

AI-050

補足資料：Module03

- プロンプト エンジニアリングについて理解する
- より効果的なプロンプトを作成する
- 精度を高めるようにコンテキストを提供する

- 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>

- **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

- <https://chatgpt-lab.com/n/nd8dde92016ec>

- 「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>

- 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。

- プロンプトエンジニアリングよりも前に、さまざまなエンドポイントからどのように利用できるかを検討することが重要
- **Completion (gpt3以前)** と **ChatCompletion (gpt-35-turbo以降)** はどちらも同様の結果を得られるが、プロンプトを構築する上で **ChatCompletion** は最も柔軟性があり、チャット シナリオ向けに最適化されている
- 機能面では、**ChatCompletion** はプロンプト内の以前のメッセージを提供するための組み込みの構造に加え、AI モデルのシステム メッセージを定義するオプションがある。**Completion** を使う場合、この機能はいわゆる “メタ プロンプト” を使って実現できる
- **ChatCompletion** はチャット以外のシナリオでも使用できる

- モデルのパラメーターを調整することも、応答に大きな影響を与える可能性がある
- 特に **temperature** と **top_p (top_probability)** は、いずれもモデルのランダム性を制御するので、モデルの応答に影響を与える可能性が最も高い
- 同じプロンプトを使ってこれらのパラメーターを調整し、応答にどのような影響が出るかを試す
- **temperature** または **top_p** を変更する際は、一度に両方を変更しない

- 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) 9

- 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す英単語は、` でくくりにし、その後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 の一般的な名詞の例として

基礎コンテンツ (前提知識)

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。

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.

セクションマーカ
(区切り線)

====

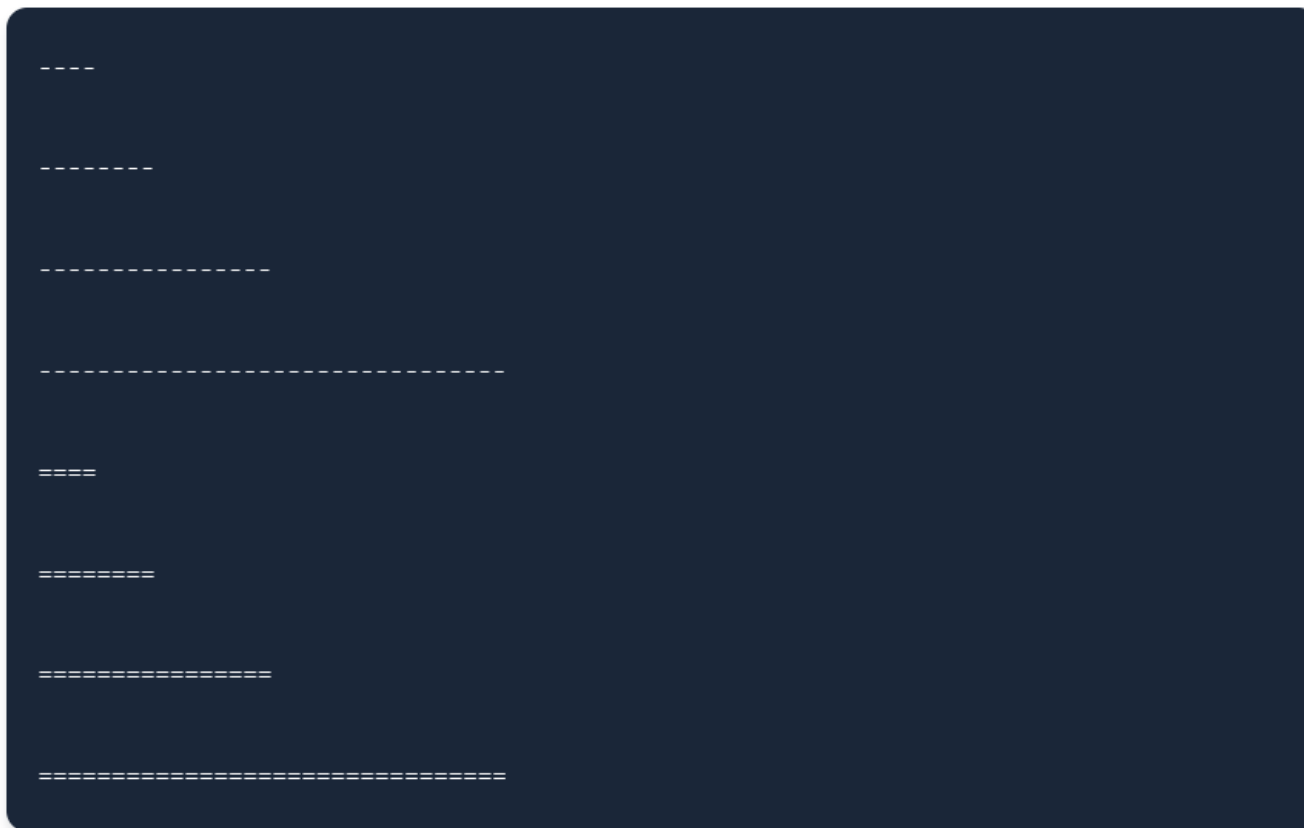
この文章を日本語に翻訳してください。
文章の中の例として示される英単語は ` でくくり、その前後にスペースを入れてください。ただし、一般的な名詞は、 ` でくくらないでください。

====

たとえば Azure OpenAI は、一般的な名詞の例です。

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。

<https://zenn.dev/zawawahoge/articles/4519112f1ad5ed>

- 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>

キュー

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;  
}
```


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・ヴァンロッサム
実行環境	.NET Framework	Pythonインタプリタ
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アプリケーションやビジネスアプリケーション	Webアプリケーションやデータサイエンス、機械学習
構文の厳密さ	厳密	柔軟

- 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

The screenshot displays the Azure AI Studio Chat playground interface. On the left is a sidebar with navigation links: Azure OpenAI, プレイグラウンド (Playground), チャット (Chat), 入力候補 (Input suggestions), DALL-E (Preview), 管理 (Manage), デプロイ (Deploy), モデル (Model), and Data files. The main area is titled 'Chat playground' and includes a 'Deploy to' button and links for 'Import setup', 'Export setup', and 'パネルを表示' (Show panel). Three panels are visible: 'アシスタントのセットアップ' (Assistant setup), 'チャットセッション' (Chat session), and 'Configuration'. The 'アシスタントのセットアップ' panel has a 'システムメッセージ' (System message) section with a text area containing 'You are an AI assistant that helps people find information.' and an '例' (Example) section with a user prompt and an assistant response. The 'チャットセッション' panel shows a '会話履歴' (Conversation history) section with a blue callout box. The 'Configuration' panel includes a 'デプロイ' (Deploy) dropdown set to 'deploy-gpt-4-32k', a 'セッションの設定' (Session settings) section with a '過去のメッセージを含む' (Include previous messages) slider set to 10, and a '現在のトークン数' (Current token count) indicator showing 0/32768. Blue callout boxes highlight the 'システムメッセージ' and the '例 (フューショット学習)' (Example (Few-shot learning)) sections in the assistant setup panel, and the '会話履歴' (Conversation history) section in the chat session panel.

システムメッセージ


例 (フューショット学習)

会話履歴

アシスタントのセットアップ



 システム メッセージ  Add your data (preview)

 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? [詳細情報](#)

Use a system message template

Default




システム メッセージ ⓘ

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、説明ができます。



例 ⓘ

 Add an example

Azure OpenAI Serviceとはどのようなものですか？



Azure OpenAI 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（AI）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。AI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 ⓘ

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。

- モデルに対して、こちらが意図する回答を促す目的で、あらかじめ例（サンプル）を列挙する手法
 - さまざまな目的で利用可能
 - インテント分類
 - 論理思考の誘導
 - **ReAct**に沿った回答誘導
- 言語モデルによる思考(Reasoning) に「検索」などの外部ツールの処理(Acting) を柔軟に組み合わせる方法
 - few-shot プロンプト(例示) などで言語モデルに「行動」(act) を推薦させ、外部ツールで実際にその行動をおこなって結果を追加し、また行動を推薦させる、というループを実装
 - 回答精度をあげる目的以外に、言語モデルのみで不可能な他操作との連携一般でも使用可能
 - 開発にはLangChain などライブラリを使用可能
 - より高度な方式では、強化学習、模倣学習なども使用

チャット セッション

特にカスタマイズをしていない場合。

✓ チャットをクリアする  コードの表示 ☐ Show raw JSON

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。

- 今回は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した

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。

例 ⓘ

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

「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。

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。



...

ポジティブ

AIに講義をやらせたほうがよいと思います。



...

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もう少し慎重に見極めたいですね。



...

ニュートラル

- **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。

チャットセッション

☒ 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ことがありますか？



私の名前は山田です。



それは素晴らしいですね。良い天気は気分を良くします。何かお手伝いできることがありますか？



今日はいい天気です。



はい、山田さん。あなたの名前を覚えています。どのようにお手伝いできるかお知らせください。



ところで、私の名前を覚えていますか？

Configuration

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 ① 10

現在のトークン数 ①

Input tokens progress indicator

162/32768

AIが、ここまでの会話の内容を記憶しているように応答する。

チャットセッション

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。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。



それは素晴らしいですね。良い天気は気分を良くします。何かお手伝いできることがありますか？

ところで、私の名前を覚えていますか？



はい、山田さん。あなたの名前を覚えています。どのようにお手伝いできるかお知らせください。

Configuration



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 ① 10

現在のトークン数 ①

Input tokens progress indicator

162/32768

チャットセッション

🗑️ チャットをクリアする 📄 コードの表示 ☒ Show raw JSON

The screenshot shows a chat interface with the following messages:

- User: こんにちは、私の名前は山田です。
- AI: こんにちは、山田さん。どのようにお手伝いできるかお知らせください。
- User: 今日はいい天気ですね
- AI: 私はAIアシスタントなので、天候を直接感じることはできませんが、あなたの地域の天気予報を提供できます。どの地域の天気を知りたいですか？
- User: ところで、私の名前は？
- AI: 申し訳ありませんが、私はAIアシスタントなので、個々のユーザーの名前を知る能力はありません。ご自身でお名前を覚えていただけますか？

Configuration



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

1



現在のトークン数

この数字を小さくすると...

以前の会話の内容を踏まえることができなくなる。

- **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) = (20000² - 15²) = 399,999,985
よって、(20000+15)*(20000-15) = 399,999,985である。

✗ 答えは399,999,775

(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よう。



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。

$$(20000)^2 - (15)^2 = 400,000,000 - 225$$

$$= 399,999,775$$

よって、 $(20000+15)*(20000-15) = 399,999,775$ となります。

○正解

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

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

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」「あの一」など、何か話す前の音声)

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」「あの一」など、何か話す前の音声）

特にそのような機能はない

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

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

正解。