

OpenAI의 모델들의 이름들과 가격은 아래의 주소에서 확인하세요.

<https://openai.com/api/pricing/>

## ✓ 1. 셋팅하기

코랩에서 OpenAI ChatGPT API를 실습하려면 openai 라이브러리를 설치해야 합니다. 다음 명령어를 입력합니다

```
!pip install openai
```

```
Requirement already satisfied: openai in /usr/local/lib/python3.11/dist-packages (1.66.3)
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) (2.10.6)
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) (4.12.2)
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==2.27.2 in /usr/local/lib/python3.11/dist-packages (from pydantic<3,>=1.9.0->openai) (2.27.2)
```

마지막 부분과 같이 'Successfully installed openai'라는 문구를 확인했다면 import openai를 통해 라이브러리를 불러옵니다. 그 후 openai.api\_key의 값으로 OpenAI API 키 발급하는 사이트(<https://platform.openai.com/account/api-keys>) 에서 얻은 키 값을 입력합니다.

```
import openai
```

```
client = openai.OpenAI(api_key="sk-proj-jJu3_uqdIS19soH7_PSyfYsPBjNVfc04cUBMCxyBMn9UMy9in6vP_DBEyGJPHwekEP9Mw6pQS0T3BIbkFJz6A0Qa4sk0u749VVlgVD_BP4GcXUW")
```

사용량 확인하는 방법

<https://platform.openai.com/account/usage>

## ✓ 2. 답변 얻기

본격적으로 ChatGPT API를 사용해봅시다. 사용하는 코드의 형식은 다음 과 같습니다. model의 값으로 작성한 gpt-4o는 OpenAI의 API를 사용할 때 실행되는 ChatGPT의 다른 이름입니다. ChatGPT를 사용하기 위 해서는 모델의 값으로 gpt-4o 값을 사용해야 합니다. ChatGPT에게 작성할 프롬프트는 content의 값으로 입력하며, 이때 큰따옴표(")로 앞뒤를 감싸야 합니다.

```
completion = client.chat.completions.create(
    model="<사용하고자 하는 모델명>",
    messages=[
        {"role": "system", "content": "<정해 주고 싶은 역할>"},
        {"role": "user", "content": "<작성하고 싶은 Prompt>"}
    ]
)
```

API를 활용해 ChatGPT에게 “2020년 월드시리즈에서 누가 우승했어?”라 고 물어보겠습니다.

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "user", "content": "2020년 월드시리즈에서는 누가 우승했어?"}
    ]
)
```

# 답변을 받는 방법

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

🔄 2020년 월드시리즈에서는 로스앤젤레스 다저스가 우승했습니다. 다저스는 탬파베이 레이스를 상대로 시리즈에서 4승 2패로 승리하며 챔피언이 되었습니다. 이는 다저스의 32년 만의 우승이었습니다.

## ✓ 3. 역할 부여해 주기

ChatGPT API를 통해 ChatGPT를 사용할 때는 ChatGPT에게 역할 을 지시할 수 있습니다. 여기서 역할 지시란 ChatGPT가 앞으로 답변할 때 ChatGPT가 해당 역할로서 답변을 하라는 의미입니다. 앞으로 안내하는 역 할 지시 방법은 항상 지시한 역할대로 동작한다는 보장은 없지만 역 할 지시문 에 따라 답변 자체의 방향성을 바꿔버리기도 합니다.

역할을 지시하려면 기존 코드에서 messages=[] 안에 {"role": "system", "content": "<ChatGPT가 수행할 역할>"}를 추가로 작성합니다. 예를 들 어, ChatGPT에게 친절하게 답변해주는 비서라는 역할을 부여해봅시다. 코드 는 다음과 같습니다.

이와 같이 거대 언어 모델의 전체적인 방향성을 지시하는 입력을 시스템 프롬프트 라고 합니다.

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": "너는 친절하게 답변해주는 비서야. 답변은 반말로 해줘"},
        {"role": "user", "content": "2020년 월드시리즈에서는 누가 우승했어?"}
    ]
)
```

# 답변을 받는 방법

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

🔄 2020년 월드시리즈에서는 로스앤젤레스 다저스가 우승했어. 탬파베이 레이스를 상대로 4승 2패로 이겼어.

동일하게 로스앤젤레스 다저스가 우승했다는 답변을 하지만 답변이 달라집니다.

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": "You must only answer users' questions in English. This must be honored. You must only answer in English."},
        {"role": "user", "content": "2020년 월드시리즈에서는 누가 우승했어?"}
    ]
)
```

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

🔄 The Los Angeles Dodgers won the 2020 World Series.

이번에는 입력을 번역하라고 해봅시다.

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": "사용자의 질문에 답변을 하지말고 한글 입력을 영어로 번역하십시오."},
        {"role": "user", "content": "2020년 월드시리즈에서는 누가 우승했어?"}
    ]
)
```

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

🔄 Who won the World Series in 2020?

ChatGPT가 사용자의 질문에 대답하는 것이 아니라 질문을 영어로 번역하여 답변합니다.

이러한 시스템 프롬프트는 웹 사이트에서 사용하는 GPT-4에도 이미 적용되어져 있습니다.  
다만 UI 상으로 보여지지 않을 뿐입니다.

GPT-4 시스템 프롬프트 유출: <https://news.hada.io/topic?id=13326>

## ✓ 4. 이전 대화를 포함하여 답변하기

ChatGPT는 답변할 때 이전 질문과 답변을 모두 고려하여 답변하는 특징이 있습니다. ChatGPT API를 이용하면 ChatGPT에게 답변을 요청할 때 ChatGPT에게 '앞서 네가 이런 답변을 한 상태였다'라는 정보를 주는 것이 가능합니다. 이것은 사용자가 가정하는 것이지만 ChatGPT는 마치 과거에 ChatGPT 자신이 답변한 것으로 가정하고, 추가 답변을 합니다.

이렇게 질문을 작성하려면 `messages=[]` 안에

`{"role": "user", "content": "<사용자의 이전 프롬프트>"}`를 작성한 후

`{"role": "assistant", "content": "<ChatGPT의 이전 답변>"}`을 추가로 작성하고,

다시 `{"role": "user", "content": "<ChatGPT 이전 답변 이후의 사용자의 프롬프트>"}`를 번갈아 작성하면 됩니다.

실제 코드를 통해 확인해봅시다.

다음 코드에서는 사용자가 ChatGPT에게 "2002년 월드컵에서 가장 화제가 되었던 나라는 어디야?"라고 질문한 후,

ChatGPT가 "바로 예상을 뚫고 4강 진출 신화를 일으킨 한국입니다."라고 답변했다고 가정하고, 사용자가 다시 "그 나라가 화제가 되었던 이유를 자세하게 설명해줘"라고 질문한 상태를 가정하고 ChatGPT의 답변을 얻습니다.

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "user", "content": "2002년 월드컵에서 가장 화제가 되었던 나라는 어디야?"},
        {"role": "assistant", "content": "바로 예상을 뚫고 4강 진출 신화를 일으킨 한국입니다."},
        {"role": "user", "content": "그 나라가 화제가 되었던 이유를 자세하게 설명해줘"}
    ]
)
```

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

🔄 2002년 월드컵에서 한국이 화제가 되었던 이유는 여러 가지가 있습니다.

1. **\*\*역대 최고 성적\*\***: 한국은 2002년 한일 월드컵에서 4강에 진출하며 아시아 국가로서는 사상 처음으로 월드컵 준결승에 오르는 성과를 이뤘습니다. 이는 당시 개최국 이점을 가지고 있었지만, 뛰어난 경기력과 팀워크가 기반
2. **\*\*강호 팀들 격파\*\***: 한국은 본선 경기에서 세계적인 강호들을 물리치며 주목을 받았습니다. 특히 16강전에서는 이탈리아를, 8강전에서는 스페인을 승부차기 끝에 꺾고 4강에 오르는 기염을 토했습니다.
3. **\*\*히딩크 감독의 리더십\*\***: 당시 대한민국 축구 국가대표팀 감독이었던 거스 히딩크는 혁신적인 훈련 방식과 전략으로 팀을 조련하여 성과를 내며 '국민 영웅'으로 칭송받았습니다.
4. **\*\*열정적인 응원과 붉은악마\*\***: 월드컵 개최국으로서 한국 국민들의 뜨거운 응원과 '붉은악마'로 대표되는 팬들의 열정적인 서포터도 큰 화제가 되었습니다. 길거리 응원과 같은 모습들은 세계적으로도 주목받았습니다.

이러한 요소들이 조합되어 한국은 2002년 월드컵에서 가장 화제가 된 나라 중 하나가 되었습니다.

## ✓ 5. 너무나 쉽게 AI 서비스 개발해보기

## ✓ 5-1. 뉴스 기사를 분석해주는 인공지능

앞으로 AI 서비스를 개발할 때는 정해주고 싶은 역할에 ChatGPT가 수행할 작업을 구체적으로 명시해주면 됩니다.

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": "<정해 주고 싶은 역할>"},
        {"role": "user", "content": "<작성하고 싶은 Prompt>"}
    ]
)
```

```
def return_answer(input_text=''):
    system_prompt = """특정 문서가 입력되면 다음과 같은 형태로 문서를 분석하십시오.
```

1. 주어진 입력:에 대해서 반드시 주제:, 요약:, 가능한 질문: 이 세가지를 순차적으로 작성해야 합니다.
2. 주제:는 입력 문서의 주제를 한 줄로 요약합니다.
3. 요약:은 입력 문서를 5줄로 요약합니다.
4. 가능한 질문:은 입력 문서로부터 사람들이 할 수 있는 질문 세 가지를 파이썬 리스트 형태로 작성합니다.
5. 가능한 질문:이 반드시 ["질문1", "질문2", "질문3"]과 같이 파이썬 리스트 형태로 작성되어야 하는 점에 유의하십시오.

이제 시작합니다."""

```
user_content = "입력: " + input_text + '\n답변:'
```

```
print('최종 유저 프롬프트')
print('=' * 50)
print(user_content)
print('=' * 50)
```

```
response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": system_prompt},
        {"role": "user", "content": user_content}
    ]
)
return response.choices[0].message.content
```

test = '''새처럼 움직이는 항공 로봇 '레이븐'.

스위스 로잔 연방 공대(EPFL) 신원동 박사팀이 개발했는데요.

연구팀은 새의 엉덩이와 발목, 발을 참고해 뒷다리를 만들었습니다.

실제 새처럼 땅에서 걷고, 틈새를 건너뛰고, 위로 뛰어오릅니다.

경충 뛰면서 날아오르는 이 기능이 핵심인데요.


기존 고정익 항공기처럼 지상을 달리지 않고 제자리에서 날아오릅니다.

연구팀은 레이븐의 점프 이륙이 점프하지 않고 날아오를 때보다 에너지 효율이 더 높다고 설명했는데요.

또 기존 고정익 항공기에 이 기능을 도입하면 복잡한 지형에서도 효율적으로 쓸 수 있을 거라고 덧붙였습니다.

#새 #로봇 #항공 #공대 #스위스 #신원동 #박사 #점프'''

```
result = return_answer(input_text=test)
print(result)
```

 최종 유저 프롬프트

입력: 새처럼 움직이는 항공 로봇 '레이븐'.

스위스 로잔 연방 공대(EPFL) 신원동 박사팀이 개발했는데요.

연구팀은 새의 엉덩이와 발목, 발을 참고해 뒷다리를 만들었습니다.

실제 새처럼 땅에서 걷고, 틈새를 건너뛰고, 위로 뛰어오릅니다.

경충 뛰면서 날아오르는 이 기능이 핵심인데요.

기존 고정익 항공기처럼 지상을 달리지 않고 제자리에서 날아오릅니다.

연구팀은 레이븐의 점프 이륙이 점프하지 않고 날아오를 때보다 에너지 효율이 더 높다고 설명했는데요.

또 기존 고정익 항공기에 이 기능을 도입하면 복잡한 지형에서도 효율적으로 쓸 수 있을 거라고 덧붙였습니다.

#새 #로봇 #항공 #공대 #스위스 #신원동 #박사 #점프

답변:

주제: 스위스 로잔 연방 공대에서 개발한 새처럼 움직이는 항공 로봇 '레이븐'

요약: 스위스 로잔 연방 공대의 신원동 박사팀이 개발한 항공 로봇 '레이븐'은 새의 뒷다리를 참고하여 설계되었습니다. 이 로봇은 실제 새처럼 땅에서 걷고, 틈새를 건너뛰고, 뛰어오르는 등 다양한 움직임이 가능합니다. 레이:

가능한 질문: ["레이븐 로봇의 주요 기능은 무엇인가요?", "레이븐의 점프 이륙 방식이 기존 항공기에 비해 어떤 장점이 있나요?", "이 연구의 실용적 응용 가능성은 어떻게 평가되나요?"]

```
test = ''
```

꿈의 에너지' 라고 불리는 핵융합 에너지가 5년 뒤 현실이 될지도 모른다. 마이크로소프트가 5년 뒤부터 핵융합으로 만든 전기를 사서 쓰겠다는 전기공급 계약을 마이크로소프트에게 전기를 공급할 곳은 미국 스타트업 '헬리온 에너지' 다. 5월 10일 핵융합 발전 스타트업 헬리온 에너지가 마이크로소프트와 계약을 맺었다고 하지만 전문가들의 전망은 긍정적이지만은 않다. 사용할 수 있는 정도의 전기를 만들어 내는 실증 시험까지 거친 핵융합 연구가 아직 없기 때문이다. 대신 현실적 핵융합 발전 연구는 크게 공공 개발과 민간 개발, 둘로 나뉘는데 둘 중 핵융합 발전 목표 시기가 조금 더 빠른 민간 개발의 목표 시기가 2030년경이다. 다만 지난

```
result = return_answer(input_text=test)
print(result)
```

🔄 최종 유저 프롬프트

```
=====
입력:
꿈의 에너지' 라고 불리는 핵융합 에너지가 5년 뒤 현실이 될지도 모른다. 마이크로소프트가 5년 뒤부터 핵융합으로 만든 전기를 사서 쓰겠다는 전기공급 계약을 맺었다.

마이크로소프트에게 전기를 공급할 곳은 미국 스타트업 '헬리온 에너지' 다. 5월 10일 핵융합 발전 스타트업 헬리온 에너지가 마이크로소프트와 계약을 맺었다고 밝혔다. 계약 조건은 2028년부터 헬리온 에너지가 핵융합 발전을 하지만 전문가들의 전망은 긍정적이지만은 않다. 사용할 수 있는 정도의 전기를 만들어 내는 실증 시험까지 거친 핵융합 연구가 아직 없기 때문이다. 대신 현실적인 목표 시기는 2040년 이후가 될 것으로 보인다.

핵융합 발전 연구는 크게 공공 개발과 민간 개발, 둘로 나뉘는데 둘 중 핵융합 발전 목표 시기가 조금 더 빠른 민간 개발의 목표 시기가 2030년경이다. 다만 지난해 11월 유럽 위원회가 낸 보고서에서는 이 시기를 시험 운영 목

답변:
=====
주제: 마이크로소프트가 핵융합 에너지 전기를 공급받기 위한 계약과 핵융합 발전의 현실적인 목표 시기

요약: 마이크로소프트는 헬리온 에너지와 2028년부터 핵융합으로 만든 전기를 공급받는 계약을 맺었다. 이 계약에 따르면 헬리온 에너지는 2029년까지 50메가와트 이상의 전기를 공급해야 하며, 그렇지 못할 경우 위약금을 내야

가능한 질문: ["마이크로소프트가 헬리온 에너지와 계약한 조건은 무엇인가요?", "핵융합 에너지의 현실적인 상용화 시기는 언제로 예측되고 있나요?", "유럽 위원회는 핵융합 발전의 시험 운영 목표 시기를 어떻게 전망했나요?"
```

## ✓ 5-2. 키워드 추출

```
def return_answer(input_text):
    system_prompt = """You are an expert at extracting keywords from a given sentence. Extract keywords from a given sentence that are key to the conte.

[Example]
input: '외신에 따르면 중국 이카이글로벌 보도를 인용해 모더나가 중국 전용 mRNA 백신 개발을 위해 중국에 최대 10억달러(약 1조3017억원)를 투자하기로 했
keyword: ['모더나', 'mRNA', '1조3017억원']

Now the sentence you want to extract keywords from.
The output format must be a list in Python.
"""

    user_content = "input: " + input_text + "nkeyword: "
```

```

print('최종 유저 프롬프트')
print('=' * 50)
print(user_content)
print('=' * 50)

response = client.chat.completions.create(
    model="gpt-4o",
    messages=[
        {"role": "system", "content": system_prompt},
        {"role": "user", "content": user_content}
    ],
    temperature=0
)
return response.choices[0].message.content

```

text = '''3일 업계에 따르면 서울 신라호텔의 더 파크뷰는 12월 1~20일 저녁 가격을 19만5000원으로, 21~31일 저녁 가격을 21만 5000원으로 각각 인상한다. 평  
롯데호텔 서울의 뷔페 라세느는 12월 평일·주말 저녁 가격을 19만원으로 기존 대비 1만원 올린다. 크리스마스 연휴 때인 23~25일과 연말 30~31일 저녁 가격은 ;  
위커힐 호텔앤리조트는 그랜드 위커힐에서 운영하던 더뷔페를 비스타 위커힐로 확장 이전해 이날 가격을 올렸다. 평일·주말 저녁 가격은 15만9000원에서 18만9000  
그랜드 인터컨티넨탈 서울 파르나스와 인터컨티넨탈 서울 코엑스도 다음주께 인상된 가격을 공지할 계획이다.'''

```

return_answer(input_text=text)

```



최종 유저 프롬프트

```

=====
input: 3일 업계에 따르면 서울 신라호텔의 더 파크뷰는 12월 1~20일 저녁 가격을 19만5000원으로, 21~31일 저녁 가격을 21만 5000원으로 각각 인상한다. 평소 평일·주말 저녁 가격이 18만5000원이던 것을 고려하면 각각 1만
롯데호텔 서울의 뷔페 라세느는 12월 평일·주말 저녁 가격을 19만원으로 기존 대비 1만원 올린다. 크리스마스 연휴 때인 23~25일과 연말 30~31일 저녁 가격은 20만5000원이다. 롯데호텔 서울의 라세느도 현재 크리스마스 이브
위커힐 호텔앤리조트는 그랜드 위커힐에서 운영하던 더뷔페를 비스타 위커힐로 확장 이전해 이날 가격을 올렸다. 평일·주말 저녁 가격은 15만9000원에서 18만9000원으로 3만원 인상됐다. 다만 위커힐은 이번 가격 책정에 인플레
그랜드 인터컨티넨탈 서울 파르나스와 인터컨티넨탈 서울 코엑스도 다음주께 인상된 가격을 공지할 계획이다.
keyword:
=====
['서울 신라호텔', '더 파크뷰', '19만5000원', '21만5000원', '롯데호텔 서울', '라세느', '19만원', '20만5000원', '위커힐 호텔앤리조트', '더뷔페', '비스타 위커힐', '18만9000원', '인플레이션', '그랜드 인터컨티넨탈
서울 파르나스', '인터컨티넨탈 서울 코엑스']

```

### 5-3. 제목 추출



```
def return_answer(text):
    prompt = '''주어진 텍스트로부터 적절한 제목을 만들어줘. 제목은 간결하고 너무 길어서는 안 돼.'''

    response = client.chat.completions.create(
        model="gpt-4o",
        messages=[
            {"role": "system", "content": prompt},
            {"role": "user", "content": text}
        ]
    )
    return response.choices[0].message.content
```

input\_text = '''지난 11일 국회 정무위원회 전체 회의에선 이복현 금융감독원장의 해외 출장길이 도마 에 올랐다. 이 원장이 지난 8일부터 닷새간 동남아 싱가포르 이 원장의 이번 출장은 국내 금융사의 투자 유치와 해외진출 확대를 지원하기 위해서다. 이 출장에는 윤종규 KB금융그룹 회장, 함영주 하나금융그룹 회장을 비롯해 야당의 지적처럼 금융권 일각에선 이번 출장을 두고 다소 의아하다는 시각이 있었다. 금감원이 해외에서 IR 행사를 하긴 하지만, 금융위원장이 아니라 금감원장이

```
return_answer(input_text)
```

🔄 '금감원장의 해외 출장 논란'

#### ✓ 5-4. 감성 분류

```
def return_answer(text):
    prompt = '''주어진 텍스트가 긍정인지 중립인지 부정인지 예측하시오. 당신의 답변은 오로지 [positive, negative, neutral] 셋 중 하나여야만 합니다.
    ex) 'SK하이닉스가 2분기 실적에서 역대 최고를 기록하였다' => "positive"'''

    response = client.chat.completions.create(
        model="gpt-4o",
        messages=[
            {"role": "system", "content": prompt},
            {"role": "user", "content": text + " => "}
        ]
    )
    return response.choices[0].message.content
```

return\_answer('씨씨에스 로고 씨씨에스 주가가 상한가다. 3일 기준 씨씨에스는 29.84%(825원) 상승한 3590원에 거래를 마쳤다. 상온 초전도체 이슈가 부각되면서

🔄 '"positive"'

