

-
- 출처: LangChain 공식 문서 또는 해당 교재명
 - 원본 URL: <https://smith.langchain.com/hub/teddynote/summary-stuff-documents>
-

✓ ConversationBufferMemory

- 대화 내용을 그대로 통째로 저장하는 메모리
- 가장 기본적인 형태 의 메모리
- 역할:
 - 이전 대화 내용을 그대로 기억해서 필요한 경우 다시 불러오는 역할
 - 즉 통째로 녹화된 비디오 테이프
 - 이 메모리는 메시지를 저장한 다음 변수에 메시지를 추출할 수 있게 해줌
 - 먼저 문자열로 추출 할 수 있음

```
from langchain.memory import ConversationBufferMemory
```

```
memory = ConversationBufferMemory()  
memory
```

- 셀 출력 = 오류 메시 출력됨

```
/var/folders/h3/l7wnkv352kqftv0t8ctl2ld40000gn/T/ipykernel\_89650/2223904900.py:1: LangChainDeprecationWarning:  
memory = ConversationBufferMemory()
```

-
- 의미: 랭체인에서 더 이상 사용하지 않을 예정인 기능에 대한 경고
 - 첫번째 줄의 의미

```
/var/folders/h3/l7wnkv352kqftv0t8ctl2ld40000gn/T/ipykernel\_89650/2223904900.py:1: LangChainDeprecationWarn
```

- 메모리 기능이 새로운 방식으로 변경되었으니, 위 주소의 마이그레이션 가이드를 참고하여 코드를 수정하라는 의미
-

- 두번째 줄의 의미

```
memory = ConversationBufferMemory()
```

- 현재 사용하고 있는 ConversationBufferMemory() 코드가 곧 지원 중단될 예정이라는 의미
-

- memory에 대한 출력값

```
ConversationBufferMemory(chat_memory=InMemoryChatMessageHistory(messages=[]))
```

```
memory.save_context(
    inputs={
        "human": "안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?"
    },
    outputs={
        "ai": "안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?"
    },
)
```

memory

- 셀 출력

```
ConversationBufferMemory(chat_memory=InMemoryChatMessageHistory(messages=[HumanMessage(content='안녕하세요, |
```

```
print(type(memory)) # <class 'langchain.memory.buffer.ConversationBufferMemory'>
```

가독성 좋게 출력해보기

```
from langchain.schema import HumanMessage, AIMessage
```

```
def pretty_print_memory(memory):
    # InMemoryChatMessageHistory.messages 에 저장된 HumanMessage/AIMessage 들
    messages = memory.chat_memory.messages
```

```
print("📖 대화 히스토리\n")
for i, msg in enumerate(messages, start=1):
    # 메시지 타입에 따라 역할(role) 지정
    if isinstance(msg, HumanMessage):
        role = "사용자"
        prefix = "👤 "
    elif isinstance(msg, AIMessage):
        role = "어시스턴트"
        prefix = "🤖 "
    else:
        role = msg.__class__.__name__
        prefix = ""
```

```
    # 출력
    print(f"{i:>2}. {prefix} {role}:")
    # 개행문자(\n) 포함된 컨텐츠도 잘 보이도록
    for line in msg.content.split("\n"):
        print("    ", line)
    print()
```

호출 예시

```
pretty_print_memory(memory)
```

- 셀 출력

📖 대화 히스토리

1. 👤 사용자:

안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?

2. 🤖 어시스턴트:

안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?

- memory의 load_memory_variables({}) 함수 = 메시지 히스토리를 반환

'history' 키에 저장된 대화 기록 확인해보기

```
print(memory.load_memory_variables({})["history"])
```

- 셀 출력

Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?

AI: 안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?

- **save_context(inputs, outputs)** 메서드 사용 → 대화 기록 저장할 수 있음

- **save_context(inputs, outputs)** 메서드 = **inputs**와 **outputs** 두 개의 인자 필요

- **inputs** = 사용자의 입력

- **outputs** = AI의 출력

- **history** 키 = 대화기록 저장

- **load_memory_variables** 메서드 사용 → 저장된 대화 기록 확인 가능

inputs: dictionary(key: "human" or "ai", value: 질문)

outputs: dictionary(key: "ai" or "human", value: 답변)

```
memory.save_context(
    inputs={"human": "네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?"},
    outputs={
        "ai": "감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행하겠습니다."
    },
)
```

2개의 대화 저장하기

```
memory.save_context(
    inputs={"human": "사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?"},
    outputs={
        "ai": "업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로 발송된 인증번호를 입력해 주시면 됩니다."
    },
)

memory.save_context(
    inputs={"human": "인증번호를 입력했습니다. 계좌 개설은 이제 어떻게 하나요?"},
    outputs={
        "ai": "본인 인증이 완료되었습니다. 이제 원하시는 계좌 종류를 선택하고 필요한 정보를 입력해 주세요. 예금 종류, 통화 종류 등을 선택할 수 있습니다."
    },
)
```

history에 저장된 대화 기록 확인하기

```
print(memory.load_memory_variables({})["history"])
```

- 셀 출력

Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?

AI: 안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?

Human: 네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?

AI: 감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행하겠습니다.

Human: 사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?

AI: 업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로 발송된 인증번호를 입력해 주시면 됩니다.

Human: 인증번호를 입력했습니다. 계좌 개설은 이제 어떻게 하나요?

AI: 본인 인증이 완료되었습니다. 이제 원하시는 계좌 종류를 선택하고 필요한 정보를 입력해 주세요. 예금 종류, 통화 종류 등을 선택할 수 있습니다.

- return_messages=True로 설정 → **HumanMessage** 와 **AIMessage** 객체 반환

```
memory = ConversationBufferMemory(return_messages=True)
```

```
# 첫번째 대화
```

```
memory.save_context(
    inputs={
        "human": "안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?"
    },
    outputs={
        "ai": "안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?"
    },
)
```

```
# 두번째 대화
```

```
memory.save_context(
    inputs={"human": "네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?"},
    outputs={
        "ai": "감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행하겠습니다."
    },
)
```

```
# 세번째 대화
```

```
memory.save_context(
    inputs={"human": "사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?"},
    outputs={
        "ai": "업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로 발송된 인증번호를 입력해 주시면 됩니다."
    },
)
```

```
# history에 저장된 대화 기록 확인하기
```

```
print(memory.load_memory_variables({})["history"])
```

- 셀 출력

```
[HumanMessage(content='안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?', additional_kwargs={},
```

```
# 보기 좋게 출력해보기
```

```
from langchain.schema import HumanMessage, AIMessage
```

```
def pretty_print_history(history):
```

```
    """
```

```
    history: List[HumanMessage | AIMessage]
```

```
    HumanMessage는 사용자, AIMessage는 어시스턴트로 구분해 예쁘게 출력합니다.
```

```
    """
```

```
    print("📌 대화 히스토리\n")
```

```
    for idx, msg in enumerate(history, start=1):
```

```
        # 역할과 이모지 결정
```

```
        if isinstance(msg, HumanMessage):
```

```
            role, emoji = "사용자", "👤"
```

```
        elif isinstance(msg, AIMessage):
```

```
            role, emoji = "어시스턴트", "🤖"
```

```
        else:
```


```
            role, emoji = msg.__class__.__name__, ""
```







```
# 헤더 출력
print(f"{idx:02d}. {emoji} {role}:")

# 본문 출력 (줄바꿈 후 들여쓰기 유지)
for line in msg.content.split("\n"):
    print("    " + line)
print() # 메시지 간 빈 줄
```

```
# 실제로 호출하는 부분
history = memory.load_memory_variables({})["history"]
pretty_print_history(history)
```

- 셀 출력

 대화 히스토리

01.  사용자:
안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?
02.  어시스턴트:
안녕하세요! 계좌 개설을 원하신다니 기쁩니다. 먼저, 본인 인증을 위해 신분증을 준비해 주시겠어요?
03.  사용자:
네, 신분증을 준비했습니다. 이제 무엇을 해야 하나요?
04.  어시스턴트:
감사합니다. 신분증 앞뒤를 명확하게 촬영하여 업로드해 주세요. 이후 본인 인증 절차를 진행하겠습니다.
05.  사용자:
사진을 업로드했습니다. 본인 인증은 어떻게 진행되나요?
06.  어시스턴트:
업로드해 주신 사진을 확인했습니다. 이제 휴대폰을 통한 본인 인증을 진행해 주세요. 문자로 발송된 인증번호를 입력해 주시면 됩니다.

```
from langchain_core.prompts import ChatPromptTemplate

ChatPromptTemplate.from_messages(
    [
        ("system", "당신은 친절한 AI 봇입니다."),
        ("human", "대한민국의 수도는 어디야?"),
    ]
)
```

- 셀 출력

```
ChatPromptTemplate(input_variables=[], input_types={}, partial_variables={}, messages=[SystemMessagePromptTemplate.from_text('You are a helpful assistant.')]


```

Chain에 적용

```
# 환경변수 처리 및 클라어트 생성
from langsmith import Client
from langchain.prompts import PromptTemplate
from langchain.prompts import ChatPromptTemplate
from langchain.chains import ConversationChain
```

```

import os
import json

# 클라이언트 생성
api_key = os.getenv("LANGSMITH_API_KEY")
client = Client(api_key=api_key)

# LangSmith 추적 설정하기 (https://smith.langchain.com)
# LangSmith 추적을 위한 라이브러리 임포트
from langsmith import traceable

# LangSmith 환경 변수 확인

print("\n--- LangSmith 환경 변수 확인 ---")
langchain_tracing_v2 = os.getenv('LANGCHAIN_TRACING_V2')
langchain_project = os.getenv('LANGCHAIN_PROJECT')
langchain_api_key_status = "설정됨" if os.getenv('LANGCHAIN_API_KEY') else "설정되지 않음"
langchain_org = "설정됨" if os.getenv('LANGCHAIN_ORGANIZATION') else "설정되지 않음"

if langchain_tracing_v2 == "true" and os.getenv('LANGCHAIN_API_KEY') and os.getenv('LANGCHAIN_ORGANIZATION'):
    print(f"✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='{langchain_tracing_v2}')"
    print(f"✅ LangSmith 프로젝트: '{langchain_project}'")
    print(f"✅ LangSmith API Key: {langchain_api_key_status}")
    print(" -> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.")
else:
    print("❌ LangSmith 추적이 완전히 활성화되지 않았습니다. 다음을 확인하세요:")
    if langchain_tracing_v2 != "true":
        print("  - LANGCHAIN_TRACING_V2가 'true'로 설정되어 있지 않습니다.")
    if not os.getenv('LANGCHAIN_API_KEY'):
        print("  - LANGCHAIN_API_KEY가 설정되어 있지 않습니다.")
    if not langchain_project:
        print("  - LANGCHAIN_PROJECT가 설정되어 있지 않습니다.")

```

"@traceable" 주석은 허용되지 않습니다. 허용되는 값은 다음과 같습니다. [@param, @title, @markdown]

• 셀 출력

```

--- LangSmith 환경 변수 확인 ---
✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='true')
✅ LangSmith 프로젝트: 'LangChain-practice'
✅ LangSmith API Key: 설정됨
-> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.

```

```

from langchain_google_genai import ChatGoogleGenerativeAI

```

```

# LLM 생성
gemini_lc = ChatGoogleGenerativeAI(
    model="gemini-2.5-flash-lite",
    temperature=0.7,
    max_output_tokens=4096,
)

```

```

from langchain.chains import ConversationChain
from langchain.memory import ConversationBufferMemory

```

```

# ConversationChain 생성하기

```

```

conversation = ConversationChain(
    llm=gemini_lc,
    memory=ConversationBufferMemory(),
)

```

```

# ConversationBufferMemory 사용

```

• 셀 출력

- ConversationChain을 사용하여 대화 진행하기

```
# 대화를 시작합니다.
```

```
response = conversation.predict(
    input="안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?"
)
print(response)
```

- 셀 출력 (3.9s)

안녕하세요! 비대면으로 은행 계좌를 개설하고 있으시다니, 정말 편리한 시대에 살고 있다는 생각이 드네요. 제가 알고 있는 정보에 따르면, 요즘은 비대면으로 은행 계좌를 개설할 수 있습니다. 어떤 은행의 계좌를 개설하고 싶으신가요? 은행마다 약간씩 절차가 다를 수 있지만, 일반적인 과정은 다음과 같습니다.

1. ****은행 앱 또는 웹사이트 접속:**** 먼저, 이용하시려는 은행의 모바일 앱을 다운로드하시거나 공식 웹사이트에 접속하셔야 합니다. 대부분의 은행은 모바일 앱과 웹사이트를 제공합니다.
2. ****비대면 계좌 개설 메뉴 선택:**** 앱이나 웹사이트에 접속하시면 "계좌 개설", "비대면 계좌 개설", "신규 계좌 신청" 등과 같은 메뉴를 찾으실 수 있습니다.
3. ****본인 인증:**** 이 단계가 가장 중요합니다. 비대면으로 계좌를 개설하기 위해서는 본인임을 확실하게 인증해야 하거든요. 일반적으로 다음과 같은 인증 방법을 사용합니다.
 - * ****휴대폰 본인 인증:**** 가입하신 통신사를 통해 휴대폰으로 인증번호를 받아 입력하는 방식입니다.
 - * ****신분증 촬영:**** 주민등록증이나 운전면허증과 같은 실물 신분증을 촬영하여 제출하는 방식입니다. 이때, 신분증의 정보가 명확하게 나타나도록 촬영해야 합니다.
 - * ****영상 통화 또는 녹화:**** 일부 은행에서는 추가적인 본인 확인을 위해 상담원과 영상 통화를 하거나, 본인 인증 절차를 녹화하는 방식을 사용합니다.
4. ****정보 입력:**** 계좌 개설을 위해 필요한 정보들을 입력해야 합니다. 보통 이름, 주민등록번호, 주소, 연락처 등 기본적인 개인 정보와 함께 계좌 종류, 입출금 통장, 예금 통장, 적금 통장 등 다양한 상품이 있습니다. 상품별로 세부 조건이 다르니 꼭 확인하세요.
5. ****상품 선택:**** 어떤 종류의 계좌를 개설할지 선택하셔야 합니다. 입출금 통장, 예금 통장, 적금 통장 등 다양한 상품이 있습니다. 상품별로 세부 조건이 다르니 꼭 확인하세요.
6. ****약관 동의:**** 계좌 개설에 필요한 각종 약관과 개인정보 수집 및 이용 동의 등을 꼼꼼히 확인하시고 동의하시면 됩니다.
7. ****계좌 개설 완료:**** 모든 절차가 정상적으로 완료되면, 비대면으로 개설된 계좌 정보와 함께 OTP(일회용 비밀번호 생성기)나 보안카드 발급을 받으실 수 있습니다.

****주의하실 점:****

- * ****실명 확인:**** 비대면 계좌 개설 시에는 반드시 본인 명의의 실명 확인된 휴대폰이나 신분증이 필요합니다.
- * ****미성년자:**** 미성년자의 경우, 법정대리인의 동의나 별도의 절차가 필요할 수 있습니다.
- * ****기존 고객:**** 이미 해당 은행에 계좌가 있으신 경우, 절차가 더 간소화될 수도 있습니다.

혹시 특정 은행을 염두에 두고 계신가요? 은행 이름을 알려주시면 해당 은행의 비대면 계좌 개설 특징에 대해 더 자세히 안내해 드릴 수도 있습니다.

```
print(type(response)) # <class 'str'>
```

- 이전의 대화 기록을 기억하고 있는지 확인해보기

```
# 이전 대화내용을 볼렛포인트로 정리해 달라는 요청 보내보기
```

```
response = conversation.predict(
    input="이전 답변을 볼렛포인트 형식으로 정리하여 알려주세요."
)
print(response)
```

- 셀 출력 (1.7s)

네, 이전 답변 내용을 요청하신 볼렛포인트 형식으로 깔끔하게 정리해 드리겠습니다!

****비대면 은행 계좌 개설 절차 (일반적인 과정):****

- * ****앱/웹사이트 접속:****
 - * 이용하려는 은행의 모바일 앱 다운로드 또는 공식 웹사이트 접속
- * ****메뉴 선택:****
 - * "계좌 개설", "비대면 계좌 개설", "신규 계좌 신청" 등 관련 메뉴 선택
- * ****본인 인증:****
 - * ****휴대폰 본인 인증:**** 통신사를 통한 인증번호 입력
 - * ****신분증 촬영:**** 주민등록증, 운전면허증 등 실물 신분증 촬영 제출 (밝은 곳에서 명확하게)
 - * ****영상 통화/녹화:**** 일부 은행에서 상담원과의 영상 통화 또는 절차 녹화 진행
- * ****정보 입력:****
 - * 이름, 주민등록번호, 주소, 연락처 등 기본 개인 정보 입력
 - * 계좌 사용 목적, 직업 등 추가 정보 요청 가능
- * ****상품 선택:****
 - * 입출금 통장, 예금 통장, 적금 통장 등 원하는 계좌 상품 선택 (금리, 혜택 비교)
- * ****약관 동의:****
 - * 관련 약관 및 개인정보 수집/이용 동의 꼼꼼히 확인 후 동의
- * ****계좌 개설 완료:****
 - * 개설된 계좌 정보 확인
 - * OTP 또는 보안카드 발급 관련 안내 확인 (인터넷/모바일 뱅킹 필수)

****비대면 계좌 개설 시 주의사항:****

- * ****실명 확인:**** 본인 명의의 실명 확인된 휴대폰 또는 신분증 필수
- * ****미성년자:**** 법정대리인 동의 또는 별도 절차 필요 가능성 있음
- * ****기존 고객:**** 해당 은행 기존 고객의 경우 절차 간소화될 수 있음

이 외에 특정 은행에 대해 더 궁금하신 점이 있다면 언제든지 다시 물어봐 주세요! 😊

▽ LangChain v0.2.0부터 메모리 사용 방식 변경

- 바뀐 버전으로 다시 사용해보기

```
from langsmith import Client
from langchain.prompts import PromptTemplate
from langchain.prompts import ChatPromptTemplate
from langchain.chains import ConversationChain
from langchain.memory import ConversationBufferMemory
from langchain_core.prompts import ChatPromptTemplate, MessagesPlaceholder
```

```
import os
import json
```

```
# 클라이언트 생성
api_key = os.getenv("LANGSMITH_API_KEY")
client = Client(api_key=api_key)
```

```
# LangSmith 추적 설정하기 (https://smith.langchain.com)
# LangSmith 추적을 위한 라이브러리 импорт
from langsmith import traceable
```

```
# LangSmith 환경 변수 확인
```

"@traceable" 주석은 허용되지 않습니다. 허용되는 값은 다음과 같습니다. `[@param, @title, @markdown]`




```

print("\n--- LangSmith 환경 변수 확인 ---")
langchain_tracing_v2 = os.getenv('LANGCHAIN_TRACING_V2')
langchain_project = os.getenv('LANGCHAIN_PROJECT')
langchain_api_key_status = "설정됨" if os.getenv('LANGCHAIN_API_KEY') else "설정되지 않음"
org = "설정됨" if os.getenv('LANGCHAIN_ORGANIZATION') else "설정되지 않음"

if langchain_tracing_v2 == "true" and os.getenv('LANGCHAIN_API_KEY') and os.getenv('LANGCHAIN_ORGANIZATION'):
    print(f"✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='true')")
    print(f"✅ LangSmith 프로젝트: '{langchain_project}'")
    print(f"✅ LangSmith API Key: {langchain_api_key_status}")
    print(" -> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.")
else:
    print("❌ LangSmith 추적이 완전히 활성화되지 않았습니다. 다음을 확인하세요.")
    if langchain_tracing_v2 != "true":
        print(f" - LANGCHAIN_TRACING_V2가 'true'로 설정되어 있지 않습니다.")
    if not os.getenv('LANGCHAIN_API_KEY'):
        print(" - LANGCHAIN_API_KEY가 설정되어 있지 않습니다.")
    if not langchain_project:
        print(" - LANGCHAIN_PROJECT가 설정되어 있지 않습니다.")

```

- 셀 출력

```

--- LangSmith 환경 변수 확인 ---
✅ LangSmith 추적 활성화됨 (LANGCHAIN_TRACING_V2='true')
✅ LangSmith 프로젝트: 'LangChain-prantice'
✅ LangSmith API Key: 설정됨
-> 이제 LangSmith 대시보드에서 이 프로젝트를 확인해 보세요.

```

```

from langchain_google_genai import ChatGoogleGenerativeAI

```

```

# LLM 생성

```

```

gemini_lc = ChatGoogleGenerativeAI(
    model="gemini-2.5-flash-lite",
    temperature=0.7,
    max_output_tokens=4096,
)

```

```

# ConversationChain 생성하기

```

```

# memory_key와 input_key를 명시적으로 설정해야 함

```

```

from langchain.chains import ConversationChain
from langchain.memory import ConversationBufferMemory
from langchain_core.prompts import ChatPromptTemplate, MessagesPlaceholder

```

```

conversation = ConversationChain(
    llm=gemini_lc,

    # (변경) chat_history를 memory_key로 명시적 설정
    # memory=ConversationBufferMemory(memory_key="chat_history"), # 오류 발생
    # (수정) memory_key를 'history'로 설정하여 ConversationChain의 기본 프롬프트와 일치
    memory=ConversationBufferMemory(
        return_messages=True,
        memory_key="history"),

    # 추론 과정을 확인을 위해 True로 설정
    verbose=True
)

```

```

# ANSI 코드를 사용하여 텍스트에 색상을 추가하는 함수

```

```

def color_text(text, color_code):
    return f"\033[{color_code}m{text}\033[0m"

```

첫번째 대화

```
response = conversation.predict(input="안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?")
print("AI:", response)
```

- 실수로 같은 질문을 두 번 입력함 (2.3s)

-  ANSI 형식으로 나와서 html로 시도했으나 잘 되지 않음

```
<span style="color: white;"><b>Entering new ConversationChain chain...</b></span>
<span style="color: white;"><b>Prompt after formatting:</b></span>
<span style="color: green;"><b>The following is a friendly conversation between a human and an AI. Th

<span style="color: green;"><b>Current conversation:</b></span>
<span style="color: green;"><b>Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요</b></span>
<span style="color: green;"><b>AI: 안녕하세요! 비대면으로 은행 계좌를 개설하시고 싶으시다니, 정말 편리한 시대에 살고 있죠!

<span style="color: green;"><b>일반적으로 비대면 계좌 개설은 크게 두 가지 방법으로 진행될 수 있어요.</b></span>
```

1. **은행 자체 모바일 앱을 이용하는 방법:**

- * 대부분의 은행들이 자체적으로 개발한 모바일 뱅킹 앱을 가지고 있어요. 이 앱을 통해 계좌 개설 절차를 진행하는 것이 가장 일반
- * **준비물:**
 - * **본인 명의 스마트폰:** 안드로이드나 iOS 운영체제를 사용하는 스마트폰이면 됩니다.
 - * **신분증:** 주민등록증 또는 운전면허증이 필요해요. 신분증 촬영 시 빛 반사 없이 선명하게 나오도록 주의해야 합니다.
 - * **본인 명의 휴대폰:** 본인 인증을 위해 필요합니다.
 - * **기존에 보유하고 있는 다른 은행 계좌 (선택 사항):** 본인 확인 절차에서 신분증 인증 외에, 이미 보유하고 있는 다른
- * **절차:**
 1. **은행 앱 다운로드:** 이용하시려는 은행의 모바일 앱을 스마트폰에 설치합니다. (예: KB국민은행 스타뱅킹, 신한은행 S
 2. **계좌 개설 메뉴 선택:** 앱을 실행한 후, 보통 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등의 메뉴를 찾습니다
 3. **상품 선택:** 개설하시려는 계좌 종류 (입출금 통장, 저금 통장 등)를 선택합니다. 보통 입출금 통장을 먼저 개설하는 ?
 4. **본인 확인:**
 - * **신분증 촬영:** 안내에 따라 신분증을 촬영합니다.
 - * **영상 통화 또는 타행 계좌 인증:** 은행에 따라서는 상담원과 짧은 영상 통화를 하거나, 기존에 가지고 계신 다른
 5. **정보 입력:** 이름, 생년월일, 주소, 직업 등 필요한 정보를 입력합니다.
 6. **약관 동의:** 상품 이용 약관 및 개인정보 수집/이용 동의 등을 진행합니다.
 7. **계좌 개설 완료:** 모든 절차가 정상적으로 완료되면 계좌가 개설됩니다.

2. **정부 지원 서비스 (SCINet 등)를 이용하는 방법:**

- * 일부 은행에서는 정부에서 제공하는 신원확인 시스템과 연계하여 비대면 계좌 개설을 지원하기도 합니다. 하지만 이 방법은 아직 보

어떤 은행을 이용하실 예정이신가요? 혹시 특정 은행을 염두에 두고 계시다면, 해당 은행의 이름이나 앱 이름을 알려주시면 제가 더 구체

Human: 준비물을 알려주세요.

AI: 네, 비대면 계좌 개설을 위해 필요한 준비물들을 다시 한번 자세히 알려드릴게요! 😊

가장 일반적인 방법인 **은행 자체 모바일 앱을 이용한 비대면 계좌 개설**을 기준으로 말씀드리겠습니다.

3. **본인 명의 스마트폰:**

- * 이것은 필수입니다! 안드로이드 스마트폰이든 아이폰이든 상관없습니다.
- * 스마트폰으로 은행 앱을 다운로드하고, 카메라로 신분증을 촬영하며, 본인 인증 절차를 진행하기 때문에 꼭 필요해요.
- * **주의할 점:** 스마트폰의 카메라 기능이 정상적으로 작동해야 합니다.

4. **신분증:**

- * **주민등록증** 또는 **운전면허증**이 필요합니다.
- * **가장 중요한 부분은 신분증을 촬영할 때입니다.**
 - * **빛 반사를 최소화하세요.** 형광등 불빛이나 창문 빛이 신분증에 직접적으로 반사되면 인식이 잘 안 될 수 있습니다.
 - * **신분증의 모든 정보가 선명하게 보이도록 촬영해야 합니다.** 글자가 흐릿하거나 잘리면 안 돼요.
 - * **신분증의 네 모서리가 모두 나오도록 촬영**하는 것이 좋습니다.
 - * **신분증을 평평한 곳에 놓고 촬영**하면 더 깔끔하게 찍을 수 있어요.

5. ****본인 명의 휴대폰:****

- * 이것 역시 필수입니다! 계좌 개설 과정에서 본인 확인을 위해 휴대폰으로 인증번호를 받거나, 경우에 따라서는 은행 상담원과의 영
- * ****알뜰폰(MVNO)을 사용하시는 경우, 일부 은행에서는 비대면 계좌 개설이 제한될 수도 있습니다.**** 이 점은 이용하시려는 은행

6. ****기존에 보유하고 있는 다른 은행 계좌 (본인 확인용, 선택 사항):****

- * 이것은 ****모든 은행에서 필수로 요구하는 것은 아니지만, 많은 은행에서 사용되는 추가적인 본인 확인 방법****입니다.
- * 신분증 촬영만으로는 본인 확인이 부족하다고 판단될 경우, ****이전에 사용하시던 다른 은행 계좌로 '1원' 또는 '소액'을 입금****
- * 따라서, ****본인 명의로 개설되어 있고 현재 사용 중인 다른 은행 계좌가 하나쯤 있다면**** 비대면 계좌 개설이 더 원활하게 진행될

요약하자면, ****스마트폰, 신분증, 그리고 본인 명의 휴대폰**** 이 세 가지가 가장 중요하다고 보시면 됩니다. 😊

혹시 이 중에서 준비하시면서 어려운 점이 있으시거나, 특정 은행에서 요구하는 사항이 궁금하시면 언제든지 다시 물어봐 주세요!

• 첫번째 셀 출력 (3.7s)

```
<span style="color: green;"><b>Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?</b></spa
<span style="color: green;"><b>AI:</b></span>
<span style="color: green;"><b>Finished chain.</b></span>
<span style="color: green;"><b>AI: 안녕하세요! 비대면으로 은행 계좌를 개설하시고 싶으시다니, 정말 편리한 시대에 살고 있죠! 제가
```

일반적으로 비대면 계좌 개설은 크게 두 가지 방법으로 진행될 수 있어요.

1. ****은행 자체 모바일 앱을 이용하는 방법:****

- * 대부분의 은행들이 자체적으로 개발한 모바일 뱅킹 앱을 가지고 있어요. 이 앱을 통해 계좌 개설 절차를 진행하는 것이 가장 일반적입니다
- * ****준비물:****
 - * ****본인 명의 스마트폰:**** 안드로이드나 iOS 운영체제를 사용하는 스마트폰이면 됩니다.
 - * ****신분증:**** 주민등록증 또는 운전면허증이 필요해요. 신분증 촬영 시 빛 반사 없이 선명하게 나오도록 주의해야 합니다.
 - * ****본인 명의 휴대폰:**** 본인 인증을 위해 필요합니다.
 - * ****기존에 보유하고 있는 다른 은행 계좌 (선택 사항):**** 본인 확인 절차에서 신분증 인증 외에, 이미 보유하고 있는 다른 은행
- * ****절차:****
 1. ****은행 앱 다운로드:**** 이용하시려는 은행의 모바일 앱을 스마트폰에 설치합니다. (예: KB국민은행 스타뱅킹, 신한은행 SOL, 우
 2. ****계좌 개설 메뉴 선택:**** 앱을 실행한 후, 보통 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등의 메뉴를 찾습니다.
 3. ****상품 선택:**** 개설하시려는 계좌 종류 (입출금 통장, 적금 통장 등)를 선택합니다. 보통 입출금 통장을 먼저 개설하는 경우가
 4. ****본인 확인:****
 - * ****신분증 촬영:**** 안내에 따라 신분증을 촬영합니다.
 - * ****영상 통화 또는 타행 계좌 인증:**** 은행에 따라서는 상담원과 짧은 영상 통화를 하거나, 기존에 가지고 계신 다른 은행
 5. ****정보 입력:**** 이름, 생년월일, 주소, 직업 등 필요한 정보를 입력합니다.
 6. ****약관 동의:**** 상품 이용 약관 및 개인정보 수집/이용 동의 등을 진행합니다.
 7. ****계좌 개설 완료:**** 모든 절차가 정상적으로 완료되면 계좌가 개설됩니다.

2. ****정부 지원 서비스 (SCINet 등)를 이용하는 방법:****


- * 일부 은행에서는 정부에서 제공하는 신원확인 시스템과 연계하여 비대면 계좌 개설을 지원하기도 합니다. 하지만 이 방법은 아직 보편적이

****어떤 은행을 이용하실 예정이신가요?**** 혹시 특정 은행을 염두에 두고 계시다면, 해당 은행의 이름이나 앱 이름을 알려주시면 제가 더 구체적인

두 번째 대화 (이전 내용을 기억하는지 확인)

```
response = conversation.predict(input="준비물을 알려주세요.")
print("AI:", response)
```

• 셀 출력 (3.1s)

-  두번째 출력화면

****> Entering new ConversationChain chain...****

****Prompt after formatting:****

*****The following is a friendly conversation between a human and an AI. The AI is talkative and provides**

****Current conversation:****

****Human:** 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?**

****AI:** 안녕하세요! 비대면으로 은행 계좌를 개설하시고 싶으시다니, 정말 편리한 시대에 살고 있죠! 제가 아는 선에서 최대한 자세하게 안내

일반적으로 비대면 계좌 개설은 크게 두 가지 방법으로 진행될 수 있어요.

1. **은행 자체 모바일 앱을 이용하는 방법:**

- * 대부분의 은행들이 자체적으로 개발한 모바일 뱅킹 앱을 가지고 있어요. 이 앱을 통해 계좌 개설 절차를 진행하는 것이 가장 일반적
- * ****준비물:****
 - * ****본인 명의 스마트폰:**** 안드로이드나 iOS 운영체제를 사용하는 스마트폰이면 됩니다.
 - * ****신분증:**** 주민등록증 또는 운전면허증이 필요해요. 신분증 촬영 시 빛 반사 없이 선명하게 나오도록 주의해야 합니다.
 - * ****본인 명의 휴대폰:**** 본인 인증을 위해 필요합니다.
 - * ****기존에 보유하고 있는 다른 은행 계좌 (선택 사항):**** 본인 확인 절차에서 신분증 인증 외에, 이미 보유하고 있는 다른
- * ****절차:****
 1. ****은행 앱 다운로드:**** 이용하시려는 은행의 모바일 앱을 스마트폰에 설치합니다. (예: KB국민은행 스타뱅킹, 신한은행 S
 2. ****계좌 개설 메뉴 선택:**** 앱을 실행한 후, 보통 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등의 메뉴를 찾습니다.
 3. ****상품 선택:**** 개설하시려는 계좌 종류 (입출금 통장, 적금 통장 등)를 선택합니다. 보통 입출금 통장을 먼저 개설하는 것
 4. ****본인 확인:****
 - * ****신분증 촬영:**** 안내에 따라 신분증을 촬영합니다.
 - * ****영상 통화 또는 타행 계좌 인증:**** 은행에 따라서는 상담원과 짧은 영상 통화를 하거나, 기존에 가지고 계신 다른
 5. ****정보 입력:**** 이름, 생년월일, 주소, 직업 등 필요한 정보를 입력합니다.
 6. ****약관 동의:**** 상품 이용 약관 및 개인정보 수집/이용 동의 등을 진행합니다.
 7. ****계좌 개설 완료:**** 모든 절차가 정상적으로 완료되면 계좌가 개설됩니다.

2. **정부 지원 서비스 (SCINet 등)를 이용하는 방법:**

- * 일부 은행에서는 정부에서 제공하는 신원확인 시스템과 연계하여 비대면 계좌 개설을 지원하기도 합니다. 하지만 이 방법은 아직 보편

****어떤 은행을 이용하실 예정이신가요?**** 혹시 특정 은행을 염두에 두고 계시다면, 해당 은행의 이름이나 앱 이름을 알려주시면 제가 더 구체

****Human:** 준비물을 알려주세요.**

****AI:****

****> Finished chain.****

****AI:** 네, 비대면 계좌 개설을 위해 필요한 준비물들을 다시 한번 자세히 알려드릴게요! 😊**

가장 일반적인 방법인 ****은행 자체 모바일 앱을 이용한 비대면 계좌 개설****을 기준으로 말씀드리겠습니다.

3. **본인 명의 스마트폰:**

- * 이것은 필수입니다! 안드로이드 스마트폰이든 아이폰이든 상관없습니다.
- * 스마트폰으로 은행 앱을 다운로드하고, 카메라로 신분증을 촬영하며, 본인 인증 절차를 진행하기 때문에 꼭 필요해요.
- * ****주의할 점:**** 스마트폰의 카메라 기능이 정상적으로 작동해야 합니다.

4. **신분증:**

- * ****주민등록증**** 또는 ****운전면허증****이 필요합니다.
- * ****가장 중요한 부분은 신분증을 촬영할 때입니다.****
 - * ****빛 반사를 최소화하세요.**** 형광등 불빛이나 창문 빛이 신분증에 직접적으로 반사되면 인식이 잘 안 될 수 있습니다.
 - * ****신분증의 모든 정보가 선명하게 보이도록 촬영해야 합니다.**** 글자가 흐릿하거나 잘리면 안 돼요.
 - * ****신분증의 네 모서리가 모두 나오도록 촬영****하는 것이 좋습니다.
 - * ****신분증을 평평한 곳에 놓고 촬영****하면 더 깔끔하게 찍을 수 있어요.

5. **본인 명의 휴대폰:**

- * 이것 역시 필수입니다! 계좌 개설 과정에서 본인 확인을 위해 휴대폰으로 인증번호를 받거나, 경우에 따라서는 은행 상담원과의 영
- * ****알뜰폰(MVNO)을 사용하시는 경우,** 일부 은행에서는 비대면 계좌 개설이 제한될 수도 있습니다.** 이 점은 이용하시려는 은행

6. ****기존에 보유하고 있는 다른 은행 계좌 (본인 확인용, 선택 사항):****

- * 이것은 ****모든 은행에서 필수로 요구하는 것은 아니지만, 많은 은행에서 사용되는 추가적인 본인 확인 방법****입니다.
- * 신분증 촬영만으로는 본인 확인이 부족하다고 판단될 경우, ****이전에 사용하시던 다른 은행 계좌로 '1원' 또는 '소액'을 입금****
- * 따라서, ****본인 명의로 개설되어 있고 현재 사용 중인 다른 은행 계좌가 하나쯤 있다면**** 비대면 계좌 개설이 더 원활하게 진행될


요약하자면, ****스마트폰, 신분증, 그리고 본인 명의 휴대폰**** 이 세 가지가 가장 중요하다고 보시면 됩니다. 😊

혹시 이 중에서 준비하시면서 어려운 점이 있으시거나, 특정 은행에서 요구하는 사항이 궁금하시면 언제든지 다시 물어봐 주세요!

세 번째 대화

```
response = conversation.predict(input="볼렛포인트로 정리해서 알려주세요.")  
print("AI:", response)
```

• 셀 출력 (2.0s)

-  세번째 출력 화면

****Entering new ConversationChain chain...****

****Prompt after formatting:****

****The following is a friendly conversation between a human and an AI. The AI is talkative and provide**

****Current conversation:****

****Human:** 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?******

****AI:** 안녕하세요! 비대면으로 은행 계좌를 개설하시고 싶으시다니, 정말 편리한 시대에 살고 있죠! 제가 아는 선에서 최대한 자세하게 안내

일반적으로 비대면 계좌 개설은 크게 두 가지 방법으로 진행될 수 있어요.

1. ****은행 자체 모바일 앱을 이용하는 방법:****

- * 대부분의 은행들이 자체적으로 개발한 모바일 뱅킹 앱을 가지고 있어요. 이 앱을 통해 계좌 개설 절차를 진행하는 것이 가장 일반적
- * ****준비물:****
 - * ****본인 명의의 스마트폰:**** 안드로이드나 iOS 운영체제를 사용하는 스마트폰이면 됩니다.
 - * ****신분증:**** 주민등록증 또는 운전면허증이 필요해요. 신분증 촬영 시 빛 반사 없이 선명하게 나오도록 주의해야 합니다.
 - * ****본인 명의의 휴대폰:**** 본인 인증을 위해 필요합니다.
 - * ****기존에 보유하고 있는 다른 은행 계좌 (선택 사항):**** 본인 확인 절차에서 신분증 인증 외에, 이미 보유하고 있는 다른
- * ****절차:****
 1. ****은행 앱 다운로드:**** 이용하시려는 은행의 모바일 앱을 스마트폰에 설치합니다. (예: KB국민은행 스타뱅킹, 신한은행 S
 2. ****계좌 개설 메뉴 선택:**** 앱을 실행한 후, 보통 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등의 메뉴를 찾습니다
 3. ****상품 선택:**** 개설하시려는 계좌 종류 (입출금 통장, 적금 통장 등)를 선택합니다. 보통 입출금 통장을 먼저 개설하는 것
 4. ****본인 확인:****
 - * ****신분증 촬영:**** 안내에 따라 신분증을 촬영합니다.
 - * ****영상 통화 또는 타행 계좌 인증:**** 은행에 따라서는 상담원과 짧은 영상 통화를 하거나, 기존에 가지고 계신 다른
 5. ****정보 입력:**** 이름, 생년월일, 주소, 직업 등 필요한 정보를 입력합니다.
 6. ****약관 동의:**** 상품 이용 약관 및 개인정보 수집/이용 동의 등을 진행합니다.
 7. ****계좌 개설 완료:**** 모든 절차가 정상적으로 완료되면 계좌가 개설됩니다.

2. ****정부 지원 서비스 (SCINet 등)를 이용하는 방법:****

- * 일부 은행에서는 정부에서 제공하는 신원확인 시스템과 연계하여 비대면 계좌 개설을 지원하기도 합니다. 하지만 이 방법은 아직 보

****어떤 은행을 이용하실 예정이신가요?**** 혹시 특정 은행을 염두에 두고 계시다면, 해당 은행의 이름이나 앱 이름을 알려주시면 제가 더 구체

Human: 준비물을 알려주세요.

AI: 네, 비대면 계좌 개설을 위해 필요한 준비물들을 다시 한번 자세히 알려드릴게요! 😊

가장 일반적인 방법인 ****은행 자체 모바일 앱을 이용한 비대면 계좌 개설****을 기준으로 말씀드리겠습니다.

1. ****본인 명의의 스마트폰:****

- * 이것은 필수입니다! 안드로이드 스마트폰이든 아이폰이든 상관없습니다.
- * 스마트폰으로 은행 앱을 다운로드하고, 카메라로 신분증을 촬영하며, 본인 인증 절차를 진행하기 때문에 꼭 필요해요.
- * ****주의할 점:**** 스마트폰의 카메라 기능이 정상적으로 작동해야 합니다.

2. ****신분증:****

- * ****주민등록증**** 또는 ****운전면허증****이 필요합니다.
- * ****가장 중요한 부분은 신분증을 촬영할 때입니다.****
 - * ****빛 반사를 최소화하세요.**** 형광등 불빛이나 창문 빛이 신분증에 직접적으로 반사되면 인식이 잘 안 될 수 있습니다.
 - * ****신분증의 모든 정보가 선명하게 보이도록 촬영해야 합니다.**** 글자가 흐릿하거나 잘리면 안 돼요.
 - * ****신분증의 네 모서리가 모두 나오도록 촬영****하는 것이 좋습니다.
 - * ****신분증을 평평한 곳에 놓고 촬영****하면 더 깔끔하게 찍을 수 있어요.

3. ****본인 명의 휴대폰:****

- * 이것 역시 필수입니다! 계좌 개설 과정에서 본인 확인을 위해 휴대폰으로 인증번호를 받거나, 경우에 따라서는 은행 상담원과의 영
- * ****알뜰폰(MVNO)**을 사용하시는 경우, 일부 은행에서는 비대면 계좌 개설이 제한될 수도 있습니다.** 이 점은 이용하시려는 은행

4. ****기준에 보유하고 있는 다른 은행 계좌 (본인 확인용, 선택 사항):****

- * 이것은 ****모든 은행에서 필수로 요구하는 것은 아니지만, 많은 은행에서 사용되는 추가적인 본인 확인 방법****입니다.
- * 신분증 촬영만으로는 본인 확인이 부족하다고 판단될 경우, ****이전에 사용하시던 다른 은행 계좌로 '1원' 또는 '소액'을 입금****
- * 따라서, ****본인 명의로 개설되어 있고 현재 사용 중인 다른 은행 계좌가 하나쯤 있다면**** 비대면 계좌 개설이 더 원활하게 진행될

요약하자면, ****스마트폰, 신분증, 그리고 본인 명의 휴대폰**** 이 세 가지가 가장 중요하다고 보시면 됩니다. 😊

혹시 이 중에서 준비하시면서 어려운 점이 있으시거나, 특정 은행에서 요구하는 사항이 궁금하시면 언제든지 다시 물어봐 주세요!

****Human:** 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?**

****AI:** 안녕하세요! 비대면으로 은행 계좌를 개설하시고 싶으시다면, 정말 편리한 시대에 살고 있죠! 제가 아는 선에서 최대한 자세하게 안내해 드

일반적으로 비대면 계좌 개설은 크게 두 가지 방법으로 진행될 수 있어요.

1. ****은행 자체 모바일 앱을 이용하는 방법:****

- * 대부분의 은행들이 자체적으로 개발한 모바일 뱅킹 앱을 가지고 있어요. 이 앱을 통해 계좌 개설 절차를 진행하는 것이 가장 일반적입니다
- * ****준비물:****
 - * ****본인 명의 스마트폰:**** 안드로이드나 iOS 운영체제를 사용하는 스마트폰이면 됩니다.
 - * ****신분증:**** 주민등록증 또는 운전면허증이 필요해요. 신분증 촬영 시 빛 반사 없이 선명하게 나오도록 주의해야 합니다.
 - * ****본인 명의 휴대폰:**** 본인 인증을 위해 필요합니다.
 - * ****기준에 보유하고 있는 다른 은행 계좌 (선택 사항):**** 본인 확인 절차에서 신분증 인증 외에, 이미 보유하고 있는 다른 은행
- * ****절차:****
 1. ****은행 앱 다운로드:**** 이용하시려는 은행의 모바일 앱을 스마트폰에 설치합니다. (예: KB국민은행 스타뱅킹, 신한은행 SOL, 우
 2. ****계좌 개설 메뉴 선택:**** 앱을 실행한 후, 보통 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등의 메뉴를 찾습니다.
 3. ****상품 선택:**** 개설하시려는 계좌 종류 (입출금 통장, 적금 통장 등)를 선택합니다. 보통 입출금 통장을 먼저 개설하는 경우가
 4. ****본인 확인:****
 - * ****신분증 촬영:**** 안내에 따라 신분증을 촬영합니다.
 - * ****영상 통화 또는 타행 계좌 인증:**** 은행에 따라서는 상담원과 짧은 영상 통화를 하거나, 기존에 가지고 계신 다른 은행
 5. ****정보 입력:**** 이름, 생년월일, 주소, 직업 등 필요한 정보를 입력합니다.
 6. ****약관 동의:**** 상품 이용 약관 및 개인정보 수집/이용 동의 등을 진행합니다.
 7. ****계좌 개설 완료:**** 모든 절차가 정상적으로 완료되면 계좌가 개설됩니다.

2. ****정부 지원 서비스 (SCINet 등)를 이용하는 방법:****

- * 일부 은행에서는 정부에서 제공하는 신원확인 시스템과 연계하여 비대면 계좌 개설을 지원하기도 합니다. 하지만 이 방법은 아직 보편적이

****어떤 은행을 이용하실 예정이신가요?**** 혹시 특정 은행을 염두에 두고 계시다면, 해당 은행의 이름이나 앱 이름을 알려주시면 제가 더 구체적인

****Human:** 불렛포인트로 정리해서 알려주세요.**

****AI:****

****Finished chain.****

AI: 네, 비대면 계좌 개설 절차를 불렛 포인트로 깔끔하게 정리해 드릴게요! 😊

****비대면 은행 계좌 개설 절차 (모바일 앱 이용 시)****

- * ****1단계: 준비물 확인 및 확보****
 - * 본인 명의 스마트폰 (안드로이드 또는 iOS)
 - * 신분증 (주민등록증 또는 운전면허증) - 빛 반사 없이 선명하게 촬영 가능한 상태
 - * 본인 명의 휴대폰 (본인 인증용)
 - * (선택 사항) 기존에 보유하고 있는 다른 은행 계좌 (타행 계좌 인증 시 필요)
- * ****2단계: 은행 모바일 앱 다운로드 및 설치****
 - * 이용할 은행의 공식 모바일 뱅킹 앱을 스마트폰에 설치합니다.
 - * *예시: KB국민은행 스타뱅킹, 신한은행 SOL, 우리은행 WON뱅킹, 하나은행 하나원큐 등*
- * ****3단계: 앱 내 계좌 개설 절차 시작****
 - * 앱 실행 후 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등 관련 메뉴를 선택합니다.
- * ****4단계: 개설할 상품 선택****
 - * 원하는 계좌 종류 (보통 입출금 통장)를 선택합니다.
- * ****5단계: 본인 확인 절차 진행****
 - * ****신분증 촬영:**** 앱 안내에 따라 신분증을 촬영합니다.
 - * ****추가 본인 인증:****
 - * 영상 통화 (은행 상담원 연결)
 - * 타행 계좌 인증 (기존 계좌로 소액 입금 및 입금자명 확인)
 - * *이 과정은 은행마다 다를 수 있습니다.*
- * ****6단계: 정보 입력****
 - * 이름, 생년월일, 주소, 직업 등 필요한 개인 정보를 정확하게 입력합니다.
- * ****7단계: 약관 동의****
 - * 상품 이용 약관, 개인정보 수집 및 이용 동의 등 관련 내용을 확인하고 동의합니다.
- * ****8단계: 계좌 개설 완료****
 - * 모든 절차가 정상적으로 완료되면 계좌 개설이 완료됩니다.

이 순서대로 진행하시면 어렵지 않게 비대면 계좌를 개설하실 수 있을 거예요! 혹시 특정 은행이나 절차에서 막히는 부분이 있다면 언제든지 다시 들

대화 기록 확인

```
print("\n--- 전체 대화 기록 ---")
```

```
# (변경) load_memory_variables() 함수가 딕셔너리 형태로 반환하며,  
# memory_key로 지정한 "chat_history" 키를 통해 대화 기록에 접근함  
print(conversation.memory.load_memory_variables({})["history"])
```

• 셀 출력

--- 전체 대화 기록 ---

Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?

AI: 안녕하세요! 비대면으로 은행 계좌를 개설하시고 싶으시다니, 정말 편리한 시대에 살고 있죠! 제가 아는 선에서 최대한 자세하게 안내해 드릴게

일반적으로 비대면 계좌 개설은 크게 두 가지 방법으로 진행될 수 있어요.

1. ****은행 자체 모바일 앱을 이용하는 방법:****

- * 대부분의 은행들이 자체적으로 개발한 모바일 뱅킹 앱을 가지고 있어요. 이 앱을 통해 계좌 개설 절차를 진행하는 것이 가장 일반적입니다
- * ****준비물:****
 - * ****본인 명의의 스마트폰:**** 안드로이드나 iOS 운영체제를 사용하는 스마트폰이면 됩니다.

- * ****신분증:**** 주민등록증 또는 운전면허증이 필요해요. 신분증 촬영 시 빛 반사 없이 선명하게 나오도록 주의해야 합니다.
- * ****본인 명의 휴대폰:**** 본인 인증을 위해 필요합니다.
- * ****기존에 보유하고 있는 다른 은행 계좌 (선택 사항):**** 본인 확인 절차에서 신분증 인증 외에, 이미 보유하고 있는 다른 은행 :
- * ****절차:****
 1. ****은행 앱 다운로드:**** 이용하시려는 은행의 모바일 앱을 스마트폰에 설치합니다. (예: KB국민은행 스타뱅킹, 신한은행 SOL, 우
 2. ****계좌 개설 메뉴 선택:**** 앱을 실행한 후, 보통 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등의 메뉴를 찾습니다.
 3. ****상품 선택:**** 개설하시려는 계좌 종류 (입출금 통장, 저금 통장 등)를 선택합니다. 보통 입출금 통장을 먼저 개설하는 경우가
 4. ****본인 확인:****
 - * ****신분증 촬영:**** 안내에 따라 신분증을 촬영합니다.
 - * ****영상 통화 또는 타행 계좌 인증:**** 은행에 따라서는 상담원과 짧은 영상 통화를 하거나, 기존에 가지고 계신 다른 은행 계
 5. ****정보 입력:**** 이름, 생년월일, 주소, 직업 등 필요한 정보를 입력합니다.
 6. ****약관 동의:**** 상품 이용 약관 및 개인정보 수집/이용 동의 등을 진행합니다.
 7. ****계좌 개설 완료:**** 모든 절차가 정상적으로 완료되면 계좌가 개설됩니다.

2. ****정부 지원 서비스 (SCINet 등)를 이용하는 방법:****

- * 일부 은행에서는 정부에서 제공하는 신원확인 시스템과 연계하여 비대면 계좌 개설을 지원하기도 합니다. 하지만 이 방법은 아직 보편적이

****어떤 은행을 이용하실 예정이신가요?*** 혹시 특정 은행을 염두에 두고 계시다면, 해당 은행의 이름이나 앱 이름을 알려주시면 제가 더 구체적인 Human: 준비물을 알려주세요.

AI: 네, 비대면 계좌 개설을 위해 필요한 준비물들을 다시 한번 자세히 알려드릴게요! 😊

가장 일반적인 방법인 ****은행 자체 모바일 앱을 이용한 비대면 계좌 개설****을 기준으로 말씀드리겠습니다.

1. ****본인 명의 스마트폰:****

- * 이것은 필수입니다! 안드로이드 스마트폰이든 아이폰이든 상관없습니다.
- * 스마트폰으로 은행 앱을 다운로드하고, 카메라로 신분증을 촬영하며, 본인 인증 절차를 진행하기 때문에 꼭 필요해요.
- * ****주의할 점:**** 스마트폰의 카메라 기능이 정상적으로 작동해야 합니다.

2. ****신분증:****

- * ****주민등록증**** 또는 ****운전면허증****이 필요합니다.
- * ****가장 중요한 부분은 신분증을 촬영할 때입니다.****
 - * ****빛 반사를 최소화하세요.**** 형광등 불빛이나 창문 빛이 신분증에 직접적으로 반사되면 인식이 잘 안 될 수 있습니다.
 - * ****신분증의 모든 정보가 선명하게 보이도록 촬영해야 합니다.**** 글자가 흐릿하거나 잘리면 안 돼요.
 - * ****신분증의 네 모서리가 모두 나오도록 촬영****하는 것이 좋습니다.
 - * ****신분증을 평평한 곳에 놓고 촬영****하면 더 깔끔하게 찍을 수 있어요.

3. ****본인 명의 휴대폰:****

- * 이것 역시 필수입니다! 계좌 개설 과정에서 본인 확인을 위해 휴대폰으로 인증번호를 받거나, 경우에 따라서는 은행 상담원과의 영상 통화
- * ****알뜰폰(MVNO)을 사용하시는 경우, 일부 은행에서는 비대면 계좌 개설이 제한될 수도 있습니다.**** 이 점은 이용하시려는 은행에 미리

4. ****기존에 보유하고 있는 다른 은행 계좌 (본인 확인용, 선택 사항):****

- * 이것은 ****모든 은행에서 필수로 요구하는 것은 아니지만, 많은 은행에서 사용되는 추가적인 본인 확인 방법****입니다.
- * 신분증 촬영만으로는 본인 확인이 부족하다고 판단될 경우, ****이전에 사용하시던 다른 은행 계좌로 '1원' 또는 '소액'을 입금****하여 하
- * 따라서, ****본인 명의로 개설되어 있고 현재 사용 중인 다른 은행 계좌가 하나쯤 있다면**** 비대면 계좌 개설이 더 원활하게 진행될 수 있

요약하자면, ****스마트폰, 신분증, 그리고 본인 명의 휴대폰**** 이 세 가지가 가장 중요하다고 보시면 됩니다. 😊

혹시 이 중에서 준비하시면서 어려운 점이 있으시거나, 특정 은행에서 요구하는 사항이 궁금하시면 언제든지 다시 물어봐 주세요!

Human: 안녕하세요, 비대면으로 은행 계좌를 개설하고 싶습니다. 어떻게 시작해야 하나요?

AI: 안녕하세요! 비대면으로 은행 계좌를 개설하시고 싶으시다니, 정말 편리한 시대에 살고 있죠! 제가 아는 선에서 최대한 자세하게 안내해 드릴게

일반적으로 비대면 계좌 개설은 크게 두 가지 방법으로 진행될 수 있어요.

1. ****은행 자체 모바일 앱을 이용하는 방법:****

- * 대부분의 은행들이 자체적으로 개발한 모바일 뱅킹 앱을 가지고 있어요. 이 앱을 통해 계좌 개설 절차를 진행하는 것이 가장 일반적입니다
- * ****준비물:****
 - * ****본인 명의 스마트폰:**** 안드로이드나 iOS 운영체제를 사용하는 스마트폰이면 됩니다.
 - * ****신분증:**** 주민등록증 또는 운전면허증이 필요해요. 신분증 촬영 시 빛 반사 없이 선명하게 나오도록 주의해야 합니다.
 - * ****본인 명의 휴대폰:**** 본인 인증을 위해 필요합니다.

- * **기존에 보유하고 있는 다른 은행 계좌 (선택 사항):** 본인 확인 절차에서 신분증 인증 외에, 이미 보유하고 있는 다른 은행 :
- * **절차:**
 1. **은행 앱 다운로드:** 이용하시려는 은행의 모바일 앱을 스마트폰에 설치합니다. (예: KB국민은행 스타뱅킹, 신한은행 SOL, 우
 2. **계좌 개설 메뉴 선택:** 앱을 실행한 후, 보통 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등의 메뉴를 찾습니다.
 3. **상품 선택:** 개설하시려는 계좌 종류 (입출금 통장, 적금 통장 등)를 선택합니다. 보통 입출금 통장을 먼저 개설하는 경우가
 4. **본인 확인:**
 - * **신분증 촬영:** 안내에 따라 신분증을 촬영합니다.
 - * **영상 통화 또는 타행 계좌 인증:** 은행에 따라서는 상담원과 짧은 영상 통화를 하거나, 기존에 가지고 계신 다른 은행 가
 5. **정보 입력:** 이름, 생년월일, 주소, 직업 등 필요한 정보를 입력합니다.
 6. **약관 동의:** 상품 이용 약관 및 개인정보 수집/이용 동의 등을 진행합니다.
 7. **계좌 개설 완료:** 모든 절차가 정상적으로 완료되면 계좌가 개설됩니다.

2. **정부 지원 서비스 (SCINet 등)를 이용하는 방법:**

- * 일부 은행에서는 정부에서 제공하는 신원확인 시스템과 연계하여 비대면 계좌 개설을 지원하기도 합니다. 하지만 이 방법은 아직 보편적이

****어떤 은행을 이용하실 예정이신가요?*** 혹시 특정 은행을 염두에 두고 계시다면, 해당 은행의 이름이나 앱 이름을 알려주시면 제가 더 구체적인 Human: 볼렛포인트로 정리해서 알려주세요.

AI: 네, 비대면 계좌 개설 절차를 볼렛 포인트로 깔끔하게 정리해 드릴게요! 😊

****비대면 은행 계좌 개설 절차 (모바일 앱 이용 시)****

- * ****1단계: 준비물 확인 및 확보****
 - * 본인 명의 스마트폰 (안드로이드 또는 iOS)
 - * 신분증 (주민등록증 또는 운전면허증) - 빛 반사 없이 선명하게 촬영 가능한 상태
 - * 본인 명의 휴대폰 (본인 인증용)
 - * (선택 사항) 기존에 보유하고 있는 다른 은행 계좌 (타행 계좌 인증 시 필요)
- * ****2단계: 은행 모바일 앱 다운로드 및 설치****
 - * 이용할 은행의 공식 모바일 뱅킹 앱을 스마트폰에 설치합니다.
 - * *예시: KB국민은행 스타뱅킹, 신한은행 SOL, 우리은행 WON뱅킹, 하나은행 하나원큐 등*
- * ****3단계: 앱 내 계좌 개설 절차 시작****
 - * 앱 실행 후 '계좌 개설', '비대면 계좌 개설', '상품 가입' 등 관련 메뉴를 선택합니다.
- * ****4단계: 개설할 상품 선택****
 - * 원하는 계좌 종류 (보통 입출금 통장)를 선택합니다.
- * ****5단계: 본인 확인 절차 진행****
 - * ****신분증 촬영:**** 앱 안내에 따라 신분증을 촬영합니다.
 - * ****추가 본인 인증:****
 - * 영상 통화 (은행 상담원 연결)
 - * 타행 계좌 인증 (기존 계좌로 소액 입금 및 입금자명 확인)
 - * *이 과정은 은행마다 다를 수 있습니다.*
- * ****6단계: 정보 입력****
 - * 이름, 생년월일, 주소, 직업 등 필요한 개인 정보를 정확하게 입력합니다.
- * ****7단계: 약관 동의****
 - * 상품 이용 약관, 개인정보 수집 및 이용 동의 등 관련 내용을 확인하고 동의합니다.
- * ****8단계: 계좌 개설 완료****
 - * 모든 절차가 정상적으로 완료되면 계좌 개설이 완료됩니다.

이 순서대로 진행하시면 어렵지 않게 비대면 계좌를 개설하실 수 있을 거예요! 혹시 특정 은행이나 절차에서 막히는 부분이 있다면 언제든지 다시 돌

```
print(type(conversation.memory.load_memory_variables({})["history"])) # <class 'str'>
print(type(response)) # <class 'str'>
print(type(memory)) # <class 'langchain.memory.buffer.Conve
```

```
<class 'str'>
<class 'str'>
<class 'langchain.memory.buffer.ConversationBufferMemory'>
```

- test

```
# ConversationChain 생성하기
# memory_key와 input_key를 명시적으로 설정해야 함

from langchain.chains import ConversationChain
from langchain.memory import ConversationBufferMemory
from langchain_core.prompts import ChatPromptTemplate, MessagesPlaceholder


conversation = ConversationChain(
    llm=gemini_lc,

    # (변경) chat_history를 memory_key로 명시적 설정
    # memory=ConversationBufferMemory(memory_key="chat_history"), # 오류 발생
    # (수정) memory_key를 'history'로 설정하여 ConversationChain의 기본 프롬프트와 일치
    memory=ConversationBufferMemory(
        return_messages=True,
        memory_key="history"),

    # 추론 과정 확인을 위해 True로 설정
    verbose=True
)

# 첫번째 대화

response = conversation.predict(input="Langchain을 배우고 싶어요. 어떻게 해야 하나요?")
print("AI:", response)
```

- 셀 출력 (7.0s)
 - 출력 화면

```
**Entering new ConversationChain chain...**
**Prompt after formatting:**
**The following is a friendly conversation between a human and an AI. The AI is talkative and provide

**Current conversation:**
[]
**Human: Langchain을 배우고 싶어요. 어떻게 해야 하나요?**
**AI:**

**Finished chain.**
AI: 안녕하세요! LangChain을 배우고 싶으시다니 정말 좋은 생각이세요! LangChain은 대규모 언어 모델(LLM)을 활용한 애플리케이션을

LangChain을 배우는 데에는 여러 가지 방법이 있는데, 제가 생각하기에 가장 효과적인 순서대로 말씀드릴게요.

**1. LangChain의 기본 개념 이해하기:**

*   **LLM (Large Language Model) 이란?** LangChain은 LLM을 기반으로 작동하기 때문에 LLM이 무엇인지, 어떻게 작동하는지
*   **LangChain의 목표와 장점:** LangChain이 왜 만들어졌는지, 어떤 문제를 해결해주고 어떤 장점을 제공하는지 이해하면 학습 동
*   **핵심 컴포넌트:** LangChain은 여러 모듈로 구성되어 있어요. 이 모듈들이 어떻게 서로 연결되고 작동하는지 큰 그림을 이해하는
    *   **Models:** 다양한 LLM과의 인터페이스를 제공합니다. (e.g., OpenAI, Hugging Face, Anthropic)
    *   **Prompts:** LLM에 전달할 질문이나 지시사항을 효과적으로 관리하고 구성하는 방법을 제공합니다. Prompt templates,
    *   **Chains:** 여러 LLM 호출이나 다른 유틸리티를 순서대로 연결하여 복잡한 워크플로우를 만드는 핵심 기능입니다. 간단한 LL
    *   **Indexes:** 외부 데이터를 LLM이 이해하고 활용할 수 있는 형태로 저장하고 검색하는 방법을 제공합니다. Document Loa
```

- * **Agents:** LLM이 스스로 어떤 도구를 사용하고 어떤 순서로 작업을 수행할지 결정하도록 만드는 기능입니다. LLM이 마치 '자신의 생각과 계획을 스스로 결정'하는 것처럼 행동할 수 있도록 하는 기능입니다.
- * **Memory:** 이전 대화 내용을 기억하여 LLM이 맥락을 유지하도록 돕는 기능입니다. ConversationBufferMemory, ConversationSummaryMemory 등 다양한 Memory 클래스가 제공됩니다.

2. 공식 문서와 튜토리얼 활용하기:

- * **LangChain 공식 문서:** LangChain의 가장 정확하고 최신 정보를 얻을 수 있는 곳은 당연히 공식 문서입니다.
 - * **Python 문서:** https://python.langchain.com/docs/get_started/introduction
 - * **JavaScript 문서:** https://js.langchain.com/docs/get_started/introduction
- * 저는 주로 Python 버전을 기반으로 학습되었기 때문에 Python 문서가 더 익숙하지만, JavaScript 버전도 매우 잘 되어 있습니다.
- * **"Get Started" 섹션:** 문서를 처음 보시면 "Get Started" 섹션이 있을 거예요. 여기서 LangChain 설치 방법부터 간단한 예제까지 다루고 있습니다.
- * **"Use Cases" 또는 "Cookbook" 섹션:** 기본적인 개념을 익히셨다면, 실제 다양한 Use Case들을 살펴보세요. 질문 답변 봇, 요약기, 분류기 등 다양한 애플리케이션을 만드는 방법을 배울 수 있습니다.

3. 직접 코드를 작성하고 실행해보기:

- * **간단한 예제부터 시작:** 공식 문서에 나오는 가장 기본적인 예제부터 직접 타이핑하고 실행해보세요. 예를 들어, 간단한 LLM 호출 예제부터 시작합니다.
- * **IDE (통합 개발 환경) 사용:** VS Code, PyCharm 같은 IDE를 사용하면 코드 자동 완성, 디버깅 기능 등을 활용할 수 있어 개발 효율이 높아집니다.
- * **API 키 설정:** OpenAI API 키와 같이 필요한 API 키를 설정하는 방법을 익히고, 실제 LLM을 호출해보면서 결과를 확인하는 과정을 경험해보세요.

4. 커뮤니티와 자료 활용하기:

- * **GitHub 저장소:** LangChain의 GitHub 저장소는 코드를 직접 보고, 이슈를 확인하고, 최신 업데이트를 파악하는 데 아주 좋습니다.
 - * **Python:** <https://github.com/langchain-ai/langchain>
 - * **JavaScript/TypeScript:** <https://github.com/langchain-ai/langchainjs>
- * **Discord 채널:** LangChain 커뮤니티는 Discord에서 매우 활발합니다. 궁금한 점이 생기거나 막히는 부분이 있을 때 질문하고 도움을 받을 수 있습니다.
- * **YouTube 튜토리얼:** 많은 개발자들이 LangChain 사용법, 특정 기능 구현 방법 등을 담은 YouTube 영상을 올리고 있습니다. 검색을 통해 찾아보세요.
- * **블로그 및 기술 문서:** Medium, 개인 블로그 등에도 LangChain 관련 유용한 정보들이 많이 공유되고 있습니다.

5. 특정 기능에 집중해서 학습하기:

- * 처음부터 모든 것을 다 알려고 하기보다는, 본인이 만들고 싶은 애플리케이션의 핵심 기능에 집중해서 학습하는 것이 효과적일 수 있습니다.
 - * **챗봇을 만들고 싶다면:** Memory, Chains, LLM 모델 사용법에 집중
 - * **문서 요약기를 만들고 싶다면:** Document Loaders, Text Splitters, Summarization Chains에 집중
 - * **질의응답 시스템을 만들고 싶다면:** Vector Stores, Retrievers, RetrievalQA Chain에 집중

제가 학습한 정보를 기반으로 몇 가지 더 구체적인 팁을 드리자면:

- * **LangChain Expression Language (LCEL):** 최근 LangChain에서는 LCEL이라는 새로운 방식을 도입해서 체인을 더 유연하고 직관적으로 만들 수 있게 되었습니다.
- * **LangSmith:** LangChain 애플리케이션을 디버깅하고 모니터링하는 데 LangSmith라는 도구가 있습니다. 이 도구를 사용하면 LLM 호출의 성능과 비용을 추적할 수 있습니다.
- * **LLM의 이해:** LangChain은 LLM을 잘 활용하기 위한 도구이기 때문에, LLM 자체에 대한 이해도가 높을수록 LangChain을 더 효과적으로 활용할 수 있습니다.


어떤 언어로 LangChain을 배우고 싶으신가요? Python인가요, 아니면 JavaScript/TypeScript인가요? 알려주시면 해당 언어에 맞춰 더 자세한 안내를 드릴 수 있습니다.

LangChain 학습 여정을 응원합니다! 궁금한 점이 있다면 언제든지 다시 물어보세요! 😊

첫번째 대화

```
response = conversation.predict(input="Langchain을 배우기 좋은 GitHub 저장소 3개만 추천해주세요.")
print("AI:", response)
```

• 셀 출력 (3.3s)

-  결과 출력 화면

```
**Entering new ConversationChain chain...**
**Prompt after formatting:**
**The following is a friendly conversation between a human and an AI. The AI is talkative and provides
detailed answers.

**Current conversation:**
```

****[HumanMessage(content='Langchain을 배우고 싶어요. 어떻게 해야 하나요?', additional_kwargs={}, response_metadata= {})]**

- LangChain은 대규모 언어 모델(LLM)을 활용한 애플리케이션을 쉽고 빠르게 구축할 수 있도록 도와주는 아주 강력한 프레임워크입니다.
- 저는 2023년 10월까지의 정보로 학습되었기 때문에, LangChain의 최신 업데이트나 특정 버전에 대한 아주 세세한 부분까지는 실시간으로 LangChain을 배우는 데에는 여러 가지 방법이 있는데, 제가 생각하기에 가장 효과적인 순서대로 말씀드릴게요.

****1. LangChain의 기본 개념 이해하기:****

****LLM (Large Language Model)이란?**** LangChain은 LLM을 기반으로 작동하기 때문에 LLM이 무엇인지, 어떻게 작동하는지에 대해 먼저 이해하는 것이 중요합니다.

- **LangChain의 목표와 장점:**** LangChain이 왜 만들어졌는지, 어떤 문제를 해결해주고 어떤 장점을 제공하는지 이해하면 학습 동기가 생길 것입니다.
- **핵심 컴포넌트:**** LangChain은 여러 모듈로 구성되어 있어요. 이 모듈들이 어떻게 서로 연결되고 작동하는지 큰 그림을 이해하는 것이 중요합니다.
- **Models:**** 다양한 LLM과의 인터페이스를 제공합니다. (e.g., OpenAI, Hugging Face, Anthropic)
- **Prompts:**** LLM에 전달할 질문이나 지시사항을 효과적으로 관리하고 구성하는 방법을 제공합니다. Prompt templates, prompt engineering 등
- **Chains:**** 여러 LLM 호출이나 다른 유틸리티를 순서대로 연결하여 복잡한 워크플로우를 만드는 핵심 기능입니다. 간단한 LLM Chain부터 시작합니다.
- **Indexes:**** 외부 데이터를 LLM이 이해하고 활용할 수 있는 형태로 저장하고 검색하는 방법을 제공합니다. Document Loaders, Vector Stores 등
- **Agents:**** LLM이 스스로 어떤 도구를 사용하고 어떤 순서로 작업을 수행할지 결정하도록 만드는 기능입니다. LLM이 마치 '추론'하는 것처럼 작동합니다.
- **Memory:**** 이전 대화 내용을 기억하여 LLM이 맥락을 유지하도록 돕는 기능입니다. ConversationBufferMemory, ConversationSummaryMemory 등

****2. 공식 문서와 튜토리얼 활용하기:****

- **LangChain 공식 문서:**** LangChain의 가장 정확하고 최신 정보를 얻을 수 있는 곳은 당연히 공식 문서입니다.
- **https://python.langchain.com/docs/get_started/introduction****
- **https://js.langchain.com/docs/get_started/introduction****
- 저는 주로 Python 버전을 기반으로 학습되었기 때문에 Python 문서가 더 익숙하지만, JavaScript 버전도 매우 잘 되어 있습니다.
- **"Get Started" 섹션:**** 문서를 처음 보시면 "Get Started" 섹션이 있을 거예요. 여기서 LangChain 설치 방법부터 간단한 예제부터 시작합니다.
- **"Use Cases" 또는 "Cookbook" 섹션:**** 기본적인 개념을 익히셨다면, 실제 다양한 Use Case들을 살펴보세요. 질문 답변 봇, 요약기, 분류기 등

****3. 직접 코드를 작성하고 실행해보기:****

****간단한 예제부터 시작:**** 공식 문서에 나오는 가장 기본적인 예제부터 직접 타이핑하고 실행해보세요. 예를 들어, 간단한 LLM 호출 (LLM에 메시지를 보내고 응답을 받아오는 것)부터 시작합니다.

- **IDE (통합 개발 환경) 사용:**** VS Code, PyCharm 같은 IDE를 사용하면 코드 자동 완성, 디버깅 기능 등을 활용할 수 있어 편리합니다.
- **API 키 설정:**** OpenAI API 키와 같이 필요한 API 키를 설정하는 방법을 익히고, 실제 LLM을 호출해보면서 결과를 확인하는 과정을 경험합니다.

****4. 커뮤니티와 자료 활용하기:****

- **GitHub 저장소:**** LangChain의 GitHub 저장소는 코드를 직접 보고, 이슈를 확인하고, 최신 업데이트를 파악하는 데 아주 좋습니다.
- **Python:**** https://github.com/langchain-ai/langchain
- **JavaScript/TypeScript:**** https://github.com/langchain-ai/langchainjs
- **YouTube 튜토리얼:**** 많은 개발자들이 LangChain 사용법, 특정 기능 구현 방법 등을 담은 YouTube 영상을 올리고 있습니다. "LangChain Tutorial"로 검색해보세요.
- **블로그 및 기술 문서:**** Medium, 개인 블로그 등에도 LangChain 관련 유용한 정보들이 많이 공유되고 있습니다.

****5. 특정 기능에 집중해서 학습하기:****

- 처음부터 모든 것을 다 알려고 하기보다는, 본인이 만들고 싶은 애플리케이션의 핵심 기능에 집중해서 학습하는 것이 효과적일 수 있습니다.
- **챗봇을 만들고 싶다면:**** Memory, Chains, LLM 모델 사용법에 집중
- **문서 요약과 검색을 하고 싶다면:**** Document Loaders, Text Splitters, Summarization Chains에 집중
- **질의응답 시스템을 만들고 싶다면:**** Vector Stores, Retrievers, RetrievalQA Chain에 집중

****제가 학습한 정보를 기반으로 몇 가지 더 구체적인 팁을 드리자면:****

- **LangChain Expression Language (LCEL):**** 최근 LangChain에서는 LCEL이라는 새로운 방식을 도입해서 체인을 더 유연하고 강력하게 만들었습니다.
- **LangSmith:**** LangChain 애플리케이션을 디버깅하고 모니터링하는 데 LangSmith라는 도구가 있습니다. 이 도구를 사용하면 LLM의 성능을 추적하고 문제를 해결할 수 있습니다.
- **LLM의 이해:**** LangChain은 LLM을 잘 활용하기 위한 도구이기 때문에, LLM 자체에 대한 이해도가 높을수록 LangChain을 더 효과적으로 활용할 수 있습니다.

어떤 언어로 LangChain을 배우고 싶으신가요? Python인가요, 아니면 JavaScript/TypeScript인가요? 알려주시면 해당 언어에 맞춰 더 자세한 안내를 드릴 수 있습니다.

LangChain 학습 여정을 응원합니다! 궁금한 점이 있다면 언제든지 다시 물어보세요! 😊',

****additional_kwargs={}, response_metadata={}]****

****Human:** Langchain을 배우기 좋은 GitHub 저장소 3개만 추천해주세요.**

****AI:****

****Finished chain.****

****AI:** 물론입니다! LangChain을 배우는 데 도움이 될 만한 GitHub 저장소를 3가지 추천해 드릴게요. LangChain 개발의 중심지이기!

1. ****LangChain Python (핵심 저장소)****

- * ****URL:**** <https://github.com/langchain-ai/langchain>
- * ****추천 이유:**** 이것이 바로 LangChain의 메인 Python 저장소입니다. LangChain의 거의 모든 기능, 모듈, 컴포넌트에 대한 코드가 포함되어 있습니다.
 - * ****실제 코드 확인:**** LangChain이 내부적으로 어떻게 작동하는지, 각 컴포넌트(Models, Chains, Indexes, Agents 등)의 역할과 상호작용을 이해할 수 있습니다.
 - * ****예제 코드:**** `examples` 폴더나 `cookbook` 섹션에 다양한 기능별 예제 코드가 포함되어 있어, 특정 기능을 구현하는 방법을 빠르게 배울 수 있습니다.
 - * ****최신 업데이트:**** LangChain 팀이 직접 관리하기 때문에 가장 최신의 변경 사항과 새로운 기능에 대한 정보를 얻을 수 있습니다.
 - * ****이슈 트래킹:**** 개발 중에 발생하는 문제점이나 버그, 그리고 사용자들의 질문과 피드백을 볼 수 있어서, 실제 개발 과정에서 마주칠 수 있는 문제를 해결하는 데 도움이 됩니다.

2. ****LangChain Cookbook (예제 중심)****

- * ****URL:**** <https://github.com/langchain-ai/langchain-cookbook>
- * ****추천 이유:**** 이 저장소는 LangChain의 다양한 사용 사례와 튜토리얼을 코드로 제공하는 데 특화되어 있습니다. 공식 문서의 예제를 직접 실행해볼 수 있는 장점이 있습니다.
 - * ****다양한 Use Cases:**** 챗봇, 문서 요약, 데이터 분석, RAG(Retrieval Augmented Generation) 시스템 구축 등 다양한 사용 사례를 다룹니다.
 - * ****실습 중심:**** 각 예제는 특정 문제를 해결하기 위해 LangChain의 어떤 컴포넌트들을 어떻게 조합해야 하는지를 명확하게 보여줍니다.
 - * ****단계별 학습:**** 초급부터 고급까지 다양한 난이도의 예제를 제공하여, 자신의 수준에 맞춰 학습 범위를 넓혀갈 수 있습니다.

3. ****LangChain JS (JavaScript/TypeScript 사용자용)****

- * ****URL:**** <https://github.com/langchain-ai/langchainjs>
- * ****추천 이유:**** 만약 JavaScript 또는 TypeScript를 사용하여 LangChain을 배우고 싶으시다면, 이 저장소가 필수적입니다.
 - * ****프론트엔드 및 Node.js 개발:**** 웹 애플리케이션이나 Node.js 기반 백엔드에서 LangChain을 사용하고 싶을 때 가장 적합한 리소스입니다.
 - * ****동일한 개념, 다른 언어:**** Python에서 배우는 LangChain의 핵심 개념(Chains, Agents, Prompts 등)은 JavaScript/TypeScript에서도 동일하게 적용됩니다.
 - * ****예제 및 문서:**** Python 저장소와 유사하게 예제 코드와 함께 JavaScript/TypeScript 환경에서의 LangChain 사용법을 상세히 설명합니다.

이 세 가지 저장소를 주로 살펴보면 LangChain의 기본적인 사용법부터 고급 활용법까지, 그리고 다양한 예제들을 통해 실질적인 개발 능력이 향상될 것입니다.

특히 Python 저장소의 `examples` 폴더와 LangChain Cookbook 저장소의 예제 코드를 직접 실행해보고, 어떻게 작동하는지 이해하는 것이 가장 중요합니다.

LangChain 학습, 파이팅입니다! 💪

```

from langchain.schema import HumanMessage, AIMessage

def pretty_print_history(memory, memory_key="history"):
    """
    memory.load_memory_variables({})[memory_key] 로 꺼낸
    List[HumanMessage|AIMessage]을 가독성 좋게 포매팅해 출력합니다.
    """
    history = memory.load_memory_variables({})[memory_key]
    print("\n📄 전체 대화 기록\n")
    for idx, msg in enumerate(history, start=1):
        if isinstance(msg, HumanMessage):
            role, emoji = "사용자", "👤"
        elif isinstance(msg, AIMessage):
            role, emoji = "어시스턴트", "🤖"
        else:
            role, emoji = msg.__class__.__name__, ""

        print(f"{idx:02d}. {emoji} {role}:")
        for line in msg.content.split("\n"):
            print("    " + line)
        print()

# 호출: memory_key="history" 로 만든 상태라면
pretty_print_history(conversation.memory, "history")

```



📄 전체 대화 기록

01. 👤 사용자:
LangChain을 배우고 싶어요. 어떻게 해야 하나요?
02. 🤖 어시스턴트:
안녕하세요! LangChain을 배우고 싶으시다니 정말 좋은 생각이세요! LangChain은 대규모 언어 모델(LLM)을 활용한 애플리케이션을 쉽고 LangChain을 배우는 데에는 여러 가지 방법이 있는데, 제가 생각하기에 가장 효과적인 순서대로 말씀드릴게요.
 - **1. LangChain의 기본 개념 이해하기:****
 - * ****LLM (Large Language Model) 이란?**** LangChain은 LLM을 기반으로 작동하기 때문에 LLM이 무엇인지, 어떻게 작동하는지도
 - * ****LangChain의 목표와 장점:**** LangChain이 왜 만들어졌는지, 어떤 문제를 해결해주고 어떤 장점을 제공하는지 이해하면 학습 동기
 - * ****핵심 컴포넌트:**** LangChain은 여러 모듈로 구성되어 있어요. 이 모듈들이 어떻게 서로 연결되고 작동하는지 큰 그림을 이해하는 것
 - * ****Models:**** 다양한 LLM과의 인터페이스를 제공합니다. (e.g., OpenAI, Hugging Face, Anthropic)
 - * ****Prompts:**** LLM에 전달할 질문이나 지시사항을 효과적으로 관리하고 구성하는 방법을 제공합니다. Prompt templates, e
 - * ****Chains:**** 여러 LLM 호출이나 다른 유틸리티를 순서대로 연결하여 복잡한 워크플로우를 만드는 핵심 기능입니다. 간단한 LLM
 - * ****Indexes:**** 외부 데이터를 LLM이 이해하고 활용할 수 있는 형태로 저장하고 검색하는 방법을 제공합니다. Document Load
 - * ****Agents:**** LLM이 스스로 어떤 도구를 사용하고 어떤 순서로 작업을 수행할지 결정하도록 만드는 기능입니다. LLM이 마치 '추
 - * ****Memory:**** 이전 대화 내용을 기억하여 LLM이 맥락을 유지하도록 돕는 기능입니다. ConversationBufferMemory, Conve
 - **2. 공식 문서와 튜토리얼 활용하기:****
 - * ****LangChain 공식 문서:**** LangChain의 가장 정확하고 최신 정보를 얻을 수 있는 곳은 당연히 공식 문서입니다.
 - * ****[https://python.langchain.com/docs/get_started/introduction](https://python.langchain.com/do**
 - * ****[https://js.langchain.com/docs/get_started/introduction](https://js.langchain.com/docs/get_s**
 - * 저는 주로 Python 버전을 기반으로 학습되었기 때문에 Python 문서가 더 익숙하지만, JavaScript 버전도 매우 잘 되어 있습
 - * ****"Get Started" 섹션:**** 문서를 처음 보시면 "Get Started" 섹션이 있을 거예요. 여기서 LangChain 설치 방법부터 간단한
 - * ****"Use Cases" 또는 "Cookbook" 섹션:**** 기본적인 개념을 익히셨다면, 실제 다양한 Use Case들을 살펴보세요. 질문 답변 봇,
 - **3. 직접 코드를 작성하고 실행해보기:****
 - * ****간단한 예제부터 시작:**** 공식 문서에 나오는 가장 기본적인 예제부터 직접 타이핑하고 실행해보세요. 예를 들어, 간단한 LLM 호출 (
 - * ****IDE (통합 개발 환경) 사용:**** VS Code, PyCharm 같은 IDE를 사용하면 코드 자동 완성, 디버깅 기능 등을 활용할 수 있어 학
 - * ****API 키 설정:**** OpenAI API 키와 같이 필요한 API 키를 설정하는 방법을 익히고, 실제 LLM을 호출해보면서 결과를 확인하는 것
 - **4. 커뮤니티와 자료 활용하기:****
 - * ****GitHub 저장소:**** LangChain의 GitHub 저장소는 코드를 직접 보고, 이슈를 확인하고, 최신 업데이트를 파악하는 데 아주 좋습
 - * ****Python:**** [<https://github.com/langchain-ai/langchain>](https://github.com/langchain-ai/langch
 - * ****JavaScript/TypeScript:**** [<https://github.com/langchain-ai/langchainjs>](https://github.com/la
 - * ****Discord 채널:**** LangChain 커뮤니티는 Discord에서 매우 활발합니다. 궁금한 점이 생기거나 막히는 부분이 있을 때 질문하면
 - * ****YouTube 튜토리얼:**** 많은 개발자들이 LangChain 사용법, 특정 기능 구현 방법 등을 담은 YouTube 영상을 올리고 있습니다. '
 - * ****블로그 및 기술 문서:**** Medium, 개인 블로그 등에도 LangChain 관련 유용한 정보들이 많이 공유되고 있습니다.
 - **5. 특정 기능에 집중해서 학습하기:****

- * 처음부터 모든 것을 다 알려고 하기보다는, 본인이 만들고 싶은 애플리케이션의 핵심 기능에 집중해서 학습하는 것이 효과적일 수 있습니다
- * **챗봇을 만들고 싶다면:** Memory, Chains, LLM 모델 사용법에 집중
- * **문서 요약을 하고 싶다면:** Document Loaders, Text Splitters, Summarization Chains에 집중
- * **질의응답 시스템을 만들고 싶다면:** Vector Stores, Retrievers, RetrievalQA Chain에 집중

****제가 학습한 정보를 기반으로 몇 가지 더 구체적인 팁을 드리자면:****

- * ****LangChain Expression Language (LCEL):**** 최근 LangChain에서는 LCEL이라는 새로운 방식을 도입해서 체인을 더 유연하게

• 가독성 좋게 출력해보기

📖 전체 대화 기록

01. 👤 사용자:

Langchain을 배우고 싶어요. 어떻게 해야 하나요?

02. 🤖 어시스턴트:

안녕하세요! LangChain을 배우고 싶으시다니 정말 좋은 생각이세요! LangChain은 대규모 언어 모델(LLM)을 활용한 애플리케이션을 쉽고

LangChain을 배우는 데에는 여러 가지 방법이 있는데, 제가 생각하기에 가장 효과적인 순서대로 말씀드릴게요.

****1. LangChain의 기본 개념 이해하기:****

- * ****LLM (Large Language Model) 이란?**** LangChain은 LLM을 기반으로 작동하기 때문에 LLM이 무엇인지, 어떻게 작동하는지
- * ****LangChain의 목표와 장점:**** LangChain이 왜 만들어졌는지, 어떤 문제를 해결해주고 어떤 장점을 제공하는지 이해하면 학습 동
- * ****핵심 컴포넌트:**** LangChain은 여러 모듈로 구성되어 있어요. 이 모듈들이 어떻게 서로 연결되고 작동하는지 큰 그림을 이해하는 것
- * ****Models:**** 다양한 LLM과의 인터페이스를 제공합니다. (e.g., OpenAI, Hugging Face, Anthropic)
- * ****Prompts:**** LLM에 전달할 질문이나 지시사항을 효과적으로 관리하고 구성하는 방법을 제공합니다. Prompt templates, e
- * ****Chains:**** 여러 LLM 호출이나 다른 유틸리티를 순서대로 연결하여 복잡한 워크플로우를 만드는 핵심 기능입니다. 간단한 LLM
- * ****Indexes:**** 외부 데이터를 LLM이 이해하고 활용할 수 있는 형태로 저장하고 검색하는 방법을 제공합니다. Document Load
- * ****Agents:**** LLM이 스스로 어떤 도구를 사용하고 어떤 순서로 작업을 수행할지 결정하도록 만드는 기능입니다. LLM이 마치 '자
- * ****Memory:**** 이전 대화 내용을 기억하여 LLM이 맥락을 유지하도록 돕는 기능입니다. ConversationBufferMemory, Conv

****2. 공식 문서와 튜토리얼 활용하기:****

- * ****LangChain 공식 문서:**** LangChain의 가장 정확하고 최신 정보를 얻을 수 있는 곳은 당연히 공식 문서입니다.
- * ****https://python.langchain.com/docs/get_started/introduction**
- * ****https://js.langchain.com/docs/get_started/introduction**
- * 저는 주로 Python 버전을 기반으로 학습되었기 때문에 Python 문서가 더 익숙하지만, JavaScript 버전도 매우 잘 되어 있습
- * ****"Get Started" 섹션:**** 문서를 처음 보시면 "Get Started" 섹션이 있을 거예요. 여기서 LangChain 설치 방법부터 간단한
- * ****"Use Cases" 또는 "Cookbook" 섹션:**** 기본적인 개념을 익히셨다면, 실제 다양한 Use Case들을 살펴보세요. 질문 답변 봇,

****3. 직접 코드를 작성하고 실행해보기:****

- * ****간단한 예제부터 시작:**** 공식 문서에 나오는 가장 기본적인 예제부터 직접 타이핑하고 실행해보세요. 예를 들어, 간단한 LLM 호출
- * ****IDE (통합 개발 환경) 사용:**** VS Code, PyCharm 같은 IDE를 사용하면 코드 자동 완성, 디버깅 기능 등을 활용할 수 있어
- * ****API 키 설정:**** OpenAI API 키와 같이 필요한 API 키를 설정하는 방법을 익히고, 실제 LLM을 호출해보면서 결과를 확인하는 것

****4. 커뮤니티와 자료 활용하기:****

- * ****GitHub 저장소:**** LangChain의 GitHub 저장소는 코드를 직접 보고, 이슈를 확인하고, 최신 업데이트를 파악하는 데 아주 좋습
- * ****Python:**** https://github.com/langchain-ai/langchain
- * ****JavaScript/TypeScript:**** https://github.com/langchain-ai/langchainjs
- * ****Discord 채널:**** LangChain 커뮤니티는 Discord에서 매우 활발합니다. 궁금한 점이 생기거나 막히는 부분이 있을 때 질문하면
- * ****YouTube 튜토리얼:**** 많은 개발자들이 LangChain 사용법, 특정 기능 구현 방법 등을 담은 YouTube 영상을 올리고 있습니다.
- * ****블로그 및 기술 문서:**** Medium, 개인 블로그 등에도 LangChain 관련 유용한 정보들이 많이 공유되고 있습니다.

****5. 특정 기능에 집중해서 학습하기:****

- * 처음부터 모든 것을 다 알려고 하기보다는, 본인이 만들고 싶은 애플리케이션의 핵심 기능에 집중해서 학습하는 것이 효과적일 수 있습니다

- * ****챗봇을 만들고 싶다면:**** Memory, Chains, LLM 모델 사용법에 집중
- * ****문서 요약을 하고 싶다면:**** Document Loaders, Text Splitters, Summarization Chains에 집중
- * ****질의응답 시스템을 만들고 싶다면:**** Vector Stores, Retrievers, RetrievalQA Chain에 집중

****제가 학습한 정보를 기반으로 몇 가지 더 구체적인 팁을 드리자면:****

- * ****LangChain Expression Language (LCEL):**** 최근 LangChain에서는 LCEL이라는 새로운 방식을 도입해서 체인을 더 유연
- * ****LangSmith:**** LangChain 애플리케이션을 디버깅하고 모니터링하는 데 LangSmith라는 도구가 있습니다. 이 도구를 사용하면 차
- * ****LLM의 이해:**** LangChain은 LLM을 잘 활용하기 위한 도구이기 때문에, LLM 자체에 대한 이해도가 높을수록 LangChain을 더

어떤 언어로 LangChain을 배우고 싶으신가요? Python인가요, 아니면 JavaScript/TypeScript인가요? 알려주시면 해당 언어에 맞춰

LangChain 학습 여정을 응원합니다! 궁금한 점이 있다면 언제든지 다시 물어보세요! 😊

```

---

```markdown

03. 👤 사용자:

Langchain을 배우기 좋은 GitHub 저장소 3개만 추천해주세요.

04. 🤖 어시스턴트:

물론입니다! LangChain을 배우는 데 도움이 될 만한 GitHub 저장소를 3가지 추천해 드릴게요. LangChain 개발의 중심지이기도 하고,

1. ****LangChain Python (핵심 저장소)****

- * ****URL:**** <https://github.com/langchain-ai/langchain>
- * ****추천 이유:**** 이것이 바로 LangChain의 메인 Python 저장소입니다. LangChain의 거의 모든 기능, 모듈, 컴포넌트에 대해
 - * ****실제 코드 확인:**** LangChain이 내부적으로 어떻게 작동하는지, 각 컴포넌트(Models, Chains, Indexes, Agents)
 - * ****예제 코드:**** `examples` 폴더나 `cookbook` 섹션에 다양한 기능별 예제 코드가 포함되어 있어, 특정 기능을 구현하
 - * ****최신 업데이트:**** LangChain 팀이 직접 관리하기 때문에 가장 최신의 변경 사항과 새로운 기능에 대한 정보를 얻을 수
 - * ****이슈 트래킹:**** 개발 중에 발생하는 문제점이나 버그, 그리고 사용자들의 질문과 피드백을 볼 수 있어서, 실제 개발 과정

2. ****LangChain Cookbook (예제 중심)****

- * ****URL:**** <https://github.com/langchain-ai/langchain-cookbook>
- * ****추천 이유:**** 이 저장소는 LangChain의 다양한 사용 사례와 튜토리얼을 코드로 제공하는 데 특화되어 있습니다. 공식 문서의
 - * ****다양한 Use Cases:**** 챗봇, 문서 요약, 데이터 분석, RAG(Retrieval Augmented Generation) 시스템 구축
 - * ****실습 중심:**** 각 예제는 특정 문제를 해결하기 위해 LangChain의 어떤 컴포넌트들을 어떻게 조합해야 하는지를 명확하게
 - * ****단계별 학습:**** 초급부터 고급까지 다양한 난이도의 예제를 제공하여, 자신의 수준에 맞춰 학습 범위를 넓혀갈 수 있습니

3. ****LangChain JS (JavaScript/TypeScript 사용자용)****

- * ****URL:**** <https://github.com/langchain-ai/langchainjs>
- * ****추천 이유:**** 만약 JavaScript 또는 TypeScript를 사용하여 LangChain을 배우고 싶으시다면, 이 저장소가 필수적입니다
 - * ****프론트엔드 및 Node.js 개발:**** 웹 애플리케이션이나 Node.js 기반 백엔드에서 LangChain을 사용하고 싶을 때 가장
 - * ****동일한 개념, 다른 언어:**** Python에서 배우는 LangChain의 핵심 개념(Chains, Agents, Prompts 등)은 Jav
 - * ****예제 및 문서:**** Python 저장소와 유사하게 예제 코드와 함께 JavaScript/TypeScript 환경에서의 LangChain

이 세 가지 저장소를 주로 살펴보면 LangChain의 기본적인 사용법부터 고급 활용법까지, 그리고 다양한 예제들을 통해 실질적인 개발 능력