OpenAI API 사용 기초

필요 라이브러리 설치

```
!pip install openai
```

다음을 코드 앞에서 추가

```
import openai
openai.api_key = "sk-..."
```

토큰

자연어는 토큰으로 쪼개어져서 처리됨

```
I'm a boy -> I 'm a boy
나는 소년입니다 -> 나 는 소년 이 ㅂ 니다
```

OpenAI 에서는 tiktoken이라는 라이브러리를 사용하여 처리함

```
!pip installl tiktoken
```

인코딩

```
encoding = tiktoken.get_encoding("cl100k_base")
print(encoding.encode("I'm a boy"))
```

```
[40, 2846, 264, 8334]
```

디코딩

```
print(encoding.decode([40]))
print(encoding.decode([2846]))
print(encoding.decode([264]))
print(encoding.decode([8334]))
```

```
I
'm
a
boy
```

모델 별로 사용되는 인코딩들이 정해져 있다.

Encoding name	OpenAI models
`cl100k_base`	`gpt-4`, `gpt-3.5-turbo`, `text-embedding-ada-002`
`p50k_base`	Codex models, `text-davinci-002`, `text-davinci-003`
`r50k_base` (or `gpt2`)	GPT-3 models like `davinci`

입력받은 문장을 토큰으로 짜른 후에 처리한다.

실제 과금을 할때에도 토큰을 단위로 처리.

Model	Base models	Fine-tuned models	
	Usage (답변 길이에 따른 과금)	Training (학습 데이터에 따른 과금)	Usage (답변 길이에 따른 과금)
(GPT-3) Ada : fastest	$0.0004 / 1K tokens	$0.0004 / 1K tokens	$0.0016 / 1K tokens
(GPT-3) Babbage	$0.0005 / 1K tokens	$0.0006 / 1K tokens	$0.0024 / 1K tokens
(GPT-3) Curie	$0.0020 / 1K tokens	$0.0030 / 1K tokens	$0.0120 / 1K tokens
(GPT-3) Davinci	$0.0200 / 1K tokens	$0.0300 / 1K tokens	$0.1200 / 1K tokens
(GPT-3.5) turbo	$0.0020 / 1K tokens	No service provided	
(GPT-4) 8K	Prompt $0.03 / 1K tokens Completion $0.06 / 1K to kens		
(GPT-4) 32K	Prompt $0.06 / 1K tokens Completion $0.12 / 1K to kens		

각 토큰은 토큰 그대로 사용하지 않고, 토큰 인덱스 번호로 사용된다.

임베딩

토큰은 특정 길이 숫자들(벡터)로 임베딩 된다.

비슷한 문맥의 토큰들이 비슷한 곳에 위치함

토큰 뿐 아니라 문장 자체를 임베딩 하기도 함

호출 시의 파라매터

기본 호출

```
MODEL = "gpt-3.5-turbo"
response = openai.ChatCompletion.create(
    model=MODEL,
    messages=[
        {"role": "system", "content": "You are a helpful assistant."},
        {"role": "user", "content": "Knock knock."},
        {"role": "assistant", "content": "Who's there?"},
        {"role": "user", "content": "Orange."},
    ],
```

```
    temperature=0,
)
```

```
{
  "choices": [
    {
      "finish_reason": "stop", # "stop" 내지는 최대 토큰
      "index": 0,
      "message": {
        "content": "Orange who?",
        "role": "assistant"
      }
    }
  ],
  "created": 1684925547,
  "id": "chatcmpl-7JgORG1viLggNxfJlhMmmc4zgtnlL", # request ID
  "model": "gpt-3.5-turbo-0301", # 모델 이름
  "object": "chat.completion", # 결과 객체 이름
  "usage": {
    "completion_tokens": 3,
    "prompt_tokens": 39,
    "total_tokens": 42
  }
}
```

좋은 예제 : https://github.com/openai/openai-cookbook/blob/main/examples/How_to_format_inputs_to_ChatGPT_models.ipynb

3가지 role

- system : 시스템의 역할/행위자(personalities or behaviors)를 설정한다.
- assistant : 모델이 이전에 출력한 응답
- user : 사용자의 입력

Zero/One-shot/Few-shot Learning

방법	내용
Prompt	**Write a short alliterative sentence about a curious cat exploring a garden** * alliterative(두운): 문장 내 각 단어마다 첫 글자를 동일하게 하여, 반복을 통해 운율을 형성
Zero-shot learning	**[Prompt에 포함시킨 예시 정보]** - **[ChatGPT의 답변]** A cat looks at flowers in the garden
One-shot learning	**[Prompt에 포함시킨 예시 정보]** Peter Piper picked a peck of pickled peppers. **[ChatGPT의 답변]** Curious cat cautiously checking colorful cabbages.
Few-shot learning	**[Prompt에 포함시킨 예시 정보]** Example 1: Peter Piper picked a peck of pickled peppers. Example 2: She sells seashells by the seashore. Example 3: How can a clam cram in a clean cream can? **[ChatGPT의 답변]** Curious cat crept cautiously, contemplating captivating, colorful carn ations.

prompt에서 주는 샘플

from https://www.samsungsds.com/kr/insights/chatgpt_whitepaper2.html?moreCnt=0&backTypeId=&category=

Few Shot Learning 호출 예

```
# An example of a faked few-shot conversation to prime the model into translating business jargon to simpler speech
response = openai.ChatCompletion.create(
    model=MODEL,
    messages=[
        {"role": "system", "content": "You are a helpful, pattern-following assistant."},
        {"role": "user", "content": "Help me translate the following corporate jargon into plain English."},
        {"role": "assistant", "content": "Sure, I'd be happy to!"},
        {"role": "user", "content": "New synergies will help drive top-line growth."},
        {"role": "assistant", "content": "Things working well together will increase revenue."},
        {"role": "user", "content": "Let's circle back when we have more bandwidth to touch base on opportunities for increased leverag
e."},
        {"role": "assistant", "content": "Let's talk later when we're less busy about how to do better."},
        {"role": "user", "content": "This late pivot means we don't have time to boil the ocean for the client deliverable."},
    ],
    temperature=0,
)

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