

AI CODE EDITOR:

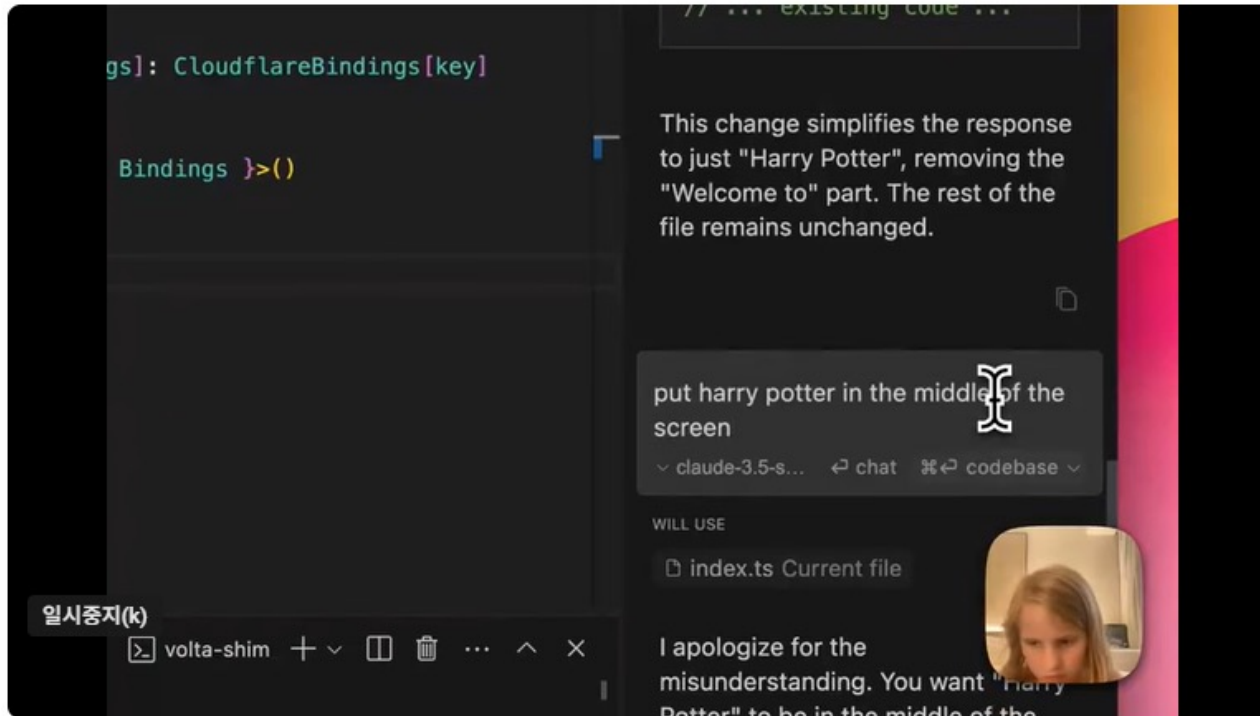


LEE KWAN HO

25.05.24

Cursor 소개

- Anysphere란 회사가 2023년 7월 4일 출시한 AI code editor
 - 현재 기업가치가 90억달러로 평가받고 있음
- 코딩을 모르는 8살의 꼬마 아이가 CURSOR을 이용해 45분만에 해리포터 챗봇을 만드는 영상이 큰 화제가 됨

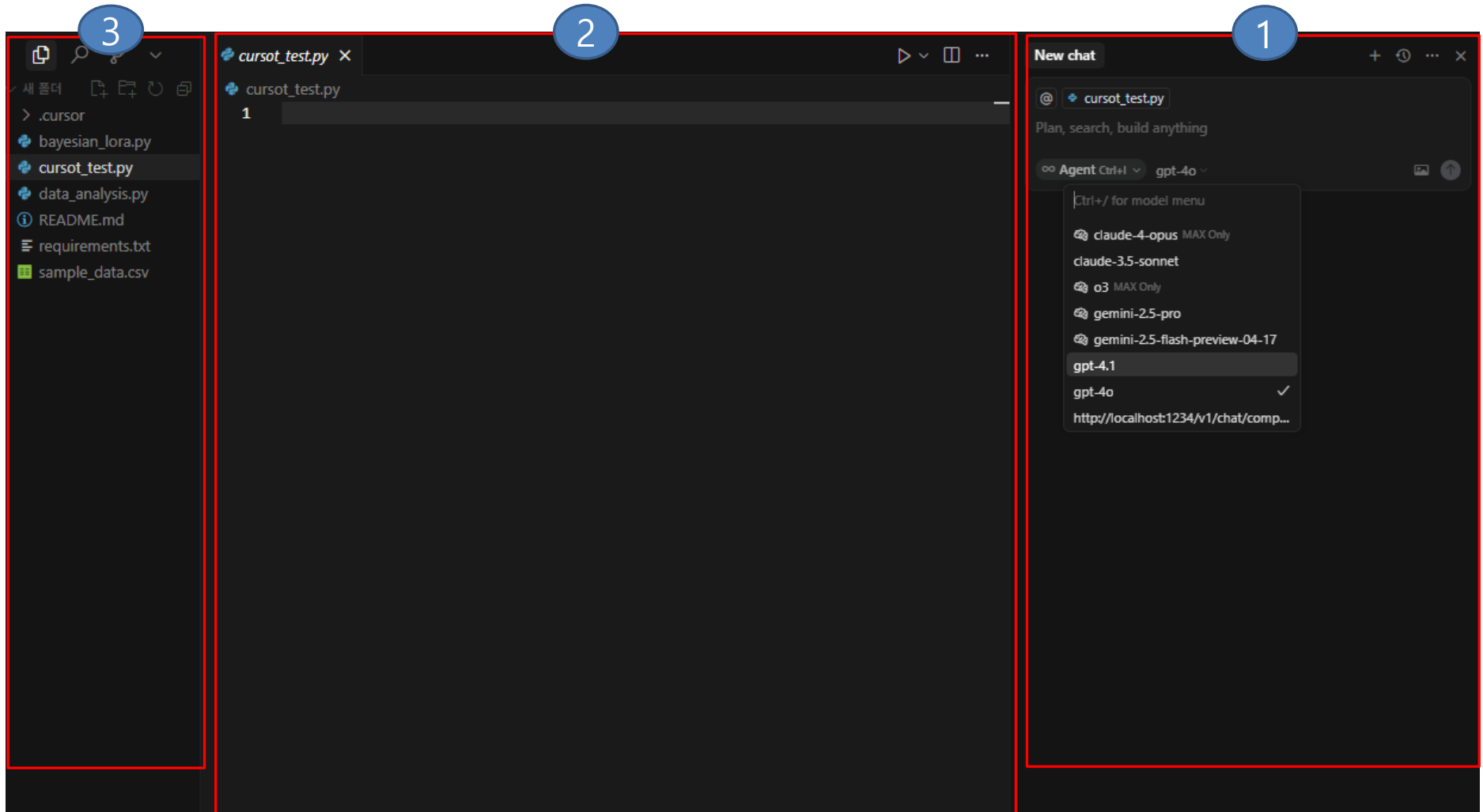


Cursor 의미

- 현재 AI CODE EDITOR 툴들이 많은 관심을 받고 있음
 - VIBE coding이라는 말이 새로 생김
 - 25년 5월 16일 OPEN AI도 AI code editor 툴 **codex**를 발표
- 코딩 관련 업무의 효율성을 올릴 수 있는 툴
- 데이터나 코드를 외부에 유출하지 않으면서도 CURSOR를 사용할 수 있는 방법

Cursor 사용방법

- 시작화면 : 코드 편집기 옆에 LLM이 장착 되어 있음



Cursor - code generation

- 자연어로 코드를 생성, LLM이 생성한 코드가 바로 내 파일에 작성됨

The screenshot displays the Cursor IDE interface. The main editor window shows a Python file named `cursor_test.py` with the following code:

```
1 import numpy as np
2 from sklearn.linear_model import LinearRegression
3
4 # 예시 데이터 (X: 입력, y: 출력)
5 X = np.array([[1], [2], [3], [4], [5]])
6 y = np.array([2, 4, 6, 8, 10])
7
8 # 선형 회귀 모델 생성 및 학습
9 model = LinearRegression()
10 model.fit(X, y)
11
12 # 예측
13 demo_X = np.array([[6], [7]])
14 demo_pred = model.predict(demo_X)
15
16 print("예측값:", demo_pred)
17 print("가중치(가중치):", model.coef_)
18 print("절편:", model.intercept_)
19
```

The right sidebar shows the "선형 회귀 모델 만들기" (Create Linear Regression Model) panel. It contains the following text:

선형 회귀 모델을 만들고 싶어

Restore checkpoint

선형 회귀 모델을 만들기 위해서는 Python의 대표적인 라이브러리인 scikit-learn을 사용하는 것이 가장 쉽고 일반적입니다. 아래는 간단한 선형 회귀 모델 예제 코드입니다.

이 코드를 `cursor_test.py` 파일에 추가해드릴까요, 아니면 다른 파일에 추가할까요?

우선 예시 코드를 보여드릴게요.

python Apply to cursor_test...

```
import numpy as np
from sklearn.linear_model import LinearRegression

# 예시 데이터 (X: 입력, y: 출력)
X = np.array([[1], [2], [3], [4], [5]])
y = np.array([2, 4, 6, 8, 10])

# 선형 회귀 모델 생성 및 학습
model = LinearRegression()
model.fit(X, y)

# 예측
X_test = np.array([[6], [7]])
y_pred = model.predict(X_test)

print("예측값:", y_pred)
print("가중치(가중치):", model.coef_)
print("절편:", model.intercept_)
```

이 코드는 다음을 수행합니다:

- 입력 데이터(X)와 출력 데이터(y)를 준비합니다.
- 선형 회귀 모델을 생성하고 학습합니다.
- 새로운 입력값에 대해 예측을 수행합니다.
- 모델의 가중치(가중치)와 절편을 출력합니다.

원하신다면, 이 코드를 바로 `cursor_test.py`에 추가해드릴 수 있습니다.

진행해도 괜찮을까요?

Cursor - Edit code & Question

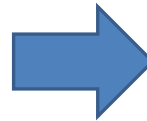
- 수정하고 싶은 코드를 지정해서 수정하고 수정한 근거에 대해 질문 가능

```
cursor_test.py > ...
1  import numpy as np
2  from sklearn.linear_model import LinearRegression
3
4  # 예시 데이터 (X: 입력, y: 출력)
5  X = np.array([[1], [2], [3], [4], [5]])
6  y = np.array([2, 4, 6, 8, 10])
7
8  # 선형 회귀 모델 생성 및 학습
9  model = LinearRegression()
10 model.fit(X, y)
11
12 # 예측
13 demo_X = np.array([[6], [7]])
14 demo_pred = model.predict(demo_X)
15
16 print("예측값:", demo_pred)
17 print("기울기(가중치):", model.coef_)
18 print("절편:", model.intercept_)
19
```

로지스틱 회귀분석 모형으로 바꾸고 싶어

gpt-4.1

Edit Selection



```
model = LinearRegression()
model.fit(X, y)
from sklearn.linear_model import LogisticRegression

# y 값을 0과 1로 변환 (예시: 5 이상이면 1, 아니면 0)
y_binary = (y >= 5).astype(int)

model = LogisticRegression()
model.fit(X, y_binary)
```

Cursor - Edit code & Question

- 수정하고 싶은 코드를 지정해서 수정하고 수정한 이유 대해 질문 가능
 - 또는 이 부분을 최적화 해줘, 더 효율적으로 바꿔줘 와 같은 것도 가능

```
왜 이런식으로 수정한거야?
```

gpt-4.1

Edit Selection

```
# y 값을 0과 1로 변환 (예시: 5 이상이면 1, 아니면 0)
y_binary = (y >= 5).astype(int)

model = LogisticRegression()
model.fit(X, y_binary)
```

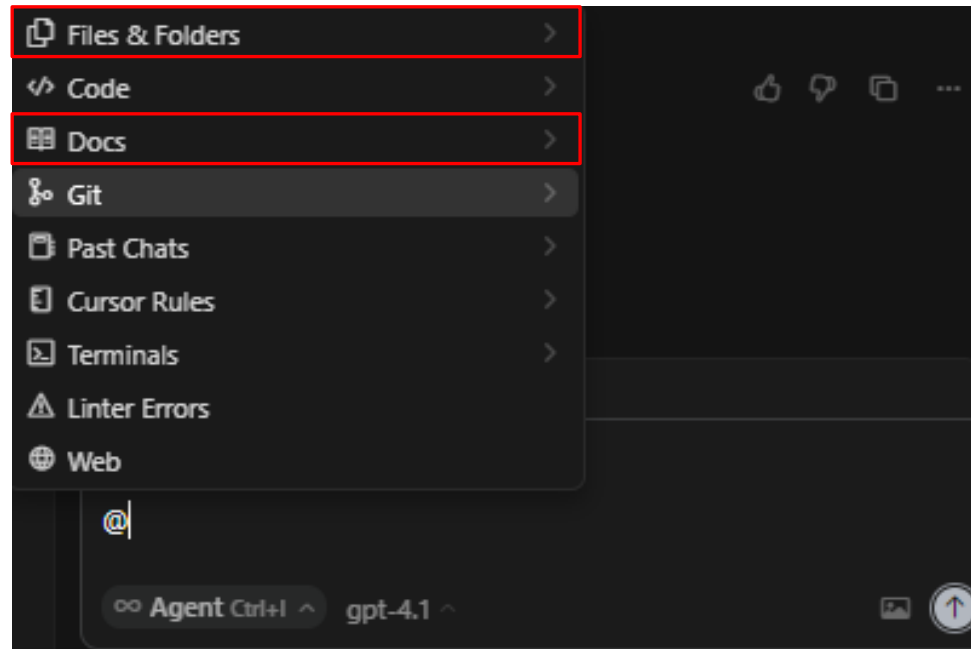


```
# y 값을 0과 1로 변환하는 이유는 로지스틱 회귀가 이진 분류 문제에 사용되기 때문입니다.
# 여기서는 y 값이 5 이상이면 1, 아니면 0으로 변환하여 이진 분류 문제로 만듭니다.
y_binary = (y >= 5).astype(int)

# 변환된 이진 레이블을 사용하여 로지스틱 회귀 모델을 학습합니다.
model = LogisticRegression()
model.fit(X, y_binary)
```

Cursor - Chat

- 코드에 대해서 LLM과 대화가 가능하다.



- Files & Folders : 프로젝트의 특정 파일을 참조하고 싶은 경우
- Docs : 사용할 라이브러리에 대한 문서를 참조하고 싶은 경우

Cursor - Chat

- Folder안에 있는 데이터 파일에 대해서 분석해줘 와 같은 질문 가능

2

폴더에 있는 csv파일을 분석하는 코드를 짜줘

CSV데이터 –
Tip을 얼마나 주는지에
관한 데이터

- 관측치 244, 변수 7개
- Tip
- Total_bill
- Sex
- Smoker
- Day
- Time
- Size

연동되어 있는
폴더와 파일들

Files & Folders

- analyze_tips.py
- local_llm_cursor \
- local_image_model.py
- tip.csv
- requirements.txt
- local_llm_down.py
- README.md
- ollama_chat.py

최소값, 최대값 등
자, 건수 비교

1

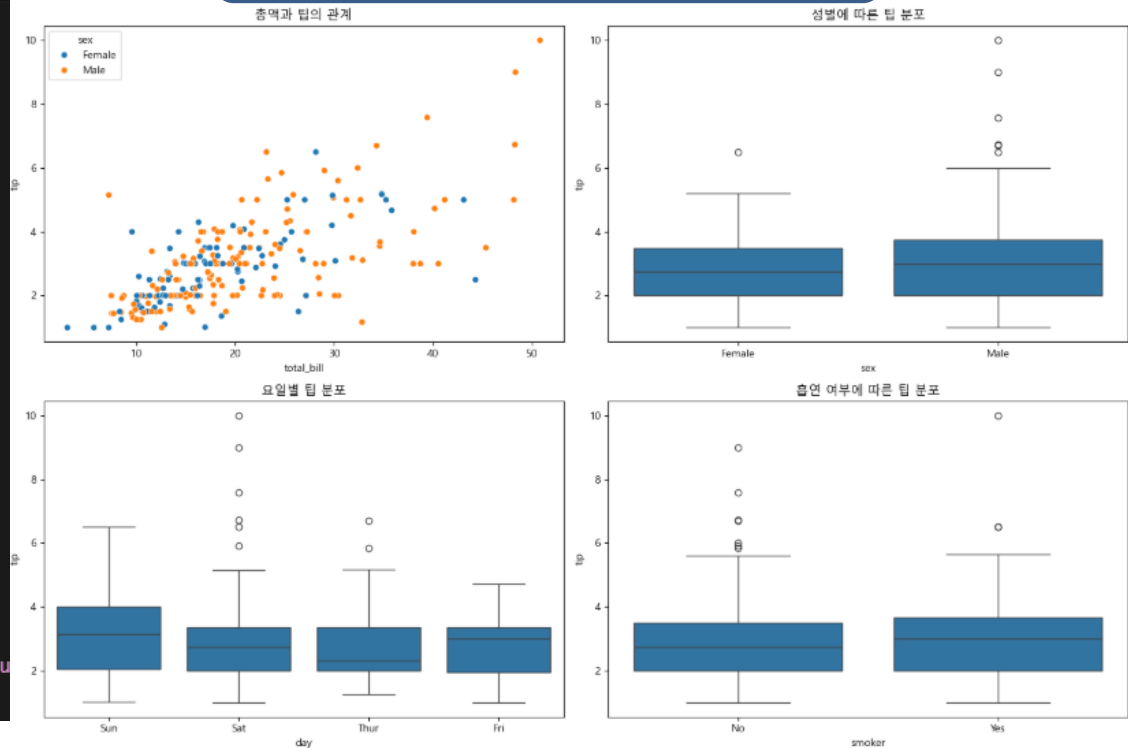
Cursor - Chat

- 바로 실행하여 확인 가능

Cursor가 생성한 코드

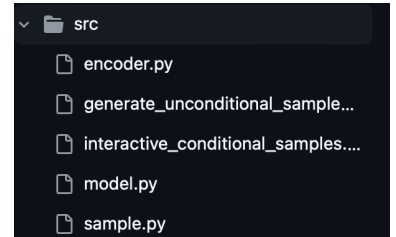
```
analyze_tips.py > ...
1 import pandas as pd
2 import matplotlib.pyplot as plt
3 import seaborn as sns
4 import matplotlib.font_manager as fm
5 from sklearn.linear_model import LinearRegression
6 import numpy as np
7
8 # 한글 폰트 설정
9 plt.rcParams['font.family'] = 'Malgun Gothic' # 윈도우의 경우
10 plt.rcParams['axes.unicode_minus'] = False # 마이너스 기호 깨짐 방지
11
12 # 데이터 로드
13 df = pd.read_csv('tip.csv')
14
15 # 기본 통계량 확인
16 print("\n=== 기본 통계량 ===")
17 print(df.describe())
18
19 # 성별에 따른 팁 분석
20 print("\n=== 성별에 따른 팁 분석 ===")
21 gender_tip_stats = df.groupby('sex')['tip'].agg(['mean', 'std', 'count'])
22 print(gender_tip_stats)
```

코드 실행 결과



Cursor - Chat

- Folder안에 있는 여러 개의 파일에 대한 설명 요청 가능
 - → Chat GPT는 폴더를 첨부할 수 없음



src폴더 안에 model.py의 역할에 대해서 궁금해

model.py 파일은 GPT-2 모델의 핵심 구조와 연산을 정의하는 파일입니다. 주요 역할은 다음과 같습니다:

1. 하이퍼파라미터 및 유틸리티 함수 정의

- `default_hparams()`: 모델의 기본 하이퍼파라미터 (`n_vocab`, `n_ctx`, `n_embd`, `n_head`, `n_layer` 등)를 설정합니다.
- `shape_list`, `softmax`, `gelu`, `norm` 등: 텐서 연산, 정규화, 활성화 함수 등 다양한 유틸리티 함수들을 제공합니다.

2. 트랜스포머의 핵심 구성 요소 구현

- `conv1d`, `attention_mask`, `attn`, `mlp`, `block` 등: GPT-2의 트랜스포머 블록을 구성하는 핵심 연산(어텐션, MLP, 레이어 정규화 등)을 직접 구현합니다.

3. 모델 전체 구조 정의

- `model()`: 입력 토큰(`x`)과 과거 상태(`past`)를 받아, 임베딩 → 여러 트랜스포머 블록 → 최종 정규화 → 로짓(logits) 계산까지 전체 GPT-2 모델의 순전파(forward) 과정을 정의합니다.
- 이 함수는 결과로 각 레이어의 present state(어텐션 캐시)와, 다음 토큰 예측을 위한 logits를 반환합니다.

즉, 이 파일은 GPT-2의 트랜스포머 모델을 TensorFlow로 직접 구현한 핵심 파일로, 모델의 구조, 연산, 하이퍼파라미터, 순전파 과정을 모두 포함하고 있습니다.

더 궁금한 부분이나, 특정 함수에 대한 상세 설명이 필요하면 말씀해주세요!

5. GPT2LMHeadModel

- GPT2Model 위에 언어 모델링용 Linear Head 추가
- 최종적으로 token-wise logits 반환

python

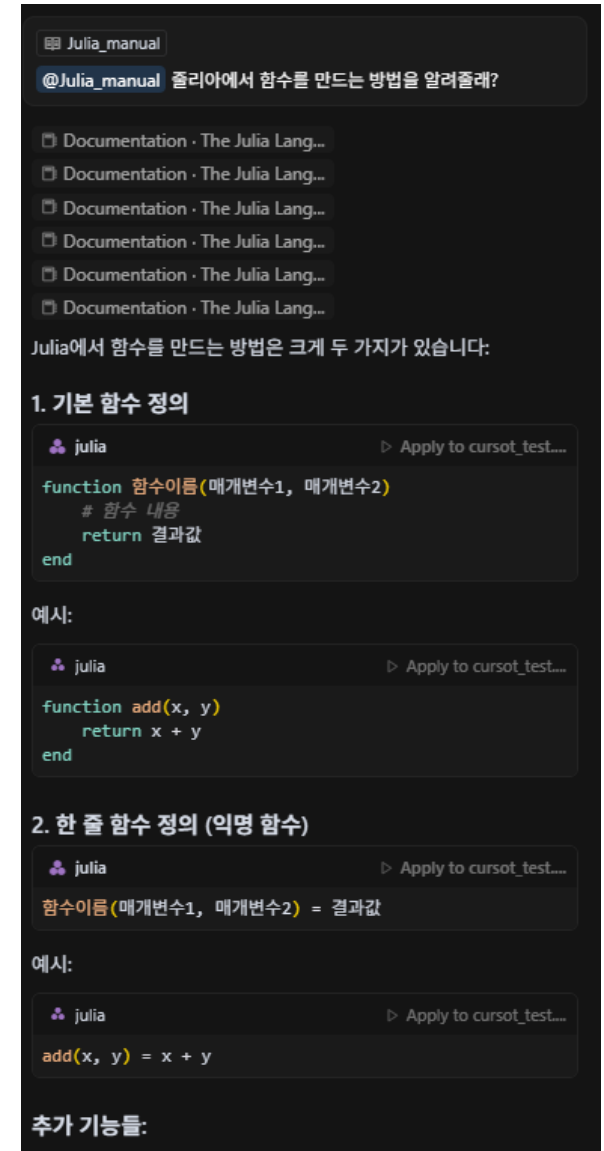
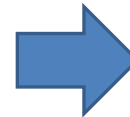
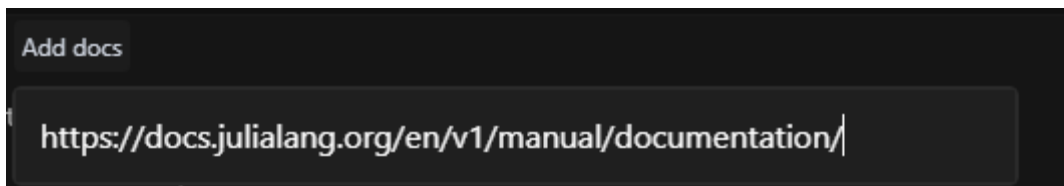
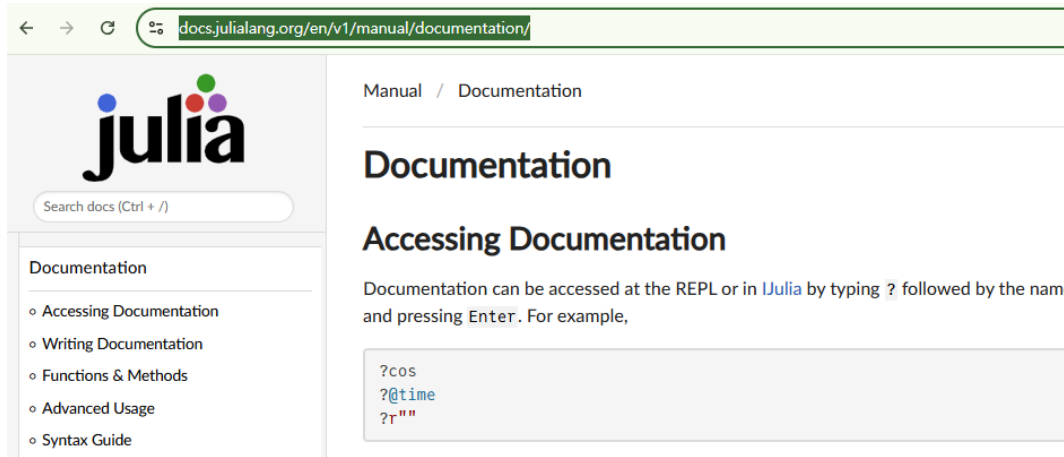
```
class GPT2LMHeadModel(nn.Module):
    def forward(self, input_ids):
        hidden_states = self.transformer(input_ids)
        logits = self.lm_head(hidden_states)
        return logits
```

ChatGPT

CURSOR

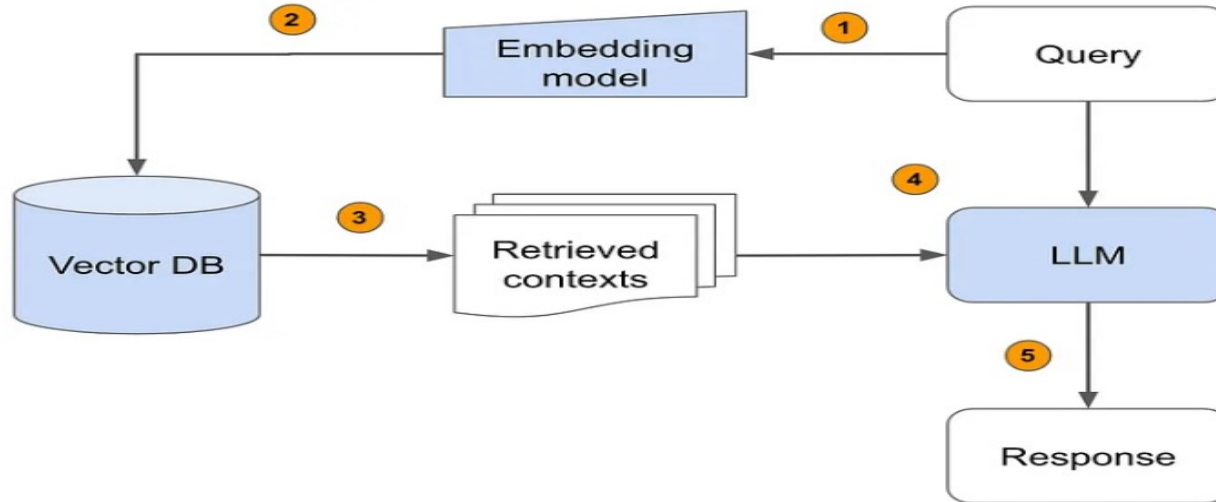
Cursor - Chat

- 라이브러리의 사용 설명서에 대한 문서를 참조한 코딩 및 수정 가능

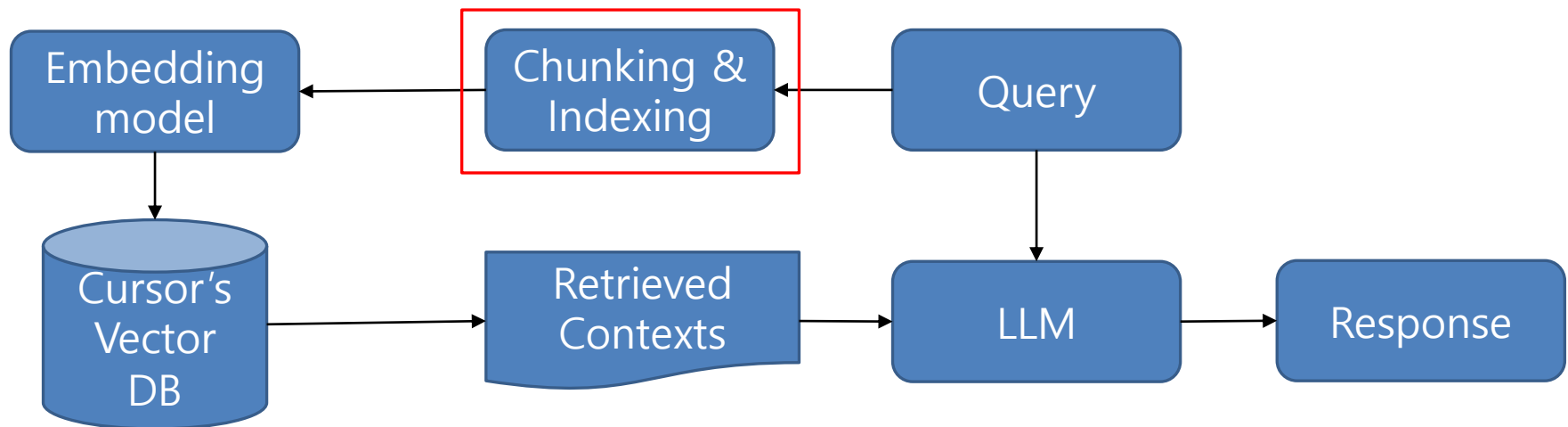


Cursor의 RAG 활용

- 일반적인 LLM + RAG 활용 방식

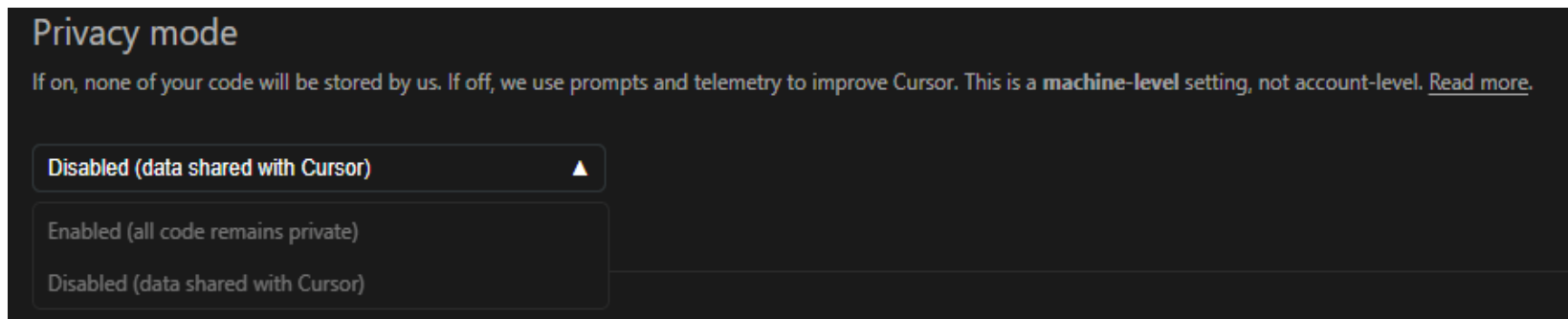


- Cursor



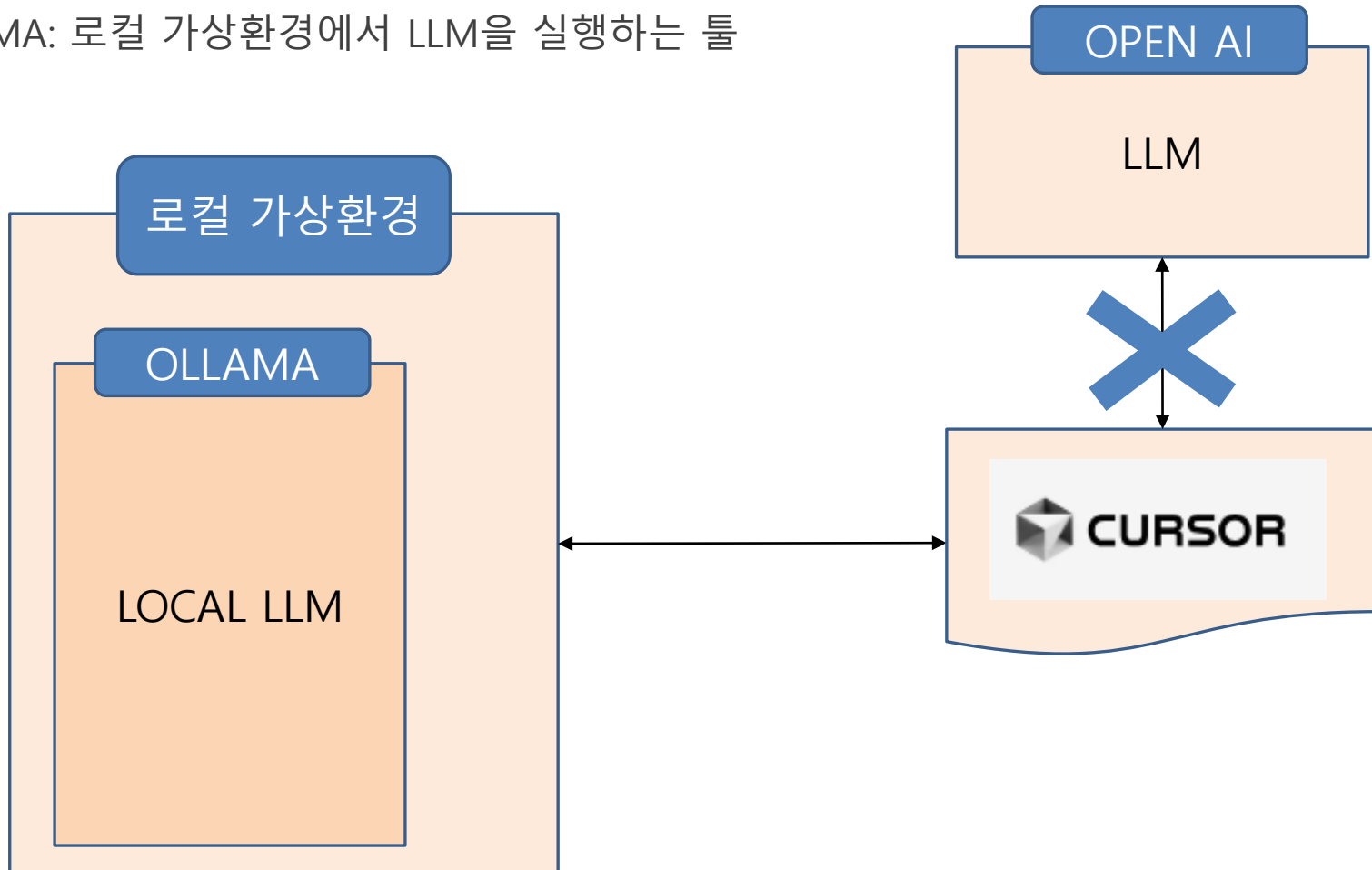
Cursor - Privacy Mode

- **Privacy Mode:** 나의 코드가 외부(자체 DB)에 저장되지 않는 기능이 존재
 - Privacy Mode를 사용하면 사용자의 INPUT 코드를 자체 vector DB에 저장하지 않음
 - 하지만 LLM의 인풋으로는 사용자의 코드와 데이터가 들어가기 때문에 의미가 없음
 - 해결 방안 LOCAL LLM을 활용



Cursor - Privacy mode

- Local LLM을 사용하는 방법 가능
 - Local LLM : 내 컴퓨터 가상환경에서 작동하는 LLM
 - OLLAMA: 로컬 가상환경에서 LLM을 실행하는 툴



감사합니다
