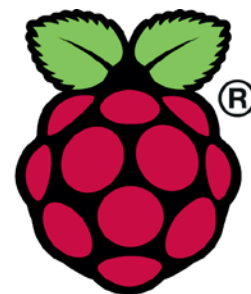
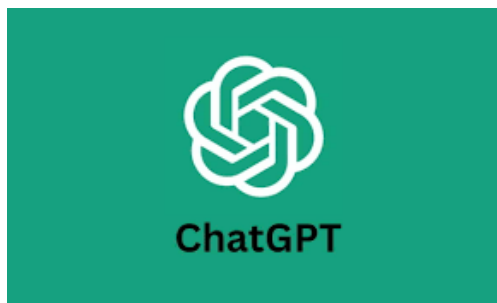


抜粋

はじめてのchatGPT開発キット

RaspberryPiでchatGPTの基礎から関連APIで応用まで習得、体験
基本編



Raspberry Pi

スペクトラム・テクノロジー株式会社

<https://spectrum-tech.co.jp>

sales1@spectrum-tech.co.jp

開発キット 目次

ページ

• 開発キット 接続構成	<u>4</u>
• 開発キット 設定	
1. Raspberry Pi起動	<u>5</u>
2. モニター・キーボード接続	<u>6</u>
3. 初期設定	
(1) パスワード設定、SU切り替え	<u>7</u>
(2) ネットワーク設定	<u>8</u>
• RaspberryPi運用マニュアル	
• RaspberryPiについて	<u>9</u>
• Linux基本コマンド	<u>10</u>
• RaspberryPi基本操作	<u>11</u>
• 日常運用(ウイルススキャン、更新)	<u>12</u>
• 開発キット 全体像	<u>14</u>
• ハード概要	<u>15</u>
• ソフト概要	<u>16</u>
1. Chatgpt概要	
① Chatgpt設定	<u>17</u>
② ChatGPT使い方	<u>19</u>

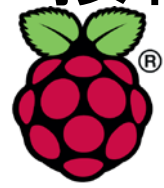
抜粋版のためページと一致しません

開発キット 目次

	ページ
2. ChatGPT機能(ipynbで説明)	
一覧	25
① GPT	26
② Embedding	29
③ 画像生成	42
④ Fine-tuning	43
⑤ 音声認識	47
⑥ 感情分析	49
⑦ その他	50
3. Openai api(pythonで説明)	
• 一覧	51
① Chat	52
② Completions	53
③ 画像生成	54
④ Embeddings	57
⑤ 音声認識	58
⑥ Files	59
⑦ Fine-tuning	61
⑧ Moderations	63
⑨ Edits	64
4. Openai事例(英語、日本語版で説明)	
一覧	65
各事例(①から④)	67

抜粋版のためページと一致しません

開発キット 接続構成



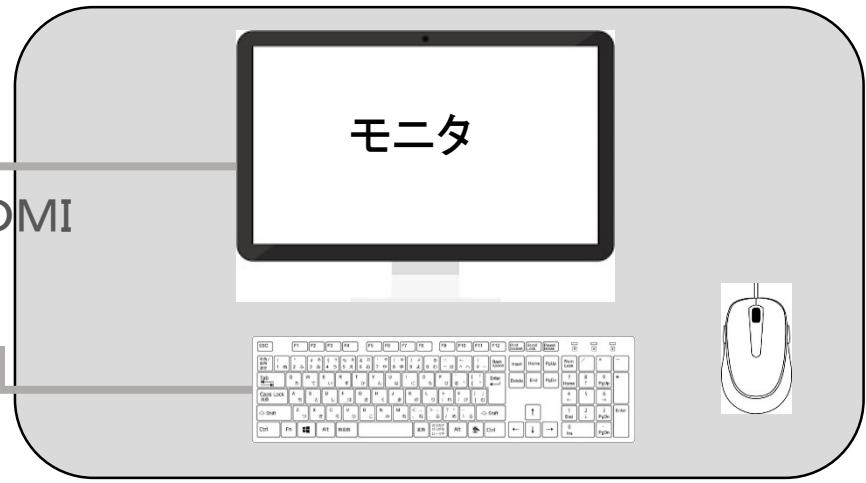
Raspberry Pi 4B

Raspberry Pi



microHDMI

USB



マイクロUSB電源 (TypeC)

USB電源
5V3A

開発キット設定

1. Raspberry Pi起動

① マイクロSDカードを挿入



- ・Raspberry Piの裏面からマイクロSDカードを挿入します。
- ・Raspbian OS(Bullseye系列)と必要なモジュール、ドライバをインストール済です。

② 電源ケーブル接続



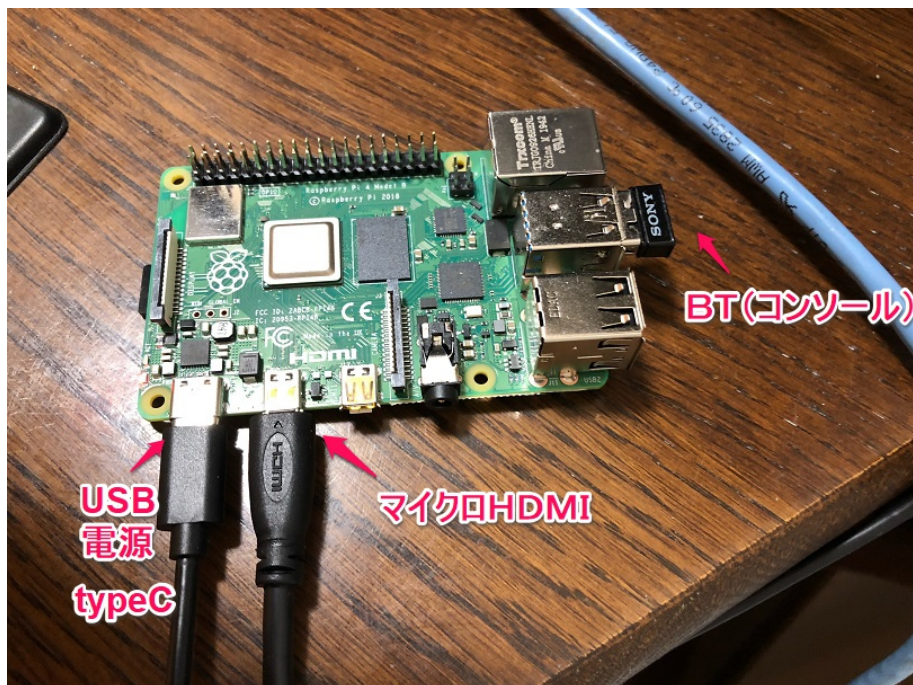
- ・Raspberry PiのマイクロUSB端子に電源コードを挿入します。
- ・電源コードは付属しておりません。必要な場合は、オプションで販売しております。5V3A必要なため、AC100Vから変換器経由で給電してください。**パソコンのusbの場合起動しないことがあります。**

開発キット設定

2. モニター、キーボード接続

Raspberry Piとモニタ(HDMI接続)、マウス、キーボード(USB接続)を接続し、初期設定を行います。

ノートPCの場合は、
表示されません



① モニタ接続

- モニター(TV、PCでHDMI端子のあるもの)を準備します。
- マイクロHDMIケーブルにより、Raspberry Piとモニタを接続します。

② マウス、キーボード接続

- デスクトップPCに接続しているマウス、キーボードをUSBでRaspberry Piと接続します。
- マウス、キーボードがBluetoothで接続されている場合は、本体のBluetooth USBを外してRaspberry Piと接続します。

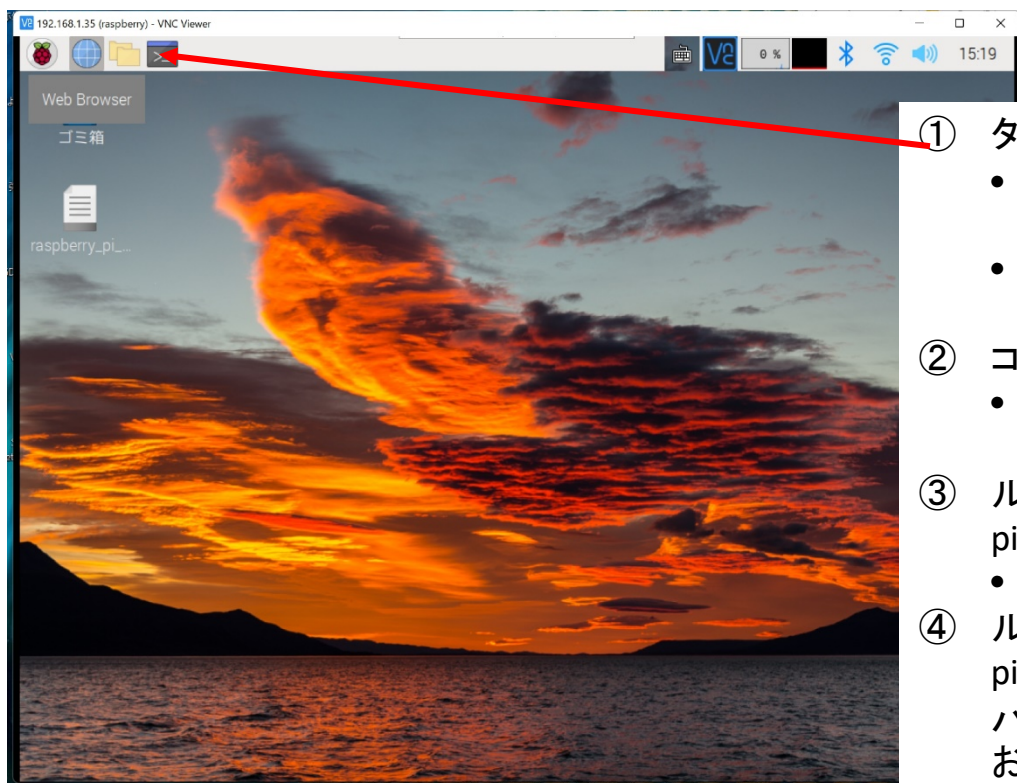
vnc接続も可能ですが、
jupyter notebookが遅くて
動作しません

開発キット設定

3. 初期設定

(1) パスワード設定、SU切り替え

Raspberry Piが立ち上がったことを確認し、各種設定を行います。



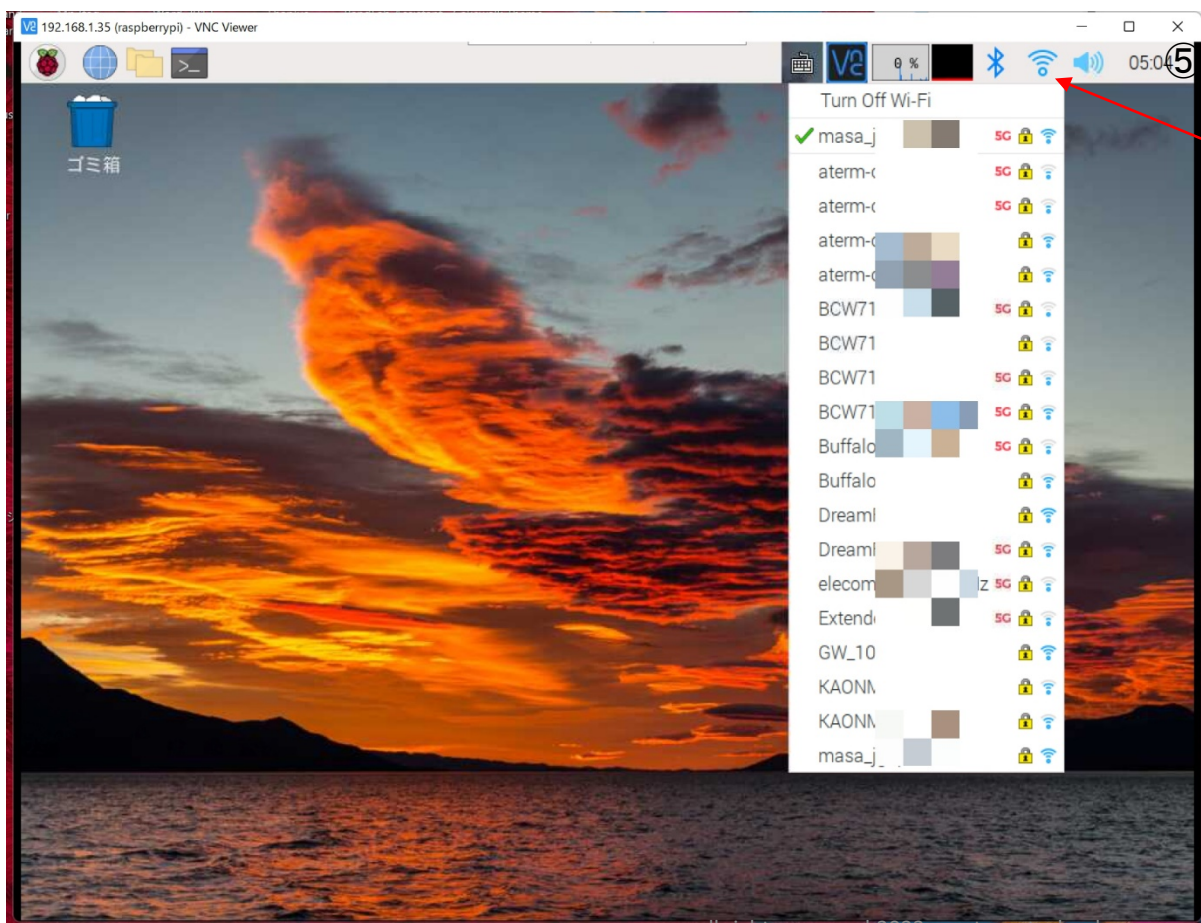
- ① ターミナルの立ち上げ
 - 上のバーのターミナルをクリックして、ターミナルウインドウを開きます
 - 初期は、ユーザアカウント: pi, ホスト名: raspberrypiで立ち上がります: pi@raspberrypi:~ \$
- ② コマンド入力への切り替え
 - 日本語と英語の切り替えは全角/半角のキーで行います。
- ③ ルートユーザのパスワード設定
 - pi@raspberrypi:~ \$ **sudo passwd root**
 - 2回入力してください。
- ④ ルートユーザへの切り替え
 - pi@raspberrypi:~ \$ **su -**
 - パスワード: カーソルは動きませんが入力してentをおしてください。
 - root@raspberrypi:~#

開発キット設定

3. 初期設定

(2) ネットワーク設定

Raspberry Piが立ち上がったことを確認し、各種設定を行います。



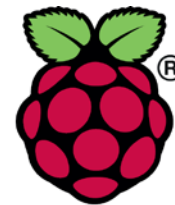
ネットワーク設定

• WiFiの場合

- 上段のLANマークをクリックして、接続したい、WiFi(SSID)を選びます。
- パスワードを入力します。
- 次回からは自動で接続されます。

• LANの場合

- 自動で接続されます。
- IPアドレスは、LAN上のDHCPから自動で取得します。
- IPアドレスをマニュアルで設定することも可能です。



Pi運用マニュアル

1. Raspberry Piについて

既に全世界で1000万台以上販売された手のひらサイズのコンピュータです。
LinuxベースのRasbianOSで動作しております。

2. Linux基本コマンド

① システム関係

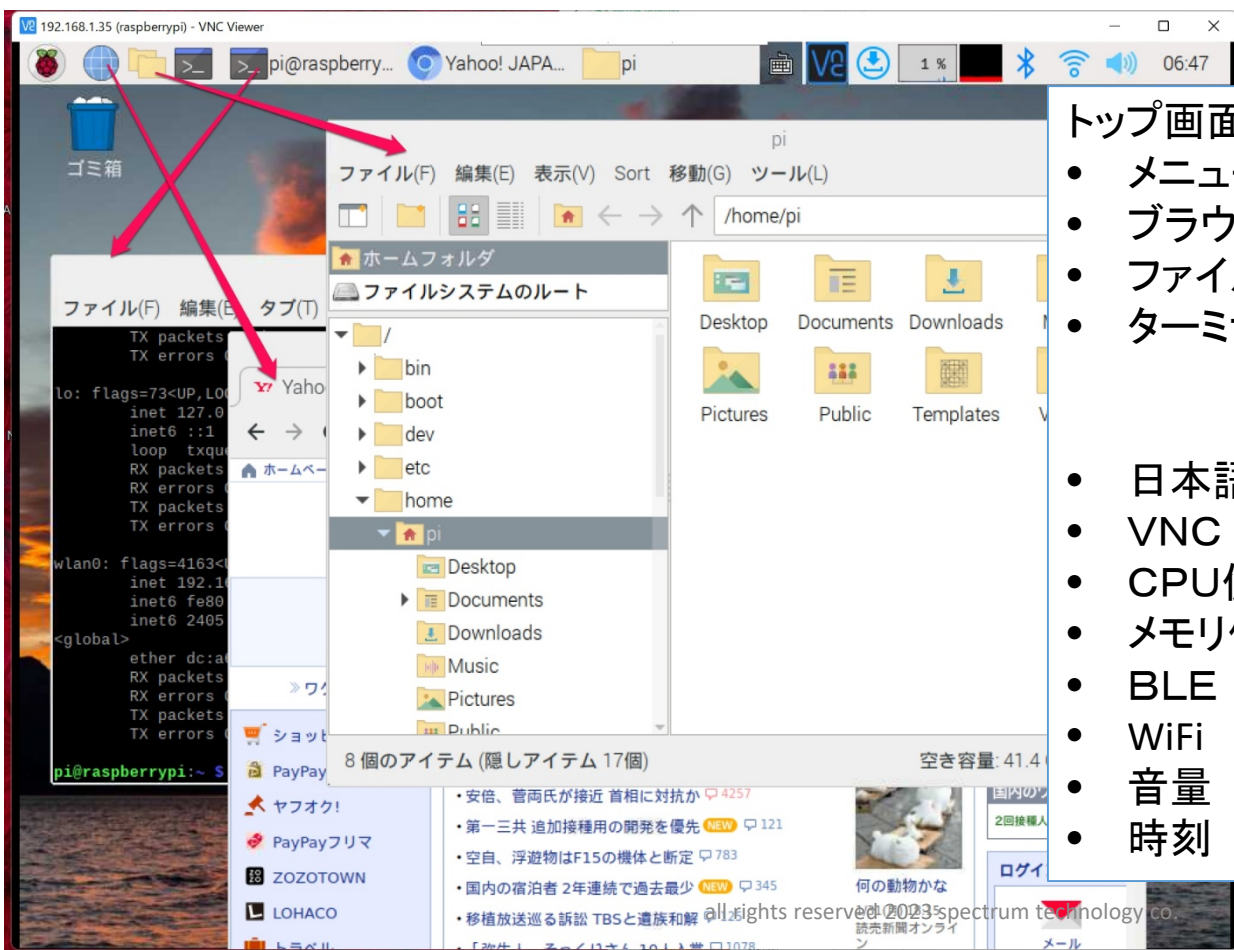
- 起動: 電源を入れると自動で起動します。
- 再起動: # reboot
又は、アプリケーション>ログアウト>再起動; 左上のメニューから
- 終了: # shutdown
又は、アプリケーション>ログアウト>シャットダウン; 左上のメニューから
- ログアウト # logout
又は、アプリケーション>ログアウト>ログアウト; 左上のメニューから
- 日本語／英語の入力切替: キーボードの全角/半角キーで切替ます。



RaspberryPi運用マニュアル

3. Raspberry Piの基本操作

① 表示画面と内容



トップ画面(上段のタスクバーで選択)

- メニュー
- ブラウザ
- ファイルマネージャ
- ターミナル

- 日本語入力
- VNC
- CPU使用率
- メモリ使用量
- BLE
- WiFi
- 音量
- 時刻

開発キット 全体像

ハードウェア

Raspberry Pi4B



+

物品追加



電源アダプタ
(オプション)



HDMIケーブル
(オプション)

OS



Rasbian OS

API関係



LangChain



画像系

Stable diffusion

(オプション)

プログラム言語
python



Triton(オプション)

AI detector

GPTZero
 crossplag™
 Copyleaks

ハード概要

本体

品名	項目	内容	備考
Raspberry Pi 4 model B	CPU	1.5GHz 64bit クアッドコアCoretex-A72	
	GPU	デュアルコア VideoCore VI® 500MHz	
	メモリ	4GB RAM	
	OS	Raspbian bullseye(Debianベース)	
	インターフェース	2.4/5GHz WiFi(802.11 bgnac), Bluetooth 5.0, BLE, 1G ether, USB 2.0x2, USB 3.0x2, micro HDMIx2, microSDカード, 40 GPIO pin	
	電源／消費電力	Micro USB Type C 3.0A	
	サイズ	85x56x18mm	
付属品		内容	備考
ケース		透明、白、黒、銀から選択できます。	
microSD 32GB		Raspbian OS, 必要なモジュールをインストールして提供します。お客様が設定するものは必要最低限のパスワード設定、WiFi設定になります。	
プログラム		Chatgpt,langchain,llamaindex, 関連pip インストール済, Pythonサンプルプログラム多数	
マニュアル		開発キット 基本編、応用編	

USB電源ケーブル、HDMIケーブルは付属していません。

別途オプション品を購入ください

ソフト概要

提供するソフトウェアの概要です。

区分	ソフト名	バージョン	備考
OS	Raspbian	Bullseye 64bit	
プログラム言語	python	3.9.2	
APIプログラム	openai	0.27.8	
	langchain	0.0.201	
	llamaindex	0.6.28	
その他	Jupyter notebook、 matplotlibなど多数のpipライブラリ		

1. ChatGPT概要

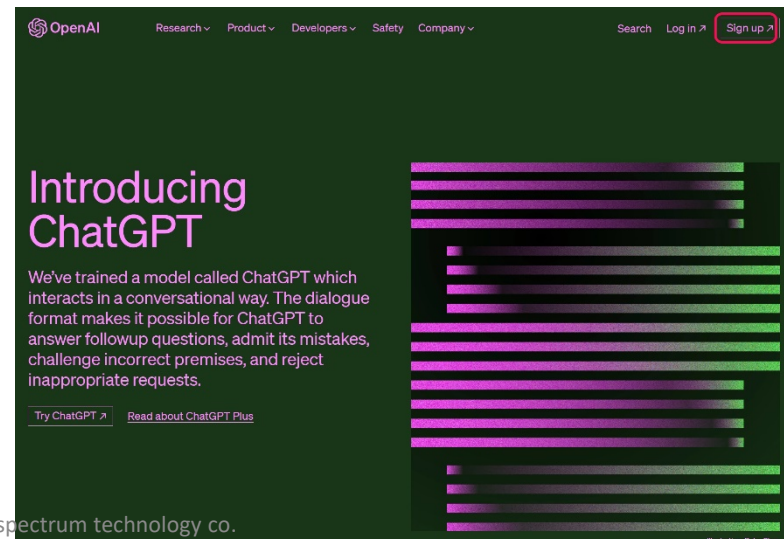
①. ChatGPT設定

- ChatGPTとは

- ChatGPT(チャットジーピーティー、英語: Chat Generative Pre-trained Transformer) [1]は、OpenAIが2022年11月に公開した人工知能チャットボットである。原語のGenerative Pre-trained Transformerとは、「生成可能な事前学習済み変換器」という意味である[2]。OpenAIのGPT-3ファミリーの大規模な言語モデルに基づいて構築されており、教師あり学習と強化学習の両方の手法を使って転移学習され、機械学習のサブセットである深層学習を使って開発されている。
- 生成AIの代表格で2021年9月までのデータを学習している。それ以降は未対応

- ChatGPT設定

- <https://openai.com/blog/chatgpt>
- 上記にアクセスし、サインイン



1. ChatGPT概要

②. ChatGPT使い方

- chatGPTトップページ <https://chat.openai.com/>
- Apiトップページ <https://platform.openai.com/overview>
- 料金ページ <https://platform.openai.com/account/usage>
- Openai api key生成
 - <https://platform.openai.com/account/api-keys>
 - Apiを使用するためのapi keyを生成します。
 - 生成されたkeyをコピーして保存。
 - export OPENAI_API_KEY='sk-xxxxx' と端末に入力し、使用します。

The screenshot illustrates the process of creating a new OpenAI API key. The main interface shows the 'API keys' section with a table of existing keys and a '+ Create new secret key' button. A modal window titled 'Create new secret key' is open, showing a text input field with 'My Test Key' and a 'Create secret key' button. A second modal window shows the generated key in a text box with a copy icon.

all rights reserved 2023 spectrum technology co.

1. ChatGPT概要

②. ChatGPT使い方

- chatGPTトップページ <https://chat.openai.com/>
- Apiトップページ <https://platform.openai.com/overview>
- 料金ページ <https://platform.openai.com/account/usage>
- Hello world 1

- <https://platform.openai.com/docs/libraries/python-bindings>

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-python/examples
```

```
$ python3 helloworld_ch1.py
```

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'
$ cd /home/pi/Documents/openai/openai-
python/examples
$ python3 helloworld_ch1.py
```

- 各自のキーを入力
- ターミナル毎に1回入力

```
~/Documents/openai/openai-python/examples/helloworld_ch1.py - mousepad
ファイル(F) 編集(E) 検索(S) 表示(V) 文書(D) ヘルプ(H)
test_cmd0804 x 生成AI使い方メモ0516.txt x helloworld_ch1.py x
1 import openai
2
3 completion = openai.ChatCompletion.create(model="gpt-3.5-turbo", messages=[{"role": "user",
4 print(completion.choices[0].message.content)
5
```

```
.../lxplug-ptbatt_0.22_arm64.deb を展開する準備をしています ...
lxplug-ptbatt (0.22) で (0.21 に) 上書き展開しています ...
lxplug-ptbatt (0.22) を設定しています ...
lxplug-cputemp (0.12) を設定しています ...
pi@raspberrypi:~ $ export OPENAI_API_KEY='sk-NTyF3uEoVWq9H5k'
pi@raspberrypi:~ $ cd /home/pi/Documents/openai/openai-python/examples
pi@raspberrypi:~/Documents/openai/openai-python/examples $ python3 helloworld_ch
.py
Hello! How can I assist you today?
pi@raspberrypi:~/Documents/openai/openai-python/examples $ sudo apt install kde-
```

- Hello worldと質問
- Hello! How can I assist you today?と回答

1. ChatGPT概要

②. ChatGPT使い方

- chatGPTトップページ <https://chat.openai.com/>
- Apiトップページ <https://platform.openai.com/overview>
- 料金ページ <https://platform.openai.com/account/usage>
- Hello world 2

- <https://platform.openai.com/docs/guides/gpt>

\$ export OPENAI_API_KEY='sk-xxxxx'

\$ cd /home/pi/Documents/openai/openai-python/examples

\$ python3 helloworld_ch2.py

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'
$ cd /home/pi/Documents/openai/openai-
python/examples
$ python3 helloworld_ch2.py
```

各自のキーを入力

```
~/Documents/openai/openai-python/examples/helloworld_ch2.py - Mousepad
ファイル(F) 編集(E) 検索(S) 表示(V) 文書(D) ヘルプ(H)

test_cmd0804 × 生成AI使い方メモ0516.txt × helloworld_ch2.py

1 # Note: you need to be using OpenAI Python v0.27.0 for the code below to work
2 import openai
3
4 messages=[
5     {"role": "system", "content": "You are a helpful assistant."},
6     {"role": "user", "content": "Who won the world series in 2020?"},
7     {"role": "assistant", "content": "The Los Angeles Dodgers won the World Series in
8     2020."},
9     {"role": "user", "content": "Where was it played?"}
10 ]
11 completion=openai.ChatCompletion.create(
12     model="gpt-3.5-turbo",
13     messages=messages
14 )
15
16 print(completion)
17
```

- 2020年WSの場所？と質問
- Texasと回答
- token数なども記載

all rights reserved 2023 spectrum technology co.

```
pi@raspberrypi: ~/Documents/openai/openai-python/examples
mailcap (3.69) のトリガを処理しています ...
desktop-file-utils (0.26-1) のトリガを処理しています ...
hicolor-icon-theme (0.17-2) のトリガを処理しています ...
gnome-menus (3.36.0-1) のトリガを処理しています ...
pi@raspberrypi:~/Documents/openai/openai-python/examples $ python3 helloworld_ch2.py
{
  "id": "chatcmpl-7aYFyDZHSrI2qU5ZehSCPumUL3V9t",
  "object": "chat.completion",
  "created": 1688945846,
  "model": "gpt-3.5-turbo-0613",
  "choices": [
    {
      "index": 0,
      "message": {
        "role": "assistant",
        "content": "The 2020 World Series was played at the Globe Life Field in Arlington, Texas."
      },
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 53,
    "completion_tokens": 18,
    "total_tokens": 71
  }
}
```

2. ChatGPT機能 一覧

項番	項目	api名等	内容	方式	備考
1	GPT	Chat completion api	チャット モデルはメッセージのリストを入力として受け取り、モデルが生成したメッセージを出力として返します。	ipynb	
2		Function calling	json形式で定義した機能呼び出して使用	ipynb	
3		Completion api	Chat completionのlist形式で会話を定義したものがpromptへ	ipynb	
4	Embedding	How to get embeddings	エンベディングを取得するには、選択したエンベディング モデル ID とともにテキスト文字列をエンベディング API エンドポイントに送信します	ipynb	
5		How to count tokens with tiktoken.ipynb	token数を算出	ipynb	
6		Obtain dataset	datasetのロード	ipynb	
7		Data visualization in 2D	データの可視化	ipynb	
8		Data visualization in 3D	データの可視化	ipynb	
9		Embedding as a text feature encoder for ML algorithms	スコアの予測	ipynb	
10		Classification using the embedding features	5段階評価の分類	ipynb	
11		Zero-shot classification	3をzeroとして再評価、1-2:nega, 4-5:posi、precision、recallの確率が向上する	ipynb	
12		Clustering	データからクラスタ抽出	ipynb	
13		Text search using embeddings	文字検索	ipynb	
14		Code search using embeddings	プログラミング・コード検索	ipynb	
15		Recommendations using embeddings	レコメンデーション	ipynb	
16	画像生成	DALL-Eを使った画像生成、加工	画像生成、加工	ipynb	
17	Fine-tuning	分類	tweet記事のhockey(600), baseball(597)の記事を使って、学習と作成されたモデルで、検証を行う。作成方法、削除なども	ipynb	時間かかる。課金注意
18		Q&A事例	東京オリンピックのデータを学習	ipynb	時間かかる。課金注意
19	音声認識	音声認識、翻訳	mp4の音声データから認識、翻訳	ipynb	
20	感情分析	感情分析	文字から感情を分析	ipynb	
21	その他	トークン制限	最大のトークン数	-	
22		課金	課金アラーム設定	-	

2. ChatGPT機能

①. GPT

• Apiトップページ <https://platform.openai.com/overview>

• Function calling

• <https://platform.openai.com/docs/guides/gpt/function-calling>

• json形式で定義した機能呼び出して使用。

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-python/examples
```

```
$ python3 function_call.py
```

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-python/examples
```

```
$ python3 function_call.py
```

```
~/Documents/openai/openai-python/examples/function_call.py - Mousepad
ファイル(F) 編集(E) 検索(S) 表示(V) 文書(D) ヘルプ(H)
test_cmd0804 x 生成AI使い方メモ0516.txt x function_call.py x
1 import openai
2 import json
3
4
5 # Example dummy function hard coded to return the same weather
6 # In production, this could be your backend API or an external API
7 def get_current_weather(location, unit="fahrenheit"):
8     """Get the current weather in a given location"""
9     weather_info = {
10         "location": location,
11         "temperature": "72",
12         "unit": unit,
13         "forecast": ["sunny", "windy"]
14     }
15     return json.dumps(weather_info)
16
17
18 def run_conversation():
19     # Step 1: send the conversation and available functions to gpt
20     messages = [{"role": "user", "content": "What's the weather like in Boston?"}]
21     functions = [
22         {
23             "name": "get_current_weather",
24             "description": "Get the current weather in a given location",
25             "parameters": {
26                 "type": "object",
27                 "properties": {

```

• Get_current_weatherの関数を定義して、呼び出して回答に使用

```
pi@raspberrypi: ~/Documents/openai/openai-python/examples
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
{"completion_tokens": 18,
 "total_tokens": 71
}
pi@raspberrypi:~/Documents/openai/openai-python/examples $ python3 function_call.py
{"id": "chatcmpl-7aZbdXAqUme5UmGyqJaUjQL3q00u8",
 "object": "chat.completion",
 "created": 1688951033,
 "model": "gpt-3.5-turbo-0613",
 "choices": [
   {
     "index": 0,
     "message": {
       "role": "assistant",
       "content": "The current weather in Boston is 72 degrees Fahrenheit with sunny and windy conditions."
     },
     "finish_reason": "stop"
   }
 ],
 "usage": {
   "prompt_tokens": 72,
   "completion_tokens": 16,
   "total_tokens": 88
 }
 }
```

2. ChatGPT機能

①. GPT

• Apiトップページ <https://platform.openai.com/overview>

• Completion api

• <https://platform.openai.com/docs/guides/gpt/completions-api>

• Chat completionのlist形式で会話を定義したものがpromptへ。どちらを使うかは、tokens数、回答の速度などにより使い分け

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-python/examples
```

```
$ python3 comp1.py
```

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'
$ cd /home/pi/Documents/openai/openai-
python/examples
$ python3 comp1.py
```

```
~/Documents/openai/openai-python/examples/comp1.py - Mousepad
ファイル(F) 編集(E) 検索(S) 表示(V) 文書(D) ヘルプ(H)
test_cmd0804 × 生成AI使い方メモ0516.txt × comp1.py
1 import openai
2
3 response = openai.Completion.create(
4     model="text-davinci-003",
5     prompt="Write a tagline for an ice cream shop."
6 )
7 print(response)
```

• アイスクリーム屋の宣伝文句

```
pi@raspberrypi: ~/Documents/openai/openai-python/examples
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
}
},
"usage": {
  "prompt_tokens": 72,
  "completion_tokens": 16,
  "total_tokens": 88
}
}
pi@raspberrypi:~/Documents/openai/openai-python/examples $ python3 comp1.py
{
  "id": "cml-7aZhGVQgjrVdQXo7nzTuzW5hm8PyG",
  "object": "text_completion",
  "created": 1688951382,
  "model": "text-davinci-003",
  "choices": [
    {
      "text": "\n\n\"Indulge in Delicious Creamy Treats!\"",
      "index": 0,
      "logprobs": null,
      "finish_reason": "stop"
    }
  ],
  "usage": {
    "prompt_tokens": 10,
    "completion_tokens": 13,
    "total_tokens": 23
  }
}
```


2. ChatGPT機能

②. Embedding(埋め込み)

- Apiページ <https://platform.openai.com/docs/guides/embeddings>
- 用途: 検索、クラスタ、レコメンド、異常検知、分類に使用
- Clustering
 - <https://platform.openai.com/docs/guides/embeddings/use-cases>
 - obtain_dataの食レポから4つのクラスタを発見: pet food, food drink, positivex2。

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-cookbook/examples
```

```
$ jupyter notebook
```

```
• Clustering.ipynb
```

```
jupyter Clustering Last Checkpoint: 先週金曜日 10:04 (autosaved)
```

```
File Edit View Insert Cell Help Trusted Python 3 (ipykernel)
```

```
In [1]: import openai
import os
os.environ["OPENAI_API_KEY"] = "sk-xxxxx"

response = openai.ChatCompletion.create(
    model="gpt-3.5-turbo",
    messages=[{"role": "user", "content": "I have a list of reviews for a product. Can you categorize them into clusters?"}],
    temperature=0,
)

print(response["choices"][0]["text"].replace("\n", ""))

sample_cluster_rows = diff[Cluster == 1]
for j in range(len(sample_cluster_rows)):
    print(sample_cluster_rows.Score.values[j])
    print(sample_cluster_rows.Summary.values[j])
    print(sample_cluster_rows.Text.str[0:100].values[j])

print("\n * 100")
```

Cluster 0 Theme: All of the reviews are positive and the customers are satisfied with the product they purchased.

5. Loved these gluten free healthy bars, saved \$\$ ordering on Amazon: These Kind Bars are so good and healthy & gluten free. My daughter

1. Should advertise coconut as an ingredient more prominently: First, these should be called Mac + Coconut bars, as Coconut is the #2

5. very good!!! just like the runts
great flavor, def worth getting
I even o

5. Excellent product: After scouring every store in town for orange peels and not finding an

5. delicious: Gummi Frogs have been my favourite candy that I have ever tried. of co

Cluster 1 Theme: All of the reviews are about pet food.

2. Messy and apparently undelicious: My cat is not a huge fan. Sure, she'll lap up the gravy, but leaves th

4. The cats like it: My 7 cats like this food but it is a little yucky for the human. Plece

5. cant get enough of it!!!! Our lil shih tzu puppy cannot get enough of it. Everytime she sees the

1. Food caused illness: I switched my cats over from the Blue Buffalo Wilderness Food to this

5. My furbabies LOVE these!: Shake the container and they come running. Even my boy cat, who isn't

Cluster 2 Theme: All of the reviews are positive and express satisfaction with the product.

5. fog Chaser Coffee: This coffee has a full body and a rich taste. The price is far below t

5. Excellent taste: This is to me a great coffee, once you try it you will enjoy it. this

4. Good, but not Wolfgang Puck good: Honestly, I have to admit that I expected a little better. That's not

5. Just My Kind of Coffee: Coffee Masters Hazelnut coffee used to be carried in a local coffee/pa

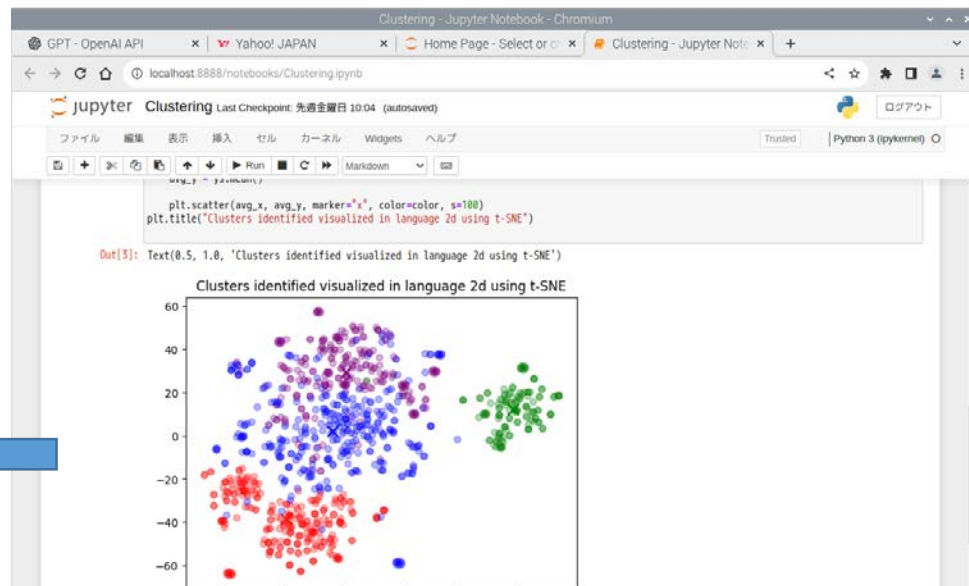
5. Rodeo Drive is Crazy Good Coffee!: Rodeo Drive is my absolute favorite and I'm ready to order more! That

Cluster 3 Theme: All of the reviews are about food or drink products.

5. Wonderful alternative to soda pop: This is a wonderful alternative to soda pop. It's carbonated for thos

5. so convenient, for so little!: I needed two vanilla beans for the Love Goddess cake that my husbands

• クラスタの内容



2. ChatGPT機能

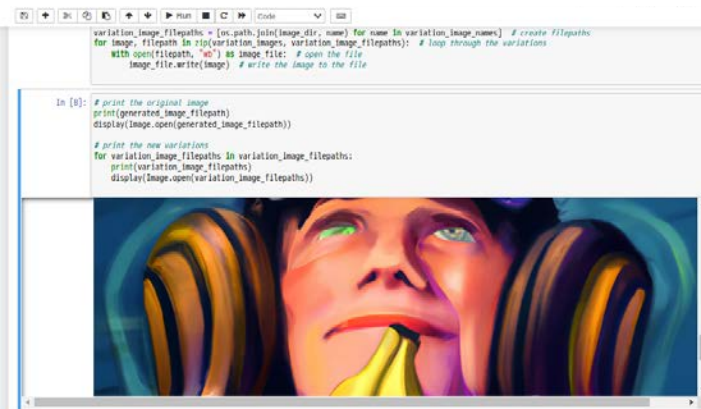
③. 画像生成

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'
$ cd /home/pi/Documents/openai/openai-cookbook/examples/dalle
$ jupyter notebook
```

- Apiページ <https://platform.openai.com/docs/guides/images>
- 用途: 文字から画像を生成、加工などを実施。著作権など要注意
- DALL-Eを使った画像生成、加工
 - <https://platform.openai.com/docs/guides/images/introduction>
 - \$ export OPENAI_API_KEY='sk-xxxxx'
 - \$ cd /home/pi/Documents/openai/openai-cookbook/examples/dalle
 - \$ jupyter notebook
 - Image_generations_edits_and_variations_with_DALL-E.ipynb

「A cyberpunk monkey hacker dreaming of a beautiful bunch of bananas, digital art」から生成
右はその変化分



2. ChatGPT機能

⑤. 音声認識

- Apiページ <https://platform.openai.com/docs/guides/speech-to-text>
- 用途: 音声を認識してテキストで出力。日本語は、漢字が今一つ
- 音声認識、翻訳
 - <https://platform.openai.com/docs/guides/speech-to-text/quickstart>

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-cookbook/examples/speech_text
```

```
$ python3 transcription.py
```

```
$ python3 translate.py
```

• 漢字が今一つ

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'
$ cd /home/pi/Documents/openai/openai-cookbook/examples/speech_text
$ python3 transcription.py
$ python3 translate.py
```

```
pi@raspberrypi: ~/Documents/openai-cookbook/examples/speech_text
ファイル(F) 編集(E) タブ(T) ヘルプ(H)

Jupyter Notebook 6.5.4 is running at:
http://localhost:8888/?token=98303b23b71e78d52ad1d40b7dded978c765cf2ca21ac451
or http://127.0.0.1:8888/?token=98303b23b71e78d52ad1d40b7dded978c765cf2ca21ac451
このノートブックサーバをシャットダウンしますか? (y/[n])y
[C 11:05:21.029 NotebookApp] シャットダウンの確認
[I 11:05:21.030 NotebookApp] Shutting down 0 kernels
[I 11:05:21.031 NotebookApp] Shutting down 0 terminals
pi@raspberrypi:~/Documents/openai/openai-cookbook/examples $ cd speech_text
pi@raspberrypi:~/Documents/openai/openai-cookbook/examples/speech_text $ python3
transcription.py
pi@raspberrypi:~/Documents/openai/openai-cookbook/examples/speech_text $ python3
transcription.py
{
  "text": "\"\u672c\u65e5\u306f\u8056\u5178\u306a\u308a \u672c\u65e5\u306f\u8056\u5178\u306a\u308a \u3066\u3069\u30c8 \u3066\u3069\u30c8\""
}
pi@raspberrypi:~/Documents/openai/openai-cookbook/examples/speech_text $ python3
transcription.py
本日は聖典なり 本日は聖典なり テスト テスト
pi@raspberrypi:~/Documents/openai/openai-cookbook/examples/speech_text $ python3
translate.py
Honjitsu wa seiten nari. Honjitsu wa seiten nari. Test. Test. Test.
pi@raspberrypi:~/Documents/openai/openai-cookbook/examples/speech_text $
```

ST Spectrum Technology

3. Openai api一覽

項番	項目	api名等	内容	方式	備考
1	Openai api	Chat		python	
2		Completions		python	
3		画像生成	画像生成	python	
			画像編集	python	
			画像変化	python	
4		embeddings		python	
5		音声認識		python	
6		Files	Fine-tuningで使用するファイル操作	python	
7		Fine-tuning	fine-tuningの生成、リスト	python	
8		Moderations	感情分析	python	
9		Edits	編集	python	

3. Openai api

②. Completions

- Apiページ <https://platform.openai.com/docs/api-reference/completions>

- 用途: api関数を説明、機能の説明と重複

- Completions

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-api
```

```
$ python3 api_req.py
```

「Say this is a test」の
promptに対して

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-api
```

```
$ python3 api_req.py
```

```
pi@raspberrypi: ~/Documents/openai/openai_api
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
{
  "role": "assistant",
  "content": "Hello! How can I assist you today?"
}
pi@raspberrypi:~/Documents/openai/openai_api $ python3 api_req.py
{
  "id": "cmpl-7azfXDI4sORRMb1HqDtbFrDeocqr",
  "object": "text_completion",
  "created": 1689051219,
  "model": "text-davinci-003",
  "choices": [
    {
      "text": "\n\nThis is indeed a test",
      "index": 0,
      "logprobs": null,
      "finish_reason": "length"
    }
  ],
  "usage": {
    "prompt_tokens": 5,
    "completion_tokens": 7,
    "total_tokens": 12
  }
}
```

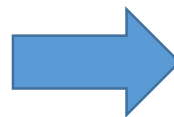
3. Openai api

③. 画像生成

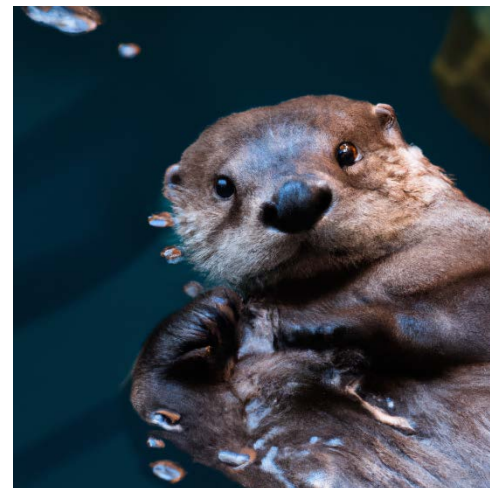
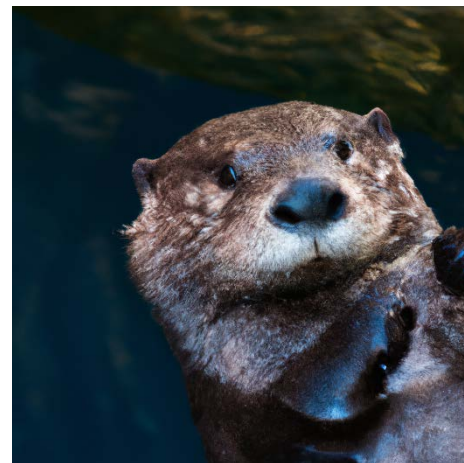
- Apiページ <https://platform.openai.com/docs/api-reference/images>
- 用途: api関数を説明、機能の説明と重複
- 画像変化

```
$ export OPENAI_API_KEY='sk-xxxxx'  
$ cd /home/pi/Documents/openai/openai-api  
$ python3 api_img_va.py
```

オリジナル



生成



コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'  
$ cd /home/pi/Documents/openai/openai-api  
$ python3 api_img_va.py
```

3. Openai api

④. embeddings

- Apiページ <https://platform.openai.com/docs/api-reference/embeddings>
- 用途: api関数を説明、機能の説明と重複
- embeddings

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai-api
```

```
$ python3 api_emb.py
```

「The food was delicious and the waiter...」の入力に対するベクトル値出力

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'  
$ cd /home/pi/Documents/openai/openai-api  
$ python3 api_emb.py
```

```
pi@raspberrypi: ~/Documents/openai/openai_api  
ファイル(F) 編集(E) タブ(T) ヘルプ(H)  
-0.02184096910059452,  
-0.014133138582110405,  
-0.015529944561421871,  
0.004111054353415966,  
0.0026444080285727978,  
-0.03139004483819008,  
0.041091497987508774,  
-0.017739437520503998,  
-0.01859021931886673,  
0.014514085836708546,  
0.010399856604635715,  
-0.015301376581192017,  
-0.01935211382806301,  
-0.002817421453073621  
  ]  
}  
],  
"model": "text-embedding-ada-002-v2",  
"usage": {  
  "prompt_tokens": 8,  
  "total_tokens": 8  
}  
}  
pi@raspberrypi:~/Documents/openai/openai_api $
```

4. Openai事例一覧(その1)

項番	項目	api名等	内容	方式	備考
1	事例	Q&A	Q&A	python	
2		Grammar correction	文法訂正	python	
3		Summarize for a 2nd grader	2年生向けに要約	python	
4		Natural language to OpenAI API	Openai api言語に変換	python	
5		Text to command	文字をコマンドに変換	python	
6		English to other languages	英語から多言語翻訳	python	
7		Natural language to Stripe API	クレジット情報取り出し	python	
8		SQL translate	sql文生成	python	
9		Parse unstructured data	非構造化データを指定に合わせて抽出し、並び替え	python	
10		Classification	会社名から業種を作成し分類	python	
11		Python to natural language	pythonのプログラムを解説	python	
12		Movie to Emoji	映画を絵文字で表現	python	
13		Calculate Time Complexity	二重ループのコード解説	python	
14		Translate programming languages	プログラム言語変換	python	
15		Advanced tweet classifier	ツイート分類	python	
16		Explain code	プログラム解説	python	
17		Keywords	キーワードを抽出	python	
18		Factual answering	事実を回答	python	
19		Ad from product description	広告文作成	python	
20		Product name generator	製品名生成	python	
21		TL;DR summarization	注釈を本文に埋め込み要約	python	
22		Python bug fixer	pythonバグ修正	python	
23		Spreadsheet creator	xls, csvファイル等生成	python	
24		JavaScript helper chatbot	javascript作成のQ&A生成	python	
25		ML/AI language model tutor	ML言語モデルのchatbot生成	python	
26		Science fiction book list maker	SF本のリスト生成	python	
27		Airport code extractor	空港の3文字略語表示	python	
28		SQL request	sql文生成	python	
29		Extract contact information	本文から連絡先抽出	python	
30		JavaScript to Python	jsからpython変換	python	

4. Openai事例一覧(その2)

項番	項目	api名等	内容	方式	備考
31	事例	Friend chat	友達とのチャットをエミュレート	python	
32		Mood to color	背景の色を生成	python	
33		Write a Python docstring	プログラム解説	python	
34		Analogy maker	相似形作成	python	
35		JavaScript one line function	jsを1行へ	python	
36		Micro horror story creator	ホラー物語生成	python	
37		Third-person converter	3人称への変換	python	
38		Notes to summary	速記を要約	python	
39		VR fitness idea generator	vrアイデア生成	python	
40		Essay outline	エッセイ要約	python	
41		Recipe creator	食材からレシピ作成	python	
42		Chat	チャット	python	
43		Marv the sarcastic chat bot	チャットボット	python	
44		Turn by turn directions	道路ナビ	python	
45		Restaurant review creator	レストラン評価	python	
46		Create study notes	学習ノート作成	python	
47		Interview questions	インタビュー質問集作成	python	

4. Openai事例

③. Summarize for a 2nd grader

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'  
$ cd /home/pi/Documents/openai/openai_api/examples  
$ cd /home/pi/Documents/openai/openai_api/examples2  
$ python3 api_summar.py
```

- ページ <https://platform.openai.com/examples>
- 内容: openaiを使った実践事例、pythonで提供。英語と日本語版の二つを提供
- 要約

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai_api/examples
```

 英語

```
$ cd /home/pi/Documents/openai/openai_api/examples2
```

 日本語

```
$ python3 api_summar.py
```

Jupiter is the fifth planet from the Sun and the largest in the Solar System. It is a gas giant with a mass one-thousandth that of the Sun, but two-and-a-half times that of all the other planets in the Solar System combined. Jupiter is one of the brightest objects visible to the naked eye in the night sky, and has been known to ancient civilizations since before recorded history. It is named after the Roman god Jupiter.[19] When viewed from Earth, Jupiter can be bright enough for its reflected light to cast visible shadows,[20] and is on average the third-brightest natural object in the night sky after the Moon and Venus.?
に対する回答

英語

Jupiter is a giant planet far away from us in space. It is very bright and can be seen from Earth. It is named after the Roman god Jupiter and has been known since ancient times.

日本語

木星は太陽系で最大の星で、太陽から5番目にあり、太陽系の他の惑星の質量を合わせ。

日本語は途中で切れる。設定は[こちら](#)

4. Openai事例

⑪. Python to natural language

- ページ <https://platform.openai.com/examples>
- 内容: openaiを使った実践事例、pythonで提供。英語と日本語版の二つを提供
- pythonのプログラムを解説

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai_api/examples
```

英語

```
$ cd /home/pi/Documents/openai/openai_api/examples2
```

日本語

```
$ python3 api_py.py
```

def
remove_common_pr
efixの関数を解説

英語

This code removes a common prefix from a dataframe column containing strings. It takes in the dataframe, the prefix to be removed, and a boolean value indicating whether to keep a single whitespace as a prefix. It then removes the prefix from the dataframe column and, if the boolean value is true, adds a single whitespace as a prefix. Finally, it returns the modified dataframe.

日本語

このコードは、指定されたプレフィックスを受け取り、xという名前のデータフレームからそのプレフィックスを削除する関数です。ws_prefixがTrueの場合、単一の空白文字をプレフィックスとして保持します。最後に、変更されたデータフレームを返します

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai_api/examples
```

```
$ cd /home/pi/Documents/openai/openai_api/examples2
```

```
$ python3 api_py.py
```

4. Openai事例

④. Essay outline

- ページ <https://platform.openai.com/examples>
- 内容: openaiを使った実践事例、pythonで提供。英語と日本語版の二つを提供
- エッセイ要約

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai_api/examples
```

 英語

```
$ cd /home/pi/Documents/openai/openai_api/examples2
```

 日本語

```
$ python3 api_essay.py
```

英語

- I. Introduction
 - A. Definition of Nikola Tesla
 - B. Overview of Tesla's life
- II. Tesla's Early Life
 - A. Childhood
 - B. Education
- III. Tesla's Contributions to Technology
 - A. Alternating Current
 - B. Radio
 - C. Robotics
- IV. Tesla's Later Life
 - A. Decline in Popularity
 - B. Death
- V. Conclusion
 - A. Summary of Tesla's Life and Contributions
 - B. Impact of Tesla's Work on Modern Technology

詳細は、原文参照

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'  
$ cd /home/pi/Documents/openai/openai_api/examples  
$ cd /home/pi/Documents/openai/openai_api/examples2  
$ python3 api_essay.py
```

日本語

- I. イントロダクション
 - A. テスラとは
 - B. テスラの貢献
- II. テスラのテクノロジーへの貢献
 - A. 電気自動車の開発
 - B. ロケット技術の進歩
 - C. スペースX
- III. 結論
 - A. テスラの貢献が世界を変えた
 - B. テ

日本語は途中で切れる。設定は[こちら](#)

4. Openai事例

④1. Recipe creator

コマンド入力

```
$ export OPENAI_API_KEY='sk-xxxxx'  
$ cd /home/pi/Documents/openai/openai_api/examples  
$ cd /home/pi/Documents/openai/openai_api/examples2  
$ python3 api_recipe.py
```

- ページ <https://platform.openai.com/examples>
- 内容: openaiを使った実践事例、pythonで提供。英語と日本語版の二つを提供
- 食材からレシピ作成

```
$ export OPENAI_API_KEY='sk-xxxxx'
```

```
$ cd /home/pi/Documents/openai/openai_api/examples
```

 英語

```
$ cd /home/pi/Documents/openai/openai_api/examples2
```

 日本語

```
$ python3 api_recipe.py
```

英語

1. Preheat oven to 375 degrees Fahrenheit.
2. Spread a single layer of Fritos in the bottom of an oven-safe dish.
3. Top the Fritos with chili.
4. Sprinkle the shredded cheese over the chili.
5. Sprinkle the diced onions over the cheese.
6. Bake in preheated oven for 15 minutes.
7. Remove from oven and top with a dollop of sour cream.
8. Serve and enjoy!

詳細は、原文参照

日本語

1. フライパンを中火で温め、チリを加えて炒めます。
2. チリが柔らかくなったら、小さめの角切りの甘い白玉ねぎまたは赤玉ねぎを加えて炒めます。
3. フリトスを加

日本語は途中で切れる。設定は[こちら](#)