

ChatGpt behind the scene:

ChatGpt is one of the product of Openai company.

Products:-

for code generation it has >> Codex

for image generation it has >> Dall-e

for text generation it has >> ChatGpt

- Query which we enter in chatgpt is called Prompt.
- Chatgpt is based on generative approach and it use transformer algorithm which was introduced by Google in 2017.
- chatgpt use modles behind the scene which is made by openai.
- GPT-3 >> it is a set of models that can understand and generate natural language(*Generative pre-trained transformer version 3*).
- **text-davanci-003** model comes under gpt-3 and it is slowest than other model of gpt-3 because it takes lot of processing power for searching due to heavy dataset.
- GPT-3 has **175 billion** parameters(token) and it takes **800 GB** to store these tokens.
- what is token?
 - ☐ we can say word is called token but sometimes when word is long then it has more than one token and every token has unique id which is depends on model to model.
 - ☐ example :- **who is carpenter**
 - it has 5 token >> who, is, car, pan, ter
 - each token has unique id

Now we use Chatgpt through python and here we use new model text-davanci-003 of gpt-3 and it can take or generate maximum 2000 token.

>> **pip install openai**

```
● PS D:\coding> pip install openai
Collecting openai
  Downloading openai-0.27.0-py3-none-any.whl (70 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 70.1/70.1 kB 643.3 kB/s eta 0:00:00
⊗ Collecting aiohttp
  Downloading aiohttp-3.8.4-cp310-cp310-win_amd64.whl (319 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 319.8/319.8 kB 2.0 MB/s eta 0:00:00
Collecting tqdm
  Downloading tqdm-4.65.0-py3-none-any.whl (77 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 77.1/77.1 kB 2.2 MB/s eta 0:00:00
Requirement already satisfied: requests>=2.20 in c:\users\shiva\appdata\local\packages\pythonsoftwarefound
kages\python310\site-packages (from openai) (2.28.1)
● Requirement already satisfied: charset-normalizer<3,>=2 in c:\users\shiva\appdata\local\packages\pythonsof
\local-packages\python310\site-packages (from requests>=2.20->openai) (2.1.0)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\shiva\appdata\local\packages\pythonsoftwaref
-packages\python310\site-packages (from requests>=2.20->openai) (2022.6.15)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\shiva\appdata\local\packages\pythonsoftwa
cal-packages\python310\site-packages (from requests>=2.20->openai) (1.26.10)
Requirement already satisfied: idna<4,>=2.5 in c:\users\shiva\appdata\local\packages\pythonsoftwarefoundat
ges\python310\site-packages (from requests>=2.20->openai) (3.3)
Collecting attrs>=17.3.0
  Downloading attrs-22.2.0-py3-none-any.whl (60 kB)
    ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 60.0/60.0 kB 3.1 MB/s eta 0:00:00
Collecting async-timeout<5.0,>=4.0.0a3
  Downloading async_timeout-4.0.2-py3-none-any.whl (5.8 kB)
Collecting aiosignal>=1.1.2
  Downloading aiosignal-1.3.1-py3-none-any.whl (7.6 kB)
Collecting frozenlist>=1.1.1
  Downloading frozenlist-1.3.3-cp310-cp310-win_amd64.whl (33 kB)
Collecting yarl<2.0,>=1.0
  Downloading yarl-1.8.2-cp310-cp310-win_amd64.whl (56 kB)
```

```
import openai
while True:
    prompt_value=input("Prompt :>> ")
    if(prompt_value=="exit"):          #if you want to close Press "exit"
        break
    else:
        # enter api key here for authentication
        openai.api_key="key_vaalue"
        output=openai.Completion.create(prompt=prompt_value,model="text-davinci-003",
max_tokens=2000)
        output_text=output["choices"][0]["text"]
        print(output_text)
```

```
connect.py U X
python > oppenai > connect.py > ...
1 import openai
2 while True:
3     prompt_value=input("Prompt :>> ")
4     if(prompt_value=="exit"):          #if you want to close Press "exit"
5         break
6     else:
7         # enter api key here for authentication
8         openai.api_key="key_vaalue"
9         output=openai.Completion.create(prompt=prompt_value,model="text-davinci-003",max_tokens=2000)
10        output_text=output["choices"][0]["text"]
11        print(output_text)
12
```

How chatgpt charge?

Chatgpt charges based on how many token it takes or generates so always gives minimum token and ask to generate minimum token.

example:- **what is computer describe in two lines ?**

here it generates minimum token so we can save our money.

```
connect.py U X
python > oppenai > connect.py > ...
1 import openai
2 while True:
3     prompt_value=input("\nPrompt :>> ")
4     if(prompt_value=="exit"):          #if you want to close Press "exit"
5         break
6     else:
7         # enter api key here for authentication
8         openai.api_key="key_vaalue"
9         output=openai.Completion.create(prompt=prompt_value,model="text-davinci-003",max_tokens=2000)
10        output_text=output["choices"][0]["text"]
11        print(output_text)
12
```

PROBLEMS OUTPUT DEBUG CONSOLE **TERMINAL** GITLENS SQL CONSOLE

PS D:\coding>
PS D:\coding> python -u "d:\coding\python\oppenai\connect.py"

Prompt :>> what is computer describe in two lines ?

Computer is an electronic device used to process data, and perform calculations and operations, based on instructions provided by the user. It consists of hardware and software components to store, retrieve, manipulate and communicate data.

Prompt :>>

python > openai > connect.py > ...

```
1 import openai
2 while True:
3     prompt_value=input("\nPrompt :>> ")
4     if(prompt_value=="exit"):          #if you want to close Press "exit"
5         break
6     else:
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL GITLENS SQL CONSOLE

Code + v

Prompt :>> my name is shivam saini

Hi Shivam, nice to meet you.

Prompt :>> who is saini

Saini is a social group of people broadly found mainly in northern India. They are also known as Shoorsaini as several lineage rigin to Rama, the epic hero of the Ramayana. The Sainis are primarily land-owning agriculturalists and are related to the Ra e in India.