



DIALOG



# OpenAI Demo

*Morten Kromberg, CTO  
(Code and examples by Brian Becker)*

# What is OpenAI?



Wikipedia says:

**OpenAI** is an American [artificial intelligence](#) (AI) research organization founded in December 2015, researching artificial intelligence with the goal of developing "safe and beneficial" [artificial general intelligence](#), which it defines as "highly autonomous systems that outperform humans at most economically valuable work".<sup>[4]</sup> As one of the leading organizations of the [AI boom](#),<sup>[5][6][7]</sup> it has developed several [large language models](#), advanced image generation models, and previously, released open-source models.<sup>[8][9]</sup> Its release of [ChatGPT](#) has been credited with starting the AI boom.<sup>[10]</sup>

# What is an LLM?



ChatGPT

Wikipedia says:

A **large language model (LLM)** is a computational **model** notable for its ability to achieve general-purpose language generation and other **natural language processing** tasks such as **classification**. Based on **language models**, LLMs acquire these abilities by learning statistical relationships from text documents during a computationally intensive **self-supervised** and **semi-supervised** training process.<sup>[1]</sup> LLMs can be used for text generation, a form of **generative AI**, by taking an input text and **repeatedly predicting the next token or word.**<sup>[2]</sup>

# Getting Started

- ◆ Follow the instructions at <https://platform.openai.com/docs/quickstart>
- ◆ You need an API Key
- ◆ Do not store it in code, and do not lose it!
  - ◆ I store it in an environment variable
- ◆ You will need to pay some money
  - ◆ (I paid \$10 + VAT)

# The OpenAI Class

- ✦ Written by Brian Becker in the last month
- ✦ Will soon move to separate repository [github.com/dyalog/openai](https://github.com/dyalog/openai)
- ✦ Demonstrates
  - ✦ Chat with LLM
  - ✦ Audio Manipulation
  - ✦ Image Creation



# General Usage

3 sub-classes, Audio, Chat and Image, each of which has methods that will create an instance:

1. Create an instance of the endpoint with most common parameters. Example:

```
i ← OpenAI.Image.Create parameters
```

2. Set any additional parameters if needed
3. Run: `r ← i.Run`
4. If ~ `r.IsOK`
  - ◆ Check `r.Data`

```
z.IsOK
```

```
0
```

```
□json z.Data
```

```
{"error":{"code":"insufficient_quota",  
"message":"You exceeded your current quota,  
please check your plan and billing details.  
For more information on this error, read the docs:  
https://platform.openai.com/docs/guides/error-codes/api-errors.",  
"param":null,"type":"insufficient_quota"}}
```

Translation: Please Enter Your Credit Card #

# Chat.Completion

"Completion" means continue a conversation with an "Assistant"

```
i ← OpenAI.Chat.Completion System User  
or i ← OpenAI.Chat.Completion System  
or i ← OpenAI.Chat.Completion ''
```

System describes the assistant.

User is the user's initial input.

```
i ← OpenAI.Chat.Completion 'you are a sarcastic assistant' 'how much is 2+2'
```



# OpenAI.Audio Methods

Method	Description	Constructor Arguments
Speech	Create Audio	Text OutFile Voice
Transcription	Create Text from Audio	AudioFile OutFile
Translate	Translate Audio to English	AudioFile OutFile

# Audio Examples

```
txt←'Suomi on kaunis maa, ja ihmiset ovat hyvin ystävällisiä.'  
s←OpenAI.Audio.Speech txt  
s.OutFile←'/tmp/finnish.wav' 1 1 specifies to overwrite  
s.Run.IsOK
```

```
]open c:\tmp\finnish.wav
```



```
ts←OpenAI.Audio.Transcription '/tmp/finnish.wav'  
ts.Run.IsOK  
ts.Response.Data.text
```

Suomi on kaunis maa ja ihmiset hyvin ystävällisiä.

```
tl←OpenAI.Audio.Translation '/tmp/finnish.wav'  
tl.Run.IsOK  
tl.Response.Data.text
```

Finland is a beautiful country and people are very friendly.

# OpenAI.Image Methods

Method	Description	Constructor Arguments
Create	Create Image	<code>prompt</code> <code>OutFile</code> <code>style</code>
Edit	Modify Image	<code>ImageFile</code> <code>prompt</code>
Vary	Create variants of Image	<code>ImageFile</code> <code>n</code>

`Prompt`  
`style`

inspiration used by the AI to generate or modify the image  
'vivid' (the default) or 'natural'

`n`

the number of images to generate from 1 (the default) to 10

`response_format`

'url' (the default, valid for 60 mins) or 'b64\_json'

`size`

'256x256', '512x512', or '1024x1024' (the default)

# Image Example

```
i←OpenAI.Image.Create 'a giraffe snowboarding'  
i.size←'512x512'  
i.n←3  
i.Run.IsOK  
i.Show
```

