



## Getting Started

### Installation (CMD, Terminal, Shell, Powershell)

```
pip install openai
# or
pip3 install openai
```

### First Prompt

```
import os
import openai
```

```
# Create, copy, and paste your API key here:
openai.api_key = "sk-123456789"
```

```
response = openai.Completion.create(
    model="text-davinci-003",
    prompt="2+2=",
    temperature=0, max_tokens=10)
```

## Using GPT-4

```
system = 'You only reply in emojis!'
prompt = 'Who are you?'
```

```
res = openai.ChatCompletion.create(
    model="gpt-4",
    messages=[
        {"role": "system",
         "content": system},
        {"role": "user",
         "content": prompt}
    ],
    max_tokens=100,
    temperature=1.2)

print(res['choices'][0]['message']['content'])
# Answer:
```

## JSON Output Format

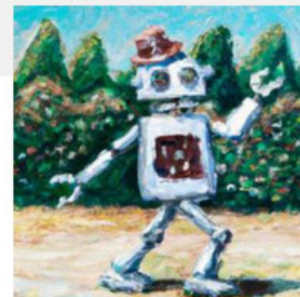
```
{
  "choices": [
    {
      "finish_reason": "stop",
      "index": 0,
      "logprobs": null,
      "text": "4\n\n2+2=4"
    }
  ],
  "created": 1682409707,
  "id": "cmpl-797uNKSjEKE5cMlod1MeXkueIetkC",
  "model": "text-davinci-003",
  "object": "text_completion",
  "usage": {
    "completion_tokens": 8,
    "prompt_tokens": 4,
    "total_tokens": 12
  }
}
```

## Generating Images Programmatically with DALL-E

```
prompt = "An oil painting of a dancing robot in the style of Monet"
```

```
response = openai.Image.create(
    prompt=prompt,
    n=1,
    size="256x256")
```

```
url = response["data"][0]["url"]
print(url)
# https://...
```



Resolution	Price
1024x1024	\$0.020 / image
512x512	\$0.018 / image
256x256	\$0.016 / image

## Example Sentiment Analysis

```
prompt = """Do sentiment analysis on the following text. Text: 'Oh, I just adore how the sun shines so brightly at 5 a.m., waking me up every single morning!'"""
```

```
response = openai.Completion.create(
    engine="text-davinci-003",
    prompt=prompt,
    max_tokens=200,
    n=1,
    stop=None,
    temperature=0.5)
```

```
sentiment = response.choices[0].text.strip()
print(sentiment)
# Sentiment: Positive
```

## Arguments Python OpenAI API Call

- ✓ **model**: Specifies the model version, e.g., 'gpt-4.0-turbo'.
- ✓ **prompt**: The input text for the model to process (e.g., question)
- ✓ **max\_tokens**: Maximum tokens in the response. Roughly equates to number of words.
- ✓ **temperature**: Controls output randomness (0 to 1). Higher value leads to more random replies.
- ✓ **top\_p**: Nucleus sampling strategy (0 to 1). Model will only consider subset of tokens whose probability exceeds top\_p.
- ✓ **n**: Number of independent completions to explore.
- ✓ **stream**: Use streaming mode (True or False) to return results incrementally (e.g., for real-time apps).
- ✓ **echo**: Include input prompt in output (True or False).
- ✓ **stop**: Stopping sequence(s) for generation (string or list of strings).
- ✓ **presence\_penalty**: Penalizes similar tokens in output.
- ✓ **frequency\_penalty**: Penalizes frequent tokens in output.

