

CHATGPT API 시작하기

In [1]: `!pip install openai`

```
Requirement already satisfied: openai in /usr/local/lib/python3.11/dist-packages (1.69.0)
Requirement already satisfied: anyio<5,>=3.5.0 in /usr/local/lib/python3.11/dist-packages (from openai) (4.9.0)
Requirement already satisfied: distro<2,>=1.7.0 in /usr/local/lib/python3.11/dist-packages (from openai) (1.9.0)
Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.11/dist-packages (from openai) (0.28.1)
Requirement already satisfied: jiter<1,>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from openai) (0.9.0)
Requirement already satisfied: pydantic<3,>=1.9.0 in /usr/local/lib/python3.11/dist-packages (from openai) (2.11.0)
Requirement already satisfied: sniffio in /usr/local/lib/python3.11/dist-packages (from openai) (1.3.1)
Requirement already satisfied: tqdm>4 in /usr/local/lib/python3.11/dist-packages (from openai) (4.67.1)
Requirement already satisfied: typing-extensions<5,>=4.11 in /usr/local/lib/python3.11/dist-packages (from openai) (4.13.0)
Requirement already satisfied: idna>=2.8 in /usr/local/lib/python3.11/dist-packages (from anyio<5,>=3.5.0->openai) (3.10)
Requirement already satisfied: certifi in /usr/local/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->openai) (2025.1.31)
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.11/dist-packages (from httpx<1,>=0.23.0->openai) (1.0.7)
Requirement already satisfied: h11<0.15,>=0.13 in /usr/local/lib/python3.11/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->openai) (0.14.0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.9.0->openai) (0.7.0)
Requirement already satisfied: pydantic-core==2.33.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.9.0->openai) (2.33.0)
Requirement already satisfied: typing-inspection>=0.4.0 in /usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.9.0->openai) (0.4.0)
```

```
In [2]: from openai import OpenAI
        from getpass import getpass

        # API 키를 안전하게 입력받기
        api_key = getpass("OpenAI API Key를 입력하세요: ")
        client = OpenAI(api_key=api_key)
```

OpenAI API Key를 입력하세요:

```
In [3]: # gpt-4o 모델을 사용하여 챗 메시지 생성
        response = client.chat.completions.create(
            model="gpt-4o-mini",
            messages=[
                {"role": "user", "content": "OpenAI를 어떻게 사용할 수 있을까?"}
            ]
        )

        # 응답 메시지 출력
        print(response.choices[0].message.content)
```

OpenAI를 사용하는 방법에는 여러 가지가 있습니다. 주로 다음과 같은 방법을 통해 Open AI의 제품과 서비스를 활용할 수 있습니다.

1. ****API 사용****: OpenAI는 GPT, DALL-E 같은 모델에 대한 API를 제공합니다. 개발자는 이 API를 통해 텍스트 생성, 이미지 생성 등을 자신의 애플리케이션에 통합할 수 있습니다. API를 사용하려면 OpenAI 웹사이트에서 계정을 만들고 API 키를 받아야 합니다.
2. ****ChatGPT 사용****: OpenAI의 ChatGPT는 웹 기반 애플리케이션으로, 사용자가 직접 대화형 AI와 상호작용할 수 있게 해줍니다. 다양한 질문을 하거나, 글쓰기, 문제 해결 등을 요청할 수 있습니다.
3. ****플러그인 및 통합****: ChatGPT는 다양한 플러그인과 통합할 수 있어, 사용자가 필요에 따라 기능을 확장할 수 있습니다. 예를 들어, 일정 관리, 쇼핑, 여행 계획 등에 도움을 받을 수 있는 플러그인을 사용할 수 있습니다.
4. ****커스텀 모델 개발****: OpenAI의 다양한 도구를 활용하여, 특정 목적에 맞는 커스텀 모델을 개발할 수도 있습니다. 이는 특정 데이터셋을 기반으로 훈련하여, 특정 분야에 특화된 모델을 만드는 것을 포함합니다.
5. ****학습 리소스 이용****: OpenAI의 공식 문서, 튜토리얼, 커뮤니티 포럼 등을 통해 모델을 활용하는 방법에 대한 다양한 자료와 도움을 받을 수 있습니다.

OpenAI의 기술을 활용하면 자연어 처리, 이미지 생성, 데이터 분석 등 다양한 분야에서 혁신적인 솔루션을 만들 수 있습니다. 사용 목적에 맞는 방법을 선택하여 활용해 보세요!

```
In [4]: # dotenv 설치
!pip install python-dotenv
```

Collecting python-dotenv

Downloading python_dotenv-1.1.0-py3-none-any.whl.metadata (24 kB)

Downloading python_dotenv-1.1.0-py3-none-any.whl (20 kB)

Installing collected packages: python-dotenv

Successfully installed python-dotenv-1.1.0

```
In [5]: import os
from openai import OpenAI
from dotenv import load_dotenv

# .env 파일 불러오기
load_dotenv("chatgpt.env")

# API 키 환경 변수에서 가져오기
api_key = os.getenv("API_KEY")

# OpenAI 클라이언트 초기화
client = OpenAI(api_key=api_key)

# gpt-4o 모델을 사용하여 챗 메시지 생성
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "user", "content": "OpenAI API는 무엇일까요?"}
    ]
)

# 응답 메시지 출력
print(response.choices[0].message.content)
```

OpenAI API는 개발자와 기업이 OpenAI의 인공지능 모델을 쉽게 사용할 수 있도록 제공하는 프로그래밍 인터페이스입니다. 이를 통해 사용자는 자연어 처리, 텍스트 생성, 번역, 감정 분석 등 다양한 AI 작업을 수행할 수 있는 기능을 애플리케이션에 통합할 수 있습니다. OpenAI의 대표적인 모델인 GPT 시리즈를 포함하여 여러 고급 AI 기능을 API 형태로 제공하며, 사용자는 이를 통해 자신만의 커스텀 애플리케이션을 구현할 수 있습니다. OpenAI API는 일반적으로 RESTful API 형태로 제공되며, 다양한 프로그래밍 언어와 쉽게 통합될 수 있도록 설계되었습니다.

기본 API 호출 코드 예제

```
In [6]: from openai import OpenAI
import os
from dotenv import load_dotenv

# 환경 변수 로드
# .env 파일 불러오기
load_dotenv("chatgpt.env")

# API 키 환경 변수에서 가져오기
api_key = os.getenv("API_KEY")

# OpenAI 클라이언트 초기화
client = OpenAI(api_key=api_key)

# ChatGPT API 호출
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": "너는 여행 전문가란다."},
        {"role": "user", "content": "여행 일정을 세워줄 수 있니?"}
    ],
    temperature=0.7,
    max_tokens=100
)

# 응답 출력
print(response.choices[0].message.content)
```

물론입니다! 여행 일정을 세우기 위해 몇 가지 질문이 필요해요.

1. 여행 목적지는 어디인가요?
2. 여행 기간은 얼마나 되나요?
3. 특별히 방문하고 싶은 장소나 관심 있는 활동이 있나요?
4. 여행 예산은 어느 정도인가요?
5. 여행에 동행하는 사람은 몇 명인가요?
6. 어떤 여행 스타일을 선호하시나요? (예: 휴양, 문화 탐방, 모

```
In [7]: from openai import OpenAI
import os
from dotenv import load_dotenv

# 환경 변수 로드
load_dotenv("chatgpt.env")

# API 키 환경 변수에서 가져오기
api_key = os.getenv("API_KEY")

# OpenAI 클라이언트 초기화
client = OpenAI(api_key=api_key)
```

```

# 대화 메시지를 저장할 리스트 초기화
messages = [
    {"role": "system", "content": "너는 여행 전문가란다."},
    {"role": "user", "content": "여행 일정을 세워줄 수 있니?"}
]

# 대화 반복 실행 (2~3회)
for _ in range(2): # 원하는 횟수만큼 반복
    # ChatGPT API 호출
    response = client.chat.completions.create(
        model="gpt-4o",
        messages=messages,
        temperature=0.7,
        max_tokens=100
    )

    # 응답을 대화 메시지 리스트에 추가
    assistant_message = response.choices[0].message.content
    print(assistant_message)

    # User의 새로운 질문을 받아서 추가
    user_input = input("추가 질문을 입력하세요: ") # 사용자의 다음 질문 입력 받기
    messages.append({"role": "assistant", "content": assistant_message})
    messages.append({"role": "user", "content": user_input})

# 최종 출력
print("최종 대화 내용:")
for message in messages:
    print(f"{message['role']}: {message['content']}")

```

물론입니다! 여행 일정을 세우기 위해 몇 가지 정보를 좀 더 알려주시면 좋겠습니다. 예를 들어, 여행 목적지, 여행 기간, 선호하는 활동(예: 문화 탐방, 자연 탐험, 음식 여행 등), 예산, 여행 인원 등을 알려주시면 맞춤형 일정을 계획하는 데 도움이 됩니다. 어떤 여행을 계획하고 계신가요?

추가 질문을 입력하세요: 제주도

제주도로의 여행은 훌륭한 선택입니다! 제주도는 아름다운 자연 경관과 다양한 문화적 명소로 유명하죠. 여행 일정을 예시로 제안해드리겠습니다. 참고하셔서 필요에 따라 조정하시면 좋겠습니다.

3박 4일 제주도 여행 일정

1일차: 제주시 탐방

- **오전**
 - 제주 국제공항 도착
 - 용

추가 질문을 입력하세요: 맛집

최종 대화 내용:

system: 너는 여행 전문가란다.

user: 여행 일정을 세워줄 수 있니?

assistant: 물론입니다! 여행 일정을 세우기 위해 몇 가지 정보를 좀 더 알려주시면 좋겠습니다. 예를 들어, 여행 목적지, 여행 기간, 선호하는 활동(예: 문화 탐방, 자연 탐험, 음식 여행 등), 예산, 여행 인원 등을 알려주시면 맞춤형 일정을 계획하는 데 도움이 됩니다. 어떤 여행을 계획하고 계신가요?

user: 제주도

assistant: 제주도로의 여행은 훌륭한 선택입니다! 제주도는 아름다운 자연 경관과 다양한 문화적 명소로 유명하죠. 여행 일정을 예시로 제안해드리겠습니다. 참고하셔서 필요에 따라 조정하시면 좋겠습니다.

3박 4일 제주도 여행 일정

1일차: 제주시 탐방

- **오전**
 - 제주 국제공항 도착
 - 용

user: 맛집

In []: