

chat GPT API

24-01-09

Overview

The OpenAI API is powered by a diverse set of models with different capabilities. You can also make customizations to our models for your specific use case.

MODEL	DESCRIPTION
GPT-4 and GPT-4 Turbo	A set of models that improve on GPT-3.5, designed to generate natural language or code.
GPT-3.5	A set of models that improve on GPT-3, designed to generate natural language or code.
DALL-E	A model that can generate and edit images from a text prompt.
TTS	A set of models that can convert text into natural-sounding speech.
Whisper	A model that can convert audio into text with high accuracy and low latency.

GPT-3.5

GPT-3.5 models can understand and generate natural language or code. Our most capable and cost-effective model in the GPT-3.5 family is `gpt-3.5-turbo` which has been optimized for chat using the [Chat Completions API](#) but works well for traditional completions tasks as well.

MODEL	DESCRIPTION	CONTEXT WINDOW	TRAINING DATA
gpt-3.5-turbo-1106	<div>New Updated GPT 3.5 Turbo</div> <div>The latest GPT-3.5 Turbo model with improved instruction following, JSON mode, reproducible outputs, parallel function calling, and more. Returns a maximum of 4,096 output tokens.</div> <div>Learn more.</div>	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo	Currently points to gpt-3.5-turbo-0613.	4,096 tokens	Up to Sep 2021
gpt-3.5-turbo-16k	Currently points to gpt-3.5-turbo-0613.	16,385 tokens	Up to Sep 2021
gpt-3.5-turbo-instruct	Similar capabilities as GPT-3.5-turbo models. Compatible with	4,096 tokens	Up to Sep 2021

가격 <https://openai.com/pricing>

GPT-3.5 터보

GPT-3.5 Turbo 모델은 성능이 뛰어나고 비용 효율적입니다.

`gpt-3.5-turbo-1106` 이 제품군의 주력 모델이며 16K 컨텍스트 창을 지원하고 대화에 최적화되어 있습니다.

`gpt-3.5-turbo-instruct` Instruct 모델이며 4K 컨텍스트 창만 지원합니다.

[GPT-3.5 터보에 대해 알아보기](#) ↗

모델	입력	산출
gpt-3.5-터보-1106	\$0.0010 / 1K 토큰	\$0.0020 / 1K 토큰
gpt-3.5-터보 지시	\$0.0015 / 1K 토큰	\$0.0020 / 1K 토큰

GPT-4

광범위한 일반 지식과 도메인 전문 지식을 갖춘 GPT-4는 자연어로 복잡한 지침을 따르고 어려운 문제를 정확하게 해결할 수 있습니다.

[GPT-4에 대해 알아보기](#)

모델	입력	산출
gpt-4	\$0.03 / 1K 토큰	\$0.06 / 1K 토큰
gpt-4-32k	\$0.06 / 1K 토큰	\$0.12 / 1K 토큰

토큰 <https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them>

Tokenizing

Token은 단어의 조각이라고 할 수 있습니다. 프롬프트가 입력되면 입력된 프롬프트는 모델에 처리되기전에 먼저 token으로 분할됩니다. 이렇게 token화 되는 방식은 정확하게 word만을 나누는것은 아니며 띄어쓰기, 쉼표, 복합어 등을 기준으로 쪼개질 수 있습니다.

<https://help.openai.com/en/articles/4936856-what-are-tokens-and-how-to-count-them> 에서 이야기하고 있는 token의 기준은 아래와 같습니다. [영어 기준]

- 1 token ~= 4 알파벳 (ex. have)
- 1 token ~= ¾ 단어
- 100 tokens ~= 75 단어
- 1-2 문장 ~= 30 tokens
- 1 구문 ~= 100 tokens
- 1,500 단어 ~= 2048 tokens

<https://sabarada.tistory.com/254>

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

GPT-3	Codex
<p>1. Task Definition: The task is to generate a list of 10 items that are related to the topic of "Artificial Intelligence".</p> <p>2. Input: The input is the topic "Artificial Intelligence".</p> <p>3. Output: The output is a list of 10 items related to Artificial Intelligence.</p> <p>4. Example Output:</p> <ul style="list-style-type: none"> 1. Machine Learning 2. Deep Learning 3. Natural Language Processing 4. Computer Vision 5. Robotics 6. Expert Systems 7. Fuzzy Logic 8. Knowledge Representation 9. Artificial Neural Networks 10. Expert Systems 	<p>1. Task Definition: The task is to generate a list of 10 items that are related to the topic of "Artificial Intelligence".</p> <p>2. Input: The input is the topic "Artificial Intelligence".</p> <p>3. Output: The output is a list of 10 items related to Artificial Intelligence.</p> <p>4. Example Output:</p> <ul style="list-style-type: none"> 1. Machine Learning 2. Deep Learning 3. Natural Language Processing 4. Computer Vision 5. Robotics 6. Expert Systems 7. Fuzzy Logic 8. Knowledge Representation 9. Artificial Neural Networks 10. Expert Systems

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: 🙌

Sequences of characters commonly found next to each other may be grouped together: 1234567890

Clear Show example

Tokens	Characters
64	252

Many words map to one token, but some don't: indivisible.

Unicode characters like emojis may be split into many tokens containing the underlying bytes: `👉👉👉👉👉👉`

Sequences of characters commonly found next to each other may be grouped together: 1234567890

TEXT	TOKEN IDS
1. The cat sat on the mat.	[1, 2, 3, 4, 5, 6, 7]
2. The dog barked at the cat.	[8, 9, 10, 11, 12, 13]
3. The cat sat on the mat.	[1, 2, 3, 4, 5, 6, 7]
4. The dog barked at the cat.	[8, 9, 10, 11, 12, 13]
5. The cat sat on the mat.	[1, 2, 3, 4, 5, 6, 7]
6. The dog barked at the cat.	[8, 9, 10, 11, 12, 13]
7. The cat sat on the mat.	[1, 2, 3, 4, 5, 6, 7]
8. The dog barked at the cat.	[8, 9, 10, 11, 12, 13]
9. The cat sat on the mat.	[1, 2, 3, 4, 5, 6, 7]
10. The dog barked at the cat.	[8, 9, 10, 11, 12, 13]

Tokenizer

The GPT family of models process text using **tokens**, which are common sequences of characters found in text. The models 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 would be tokenized by the API, and the total count of tokens in that piece of text.

GPT-3 Codex

오픈에이아이([OpenAI](#), [openai.com](#))가 개발한 대화 전문 인공지능은 **챗봇**으로, **챗**은 채팅의 줄임말이고 **GPT**는 'Generated Pre-trained Transformer'의 앞 글자를 딴 것이다. **챗GPT**는 사용자가 **대화창**에 텍스트를 입력하면 그에 맞춰 대화를 함께 나누는 서비스로, 공개 단 5일 만에 하루 이용자가 **100만** 명을 돌파하면서 돌풍을 일으키기 시작했다. 특히 질문에 대한 답변은 물론 논문 작성, 번역, 노래 작사·작곡, 코딩 작업 등 광범위한 분야의 업무 수행까지 가능하다는 점에서 기존 **AI**와는 확연히 다른 면모를 보이고 있다.

Clear Show example

Tokens	Characters
542	314

(openAI, openai.com) GPT 'Generated Pre-
 trained Transformer' GPT
 50 100
 AI

TEXT TOKEN IDS

한글의 경우 토큰 수가 더 많이 소비 된다.

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

- openAI에서 제공하는 모델 테스트 가능

Playground

Your presets

Save

View code

Share

...

SYSTEM

You are a helpful assistant.

USER

아침에는 네 발로 기고 오후에는 두 발로 걷고 저녁에는 세 발로 걷는 동물은?

ASSISTANT

인간입니다. 이는 그리스 신화 속에 등장하는 스팅크스의 수수께끼입니다.

ASSISTANT

스핑크스는 사람 얼굴과 사자 몸체를 가진 생물로, 그림자 속에 없어지기 전까지 동물 수수께끼를 가지고 사람들을 시험했었습니다. 이 수수께끼는 "아침에는 네 발로 기고 오후에는 두 발로 걷고 저녁에는 세 발로 걷는 동물은 무엇인가?"였습니다. 응답은 인간이었고, 아침에는 네 발로 기어가고(어릴 때는 기어가는 단계), 오후에는 두 발로 걷고, 저녁에는 세 발로 걷기 때문입니다.

⊕ Add message

Submit

🔄

📝 Give us feedback

Mode

Chat

Model

gpt-3.5-turbo

Temperature

1

Maximum length

256

Stop sequences

Enter sequence and press Tab

Top P

1

Frequency penalty

0

Presence penalty

0

Temperature(온도)는 0~1 사이의 값이다.

- 숫자가 클수록 따뜻하고, 재치 있고, 창의적이고, 아무 말이나 하는 경향이 있다.
- 숫자가 작을수록 차갑고, 재미 없고, 사무적이고, 옳은 말을 하는 경향이 있다.

번역 작업을 예로 들면, 온도를 0으로 설정하면 항상 똑같은 결과가 나온다. 숫자를 높일수록 자연스러운 문장이 나오지만 그때 그때 답이 다르고 틀린 답을 할 가능성이 높아진다.

Top P : Top P 숫자를 높이면 다양한 답을 얻을 수 있다
<https://wikidocs.net/218344>

사용 예시

<https://wikidocs.net/195818>

배경 설명

1. openai 라이브러리 설치

빠른 시작 언어 선택

OpenAI API 사용을 시작하려는 도구나 언어를 선택하세요.

곱슬 곱슬하다 파이썬 **Node.js**

Node.js는 웹 개발에 일반적으로 사용되는 인기 있는 JavaScript 프레임워크입니다. OpenAI는 JavaScript에서 OpenAI API를 간단하고 효율적으로 작업할 수 있게 해주는 맞춤형 **Node.js/TypeScript 라이브러리를 제공합니다.**

1단계: 노드 설정

> Node.js 설치

▼ OpenAI Node.js 라이브러리 설치

Node.js를 설치하면 OpenAI Node.js 라이브러리를 설치할 수 있습니다. 터미널/명령줄에서 다음을 실행합니다.

```
1 npm install --save openai
2 # or
3 yarn add openai
```

2. API 요청하기

▼ API 요청하기

Node.js를 구성하고 API 키를 설정한 후 마지막 단계는 Node.js 라이브러리를 사용하여 OpenAI API에 요청을 보내는 것입니다. 이렇게 하려면 `openai-test.js` 터미널이나 IDE를 사용하여 이름이 지정된 파일을 만듭니다.

파일 내부에 아래 예 중 하나를 복사하여 붙여넣습니다.

```
채팅 완료 ▼   복사

1  import OpenAI from "openai";
2
3  const openai = new OpenAI();
4
5  async function main() {
6      const completion = await openai.chat.completions.create({
7          messages: [{ role: "system", content: "You are a helpful assistant." }],
8          model: "gpt-3.5-turbo",
9      });
10
11      console.log(completion.choices[0]);
12  }
13
14  main();
```

코드를 실행하려면 `node openai-test.js` 터미널/명령줄에 입력하세요.

Role

<https://platform.openai.com/docs/api-reference/chat/create>

1. user
 - 사용자, 질문자
2. Assistant
 - 대화의 연속성을 유지하기 위해 사용
3. system
 - 역할 설정하여 답변 받음

▼ API 요청하기

Node.js를 구성하고 API 키를 설정한 후 마지막 단계는 Node.js 라이브러리를 사용하여 OpenAI API에 요청을 보내는 것입니다. 이렇게 하려면 `openai-test.js` 터미널이나 IDE를 사용하여 이름이 지정된 파일을 만듭니다.

파일 내부에 아래 예 중 하나를 복사하여 붙여넣습니다.

```
채팅 완료 ▼ 복사
1  import OpenAI from "openai";
2
3  const openai = new OpenAI();
4
5  async function main() {
6    const completion = await openai.chat.completions.create({
7      messages: [{ role: "system", content: "You are a helpful assistant." }],
8      model: "gpt-3.5-turbo",
9    });
10
11    console.log(completion.choices[0]);
12  }
13
14  main();
```

코드를 실행하려면 `node openai-test.js` 터미널/명령줄에 입력하세요.

▲ request curl

Request

- 참고 문서 :

<https://velog.io/@dy6578ekdb/%ED%94%84%EB%A1%A0%ED%8A%B8%EC%97%94%EB%93%9C-%EA%B0%9C%EB%B0%9C%EC%9E%90%EB%A5%BC-%EC%9C%84%ED%95%9C-chatGPT-API-%EC%82%AC%EC%9A%A9%EB%B2%95-React-Flask>

- 공식 문서 :

<https://platform.openai.com/docs/api-reference/chat/create>


```
prompt = f"대한민국의 대통령은 누구입니까?"
# set api key
openai.api_key = API_KEY
# Call the chat GPT API
completion = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[
        {"role": "user", "content": prompt},
    ],
    temperature=0.8,
    max_tokens=10
)
```

▲ 파이썬 요청 예제

Example request

gpt-3.5-turbo ▾

curl ▾

 Copy

```
1  curl https://api.openai.com/v1/chat/completions \
2    -H "Content-Type: application/json" \
3    -H "Authorization: Bearer $OPENAI_API_KEY" \
4    -d '{
5      "model": "gpt-3.5-turbo",
6      "messages": [
7        {
8          "role": "system",
9          "content": "You are a helpful assistant."
10       },
11       {
12         "role": "user",
13         "content": "Hello!"
14       }
15     ]
16   }'
```

▲ request curl

Response

```
{
  "result": {
    "choices": [
      {
        "finish_reason": "stop",
        "index": 0,
        "message": {
          "content": "저는 인공지능 어시스턴트이기 때문에 현재 시점에서 대통",
          "role": "assistant"
        }
      }
    ],
    "created": 1684061777,
    "id": "chatcmpl-7G3gfKva4UNOd9dzcnBn0BYlJel3j",
    "model": "gpt-3.5-turbo-0301",
    "object": "chat.completion",
    "usage": {
      "completion_tokens": 82,
      "prompt_tokens": 27,
      "total_tokens": 109
    }
  }
}
```

▲ 응답 예제

-content만 뽑아 사용

```
completion["choices"][0]["message"]["content"].encode("utf-8").decode()
```

Response



```
1  {
2    "id": "chatcmpl-123",
3    "object": "chat.completion",
4    "created": 1677652288,
5    "model": "gpt-3.5-turbo-0613",
6    "system_fingerprint": "fp_44709d6fcb",
7    "choices": [{
8      "index": 0,
9      "message": {
10        "role": "assistant",
11        "content": "\n\nHello there, how may I assist
12      },
13      "logprobs": null,
14      "finish_reason": "stop"
15    }],
16    "usage": {
17      "prompt_tokens": 9,
18      "completion_tokens": 12,
19      "total_tokens": 21
20    }
21  }
```

▲ response curl

다른 GPT API 모델 <https://platform.openai.com/docs/models>

1. Davinci

- 주요 기능: 1,750억 개의 매개변수가 있는 가장 크고 가장 유능한 GPT-3 모델로, 매우 일관되고 상황에 맞는 응답을 생성합니다.
- 작동 방식: 입력 프롬프트를 분석하고 방대한 양의 데이터에서 학습한 패턴을 기반으로 응답을 생성하여 작동합니다.
- 적용 사례: Davinci는 언어 번역, 콘텐츠 생성 및 제품 추천과 같은 복잡한 작업에 이상적이라 할 수 있습니다.
- 비용: 0.02달러 / 1000 토큰.

2. Curie

- 주요 기능: 67억 개의 매개변수가 있는 GPT-3 모델로, 컨텍스트 이해가 필요한 자연어 처리 작업에 이상적입니다.
- 작동 방식: 입력 프롬프트를 분석하고 방대한 양의 데이터에서 학습한 패턴을 기반으로 응답을 생성하여 작동합니다.
- 적용 사례: Curie는 챗봇, 콘텐츠 생성 및 언어 번역에 이상적입니다.
- 비용: 0.002달러 / 1000 토큰.

3. Babbage

- 주요 기능: 13억 개의 매개변수가 포함된 GPT-3 모델로, 더 단순한 자연어 처리 작업에 이상적입니다.
- 작동 방식: 입력 프롬프트를 분석하고 방대한 양의 데이터에서 학습한 패턴을 기반으로 응답을 생성하여 작동합니다.
- 적용 사례: Babbage는 간단한 챗봇, 언어 번역 및 Q&A 애플리케이션에 이상적입니다.
- 비용: 0.005달러 / 1000 토큰.

4. Ada

- 주요 기능: 12억 개의 매개변수가 포함된 GPT-3 모델로, 텍스트 프롬프트에 대해 일관되고 적절한 응답을 생성하는 데 이상적입니다.
- 작동 방식: 입력 프롬프트를 분석하고 방대한 양의 데이터에서 학습한 패턴을 기반으로 응답을 생성하여 작동합니다.
- 적용 사례: Ada는 챗봇, 콘텐츠 생성 및 언어 번역에 이상적입니다.
- 비용: 0.0004달러 / 1000 토큰.

참고 영상 (웹)

- https://www.youtube.com/watch?v=LX_DXLlaymg

실습 코드 (챗비티와 간단한 채팅)

- <https://www.youtube.com/watch?v=8L3M8G1UsYg&t=575s>
- https://github.com/LaurentiuGabriel/chatbot_demo/blob/main/Main.js

참고1 (json)

- <https://hdhdeveloper.tistory.com/136>

참고 2 (파이썬)

- <https://velog.io/@dy6578ekdb/%ED%94%84%EB%A1%A0%ED%8A%B8%EC%97%94%EB%93%9C-%EA%B0%9C%EB%B0%9C%EC%9E%90%EB%A5%BC-%EC%9C%84%ED%95%9C-chatGPT-API-%EC%82%AC%EC%9A%A9%EB%B2%95-React-Flask>
- <https://velog.io/@yule/OpenAI-API-%EB%B0%9C%EA%B8%89>

최종 사용자 ID

- <https://platform.openai.com/docs/guides/safety-best-practices?lang=curl>

포스트맨

- <https://binit.tistory.com/17>