



OpenAI GPT-4 ChatGPT LangChain 人口知能プログラミング実践入門

PDF版ダウンロードリンク

<https://www.borndigital.co.jp/book/30237.html>

Python3.10

LlamaIndex0.6.12

LangChain0.0.181

ChatGPT、OpenAI Playground、DALL-E、GitHub Copilotの利用法

各種モデル別の機能や応用キーワード

ChatGPT

大規模人口知能自然言語処理モデル

- テキスト・文章生成
- 質問応答
- 要約
- 翻訳
- プログラム生成

OpenAI Playground

複数のモデルから利用環境に応じた適切なモデルを選択し、各種オプションにより生成AIの制御が行える、「挿入モード」「編集モード」といったテキスト生成機能もついている

- テキスト生成
- 質問応答
- 要約
- 翻訳
- プログラム生成

DALL-E

DALLEはOpenAIが開発したGPT-3を自然言語から画像生成するよう学習させたAIモデルである、文章の意味を理解し、もっとらしい画像を生成してくれる画像生成Webユーザ

—インターフェース

- テキストからの画像生成
- バリエーション
- インペインティング
- アウトペインティング

GitHub Copilot

OpenAIとGitHubが共同開発した人口知能によるプログラミング開発支援ツール

- コメントからプログラムコード生成
- コードの自動補完支援

対応プログラミング言語：

Python

JavaScript

C++

C#

Ruby

Go

TypeScript

ChatGPTの利用法

大規模人口知能自然言語処理モデル

- テキスト・文章生成
- 質問応答
- 要約
- 翻訳
- プログラム生成

ChatGPTのアカウント登録し方

メールアドレス、携帯電話番号、PCやインターネットがあれば利用可能

登録の流れ：

- 1.OpenAIのホームページへアクセス <https://openai.com/>
- 2.登録画面 SignUpをクリック
- 3.メールアドレスを登録する
- 4.パスワードを入力する
- 5.Continueをクリックしてメールアドレスを認証する
- 6.携帯電話番号を登録する
- 7.登録された携帯電話にコードが送られる
- 8.コードを入力
- 9.Next→Next→Doneをクリックしたら登録完了（ログイン画面が表示される）

登録後のログイン流れ：

- 1.ログイン画面 <https://platform.openai.com/apps>
- 2.ChatGPTを選択
- 3.メールアドレスやPW入力したらログイン完了

ChatGPTの画面構成

必要に応じて各機能ボタンをクリックするだけで簡単に操作ができる

入力欄について

Send a message... の欄にテキスト（日本語/英語/など）入力した言語でリアルタイムに対話形式で回答が得られる

新規質問について

話題を変えて新しい質問をしたいときは画面左上のNew chatのボタンを押せば新規の質問や回答のやり取りができる

その他操作機能について

- 1.Clear Conversations:チャット履歴を全消去
- 2.Upgrade to Plus:GPT-3.5からGPT-4.0にアップグレードする（有料版）
- 3.Dark Mode:ダークモードとライトモードを切り替える
- 4.Get Help:ヘルプ

5.Log Out:ログアウト

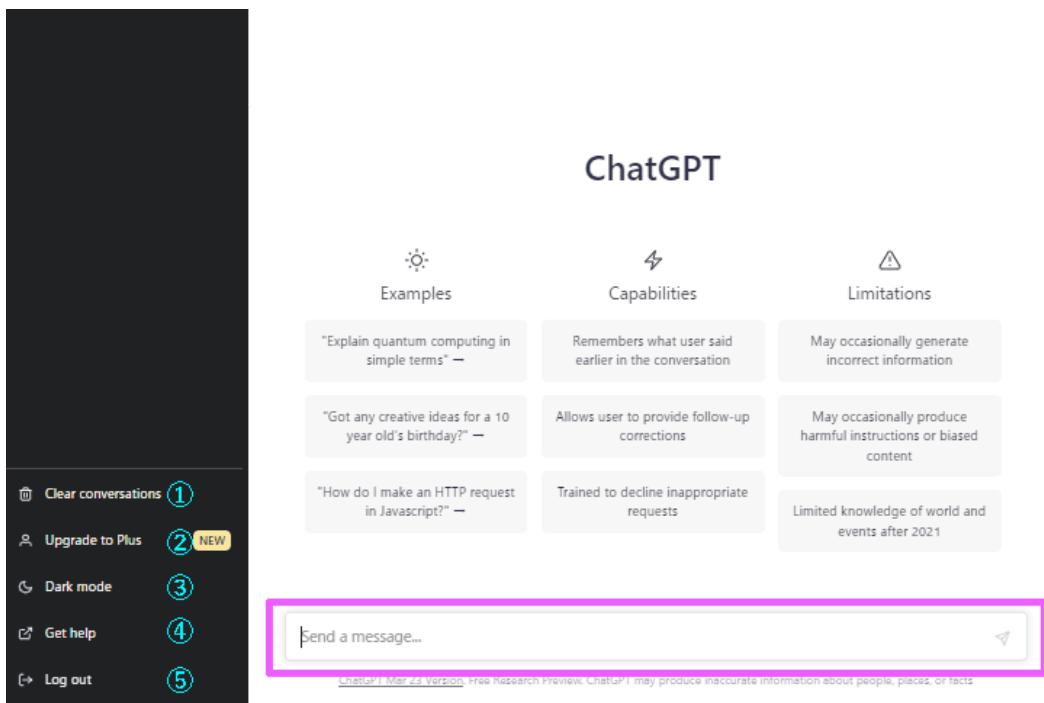


図2.1

出所: Socratetsu.jpより. 公式Webサイト |chatGPTウェブ版画面

チャット履歴名（タイトル名）の編集について

1.表示されているチャット履歴の件名・タイトル名編集は画面左上の検索履歴キーワード表示がでている箇所にマウスを当てる

2.表示されたマークを下記の順で操作すれば編集ができる

3.鉛筆マーク→タイトル名の変更→✓マーク→編集完了

チャット履歴一部のみ削除について

1.表示されているチャットの履歴は画面左上の検索履歴キーワード表示がでている箇所にマウスを当てる

2.表示されたマークを下記の順で操作すれば編集ができる

3.ゴミ箱マーク→✓マーク→削除完了

注意：

表示されているチャット履歴の件名・タイトル名編集は画面左上の検索履歴キーワード表示（※1）がでている箇所にマウスを当てると鉛筆マーク、ゴミ箱マークや×マークが現れる（※2）

（※1）検索内容のキーワードが履歴件名として検索順番で左上のNew chatのボタンの下のエリアの表示される

(※2) 新規画面では画面左上の検索履歴キーワードがないためこれらのマークは表示されない

鉛筆マーク：タイトル名の変更

ゴミ箱マーク：選択された欄のチャット履歴のみが削除される

✓マーク：確定

✗マーク：キャンセル

各種機能の使い方

テキスト生成の仕方

メッセージ欄に「テキストを入力」し、エンターを押すと入力した言語でリアルタイムに対話の結果が表示される

実践テーマの例：

- 物語の生成
- 歌詞の生成
- テキストベースゲームの生成
- その他

実践例：物語の生成について

メッセージ欄に「物語を書いてください。」と入力し、エンターを押すと物語のタイトルが決まってないためChatGPTがかてにテーマを選んで表示してくれる、この例では「星の舞踏会」が結果として表示される

結果の文字数：1407字

入力テキスト例：

物語を書いてください。

出力テキスト例：

もちろん、物語を書きます！以下は、架空の物語です。

タイトル：星の舞踏会

昔々、遙か彼方の宇宙には、美しい星々が輝く星の舞踏会という神秘的なイベントがありました。この舞踏会では、全宇宙の星が一堂に会し、光り輝くドレスや風変わりな衣装をまとった星々が美しい舞を踊るのです。

ある夜、地球の星であるリリーは、星の舞踏会への招待状を手に入れました。リリーは興奮して喜び、家族や友人たちにその嬉しい知らせを伝えました。しかし、リリーは自分には特別なドレスがなく、舞踏会に出席できるかどうか心配しました。

そんなリリーの心配を察した親友のコスモスは、星の精霊たちに助けを求めるにしました。彼らは星のエネルギーで美しいドレスを作り上げ、リリーの前に贈りました。リリーはそのドレスの美しさに目を奪われ、涙を流しながら感謝の言葉を述べました。

舞踏会の夜、リリーは美しいドレスを身にまとい、コスモスと共に星の舞踏会の場に姿を現しました。そこには宇宙の星々が輝き、幻想的な光が舞い踊る壮大な光景が広がっていました。

リリーは自分自身を忘れ、音楽に身を委ねて踊り始めました。彼女の踊りは美しく、宇宙の星々もリリーの踊りに感動しました。一晩中、彼女は星々と共に美しい舞を繰り広げ、星の舞踏会は大いに盛り上りました。

舞踏会が終わり、リリーはコスモスに感謝の言葉を述べました。「あなたがいなければ、私はこんな素晴らしい経験をすることはませんでした。本当にありがとうございました」とリリーは言いました。

その後、リリーは地球に帰る途中、星の精霊たちから美しいペンダントを贈られました。それは星の舞踏会での思い出を刻み込んだ特別な贈り物でした。

以来、リリーはその美しいペンダントを大切にし、星の舞踏会の素晴らしい思い出を胸に秘めていました。そして、彼女はいつでも星の舞踏会のような輝かしい夢を追いかけることを決めたのです。

こうして、星の舞踏会は物語の終わりではなく、新たな始まりとなりました。星々が輝く宇宙で、誰もが夢と希望を追い求める冒険が続いていくのでした。

質問応答の仕方

メッセージ欄に「質問を入力」し、エンターを押すと入力した言語でリアルタイムに対話の結果が表示される

質問テーマの例：

- 人口知能について教えてください。
- ChatGPTの発明者は誰ですか？
- 料理レシピ
- その他

実践例：料理レシピ

メッセージ欄に「おせち料理のレシピを書いてください。」と入力し、エンターを押すとレシピが結果として表示される

入力テキスト例：

おせち料理のレシピを書いてください。

出力テキスト例：

おせち料理は、日本のお正月に特別な料理セットとして楽しまれる伝統的な料理です。具材や味付けは地域や家庭によって異なりますが、以下は一般的なおせち料理の代表的な具材とレシピの例です：

黒豆(くろまめ)：

甘く煮た黒豆は、一年の幸運を招くとされています。

材料：

黒豆、砂糖、醤油、水

作り方：

黒豆を洗って水に浸し、煮る前に一晩置きます。煮る時に砂糖と醤油を加えて煮込み、水分がなくなるまで煮詰めます。

伊達巻(だてまき)：卵焼きを巻いて円筒状にし、きれいな模様をつけた料理です。

材料：卵、砂糖、醤油、だし、にんじん(細切り)

作り方：卵に砂糖と醤油、だしを加えて混ぜ、薄焼き卵を作ります。それをにんじんを巻き込みながら巻いて円筒状にします。

かまぼこ：魚のすり身を成型して焼いたり蒸したりした料理です。

材料：かまぼこ用の魚のすり身、片栗粉、酒、砂糖、塩、顆粒だし、食紅(赤色)

作り方：魚のすり身に調味料を加え、泡立て器でよく混ぜます。その後、蒸し器で蒸したり、オーブンで焼いたりして形を整えます。

たたきごぼう：ごぼうを細かく切って煮物にした料理です。

材料：ごぼう、だし、砂糖、醤油、酒、みりん

作り方：ごぼうを細切りにし、煮物用の調味料で煮込んで味を染み込ませます。

田作り(たづくり)：魚のすり身を酒蒸しにした料理です。

材料：魚のすり身、酒、だし、砂糖、塩、卵黄、おろし生姜

作り方：魚のすり身に調味料と卵黄を加えてよく混ぜ、蒸し器で蒸します。完成したらおろし生姜を添えて食べます。

これらは一部の代表的なおせち料理の例ですが、おせち料理には他にもさまざまな具材や料理が含まれます。おせち料理はお正月を迎える特別な料理セットであり、家族や友人と一緒に楽しむことで、新しい年の幸せを願う日本の伝統的な習慣です。

要約の仕方

メッセージ欄に「質問を入力」し、エンターを押すと入力した言語でリアルタイムに対話の結果が表示される

- 人口知能について教えてください。
- 誰でも分かるように短く簡潔にしてください。
- もっと短くお願いします。

実践例：人口知能について

メッセージ欄に「人口知能について教えてください。」と入力し、エンターを押すと1616文字の結果が表示される、さらに短くしてと要求すると176文字の回答が得られる、繰り返して最もまとめたテキスト40文字の要約ができる

結果の文字数：1616字（入力1回目） 結果の文字数：176字（入力2回目） 結果の文字数：40字（入力3回目）

注意：文字数は質問によって変わる

入力1回目のテキスト例：

人口知能について教えてください。

出力テキスト例：

人工知能(Artificial Intelligence, AI)は、コンピューターや機械に人間の知的な活動を模倣させる技術やシステムの総称です。AIは、データを学習し、パターンを認識し、問題を解決する能力を持ちます。AIは、さまざまなタスクを実行するためのアルゴリズムやモデルを使用して、人間に似た知的な行動を行うことが可能です。

AIは大きく以下のようなタイプに分類されます：

弱いAI(ナローAI)：特定のタスクに対して限られた範囲の知能を持つAIです。例えば、チェスをプレイするAIや音声認識システムなどがあります。弱いAIは特定のタスクにおいては高い精度を持つことができますが、広範な知識や理解力を持ちません。

強いAI(ジェネラルAI)：幅広い領域で人間のような知的な能力を持つAIです。つまり、あらゆるタスクにおいて人間と同等の知能を持ち、問題を解決できる理想的なAIを指します。しかし、現在の技術では強いAIの実現は難しいとされています。

AIの主な技術としては以下のものがあります：

機械学習：データを解析し、パターンを学習する技術です。教師あり学習、教師なし学習、強化学習などがあります。

ニューラルネットワーク：脳の神経細胞を模倣したモデルで、画像認識、自然言語処理などに広く利用されています。

自然言語処理(NLP)：人間の自然な言語をコンピュータが理解し、処理する技術です。機械翻訳、テキスト分析、チャットボットなどに使われます。

コンピュータビジョン：画像や動画を解析し、物体やパターンを認識する技術です。顔認識、物体検知などが代表的な応用です。

AIは現代の社会に広範囲に応用されており、自動運転車、推薦システム、医療診断、金融取引、製造業など、さまざまな分野で利用されています。AIの進化と応用は

今後も進んでいくことが予想されており、社会に大きな影響を与えるテクノロジーとして注目されています。

入力2回目のテキスト例：

誰でも分かるように短く簡潔にしてください。

出力テキスト例：

AI(人工知能)は、人間の知的な活動を模倣する技術で、機械学習やニューラルネットワークを利用してデータを学習します。自然言語処理やコンピュータビジョンなど、多くの分野で応用されており、自動運転車や医療診断など、社会に大きな影響を与えるテクノロジーです。

入力3回目のテキスト例：

もっと短くお願いします。

出力テキスト例：

AI(人工知能)は人間の知的な活動を模倣する技術で、多くの分野で利用される。

翻訳の仕方

メッセージ欄に「翻訳したい言葉または文章」を入力し、英訳をお願いします。と指示を入れてエンターを押すと英語でリアルタイムに翻訳の結果が表示される

実践例：

メッセージ欄に「翻訳したい言葉または文章」と入力し、英訳をお願いします。と指示を入れてエンターを押すと英訳版が表示される

入力テキスト例：

人工知能(Artificial Intelligence, AI)は、コンピューターや機械に人間の知的な活動を模倣させる技術やシステムの総称です。AIは、データを学習し、パターンを認識し、問題を解決する能力を持ちます。AIは、さまざまなタスクを実行するためのアルゴリズムやモデルを使用して、人間に似た知的な行動を行うことが可能です。この文章を英訳してください。

出力テキスト例：

Artificial Intelligence (AI) is a collective term for technologies and systems that mimic human intellectual

activities in computers and machines. AI has the ability to learn from data, recognize patterns, and solve problems. It can perform various tasks using algorithms and models, enabling human-like intelligent behavior.

プログラム生成

よく使われる簡単なコードまたは実行してエラーが出た場合の指摘に関する書き直しなどについてはChatGPTに頼める

メッセージ欄に「依頼内容を入力」し、エンターを押すとリアルタイムにプログラミングコードが表示される

注意：2021年までの情報なので以後に更新されたコードについては回答が得られない

実践例：

メッセージ欄に「ファイルの書き込み関数を書いてください」と入力し、エンターを押すとコードが表示される

入力テキスト例：

ファイルの書き込み関数を書いてください

出力テキスト例：

```
def write_to_file(file_path, content):
    try:
        with open(file_path, 'w', encoding='utf-8') as file:
            file.write(content)
            print("ファイルの書き込みが完了しました。")
    except Exception as e:
        print("ファイルの書き込みエラー：" , e)

# 使用例
file_path = 'example.txt'
content = 'これはファイルに書き込まれる内容です。'
write_to_file(file_path, content)
```

OpenAI Playgroundの利用法

利用料金・トークン数について

利用料金は入出力領域のテキストのトークン数で計算される、図の赤枠の数字はトークン数を表している、

OpenAI APIのサイト内で利用できる、無料枠（3か月以内の18ドル分）で試用できる

各利用条件によって利用料金が変わる、高度な設定（プロレベル）についてはさらに詳しい利用料金がある

用途別の料金参考例

1トークン:LLM処理のテキストの単位

英語の場合、1単語が1～2トークン(例:Artificial)

日本語の場合、1文字が1～2トークン(例:あ)

表：1Kトークン当たりの料金

Mode	1K当たりの料金 (USD)	(JPY)
Input	0.0015	0.21
Output	0.002	0.28
Fine-tuning	0.0006	0.08
Embeddings	0.0001	0.01

為替レート:2023年7月27日時点で1 USD = 140.33 JPYで計算された数字

単語説明

Input/Output:入出力

Fine-tuning :ファインチューニング

Embedding :埋め込み生成

注意：

利用料金は異なる場合があるので各自のアカウント/アカウントアイコン/Manage account/Usageの順でボタンを押して確認する必要がある

お支払方法登録が必要で、利用制限設定をすることで利用の予算管理もできる

左メニューのBilling/Usage Limitで指定できる

Soft Limit:

月利用料金設置の限度が到達したときに通知をする限度設定プラン

Hard Limit :

月利用料金設置の限度が到達したときにリクエストを拒否するする設定プラン

OpenAI Playgroundのアカウント登録し方

OpenAI APIのサイト内で利用できる、無料枠（3か月以内の18ドル分）で試用できる

メールアドレス、携帯電話番号、PCやインターネットがあれば利用可能

登録の流れ：

1.OpenAI APIのホームページへアクセス <https://openai.com/blog/openai-api>

2.登録画面 SignUpをクリック

3.ChatGPT登録済の場合はログインをし、上端の「Play Ground」をクリック

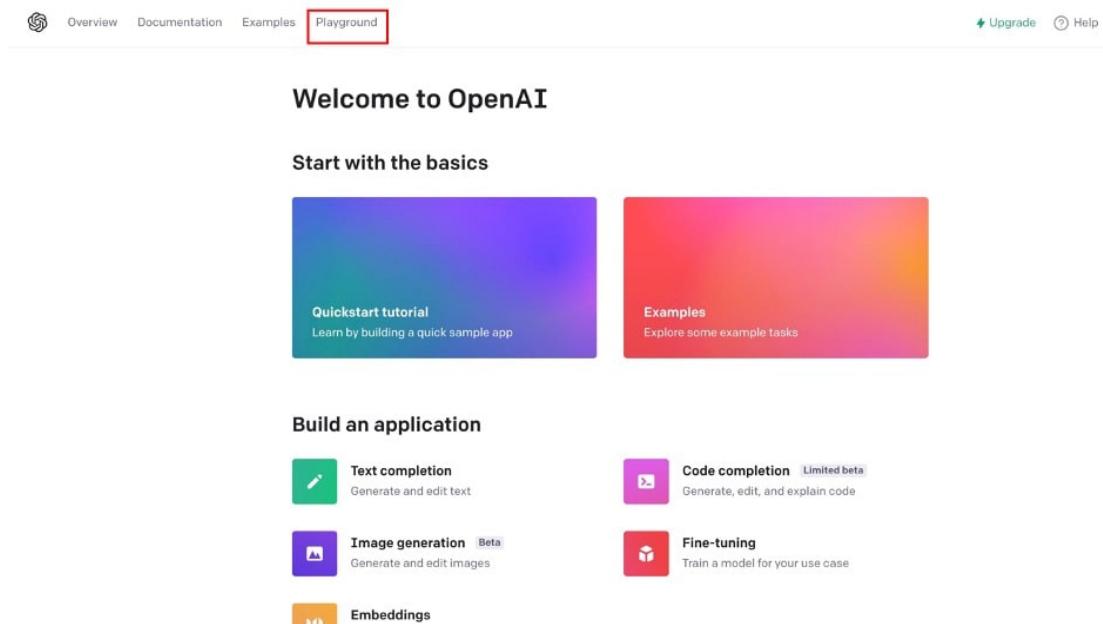


図2.2

出所: AI Life Hacksより. 公式Webサイト |Playground アクセス方法

4.OpenAI Playgroundの画面が表示される

Open AI Playgroundの画面構成

各種パラメーター設定について

Playground画面の右側で「どのAIモデルを使用するか」や「モデルのパラメータ」などの設定ができる

料金かからない設定あるいは品質の高い設定などさまざまな設定が可能である

利用前に目的にあったパラメーターについての理解が必須である

各種ボタン説明

Submit : テキスト生成依頼

Undo Last :直前操作の取り消し

Regenerate :テキスト再生成

Show history :履歴表示

パラメーター：図の右側がパラメーター設定のエリア

OpenAI Playgroundの画面

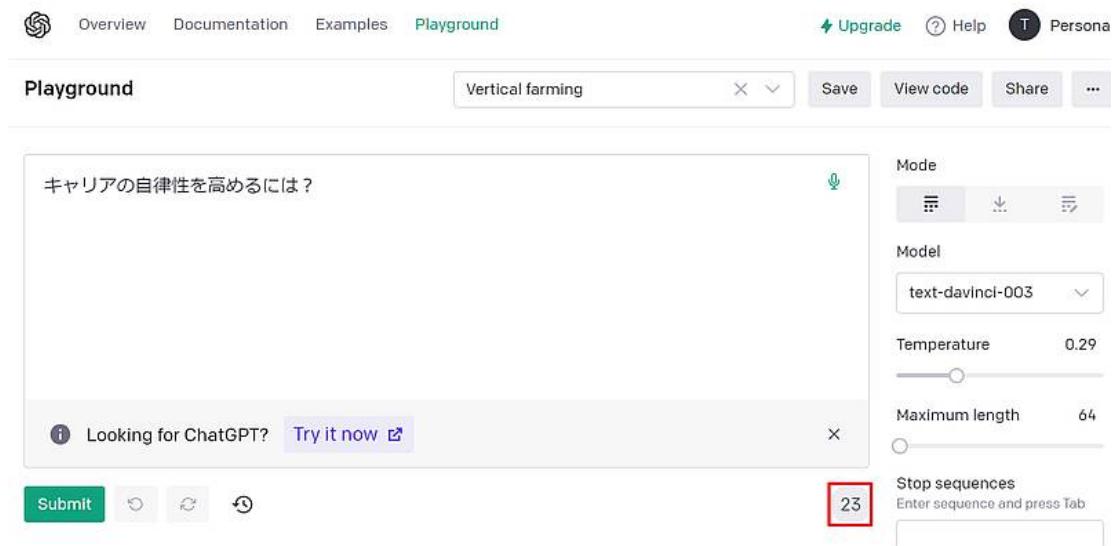


図2.3

出所: AutoWorkerより. 公式Webサイト |Open AI Playground サイト メインページ

設定モード

Complete :テキスト生成モード

Chat :チャットモード

Insert :挿入モード

Edit :編集モード

モデル選択基準表

モード	モデル	ベースモデル
Complete	text-davinci-003	GPT-3.5
	text-curie-001	GPT-3
	text-babbage-001	GPT-3
	text-davinci-003	GPT-3
Chat	gpt-4	GPT-4
	gpt-3.5-turbo	GPT-3.5
Insert	text-davinci-003	GPT-3.5
Edit	text-davinci-edit-001	GPT-3
	codedavinci-edit-001	GPT-3

パラメーターの決め方基準

パラメーター	説明
Temperature	ランダムさ、創造的にするには0.9、答えがある場合は0を推進。 Top Pと同時変更は非推進（デフォルト：0.7、0~1）
Maximum length	最大トークン数（デフォルト：256）
Stop sequence	停止文字列（デフォルト：なし）
Top P	nucleusサンプリング。0.1は、上位10%の確率を持つトークンのみの考慮を意味する Temperatureと同時変更は非推進（デフォルト：1）
Frequency penalty	生成したテキストに基づいて、新しいトークンに課すペナルティ。 同じテキスト出力の可能性を減らす（デフォルト：0,0~2）
Presence penalty	生成したテキストに基づいて、新しいトークンに課すペナルティ。 同じトークンの繰り返しの可能性を減らす（デフォルト：0,0~2）
Best of	サーバー側でbest_of個の結果を生成し、最良を返す（デフォルト：1）
Inject start text	書式補正のためコンプリーション前に追加するテキスト
Inject restart text	書式補正のためコンプリーション語に追加するテキスト
Show probabilities	トークンの生成される確率を色分けで表現 Off:オフ Most likely: 最も確率が高いものを色分け Least likely: 最も確率が低いものを色分け Full spectrum: 両方を2色で色分け

入力欄について

必要に応じて各機能ボタンをクリックするだけで簡単に操作ができる

画面の真ん中にある入力バーに質問を記入し、“Submit”ボタンを押すことで、AIツールが回答してくれる

入出力エリアの例

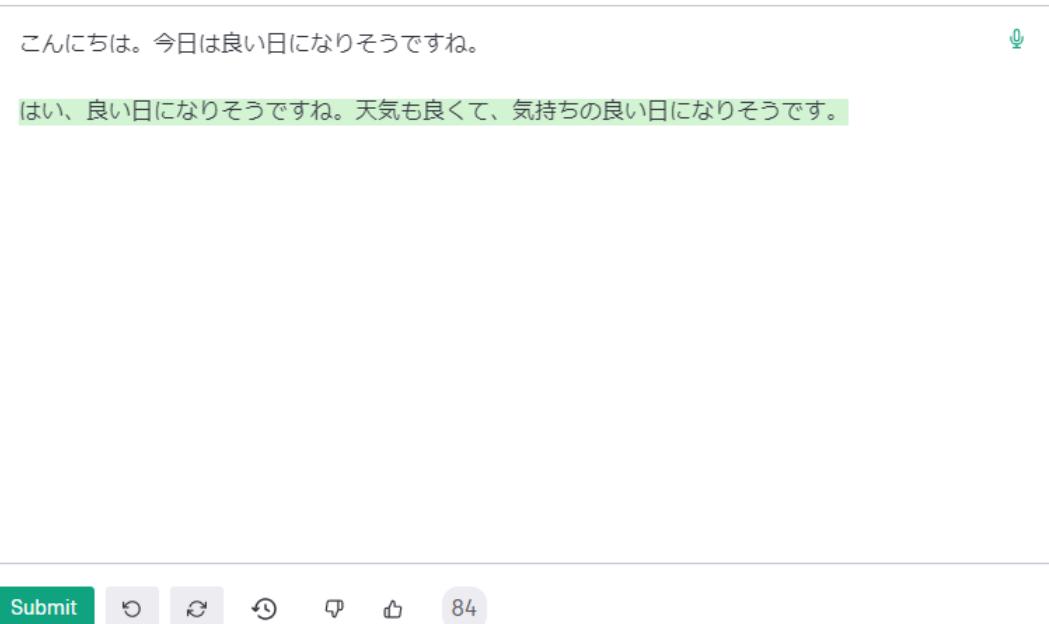


図2.4

出所: ClassMethod.Incより. 公式Webサイト |Open AI Playground入力画面

テキスト生成モード

テキスト生成の場合は完全モードCompleteを選んでモデルはモデル選択基準通りに選ぶこと

文章生成

テーブル：パラメーターの例

モード	モデル
Complete	text-davinci-003
Temperatur	0.7
Maximum length	256
TopP	1

実践例：物語の生成について

テキスト生成の依頼内容を入力：

物語を書いてください
コミュ障でぼっちな学生があることをきっかけにロックバンドに加入し、慣れない人間関係を通じて活動していく物語

出力：

高校生の「○○さん」は音楽が大好きだった。毎日楽器の練習を通じて時間を過ごしていた。しかし、かれはコミュ障でぼっちなので、友達も少なかった

質問応答

実践例：知りたい内容を質問してテキスト生成について

質問の依頼内容を入力：

人口知能について教えてください

出力：

人工知能(Artificial Intelligence, AI)は、コンピューターや機械に人間の知的な活動を模倣させる技術やシステムの総称です。AIは、データを学習し、パターンを認識し、問題を解決する能力を持ちます。AIは、さまざまなタスクを実行するためのアルゴリズムやモデルを使用して、人間に似た知的な行動を行うことが可能です。

要約

実践例：要約したいテキストの生成について

質問の依頼内容を入力：

以下の文章を1分で要約してください。

人工知能(Artificial Intelligence, AI)は、コンピューターや機械に人間の知的な活動を模倣させる技術やシステムの総称です。AIは、データを学習し、パターンを認識し、問題を解決する能力を持ちます。AIは、さまざまなタスクを実行するためのアルゴリズムやモデルを使用して、人間に似た知的な行動を行うことが可能です。

出力：

AI(人工知能)は人間の知的な活動を模倣する技術で、多くの分野で利用される。

プログラム生成

実践例：プログラミングコードの生成について

プログラミング生成の依頼内容を入力：

```
#"Hello World!"の表示コード  
def helloworld():
```

出力：

```
print("Hello World!")
helloworld( )
```

チャットモード

チャットモードはテキスト生成モードより利用料金が安いため、精度の要求がなければチャットモードをテキスト生成の代わりに使える

ただし、チャットモードの実践手順は異なる

設定の流れ

チャートの場合はチャートモードChatを選んでモデルはモデル選択基準通りに選ぶこと

チャートモードでは入出力領域の場所が、1つの「SYSTEM」と複数の「USER」「ASSISTANT」にユーザーとチャートAIのメッセージのやり取りを入力する

「Add message」ボタンで、「USER」「ASSISTANT」を追加できる

単語の説明

「SYSTEM」：システムに設定されたチャートAIのキャラクター紹介の欄

「USER」：入力欄

「ASSISTANT」：出力欄

図：

チャート入力をSubmitをしていないため出力欄が表示されていない状態の画面

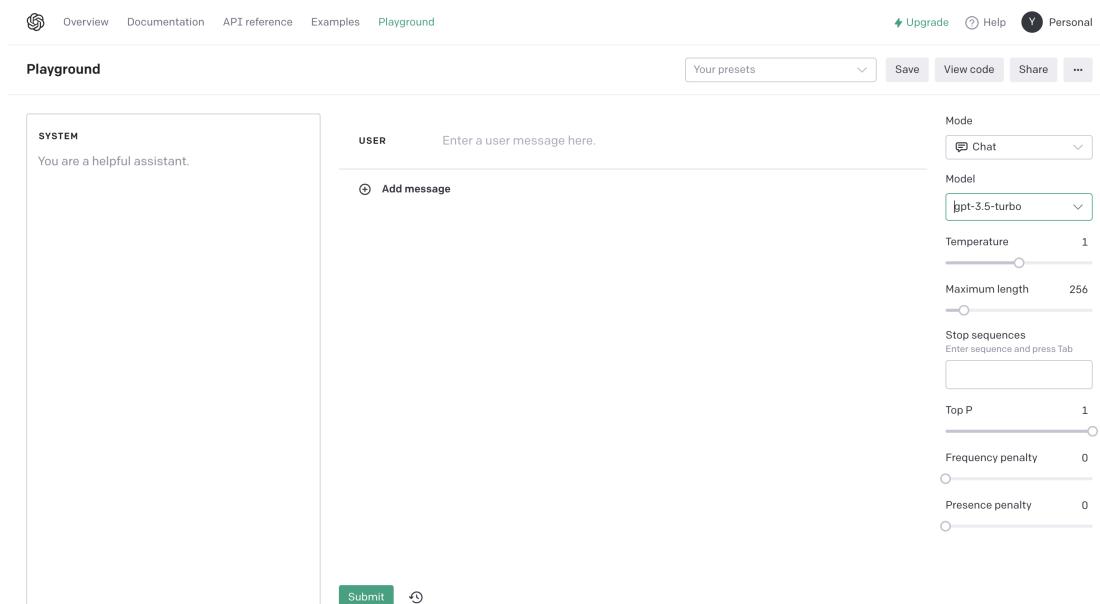


図2.5

出所: TechTechMediaより. 公式Webサイト |OpenAI Playgroundチャート画面

テーブル：パラメーターの例

モード	モデル
Chat	gpt-3.5-turbo
Temperatur	0
Maximum length	256
TopP	1

チャートの実践例について

「SYSTEM」:女子学生キャラクターのチャートAIです。

「USER」:おはよう

「ASSISTANT」:おはようございます。おにいちゃん。今日はどんな予定がありますか？

挿入モード

挿入モードはテキスト生成のテキストの続きではなく間のテキストのテキストを生成するモードで、物語のシーンの間を繋ぐテキストを生成したり、プログラムにコメントを挿入したりするときに使われる

この場合は挿入モードInsertを選んでモデルはモデル選択基準通りに選ぶこと

テーブル：パラメーターの例

モード	モデル
Insert	text-davinci-003
Temperatur	0
Maximum length	256
TopP	1

挿入モードはテキスト生成の入出力領域は同じで、テキスト挿入したいときにInsertを押すと画面が2つに分かれる

Insertボタンの位置

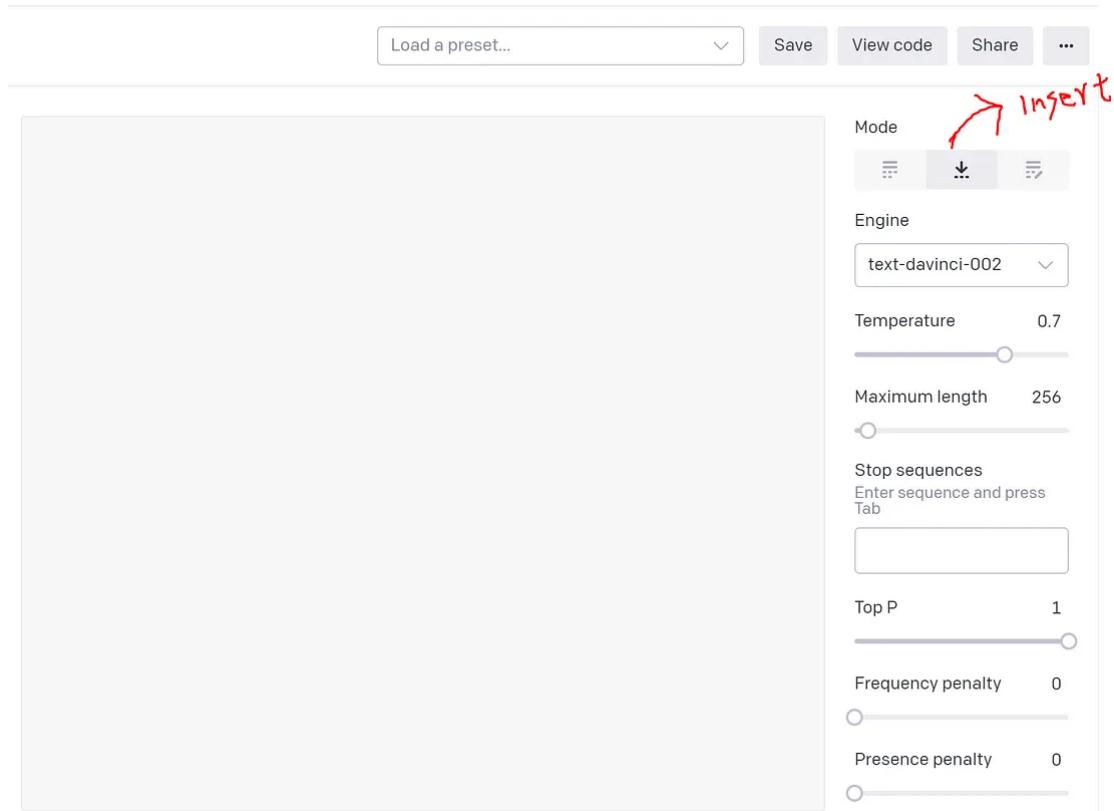


図2.6

出所:Ramsri Gouthamより. 公式Webサイト |OpenAI Playground挿入ボタンの画面

Insert(左) エリア内に挿入テキストを入力してください

画面が2つに分かれたイメージ図

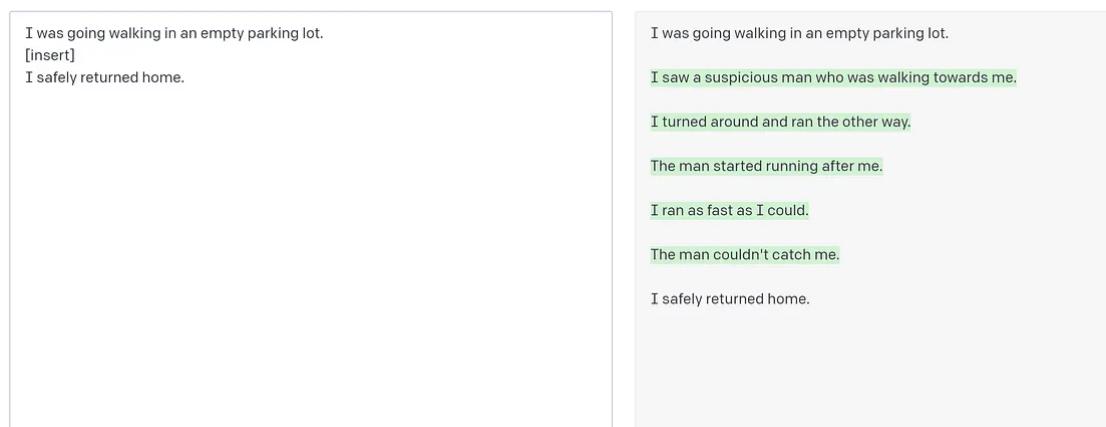


図2.7

出所:Ramsri Gouthamより. 公式Webサイト |OpenAI Playground実行後入出力画面

Submitボタンを押すとInsert(左) と同様な位置（右）に挿入テキストが表示され、テキストの挿入が完成する

テキスト挿入の実践例について

入力：

```
def helloworld( ):
    """
    説明:[Insert]
    """
    print("Hello World!")
helloworld()
```

出力：

```
def helloworld( ):
    """
    説明:Hello Worldを表示する関数
    """
    print("Hello World!")
helloworld()
```

編集モード

編集モードは任意のテキストとどのように編集するかを入力し、任意のテキストの編集結果を出力する

この場合は編集モードInsertを選んでモデルはモデル選択基準通りに選ぶこと

テーブル：パラメーターの例

モード	モデル
Edit	text-davinci-edit-001または code-davinci-edit-001
Temperatur	0
TopP	1

編集モードはテキスト生成の入出力領域は同じで、テキスト編集したいときにEditモードを選ぶ画面が3つに分かれる

単語の説明

「Input」：入力欄

「Instruction」：命令領域

「右側の欄」：出力欄

入力欄に任意のテキストを入力し、命令領域にどのように編集するかを入力する

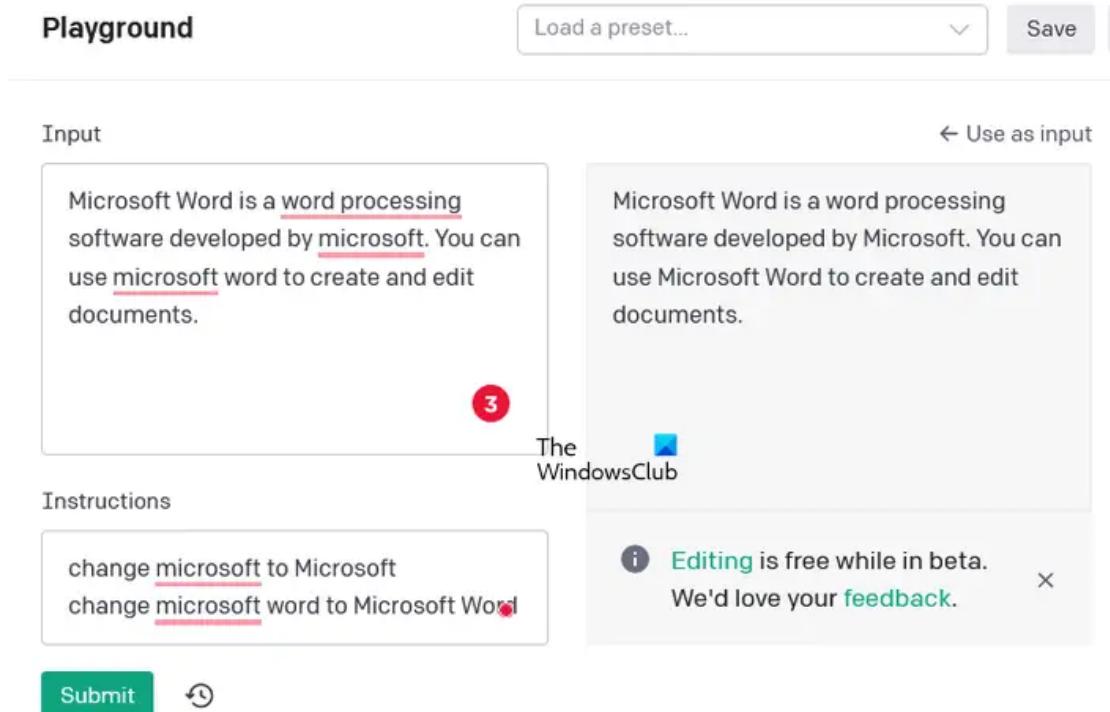


図2.8

出所:The Windows Clubより. 公式Webサイト |OpenAI Playgroundテキスト編集画面

Submitボタンを押すと右側の出力欄に編集済の任意のテキストが表示され、テキストの編集が完成する

テキスト編集の実践例について

- 1.入力：今日はとても楽かった
- 2.命令：誤字脱字を修正してください
- 3.出力：今日はとても楽しかった

DALL-Eの利用法

DALL-Eの利用料金

18ドル分のクレジットが無料枠で、3ヶ月以内の利用期限がある

画像生成はサイズと枚数によって料金が変わる

画像1枚当たりの料金表

画像サイズ	料金	日本円の目安
1024x1024	0.02 ドル	2.60円 (130円/ドルレートで計算)
512x512	0.018 ドル	2.34円 (130円/ドルレートで計算)
256x256	0.016 ドル	2.08円 (130円/ドルレートで計算)

DALL-Eのアカウント登録し方

利用料金がかかる（3か月以内の無料枠で試用できる）

登録の流れ：

chatGPTアカウントがあれば利用可能



ChatGPTのログイン状態でDALE-2のホームページへアクセス

<https://openai.com/blog/dall-e/>



<https://labs.openai.com/>

DALL-Eの画面が表示される

DALL-Eの画面構成

ボタン名

DALL-E:

画像生成画面ボタン

History:

履歴画面ボタン

Collection:

生成された作品コレクション画面ボタン

Surprise me:

ランダム切替ボタン

Generate:

生成確定ボタン

Upload an image to edit:

PCから画像生成の元として利用したい画像がある場合はこちらで画像ファイルをアップロードして画像のGenerateが可能

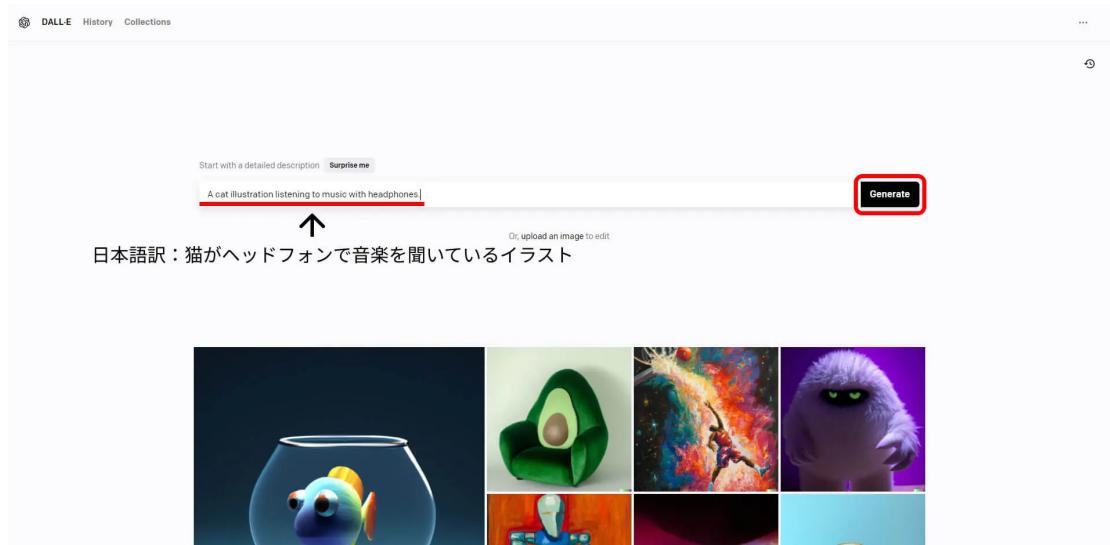


図2.9 出所:優良Webより. 公式Webサイト |DALL-Eの画像フォルダアップロードする画面

Variations機能:

生成結果の画像が拡大された状態でバリエーション画像を生成する機能ボタンが表示される

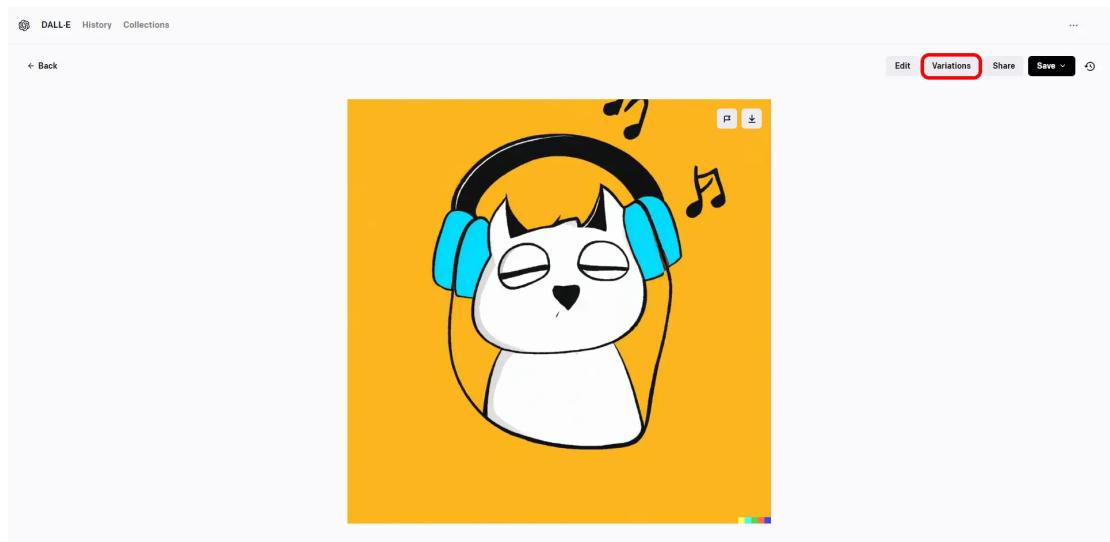


図2.10

出所:優良Webより. 公式Webサイト |DALL-Eの画像フォルダアップロードする画面

Editボタン機能の説明について

生成結果の画像が拡大された状態で編集機能Editボタンが表示される、ボタンを押すと5つの編集マークが画像の下に表示される

➤ : フレーム移動マーク

追加で画像を生成する枠の位置を移動する

👉 : 全体移動マーク

画像位置をずらす

消しゴムマーク：

画像の任意の場所を丸い形で消す

選択して太さ（右側の○—○バー）を調整し、書き換える場所を消す

+フレーム：

画像のアップロード

任意の位置に追加した枠に追加画像を生成する

フレーム+：

フレーム追加

新たに生成枠を追加する（Add generation frame）

ローカルの画像ファイルを選択し、追加された枠に配置する

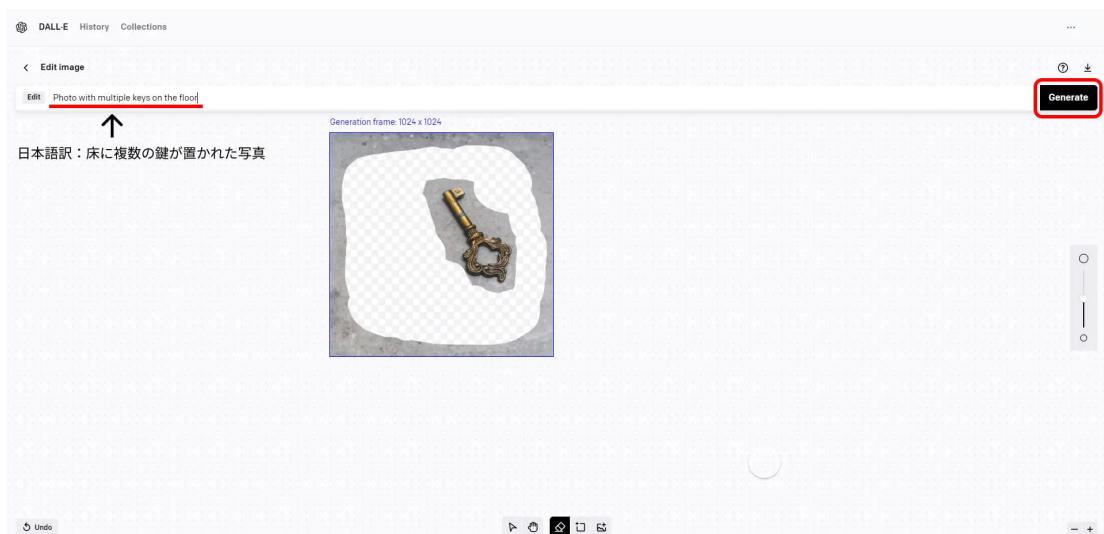


図2.11

出所：優良Webより。公式Webサイト |DALL-Eの編集ボタン機能説明画面

その他のボタン

-/+ (右下)：拡縮

↓:ダウンロード

Undo (左下) :やり直し

Acceptボタン:決定（編集後画像に当てると表示されるボタン）

Cancelボタン:キャンセル（編集後画像に当てると表示されるボタン）

DALL-Eの画像生成し方

表示された画面の中央のテキス入力欄に生成したい画像の説明文を入力し、Generateボタンを押すだけで簡単に画像生成ができる

画像生成の実践例

入力：猫がヘッドフォンで音楽を聴いているイラスト

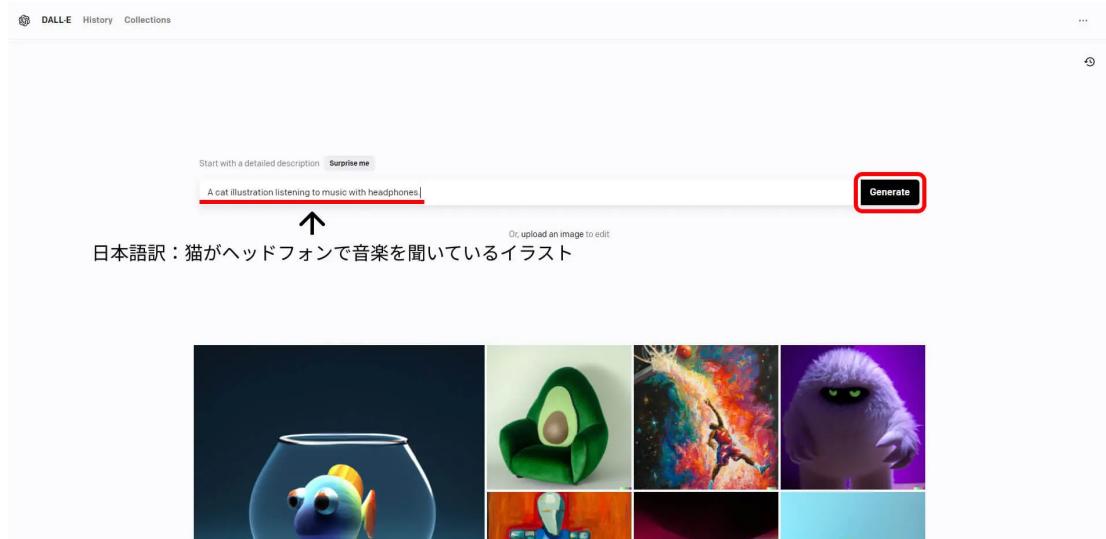


図2.12

出所:優良Webより. 公式Webサイト |編集画面の枠内に作成したい画像を表すテキストの入力例

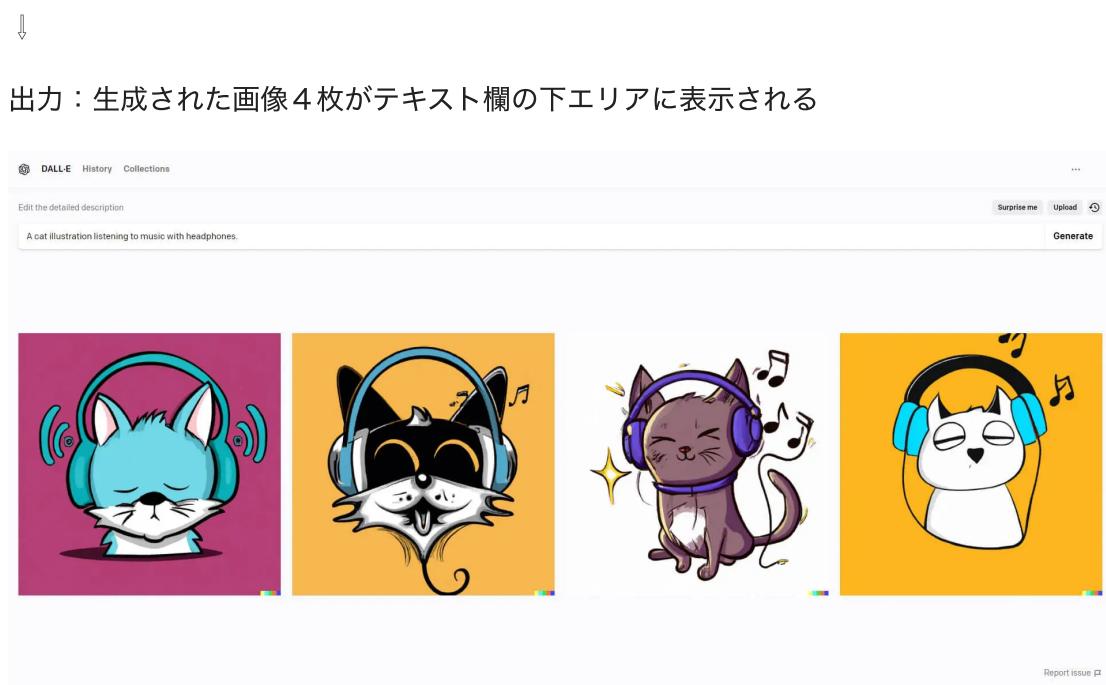


図2.13

出所:優良Webより. 公式Webサイト |編集画面のテキスト生成出力画像4枚の例



中から利用したい1枚の画像をクリックして拡大し、ダウンロード

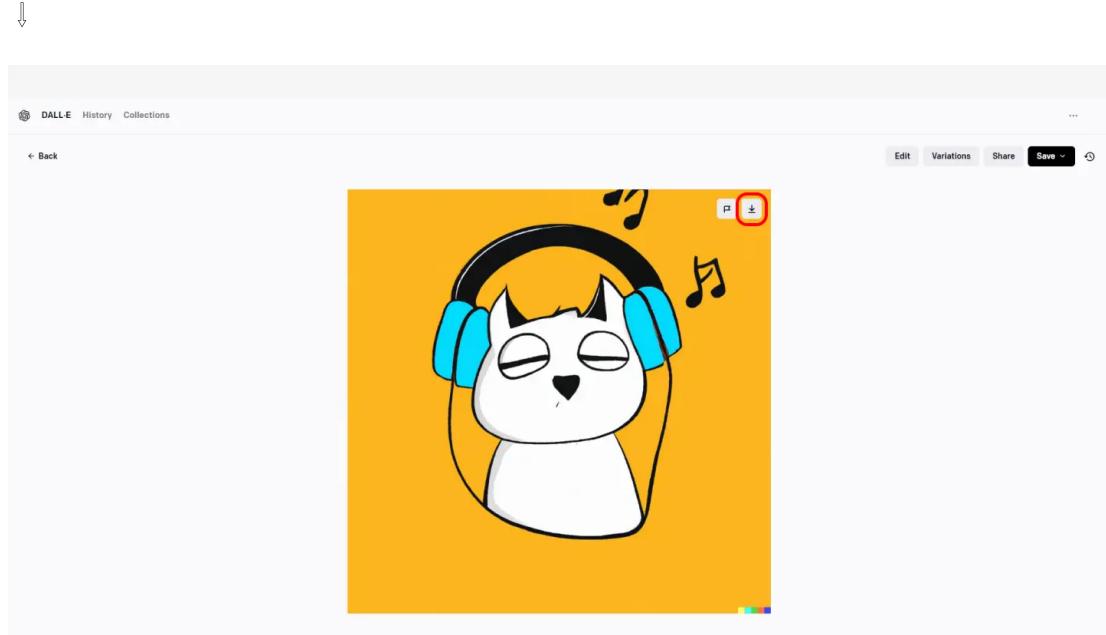


図2.14

出所:優良Webより. 公式Webサイト |選択した拡大画像の例

コレクションのお気に入りとして保存も可能（この場合Saveで保存）

↓

ここからはバリエーション、編集、インペインティング、アウトペインティングの機能を使うことが可能（続きご参照）

バリエーション機能について

選択された好きな画像が拡大表示されたら「Variations機能ボタン」を押すとさらに画像が4枚追加され合計5枚の画像が表示される

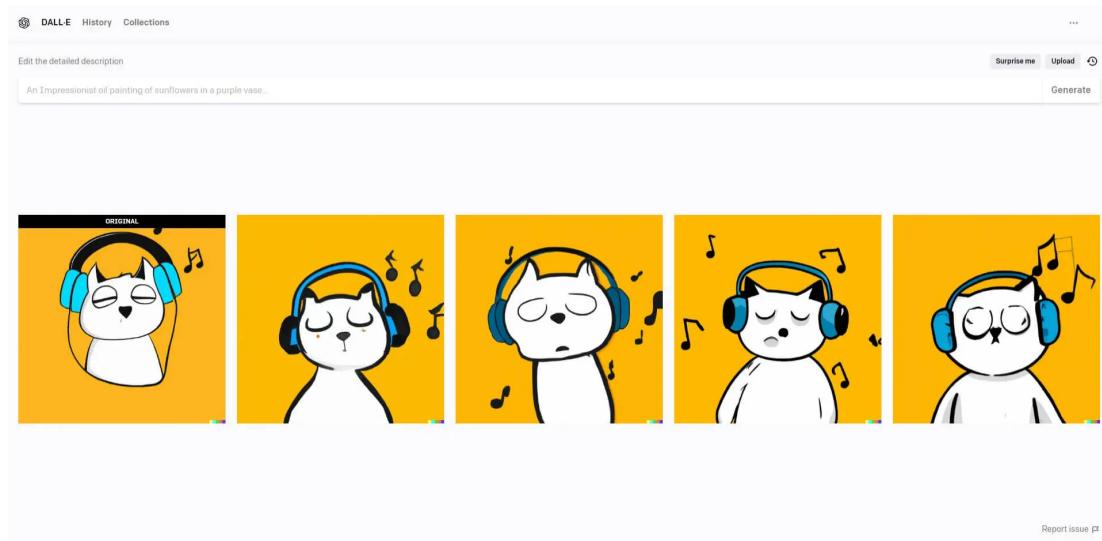


図2.15

出所:優良Webより. 公式Webサイト |選択した画像のバリエーション例

中から利用したい1枚の画像をクリックして拡大し、ダウンロード



コレクションのお気に入りとして保存も可能（この場合Saveで保存）

インペインティング機能について

インペインティング機能で画像内的一部分を描き換えることができる

画像の一部を削除し、その部分のみをAIが編集する実践例

編集したい画像を拡大する・または画像ファイルをアップロードする



Editボタンを押す



消しゴムボタンで消したいエリアを消す、消されたエリア内に編集したい内容が表示されるようになる

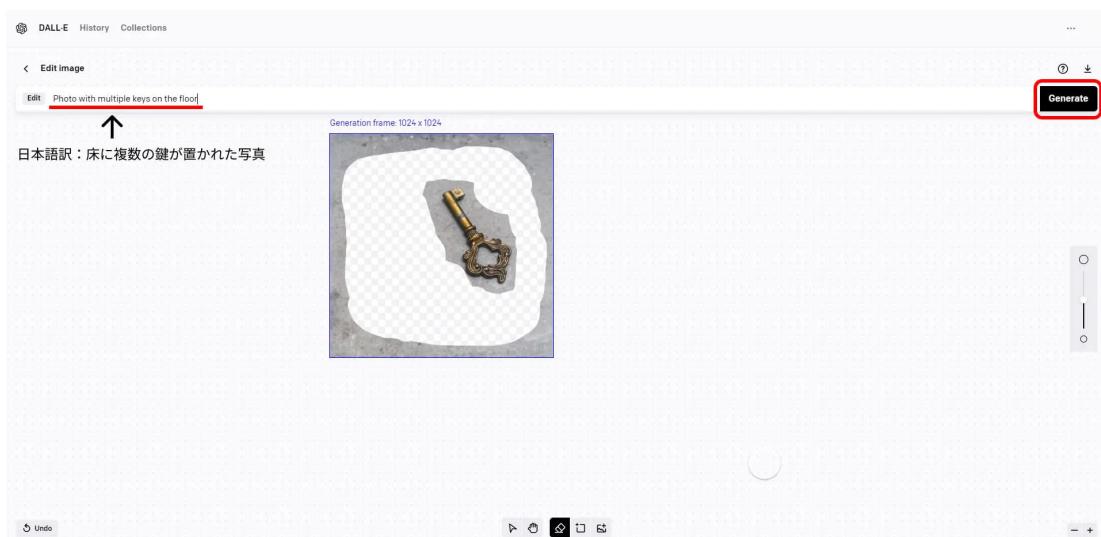


図2.16

出所:優良Webより. 公式Webサイト |編集テキストの入力例



中央のテキス欄に編集内容を入力する

入力：

床に複数の鍵が置かれた写真 (*)

((*)写真では英語版の入力内容が表示されているがこの入力内容は 日本語に訳したテキスト文章である)



図2.17

出所:優良Webより. 公式Webサイト |AIに編集して欲しい部分を消しゴムで削除する例



Generate



出力 :

消されたエリア内に床に複数の鍵が置かれた状態の画像4枚が表示される、元の鍵はそのまま元の位置にあることが確認できる

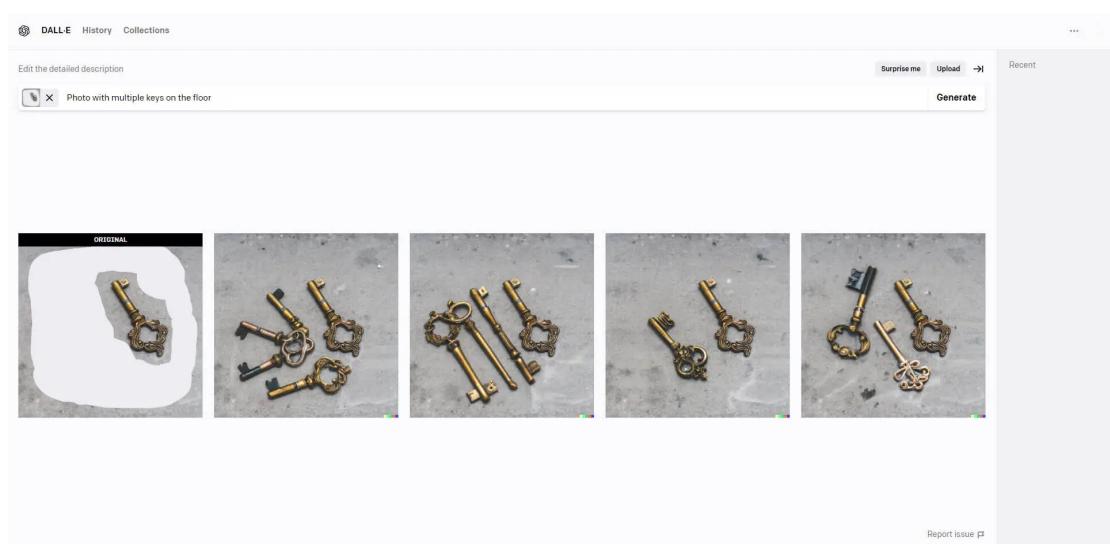


図2.18

出所:優良Webより. 公式Webサイト |AIが編集して出力した画像4枚の例



中から利用したい1枚の画像をクリックして拡大し、ダウンロード



コレクションのお気に入りとして保存も可能（この場合Saveで保存）

アウトペインティング機能について

アウトペインティングは画像の枠外を入力テキストに従って描き足すことができる

画像の右側に新たに画像を追加配置する実践例

編集したい画像を拡大する・または画像ファイルをアップロードする



Editボタンを押す



編集画面で、「Add generation frame」と書かれたボタンをクリックする

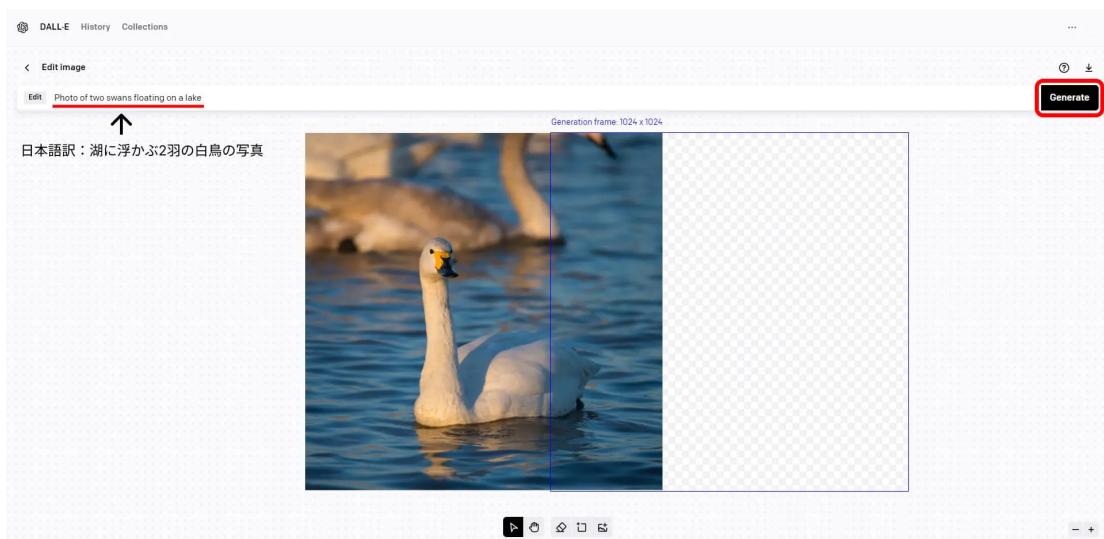


図2.19

出所:優良Webより. 公式Webサイト |元画像の外に枠を描き足す例



表示された枠（画像外の空の右枠）を、拡張したい範囲に移動させる。



➤と👉を使って、元画像とフレームの位置を自由に調整可能

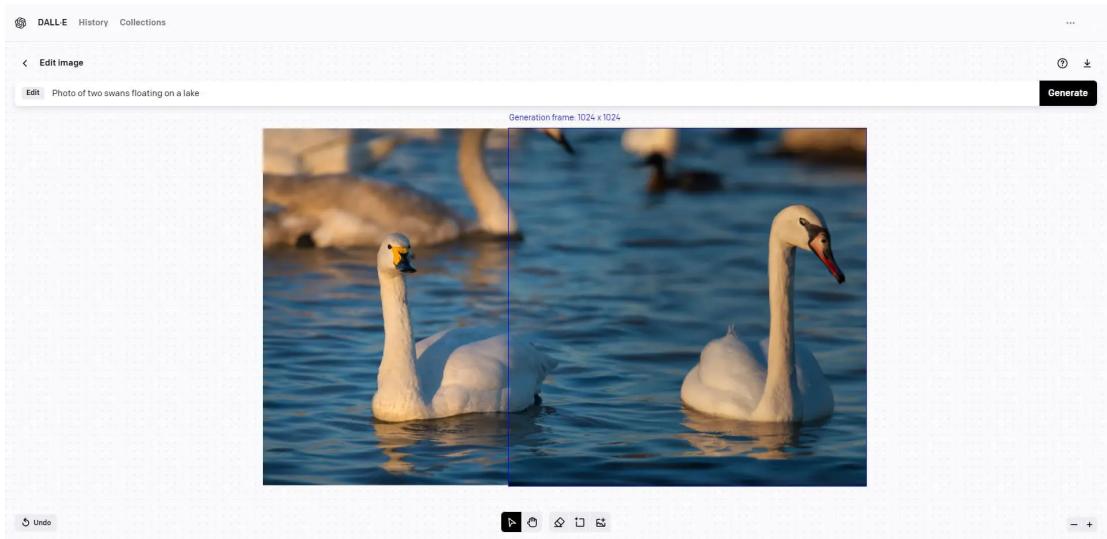


図2.20

出所:優良Webより. 公式Webサイト |元画像の拡張編集出力例

↓

完成形のイメージを伝えるテキストを入力する

入力:湖に浮かぶ2羽の白鳥の写真

↓

「Generate」をクリックする

↓

出力:4枚のアウトペインティングが生成される

↓

生成画像を左右ボタンで確認し、Acceptボタンで決定

↓

ダウンロードボタンでダウンロード

↓

コレクションのお気に入りとして保存も可能（この場合Saveで保存）

GitHub Copilotの利用法

GitHub Copilot利用をサポートするエディターは4種類ある

VSCode

VS

JetBrains

以後はVSCodeエディタベースで解説する

GitHub Copilotの利用料金

月10ドル（約1,300円、130円/ドルレートで計算） 年100ドル 60日間無料トライアル

詳細利用料金

<https://github.com/features/copilot#pricing>

GitHubのアカウント登録し方

GitHub社が提供しているオープンソースプロジェクトなどのソースコードを公開し、管理することができるプラットフォームである

GitHub <https://github.com>

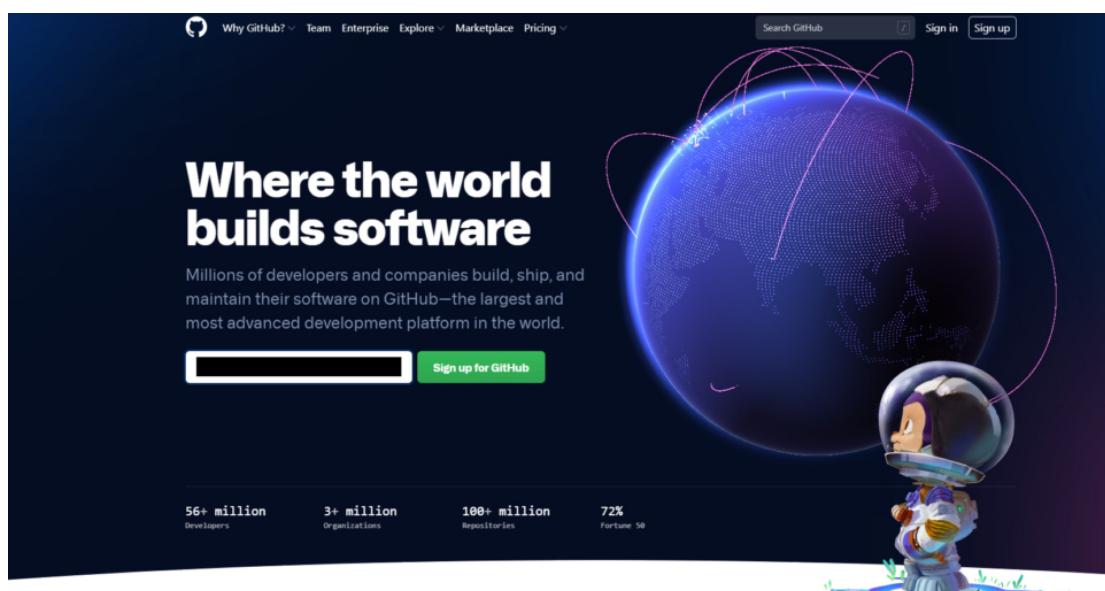


図2.21 出所:Classmethod.jpより. 公式Webサイト |GitHubのホーム画面

登録の流れ

ボックスに登録用のメールアドレスを入力し、[Sign up for GitHub] ボタンをクリックする。



[Create your account] 画面が表示されたら必要項目を入力する。

- Username : 任意のユーザー名
- Email address : GitHub登録用メールアドレス

- Password : 文字数は15文字以上、または数字と小文字を含む8文字以上のパスワードを指定する)
- Email preferences : 製品の最新情報、お知らせなどを不定期にメール受け取りたい場合にはチェックを入れる
- Verify your account : [検証する] ボタンをクリックし、画面の指示に従って画像を選択することで、ロボットではないことを証明する

↓

入力後、アクティブになった [Create account] ボタンをクリックする。

↓

[Welcome to GitHub] 画面が表示される。

↓

下のほうにスクロールして [Complete setup] ボタンをクリックする。

↓

[Please verify your email address] 画面が表示される。

↓

登録用に入力したメールアドレス宛に、GitHubからメールが届いていることを確認する。※必要に応じてメール設定で、下記のメールを受信できるようにしておいてください。 メール件名: [GitHub] Please verify your email address. 送信メールアドレス: GitHub (noreply@github.com)

↓

メール本文の [Verify email address] ボタンをクリックする。

↓

GitHub画面上で [Your email was verified.] メッセージが表示されれば登録が完了

↓

ここで操作を終了する場合は、 [skip this for now] ボタンをクリックする。

GitHub Copilotのアカウント登録し方

登録の流れ

事前にGitHubのアカウント登録や初期設定完了させる

↓

GitHub管理画面右上のメニューから「Settings」をクリックする

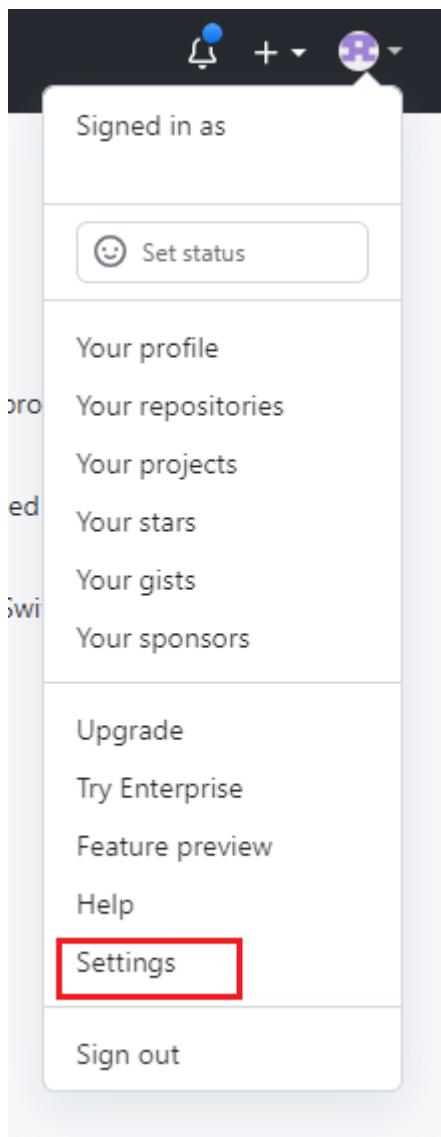


図2.22

出所:Kagoya Japan Inc.より. 公式Webサイト |GitHubのホーム設定ボタン

↓

次に画面左側に表示されたメニューの中から、「Copilot」を選ぶ

↓

GitHub Copilotの管理画面が表示される

↓

画面中央付近にある「Start Free trial」をクリックする

↓

「Monthly Plan(10/月)」「Yearly Plan (100/年)」のいずれかを選んで「Get access GitHub Copilot」をクリックする

*このまま設定を進めるとGitHub Copilot無料トライアルの申込が完了し、60日間はGitHub Copilotを無料で利用でき、その間にトライアルの解除も可能で、トライアルを

解除すれば課金はされない

GitHub Copilotのプランを「キャンセル」したい場合

GitHubの管理画面から「Settings」→「Access」→「Billing and plans」→「Plans and usage」→契約プランの管理画面が表示されるので、「GitHub Copilot」欄の「Edit」→「Cancel trial」を押せばプランをキャンセルできる



申込情報の入力画面の指示に従い氏名や住所などを入力した上で、「Save」をクリックする



支払情報の入力画面で支払い方法をクレジットカードもしくはPayPalで設定し、画面の指示に従い支払情報を入力して「Save Payment information」をクリックする

*次ページ以降は画面の指示に従い入力を進める



支払方法としてクレジットカードを選ぶと、以下のような確認画面が表示され、内容を確認して間違いがなければ「Submit」をクリックする



ソースコードをGitHubが活用するのを許可するか確認する画面が表示され、いずれか選択し「Save and get started」をクリックする

英語表示

Suggestions matching public code

公開コードに一致する提案を許可するか
(Allow許可/Block否定)

Allow GitHub to use my code snippets for product improvements

自分のコードを製品の改善に利用することを
許可するか (ON/OFF) |

日本語解説

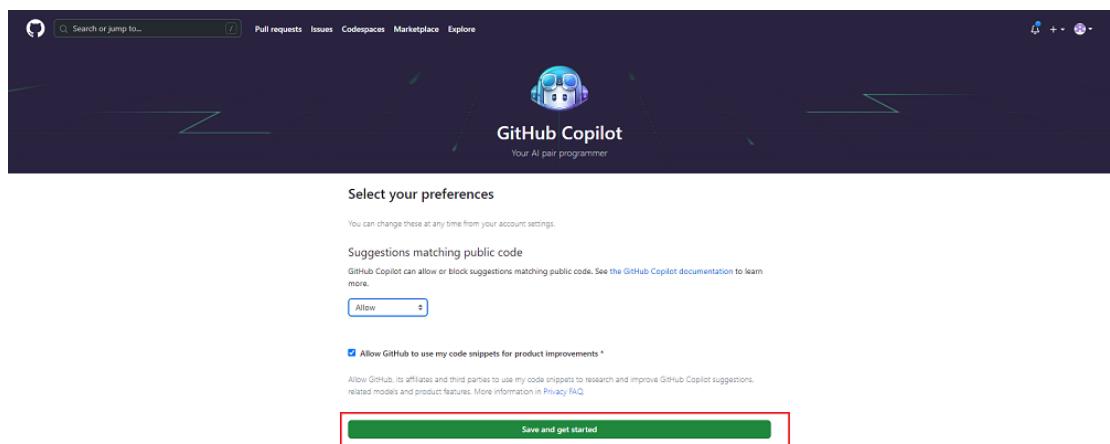


図2.23

出所:Kagoya Japan Inc.より. 公式Webサイト |GitHub Copilotの登録は完了画面



これでGitHub Copilotの登録は完了

VSCodeエディター・Copilotプラグインのインストール

VS Codeのインストールの流れ

VS Codeのインストール用ファイルのダウンロードサイト

Download Visual Studio Code:

<https://code.visualstudio.com/download>



使うPCの仕様別でダウンロードするファイルを選択

https://code.visualstudio.com/download?_fsi=qVaReeKa



例:Windows11の場合

<https://code.visualstudio.com/docs/?dv=win%E3%82%92%E3%82%AF%E3%83%AA%E3%83%83%E3%82%AF%E3%81%9C>



ファイルを開き、利用条件の同意してインストール完了



VS Codeを起動し、画面左側の「拡張機能」メニューをクリックする

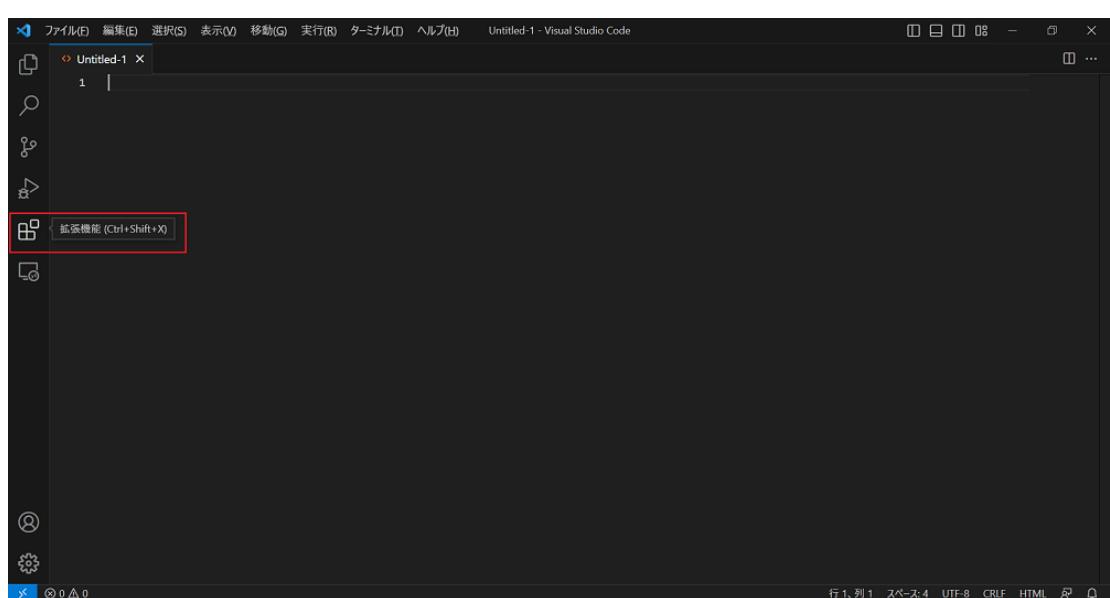


図2.24

出所:Kagoya Japan Inc.より. 公式Webサイト | VS Code画面左側の「拡張機能」メニュー

—

↓

拡張機能の中から「GitHub Copilot」を検索しインストールする

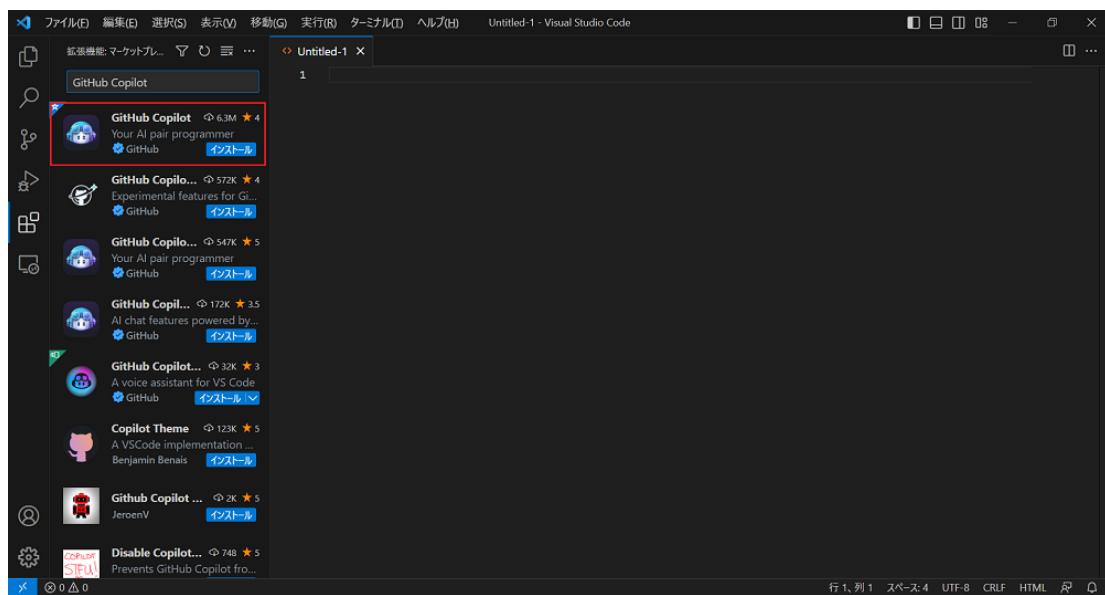


図2.25

出所:Kagoya Japan Inc.より. 公式Webサイト | 「GitHub Copilot」を検索しインストールする画面の例

↓

GitHubへのサインインを求められるので、画面の指示に従いサインイン作業を行う

↓

これでプラグインのインストールは完了

GitHub Copilotの使い方

GitHub Copilotは、対応VS Codeエディタに専用プラグインGithub Copilotの拡張機能をインストールすることで使えるようになる

GitHub CopilotのサイトでGitHub Copilotを有効化

VSCodeの下端に表示されるGitHub Copilot状態アイコンで有効化・無効化を切り替えることができる

ファイルの作成し方

New file



Save as



作成するファイル名 `hello_copilot.py` を指定する



ファイル保存先を選ぶ



保存

コードの候補の表示し方

```
def helloworld( ):
```

上記のコードを入力しただけで以下の候補が薄灰色で表示される

```
print("Hello World!") #コードの候補
```



```
def helloworld( ):
```

Tabキーを押すだけでコード入力タイプせずにコードがかてに入力されるようになる

```
print("Hello World")
```

コードの代替候補の表示

ctrl+Enterを押すとさらに代替候補が表示される

```
print("Hello World!")
```

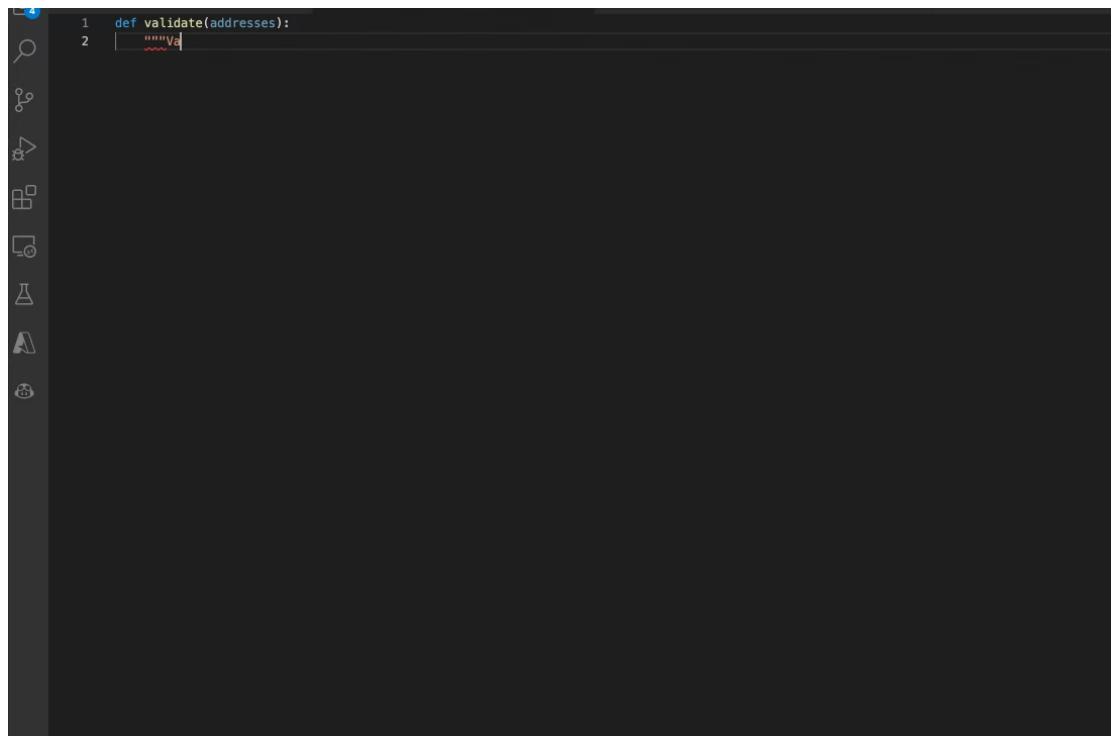
```
print("Hello, World!") #間にカンマがある候補
```

```
print("Hello World")#!マークがない候補
```

```
print("Hello world!") #wが小文字の候補
```

その他の候補例：

図のようにテキストが進みにつれて候補が次から次提案される



A screenshot of a code editor interface, likely Quora's code search feature. On the left is a vertical toolbar with icons for search, file operations, and other functions. The main area shows two lines of Python code:

```
1 def validate(addresses):  
2     """Va|
```

The cursor is at the end of the word "Va". A dropdown menu on the right side of the editor lists several code snippets as suggestions, starting with "validate" and "ValueError".

図2.26

出所:Quiita.comより. 公式Webサイト |コードの予測候補画面の例

コメントからのコードの候補の表示

入力：

```
def read_file(filename):
```

出力：

with open(filename, 'r')as f: #以下のように候補コードが薄く表示される、このコード
セルでは例なので薄くなっていない return f.read()