

6.2.1 사이드바 만들기

<코드> ch06_chatbot.py

```
import streamlit as st

# 1) main() 함수 선언
def main():
    # 2) 메인 화면 구성
    st.set_page_config(layout="wide")
    st.title("친근한 AI 챗봇")
    st.caption("스트림과 OpenAI API를 활용한 간단한 챗봇")
    # 3) 사이드바 구성
    with st.sidebar:
        st.subheader("OpenAI API Key 설정")
        # 4) 입력 위젯 유형 설정(비밀번호)
        openai_api_key = st.text_input("OpenAI API Key", type="password")
        st.write("[OpenAI API Key받기](https://platform.openai.com/account/api-keys)")

# 5) main() 함수 실행
if __name__ == "__main__":
    main()
```

</코드>

6.2.2 입력창 및 대화창 만들기

<코드> ch06_chatbot.py

```
import streamlit as st

# 1) main() 함수 선언
def main():

    # 2) 메인 화면 구성

    st.set_page_config(layout="wide")

    st.title("친근한 AI 챗봇")

    st.caption("스트림과 OpenAI API를 활용한 간단한 챗봇")

    # 3) 사이드바 구성

    with st.sidebar:

        st.subheader("OpenAI API Key 설정")

        # 4) 입력 위젯 유형 설정(비밀번호)

        openai_api_key = st.text_input("OpenAI API Key", type="password")

        st.write("[OpenAI API Key받기](https://platform.openai.com/account/api-keys)")

    # 6) 입력창 및 대화창 구현

    user_input = st.chat_input("무엇이 궁금한가요?")

    if user_input:

        # 7) 사용자 질문 출력

        with st.chat_message("user"):

            st.write(user_input)

        # 8) 챗봇 응답 출력

        with st.chat_message("assistant"):

            st.write("안녕! 난 친구 봇이야.😊")

# 5) main() 함수 실행
if __name__ == "__main__":

    main()
```

</코드>

6.2.3 OpenAI API로 요청 및 응답 처리하기

<코드> ch06_chatbot.py

```
import streamlit as st

from openai import OpenAI # 9) openai 패키지에서 OpenAI 클래스 불러오기

# 1) main() 함수 선언
def main():
    # 2) 메인 화면 구성
    st.set_page_config(layout="wide")
    st.title("친근한 AI 챗봇")
    st.caption("스트림릿과 OpenAI API를 활용한 간단한 챗봇")

    # 3) 사이드바 구성
    with st.sidebar:
        st.subheader("OpenAI API Key 설정")
        # 4) 입력 위젯 유형 설정(비밀번호)
        openai_api_key = st.text_input("OpenAI API Key", type="password")
        st.write("[OpenAI API Key받기](https://platform.openai.com/account/api-keys)")

    # 10) OpenAI 클라이언트 생성
    client = OpenAI(api_key=openai_api_key)

    # 6) 입력창 및 대화창 구현
    user_input = st.chat_input("무엇이 궁금한가요?")

    if user_input:
        # 7) 사용자 질문 출력
        with st.chat_message("user"):
            st.write(user_input)

        # 11) API 요청 및 응답(기존 8)번 코드 지우고 작성)
        with st.chat_message("assistant"):
            response = client.chat.completions.create(
                model="gpt-4o-mini",
                messages=[{"role": "assistant", "content": user_input}],
                stream=True,
```

```
)  
  
# 12) 응답 결과 출력  
  
st.write_stream(response)
```

```
# 5) main() 함수 실행  
  
if __name__ == "__main__":  
    main()
```

</코드>

6.2.4 과거 대화 내역 관리하기

<코드> ch06_chatbot.py

```
import streamlit as st

from openai import OpenAI # 9) openai 패키지에서 OpenAI 클래스 불러오기


# 1) main() 함수 선언
def main():
    # 2) 메인 화면 구성
    st.set_page_config(layout="wide")
    st.title("친근한 AI 챗봇")
    st.caption("스트림릿과 OpenAI API를 활용한 간단한 챗봇")

    # 3) 사이드바 구성
    with st.sidebar:
        st.subheader("OpenAI API Key 설정")
        # 4) 입력 위젯 유형 설정(비밀번호)
        openai_api_key = st.text_input("OpenAI API Key", type="password")
        st.write("[OpenAI API Key받기](https://platform.openai.com/account/api-keys)")

    # 13) 시스템 프롬프트 추가
    system_message = """너의 이름은 친구 봇이야.
너는 항상 반말을 하는 챗봇이야. 절대로 다나까 같은 높임말로 끝내지 마.
항상 반말로 친근하게 대답해줘.
영어로 질문을 받아도 무조건 한글로 답변해줘.
한글이 아닌 답변을 하게 되면 다시 생각해서 답변을 꼭 한글로 만들어줘.
모든 답변 끝에 답변에 맞는 이모티콘도 추가해줘.
"""

    # 14) 대화 내용 관리를 위한 세션 상태 설정
    if "messages" not in st.session_state:
        st.session_state.messages = [{"role": "system", "content": system_message}]

    # 15) 기존 대화 내역 표시

    idx = 0 # 대화 내역 순번 기록용

    for message in st.session_state.messages:
```

```
if idx > 0: # 시스템 프롬프트는 표시하지 않음
```

```
    with st.chat_message(message["role"]):
```

```
        st.write(message["content"])
```

```
    idx = idx + 1
```

```
# 10) OpenAI 클라이언트 생성
```

```
client = OpenAI(api_key=openai_api_key)
```

```
# 6) 입력창 및 대화창 구현
```

```
user_input = st.chat_input("무엇이 궁금한가요?")
```

```
if user_input:
```

```
    # 16) 세션 상태 리스트에 사용자 질문 추가
```

```
    st.session_state.messages.append({"role": "user", "content": user_input})
```

```
# 7) 사용자 질문 출력
```

```
with st.chat_message("user"):
```

```
    st.write(user_input)
```

```
# 17) API 요청 및 응답(기존 11~12)번 코드 지우고 작성)
```

```
with st.chat_message("assistant"):
```

```
    stream = client.chat.completions.create(
```

```
        model="gpt-4o-mini",
```

```
        messages=st.session_state.messages,
```

```
        stream=True,
```

```
    )
```

```
    response = st.write_stream(stream)
```

```
# 18) 세션 상태 리스트에 챗봇 응답 추가
```

```
st.session_state.messages.append({"role": "assistant", "content": response})
```

```
# 5) main() 함수 실행
```

```
if __name__ == "__main__":
```

```
    main()
```

</코드>