AI Agents & LLM

Resources & Links : Link1

If you want to play around with AI models that are open source in nature : Ollama allows you to run those defined models locally on your computer.

Once you run Ollama on your cmd , you can run it locally on the localhost : http://localhost:11434/

Let's understand how to build a commercial project through an open source model we fetch from Ollama.

- LLama 3.2
- GPT-OSS 3.B

Example 1 : Let's build a Website Summarizer that summarizes the content with AI : Link

- Writing a program that takes a website url .
- Scrapes the content from the website.
- Summarizes the content using the ChatGPT model (Frontier Models)

What are Frontier Models ?

- Frontier models are meant to be ***closed-source in nature*** such as (ChatGPT, Claude and many more).
- They are the highest scale of the models.

Closed-Source Frontier Models :

- GPT from OpenAI
- Claude from Anthropic
- Gemini from Google.
- Command R from Cohere.
- Perplexity.

Open-Source Frontier Models :

- Llama from Meta
- Mixtral from Mistral
- Qwen from Alibaba Cloud
- Gemma from Google

- Phi from Microsoft.

Three ways to use models :

- **Chat Interfaces :** ChatGPT
- **Cloud APIs :**
  - Frameworks like LangChain
  - Managed AI cloud services :
    - Amazon BedRock
    - Google Vertex
    - Azure ML
- **Direct inference :**
  - With the huggingFace Transformers library
  - With Ollama to run locally.

Frontier models & End-user UIs :

- **OpenAI :**
  - Models : GPT, o1
  - Chat: ChatGPT
- **Anthropic** :
  - Models : Claude
  - Chat : Claude
- **Google** :
  - Models : Gemini
  - Chat : Gemini Advance
- **Cohere** :
  - Command R +
  - Command R +
- **Meta** :
  - Llama
  - Meta.AI
- **Perplexity** :
  - Models : Perplexity
  - Search : Perplexity

Pros of such Frontier Models :

- **Synthesizing information :**
  - Answering the question in depth with a structured, well researched answer and often including a summary.
- **Fleshing out a skeleton :**

    - From a couple of notes, building out a well crafted email, or a blog post and iterating on it with you until perfect.
- **Coding :**
    - The ability to write and debug code is remarkable, far overtaking Stack Overflow as the resource for engineers.

Limitations of Frontier models :

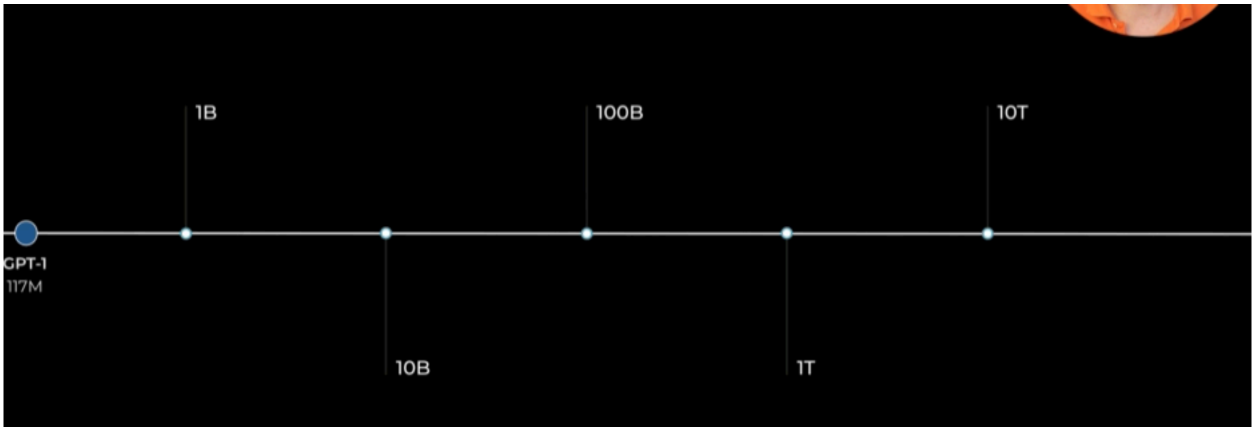
- **Specialized domains** : Most are not PhD level, but closing in.
- **Recent events** : Limited knowledge beyond training cut-off date.
- **Can confidently make mistakes :** Some curious blindspots.

Number of parameters in models (log scale) :

- **Parameters** == Are also defined as weights.

Eg :-

- **GPT-1** : 117 Million parameters.
- **GPT-2** : 1.5 Billion parameters.
- **GPT-3** : 175 Billion parameters.
- **GPT-4** : 1.76 Trillion parameters.



Tokens :

- **Tokens are individual units which get passed into a model.**

A] In the early days, neural networks were trained at the character level (character by character). Small vocab, but expects too much from the network.

- *Predict the next character in this sequence.*

B] Then later, the models were trained based on each possible word.

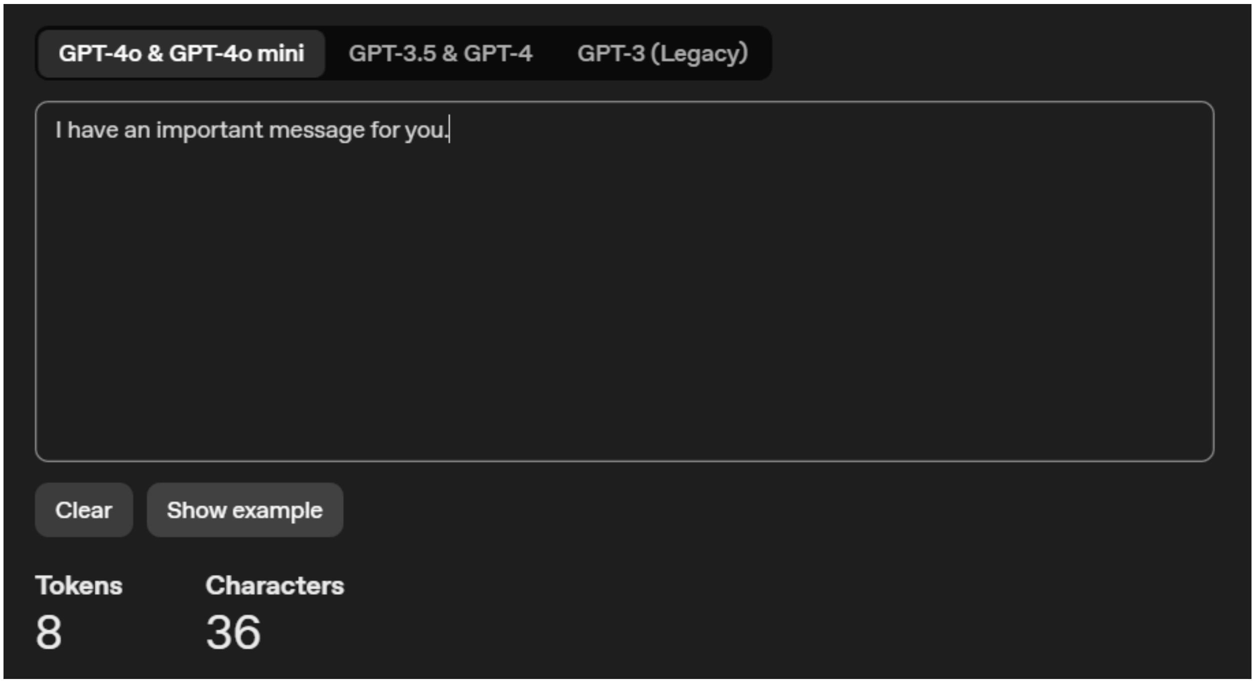
- *Predict the next word in this sequence.*
- Much easier to learn from, but leads to enormous vocabs with rare words omitted.

C] The breakthrough was to work with chunks of words, called “tokens” (During the initial phases of GPT)

- A middle ground; manageable vocab, and useful information for the neural network.
- In addition, it elegantly handles word stems.
- ***Chunks (Tokens)*** → sometimes have a complete word or sometimes have a partial incomplete word.

And train the model in such a way where it takes a series of tokens and outputs tokens based on the tokens that were passed in the model already.

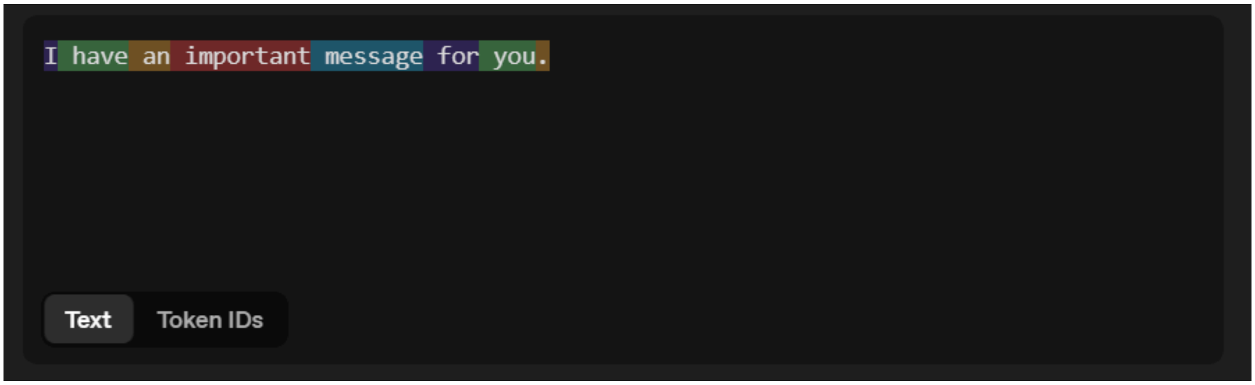
Example : https://platform.openai.com/tokenizer.



I have an important message for you.

Tokens
8

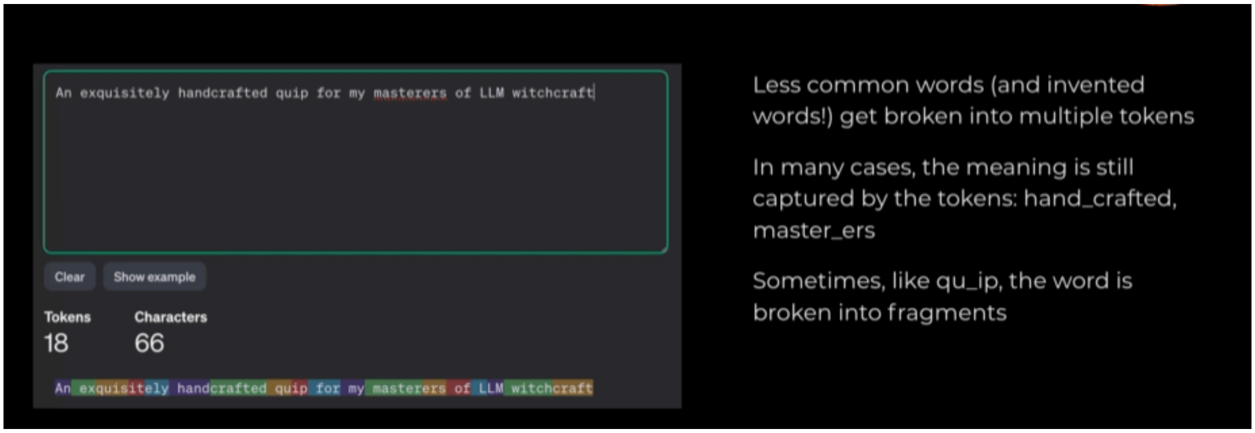
Characters
36



I have an important message for you.

- For common words, 1 word maps to 1 token + gap between words is also included as part of the word / token.

Example 2 :



```
An exquisitely handcrafted quip for my masterers of LLM witchcraft
```

Tokens 18

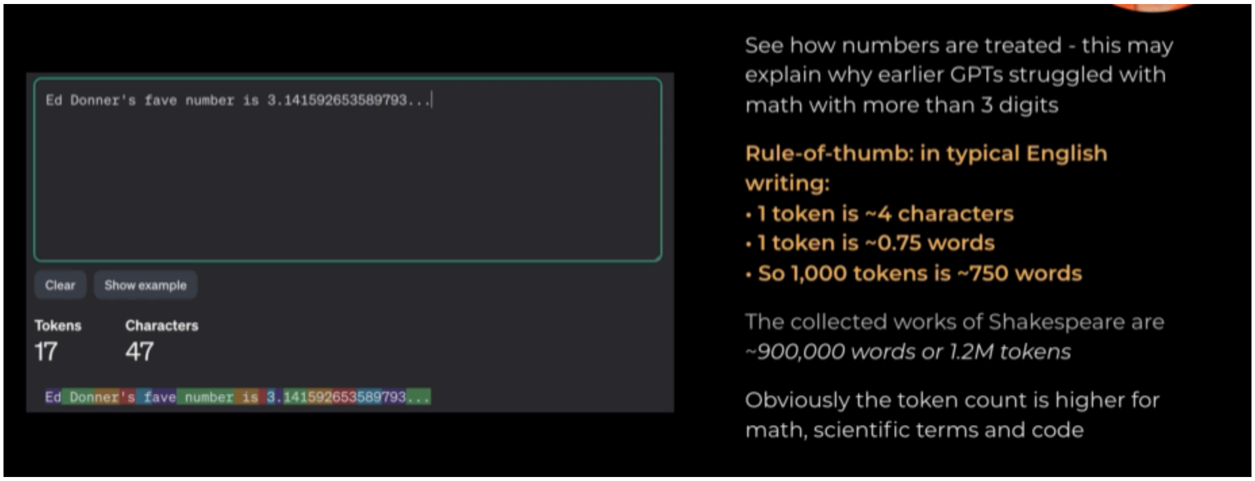
Characters 66

```
An exquisitely handcrafted quip for my masterers of LLM witchcraft
```

Less common words (and invented words!) get broken into multiple tokens

In many cases, the meaning is still captured by the tokens: hand_crafted, master_ers

Sometimes, like qu_ip, the word is broken into fragments



```
Ed Donner's fave number is 3.141592653589793...
```

Tokens 17

Characters 47

```
Ed Donner's fave number is 3.141592653589793...
```

See how numbers are treated - this may explain why earlier GPTs struggled with math with more than 3 digits

Rule-of-thumb: in typical English writing:

- **1 token is ~4 characters**
- **1 token is ~0.75 words**
- **So 1,000 tokens is ~750 words**

The collected works of Shakespeare are *~900,000 words or 1.2M tokens*

Obviously the token count is higher for math, scientific terms and code

Context Window :

- Max number of tokens that the model can consider when generating the next token.
- Includes the original input prompt, subsequent conversation, the latest input prompt and almost all the output prompt. ***(It forms a huge long prompt string when the user enters the message to the model + adds up other conservation prompts as the user moves deep into the conversations).***
- It governs how well the model can remember references, content and context.
- ***Particularly important for multi-shot prompting where the prompt includes examples, or for long conversations.***
- *Or questions on the complete works of Shakespare !*
- ***Example : Some models are trained to handle 1.2M tokens in just a single prompt (Daamnn !!!!)***

API costs:

- Chat interfaces typically have Pro plans with a monthly subscription. Rate-limited, but no per-usage charge.
- APIs typically have no subscription, but charge per API call.
- The cost is based on the number of input tokens and the number of output tokens.

To compare different frontier models on standard benchmarks, Link

One-Shot Prompting :- It helps us in dealing with the nuanced tasks by providing one example and once it's trained on that example, it starts giving us structured outputs as desired.

- This is an excellent use case for an LLM, because it requires nuanced understanding.

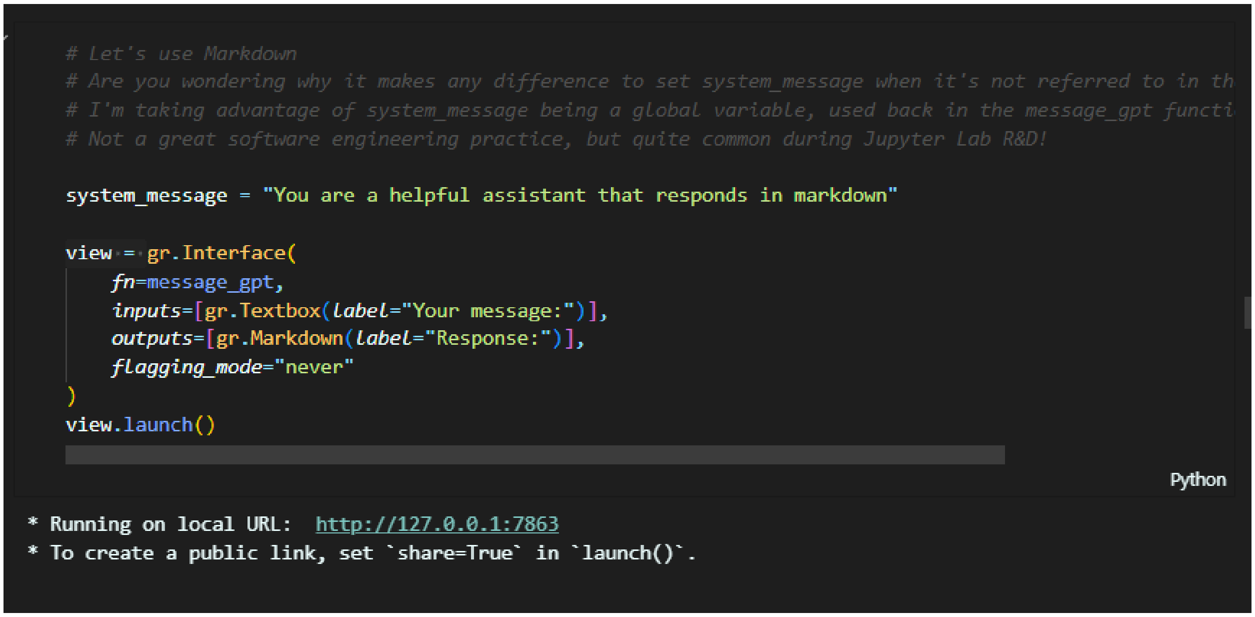
Structured Outputs → Forces the LLM to respond in a particular way.

Adversarial Conversation between ChatBots :

- Configuring two chatbots in mind with the API , we can conduct real-time conversation between two or more models.

Gradio UI :

- *If you are looking for a way to introduce a UI which interacts with the frontier LLM models easily, Gradio python library helps you with it.*

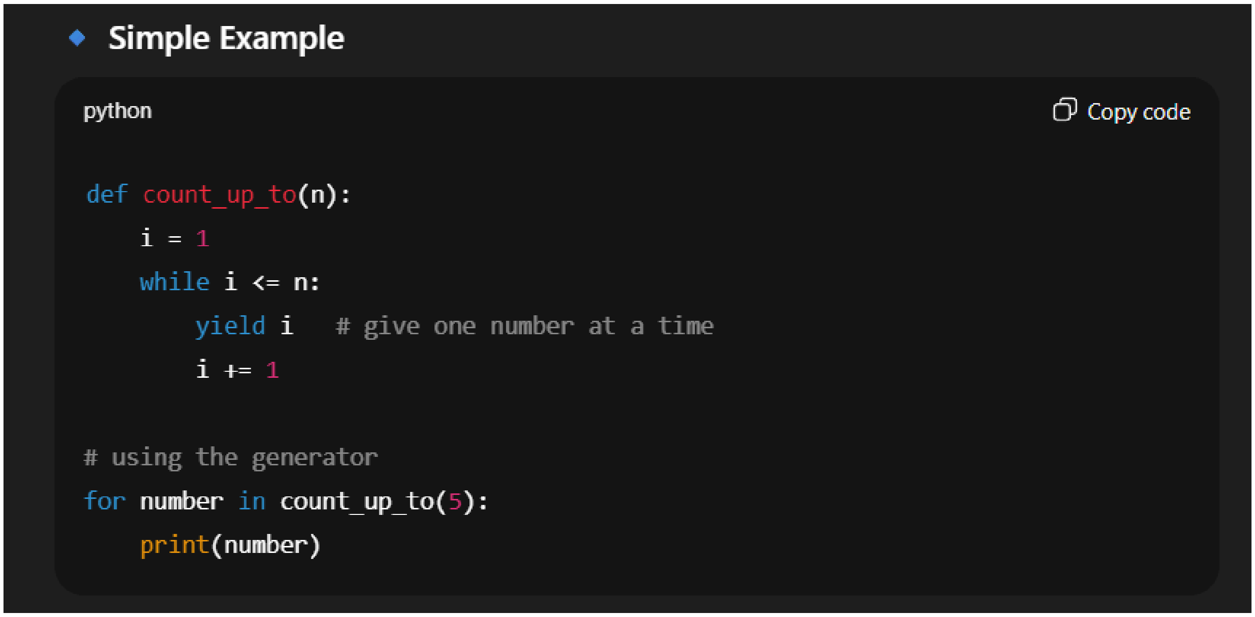


```python
# Let's use Markdown
# Are you wondering why it makes any difference to set system_message when it's not referred to in th
# I'm taking advantage of system_message being a global variable, used back in the message_gpt functi
# Not a great software engineering practice, but quite common during Jupyter Lab R&D!

system_message = "You are a helpful assistant that responds in markdown"

view = gr.Interface(
    fn=message_gpt,
    inputs=[gr.Textbox(label="Your message:")],
    outputs=[gr.Markdown(label="Response:")],
    flagging_mode="never"
)
view.launch()
```

```
* Running on local URL:  http://127.0.0.1:7863
* To create a public link, set `share=True` in `launch()`.
```

yield in Python :

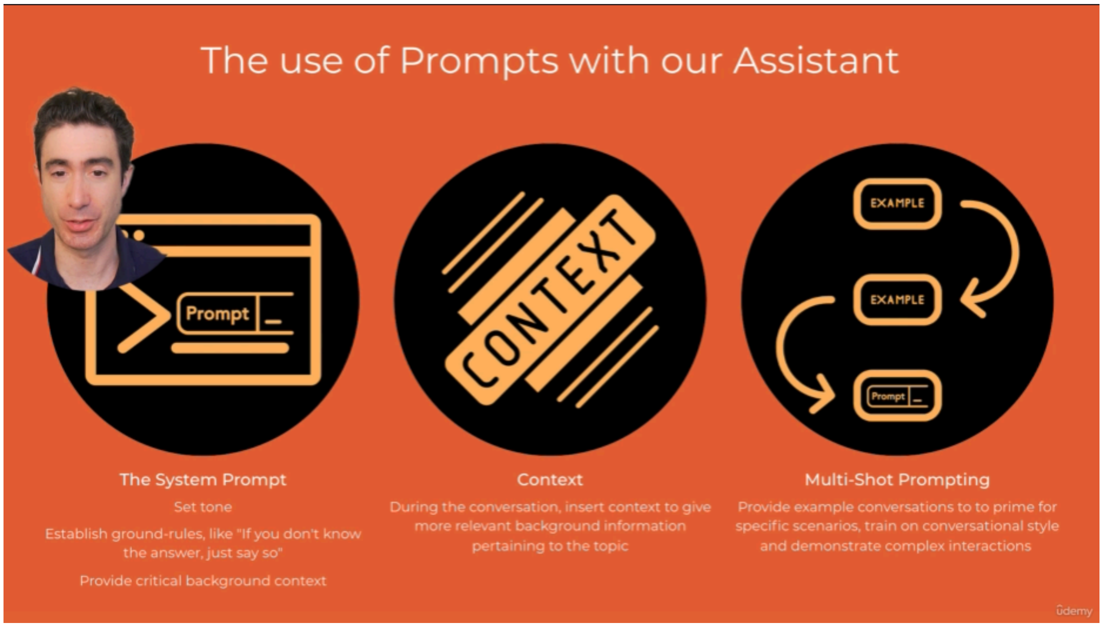
- *It's used **inside a function to turn into a generator***
- Instead of returning from a function , we use the keyword ***yield = forms a generator fn which yields each element one by one instead of all at once.***



◆ Simple Example

```python
def count_up_to(n):
    i = 1
    while i <= n:
        yield i   # give one number at a time
        i += 1

# using the generator
for number in count_up_to(5):
    print(number)
```

Use of Prompts with our Assistant :



The use of Prompts with our Assistant

The System Prompt

Set tone

Establish ground-rules, like "If you don't know the answer, just say so"

Provide critical background context

Context

During the conversation, insert context to give more relevant background information pertaining to the topic

Multi-Shot Prompting

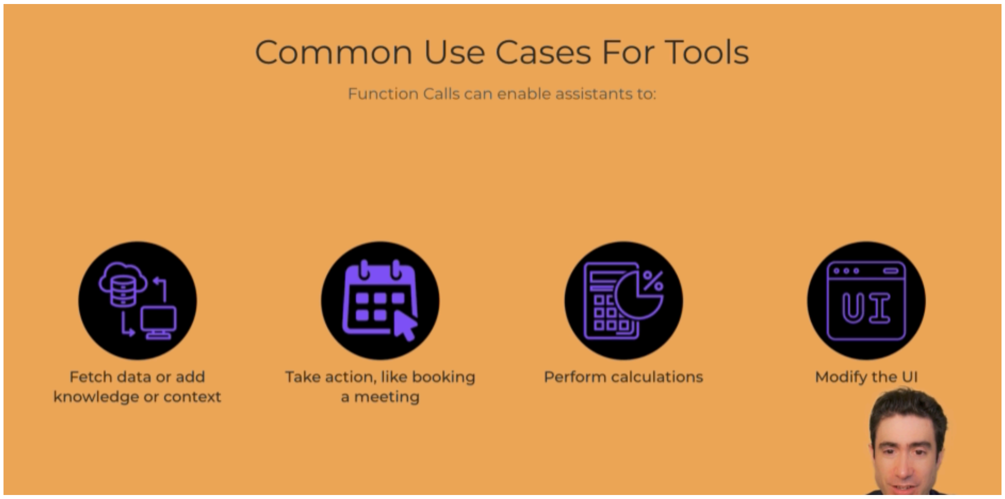
Provide example conversations to to prime for specific scenarios, train on conversational style and demonstrate complex interactions

RAG :- It helps you in finding extra information that's relevant to the prompt, and adding it into the context of the message sent to the LLM.

AI Tools :

- **Allows frontier models to connect with external functions**
  - Richer responses by extending knowledge
  - Ability to carry out actions within the application.
  - Enhanced capabilities, like calculations.
- **How it works :**
  - In a request to the LLM, specify available tools.
  - The reply is either Text, or a request to run a tool.
  - We run the tool and call the LLM with the results.

Common Examples of AI Tools :



Common Use Cases For Tools

Function Calls can enable assistants to:

Fetch data or add knowledge or context

Take action, like booking a meeting

Perform calculations

Modify the UI

What are Agents :

- Software entities that can autonomously perform tasks.

Common characteristics of an Agent :

- Autonomous
- Goal-oriented
- Task specific

Designed to work as part of an Agent Framework to solve complex problems with limited human involvement.

- Memory / persistence.
- Decision-making / orchestration.
- Planning capabilities.
- Use of tools; potentially connecting to databases or the internet.

What are we about to do ?

- **Image Generation** : Use the OpenAI interface to generate images.
- **Make Agents :** Create Agents to generate sound and images for our store.
- **Make an Agent Framework** : Teach our AI Assistant to speak and draw.

What's an Agent AI :

1. Breaking a complex problem into smaller steps, with multiple LLMs carrying out specialized tasks
2. The ability for LLMs to use Tools to give them additional capabilities
3. The 'Agent Environment' which allows Agents to collaborate
4. An LLM can act as the Planner, dividing bigger tasks into smaller ones for the specialists
5. The concept of an Agent having autonomy / agency, beyond just responding to a prompt - such as Memory

HuggingFace Platform :

- The ubiquitous platform for LLM engineers :
  - **Models :** Over 800,000 Open Source models of all shapes and sizes.
  - **Datasets :** A treasure trove of 200,000 datasets.
  - **Spaces** : Apps, many built in Gradio, including leaderboards.

It's gives access to multiple libraries :

- **Hub** : Library allows you to login to a hugging face and both download and upload things like data sets and models.
- **DataSets** : Gives immediate access to data repositories in HuggingFace.
- **Transformers** : Provides wrapper code around LLMs that follow the transform architectures.
- **PEFT (Parameter Efficient Fine Tuning)** : Allows us to train LLMs without needing to work with all of the billions parameters.
- **TRL (Transformer Reinforcement Learning)** : Ability to do things like reward modelling, supervised fine tuning.
- **Accelerate** : It allows both training and inference to run at scale in an efficient, adaptable way.