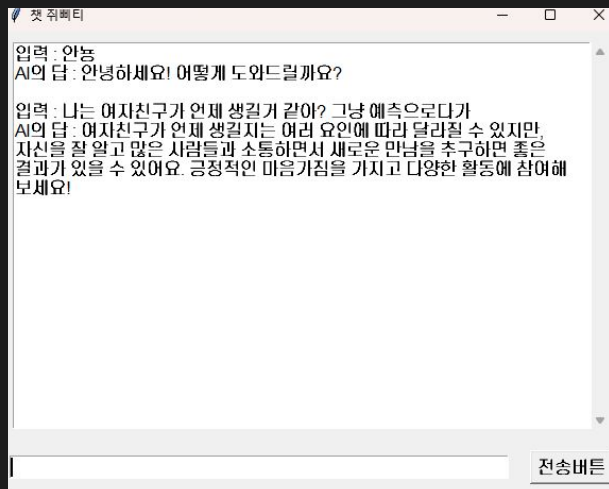


챗 쥐베티

최초의 파이썬용 AI 챗봇 프로그램



팀원 소개





프로젝트 개발 총괄
류용헌

1인 개발!!

만들게 된 계기에 대해

저는 챗 지피티라는 것을 몰랐는데
제 친구들을 포함해서 다른 사람들까지 다 쓰더라고요
그래서 제 친구한테 물어서 챗 지피티라는 것을 봤는데
와~ 굉장히 돈이 될 것 같더라고요 (자본주의)
그래서 저도 처음이지만 앞으로의 번영을 위해
챗 지피티같은 **AI 챗봇을 만들어 볼겁니다.**



프로젝트 개발 과정 소개



??

솔직히 처음에는 감이 안잡혔는데, 선린인고에서 **개최한 2025 SSF**에 AI쪽으로
참가하고 오니깐 신기하게 감이 잡혀버렸습니다(?)

첫 과정으로는 OpenAI API 키를 얻을 겁니다.



OpenAI?? 지금 Chat GPT로 Chat GPT를 만드려는거?? 라고 하실 수 있는데
(솔직히 다른 대형 회사들도 OpenAI 쓰잖아요.. 쏿)

그냥 OpenAI API 키를 발급받으면 아무런 효과가 없어요.

왜냐? GPT 모델을 사지 않았기 때문입니다.

(사실 우리가 하는 챗봇과의 대화는 누군가가 API 사용료를 내고 있기 때문에
사용이 가능한겁니다.)

근데 솔직히 제가 돈이 어딴어요. 고기 사먹을 돈도 없는데ㅠㅠ

그래서 발견한 것이 무료로 API를 사용하는 대가로, 대화 데이터를 OpenAI 측에 넘겨주는 그런
사기적인 속성을 켜버렸습니다.



아무튼 그렇게 해서 **API키를 사용할 수 있게 되었습니다.**
그리고 VScode를 이용해 기본적인 AI 문법을 작성해봤습니다.

```
import openai

client = openai.OpenAI(api_key="API키 비밀유지 ^^")

messages = [
    {"role": "system", "content": "너는 정보를 전달해주는 AI야."}
]

user_input = input("입력 : ")

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

completion = client.chat.completions.create(
    model="gpt-4o-mini",
    messages=messages
)

ai_response = completion.choices[0].message.content
print("AI의 답 : ", ai_response)
```


대화 test

입력 : 안녕

AI의 답 : 안녕하세요! 어떻게 도와드릴까요?

입력 : 여자친구 만드는 법 알려줘

AI의 답 : 여자친구를 만드는 것은 여러 요소가 필요합니다. 아래의 몇 가지 팁을 참고해 보세요:

1. ****자신을 알아가세요****: 자신의 취향, 관심사, 강점, 약점을 파악하세요. 자신을 이해하는 것이 다른 사람과의 관계에서도 중요한 출발점입니다.
2. ****사회적 네트워크 확장****: 친구와의 모임, 취미 활동, 또는 동호회에 참여하여 새로운 사람들을 만날 기회를 늘리세요. 다양한 사람들과 교류할수록 만남의 기회가 많아집니다.
3. ****좋은 의사소통****: 상대방과 대화할 때 관심과 존중을 보이세요. 질문을 던지고, 이야기 중에 적극적으로 경청하는 것이 중요합니다.
4. ****자신감 가지기****: 자신감 있는 태도는 매력적입니다. 자신의 외모와 성격에 자신감을 가지고 행동하세요.
5. ****상대방에 대한 관심 표현****: 상대방에게 관심을 두고, 그들의 이야기에 귀 기울이며 마음을 열어보세요. 작은 것이라도 상대방에게 호의적인 행동을 보여주는 것이 좋습니다.
6. ****데이트 제안하기****: 서로에 대한 호감이 높아진다면 자연스런게 데이트를 제안해 보세요. 커피나 영화 혹은 함께 할 수 있는 활동으로 시작할 수 있습니다.

하 하 하 하

확실히 AI가 작동되는게 맞는거 같군요.

아무튼 이렇게 텍스트로만 대화를 하면 시각적 감성이 살지가 않잖아요?
그래서 저는 tkinter라는 gui 라이브러리를 이용해서 시각적인
AI 챗봇을 만들겁니다.

```

}

def get_ai_response(user_input):
    messages.append({"role": "user", "content": user_input})

    completion = client.chat.completions.create(
        model="gpt-4o-mini",
        messages=messages
    )

    ai_response = completion.choices[0].message.content
    messages.append({"role": "assistant", "content": ai_response})
    return ai_response

def send_message():
    user_input = entry.get()
    if user_input.strip() == "":
        return

    chat_area.insert(tk.END, "입력 : " + user_input + "\n")
    entry.delete(0, tk.END)

    ai_response = get_ai_response(user_input)
    chat_area.insert(tk.END, "AI의 답 : " + ai_response + "\n\n")
    chat_area.see(tk.END)

root = tk.Tk()
root.title("첫 귀환")

chat_area = scrolledtext.ScrolledText(root, wrap=tk.WORD, width=60, height=20, font=("Arial", 12))
chat_area.pack(padx=10, pady=10)

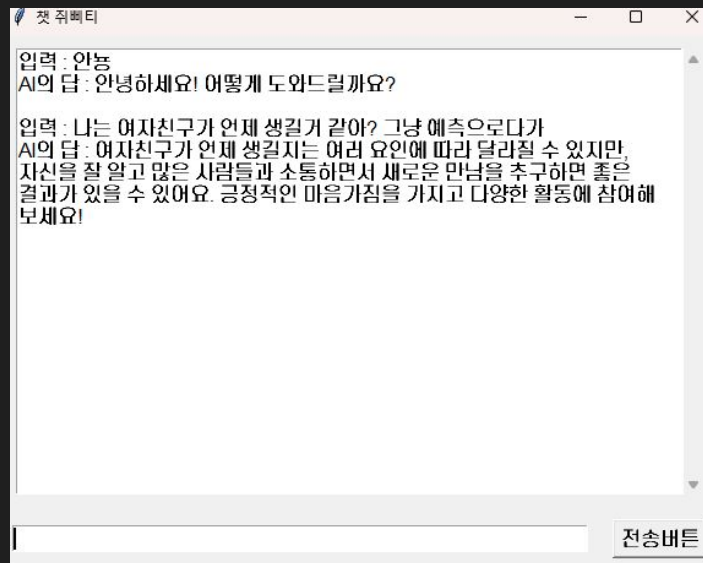
entry = tk.Entry(root, width=50, font=("Arial", 12))
entry.pack(side=tk.LEFT, padx=10, pady=10, fill=tk.X, expand=True)

send_button = tk.Button(root, text="전송 버튼", command=send_message, font=("Arial", 12))
send_button.pack(side=tk.RIGHT, padx=10, pady=10)

root.mainloop()

```

아무튼 이런 분위기로다가 코드가 완성되었습니다.
(보는건 한 순간이지만, 저는 코드를 구상하느라 죽을뻔 했습니다.)
그럼 이제 프로그램을 돌려보아야겠죠?



... 무언가 말하는게 마음에 들지는 않지만 첫 AI 챗봇 개발
성공~