OpenAI APIの使い方

OpenAI APIをPythonで使うためには、openaiという外部ライブラリが必要です。

このライブラリはGoogle Colaboratoryには最初からはインストールされていないので、これをインストールします。

```
[ ] !pip install openai
```

```
Collecting openai
  Downloading openai-0.27.9-py3-none-any.whl (75 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 75.5/75.5 kB 1.2 MB/s eta 0:00:00
Requirement already satisfied: requests>=2.20 in /usr/local/lib/python3.10/dist-packages (from openai) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from openai) (4.66.1)
Requirement already satisfied: aiohttp in /usr/local/lib/python3.10/dist-packages (from openai) (3.8.5)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (3.2.0)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (3.4)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (2.0.4)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests>=2.20->openai) (2023.7.22)
Requirement already satisfied: attrs>=17.3.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (23.1.0)
Requirement already satisfied: multidict<7.0,>=4.5 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (6.0.4)
Requirement already satisfied: async-timeout<5.0,>=4.0.0a3 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (4.0.3)
Requirement already satisfied: yarl<2.0,>=1.0 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (1.9.2)
Requirement already satisfied: frozenlist>=1.1.1 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (1.4.0)
Requirement already satisfied: aiosignal>=1.1.2 in /usr/local/lib/python3.10/dist-packages (from aiohttp->openai) (1.3.1)
Installing collected packages: openai
Successfully installed openai-0.27.9
```

先ほどインストールしたopenaiをインポートします。

```
[ ] import openai
```

次に、API Keyを設定します。

これはOpenAI APIを利用するための認証キーであるため、他人に公開してはいけません

```
[ ] openai.api_key = "<API Key>"
```

ChatGPTと会話をするには、`ChatCompletion` という機能のAPIを使います。

このとき、

- モデルの種類
- 会話

を指定して実行します。

今回はChatGPTを使うため、`model="gpt-3.5-turbo"` と指定します。

会話の内容は `messages` というパラメータで指定します。

`messages` には辞書のリストを指定し、その辞書には `"role"` と `"content"` という2つのキーを指定します。

まずは

- `"role": "user"`
- `"content": "OpenAIについて教えてください"`

という辞書が1つだけ入ったリストを `"messages"` に指定して実行してみましょう。

```
[ ]  response1 = openai.ChatCompletion.create(
         model="gpt-3.5-turbo",
         messages=[
             {"role": "user", "content": "OpenAIについて教えてください"}
         ]
     )
```

`response1` の中身を確認します。

```
[ ]  response1
```

```
<OpenAIObject chat.completion id=chatcmpl-7qzSPN0a8n69OKysrobdRGqmBihy2 at 0x7c42404b1850> JSON: {
  "id": "chatcmpl-7qzSPN0a8n69OKysrobdRGqmBihy2",
  "object": "chat.completion",
  "created": 1692863653,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
```

```
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content":
"OpenAI\u306f\u4eba\u5de5\u77e5\u80fd\uff08AI\uff09\u6280\u8853\u306e\u7814\u7a76\u3068\u958b\u767a\u3092\u884c\u3046\u4f01\u696d\u3067\u3059\u3002OpenAI\u306f\u3001\u65b0\u3
Pre-trained Transformer\uff09\u3092\u958b\u767a\u3057\u3001\u305d\u306e\u6539\u826f\u7248\u3067\u3042\u308b\u300cGPT-
3\u300d\u306f\u7279\u306b\u6ce8\u76ee\u3092\u96c6\u3081\u307e\u3057\u305f\u3002GPT-
3\u306f\u3001\u5de8\u5927\u306a\u30c7\u30fc\u30bf\u30bb\u30c3\u30c8\u3092\u5b66\u7fd2\u3057\u3066\u751f\u6210\u3055\u308c\u305f\u6587\u7ae0\u3092\u751f\u6210\u3057\u3001\u3
Programming
Interface\uff09\u3092\u63d0\u4f9b\u3057\u3001\u958b\u767a\u8005\u304cGPT\u306a\u3069\u306e\u6a5f\u80fd\u3092\u81ea\u8eab\u306e\u30a2\u30d7\u30ea\u30b1\u30fc\u30b7\u30e7\u30f3\u
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 18,
    "completion_tokens": 541,
    "total_tokens": 559
  }
}
```


ChatGPTの出力は `response1.choices[0]["message"]["content"]` に入っているため、それを出力してみます。

```
[ ] print(response1.choices[0]["message"]["content"])
```

OpenAIは人工知能（AI）技術の研究と開発を行う企業です。OpenAIは、新たなAIの進歩が協力と共有によって進められるべきであるという考えから生まれました。そのため、技術の商業利用による利益を最大化することを目的とする従来の企業とは異なり

OpenAIは、自己学習AIシステム「GPT」（Generative Pre-trained Transformer）を開発し、その改良版である「GPT-3」は特に注目を集めました。GPT-3は、巨大なデータセットを学習して生成された文章を生成し、さまざまなタスクに使用することができ

OpenAIは、技術の進歩が公共の利益を最大化する方法を追求するため、AIの研究成果を公開することを重視しています。また、法的、倫理的な問題に対処するための取り組みも行っています。

OpenAIの研究成果やテクノロジーは、研究者や開発者、企業、一般の人々に利用可能です。また、OpenAIはAPI（Application Programming Interface）を提供し、開発者がGPTなどの機能を自身のアプリケーションに統合することができるようにしていま

OpenAIの目標は、人工知能の進歩を通じて、広い範囲の問題を解決し、持続可能な未来を実現することです。

ChatGPTの出力文が長いため、続けて要約をさせてたいと思います。

そこで、次はChatGPTに対して「もっと簡潔に説明してください」と指示を出してみます。

先ほど実行したコードの `"content"` の中身を変えて実行します。

```
[ ] response2 = openai.ChatCompletion.create(
```

```
[ ] response2 = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "user", "content": "もっと簡潔に説明してください"}
        ]
    )

    print(response2.choices[0]["message"]["content"])
```

```
    もっと短く説明します。
```

すると、うまく要約できませんでした。

これは、ChatGPTはこれまでの会話の内容を知らないからです。

ChatCompletionで、これまでの会話の内容を加味したタスクを行うには、これまでの会話の内容を教えてあげる必要があります。

そこで、`messages` を以下のように指定します。

```
messages=[
        {"role": "user", "content": "OpenAIという会社について教えてください"},
        {"role": "assistant", "content": response1.choices[0]["message"]["content"]},
        {"role": "user", "content": "もっと簡潔に説明してください"}
    ]
```

`messages` には3つの辞書が入ったリストを指定します。

それぞれの辞書には `"role"` と `"content"` という2つのキーを指定します。

`"role"` の値によって、`"content"` に書くべき内容が変わります。

- `"role"` : `"user"` の場合 : `"content"` に質問や会話の内容を入力する
- `"role"` : `"assistant"` の場合 : LLMを指す。`"content"` にuserに対する返答を定義する
- `"role"` : `"system"` の場合 : どのように返答して欲しいかというメタ情報や設定情報を指定する。役割設定で使うことが多い。

```
[ ] response3 = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
```

```
[ ] response3 = openai.ChatCompletion.create(
        model="gpt-3.5-turbo",
        messages=[
            {"role": "user", "content": "OpenAIという会社について教えてください"},
            {"role": "assistant", "content": response1.choices[0]["message"]["content"]},
            {"role": "user", "content": "もっと簡潔に説明してください"}
        ]
    )
```

```
[ ] print(response3.choices[0]["message"]["content"])
```

```
OpenAIはAI技術の研究と開発を行う企業で、特にGPTという自己学習AIシステムが有名です。彼らはAIの研究成果を公開し、共有することを重視しています。また、APIを提供して開発者が彼らの技術を利用できるようにしています。彼らの目標は、AIを使
```

これまでの会話の内容を加味して要約させることができました。

最後に、Usageにアクセスし、何ドル分使用したのか確認しましょう。

数円分しか使っていないと思います。

```
[ ]
```