

으뜸  
파이썬



10강 나만의 프로젝트

# 좋은 프로그래밍 실력

- 많은 문제를 해결하는 것

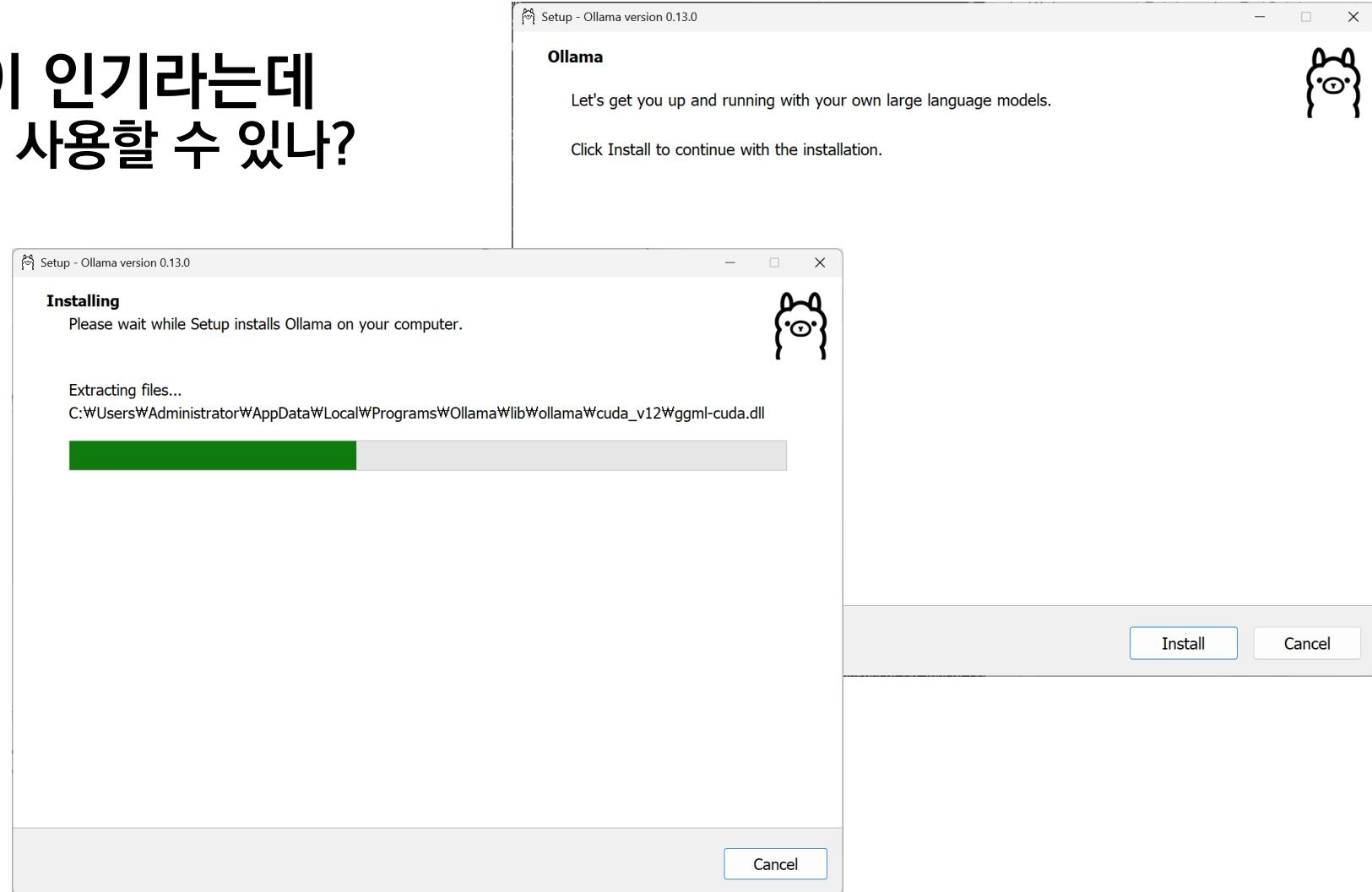
# 다양한 도구를 사용해 보자

- 요즘 인공지능이 인기라는데
  - 내 컴퓨터에서 사용할 수 있나?

<https://ollama.com/download/windows>

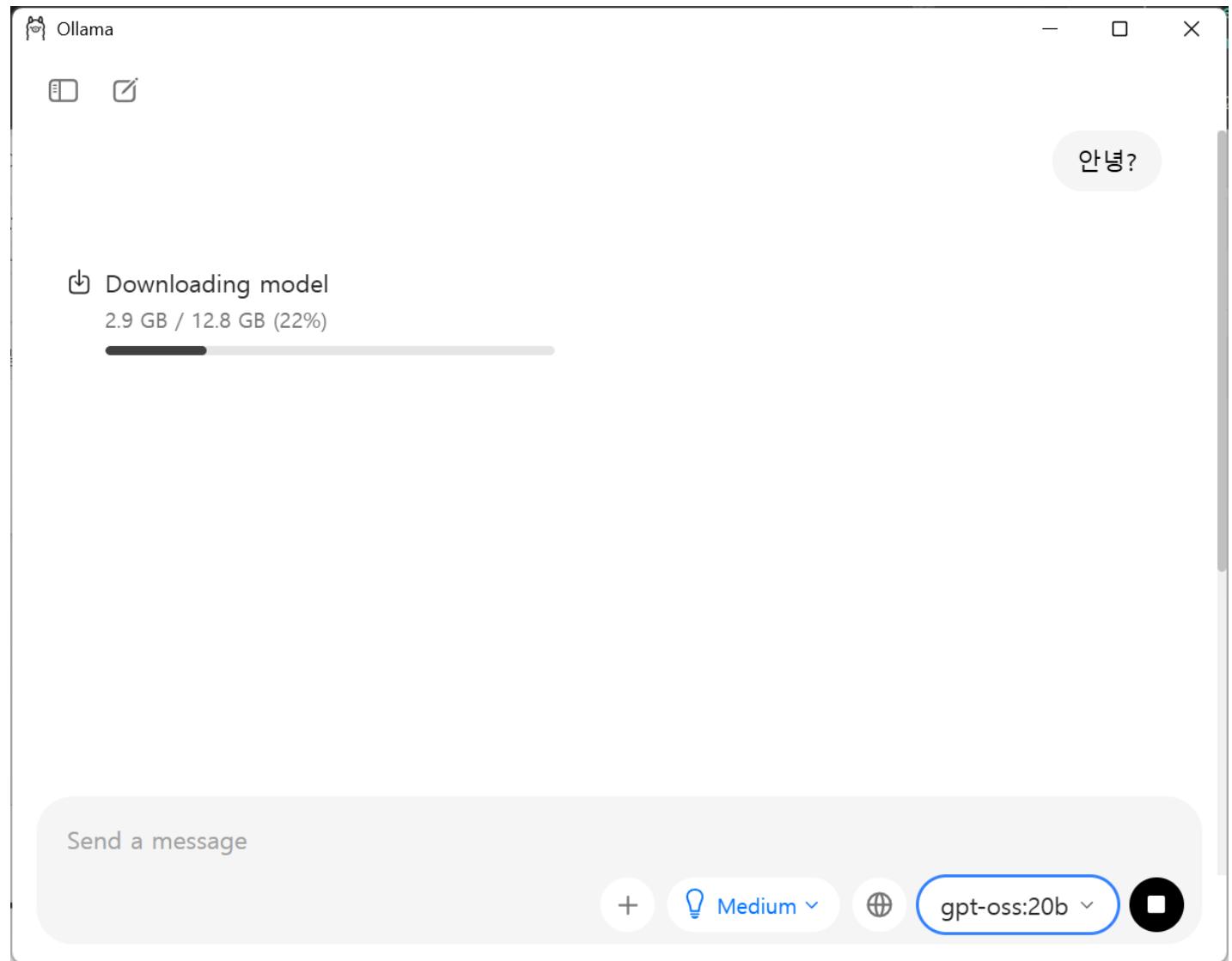
# 다양한 도구를 사용해 보자

- 요즘 인공지능이 인기라는데
  - 내 컴퓨터에서 사용할 수 있나?



# 어떻게 쓰나?

## 일단 말을 걸어 보자 (모델 다운로드)



# 학습된 모델을 가지고 옴

The screenshot shows a window titled "Ollama" with a message input field at the bottom. The message "파이썬이 뭐야?" (What is Python?) has been sent, and the AI is "Thinking for a moment". The response is provided below:

**파이썬이 뭐야?**

파이썬(Python)은 **높은 수준의, 인터프리터형**(컴파일 없이 바로 실행되는) 프로그래밍 언어입니다.

- **가독성이 뛰어나서** 초보자도 배우기 쉽습니다.
- **동적 타이핑과 자동 메모리 관리** 덕분에 코드 작성이 간편합니다.
- **풍부한 표준 라이브러리와 수많은 서드파티 패키지**(NumPy, pandas, Django, Flask 등)를 통해 과학계산, 데이터 분석, 웹 개발, 인공지능, 자동화 등 거의 모든 분야에서 사용됩니다.
- **오픈소스**이어서 누구나 자유롭게 사용·배포·개발할 수 있습니다.

Send a message

+

Medium

gpt-oss:20b

↑

# 영어 연습 프로그램을 만들어 볼까

- 질문을 하고 답을 얻어 보자
  - Ollama 패키지 설치  
% pip install ollama
  - Ollama 모델 로딩  
% ollama pull llama3
  - 물어보고 답하기 구현

# 영어 연습 프로그램을 만들어 볼까

- 질문을 하고 답을 얻어 보자

```
import ollama

def build_prompt():
    return input("prompt: ")

def ask_llm(prompt, model="llama3"):
    response = ollama.generate(
        model=model,
        prompt=prompt
    )
    return response['response']

if __name__ == "__main__":
    prompt = build_prompt()
    result = ask_llm(prompt)
    print(result)
```

```
prompt: Hello!
Hello! It's nice to meet you. Is there something I can help you with, or would you like to chat?
PS D:\ymkangOneDrive\OneDrive\문서\YMKang_Work\수업_실습\2025_2학기\2025_2_Python_문제해결\Codes>
```

```
import customtkinter as ctk
import ollama

def ask_llm():
    prompt = prompt_box.get("1.0", "end").strip()
    result = ollama.generate(model="llama3", prompt=prompt)["response"]
    output_box.delete("1.0", "end")
    output_box.insert("end", result)

ctk.set_appearance_mode("dark")

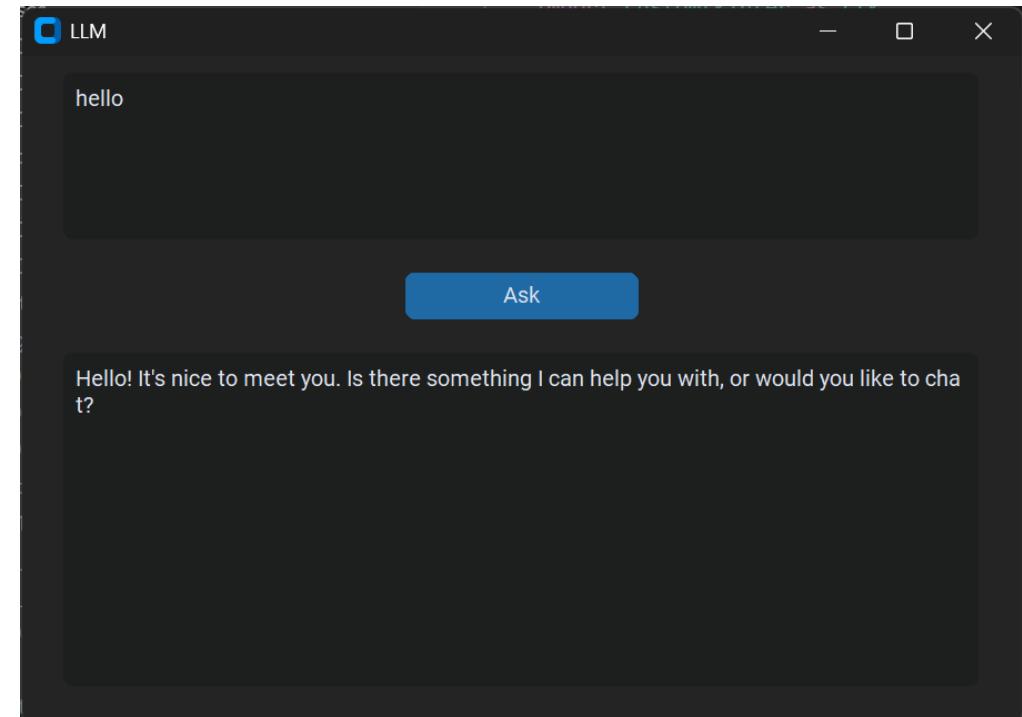
app = ctk.CTk()
app.title("LLM")
app.geometry("600x400")

prompt_box = ctk.CTkTextbox(app, width=550, height=100)
prompt_box.pack(pady=10)

run_btn = ctk.CTkButton(app, text="Ask", command=ask_llm)
run_btn.pack(pady=10)

output_box = ctk.CTkTextbox(app, width=550, height=200)
output_box.pack(pady=10)

app.mainloop()
```



```
import customtkinter as ctk
import ollama

# LLM 호출 함수
def ask_llm(prompt, model="llama3"):
    result = ollama.generate(model=model, prompt=prompt)
    return result["response"]

# 엔터 입력 시 실행되는 함수
def on_enter(event=None):
    user_input = input_box.get().strip()
    if not user_input:
        return

    # 입력창 초기화
    input_box.delete(0, "end")

    # 사용자 메시지 출력
    chat_box.configure(state="normal")
    chat_box.insert("end", f"You: {user_input}\n")
    chat_box.configure(state="disabled")
    chat_box.see("end")

    # LLM 응답 생성
    response = ask_llm(user_input)

    # LLM 응답 출력
    chat_box.configure(state="normal")
    chat_box.insert("end", f"AI: {response}\n\n")
    chat_box.configure(state="disabled")
    chat_box.see("end")

# GUI 설정
ctk.set_appearance_mode("dark")
app = ctk.CTk()
app.title("LLM Chat")
app.geometry("600x500")

# 대화 출력 창
chat_box = ctk.CTkTextbox(app, width=560, height=400, state="disabled")
chat_box.pack(pady=10)

# 입력창 (Entry 사용)
input_box = ctk.CTkEntry(app, width=560)
input_box.pack()
input_box.bind("<Return>", on_enter)

app.mainloop()
```



LLM Chat



You: hello

AI: Hello! It's nice to meet you. Is there something I can help you with, or would you like to chat?

You: 한국어도 하니?

AI: 😊 Yes, I can also speak Korean! 😊 If you want to communicate in Korean, please feel free to ask me any questions or make requests. I'll do my best to respond in Korean.

(Note: Keep in mind that while I can understand and generate text in Korean, my proficiency may not be as high as a native speaker's.)

You: 한국어로 이야기 해줘

AI: 😊

네, 한국어로 이야기를 시작할까요? 무엇에 대해 이야기를 하실 건가요? 🤗

You: 부산에 대해 이야기할까?

AI: Busan! 🎉 I'd be happy to talk about this amazing city in South Korea.

Busan is the second-largest metropolitan area in South Korea, with a population of over 3.5 million people. It's located on the southeastern coast and is known for its vibrant culture, stunning beaches, and rich history.

Here are some interesting facts about Busan:

|

# 이미지 읽어 보기

```
import customtkinter as ctk
from tkinter import filedialog
from PIL import Image, ImageTk

def load_image(path):
    global photo_image

    img = Image.open(path)
    photo_image = ImageTk.PhotoImage(img)
    img_label.configure(image=photo_image)

def open_image():
    path = filedialog.askopenfilename(
        title="이미지 선택",
        filetypes=[("Image files",
                    "*.png *.jpg *.jpeg *.bmp *.gif")]
    )
    if path:
        load_image(path)
```

```
# -----
# 메인 윈도우 구성
# -----
ctk.set_appearance_mode("system")
ctk.set_default_color_theme("blue")

root = ctk.CTk()
root.title("간단 이미지 로더")
root.geometry("800x600")

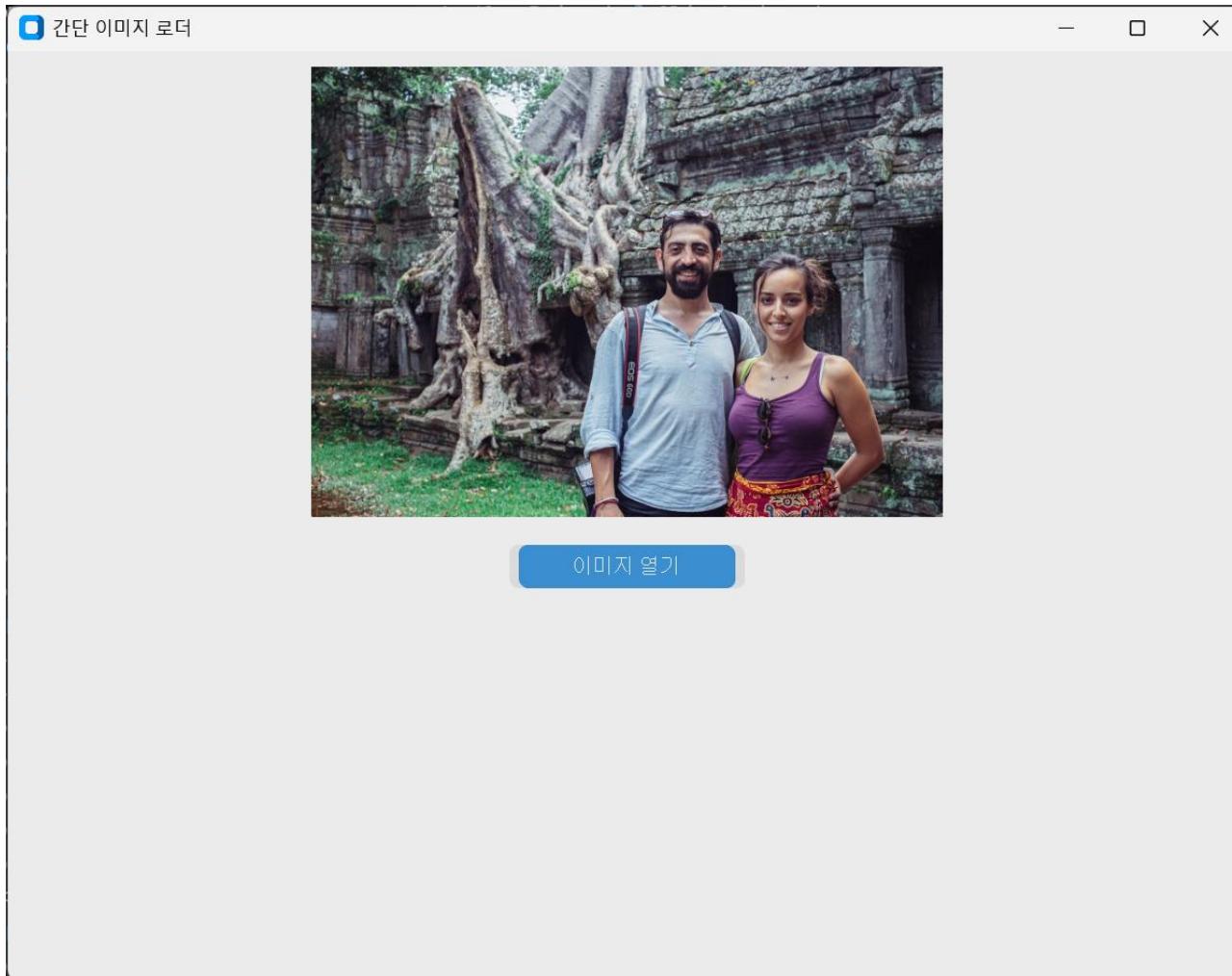
img_label = ctk.CTkLabel(root, text="")
img_label.pack(pady=10)

btn_frame = ctk.CTkFrame(root)
btn_frame.pack(pady=8)

open_btn = ctk.CTkButton(btn_frame, text="이미지 열기",
                        command=open_image)
open_btn.pack(side="left", padx=6)

root.mainloop()
```

# 이미지 읽어 보기



# 배경제거

```
pip install rembg onnxruntime
```

# 배경제거

```
import customtkinter as ctk
from tkinter import filedialog
from PIL import Image
from rembg import remove
import os

# GPU 없어도 오류 안 나게 설정
os.environ["REMGB_BACKEND"] = "cpu"

ctk.set_appearance_mode("system")
ctk.set_default_color_theme("blue")

root = ctk.CTk()
root.title("배경 제거 + 저장")
root.geometry("800x600")

# 이미지 표시 라벨
label = ctk.CTkLabel(root, text="이미지를 열어 주세요")
label.pack(expand=True, pady=20)

# 현재 이미지 보관
img = None
```

# 배경제거

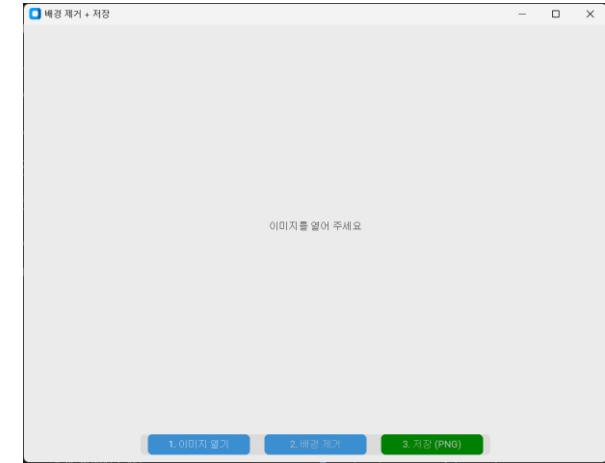
```
# 버튼 3개만
frame = ctk.CTkFrame(root)
frame.pack(pady=10)

btn_load = ctk.CTkButton(frame, text="1. 이미지 열기", command=None, width=140)
btn_load.pack(side="left", padx=10)

btn_remove = ctk.CTkButton(frame, text="2. 배경 제거", command=None, state="disabled",
                           width=140)
btn_remove.pack(side="left", padx=10)

btn_save = ctk.CTkButton(frame, text="3. 저장 (PNG)", command=None, state="disabled",
                           width=140, fg_color="green")
btn_save.pack(side="left", padx=10)

root.mainloop()
```



```
def load():
    global img
    path = filedialog.askopenfilename(filetypes=[("사진", "*.jpg *.jpeg *.png")])
    if path:
        img = Image.open(path).convert("RGBA")
        show(img)
        btn_remove.configure(state="normal")
        btn_save.configure(state="normal")

def remove_bg():
    global img
    if img:
        result = remove(img)          # 배경 제거
        img = result
        show(result)

def save():
    if img and img.mode == "RGBA":
        path = filedialog.asksaveasfilename(defaultextension=".png",
                                              filetypes=[("PNG 파일", "*.png")] )
    if path:
        img.save(path)
        ctk.CTkToplevel(root).withdraw()
```

## 주요 3개 기능

# 버튼과 기능 연결

```
# 버튼 3개만
frame = ctk.CTkFrame(root)
frame.pack(pady=10)

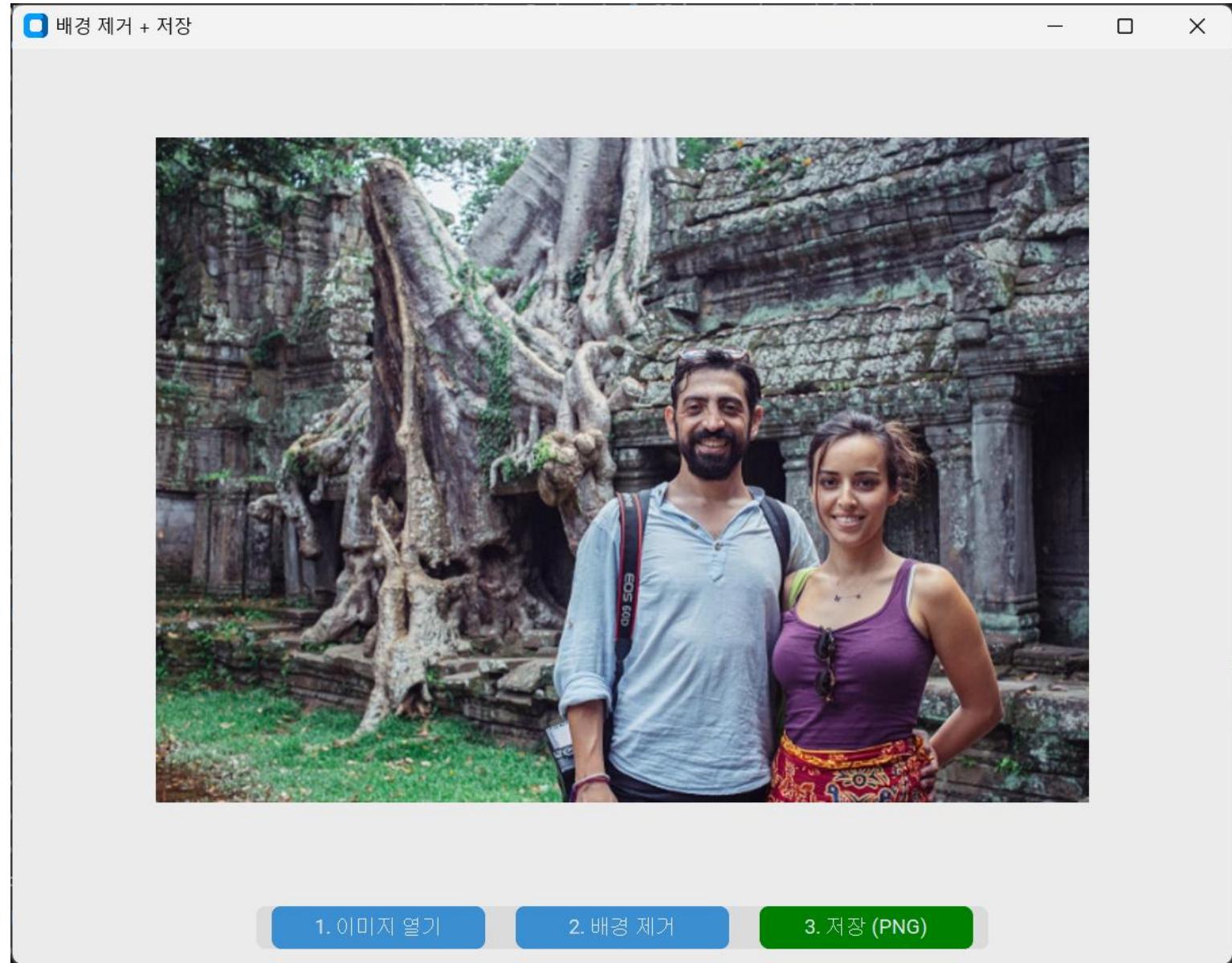
btn_load = ctk.CTkButton(frame, text="1. 이미지 열기", command=load, width=140)
btn_load.pack(side="left", padx=10)

btn_remove = ctk.CTkButton(frame, text="2. 배경 제거", command=remove_bg,
                           state="disabled", width=140)
btn_remove.pack(side="left", padx=10)

btn_save = ctk.CTkButton(frame, text="3. 저장 (PNG)", command=save, state="disabled",
                        width=140, fg_color="green")
btn_save.pack(side="left", padx=10)

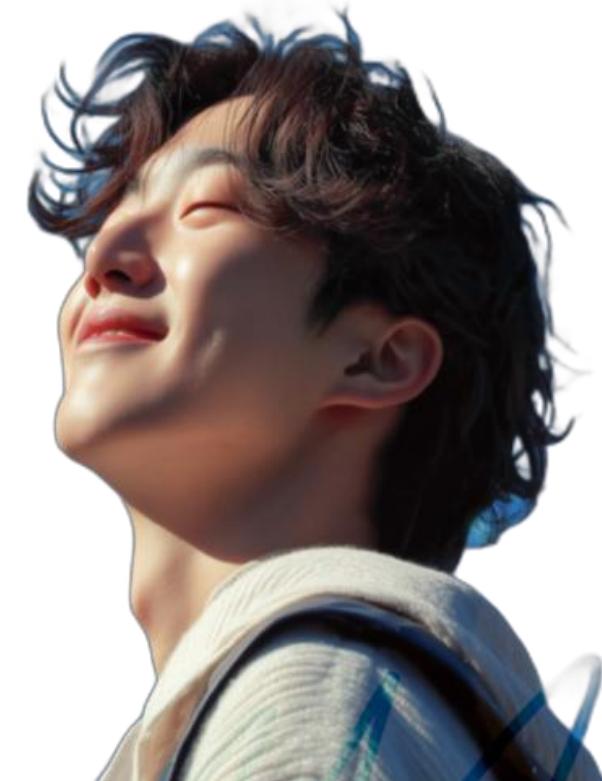
root.mainloop()
```

# 배경제거



# 배경제거

Hello There





# Questions?