 모델, 글로벌 3대장

Anthropic

Claude

Fable 5.1 · Opus 5.5 · Sonnet 5.5
Haiku 4.5

코딩 · 엑셀 실무 · 에이전트 작업 강점

OpenAI

ChatGPT

GPT-6 Astra · GPT-6.1 Sol
GPT-6 Luna

가장 넓은 사용자 · 범용성

Google

Gemini

Gemini 4 Argon
3.8 Flash

긴 문서 · 속도 · 구글 서비스 연동

그 밖의 주요 모델 - Meta Muse Spark · xAI Grok · 중국 GLM(Z.ai) · Kimi · Qwen · DeepSeek · 샤오미 MiMo

회계 · 재무 실무에서는 **Claude**가 벤치마크로도, 사용자 체감으로도 높은 평가를 받고 있다



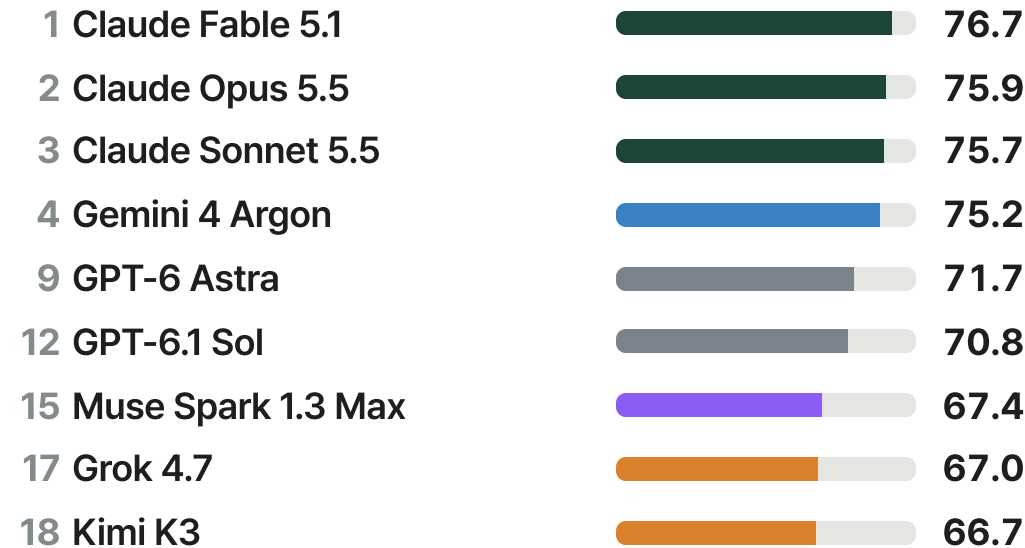
참고 · 그 외에 알아야 할 것

모델 - 회계 · 재무 분야 벤치마크

정답률 %

엑셀 재무모델링

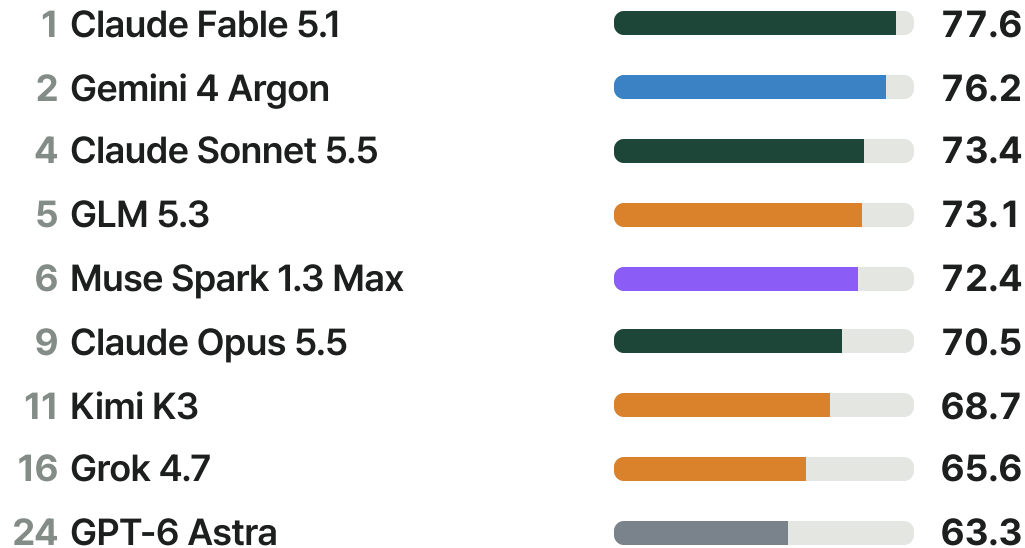
IB · PE 주니어 수준 재무모델을 엑셀로 만들기



정답률 %

세무 리서치

법인세 문제를 법령을 찾아 적용해 답하기



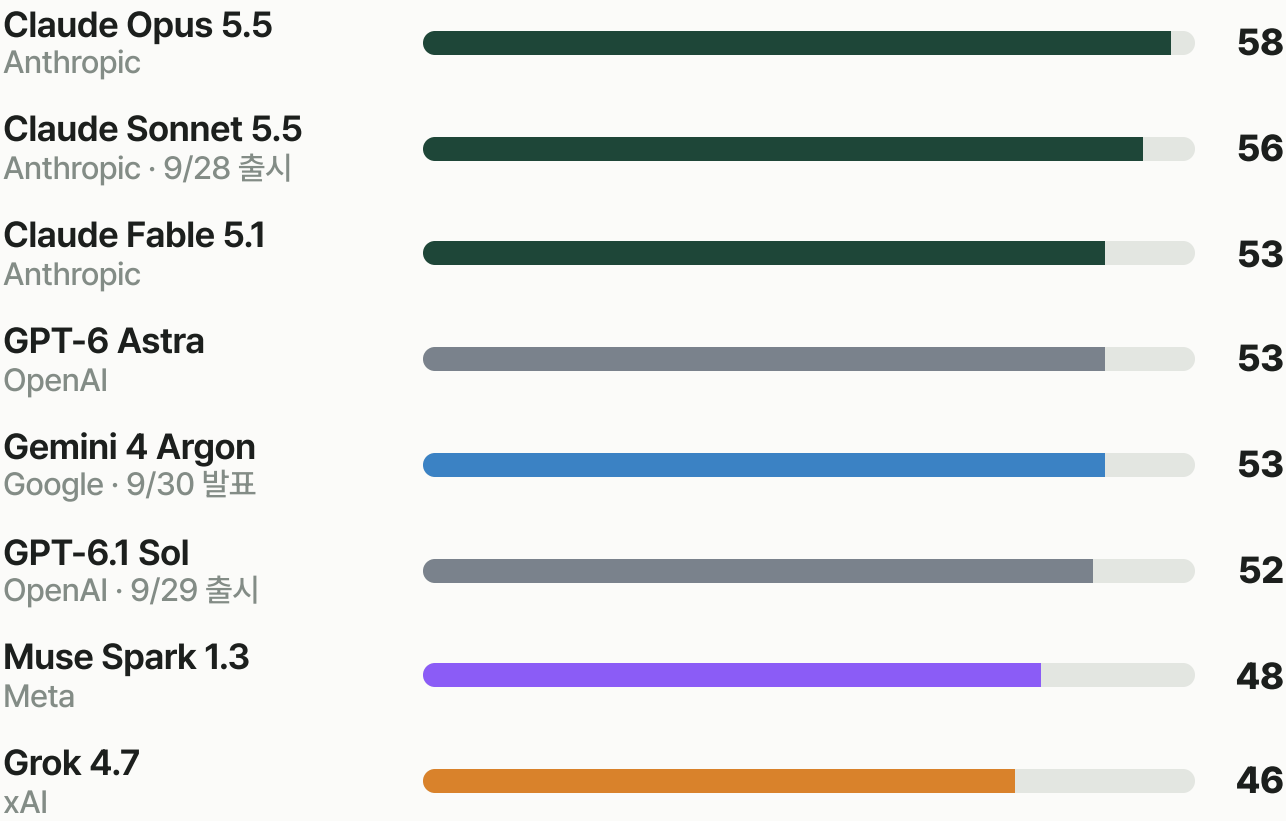
■ Anthropic ■ Google ■ OpenAI ■ Meta ■ xAI · 중국(Z.ai · Moonshot)

출처: Vals AI · 2026.10.4 조회



참고 · 그 외에 알아야 할 것

모델 - 종합 벤치마크



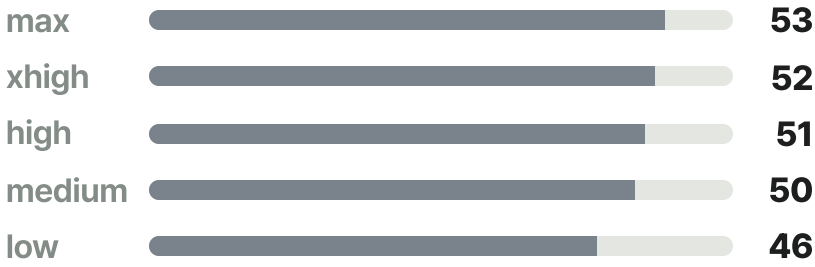
출처: Artificial Analysis · 2026.10.4 조회

같은 모델, 추론 수준만 다르게

Claude Opus 5.5



GPT-6 Astra





참고 · 그 외에 알아야 할 것

모델 - 국내 모델(소버린 AI)

소버린 AI = 나라의 데이터 · 법률 · 문화를 자국 기술로 다루는 AI

LG AI연구원

엑사원

EXAONE

한국어 · 영어 · 금융 · 특허 같은
전문 분야에 강하다

사내 서버에 직접 설치해 쓰기 좋다

네이버

하이퍼클로바X

HyperCLOVA X

국내 법령 · 행정 · 비즈니스 자료를
많이 학습했다

한국 세법 · 제도 맥락 이해에 강점

업스테이지

솔라

Solar

작고 빠른 모델 · 영수증 · 계약서
같은 문서 인식에 특화

사내 문서 검색에 쓰기 좋다

쓰는 이유는 두 가지 - 해외 서버로 보낼 수 없는 사내 보안 데이터, 그리고 국내 세법 · 행정 실무

그 외에 알아야 할 것

컨텍스트 윈도우 - 한 대화창에서의 컨텍스트 용량

컨텍스트 윈도우 = AI가 한 대화에서 한 번에 기억하는 양

기억 용량의 변화

A4 몇 장 → 책 여러 권

초창기 (2022~23)

약 4천 토큰

A4 몇 장 - PDF는 몇 장만 첨부, 대화가 길어지면 더 이상 이어지지
않음

지금

100만 토큰

책 여러 권 - 긴 자료도 한 대화에 넣을 수 있다

컨텍스트 윈도우가 차면

내용이 쌓일수록 정확도가 떨어진다

- ↓ 답변 품질이 떨어진다
쓸데없는 정보 · 맥락이 많아지면 틀린 답도 늘어난다
- ↑ 토큰 소모가 늘어난다
쌓인 맥락을 매번 다시 읽어야 하므로 (토큰)

그래서 이렇게 쓴다

새로운 작업은 새 대화창에서

주제가 바뀌면 새로 연다

길어지면 주기적으로 /compact

지난 대화를 요약해 용량을 비운다 - 3장에서 직접

그 외에 알아야 할 것

토큰 - AI 사용량을 재는 단위

① 토큰이란

AI가 읽고 쓰는 글의 단위

한국어

8토큰

대 손 총당 금 을 설정 합 니다

영어

3토큰

Set the allowance

예시(분할은 모델마다 다르다) - 한국어는
1.5~2배 짧게

전기는 kWh, AI는 토큰으로 잰다 - 용량 ·
요금 · 한도 모두 토큰 단위

② 언제 쓰나

읽을 때와 쓸 때

입력 읽을 때

질문 · 첨부 파일 · AI가 찾아 읽은 파일

출력 쓸 때

답변 · 코드 + 답하기 전 추론 과정

예) 질문 한 줄 + 엑셀 첨부 → 대부분 입력 ·
답과 코드 → 출력
같은 토큰이라도 출력이 입력보다 5배 비싸다

같은 내용도 PDF는 수 배, 이미지 · 스캔은
몇 배~수십 배 - 자료는 표 · 텍스트로

③ 토큰 예산

구독 요금제

5시간 한도 + 주간 한도
(설정 › 사용량에서 확인)

빨리 닳는 것

긴 대화 · 큰 첨부 · 상위 모델 · 커넥터
(MCP 같은 연결)

그래서 구독 요금제에서는 토큰을
아끼는 게 곧 더 오래 쓰는 법

토큰은 읽을 때와 쓸 때만 든다 - 그래서 아끼는 방법도 정해져 있다(다음 장, 4가지 원칙)

그 외에 알아야 할 것

토큰 - 아끼는 4가지 원칙

토큰은 읽을 때(입력)와 쓸 때(출력)만 든다 - 지금까지 본 개념을 합치면 아끼는 법은 네 가지

1 이미지보다 텍스트로 전달

이미지 · 스캔을 판독하면 같은 내용의 텍스트보다
몇 배~수십 배 입력 토큰

← 토큰

2 일상 작업은 가벼운 모델 · 낮은 추론으로

상위 모델은 단가가 높고, 답하기 전 생각도 출력 토큰 -
출력은 입력보다 5배

← 모델 · 추론 수준

3 새로운 작업은 새 대화창에서

질문할 때마다 앞 대화 전체를 처음부터 다시 읽는다

← 컨텍스트 윈도우

4 반복되는 작업은 코드로 고정

코드를 쓰고 결과를 읽을 때만 토큰 - 계산은 컴퓨터가

← 코드

가장 강력한 방법은 4번 - 코드가 실행되는 동안에는 토큰이 들지 않는다(계산은 컴퓨터가 한다)

그 외에 알아야 할 것

마크다운(.md) - 기호로 서식을 표시하는 글쓰기 형식

마크다운 = 기호(#, **, -)로 서식을 표시하는 글 형식 - 사람도 AI도 읽기 쉽다

CLAUDE.md - 메모장

```
# 회계팀 작업 폴더 규칙
## 폴더
- 00_정리전_받은자료 · 01_받은자료는 원본이다. 읽기만 하고 **
  고치지 않는다**
- 작업 중 파일은 02_작업, 결과물은 03_산출에 저장한다
## 엑셀
- 엑셀의 합계 · 차이 같은 계산 칸은 값이 아니라 **수식**으로
  넣는다
```



CLAUDE.md - 미리보기

```
회계팀 작업 폴더 규칙


---


폴더


- 00_정리전_받은자료 · 01_받은자료는 원본이다. 읽기만 하고
    고치지 않는다
- 작업 중 파일은 02_작업, 결과물은 03_산출에 저장한다


엑셀


- 엑셀의 합계 · 차이 같은 계산 칸은 값이 아니라 수식으로
    넣는다

```

CLAUDE.md · SKILL.md 같은 지침 파일도 전부 마크다운 - 외울 건 세 개: # 제목 · ** 강조 · - 목록 (나머지 문법은 다음 장)



참고 · 그 외에 알아야 할 것

마크다운 기본 문법

왼쪽처럼 쓰면 오른쪽처럼 보인다 - 외울 필요 없이 필요할 때 펼쳐 본다

	이렇게 쓰면	이렇게 보인다
제목	# 결산 규칙	결산 규칙
소제목	## 폴더	폴더
굵게	**고치지 않는다**	고치지 않는다
기울임	<i>*참고*</i>	참고
목록	- 원본은 읽기만	• 원본은 읽기만
번호 목록	1. 원장을 읽는다	1. 원장을 읽는다

	이렇게 쓰면	이렇게 보인다		
체크	- [] 대사 확인	☐ 대사 확인		
표	계정 금액	<table><tr><td>계정</td><td>금액</td></tr></table>	계정	금액
계정	금액			
코드	`=SUM(A1:A3)`	=SUM(A1:A3)		
링크	[회계위키](주소)	회계위키		
인용	> 문단 33	문단 33		
구분선	---			

그 외에 알아야 할 것

할루시네이션 - 그럴듯하게 지어낸 틀린 답

할루시네이션(환각) = AI가 사실이 아닌 것을 사실처럼 말하는 것

프롬프트 재고자산 평가손실 환입 근거 문단이 뭐야?

AI

네, **K-IFRS 제1002호 문단 34의2**에 따르면 평가손실 환입은 당기 매출원가에서 차감하며, 환입 한도는 **문단 34의3**에서 직전 3개년 평균으로 정하고 있습니다.

존재하지 않는 문단

예시(재구성)

왜 생기나

사실을 확인해 말하는 게 아니라 "가장 그럴듯한 다음 말"을 고르는 원리의 부작용. 거짓말이 아니다.

어떻게 막나

숫자 · 조문 · 문단은 출처를 요구하고, 근거를 직접 조회시킨다

AI는 사실을 검증해서 말하는 게 아니라 확률로 말한다 - 그래서 **별도 검증**이 중요하고, MCP로 기준서 · 법령을 직접 조회시켜 검증하게 할 수 있다(4장)

LLM의 한계 세 가지 - 클로드 코드는 이렇게 넘는다

개발자들의 도구가 비개발자에게도 필요한 이유 - 내 PC에서, 코드로 일하기 때문

LLM의 한계

챗봇

클로드 코드

① 사전학습

우리 자료를 모른다

필요한 파일을 골라 첨부하고 설명한다

→

내 PC에 있는 모든 폴더와 파일이 컨텍스트가 된다

② 글만 쓴다

직접 만들지 못한다

만든 파일을 매번 내가 내려받아야 한다

→

만든 파일을 내 PC에 바로 저장까지 한다

③ 확률 모델

매번 다르다

코드가 대화 안에만 남아 다음 달엔 새로 짤다

→

사용한 코드를 저장해 두고 다음 번에도 똑같이 재사용한다

코드는 몰라도 된다 코드는 클로드 코드가 짜고 돌리고 저장한다 - 우리는 말로 시키고 결과를 확인한다



참고 · 그 외에 알아야 할 것

엔트로픽도 같은 말을 한다

이 강의에서 한 말

반복은 코드로 고정한다

시작 전에 9쪽

입력이 같으면 몇 번을 돌려도 같은 결과

17쪽 코드

프롬프트는 아무리 잘 써도 흔들린다

16쪽 프롬프트

계산은 컴퓨터가 - 토큰을 아낀다

27쪽 토큰

엔트로픽 공식 문서

매번 똑같아야 하는 일은 코드만이 보장한다

"...deterministic reliability that only code can provide." · 엔지니어링 블로그 2025-10

코드는 결과가 늘 같아서, 일이 일관되고 반복된다

"Because code is deterministic, this workflow is consistent and repeatable." · 같은 글

코드는 매번 같고, 말은 해석이 갈린다

"Code is deterministic; language interpretation isn't." · 공식 스킬 가이드 26쪽

정렬을 AI가 글로 하면 정렬 프로그램보다 훨씬 비싸다

"...far more expensive than simply running a sorting algorithm." · 엔지니어링 블로그

이 강의의 원칙은 클로드를 만든 회사의 공식 권장과 같다

04

부록(Appendix)

AI 발전과정 · 멀티모달 · 이미지는 어떻게 만들까



참고 · 부록

AI는 어떻게 엑셀을 넘어 24시간 일하게 되었을까?



오늘 배운 개념이 등장한 순서 그대로 - 그리고 지금 여기



참고 · 부록

ChatGPT의 등장과 프롬프트 엔지니어링

특징

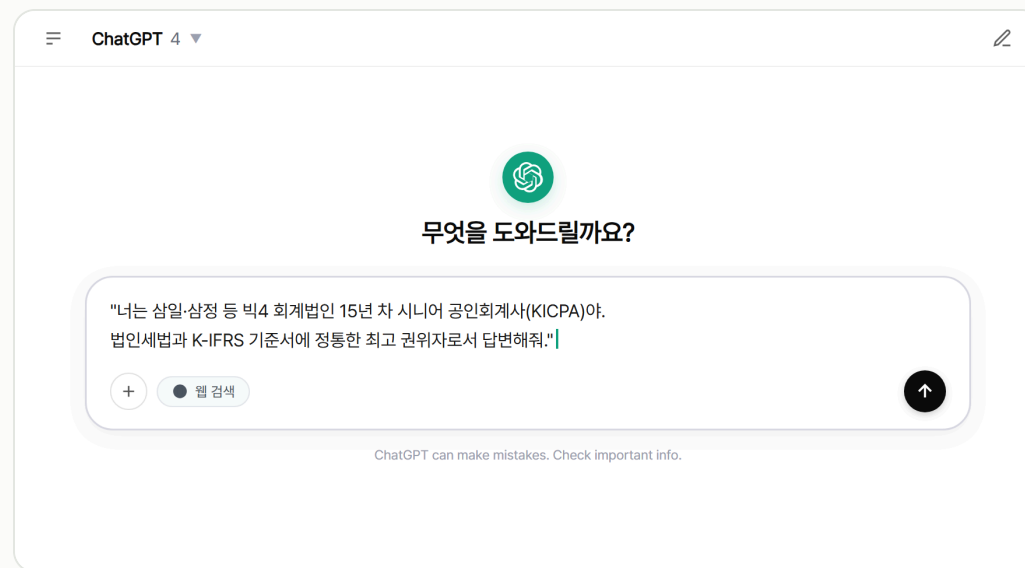
LLM(Large Language Model)의 등장

ChatGPT라는 대화형 AI가 등장하여 사용자가 **자연어로** 지시하면 알아서 답변을 해주는 시대가 열렸습니다. 이에 답변의 퀄리티를 높이기 위한 "**프롬프트 엔지니어링**"이 핵심 역량으로 대두되었습니다.

한계

할루시네이션 & 기억 용량

없는 세법 조항이나 회계 기준서를 지어내는 **할루시네이션(환각)** 현상이 심각했고, 대화창 하나당 **작업 기억(컨텍스트)**이 **A4 2~3장에 불과**해 두꺼운 책을 통째로 읽혀줄 방법이 없었습니다.



▲ ChatGPT에 페르소나(전문가 역할 부여) 프롬프트를 입력하는 실제 대화 화면

컨텍스트 윈도우 확장과 RAG(검색 증강 생성)

특징

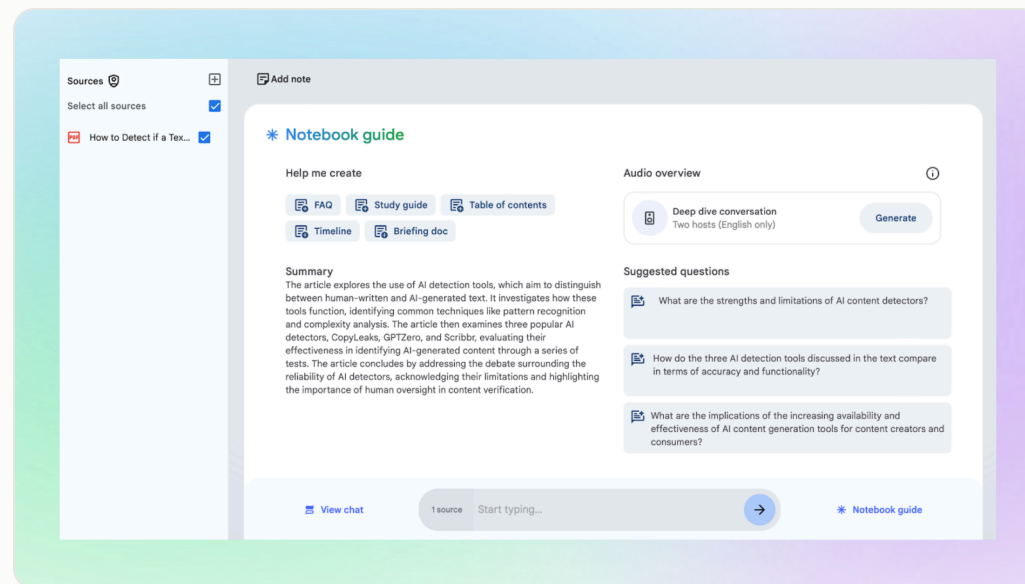
컨텍스트 윈도우 확장(4천 → 100만 토큰)과 RAG

작업 기억이 **100만 토큰(책 수십 권 분량)**으로 확장되어 맥락 (**Context**)을 저장할 수 있게 되었고, 내 문서를 검색해 "오직 이 근거 위에서만 답하라"고 요청하는 **RAG(검색 증강 생성)** 기술이 유행했습니다.

한계

'글쓰기'에 머무른 AI 활용

텍스트 답변과 요약 능력은 탁월했지만, 우리가 실제로 사용하는 프로그램이나 오피스 문서를 직접 완성하고 편집하지는 못했습니다.



▲ 구글 NotebookLM - 방대한 사내 문서를 올려두고 오직 문서 근거로만 답변하는 RAG 대표 서비스

AI 에이전트의 시대 - 커서 · 클로드 코드

특징

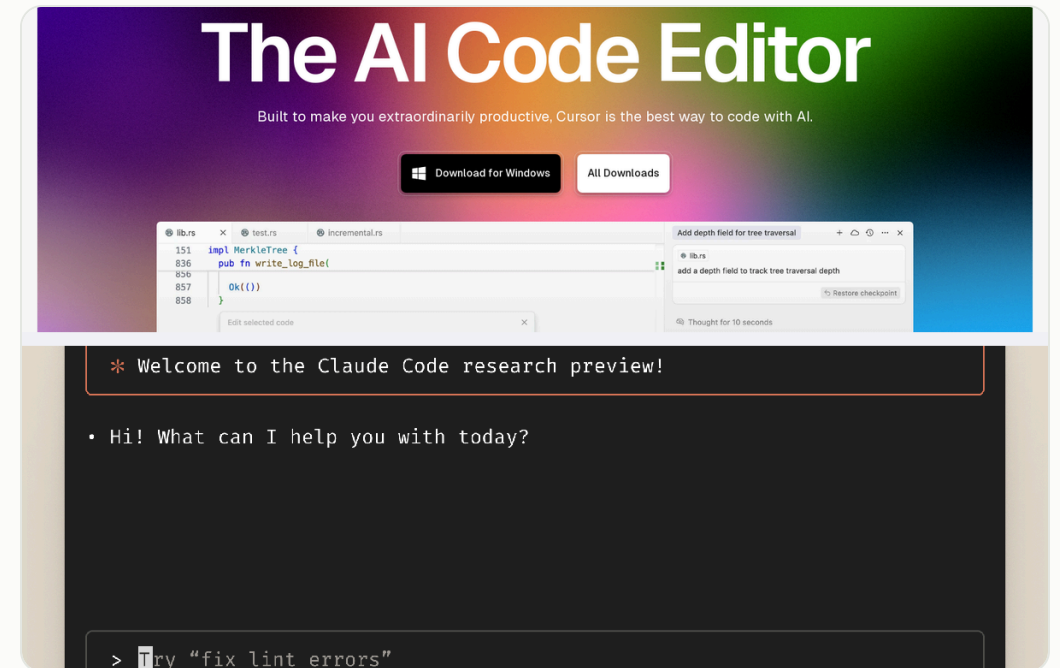
AI가 코딩하고 직접 실행·수정

Cursor와 Claude Code의 등장으로 AI가 코드를 작성하고 직접 터미널에서 실행하며 스스로 개발. 비개발자도 자연어로 자기만의 프로그램을 제작(Vibe Coding)할 수 있는 기회가 열렸습니다.

한계

오피스(엑셀·PPT) 실무의 벽

GitHub 등 오픈소스 학습으로 웹사이트나 앱은 잘 만들었지만, 실무에 필수적인 오피스(엑셀·PPT) 문서를 코드로 정교하게 다루는 것에는 여전히 한계가 있었습니다.



▲ 바이브 코딩 양대 도구 - 커서(Cursor IDE 에디터) & 클로드 코드(Claude Code CLI)



참고 · 부록

스킬 · MCP의 등장과 오피스 에이전트

특징

실무 문법(스킬)과 외부 데이터(MCP)의 결합

단순 개발자용 코드 생성을 넘어 **스킬(Skill)**을 통해 openpyxl 등 실무 서식·수식 문법을 장착하고 **MCP**로 외부 데이터를 실시간 연결하며 진짜 오피스 문서를 완성하는 에이전트로 도약했습니다.

그래서

클로드 엑셀 · 클로드 코드도 이 흐름 위에 있다

MCP(2024.11)와 **엑셀 등 오피스 스킬(2025.10)**이 먼저 나왔고, 곧이어 **클로드 엑셀(2025.10 베타)**이 엑셀 안에서 일하는 애드인으로 나왔다 - 스킬과 MCP를 그대로 함께 쓴다.



▲ 앤트로픽 공식 스킬(Skill) 패키징 구조 - 지침(SKILL.md)과 실무 양식·규칙(Resources) 탑재



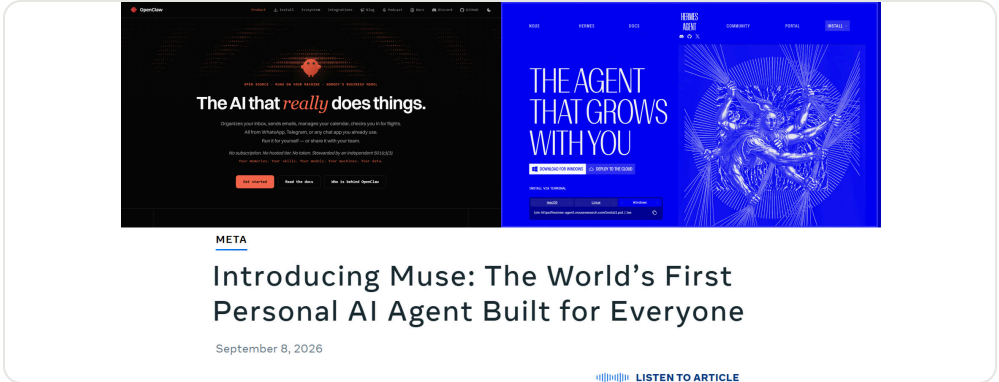
참고 · 부록

최신 동향 - 24시간 에이전트 · 컴퓨터를 직접 쓰는 AI

① 24시간 개인 에이전트

내가 자는 동안에도 일한다

메신저로 시키면 내 PC나 클라우드에서 **24시간 돌며** 메일 · 일정 · 예약 · 브라우저 작업을 대신한다. 오픈소스 **OpenClaw · Hermes Agent**에 이어 **Meta Muse(9.8) · OpenAI Dots(9.29)**까지.

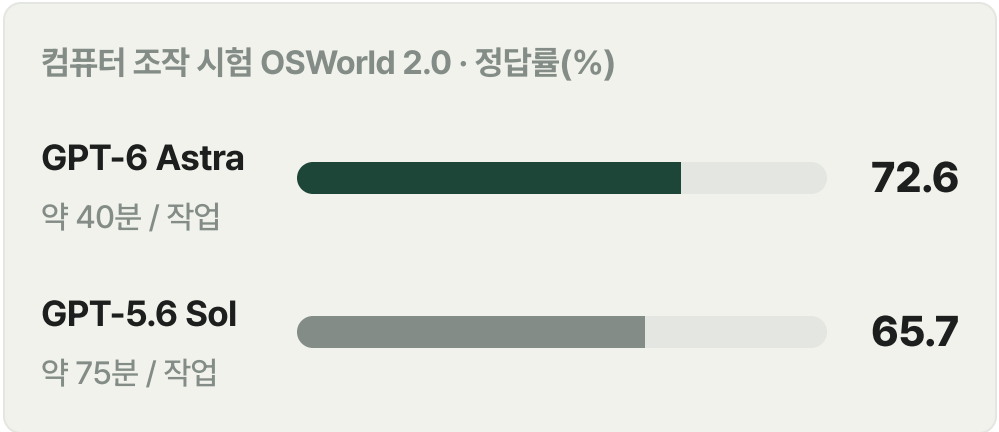


▲ OpenClaw · Hermes Agent 공식 사이트, Meta의 Muse 발표(2026.9.8)

② 컴퓨터 유즈(Computer Use)

화면을 보고 마우스 · 키보드를 다룬다

OpenAI **GPT-6 Astra(9.3)**는 브라우저 · 스프레드시트 · 데스크톱 앱을 사람처럼 조작해 **여러 단계 업무를 끝낸다**. 코드가 없는 회사 시스템도 화면으로 다루는 방향.



▲ OSWorld 2.0(오프라인) · 출처 marktechpost 2026.9.3