

# ChatGPT API 시작하기

인공지능 전공  
안정호 교수

# 02

## ChatGPT API

### 시작하기

2.1 API 사용 요금	32
2.2 API 키 발급하기	34
2.3 실습 환경 조성하기	36
2.3.1 파이썬 설치하기	37
2.3.2 파이썬 가상 환경 만들기	41
2.3.3 비주얼 스튜디오 코드 설치하기	43
2.3.4 구글 코랩 사용하기	45
2.4 ChatGPT API 기본 사용 방법	47
2.4.1 기본 질문하기	48
2.4.2 역할 부여하기	53
2.4.3 이전 대화를 포함하여 답변하기	56
2.5 OpenAI에서 제공하는 AI 서비스	57

# ChatGPT API

**API = Application Programming Interface**

S/W 애플리케이션을 구축하고 상호 작용하기 위한 규칙과 프로토콜의 집합  
명령어, 함수 포함

OpenAI에서 **ChatGPT API** 제공. 이 API를 이용해,

- 새로운 애플리케이션 개발 가능
- 기존 서비스에 ChatGPT 기능 추가 가능

ChatGPT API 개념도:

## 2.1 API 사용 요금

ChatGPT API 이용은

- 유료, 카드 등록 필요
- 최초 가입시 \$5 크레딧 제공
- ChatGPT Plus와는 완전히 별개의 서비스
- (월 정액 구독 서비스가 아닌) 사용한 양만큼 요금 지불
- 모델별로 요금 정책 상이
- 토큰 개수(Prompt Token과 Completion Token) 개수에 따라 요금 부과

최신 OpenAI API 요금정책: <https://openai.com/pricing#language-models>

## 2.1 API 사용 요금(모델 별)

- GPT-4: GPT-4는 가장 우수한 성능을 가진 모델입니다. OpenAI의 설명에 따르면 GPT-4는 GPT-3.5보다 훨씬 더 안정적이고, 창의적이며, 미묘한 지침을 처리할 수 있다고 소개하고 있습니다. GPT-4 모델은 크게 3가지 유형이 있습니다. 유료 API 사용 시 요금 정책은 다음과 같습니다.
  - 첫 번째 GPT-4 모델(gpt-4): 최대 8,192개의 토큰을 처리할 수 있습니다. 이는 GPT-3.5의 최대 길이인 4,097개와 비교하면 두 배의 처리 능력입니다. 프롬프트 토큰(prompt tokens) 1,000개에 대해 \$0.03를 부과하며, 완료 토큰(completion tokens) 1,000개에 대해 \$0.06를 부과합니다.
  - 두 번째 GPT-4 모델(gpt-4-32k): 최대 32,768개의 토큰을 처리할 수 있습니다. 프롬프트 토큰(prompt tokens) 1,000개에 대해 \$0.06를 부과하며, 완료 토큰(completion tokens) 1,000개에 대해 \$0.12를 부과합니다.
  - 세 번째 GPT-4 모델(gpt-4-turbo): 유일하게 2023년 4월까지의 학습 데이터를 제공하는 모델로, 최대 128,000개의 토큰을 처리할 수 있습니다. 뿐만 아니라 GPT-4 대비 낮은 비용을 부과하여 가성비 좋은 모델입니다. 프롬프트 토큰(prompt tokens) 1,000개에 대해 \$0.01를 부과하며, 완료 토큰(completion tokens) 1,000개에 대해 \$0.03를 부과합니다.

- ChatGPT(GPT-3.5): 가장 많이 알려진 모델로, 웹에서 무료로 사용 시 활용되는 모델입니다. ChatGPT의 성능은 davinci와 같지만, 가격은 훨씬 저렴하다고 소개하고 있습니다. 2021년 9월까지의 학습 데이터를 사용했지만, 추가 학습 기능을 제공하여 사용자가 원한다면 최신 데이터로 추가 학습이 가능합니다. ChatGPT 모델은 총 두 가지 유형이 있습니다. 유료 API 사용 시 요금 정책은 다음과 같습니다.
  - 첫 번째 ChatGPT 모델(gpt-3.5-turbo-instruct): 최대 4,097개의 토큰을 처리할 수 있습니다. 프롬프트 토큰(prompt tokens) 1,000개에 대해 \$0.0015를 부과하며, 완료 토큰(completion tokens) 1,000개에 대해 \$0.002를 부과합니다.
  - 두 번째 ChatGPT 모델(gpt-3.5-turbo-1106): 최대 16,384개의 토큰을 처리할 수 있습니다. 이는 gpt-3.5-turbo-instruct의 최대 길이인 4,097개와 비교하면 두 배의 처리 능력입니다. 프롬프트 토큰(prompt tokens) 1,000개에 대해 \$0.001를 부과하며, 완료 토큰(completion tokens) 1,000개에 대해 \$0.002를 부과합니다.
- davinci-002: 가장 뛰어나면서도 유료 전환 시 가장 비용이 많이 드는 제품입니다. 유료 전환 시 토큰 1,000개에 대해 \$0.0020를 부과합니다. 뒤에서 설명할 파라미터 값을 조절하면서 좀 더 섬세하고, 전문적인 사용이 필요하다면 davinci를 추천합니다. 하지만 GPT 기본 모델로써 명령어에 최적화되지 않은 한계가 있습니다.
- babbage-002: 짧은 텍스트를 수정하거나 분석하는 수준의 간단한 작업에 적합한 모델입니다. 좋은 작문이 필요한 작업에는 적합하지 않을 수 있습니다. 유료 전환 시 토큰 1,000개에 대해 \$0.0004를 부과합니다. 이 모델 또한 GPT 기본 모델로써 명령어에 최적화되지 않은 한계가 있습니다.

# 토큰 개수 계산

Multiple models, each with different capabilities and price points. Prices are per 1,000 tokens. You can think of tokens as pieces of words, where 1,000 tokens is about 750 words. This paragraph is 35 tokens.

OpenAI 토큰 계산 사이트:

<https://platform.openai.com/tokenizer>

## Learn about language model tokenization

OpenAI's large language models (sometimes referred to as GPT's) process text using **tokens**, which are common sequences of characters found in a set of text. The models learn to understand the statistical relationships between these tokens, and excel at producing the next token in a sequence of tokens.

You can use the tool below to understand how a piece of text might be tokenized by a language model, and the total count of tokens in that piece of text.

It's important to note that the exact tokenization process varies between models. Newer models like GPT-3.5 and GPT-4 use a different tokenizer than previous models, and will produce different tokens for the same input text.

GPT-3.5 & GPT-4 GPT-3 (Legacy)

인공지능 서비스 개발

Clear

Show example

Tokens

10

Characters

11

인공지능 서비스 개발

## 2.2 API 키 발급하기

OpenAI의 API를 사용하기 위해, OpenAI 사이트에서 **API 키(Key)**를 발급받아야 함.

API 키는 외부 프로그램과 OpenAI의 서버가 통신할 때, 사용자가 누군지 식별하는 용도로 사용.

주민등록번호와 같은 개념.

내 API 키 노출시 나도 모르게 과금될 수 있으므로 보안에 주의

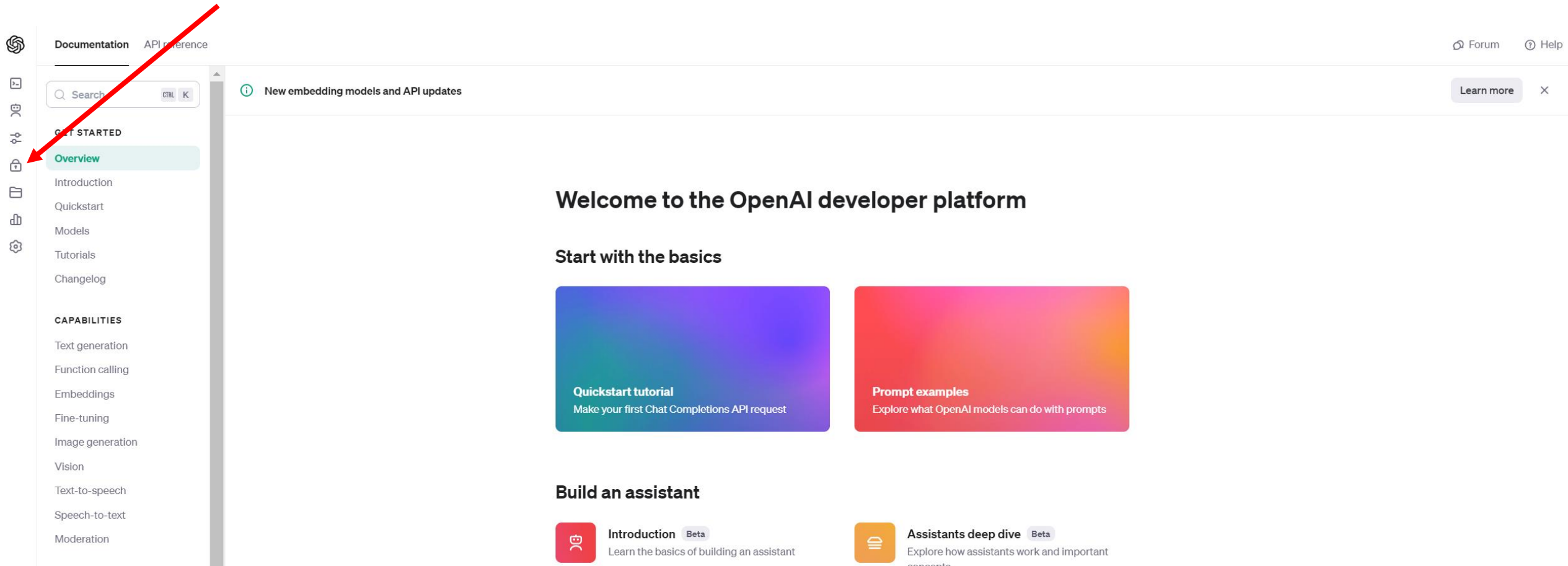
삭제, 재발급 쉬움, 권장함.

OpenAI API 홈페이지: <https://platform.openai.com/>

**필수!** 각자 openai.com에 접속하여 계정을 만들고 ChatGPT API 키를 발급 받으세요.



## 2.2 API 키 발급하기



The screenshot shows the OpenAI developer platform documentation page. On the left, a sidebar contains a search bar and a list of links. A red arrow points to the 'Overview' link under the 'GET STARTED' section. The main content area has a header 'Welcome to the OpenAI developer platform' and a section 'Start with the basics' with two cards: 'Quickstart tutorial' and 'Prompt examples'. Below this is a section 'Build an assistant' with two cards: 'Introduction' and 'Assistants deep dive'.

Documentation API reference

Search  CTRL K

GET STARTED

- Overview
- Introduction
- Quickstart
- Models
- Tutorials
- Changelog

CAPABILITIES

- Text generation
- Function calling
- Embeddings
- Fine-tuning
- Image generation
- Vision
- Text-to-speech
- Speech-to-text
- Moderation

New embedding models and API updates

Forum Help

Learn more

### Welcome to the OpenAI developer platform

#### Start with the basics

##### Quickstart tutorial

Make your first Chat Completions API request

##### Prompt examples

Explore what OpenAI models can do with prompts

#### Build an assistant



##### Introduction Beta


Learn the basics of building an assistant


##### Assistants deep dive Beta


Explore how assistants work and important concepts


다음 실습을 위해 각자 신속히 발급받으실 것을 권장합니다.


- 
-  Playground


 Assistants

 Fine-tuning

 API keys

 Storage

 Usage


 Settings

## API keys

Your secret API keys are listed below. Please note that we do not display your secret API keys again after you generate them.

Do not share your API key with others, or expose it in the browser or other client-side code. In order to protect the security of your account, OpenAI may also automatically disable any API key that we've found has leaked publicly.

Enable tracking to see usage per API key on the [Usage page](#).

NAME	SECRET KEY	TRACKING ⓘ	CREATED	LAST USED ⓘ	PERMISSIONS	
ChatGPT_API_KEY_WSTUDY	sk- . . . GbKj	Enabled	2024년 1월 10일	2024년 3월 1일	All	

+ Create new secret key

## Default organization

If you belong to multiple organizations, this setting controls which organization is used by default when making requests with the AP keys above.

Personal

Note: You can also specify which organization to use for each API request. See [Authentication](#) to learn more.

## 2.3 실습환경 조성하기

2.3.1 파이썬 설치하기

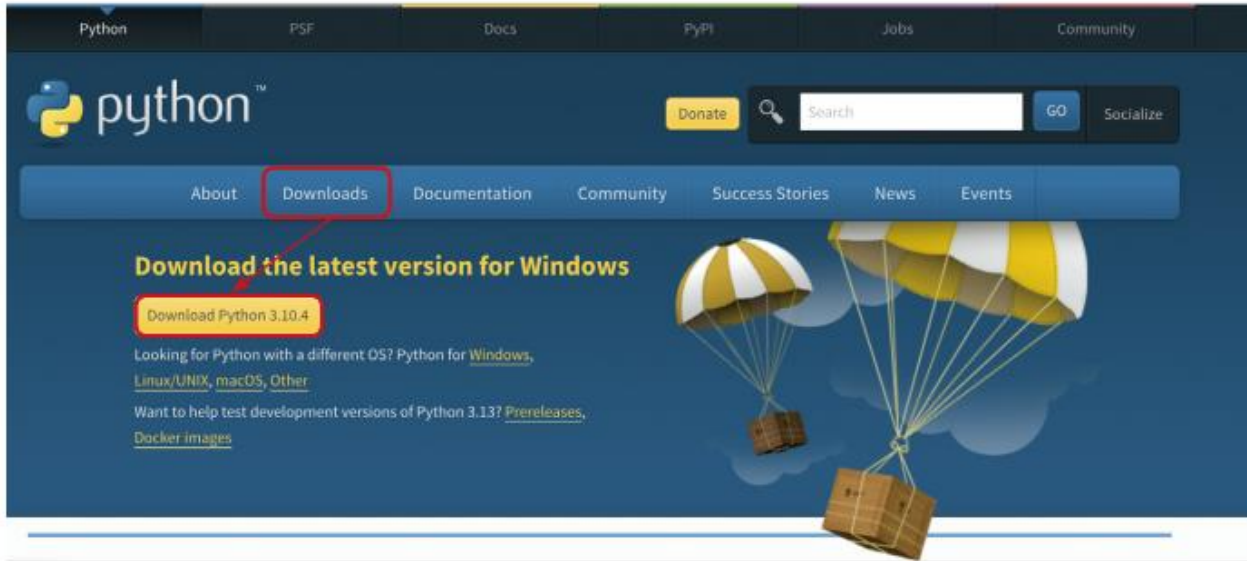
2.3.2 파이썬 가상환경 만들기

2.3.3 비주얼 스튜디오 코드 설치하기

2.3.4 구글 코랩 사용하기

## 2.3.1 파이썬 설치하기

파이썬 다운로드 페이지: <https://www.python.org/downloads/>



반드시 체크

아나콘다 아님

## 2.3.2 파이썬 가상환경 만들기

### 가상환경(virtual environment)

- 파이썬을 활용하여 프로그램을 개발할 때 반드시 알아야 할 개념 중 하나
- 하나의 PC에 여러 가상환경 만들 수 있다.
- 가상환경마다 다른 버전의 파이썬과 라이브러리 설치 가능

## 2.3.2 파이썬 가상환경 만들기

### 1. 가상환경 생성

```
# python -m venv '가상_환경_이름'  
C:\> python -m venv py_env
```

py\_env 폴더 생성

### 2. 가상환경 진입하기

```
# 윈도우: 가상_환경_이름\Scripts\activate.bat  
# macOS: source 가상_환경_이름/bin/activate  
C:\> py_env\Scripts\activate.bat
```

```
(py_env) C:\>
```

### 3. 가상환경 벗어나기

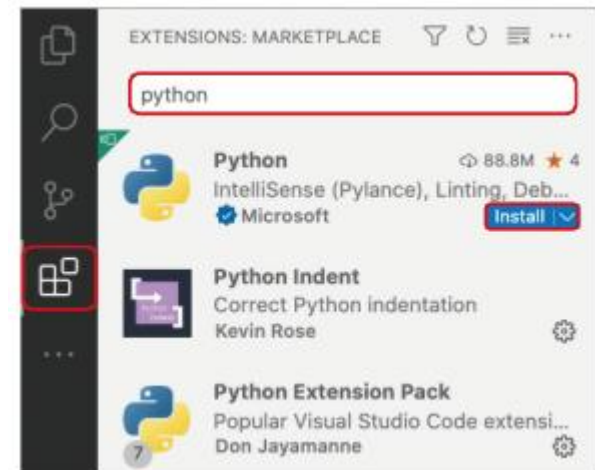
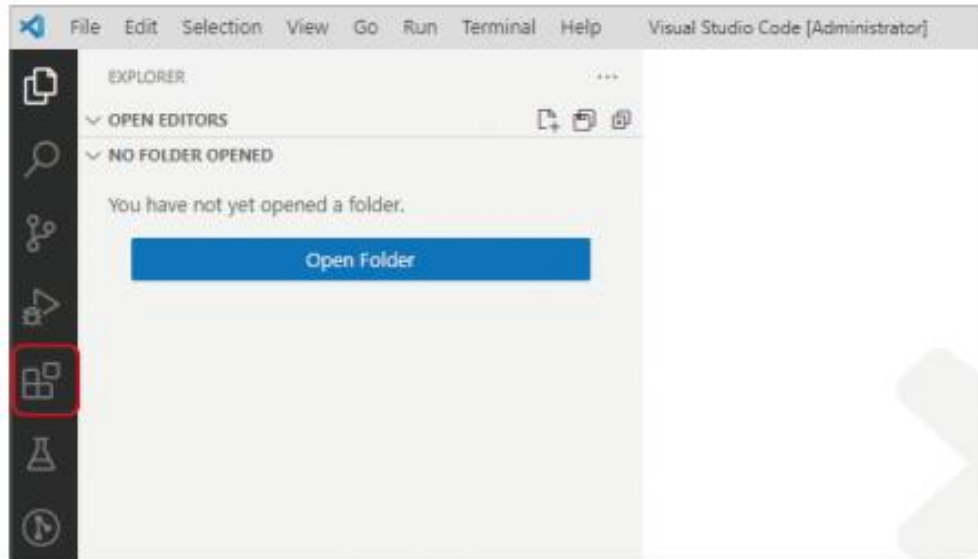
```
(py_env) C:\> deactivate  
C:\>
```

## 2.3.3 비주얼 스튜디오 코드 설치하기

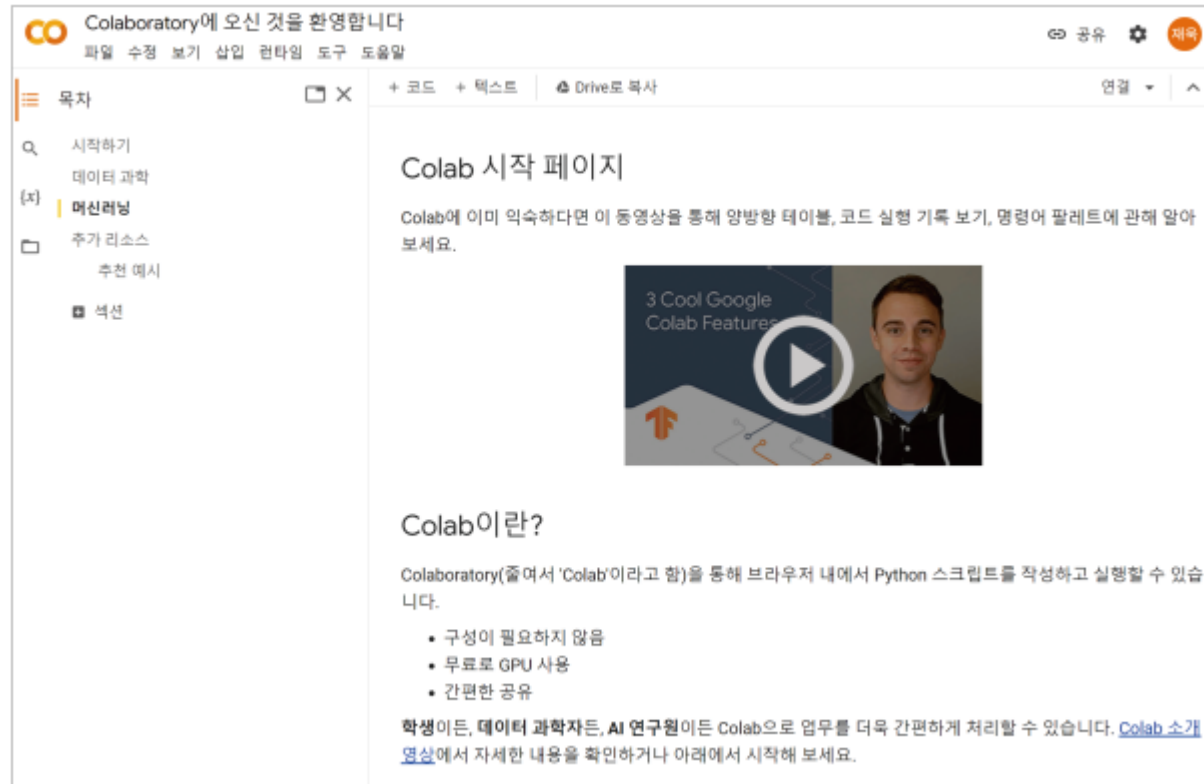
■ 비주얼 스튜디오 코드 공식 홈페이지: <https://code.visualstudio.com>

### 비주얼 스튜디오 코드(Visual Studio Code, VS Code)

- 프로그래머들에게 가장 많은 사랑을 받는 에디터
- 파이썬 확장 프로그램(Extension) 설치



## 2.3.4 구글 코랩 사용하기





## 2.4 ChatGPT API 기본 사용 방법

2.4.1 기본 질문하기

2.4.2 역할 부여하기

2.4.3 이전 대화를 포함하여 답변하기

## 2.4 ChatGPT API 기본 사용 방법

Colab 실습: ChatGPT\_API\_실습.ipynb 파일 생성

Colab에서 openai 모듈 설치

```
!pip install openai
```

```
import openai
```

여기에 2.2절 'API 키 발급하기(34쪽)'에서 발급받은  
OpenAI API 키 값을 입력합니다.

```
client = openai.OpenAI(api_key = "api key 입력")
```

## 2.4.1 기본 질문 하기

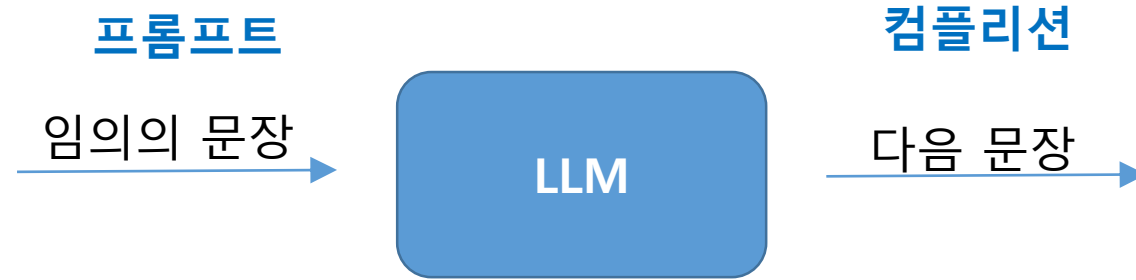
```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages=[{"role": "user", "content": "프롬프트 입력"}],  
    temperature=1,  
    top_p=1,  
    presence_penalty=1,  
    frequency_penalty=1,  
    n=1,  
    max_tokens=4000,  
    stop=None,  
    seed=특정 숫자,  
    response_format={ "type": "json_object" },  
    tools=사용자 지정 툴,  
    tools_choice="auto"  
)
```

질문하기 위한 메서드

파라미터들

- **메시지(message)**: 사용자가 입력할 프롬프트가 포함된 리스트입니다. 가장 핵심이 되는 부분으로 메시지를 활용하여 원하는 질문을 하고, 다양한 형태의 메시지 입력을 통해 프롬프트 엔지니어링을 할 수 있습니다. 기본적인 질문 방법은 [{"role": "user", "content": "프롬프트 입력"}]과 같이 content 키 안에 질문하고자 하는 프롬프트를 입력하는 방법입니다. 역할을 부여하거나 이어서 질문하는 방법은 다음 절에서 자세히 다루겠습니다.
- **온도 조절(temperature)**: 텍스트의 랜덤성(randomness)과 관련된 파라미터입니다. 온도를 높게 설정하면(최대값은 2.0) 모델이 생성하는 텍스트가 사람이 보기에는 예측에서 벗어나고 다양하며 창의적으로 보이는 경향을 띵니다. 반면, 온도를 낮게 설정하면 좀 더 전형적이고 보수적인 텍스트를 생성하는 경향을 보입니다. 일반적으로 온도 값은 0.5~1.5 사이의 값을 시작점으로 사용합니다. 적절한 온도 값은 사용자가 어떤 작업을 하고 싶은지, 어떤 프롬프트를 사용하는지에 따라 다르므로 실험적으로 찾는 것이 좋습니다. 설정값의 범위는 0.0~2.0이며, 값을 설정하지 않으면 기본값인 1로 설정됩니다.
- **핵 샘플링(top\_p)**: 다음 단어 또는 토큰이 샘플링되는 범위를 제어합니다. 응답을 생성할 때 모델은 다음 토큰의 어휘에 대한 확률 분포를 계산합니다. 예를 들어, top\_p를 0.5로 설정하면 모델이 샘플링할 때, 누적 확률이 0.5보다 큰 상위 토큰 중에서만 다음 토큰을 샘플링합니다. top\_p를 1.0으로 설정하면 모든 토큰(전체 분포)에서 샘플링하고, top\_p를 0.0으로 설정하면 항상 가장 확률이 높은 단일 토큰을 선택합니다. 설정값의 범위는 0.0~1.0이며, 값을 설정하지 않으면 기본값인 1로 설정됩니다.
- **존재 페널티(presence\_penalty)**: 단어가 이미 생성된 텍스트에 나타난 경우 해당 단어가 등장할 가능성을 줄입니다. 빈도수 페널티와 달리 존재 페널티는 과거 예측에서 단어가 나타나는 빈도수에 따라 달라지는 않습니다. OpenAI의 설명에 따르면 이 파라미터 값을 크게 설정할수록 모델이 새로운 주제에 대해 이야기할 가능성이 높아진다고 합니다. 설정값의 범위는 0.0~2.0이며, 값을 설정하지 않으면 기본값인 0으로 설정됩니다.
- **빈도수 페널티(frequency\_penalty)**: 모델이 동일한 단어를 반복적으로 생성하지 않도록 설정하는 값입니다. 이 페널티는 어떤 단어가 어느 빈도로 등장했는지에 따라 영향을 받습니다. 빈도수 페널티의 값을 높게 설정하면 모델이 이미 생성한 단어를 다시 생성하는 것을 자제하므로 좀 더 다양하고 중복되지 않은 텍스트를 생성하게 유도할 수 있습니다. 따라서 모델이 특정 단어를 반복하는 경향을 보인다면 빈도수 페널티 값을 높게 설정합니다. 설정값의 범위는 0.0~2.0이며, 값을 설정하지 않으면 기본값인 0으로 설정됩니다.
- **응답 개수(n)**: 입력 메시지에 대해 생성할 답변의 수를 설정합니다. 값을 설정하지 않으면 기본값인 1로 설정됩니다.
- **최대 토큰(max\_tokens)**: 최대 토큰 수를 제한합니다. API 사용은 토큰 수에 따라 요금이 부과되므로 최대 토큰 파라미터를 통해 답변의 길이를 조절할 수 있습니다. 값을 설정하지 않으면 모델의 최대 토큰 수에 맞춰 설정됩니다.
- OpenAI API reference: <https://platform.openai.com/docs/api-reference>

# 언어 모델 이해



temperature 와  
top\_p

Large Language Model

- 문장 생성 입출력
- 질의응답 입출력
- 논리적 사고

나는

고양이다

## 2.4.1 기본 질문 하기

```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages=[{"role": "user", "content": "Tell me how to make a pizza"}])  
print(response)
```

여기에 답변 결과가 담깁니다.

답변 결과를 출력합니다.

각자 답변을 출력해 보세요

## 2.4.1 기본 질문 하기

```
ChatCompletion(  
  id='chatcmpl-8L07P3XCYU2YX6v5Qq8DhULu6lmrV',  
  choices=[  
    Choice(finish_reason='stop',  
           index=0,  
           message=ChatCompletionMessage(content='답변내용',  
                                           role='assistant',  
                                           function_call=None,  
                                           tool_calls=None))  
  ],  
  created=1700016035,  
  model='gpt-3.5-turbo-0613',  
  object='chat.completion',  
  system_fingerprint=None,  
  usage=CompletionUsage(completion_tokens=305,  
                         prompt_tokens=15,  
                         total_tokens=320))
```

답변 결과 텍스트

사용한 토큰 수

## 2.4.1 기본 질문 하기

ChatGPT 답변 (텍스트) 추출

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

As an AI language model, I can give you the steps to make a basic pizza.

Ingredients:

- 1 pound pizza dough
  - 1 cup tomato sauce
  - 2 cups shredded mozzarella cheese
  - 1/2 cup grated parmesan cheese
  - Optional toppings: sliced pepperoni, sliced onions, sliced mushrooms, sliced bell peppers, chopped garlic, red pepper flakes, fresh basil leaves
- ... 생략 ...

## 2.4.1 기본 질문 하기

소모 토큰 확인

```
print(response.usage)
```

결과

```
CompletionUsage(completion_tokens=305, prompt_tokens=15, total_tokens=320)
```

비용 계산

```
total_bill = response.usage.prompt_tokens * 0.0015/1000 +  
response.usage.completion_tokens * 0.002/1000  
print("총 발생 비용 {} 원".format(total_bill * 1323.43))
```

결과

```
총 발생 비용 0.837069 원
```



## 2.4.2 역할 부여하기

ChatGPT API를 이용할 때 ChatGPT에게 역할을 지시할 수 있다.

역할 지시란, ChatGPT가 앞으로 답변할 때 해당 역할로서 답변하라는 의미이다.

역할 지시문에 따라 답변의 방향성이 바뀐다.

역할을 지시하려면, 아래와 같이 messages의 [ ]안에 {"role": "system", "content": "~~~"}를 추가로 작성합니다.

아래는 ChatGPT에게 친절하게 답해주는 비서 역할을 부여합니다

```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages=[  
        {"role": "system", "content": "너는 친절하게 답변해 주는 비서야"},  
        {"role": "user", "content": "2020년 월드시리즈에서는 누가 우승했어?"}]  
    )  
print(response.choices[0].message.content)
```

역할 부여

질문

### 결과

2020년 월드시리즈에서는 미국 메이저리그의 로스앤젤레스 더저스가 탬파베이 레이스를 대상으로 4승 2패로 승리하여 우승하였습니다.

# 세가지 role

<https://platform.openai.com/docs/guides/chat/introduction>

## system

선택사항으로, assitant의 동작 설정;

어시스턴트의 성격을 수정하거나 대화 전반에 걸쳐 어시스턴트가 어떻게 행동해야 하는지에 대한 구체적인 지침을 제공

## assistant

## user

## 2.4.2 역할 부여하기

```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    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)
```

역할 부여

질문

### 결과

In 2020 World Series, the Los Angeles Dodgers won the championship.

## 2.4.2 역할 부여하기

```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages=[  
        {"role": "system", "content": "You're a chatbot that refuses to answer and  
says sorry when users ask questions."},  
        {"role": "user", "content": "2020년 월드시리즈에서는 누가 우승했어?"}  
    ]  
)  
print(response.choices[0].message.content)
```

결과

I'm sorry, but I can't answer your question.

## 2.4.2 역할 부여하기

```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    messages=[  
        {"role": "system", "content": "You are a translator that translates users'  
inputs. If the input is in Korean, it must be translated into English. This must  
be strictly adhered to."},  
        {"role": "user", "content": "2020년 월드시리즈에서는 누가 우승했어?"}  
    ]  
)  
print(response.choices[0].message.content)
```

결과

Who won the World Series in 2020?

## 2.4.2 이전 대화를 포함하여 답변하기

ChatGPT는 답변할 때 이전 질문과 답변을 모두 고려하여 답변하는 특징이 있습니다. ChatGPT API를 이용하면 ChatGPT에게 답변을 요청할 때 ‘앞서 네가 이런 답변을 한 상태였다’는 정보를 전달할 수 있습니다. 이것은 사용자가 가정하는 것이지만, ChatGPT는 마치 과거에 자신이 답변한 것으로 가정하고, 추가 답변을 제공합니다.

이렇게 질문을 작성하려면 `messages=[ ]` 안에 `{"role": "user", "content": ""}`를 작성한 후 `{"role": "assistant", "content": ""}`을 추가로 작성하고, 다시 `{"role": "user", "content": ""}`를 번갈아 작성하면 됩니다. 실제 코드를 통해 확인해 봅시다.

다음 코드에서는 사용자가 ChatGPT에게 “2002년 월드컵에서 가장 화제가 되었던 나라는 어디야?”라고 질문한 후, ChatGPT가 “바로 예상을 뚫고 4강 진출 신화를 일으킨 한국입니다.”라고 답변한 상태라고 가정하고, 사용자가 다시 “그 나라가 화제가 되었던 이유를 자세하게 설명해 줘”라고 질문해 보겠습니다.

## 2.4.2 이전 대화를 포함하여 답변하기

```
response = client.chat.completions.create(  
    model="gpt-3.5-turbo",  
    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 설명해 줘"}  
    ]  
)  
print(response.choices[0].message.content)
```

### 결과

한국이 2002년 월드컵에서 화제가 되었던 이유는 다음과 같습니다.

1. 예상치 못한 좋은 성적: 한국은 이전 대회에서는 한 번도 16강을 넘어갈 수 없었지만, 2002년 대회에서는 4강까지 진출하여 세계적인 축구 강국들을 이겨내었습니다. 이러한 좋은 성적은 세계적으로 큰 관심을 불러일으켰습니다.

... 생략...

## 2.5 OpenAI에서 제공하는 서비스

OpenAI에서는 ChatGPT뿐만 아니라 자연어로 이미지를 생성하는 DALL·E(달리), 이미지를 통해 자연어를 생성하는 Vision, 음성을 텍스트로 분석해주는 Whisper(위스퍼) 등 다양한 생성형 AI 서비스의 API를 제공합니다. 이어지는 3장에서는 Whisper를 활용하여 음성 비서를 만들 예정이고, 4장에서는 DALL·E를 활용하여 텔레그램/카카오톡 메신저에서 이미지를 생성하는 서비스 만들 예정입니다. OpenAI에서 제공하는 API 서비스에 대한 자세한 설명은 다음 사이트에서 자세히 확인할 수 있습니다.

- OpenAI 플랫폼: <https://platform.openai.com/>



# 03

## 나만의 음성 비서 만들기

3.1 음성 비서 프로그램 맛보기 60

3.2 음성 비서 프로그램의 구조 63

3.3 개발 환경 구축하기 64

3.4 텍스트를 음성 파일로 변환하는 gTTS 사용법 익히기 67

3.5 음성 파일을 텍스트로 변환하는 Whisper API 사용법 익히기 70

3.6 프로그램 UI를 생성하는 스트림릿 사용법 익히기 71

3.6.1 스트림릿 설치 72

3.6.2 스트림릿 시작하기 72

3.7 스트림릿으로 음성 비서 프로그램의 UI 만들기 75

3.7.1 스트림릿의 상태를 저장하기 위한 session\_state 함수 84

3.7.2 스트림릿 오디오 레코더를 활용하여 음성 녹음하기 86

3.8 음성 파일을 텍스트로 변환하기 88

3.9 ChatGPT API로 질문하고 답변 구하기 90

3.9.1 대화 내용을 채팅 형식으로 시각화하기 92

3.10 gTTS로 답변을 음원 파일로 만들고 재생하기 94

3.11 음성 비서 프로그램 배포하기 98

3.11.1 깃허브 리포지토리 생성하기 99

3.11.2 깃허브에 코드 및 패키지 정보 업로드하기 100

3.11.3 스트림릿 사이트와 깃허브 연동하기 103

# 3.1 음성비서 프로그램 맛보기

음성비서프로그램: <https://wikibook.co.kr/chatgpt-api-voicebot/>

각자 이용해 보세요~

## 3.2 음성 비서 프로그램의 구조

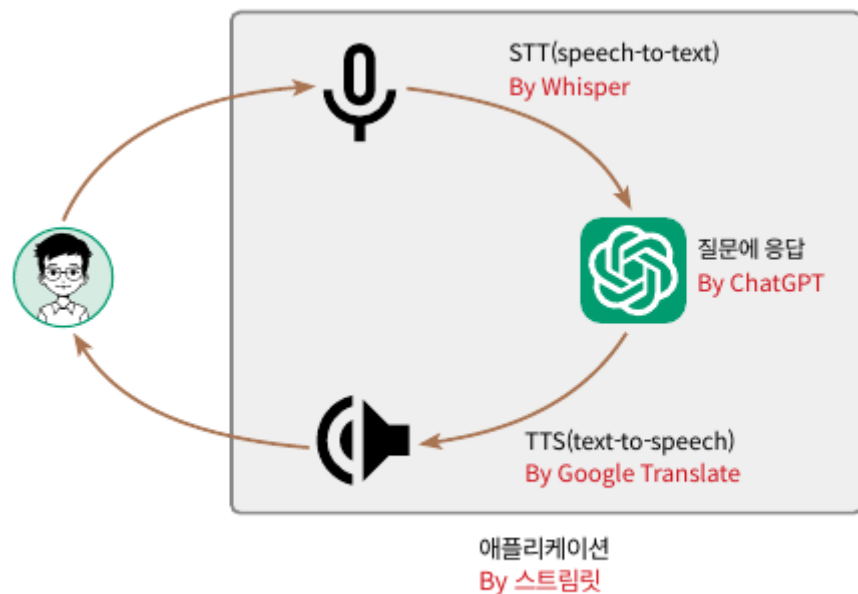


그림 3.8 음성 비서 프로그램 구조

### ■ 웹 애플리케이션 구현: 스트림릿

음성 비서 프로그램은 모두 파이썬을 활용하여 제작합니다. 기본적으로 웹 애플리케이션을 간단하게 만들 수 있게 도와주는 라이브러리인 스트림릿(Streamlit)을 활용해 구현합니다. 스트림릿을 활용하면 프론트엔드, 백엔드를 복잡하게 코딩하지 않고도 간단하게 웹 애플리케이션을 구현할 수 있습니다. 스트림릿으로 웹 애플리케이션을 개발하는 방법은 이 책의 3.6절 '프로그램 UI를 생성하는 스트림릿 사용법 익히기(71쪽)'에서 설명합니다.

### ■ 애플리케이션 배포: 깃허브, 스트림릿 클라우드

마지막으로 애플리케이션을 전 세계 사용자가 모두 사용할 수 있도록 깃허브(Github)에 코드를 올리고, 스트림릿 클라우드에 배포합니다. 애플리케이션을 인터넷에 배포하는 방법은 이 책의 3.11절 '음성 비서 프로그램 배포하기(98쪽)'에서 설명합니다.

# streamlit 사용법

streamlit 한 번에 끝내기 1강~10강(9강 제외)

[https://www.youtube.com/playlist?list=PLR4H8X\\_\\_Qok291XbTXW-9u\\_u-9XHynfmT](https://www.youtube.com/playlist?list=PLR4H8X__Qok291XbTXW-9u_u-9XHynfmT)

꼭! 한 번 이상 시청하고 오세요~

PyCharm이 아닌 VS Code로 실습함 (교재 p. 64-69 미리 읽고 오면 큰 도움)

하루 두시간,

전공 책읽기, 코딩 연습 실천해 보세요.

여러분의 위대한 도전을 응원합니다!