

ChatGPT 커스터마이징 & 프로그래밍

GPT 기반 인터랙티브 서비스 개발하기

충북대학교 의과대학
박 승



충북대학교
CHUNGBUK NATIONAL UNIVERSITY

학습 목표

1. ChatGPT 커스터마이징의 기본 개념을 이해하고, 맞춤형 서비스를 설계할 수 있다.
2. 사용자와 상호 작용이 가능한 인터랙티브 서비스를 개발할 수 있다.
3. GPT의 프롬프트 구성 요소와 파라미터의 역할을 명확히 이해한다.
4. 시각적 정보와 텍스트 데이터를 결합한 GPT 서비스를 개발할 수 있다.

ChatGPT 커스터마이징

■ OpenAI 제공 GPT 커스터마이징

- 링크: <https://chatgpt.com/gpts>



GPT

지침, 추가 지식 및 모든 스킬 조합을 결합한 ChatGPT의 맞춤형 버전을 발견하고 만듭니다.

최상위 선택 항목 글쓰기 생산성 연구 및 분석 교육 라이프스타일 프로그래밍

추천

이번 주 선별된 최상위 선택 항목

The image displays four distinct icons arranged in a 2x2 grid. The top-left icon is a red hexagon with a white stylized 'G' shape inside. The top-right icon is a green circle containing a white stylized figure of a person with arms raised. The bottom-left icon is a circular image showing a green, leafy plant growing out of a brown, textured base. The bottom-right icon is an orange circle with a white stylized icon of a book or document with a folded corner.

지금 인기

커뮤니티에서 가장 인기 있는 GPT

- 

image generator
A GPT specialized in generating and refining images with a mix of professional and friendly tone.
작업자: NAIF J ALOTAIBI
- 

Write For Me
Write tailored, engaging content with a focus on quality, relevance and precise word count.
작업자: puzzle.today
- 

Canva
Effortlessly design anything: presentations, logos, social media posts and more.
- 

Scholar GPT
Enhance research with 200M+ resources and built-in critical reading. Access Google Scholar, PubMed, JSTOR, Arxiv...

새 GPT

초안

...

만들기

만들기

구성

+

이름

GPT 이름을 입력하세요

설명

이 GPT 용도에 대한 짧은 설명 추가

지점

이 GPT의 용도는 무엇인가요? 어떻게 작동하나요? 해서는 안 되는 것이 있나요?

대화 스타터

×

지식

지식(Knowledge) 하에 파일을 업로드하면, GPT와의 대화에 파일 콘텐츠가 포함될 수 있습니다. 코드 인터프리터를 사용하면 파일을 다운로드할 수 있습니다.

파일 업로드

기능

☒ 웹 브라우징

☒ DALL·E 이미지 생성

☐ 코드 인터프리터 및 데이터 분석 (?)

작업

새 작업 만들기

미리 보기

메시지 GPT

↑

ChatGPT 커스터마이징

■ 실습 1) 한국어-영어 번역기

< Korean-English Translator
● 초안

만들기 구성

+

이름

설명

지침

1. 한국어 입력이 들어오면 영어로 번역해줘.
2. 영어 입력이 들어오면 한국어로 번역해줘.
3. 최대한 형식적이고, 문법에 맞는 표현을 채택해줘.
4. 한국어나 영어가 아닌 언어의 경우, 번역할 수 없다고 알려줘.

대화 스타터

×

×

미리 보기

충북대학교는 충청북도 청주시 서원구에 있는 대한민국의 4년제 국립 대학이며, 충청북도의 거점국립대학교이다.

1951년 독립 청주초급농과대학으로 개교하여 1953년 청주농과대학으로 승격하고, 1956년 충북대학으로 교명을 변경하였다. 1977년에 공과대학 · 농과대학 · 사범대학 · 사회과학대학 설립이 인가 되면서 국립 충북대학교로 승격됐다. 15개 대학, 일반대학원 및 8개 전문·특수대학원이 개설되어 있다. 교훈은 진리, 정의, 개척이다.



Chungbuk National University is a four-year national university located in Seowon-gu, Cheongju, Chungcheongbuk-do, South Korea, and serves as the flagship national university of Chungcheongbuk-do.

It was established in 1951 as Cheongju Provincial Junior College of Agriculture and was promoted to Cheongju College of Agriculture in 1953. The name was changed to Chungbuk College in 1956, and in 1977, it was elevated to Chungbuk National University with the approval for the establishment of the College of Engineering, College of Agriculture, College of Education, and College of Social Sciences. The university consists of 15 colleges, a general graduate school, and 8 specialized graduate schools. Its motto is Truth, Justice, and Pioneering.

ChatGPT 커스터마이징

■ 실습 1) 한국어-영어 번역기

Korean-English Translator

초안

만들기

구성

+

이름

Korean-English Translator

설명

한국어-영어 간 번역기

지침

1. 한국어 입력이 들어오면 영어로 번역해줘.

2. 영어 입력이 들어오면 한국어로 번역해줘.

3. 최대한 형식적이고, 문법에 맞는 표현을 채택해줘.

4. 한국어나 영어가 아닌 언어의 경우, 번역할 수 없다고 알려줘.

대화 스타터

Hello, User!

미리 보기

Tu seras pour moi unique au monde. Je serai pour toi unique au monde...



죄송하지만, 이 언어는 번역할 수 없습니다.

ChatGPT 커스터마이징

■ 실습 2) 강의계획서 제작 서비스

만들기

구성

+

이름

나만의 강의계획서

설명

강의계획서 자동 제작기

지침

수업 제목이 주어지면, 다음과 같은 내용을 체계적으로 작성해줘.

<요구사항>

1. 강의 목표
2. 주차별 강의 내용
3. 주차별 강의 내용의 세부 내용

대화 스타터

어떤 강의를 계획하시나요?

×

×

수업 제목이 주어지면, 다음과 같은 내용을 체계적으로 작성해줘.

<요구사항>

1. 강의 목표
2. 주차별 강의 내용
3. 주차별 강의 내용의 세부 내용

<참고사항>

1. 강의는 총 16주로 이루어져 있어.
2. 8주차는 중간고사, 16주차는 기말고사가 들어가야 해.
3. 주차별 강의 내용과 세부 내용은 표로 작성해줘.
4. 최대한 체계적이고 직관적인 강의 내용을 구성해줘.
5. 대상 학생은 공학 분야의 학부생이야.

■ 실습 2) 강의계획서 제작 서비스

선형대수학

선형대수학 강의계획서

1. 강의 목표

이 강의는 공학 분야의 학부생들을 대상으로 선형대수학의 기본 개념과 그 응용을 학습하는 것을 목표로 한다. 학생들은 벡터와 행렬, 선형 변환 등 선형대수학의 핵심 주제를 이해하고, 이를 공학 문제 해결에 적용할 수 있는 능력을 기르게 된다. 강의를 끝나면 학생들은 선형대수학의 이론적 배경을 바탕으로 실질적인 문제를 분석하고 해결할 수 있어야 한다.

2. 주차별 강의 내용

주차	강의 내용	세부 내용
1주차	선형대수학 개요	선형대수학의 기본 개념, 벡터와 스칼라의 정의, 벡터 공간
2주차	벡터와 벡터 공간	벡터의 성질, 벡터 공간의 정의, 선형 결합, 선형 독립성, 기저와 차원
3주차	행렬과 행렬 연산	행렬의 정의, 행렬의 덧셈 및 곱셈, 영행렬과 단위행렬, 전치행렬
4주차	행렬식과 역행렬	행렬식의 정의와 성질, 역행렬의 계산, 행렬식과 선형독립성의 관계
5주차	선형 연립방정식	가우스 소거법, 가우스-조던 소거법, 연립방정식의 해, 고유값 문제 개요
6주차	고유값과 고유벡터	고유값, 고유벡터의 정의, 고유값 문제 해결 방법, 대각화 개념
7주차	대각화와 응용	행렬의 대각화, 대각화를 통한 선형 변환의 간소화, 실제 문제에서의 응용
8주차	중간고사	1주차부터 7주차까지의 내용 복습 및 시험
9주차	내적 공간	내적의 정의, 내적 공간의 성질, 직교성, 직교 기저
10주차	정사영과 직교화	정사영 개념, 그람-슈미트 직교화 과정, 최소 제곱 해법
11주차	특이값 분해(SVD)	특이값 분해의 정의와 성질, SVD를 이용한 차원 축소와 데이터 분석
12주차	선형 변환	선형 변환의 정의, 선형 변환의 행렬 표현, 선형 변환의 성질
13주차	행렬의 고급 주제	행렬 분해(Cholesky, LU 분해), 대칭 행렬과 양정치 행렬의 특성
14주차	응용: 공학 문제 해결	선형대수학을 활용한 실질적 문제 해결 (신호 처리, 그래픽스, 머신러닝 등 다양한 공학적 응용)
15주차	기말고사 준비	전체 강의 내용 요약 및 문제 풀이 세션
16주차	기말고사	9주차부터 15주차까지의 내용 복습 및 시험

ChatGPT 커스터마이징

■ 실습 2) 강의계획서 제작 서비스

- 피드백 or 프롬프트 수정하기

1. 강의 내용은 영어로 작성해줘.
2. 응용:공학문제 해결, 기말고사 준비 대신 실제 수업 내용을 작성해 줘.

Week	Lecture Topic	세부 내용
1	Introduction to Linear Algebra	선형대수학의 필요성과 공학적 응용 소개, 벡터와 스칼라의 기본 개념, 벡터 공간의 예시
2	Vectors and Vector Spaces	벡터의 덧셈 및 스칼라 곱, 선형 독립성의 예시, 기저와 차원의 직관적 이해
3	Matrices and Matrix Operations	행렬 연산의 성질, 역행렬이 존재하는 조건, 전치행렬의 특성
4	Determinants and Inverse Matrices	행렬식 계산 방법(2x2, 3x3), 행렬식과 역행렬의 관계, 선형 독립성과의 연관성
5	Systems of Linear Equations	가우스 소거법과 가우스-조던 소거법을 사용한 연립방정식 풀이, 고유값 문제의 중요성에 대한 개요
6	Eigenvalues and Eigenvectors	고유값과 고유벡터 계산법, 행렬 대각화와 공학적 응용 사례
7	Diagonalization and Applications	행렬의 대각화 방법, 신호 처리나 데이터 분석 등 실제 공학적 응용에서의 대각화 활용
8	Midterm Exam	중간고사: 1주차부터 7주차까지의 주요 개념 및 문제 해결
9	Inner Product Spaces	내적 공간의 정의와 성질, 직교성의 개념 및 공학적 응용, 직교 기저의 예시
10	Projections and Orthogonalization	정사영과 그람-슈미트 직교화 과정, 과소결정 시스템에 대한 최소제곱 해법
11	Singular Value Decomposition (SVD)	특이값 분해(SVD)의 개념과 공학적 응용(차원 축소, 데이터 분석, 이미지 압축 등)
12	Linear Transformations	선형 변환의 예(회전, 스케일링, 반사 등), 선형 변환의 행렬 표현과 그 특성
13	Matrix Factorizations	LU 분해와 Cholesky 분해, 대칭행렬과 양정치 행렬의 특성 및 응용
14	Numerical Linear Algebra	대규모 선형 시스템을 푸는 수치적 방법, 반복 기법(Jacobi, Gauss-Seidel), 오차 분석과 행렬 조건수
15	Advanced Topics: Applications in Engineering	선형대수학의 공학적 응용: 신호 처리, 머신러닝, 컴퓨터 그래픽스, 제어 시스템 등 다양한 분야의 실제 문제 해결
16	Final Exam	기말고사: 9주차부터 15주차까지의 주요 개념 및 문제 풀이

■ 실습 3) 그림 생성 서비스

이름

나만의 그림 생성 툴

설명

주어진 키워드에 맞는 그림을 그려줍니다.

지침

키워드와 세부 설명에 맞는 그림을 그려줘.

<주요 사항>

1. 르누아르, 달리, 고흐 중 랜덤으로 한 사람의 화풍으로 그려줘.
2. 답변에 그림과 함께 어떤 화가의 화풍을 차용하였는지 알려 줘.
3. 답변은 한글로 해줘.

대화 스타터

어떤 그림을 그려드릴까요?

×

×

기능

☒ 웹 검색

☒ 캔버스

☒ 4o 이미지 생성

☐ 코드 인터프리터 및 데이터 분석 ?

태양 밑의 꽃밭



이 그림은 르누아르의 화풍을 차용하여 그린 "태양 밑의 꽃밭"입니다. 따뜻한 햇살이 꽃밭을 부드럽게 감싸며, 밝고 화사한 색감의 꽃들이 활짝 핀 모습을 부드러운 붓터치로 표현했습니다.

ChatGPT 커스터마이징

■ 실습 3) 그림 생성 서비스



이 사람의 특성이 잘 묻어나게 르누아르 풍으로 바꾸어줘.



이 이미지는 르누아르의 화풍을 차용하여 부드럽고 따뜻한 색감으로 표현되었습니다.

ChatGPT 커스터마이징

■ 실습 4) 지식 기반 응답 서비스

- 충북대학교 학칙: <https://drive.google.com/drive/folders/10hspG3q7X5H3JMSIQAfDq0kEYyY6pw4b?usp=sharing>

이름

충북대학교 학칙 Q&A

설명

충북대학교 학칙을 알려줍니다.

지침

1. 첨부된 규정집에 포함되어 있지 않은 내용은 없다고 꼭 말해야 해.
2. 답변에는 그 근거 규정을 꼭 명시해줘 (예시: X조 X항).

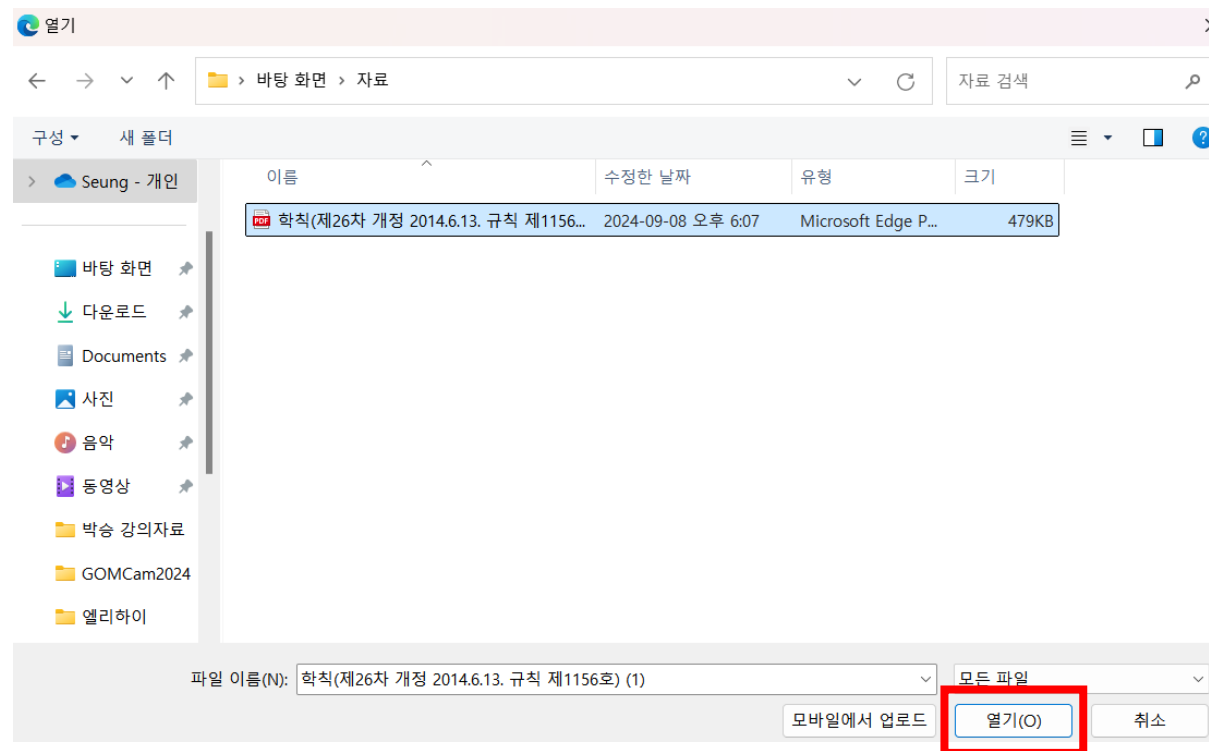
대화 스타터

어떤 것이 궁금하세요?

지식

지식(Knowledge) 하에 파일을 업로드하면, GPT와의 대화에 파일 콘텐츠가 포함될 수 있습니다. 코드 인터프리터를 사용하면 파일을 다운로드할 수 있습니다.

파일 업로드



■ 실습 4) 지식 기반 응답 서비스

2014년도 기준으로 의예과의 입학 정원은 몇명이야?

2014년도 기준으로 충북대학교 의예과의 입학 정원은 34명입니다.

학사 과정의 수업 연한은 몇년이야?

충북대학교 학사과정의 수업 연한은 4년입니다. 다만, 특별시험을 통과하거나 소정의 졸업 학점을 취득한 경우에는 1년 범위 내에서 수업 연한을 단축할 수 있습니다(제32조 1항).

대학	모집단위	입 학 정 원			
		2014학년도	2013학년도	2012학년도	2011학년도
	소계	50	50	50	50
의과 대학	의예과	34	34	25	25
	<의학과>	<25>	<25>	<25>	<25>
	간호학과	60	60	60	-
	소계	94	94	85	25
		<25>	<25>	<25>	<25>

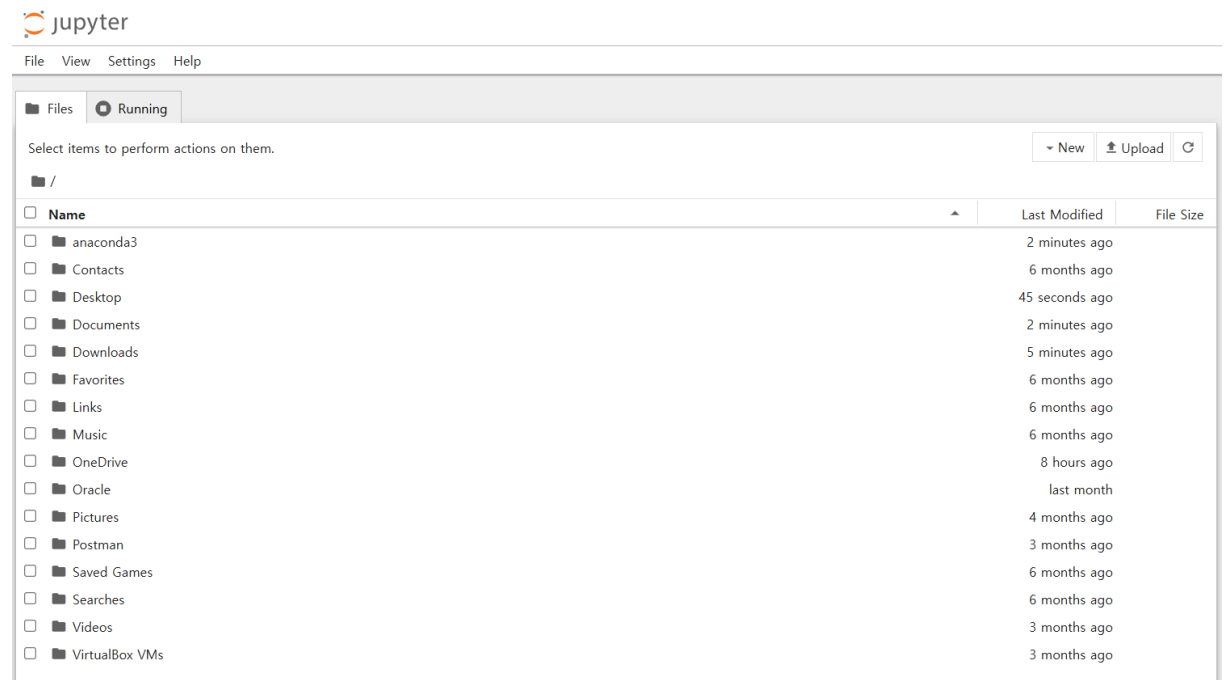
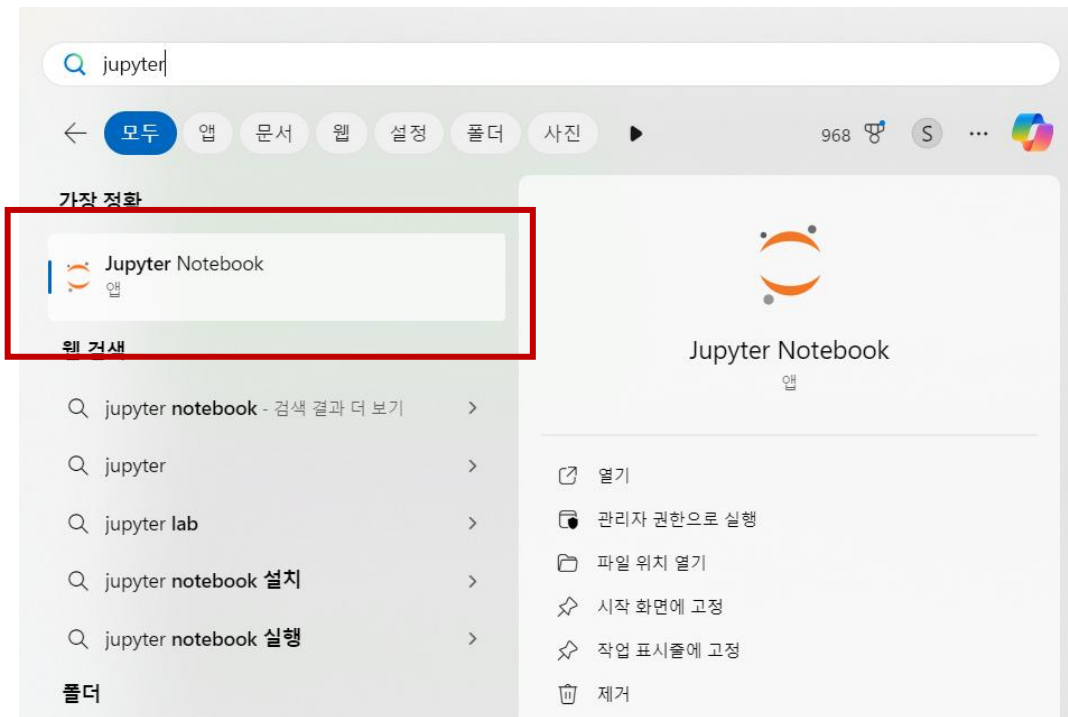
제32조(수업 연한)

① 학사과정의 수업 연한은 4년으로 한다. 다만, 특별 시험을 통과하거나 소정의 졸업 학점을 취득한 자의 수업연한을 1년 범위 내에서 단축할 수 있다. <개정 2011.2.25>

ChatGPT 프로그래밍

■ 프로그래밍 환경 세팅

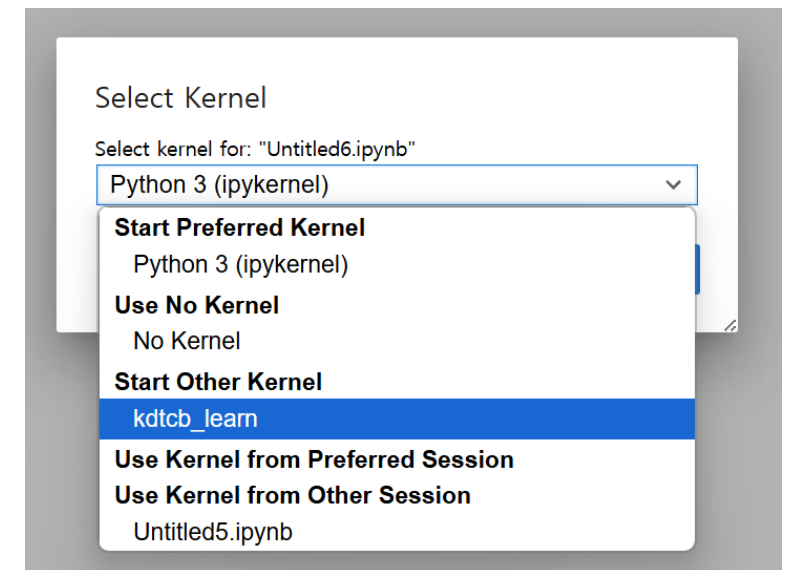
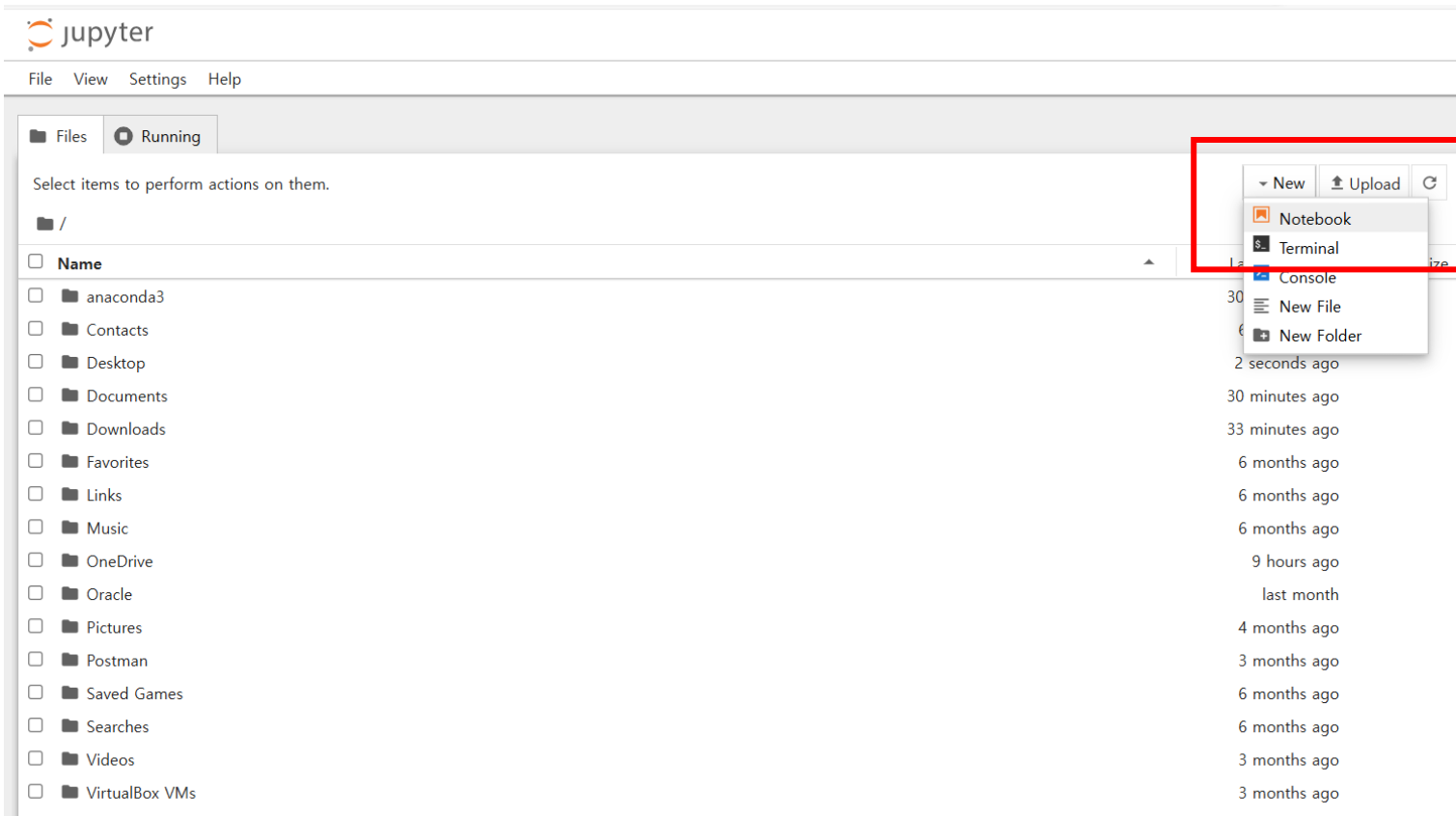
▪ 1. Jupyter Notebook 실행



ChatGPT 프로그래밍

■ 프로그래밍 환경 세팅

▪ 2. 새 notebook 파일 생성



■ 프로그래밍 환경 세팅

▪ 3. openai 라이브러리 설치

```
: !pip install openai
```

```
Requirement already satisfied: openai in c:\users\kdt-19\anaconda3\lib\site-packages (1.78.1)
Requirement already satisfied: anyio<5,>=3.5.0 in c:\users\kdt-19\anaconda3\lib\site-packages (from openai) (4.2.0)
Requirement already satisfied: distro<2,>=1.7.0 in c:\users\kdt-19\anaconda3\lib\site-packages (from openai) (1.9.0)
Requirement already satisfied: httpx<1,>=0.23.0 in c:\users\kdt-19\anaconda3\lib\site-packages (from openai) (0.27.0)
Requirement already satisfied: jiter<1,>=0.4.0 in c:\users\kdt-19\anaconda3\lib\site-packages (from openai) (0.9.0)
Requirement already satisfied: pydantic<3,>=1.9.0 in c:\users\kdt-19\anaconda3\lib\site-packages (from openai) (2.8.2)
Requirement already satisfied: sniffio in c:\users\kdt-19\anaconda3\lib\site-packages (from openai) (1.3.0)
Requirement already satisfied: tqdm>4 in c:\users\kdt-19\anaconda3\lib\site-packages (from openai) (4.66.5)
Requirement already satisfied: typing-extensions<5,>=4.11 in c:\users\kdt-19\anaconda3\lib\site-packages (from openai) (4.11.0)
Requirement already satisfied: idna>=2.8 in c:\users\kdt-19\anaconda3\lib\site-packages (from anyio<5,>=3.5.0->openai) (3.7)
Requirement already satisfied: certifi in c:\users\kdt-19\anaconda3\lib\site-packages (from httpx<1,>=0.23.0->openai) (2024.8.30)
Requirement already satisfied: httpcore==1.* in c:\users\kdt-19\anaconda3\lib\site-packages (from httpx<1,>=0.23.0->openai) (1.0.2)
Requirement already satisfied: h11<0.15,>=0.13 in c:\users\kdt-19\anaconda3\lib\site-packages (from httpcore==1.*->httpx<1,>=0.23.0->openai) (0.14.0)
Requirement already satisfied: annotated-types>=0.4.0 in c:\users\kdt-19\anaconda3\lib\site-packages (from pydantic<3,>=1.9.0->openai) (0.6.0)
Requirement already satisfied: pydantic-core==2.20.1 in c:\users\kdt-19\anaconda3\lib\site-packages (from pydantic<3,>=1.9.0->openai) (2.20.1)
Requirement already satisfied: colorama in c:\users\kdt-19\anaconda3\lib\site-packages (from tqdm>4->openai) (0.4.6)
```

■ 실습 5) GPT 기반 심리 상담 서비스 개발

- 1. 라이브러리 로드 및 api_key 연동
 - api_key는 openai 홈페이지에서 유료 결제로 사용 가능

```
import os
import openai
from openai import OpenAI

openai.api_key = "sk-proj-zcYjewY8DYjQ4gjIKzBeASFA1ru_McHtN4kA50N-QjVLGEXQx9pivZHXI7ste"
```


■ 실습 5) GPT 기반 심리 상담 서비스 개발

▪ 2. system instruction 정의

```
import openai

# 시스템 초기 설정 (역할과 답변 스타일 정의)
messages = [
    {
        "role": "system",
        "content": """
당신은 사려 깊고 공감 능력이 뛰어난 심리상담사입니다.
내담자의 고민을 경청하고, 공감을 표현하세요.
정신건강 관리를 위한 전략과 구체적 해결책을 제시해 내담자의 고민을 해소하도록 돕습니다.
반드시 한국어로 답변하세요.
답변은 20단어 이내로 제한하세요.
        """
    }
]
```

■ 실습 5) GPT 기반 심리 상담 서비스 개발

▪ 2. system instruction 정의

```
# GPT-4와 상호작용하는 함수 정의
def chat_with_gpt(user_input):
    # 사용자 입력 처리
    messages.append({"role": "user", "content": user_input})

    # GPT-4를 사용한 대화 생성 요청
    response = openai.chat.completions.create(
        model="gpt-4",
        messages=messages,
        temperature=0.7,
        max_tokens=256,
        top_p=1,
        frequency_penalty=0,
        presence_penalty=0
    )

    # GPT 응답 추출 및 처리
    assistant_reply = response.choices[0].message.content

    # 응답 메시지 기록
    messages.append({"role": "assistant", "content": assistant_reply})

    return assistant_reply
```

파라미터	의미
Response format	응답의 출력 형식 (텍스트, JSON 등)
Temperature	모델의 창의성 수준 (값이 높을수록 더 창의적인 응답)
Maximum Tokens	응답에서 생성할 최대 토큰의 수
Stop Sequences	특정 문자열이 생성되면 응답을 멈춤
Top P	모델이 다음 단어를 선택하는 확률 (값이 낮을수록 더 결정적인 답변)
Frequency Penalty	같은 단어의 반복 사용을 방지하는 페널티
Presence Penalty	새로운 주제를 더 제시하게 하는 페널티

■ 실습 5) GPT 기반 심리 상담 서비스 개발

▪ 3. 입출력 파트 구현

```
# 사용자 입력을 받아 지속적인 대화 진행
while True:
    user_input = input("You: ")

    # 종료 명령 확인 및 처리
    if user_input.lower() == 'exit':
        print("종료합니다.")
        break

    # GPT 응답 출력
    response = chat_with_gpt(user_input)
    print("GPT:", response)
```

You: 안녕 GPT야. 나 우울해.

GPT: 안녕하세요. 당신의 마음을 이해합니다. 저와 함께 얘기하면서 감정을 정리해봐요.

You: 파이썬 프로그래밍이 너무 어려워.

GPT: 어려움을 겪을 때도 있어요. 천천히 한 단계씩 해나가보면 어떨까요? 함께 해결해 나가요.

You: ↑↓ for history. Search history with c-↑/c-↓

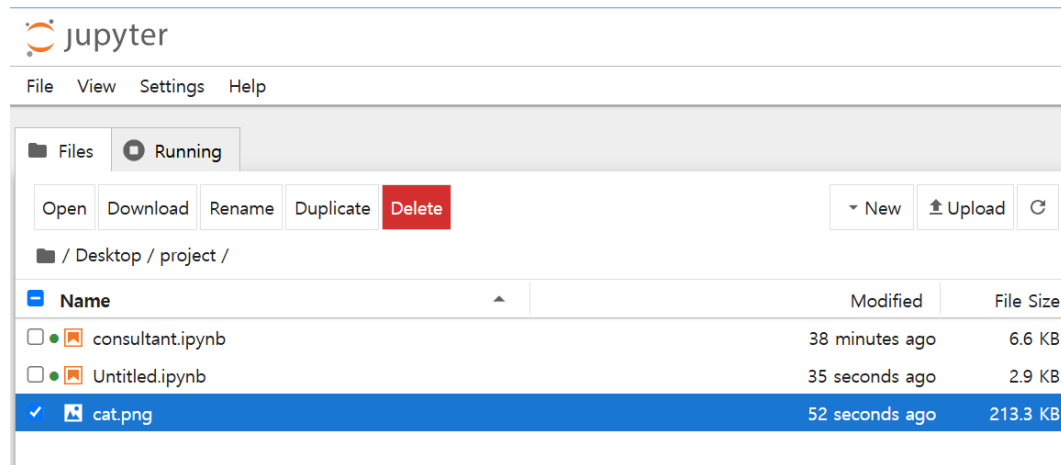
■ 실습 6) 영상 인식 및 대화 서비스 개발

- 1. 라이브러리 로드 및 api_key 연동

```
import base64
import requests

api_key = "sk-proj-zcYjewY8DYjQ4gjIKzBeASFA1ru_McHtN4kA50N-QjVLGEXQx9pivZHXI7ste6b0I7y0IT8613"
```

- 2. 프로젝트 폴더 내 "cat.png" 파일 넣기



The image shows the JupyterLab interface. At the top, there's a 'jupyter' logo and a menu bar with 'File', 'View', 'Settings', and 'Help'. Below the menu bar, there are two tabs: 'Files' and 'Running'. The 'Files' tab is active, showing a file browser. The current path is '/ Desktop / project /'. There are buttons for 'Open', 'Download', 'Rename', 'Duplicate', 'Delete', 'New', 'Upload', and a refresh icon. A table lists the files in the directory:

Name	Modified	File Size
consultant.ipynb	38 minutes ago	6.6 KB
Untitled.ipynb	35 seconds ago	2.9 KB
cat.png	52 seconds ago	213.3 KB

The 'cat.png' file is selected, indicated by a blue highlight and a checkmark in the first column.

■ 실습 6) 영상 인식 및 대화 서비스 개발

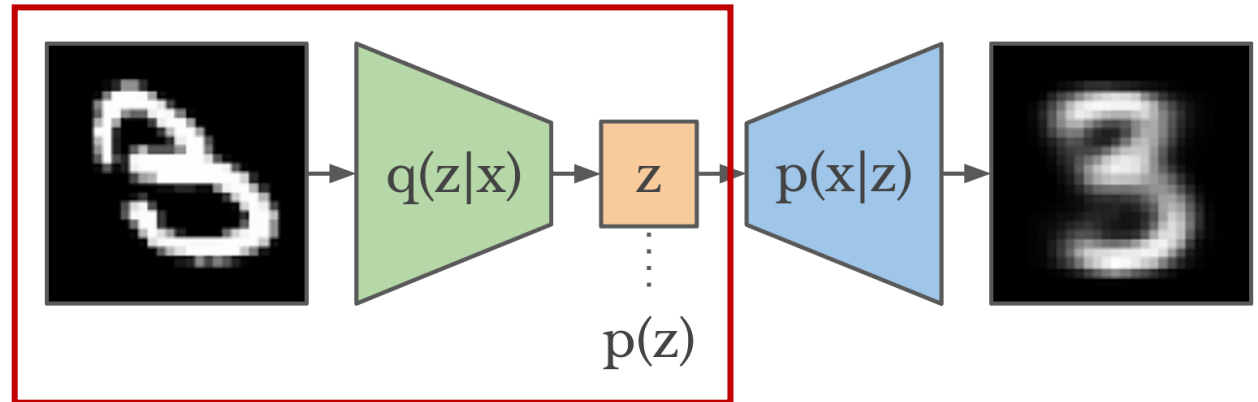
▪ 3. base64를 활용한 영상 특징 인코딩 및 프롬프트 설정

```
# 이미지를 base64로 인코딩하는 함수
def encode_image(image_path):
    with open(image_path, "rb") as img_file:
        return base64.b64encode(img_file.read()).decode('utf-8')

# 이미지 경로
image_path = "./cat.png"
base64_image = encode_image(image_path)

# OpenAI API 요청 헤더
headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {api_key}"
}

# 요청 데이터(payload) 설정
payload = {
    "model": "gpt-4o",
    "messages": [{
        "role": "user",
        "content": [
            {"type": "text", "text": "이 이미지는 어떤 이미지야?"},
            {"type": "image_url",
             "image_url": {"url": f"data:image/jpeg;base64,{base64_image}"}}
        ]
    }],
    "max_tokens": 300
}
```



■ 실습 6) 영상 인식 및 대화 서비스 개발

- 4. 출력 파트 구현

```
# API 요청 및 응답 처리
response = requests.post(
    "https://api.openai.com/v1/chat/completions",
    headers=headers,
    json=payload
)

# 응답 결과 출력
if response.status_code == 200:
    answer = response.json()['choices'][0]['message']['content']
    print("이미지 분석 결과:", answer)
else:
    print("요청 실패:", response.status_code, response.text)
```



이미지 분석 결과: 이 이미지는 고양이의 옆모습을 담고 있습니다.

고양이는 위를 바라보고 있으며, 털이 복슬복슬하고 눈이 초록색입니다.
배경은 단색입니다.

■ 실습 7) Visual Question Answering (VQA) 서비스 개발

- 1. 라이브러리 로드 및 api_key 연동

```
import base64  
import requests
```

```
api_key = "sk-proj-zcYjewY8DYjQ4gjIKzBeASFA1ru_McHtN4kA50N-QjVLGEXQx9pivZHxI7ste6b0I7y01T8613"
```

■ 실습 7) Visual Question Answering (VQA) 서비스 개발

▪ 2. 프롬프트 설정 및 대화 파트 구현

- 사용자 입력을 user_question 으로 프롬프트에 전달

```
# 이미지를 base64로 인코딩하는 함수
def encode_image(image_path):
    with open(image_path, "rb") as img_file:
        return base64.b64encode(img_file.read()).decode('utf-8')

# 이미지 경로 설정
image_path = "./cat.png"
base64_image = encode_image(image_path)

# 사용자로부터 이미지 관련 질문 입력
user_question = input("이미지에 대해 궁금한 질문을 입력하세요: ")
```

```
# OpenAI API 요청 헤더
headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {api_key}"
}

# 요청 payload 구성 (Visual Question Answering)
payload = {
    "model": "gpt-4o",
    "messages": [
        {
            "role": "user",
            "content": [
                {"type": "text", "text": user_question},
                {"type": "image_url",
                 "image_url": {"url": f"data:image/jpeg;base64,{base64_image}"}}
            ]
        }
    ],
    "max_tokens": 300
}
```


■ 실습 7) Visual Question Answering (VQA) 서비스 개발

▪ 3. 출력 파트 구현

```
# OpenAI API에 요청 보내기
response = requests.post(
    "https://api.openai.com/v1/chat/completions",
    headers=headers,
    json=payload
)

# 응답 결과 출력
if response.status_code == 200:
    answer = response.json()['choices'][0]['message']['content']
    print("질문에 대한 답변:", answer)
else:
    print("요청 실패:", response.status_code, response.text)
```



이미지에 대해 궁금한 질문을 입력하세요: 이 고양이의 색은 뭐야?

질문에 대한 답변: 고양이는 갈색과 회색의 줄무늬 털을 가지고 있으며, 턱 밑과 목 주변은 흰색입니다.

이미지에 대해 궁금한 질문을 입력하세요: 여기서 어떤 동물이 어디를 바라보고 있어?

질문에 대한 답변: 이 이미지에는 고양이가 위쪽을 바라보고 있습니다.

■ 실습 8) Ghibli 스타일 변환 서비스 개발

- 1. 라이브러리 로드, api_key 연동, 입력 영상 설정

```
import base64
import requests
import openai
from PIL import Image
from io import BytesIO
from IPython.display import display

# API 키 설정
api_key = "sk-proj-zcYjewY8DYjQ4gjIKzBeASFA1ru_McHtN4kA50N-QjVLGEXQx9pivZHXI7ste6b0I
client = openai.OpenAI(api_key=api_key)
```

```
# 이미지를 base64로 인코딩하는 함수
def encode_image(image_path):
    with open(image_path, "rb") as img_file:
        return base64.b64encode(img_file.read()).decode('utf-8')

# 이미지 경로 설정 (상대 경로)
image_path = "./cat.png"
base64_image = encode_image(image_path)
```

■ 실습 8) Ghibli 스타일 변환 서비스 개발

▪ 2. GPT를 활용한 영상 특징 추출

```
# OpenAI API 요청 헤더
headers = {
    "Content-Type": "application/json",
    "Authorization": f"Bearer {api_key}"
}

# GPT-4o를 통해 원본 이미지를 묘사 받기
payload_gpt = {
    "model": "gpt-4o",
    "messages": [
        {
            "role": "user",
            "content": [
                {"type": "text", "text": "Describe this image in detail."},
                {"type": "image_url",
                 "image_url": {"url": f"data:image/jpeg;base64,{base64_image}"}}
            ]
        }
    ],
    "max_tokens": 300
}

response_gpt = requests.post(
    "https://api.openai.com/v1/chat/completions",
    headers=headers,
    json=payload_gpt
)

if response_gpt.status_code == 200:
    description = response_gpt.json()['choices'][0]['message']['content']
else:
    print("GPT 요청 실패:", response_gpt.status_code, response_gpt.text)
    exit()

description
```

"The image depicts a close-up profile of a fluffy cat with medium to long hair. The cat is looking upwards, displaying its prominent whiskers and a well-defined jawline. Its fur is a mix of brown, gold, and black hues, creating a tabby-like pattern. The cat's eyes are bright green, standing out against its fur. Its ears are upright, and there is a slight tilt to its head, suggesting attentiveness or curiosity. The background is a soft, blurred gray, which contrasts with the sharp detail and vivid colors of the cat."

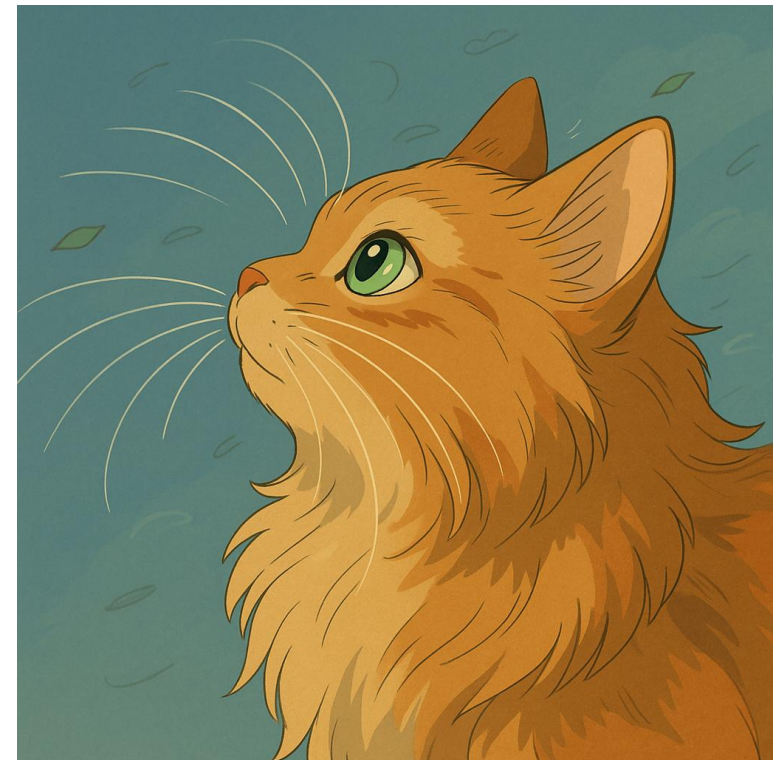
ChatGPT 프로그래밍

- 실습 8) Ghibli 스타일 변환 서비스 개발
 - 3. DALL-E를 활용한 영상 생성

```
# 프롬프트 정의
prompt = (
    "A cel-shaded, hand-drawn 2D animation in the Studio Ghibli style. "
    + description
)

# 이미지 생성 요청
response = client.images.generate(
    model="dall-e-3", # 최신 DALL-E 모델
    prompt=prompt,
    size="1024x1024",
    quality="standard",
    n=1
)

# 생성된 이미지 URL 출력
image_url = response.data[0].url
print("Generated image URL:", image_url)
```



감사합니다

Q&A



충북대학교
CHUNGBUK NATIONAL UNIVERSITY