OpenAI API with Python Introduction

Setup

API Key from OpenAI

```
https://platform.openai.com/organization/api-keys
```

```
python3 -m pip install openai
```

```
python3 -m pip install requests
```

Tested on MacOS. Should work fine on Ubuntu.

About Me Lab

This lab allows you to ask OpenAI questions and it will answer those questions in relation to where you are. We use the ip-api.com REST API to find out current location based on IP Address, and then we inject that into out OpenAI request.

lab-about-me.py

```
from openai import OpenAI
from requests import get
import os

client = OpenAI()

nationality = get('http://ip-api.com/json/').json() #Geo Data from IP
Address API
nationality = nationality['country']

os.system('clear') #Windows = 'cls'

while True:
    query = input('Question: ')
    os.system('clear')
    completion = client.chat.completions.create(
    model='gpt-4o-mini',
    messages=[
        {"role": "system", "content": "You Provide Answers to Question."},
        {"role": "user", "content": f"I am from {nationality}"},
        {"role": "user", "content": "Answer in 10 words or less"},
        {"role": "user", "content": query}
    ]
    )
    print(query)
    print(completion.choices[0].message.content)
```

Autoblog

This script allows the user to ask OpenAI to write blog posts and then saves those blog posts to an HTML file.

lab-auto-blog.py

```
from openai import OpenAI
import os

client = OpenAI()

os.system('clear') #Windows = 'cls'

while True:
    title = input('Title for Blog Post: ').title()
    os.system('clear')
    completion = client.chat.completions.create(
    model='gpt-4o-mini',
    messages=[
        {"role": "system", "content": "You are a blogger."},
        {"role": "user", "content": "Write a 100 word essay"},
        {"role": "user", "content": title}
    ]
    )

    story = completion.choices[0].message.content

    with open('blog.html', 'a') as file:
        file.write(f'<h2>{title}</h2>')
        file.write(f'<p>{story}</p>')

    print(title)
    print(story)
```

Image Gallery

This script asks Dall E to create an image. We inject a bias into that request to show how this can automatically be done so that the user will not see it.

We then download the image and save it using the 'wb' file mode. We use the timestamp for the image as the name for it, and then we write the image name within an src tag to create an image gallery.

```
from openai import OpenAI
from requests import get
import os

bias = 'add a bunny'

def ai(query):
    client = OpenAI()
    response = client.images.generate(
        model="dall-e-3",
        prompt=query,
        n=1,
        size="1024x1024"
    )

    return response

while True:
    query = input('Image to Create: ')
    query = f'{query} {bias}'
    os.system('clear')
    response = ai(query)
    pic_name = f'{response.created}.png'

    response_image = get(response.data[0].url)
    with open(pic_name, 'wb') as file:
        file.write(response_image.content)

    with open('gallery.html', 'a') as gallery:
        gallery.write(f'<img style="height:200px; width:auto;" 
src="{pic_name}">')

    print(query)
    print(response_image)
```