

LangChain를 활용한 간단한 챗봇 만들어보기

In [8]: `!pip install -Uq langchain_core langchain_openai langchain`

```
0.0/1.0 MB ? eta -:--:--
0.3/1.0 MB 9.0 MB/s eta 0:00:01
1.0/1.0 MB 18.1 MB/s eta 0:00:01
1.0/1.0 MB 12.7 MB/s eta 0:00:00
```

API_Key 인증

```
In [2]: import os
from openai import OpenAI

def init_api():
    with open("chatgpt_kict2409.env") as env:
        for line in env:
            key, value = line.strip().split("=")
            os.environ[key] = value

init_api()
# client = OpenAI(api_key = os.environ.get("API_KEY"))
os.environ["OPENAI_API_KEY"] = os.environ.get("API_KEY")
```

```
In [5]: from langchain_core.runnables import RunnablePassthrough
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser
from langchain_openai import ChatOpenAI
```

```
In [18]: # 프롬프트 템플릿 생성
# 템플릿은 system과 human으로 구성되어 있음.
prompt_template = ChatPromptTemplate.from_messages([
    ("system", "안녕하세요. 친구처럼 대화하고 이모티콘을 사용해 주세요 😊"),
    ("human", "{input_text}")
])

# LangChain 체인 생성
# RunnablePassthrough()를 쓰지 않으면 human의 변수명은 user_input으로 고정 사용해
# RunnablePassthrough()를 쓸 경우, human의 사용자 입력 변수의 이름을 "input_text"
# | prompt_template : 생성한 것을 prompt_template 체인에 연결
# | ChatOpenAI() : OpenAI의 채팅 모델을 체인에 연결
# | StrOutputParser() : AI의 응답을 문자열로 파싱 - 모델의 출력을 사람이 읽을 수
chain = {"input_text": RunnablePassthrough()} | prompt_template | ChatOpenAI() |

# 대화 루프 시작
while True:
    message = input("YOU :(quit : 종료) ")
    if message.lower() == "quit":
        break

    print(" AI : ", end="", flush=True)

    for response in chain.stream(message):
        print(response, end="", flush=True)
    print()
```

YOU :(quit : 종료) quit

실습 및 도전 과제

- [실습 1] Prompt 메시지를 변경해 보자. 여행용 챗봇 만들어보기
- [실습 2] Prompt 메시지로 할루시네이션을 없애는 프롬프트를 작성해 보자.
- [도전 1] 이전의 대화를 기능을 추가하는 것을 만들어보자.

Prompt 엔지니어링

```
In [26]: from langchain_core.runnables import RunnablePassthrough
from langchain_core.prompts import ChatPromptTemplate
from langchain_core.output_parsers import StrOutputParser
from langchain_openai import ChatOpenAI

# 할루시네이션 방지 시스템 메시지
prompt_template = ChatPromptTemplate.from_messages([
    ("system",
     "안녕하세요. 대화할 때 정확하고 신뢰할 수 있는 정보만 제공해 주세요. "
     "잘 모르는 내용은 추측하지 말고 '모르겠습니다'라고 답변해 주세요. 😊"),
    ("human", "{input_text}")
])

# LangChain 체인 생성
chain = {"input_text": RunnablePassthrough()} | prompt_template | ChatOpenAI()

# 대화 루프 시작
while True:
    message = input("YOU :(quit : 종료) ")
    if message.lower() == "quit":
        break

    print(" AI : ", end="", flush=True)

    for response in chain.stream(message):
        print(response, end="", flush=True)
    print()
```

YOU :(quit : 종료) 안녕!

AI : 안녕하세요! 무엇을 도와드릴까요? 😊

YOU :(quit : 종료) 너가 모르는 내용이 나오면 어떻게 할거니?

AI : 모르는 내용이 나오면 "모르겠습니다"라고 솔직하게 답변할 거예요. 다른 도움이 필요하시면 다른 질문을 해주세요. 😊

YOU :(quit : 종료) quit