나만의 인공지능 · 4강

두뇌를 꽂다

3강에서 **두뇌를 받았습니다.** 오늘은 그 두뇌를
채팅창 밖으로 꺼내 내 프로그램에 연결합니다.

11434 올라마 문 번호	**1234** LM 스튜디오 문 번호	**1줄** 도구를 바꿀 때 고치는 코드	**0원** 오늘도 드는 돈

01 · 어디까지 왔나

지금까지는 옮겨 놓기만 했습니다

1. **1강 · 두뇌를 만들었습니다**
   함수에서 시작해 타이타닉 두뇌를 직접 만들었습니다.
2. **2강 · 점수를 올렸습니다**
   층을 바꾸고 데이터를 손봐 순위표에 올렸습니다.
3. **3강 · 두뇌를 사 왔습니다**
   허깅페이스에서 골라 LM 스튜디오·올라마로 돌렸습니다.
4. **4강 · 오늘 — 꽂습니다**
   채팅창 밖으로 꺼내 프로그램에 연결합니다.

3강까지는 챗지피티를 내 컴퓨터로 옮겨 놓은 것에 가깝습니다.

02 · 왜

사람이 쳐야 도는 건 자동화가 아닙니다

채팅창은 **사람이 손으로 치는 곳**입니다. 내가 물어야만 답합니다.

자비스는 그렇지 않습니다. **내가 묻지 않아도 일을 합니다.**

오늘 하는 일은 이 한 문장입니다 —
내 컴퓨터의 두뇌에게 프로그램이 말을 걸게 한다.

그 통로를 API 라고 부릅니다. 실제로는 주소 하나입니다.

03 · 주소

주소는 받는 게 아니라 정해져 있습니다

도구	주소	어떻게 아나
LM 스튜디오	`http://127.0.0.1:1234`	모델 고르고 서버 켜면 **화면에 뜹니다**
올라마	`http://127.0.0.1:11434`	화면이 없습니다. **항상 이 주소입니다**

올라마는 설치하면 **자동으로 켜집니다.** 따로 실행할 필요가 없습니다.

확인은 브라우저에 주소를 치는 것 하나 — **Ollama is running** 이 뜨면 끝입니다.

127.0.0.1 은 내 컴퓨터 자신을 가리키는 주소이고, 뒤의 숫자는 문 번호입니다. 인터넷으로 나가지 않습니다.

04 · 명령어

올라마는 네 개면 됩니다

명령	하는 일
`ollama pull 모델이름`	내려받습니다
`ollama run 모델이름`	대화창에서 바로 써 봅니다 (`/bye` 로 나감)
`ollama list`	**내가 가진 두뇌 목록** — 여기서 이름을 복사합니다
`ollama ps`	지금 메모리에 올라간 두뇌

허깅페이스 GGUF 는 `ollama pull hf.co/올린사람/저장소:Q4_K_M` 로 바로 받습니다. 모델 페이지의 Use this model → Ollama 에 그 주소가 나옵니다.

05 · 핵심

코드는 한 줄만 바뀝니다

두 도구는 다른 프로그램인데 **말을 주고받는 형식이 같습니다.** 둘 다 오픈에이아이 형식입니다.

```
from openai import OpenAI

# LM 스튜디오면 1234, 올라마면 11434. 이 줄 하나만 바꾸면 됩니다.
ai = OpenAI(base_url="http://127.0.0.1:11434/v1",
            api_key="local")

답 = ai.chat.completions.create(
    model="gemma4:e2b",
    messages=[{"role": "user", "content": "안녕?"}],
)
print(답.choices[0].message.content)
```

주소 한 줄, 모델 이름 한 줄. 나머지는 손댈 것이 없습니다.

06 · 여기를 보세요

돈 내던 자리에 아무거나 적습니다

원래 `api_key` 자리에는 **돈 내고 받은 열쇠**가 들어갑니다.
오픈에이아이도, 구글도, 앤트로픽도 그렇습니다.

그런데 지금은 **아무 글자나 적어도 됩니다.**
내 컴퓨터라서 검사하는 사람이 없기 때문입니다.

이것이 소유한다는 뜻입니다.

07 · 함정

여기서 막힙니다 — 셋

1

모델 이름에 꼬리표

`llama3.2` → not found

`llama3.2:1b` → 정상

ollama list 를 통째로 복사

2

열쇠에 한글 금지

아무거나 맞지만 **영문·숫자로.**

한글을 넣으면

`UnicodeEncodeError`

3

오늘은 코랩이 아닙니다

코랩은 구글 컴퓨터입니다.

거기서 127.0.0.1 은 **구글 자신**을 가리킵니다.

내 컴퓨터에서 실행

1·2강은 코랩이었습니다. 오늘부터 성격이 바뀝니다 — 두뇌를 소유했기 때문에 생기는 불편입니다.

08 · 실습

코드가 알아서 찾게 합니다

주소도 모델 이름도 **사람이 적지 않게** 만들 수 있습니다.
켜져 있는 쪽을 찾아 붙고, 둘 다 꺼져 있으면 무엇을 켜야 하는지 알려 줍니다.

```
후보 = [("올라마", "http://127.0.0.1:11434/v1"),
        ("LM 스튜디오", "http://127.0.0.1:1234/v1")]

for 이름, 주소 in 후보:
    try:
        모델들 = [m["id"] for m in 목록받기(주소)]
        if 모델들:
            print(이름, "찾음 ·", 모델들[0]); break
    except Exception:
        continue
```

실제로 돌린 결과 — 올라마 찾음 · dama-choco:latest / gemma4:e2b / llama3.2:1b

09 · 뼈대

에이전트는 세 덩어리입니다

1. **두뇌**
   무엇을 할지 정합니다. 오늘 연결한 로컬 AI 가 여기 들어갑니다.
2. **손**
   실제로 하는 일. 파일 열기, 검색, 메일 보내기 같은 것들입니다.
3. **반복**
   다 될 때까지 두뇌에게 다시 묻습니다. 이것이 자동화입니다.

두뇌가 **주소 한 줄**로 붙어 있어서
더 좋은 모델이 나오면 그 줄만 바꾸면 됩니다.
도구는 바뀝니다. 구멍의 모양은 그대로입니다.

10 · 정리

오늘 자비스에 두뇌를 꽂았습니다

아직 눈도 귀도 입도 없습니다. 물으면 답하는 수준입니다.

손을 하나씩 달아 주면 **스스로 일하는 비서**가 됩니다.

📖 4강 이북으로 복습하기

강의 목록

Connect AI LAB · AI CITY BUILDERS