### モジュール3 プロンプトエンジニアリング



### Azure OpenAl Service を使用してプロンプト エンジニアリングを適用する

1時間3分・モジュール・7ユニット

**★★★★** 4.8 (1,290)

中級 Al エンジニア 開発者 ソリューション アーキテクト 学生 Azure OpenAl Service Azure

Azure OpenAI でのプロンプト エンジニアリングは、自然言語処理モデルのプロンプトの設計を含む 1 つの手法です。 このプロセスにより、応答の精度と関連性が向上し、モデルのパフォーマンスが 最適化されます。

800 XP

#### 学習の目的

このモジュールを完了すると、次のことができるようになります。

- プロンプト エンジニアリングの概念と、Azure OpenAI モデルのパフォーマンスを最適化するためのその役割について理解する。
- AI モデルをより適切に活用するためにプロンプトを設計および最適化する方法を知る。
- 明確な指示を含め、出力構成を要求し、コンテキストに基づく内容を使用してモデルの応答の品質を向上させる。

https://learn.microsoft.com/ja-jp/training/modules/apply-prompt-engineering-azure-openai/

### モジュール3

- プロンプトとは?
- プロンプトエンジニアリングとは?
- プロンプトの基本
- Chatプレイグラウンドの利用
- 複雑なタスクを分割する
- 知識チェック
- 講師デモ

### モジュール3

- ・プロンプトとは?
- プロンプトエンジニアリングとは?
- プロンプトの基本
- Chatプレイグラウンドの利用
- 複雑なタスクを分割する
- 知識チェック
- 講師デモ

### プロンプトとは?

- ・人工知能(生成AI)に対する指示や質問。
- ・生成AIはプロンプトに基づいて、回答を出力する
- ・生成AIをうまく使いこなすには、プロンプトの書き方が重要と なる。
- AIに対するプロンプトを開発する専門家「プロンプトエンジニア」という職種も誕生している。
  - アメリカでは今、5000万円という高額報酬の求人まで飛び出すなど、 需要が急速に拡大しています(NHK)
  - https://www3.nhk.or.jp/news/html/20230518/k10014071011000.html

### (参考) さまざまなテクニック

#### ・深津式プロンプト

- ・システム1:「命令書」「制約条件」「入力文」「出力文」
- システム2:「逆質問」
- より高度な敬語でAIに接するほどに出力が良くなる
  - <a href="https://chatgpt-lab.com/n/nd8dde92016ec">https://chatgpt-lab.com/n/nd8dde92016ec</a>
- ・「自分を信じて限界を超えてください」「成長の機会だと捉えて挑戦してください」など**感情をグッと込めたプロンプトを添える**と、LLMの出力の精度が向上。Microsoft**などの研究グループによる発表** 
  - https://note.com/akira570/n/n374b0fd79bb1

### モジュール3

- プロンプトとは?
- ・プロンプトエンジニアリングとは?
- プロンプトの基本
- Chatプレイグラウンドの利用
- 複雑なタスクを分割する
- 知識チェック
- 講師デモ

### プロンプトエンジニアリングとは?

- 人工知能(生成AI)に対する指示や質問(プロンプト)を設計 する技術。
- チャットボットは部下のようなものであり、上司が部下に対して適切な指示を出すことで、部下はより効率的に働くことができる。プロンプトエンジニアリングは、「部下に対する適切な指示の出し方の技術」に相当する。
- あるケースでうまく働くプロンプトがあったとしても、別の ケースではうまく働かない可能性がある。良いプロンプトを開 発するには、プロンプトエンジニアリングの定番の方法に加え て、試行錯誤も必要となる。

### モジュール3

- プロンプトとは?
- プロンプトエンジニアリングとは?
- ・プロンプトの基本
- Chatプレイグラウンドの利用
- 複雑なタスクを分割する
- 知識チェック
- 講師デモ

### プロンプトの基本

- 具体的な指示を出す
- プライマリコンテンツ、サポートコンテンツ、基礎コンテンツ を指定する
- セクションマーカーを使用する
- キューを使用する
- ・ 出力構造を指定する

### プロンプトの基本

- ・具体的な指示を出す
- プライマリコンテンツ、サポートコンテンツ、基礎コンテンツ を指定する
- セクションマーカーを使用する
- キューを使用する
- ・ 出力構造を指定する

# 具体的な指示を出す(Provide clear instructions)

- 例えば、生成AIに対するプロンプトとして「記事を書いて」と 指示すると、AIは何について記事を書けばいいのか分からず、 期待通りの結果を返せない可能性がある。
- この場合「環境問題についての記事を書いて」や「環境問題が 人類に与える影響について記事を書いて」と、より具体的に指 示することで、AIは環境問題についての記事を生成することが 可能になる。

### プロンプトの基本

- 具体的な指示を出す
- ・プライマリコンテンツ、サポートコンテンツ、基礎コンテンツ を指定する
- セクションマーカーを使用する
- キューを使用する
- ・ 出力構造を指定する

## プロンプトをプライマリコンテンツ、サポートコンテンツ、基礎コンテンツを指 定する

- プライマリコンテンツ(primary content)
  - ・ユーザーがAIに送信する具体的な要求や命令、あるいは処理対象の文章。
- サポートコンテンツ(supporting content)
  - プライマリコンテンツを補完するための追加情報。指示 (instructions) とも。
- 基礎コンテンツ(grounding content)
  - ・ユーザーとAIが共有する前提知識や背景情報。

Azure OpenAI processes text by breaking it down into tokens. Tokens can be words or just chunks of characters. For example, the word "hamburger" gets broken up into the tokens "ham", "bur" and "ger", while a short and common word like "pear" is a single token.

====

この文章を日本語に翻訳してください。 文章の中の例として示される英単語は`でくくり、そ の前後にスペースを入れてください。ただし、一般的 な名詞は、`でくくらないでください。

====

たとえば Azure OpenAI は、一般的な名詞の例です。

Azure OpenAI processes text by breaking it down into tokens. Tokens can be words or just chunks of characters. For example, the word "hamburger" gets broken up into the tokens "ham", "bur" and "ger", while a short and common word like "pear" is a single token.

プライマリ コンテンツ (翻訳対象の 文章)

====

この文章を日本語に翻訳してください。 文章の中の例として示される英単語は`でくくり、そ の前後にスペースを入れてください。ただし、一般的 な名詞は、`でくくらないでください。

サポート コンテンツ (指示)

====

たとえば Azure OpenAI は、一般的な名詞の例です。

基礎 コンテンツ (前提知識)

Q

Azure OpenAI processes text by breaking it down into tokens. Tokens can be words or just chunks of characters. For example, the word "hamburger" gets broken up into the tokens "ham", "bur" and "ger", while a short and common word like "pear" is a single token.

====

この文章を日本語に翻訳してください。 文章の中の例として示される英単語は`でくくり、その前後 にスペースを入れてください。ただし、一般的な名詞は、`で くくらないでください。

====

たとえば Azure OpenAI は、一般的な名詞の例です。



Azure OpenAIはテキストをトークンに分解して処理します。トークンは単語または文字のチャンクになります。例えば、hamburger という単語は ham、bur、ger というトークンに分解されます。一方、pear のような短く一般的な単語は1つのトークンとなります。

「`」でくくられた部分は、 Chatプレイグラウンドで は青字の等幅フォントで 表示される。

### プロンプトの基本

- 具体的な指示を出す
- プライマリコンテンツ、サポートコンテンツ、基礎コンテンツ を指定する
- セクションマーカーを使用する
- キューを使用する
- ・出力構造を指定する

セクション マーカー (区切り線) Azure OpenAI processes text by breaking it down into tokens. Tokens can be words or just chunks of characters. For example, the word "hamburger" gets broken up into the tokens "ham", "bur" and "ger", while a short and common word like "pear" is a single token.

====

この文章を日本語に翻訳してください。 文章の中の例として示される英単語は、でくくり、そ の前後にスペースを入れてください。ただし、一般的 な名詞は、、でくくらないでください。

====

たとえば Azure OpenAI は、一般的な名詞の例です。

### 区切り線の書き方

結論として、ChatGPT API での区切り線は、 - または = を4以上の2の冪乗(4, 8, 16, 32)文字数 分だけ書くのがおすすめです。

つまり、以下の区切り線が最有力です。

| ====                                    |  |
|---|--|
|   |  |
| ======                                  |  |
|   |  |
|   |  |
| ======================================= |  |
|   |  |
| ======================================= |  |
|   |  |

これらは1トークンとしてカウントされるため効果的です。

### プロンプトの基本

- 具体的な指示を出す
- プライマリコンテンツ、サポートコンテンツ、基礎コンテンツ を指定する
- セクションマーカーを使用する
- キューを使用する
- ・ 出力構造を指定する

キュー(cue): プロンプトの末尾に書く、モデルを目的の出力に誘導する言葉や文。 Completions API(入力候補)でよく使用される。

プロンプト

1から100まで合計するコードを書いてください。 ====

#include <stdio.h>

Completions APIが 生成した、 「キュー」に続く 部分

```
int main(){
  int i,sum=0;
  for(i=1;i<=100;i++){}
     sum += i;
  printf("%d\n",sum);
  return 0;
```

### プロンプトの基本

- 具体的な指示を出す
- プライマリコンテンツ、サポートコンテンツ、基礎コンテンツ を指定する
- セクションマーカーを使用する
- キューを使用する
- ・出力構造を指定する

#### 出力構造の指 定

C#とPythonの違いをまとめてください。1列目は比較の観点、2列目はC#について、3列目はPythonについて書いてください。Markdownを使用して表形式で出力してください。



. . .

| 比較の観点      | C#                                   | Python                        |
|------------|--------------------------------------|-------------------------------|
| 開発元        | マイクロソフト                              | グイド・ヴァンロ<br>ッサム               |
| 実行環境       | .NET Framework                       | Pythonインタプリ<br>タ              |
| オブジェクト指向言語 | クラスベース                               | クラスベース                        |
| 開発用ツール     | Visual Studio                        | -                             |
| 開発対象       | Windowsアプリケー<br>ションやビジネスア<br>プリケーション | Webアプリケーションやデータサイ<br>エンス、機械学習 |
| 構文の厳密さ     | 厳密                                   | 柔軟                            |

### モジュール3

- プロンプトとは?
- プロンプトエンジニアリングとは?
- プロンプトの基本
- ・Chatプレイグラウンドの利用
- 複雑なタスクを分割する
- 知識チェック
- 講師デモ

### Chat プレイグラウンドの利用

- システムメッセージを指定する
- 例を指定する(フューショット学習)
- 会話履歴を利用する

Deploy to ∨

#### プレイグラウンド

#### ∮ チャット

82 入力候補

■ DALL·E (Preview)

#### 管理

**"** デプロイ

🕞 モデル

Data files

Quotas

Content filters (Preview)

システム メッセージ

例 (フューショッ 卜学習)

#### Chat playground

アシスタントのセットアップ  $\times$ □ 変更の保存 Specify how the chat should act Use a template to get started, or just start writing your own system message below. Want some tips? <u>詳細情報</u> Use a system message template Default システム メッセージ ① You are an AI assistant that helps people find information. 例① ユーザー: アシスタント:

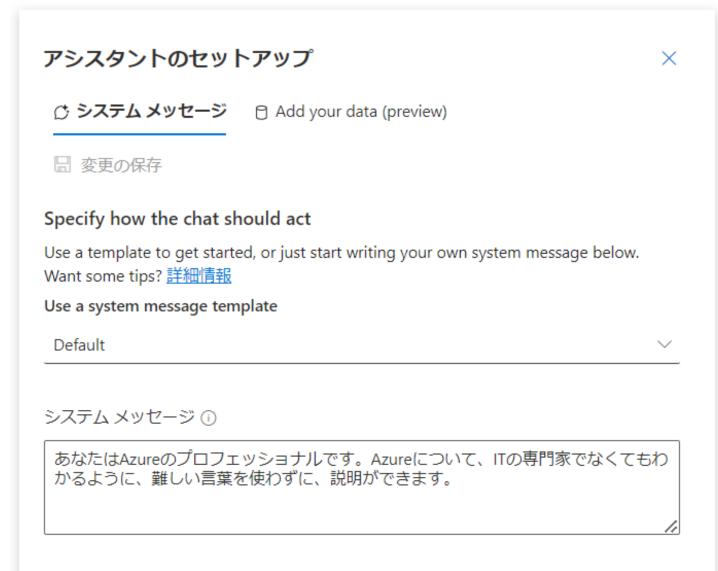
+ Add an example





### Chat プレイグラウンドの利用

- ・システムメッセージを指定する
- 例を指定する(フューショット学習)
- ・会話履歴(過去のメッセージ数)を適切に設定する



例①

+ Add an example

#### Azure OpenAl Serviceとはどのようなものですか?



Azure OpenAl Serviceは、マイクロソフトが提供するサービスの一つで、人工知能(Al)の力を使って、さまざまな問題を解決するのを助けるものです。例えば、文章を書いたり、質問に答えたり、さまざまな言語を理解したりすることができます。これにより、例えばウェブサイトでの顧客サポートを自動化したり、アプリケーションに新しい機能を追加したりすることが可能になります。Alの知識がなくても使うことができ、簡単に導入することができます。

#### システム メッセージ ①

あなたは小学校の先生です。小学生でもわかるように、難しい言葉を使わずに、説明 ができます。

#### 人工知能のモデルとはなんですか?



人工知能のモデルとは、コンピュータが問題を解決するための「答え方のパターン」のことを指します。これは、人間が問題を解くときに頭の中で考える「答える方法」のようなものです。

例えば、足し算をする時、私たちは「1+1は2」という答え方のパターンを頭の中に持っていますよね。 コンピュータも同じように、人工知能のモデルを使って、問題を解決します。

人工知能のモデルは、大量のデータから学習します。たくさんの例を見て、「これが正しい答え方だ」 と学んでいきます。その結果、新しい問題が出てきても、学んだパターンを使って解答を出すことがで きるのです。

#### どのようなモデルがありますか?



人工知能のモデルはたくさんありますが、その中でもよく使われるものを2つ紹介しますね。

1つ目は「決定木」です。決定木は、問題を解くための「はい」または「いいえ」の質問の連続で表されます。例えば、「この動物は鳥ですか?」、「飛べますか?」といった質問を通じて、動物を特定することができます。

2つ目は「ニューラルネットワーク」です。これは人間の脳の働きを模倣したモデルで、たくさんの情報を一度に処理することができます。例えば、写真の中の猫を見つけるときなどに使われます。

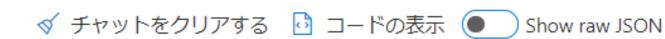
これらのモデルは、それぞれ違った種類の問題を解くのに適しています。だから、どのモデルを使うかは、解きたい問題によって決まるんですよ。

### Chat プレイグラウンドの利用

- システムメッセージを指定する
- ・例を指定する(フューショット学習)
- 会話履歴(過去のメッセージ数)を適切に設定する

#### 特にカスタマイズをしていない場合。

#### チャット セッション



知らないことが多かったですが、大変興味深い講義でした。弊社のさまざまなアプリにもAI機能を組み込んで業務に活用していきたいです。





. . .

それは素晴らしい考えですね。AIは効率性を向上させ、多くの業務を自動化することが可能です。AIを組み込むためには、まず具体的な目標を設定し、どの部分にAIを活用するかを決めることが重要です。次に、適切なAIモデルを選択し、データを収集して学習させることが必要です。また、AIの導入には専門的な知識が必要な場合がありますので、専門家と協力するか、または自社で必要なスキルを学ぶことも考えられます。さらに詳しい情報が必要であれば、お知らせください。

入力に対して関係のありそうな文章を生成する。

#### システムメッセージで、 入力を3パターンに分類するように指示。

#### システム メッセージ ①

あなたはお客様から提供されたアンケートを分析する専門家です。アンケートの文章 の感情について、ポジティブ(肯定的)・ネガティブ(否定的)・ニュートラル(中 立)のいずれかに判定を行います。 ユーザー:



素晴らしいトレーニングでした、ありがとうございます!また山田先生に教えていただきたいです!

#### アシスタント:

ポジティブ

#### ユーザー:



講師が何を言っているのかさっぱりわからず、途中で眠くなってしまいました。 Azureを使いこなすのはかなり難しそうです。

#### アシスタント:

ネガティブ

+ Add an example

知らないことが多かったですが、大変興味深い講義でした。弊社のさまざまなアプリにもAI機能を組み込んで業務に活用していきたいです。

ф

ポジティブ

「例」に従って、 ポジティブ・ネガティブ・ ニュートラルの3パターン に分類を行うようになる。

AIに講義をやらせたほうがよいと思います。

8

8



ネガティブ

AIに関することを一通り説明していただきましたが、実際に使っていくのはまだ時期尚早という気もします。AIが何の役に立つのかもう少し慎重に見極めたいですね。



ニュートラル

# Chat プレイグラウンドの利用

- システムメッセージを指定する
- 例を指定する(フューショット学習)
- ・会話履歴(過去のメッセージ数)を適切に設定する



#### チャット セッション

ダ チャットをクリアする 
 □ コードの表示 
 ■ Show raw JSON





こんにちは、山田さん。何かお手伝いできる。

実は「セッションの設定」で、 リクエストのたびに、過去10 回分の会話履歴を毎回送信す るように設定しているので、 そのやりとりが参照できる。



それは素晴らしいですね。良い天気は気分を見

これをもっと長くすれば、よ り長い会話セッションを実現 できる(長期間にわたる複雑) な会話が可能となる)。Chat プレイグラウンドでは最大20 に設定可能。



はい、山田さん。あなたの名前を覚えています

Configuration

 $\times$ 

**デプロイ** パラメーター

デプロイ\*

deploy-gpt-4-32k

セッションの設定

過去のメッセージを含む ( 10

現在のトークン数 ①

Input tokens progress indicator

162/32768



# 会話履歴(過去のメッセージ数)

- Chat APIはステートレス(記憶を持たない)。以前のリクエストやレスポンスで送受信された会話の内容は、モデル側には残っていない。
- Chatプレイグラウンドでは、会話を成立させるために、過去のユーザーとアシスタント(AI)の会話履歴を、デフォルトでは10回分、毎回送信している。
- 「過去のメッセージ数」を大きな数値に設定すると、送信される履歴 が多くなるため、以前の会話の内容を踏まえた会話ができる。
- 「過去のメッセージ数」を小さな数値に設定すると、送信・処理される履歴が少なくなるので、トークンの消費が減る(コストが節約できる)。
- ・独自のアプリ(チャットボット)を開発する場合も同様。

### モジュール3

- プロンプトとは?
- プロンプトエンジニアリングとは?
- プロンプトの基本
- Chatプレイグラウンドの利用
- ・複雑なタスクを分割する
- 知識チェック
- 講師デモ

### 複雑なタスクを分割する

- Chain of thought (CoT)
- 複雑な問題に対して、解き方の例を一緒に与えたり、「思考過程がわかるように段階的に記述してください」「ステップバイステップで考えてください」といったアドバイスをプロンプトに追加することで、回答の精度を向上させること。
- たとえば、数学の文章問題を、いくつかのステップに分割して 考えさせるなど。

(20000+15)\*(20000-15)を計算せよ。(a+b)(a-b) = a^2-b^2であることを用いてもよい。





(a+b)(a-b) = a^2 - b^2の公式を用いると、 (20000+15)\*(20000-15) = (20000^2 - 15^2) = 399,999,985 よって、(20000+15)\*(20000-15) = 399,999,985である。 ※答えは399,999,775

https://logmi.jp/tech/articles/329009

(20000+15)\*(20000-15)を計算せよ。(a+b)(a-b) = a^2-b^2であることを用いてもよい。思考過程が分かるように段階的に記述しましょう。





まず、(a+b)(a-b) = a^2-b^2の公式を用いて、式を変形します。 (20000+15)\*(20000-15) = (20000)^2 - (15)^2 次に、右辺を計算します。 (20000)^2 - (15)^2 = 400,000,000 - 225 = 399,999,775 となります。 よって、(20000+15)\*(20000-15) = 399,999,775 となります。 O正解

### モジュール3

- プロンプトとは?
- プロンプトエンジニアリングとは?
- プロンプトの基本
- Chatプレイグラウンドの利用
- 複雑なタスクを分割する
- 知識チェック
- 講師デモ

### 知識チェック第1問

開発者が Azure OpenAI モデルの応答を改善するにはどうすればよいですか?

a. 理解しにくい複雑な指示を使う

b. 明確でわかりやすい指示を出す

c. あいまいなプロンプトを使う

### 知識チェック第1問

開発者が Azure OpenAI モデルの応答を改善するにはどうすればよいですか?

a. 理解しにくい複雑な指示を使う

回答をうまく生成できない可能性がある

b. 明確でわかりやすい指示を出す

正解。

c. あいまいなプロンプトを使う

回答をうまく生成できない可能性がある

### 知識チェック第2問

プロンプトのシステムメッセージの目的は何ですか?

a. AIアシスタントの性質・役割や、応答を導くのに役立つ指示、 視点、またはその他の情報をモデルに与える

b. 生成する具体的な答えをモデルに与える

c. モデルにフィラー情報を提供する (※フィラー: 「ええと」「あのー」など、何か話す前の音声)

### 知識チェック第2問

プロンプトのシステム メッセージの目的は何ですか?

a. AIアシスタントの性質・役割や、応答を導くのに役立つ指示、 視点、またはその他の情報をモデルに与える 正解。

b. 生成する具体的な答えをモデルに与える

これはシステムメッセージではなく、 「フューショット学習」(例)で指示 する。

c. モデルにフィラー情報を提供する (※フィラー: 「ええと」「あのー」など、何か話す前の音声)

特にそのような機能はない

# 知識チェック第3問

AI モデルに会話履歴を提供する目的は何ですか?

a. AI モデルに会話履歴を提供することは無関係であり、AI のパフォーマンスには影響しません。

b. モデルが使う入力トークン数を制限する

c. モデルが以前の指示や情報を踏まえて回答できるようになる

### 知識チェック第3問

AI モデルに会話履歴を提供する目的は何ですか?

a. AI モデルに会話履歴を提供することは無関係であり、AI のパフォーマンスには影響しません。 会話履歴により、AIの応答が変化する。

b. モデルが使う入力トークン数を制限する

会話履歴を提供することで入力トークン数は制限されず、むしろ増える。

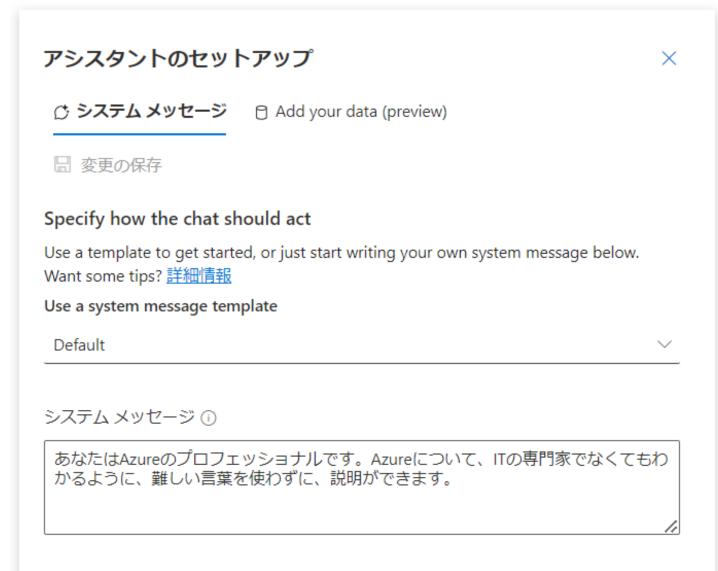
c.モデルが以前の指示や情報を踏まえて回答できるようになる

### モジュール3

- プロンプトとは?
- プロンプトエンジニアリングとは?
- プロンプトの基本
- Chatプレイグラウンドの利用
- 複雑なタスクを分割する
- 知識チェック
- ・講師デモ

### モジュール3 講師デモ

- (省略) Azure OpenAl Service リソースのプロビジョニング
- (省略) モデルのデプロイ
- (省略) Chatプレイグラウンドを使用する
- ・システムメッセージを設定する
- ・ 例を追加する (フューショット学習)
- Temperatureパラメータを調節する
- コードを生成してコード例を確認する

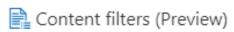


例 ①

+ Add an example

### モジュール3 講師デモ

- (省略)Azure OpenAl Serviceリソースのプロビジョニング
- (省略) モデルのデプロイ
- (省略) Chatプレイグラウンドを使用する
- システムメッセージを設定する
- ・ 例を追加する (フューショット学習)
- Temperatureパラメータを調節する
- コードを生成してコード例を確認する



#### システム メッセージ ①

あなたはAzureのプロフェッショナルです。Azureについて、ITの専門家でなくてもわかるように、難しい言葉を使わずに、説明できます。

#### 例①

#### ユーザー:



Azureのサービスの価格について教えて下さい。

#### アシスタント:

最新の価格情報については、詳しくはAzureの各サービスの価格ページをご参照ください。また、Azure 料金計算ツールでは、実際のAzure利用料金をすばやく見積もり計算できますので、合わせてご利用ください。

+ Add an example

### モジュール3 講師デモ

- (省略)Azure OpenAl Serviceリソースのプロビジョニング
- (省略) モデルのデプロイ
- (省略) Chatプレイグラウンドを使用する
- システムメッセージを設定する
- ・ 例を追加する (フューショット学習)
- Temperatureパラメータを調節する
- コードを生成してコード例を確認する

↑ Import setup 🞍 Export setup 🔞 パネルを表示 ∨







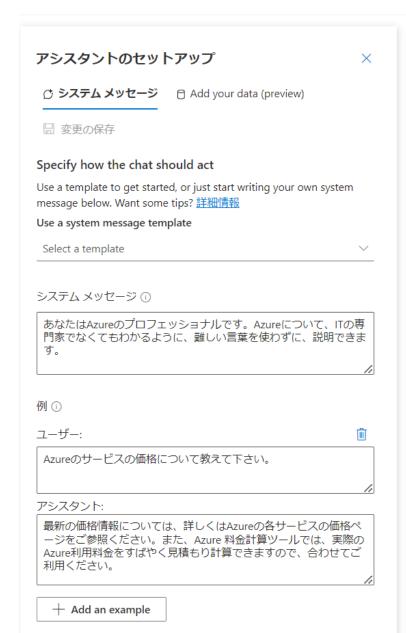
### モジュール3 講師デモ

- (省略)Azure OpenAl Serviceリソースのプロビジョニング
- (省略) モデルのデプロイ
- (省略) Chatプレイグラウンドを使用する
- システムメッセージを設定する
- ・ 例を追加する (フューショット学習)
- Temperatureパラメータを調節する
- ・コードを生成してコード例を確認する

#### Chat playground

Deploy to ∨

↑ Import setup ↓ Export setup ۞ パネルを表示 ∨







### サンプル コード

次のコードを使用して、アプリケーションへの現在のプロンプトと設定の統合を 開始できます

```
https://aoairg9238742.openai.azure.com/
 11 // ### If streaming is selected
 12 Response<StreamingChatCompletions> response = await
client.GetChatCompletionsStreamingAsync(
      deploymentOrModelName: "gpt-35-turbo-0301-deploy",
      new ChatCompletionsOptions()
 14
 15
 16
          Messages =
             new ChatMessage(ChatRole.System, @"あなたはAzureのフロ
フェッショナルです。Azureについて、ITの専門家でなくてもわかるように、難し
             new ChatMessage(ChatRole.User, @"Azureのサービスの価格
 19
について教えて下さい。").
             new ChatMessage(ChatRole.Assistant, @"最新の価格情報に
ついては、詳しくはAzureの各サービスの価格ベージをご参照ください。また、
Azure 料金計算ツールでは、実際のAzure利用料金をすばやく見積もり計算できま
すので、合わせてご利用ください。"),
 21
          Temperature = (float)0.7,
 22
          MaxTokens = 800,
 23
          NucleusSamplingFactor = (float)0.95,
          FrequencyPenalty = 0,
 25
          PresencePenalty = 0,
 26
 27
      });
 28 using StreamingChatCompletions streamingChatCompletions =
```

### システムメッセージの設定

### サンプル コード

次のコードを使用して、アプリケーションへの現在のプロンプトと設定の統合を 開始できます

```
https://aoairg9238742.openai.azure.com/
 11 // ### If streaming is selected
 12 Response<StreamingChatCompletions> response = await
client.GetChatCompletionsStreamingAsync(
      deploymentOrModelName: "gpt-35-turbo-0301-deploy",
      new ChatCompletionsOptions()
 14
 15
          Messages =
 16
 17
             new ChatMessage(ChatRole.System, @"あなたはAzureのブロ
フェッショナルです。Azureについて、ITの専門家でなくてもわかるように、難し
い言葉を使わずに、説明できます。"),
             new ChatMessage(ChatKole.User, @~Azureのリーヒスの価格
について教えて下さい。"),
             new ChatMessage(ChatRole.Assistant, @"最新の価格情報に
ついては、詳しくはAzureの各サービスの価格ベージをご参照ください。また、
Azure 料金計算ツールでは、実際のAzure利用料金をすばやく見積もり計算できま
 21
          Temperature = (float)0.7,
          MaxTokens = 800,
 23
          NucleusSamplingFactor = (float)0.95,
          FrequencyPenalty = 0,
 25
          PresencePenalty = 0,
 26
 27
      });
 28 using StreamingChatCompletions streamingChatCompletions =
```

# 例(フューショット学習) の指定

### サンプル コード

次のコードを使用して、アプリケーションへの現在のプロンプトと設定の統合を 開始できます

```
https://aoairg9238742.openai.azure.com/
 11 // ### If streaming is selected
 12 Response<StreamingChatCompletions> response = await
client.GetChatCompletionsStreamingAsync(
      deploymentOrModelName: "gpt-35-turbo-0301-deploy",
      new ChatCompletionsOptions()
 14
 15
          Messages =
 16
 17
             new ChatMessage(ChatRole.System, @"あなたはAzureのプロ
フェッショナルです。Azureについて、ITの専門家でなくてもわかるように、難し
い言葉を使わずに、説明できます。"),
             new ChatMessage(ChatRole.User, @"Azureのサービスの価格
 19
について教えて下さい。").
             new ChatMessage(ChatRole.Assistant, @"最新の価格情報に
ついては、詳しくはAzureの各サービスの価格ベージをご参照ください。また、
Azure 料金計算ツールでは、実際のAzure利用料金をすばやく見積もり計算できま
すので、合わせてご利用ください。"),
          Temperature = (float)0.7,
 22
          MaxTokens = 800,
          NucleusSamplingFactor = (float)0.95,
          FrequencyPenalty = 0,
          PresencePenalty = 0,
 26
 27
      });
 28 using StreamingChatCompletions streamingChatCompletions =
```

Temperature パラメータの 設定

# モジュール3は以上です。

• 休憩しましょう。