

# Open Interpreter

株式会社ビヨンド  
システム開発部 萬代



# 自己紹介

---

萬代陽一

株式会社ビヨンド システム開発部所属

ChatGPT 歴 : 約半年 と数ヶ月

Twitter 始めました → @chaotic\_pg

# 本日のお品書き

- Open Interpreter とは？
- Open Interpreter のメリット・デメリット
- Open Interpreter をつかってみる
- まとめ

# Open Interpreter とは？

ChatGPT の advanced data analysis のオープンソース版という位置付け

チャットで質問した内容に対して、ChatGPT を用いて回答を返す

必要に応じてインターネットからデータを取得したり、解析に必要な処理をコード化して処理することが可能

# Open Interpreter のメリット

ChatPGT に接続してチャットに対する回答を考えるため、ChatGPT ができることは基本的には可能

加えて URL を与えるとインターネット接続してデータを取得したり、データセットと解析方法の指定することでデータ解析を自分で考え行う

必要な python ライブラリなどを自分で揃えたり、実行中のマシン内のデータリソースにアクセスしたりもするため、応用範囲が広い

# Open Interpreter のデメリット

マシン内のデータアクセスやインターネットアクセスなど、セキュリティ面の不安が大きい

思ったより **トークンを消費している** (この資料作るだけで \$40 使いました)

# Open Interpreter をつかってみる

Open Interpreter を使う場合、2種類の方法がある

- ローカル環境で実行する
- クラウド環境で実行する

ローカル環境の場合、直接 PC 上で実行するのは危険なので、Docker など隔離された環境で実行したほうがよい

# Open Interpreter をつかってみる

今回は Google Colaboratory というツールを使ってみます

Google Colaboratory とは、Google が提供している Python の実行環境

Jupyter Notebook のような UI なので、実行結果が残った状態で保存できます

今回のような機械学習のちょっとした実行環境には最適です



# Open Interpreter をつかってみる

Google Colaboratory

<https://colab.research.google.com/>

ぱっと見どこから使うのかわからないですが

メニューのファイルからノートブックを新規追加を選ぶと実行画面に進めます

# Open Interpreter をつかってみる

```
!pip install open-interpreter
```

```
import interpreter
```

```
interpreter.api_key = "[ChatGPT の API キー]"
```

```
interpreter.auto_run = True
```

```
interpreter.chat()
```



これだけ！

# Open Interpreter をつかってみる

switchの売上TOP10 を円グラフに表示してください。またゲーム名をラベルにしてください。 [https://teitengame.com/2022\\_01.html](https://teitengame.com/2022_01.html) からデータを取得してください。データはHTML 内にあるのでスクレイピングしてください。目的のデータは2つ目のテーブルにあります。ゲーム名には日本語を含むので IPAゴシックを導入し表示するようにしてください

とチャットに打ってみる

妙に指示が具体的ですが、ここまで来るのに紆余曲折があったということです

# Open Interpreter をつかってみる

動き出す！

```
Collecting open-interpreter
```

```
  Downloading open_interpreter-0.1.14-py3-none-any.whl (68 kB)
```

```
68.2/68.2 kB 2.8 MB/s eta 0:00:00
```

```
Requirement already satisfied: appdirs<2.0.0,>=1.4.4 in /usr/local/lib/python3.10/dist-packages (from open-interpreter) (1.4.4)
```

```
Collecting astor<0.9.0,>=0.8.1 (from open-interpreter)
```

```
  Downloading astor-0.8.1-py2.py3-none-any.whl (27 kB)
```

```
Collecting git-python<2.0.0,>=1.0.3 (from open-interpreter)
```

```
  Downloading git_python-1.0.3-py2.py3-none-any.whl (1.9 kB)
```

```
Collecting html2image<3.0.0.0,>=2.0.4.3 (from open-interpreter)
```

```
  Downloading html2image-2.0.4.3-py3-none-any.whl (27 kB)
```

```
Collecting inquirer<4.0.0,>=3.1.3 (from open-interpreter)
```

```
  Downloading inquirer-3.1.3-py3-none-any.whl (18 kB)
```

```
Collecting ipykernel<7.0.0,>=6.26.0 (from open-interpreter)
```

```
  Downloading ipykernel-6.26.0-py3-none-any.whl (114 kB)
```

# Open Interpreter をつかってみる(ハイライト)

まず、以下の手順で進めます。

- 1 指定されたURLからHTMLを取得します。
- 2 BeautifulSoupを使用してHTMLを解析し、目的のテーブルデータを取得します。
- 3 データを整形し、ゲーム名と売上を抽出します。
- 4 matplotlibを使用して円グラフを作成します。日本語の表示にはIPAゴシックを使用します。

まずは、指定されたURLからHTMLを取得します。

```
import requests

url = 'https://teitengame.com/2022_01.html'
response = requests.get(url)
response.text
```

# Open Interpreter をつかってみる(ハイライト)

HTMLの取得が成功しました。次に、BeautifulSoupを使用してHTMLを解析し、目的のテーブルデータを取得します。テーブルは2つ目のものなので、すべてのテーブルを取得した後、インデックス1のテーブルを選択します。

```
from bs4 import BeautifulSoup

soup = BeautifulSoup(response.text, 'html.parser')
tables = soup.find_all('table')
target_table = tables[1]
target_table
```

# Open Interpreter をつかってみる(ハイライト)

目的のテーブルの抽出に成功しました。次に、このテーブルからゲーム名と売上を抽出します。テーブルの各行がゲームのデータを表しているので、各行をループしてゲーム名と売上を取得します。ゲーム名は各行の2番目のセルに、売上は最後のセルに格納されています。

```
game_sales = []
rows = target_table.find_all('tr')
for row in rows:
    cells = row.find_all('td')
    if len(cells) > 1:
        game_name = cells[1].text.strip()
        sales = cells[-1].text.strip().replace(',', '')
        game_sales.append([game_name, int(sales)])
game_sales
```

File "<stdin>", line 18, in <module>

ValueError: invalid literal for int() with base 10: '累計\n

販売本数'

# Open Interpreter をつかってみる(ハイライト)

エラーが発生しました。売上データを整数に変換しようとしたときに、'累計\n販売本数'という文字列を整数に変換できないというエラーが発生しました。これは、テーブルのヘッダー行も解析してしまったためです。

この問題を解決するために、売上データを整数に変換する前に、それが実際に数値であることを確認します。数値でない場合は、その行をスキップします。

```
game_sales = []
rows = target_table.find_all('tr')
for row in rows:
    cells = row.find_all('td')
    if len(cells) > 1:
        game_name = cells[1].text.strip()
        sales = cells[-1].text.strip().replace(',', '')
        if sales.isdigit():
            game_sales.append([game_name, int(sales)])
game_sales
```



# Open Interpreter をつかってみる(ハイライト)

盛り上がってきたところでエラーが解決できず無限ループへ...

キーを持たない型に対してどうしてもハッシュっぽくアクセスしてしまう...

# Open Interpreter をつかってみる (Docker 編)

なんやかんやしてるうちに colab を使いすぎて

うまく動かなくなってきたので Docker に切り替え...

ChatGPT を使う分には GPU とかいらないので

実行環境は何でもいいと思います (適当)

# Open Interpreter をつかってみる (Docker 編)

<https://github.com/karaage0703/open-interpreter-docker>

python が動作する Docker が必要なだけなので

Linux の Docker コンテナなら大抵要件はクリアしているはずですが

pip する手間も惜しんで上記のリポジトリから Dockerfile を拝借しました

# Open Interpreter をつかってみる (Docker 編)

docker exec の時に OpenAI の API キーを環境変数として渡せば ChatGPT を選択  
なければ Mistral 7B という LLM を使うようです

Mistral 7B という LLM もすげえやつらしいですが、マシンパワーを必要としてくるため今回は却下で

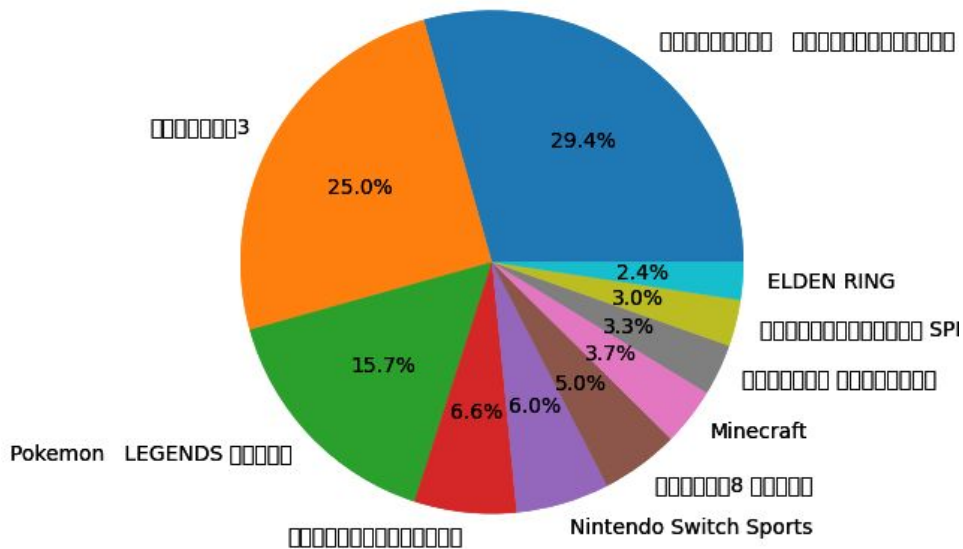
# Open Interpreter をつかってみる (Docker 編)

ビルドした後 `docker run --it` でコンテナを起動

`interpreter -y` で Open Interpreter を起動

入力受付が始まるので、チャットする、という流れ

# Open Interpreter をつかってみる (Docker 編)



# Open Interpreter をつかってみる (Docker 編)

```
{
```

```
"ポケットモンスター スカーレット・バイオレット": 4338931,
```

```
"スプラトゥーン3": 3687814,
```

```
"Pokemon LEGENDS アルセウス": 2314806,
```

```
"星のカービィ ディスカバリー": 969047,
```

```
"Nintendo Switch Sports": 887722,
```

```
"マリオカート8 デラックス": 742018,
```

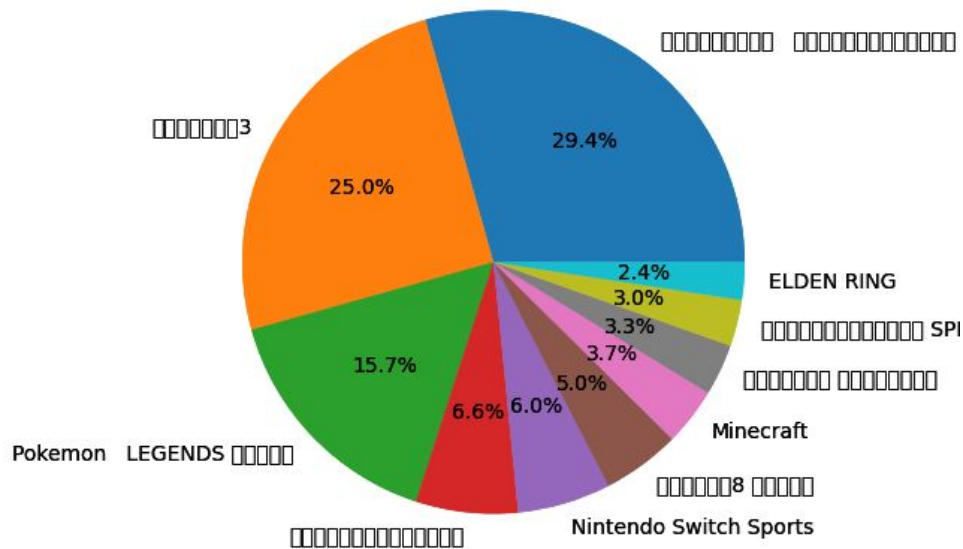
```
"Minecraft": 548415,
```

```
"マリオパーティ スーパースターズ": 485594,
```

```
"大乱闘スマッシュブラザーズ SPECIAL": 441053,
```

```
"ELDEN RING": 356711
```

```
}
```



# Open Interpreter ができること

- 写真、動画、PDF などの作成や編集
  - 指定のフォルダにある動画の字幕を全部付ける、など
- Chrome ブラウザの制御とリサーチ作業
- 大規模なデータセットのプロット、クリーニング、分析
- 等々

引用

[https://github.com/KillianLucas/open-interpreter/blob/main/docs/README\\_JA.md](https://github.com/KillianLucas/open-interpreter/blob/main/docs/README_JA.md)



# まとめ

- Open Interpreter をご紹介しました
- 抽象的な質問で回答を考えてくれる ChatGPT の持ち味にコードを実行する柔軟性を組み合わせたとても良いツールです
- 実行環境によってはセキュリティのリスクも存在するので要注意
  - Docker や Google Colaboratory がおすすめ
- Google Colaboratory を使えば安価に試せる
  - ブラウザでページを開けばすぐできる！
  - 自前で GCP に専用コンテナを立てて処理することも可能（GPU 付きインスタンスは高い！）
- ChatGPT を使うなら Docker もあり
  - colab の再接続に疲れたら



サーバのことは  
全部  
丸投げ 

ご清聴ありがとうございました！



API  
開発も  
やってます 

