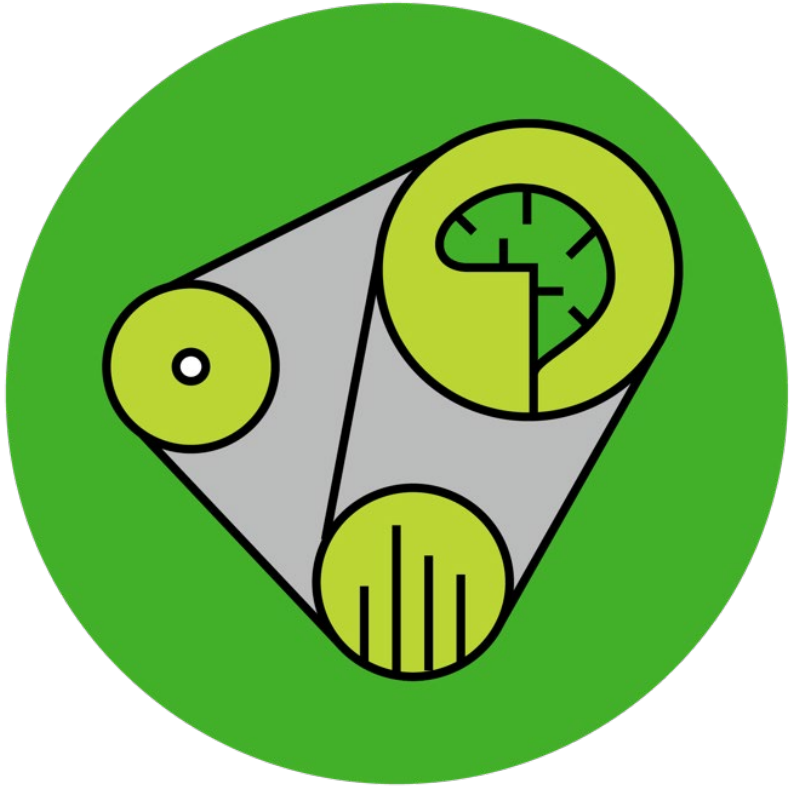
Deloitte.



AI Guild | GenAI Practicum

Session 2: The Art of Prompt Engineering

March 2025

Welcome to the AI Guild GenAI Practicum

Introduction

The Instructors

Niranjan Kumar
USI

Mahendra Rathore
USI

Week Agenda

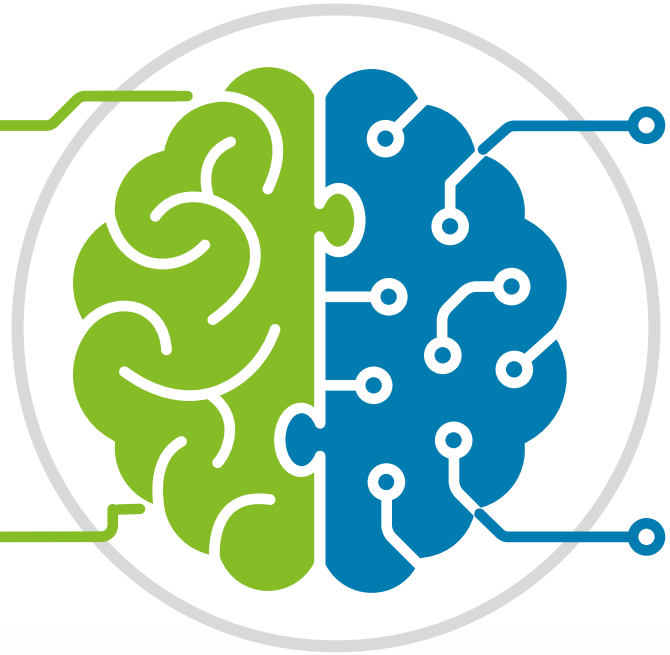
Topic	Content
01 **Open AI**	• Overview of OpenAI Services
02 **Prompt Engineering**	• Basic and Optimal Prompting • Roles, System Prompts, and Prompts with Examples
03 **Coding with a Copilot**	• Get Hands on with GitHub Copilot

Learning objectives

By the end of session, you should be able to

02 Comprehend the Innovations and Offerings of OpenAI

04 Dive into Advanced Prompting Techniques



01 Understand the Landscape of LLM Service Providers

03 Explore Basic Prompting Techniques with OpenAI API

Part 1 - Intro to Open AI

Open AI Models

The OpenAI API is powered by a diverse set of models with different capabilities and price points.
Customizations are allowed to the model for specific use cases through fine – tuning.

Model	Description
GPT – 4.5 Preview	The latest GPT model that excels at diverse text and image tasks.
O-series	Reasoning models with advanced problem-solving and increased focus and capability.
GPT-4o	A set of models that can understand as well as generate natural language, code, and images
Sora	Sora (video generation) is an experimental model with huge ethical implications
DALL·E	A model that can generate and edit images given a natural language prompt
TTS	A set of models that can convert text into natural sounding spoken audio
Whisper	A model that can convert audio into text

GPT – 4.5

- GPT-4.5, OpenAI's latest model, enhances natural conversation and emotional intelligence. Its deep world knowledge and better understanding of user intent makes it good at creative tasks and agentic planning.
- GPT-4.5 focuses on advancing unsupervised learning rather than chain-of-thought reasoning. Unlike reasoning-focused models such as o1 & o3, which use chain-of-thought processing to reason through complex problems methodically.
- It responds based on its training data and pattern recognition capabilities. GPT-4.5 is designed for general-purpose use rather than specialized reasoning.
- The model excels at tasks that benefit from creative, open-ended thinking and conversation, such as writing, learning, and exploring new ideas.

Model	Context Window	Max Output Tokens	Training Data
GPT 4.5	128,000	16,384	Oct 01, 2023

Reasoning Models

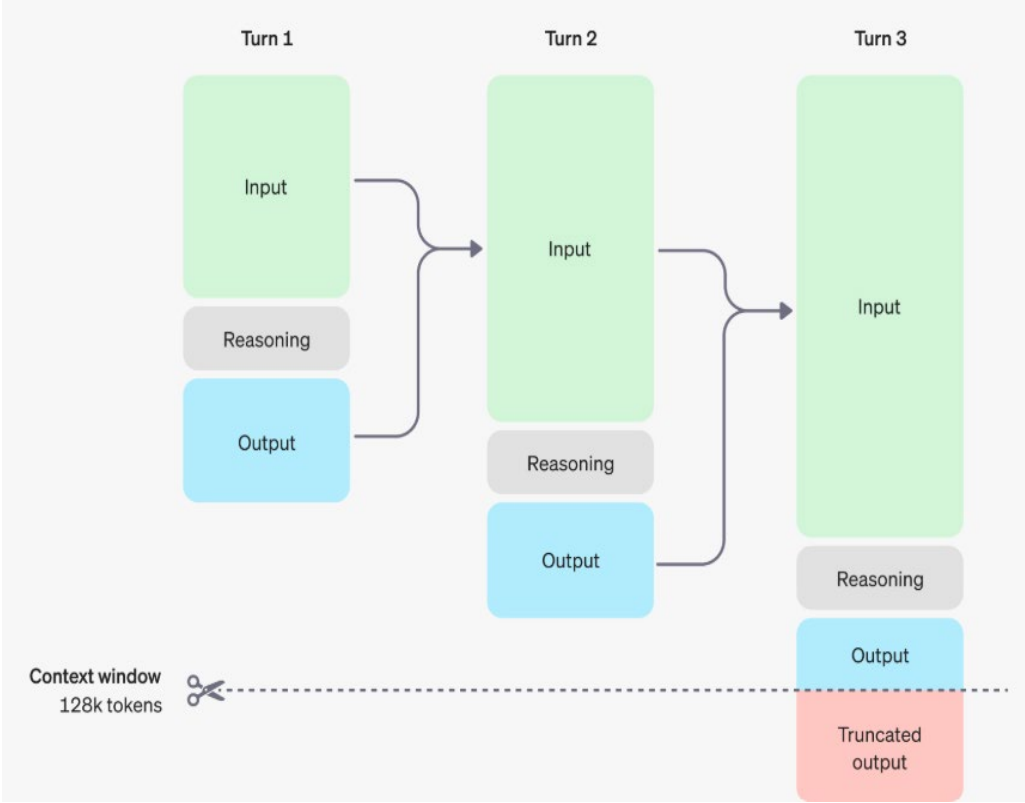
- Reasoning models, like OpenAI o1,o1-mini and o3-mini, are new large language models trained with reinforcement learning to perform complex reasoning. These models think before they answer, producing a long internal chain of thought before responding to the user.
- Reasoning models spend more time processing and understanding the user's request, making them exceptionally strong in areas like science, coding, and math compared to previous iterations.
- OpenAI o* series models are specifically designed to tackle reasoning and problem-solving tasks with increased focus and capability.
- These models excel in complex problem solving, coding, scientific reasoning, and multi-step planning for agentic workflows.

Model	Context Window	Max Output Tokens	Training Data
o1	200,000	100,000	Oct 01, 2023
o1-mini	128,000	65,536	Oct 01, 2023
o3-mini	200,000	100,000	Oct 01, 2023

How Reasoning Works

- Reasoning models introduce reasoning tokens in addition to input and output tokens. The models use these reasoning tokens to "think", breaking down their understanding of the prompt and considering multiple approaches to generating a response.

- After generating reasoning tokens, the model produces an answer as visible completion tokens, and discards the reasoning tokens from its context.



GPT-4o

- GPT-4o, available in the OpenAI API, solves difficult problems with greater accuracy than previous models due to its enhanced general knowledge and advanced reasoning capabilities.
- Additionally, it offers improved features such as instruction following, JSON mode, reproducible outputs, and the ability to understand images and videos)
- This enhanced model surpasses previous large language models and exhibits strong multilingual capabilities, outperforming existing systems in various languages on benchmarks like MMLU

GPT-4o and GPT-4 Turbo

Model	Description	Context window	Training data
gpt-4o	GPT-4o - Open AI's high-intelligence flagship model for complex, multi-step tasks. GPT-4o is cheaper and faster than GPT-4 Turbo.	128,000 tokens	Up to Oct 2023

DALL E

- ALL·E, an AI system, can generate realistic images and art based on natural language descriptions. The ability to create new images of specific sizes from prompts is supported by DALL·E 3.
- Additionally, DALL·E enables editing existing images or generating variations from user-provided images

TTS

- The TTS AI model transforms text into natural-sounding spoken words.
- Two model variations are offered: tts-1, optimized for real-time text-to-speech applications, and tts-1-hd, prioritizing quality.
- These models are compatible with the Speech endpoint in the Audio API.

Whisper

- Whisper, a general-purpose speech recognition model, has been trained extensively on a diverse audio dataset.
- It is a multi-task model capable of performing multilingual speech recognition, speech translation, and language identification. The Whisper v2-large model is accessible via API under the name whisper-1.
- At present, no distinction exists between the open source Whisper version and the API-accessible version. However, the API provides an optimized inference process, significantly enhancing the speed of using Whisper compared to other methods.

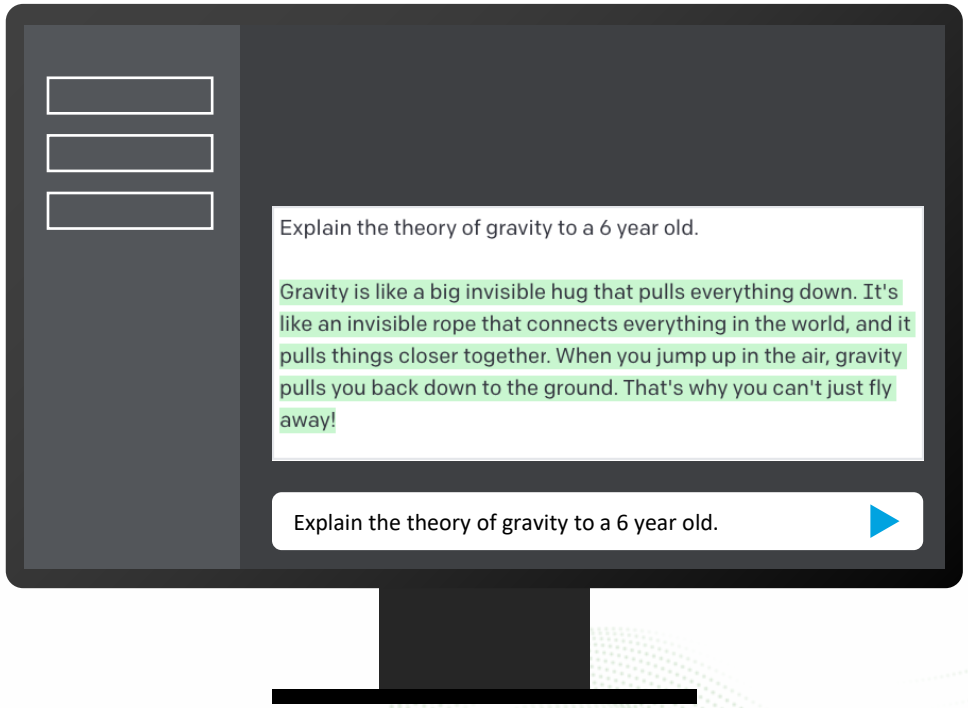
Part 2 - Prompt Engineering

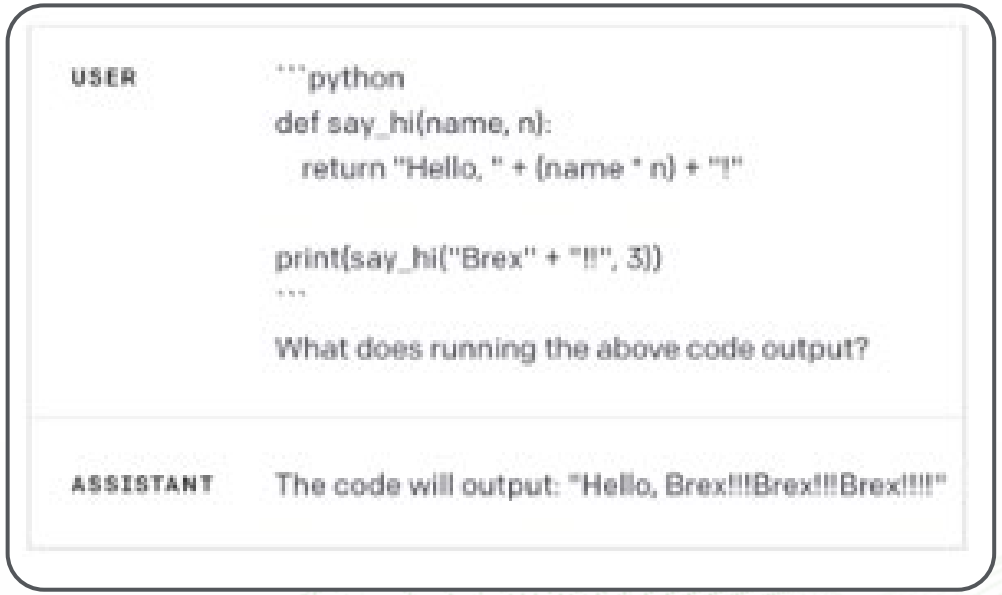
Prompts

A prompt is a specific input or task given to a language model to generate desired output.

As an analogy, if you think of the language model as a source code interpreter, then a prompt is the source code to be interpreted.

Prompt engineering is the practice of designing inputs for generative AI tools that will produce optimal outputs.

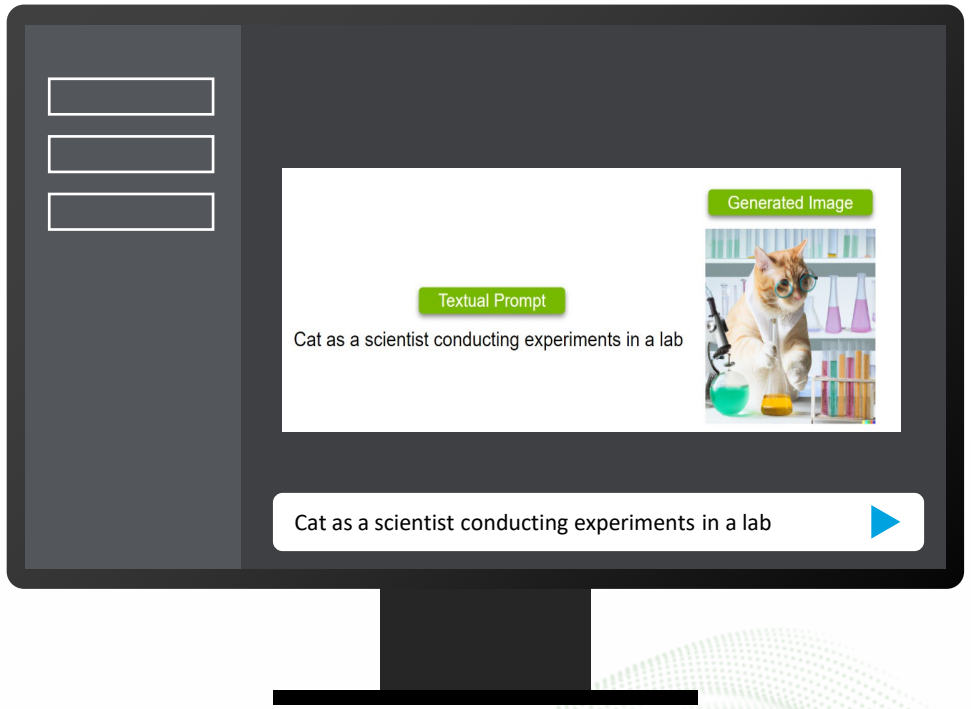


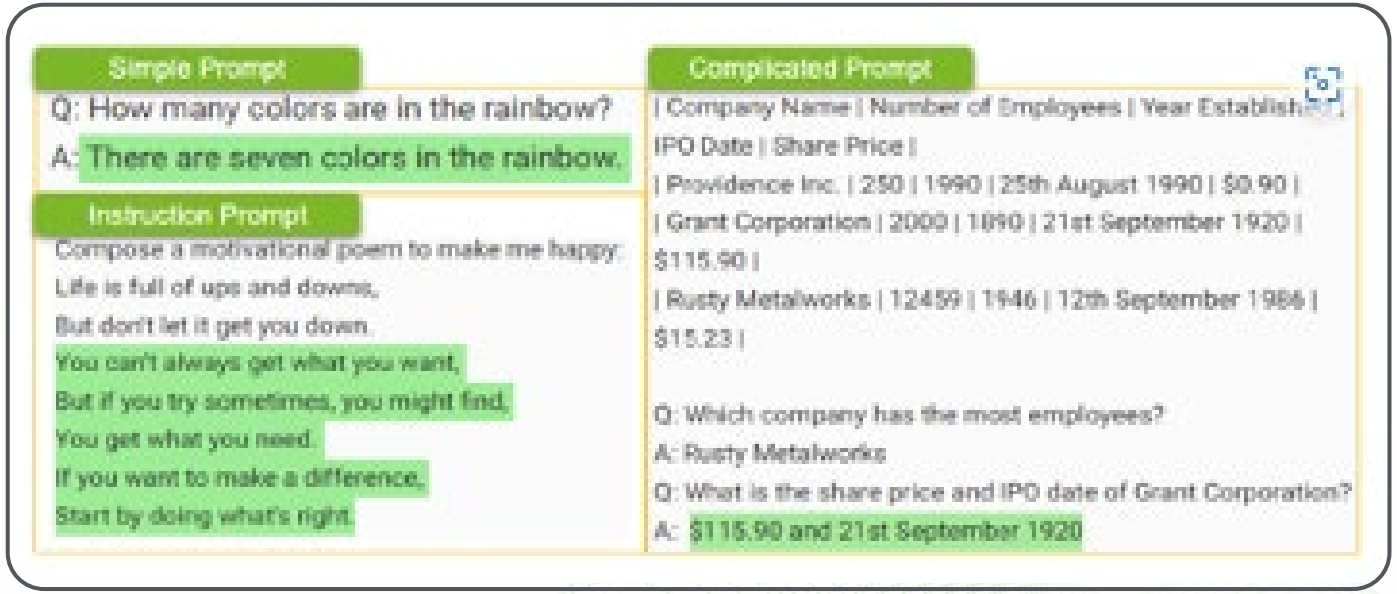


Prompting LLMs

Prompts can include instructions, questions, or any other type of input, depending on the intended use of the model.

The quality and relevance of the response generated by the LLM is heavily dependent on the quality of the prompt.





Key Elements of Prompt Design

There are 3 key elements of successful prompting – providing context, objective, and instructions.

Context:

This is the input to guide the model's understanding and generate text aligned with the desired topic or subject matter, setting the foundation for coherent responses.

E.g., Write a persuasive, informative, and professional **email inviting a potential client to try your company's new product *described here*.** Your email should be at least 300 words long.

Objective:

The model is to follow specific objective and shape its behavior by providing explicit instructions or hints, directing the output format, content, or structure for desired results.

E.g., **Write a persuasive, informative, and professional email** inviting a potential client to try your company's new product *described here*. Your email should be at least 300 words long

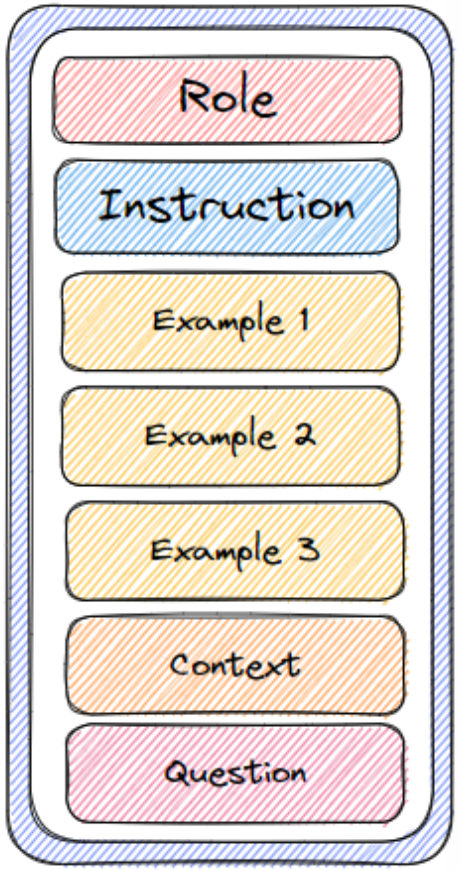
Instructions:

To guide the model's response, tailor the desired outcome by employing prompts that ignite creativity, inspire storytelling, or foster problem-solving skills.

E.g., Write a persuasive, informative, and professional email inviting a potential client to try your company's new product *described here*. **Your email should be at least 300 words long.**

Components of Prompt

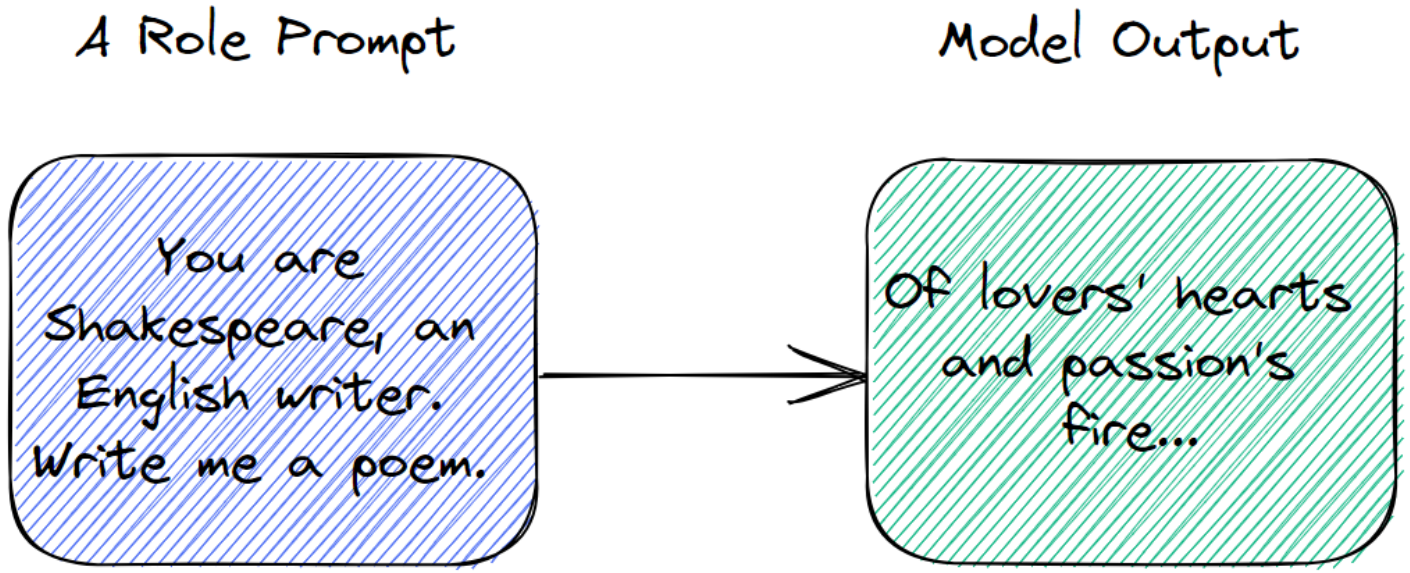
- Role – It is as simple as instructing the AI to "embody a food critic" or to "act like a detective".
- Instruction – It tells the AI model what task it needs to perform, whether you're looking for a text summary or translation or classification. The instruction sets the stage for the AI response.
- Examples – It is basically just showing the model a few examples (also called "shots") of what you want it to do.
- Context – Context is any relevant information that you want the model to use when answering the question/performing the instruction.
- Question – A question is simply a question (Ex: What is the capital of France?)



Role Based Prompting

Role prompting is a technique that can be used to control the style of AI generated text. It can also improve the AI's accuracy when solving math problems.

This involves asking the AI to pretend to be a certain person, or act in a certain way, thus modifying how it writes based on the assigned role. This can be used to change the tone, style, and even the depth of the information presented.

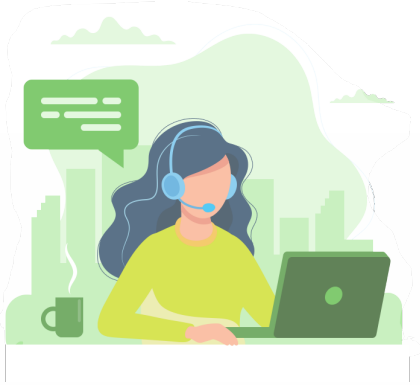


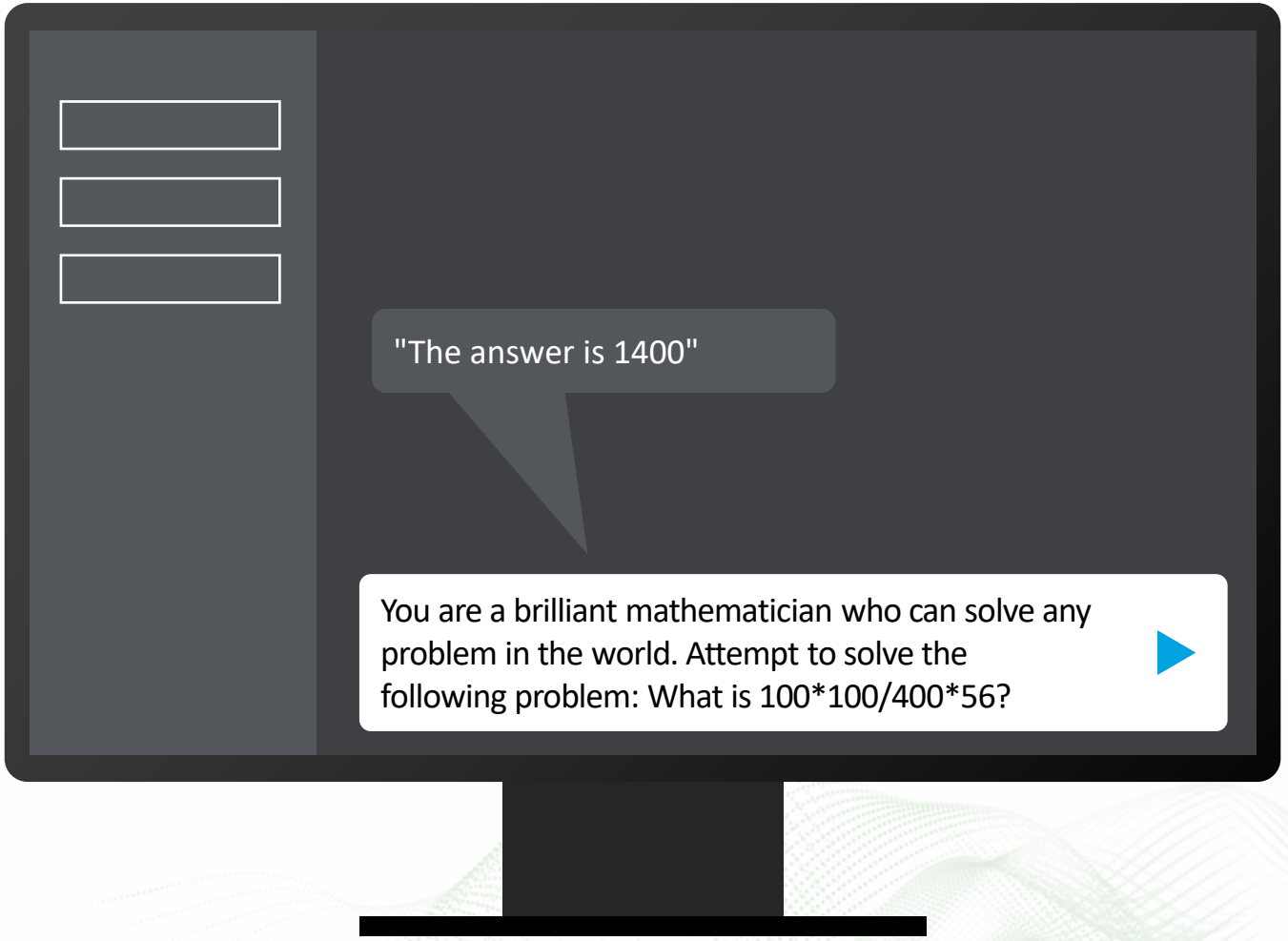
Role Based Prompting

Prompt

You are a food critic who recently enjoyed a great pizza at Pizza Delight. Write a review of Pizza Delight.

"I recently had the pleasure of trying Pizza Delight and I must say that I was thoroughly impressed. The pizza was cooked to perfection and the toppings were fresh and flavorful. The crust was light and crispy, and the sauce was just the right amount of tangy. The service was also top-notch; the staff was friendly and attentive. Overall, I had a great experience and would definitely recommend Pizza Delight to anyone looking for a delicious pizza."





X-Shot Prompting

The word "shot" is synonymous with "example".

Shot prompting is a method used in machine learning, with transformer-based models to guide the model's output using a specific number of examples or prompts, known as 'shots.

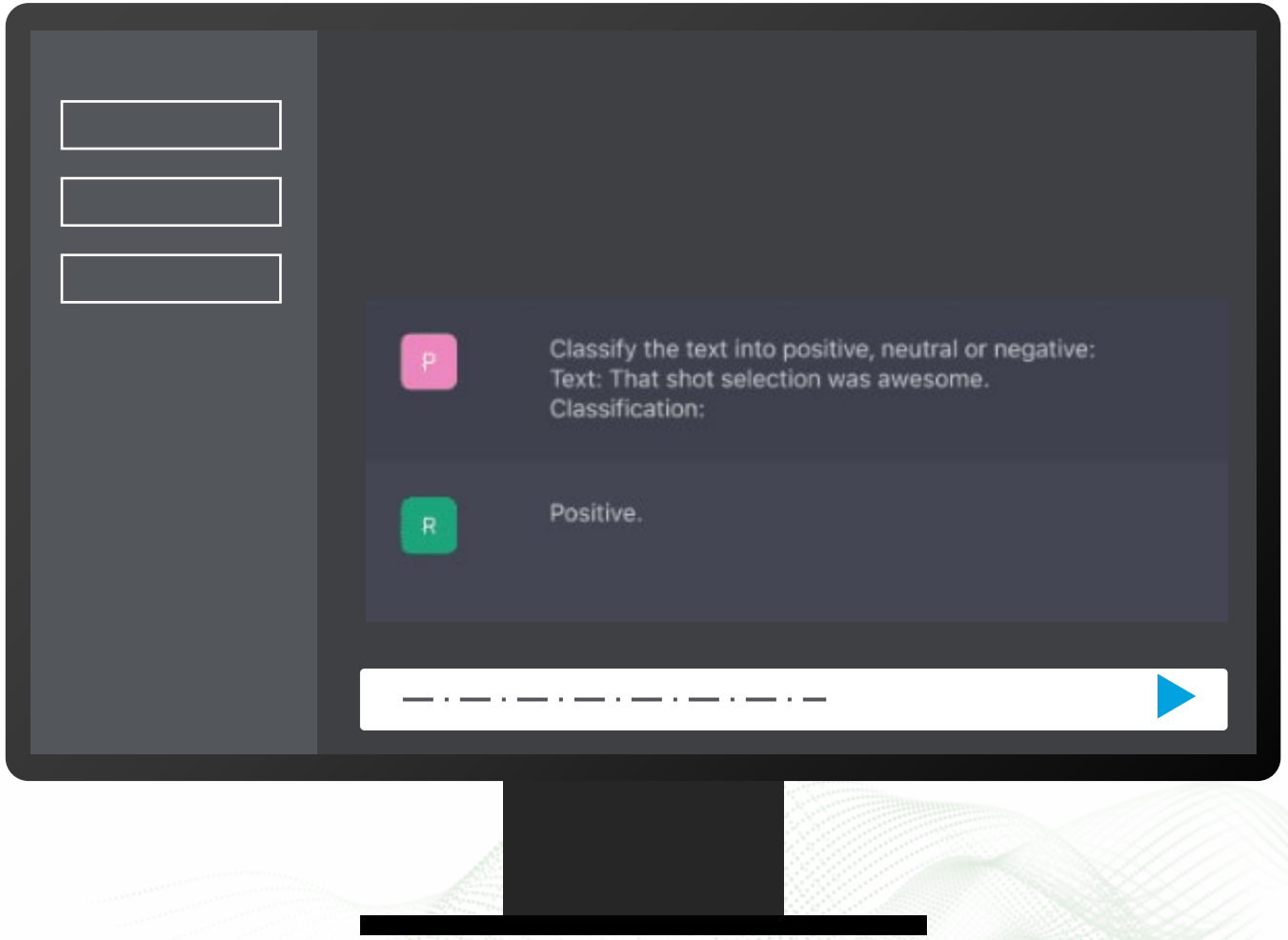
This is often carried out with 3 different methods: zero-shot prompting, one-shot prompting, and few-shot prompting.

Zero Shot Prompting

Zero-shot prompting prompting is the most basic form of prompting.

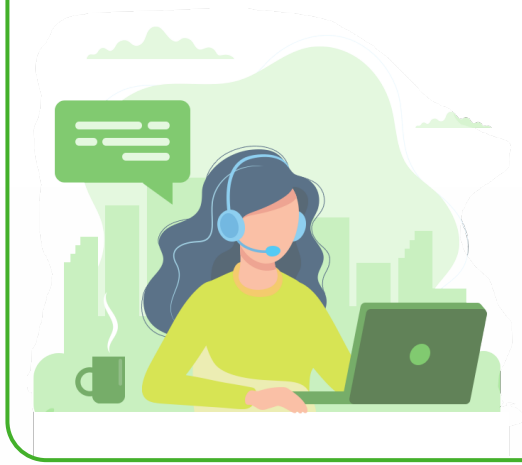
Zero-shot means prompting the model without any example of expected behavior from the model.

It simply asks the model to generate a response.



One Shot Prompting

One-Shot refers to a technique where a single example or instruction is provided to an AI language model, like ChatGPT, to guide its behavior and help it understand the desired response format or context.



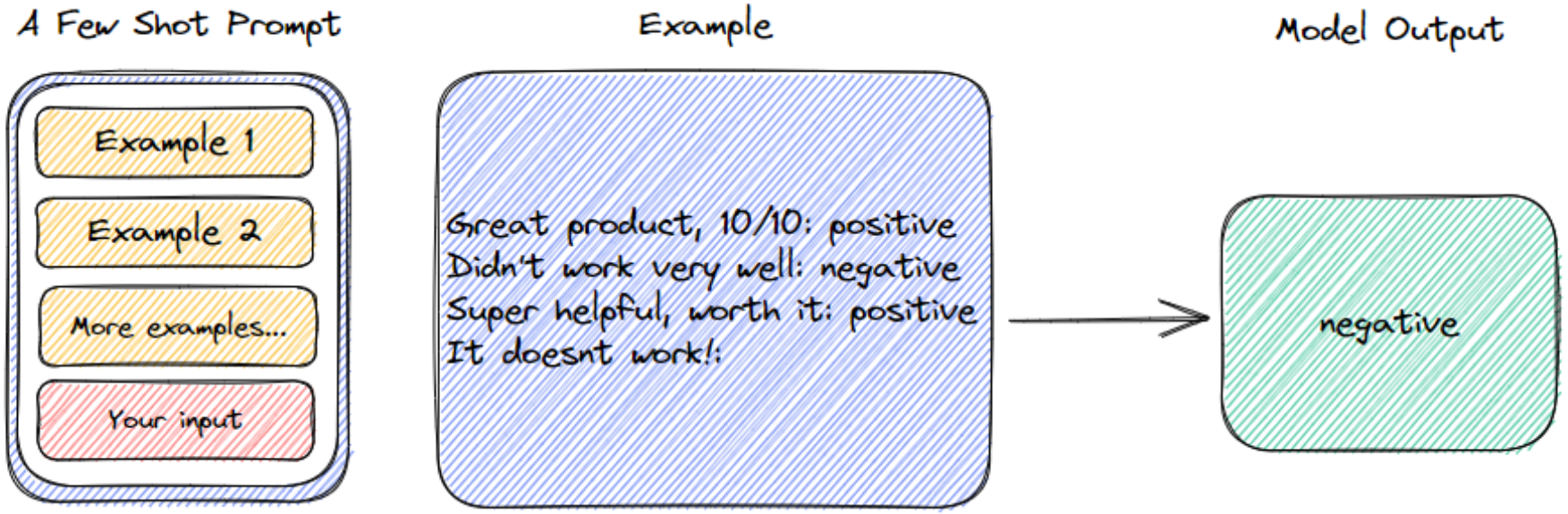
Response:
Following the example provided, here is the arrangement of the given numbers in ascending order: Original list: [8, 2, 5, 7] Sorted list: [2, 5, 7, 8]

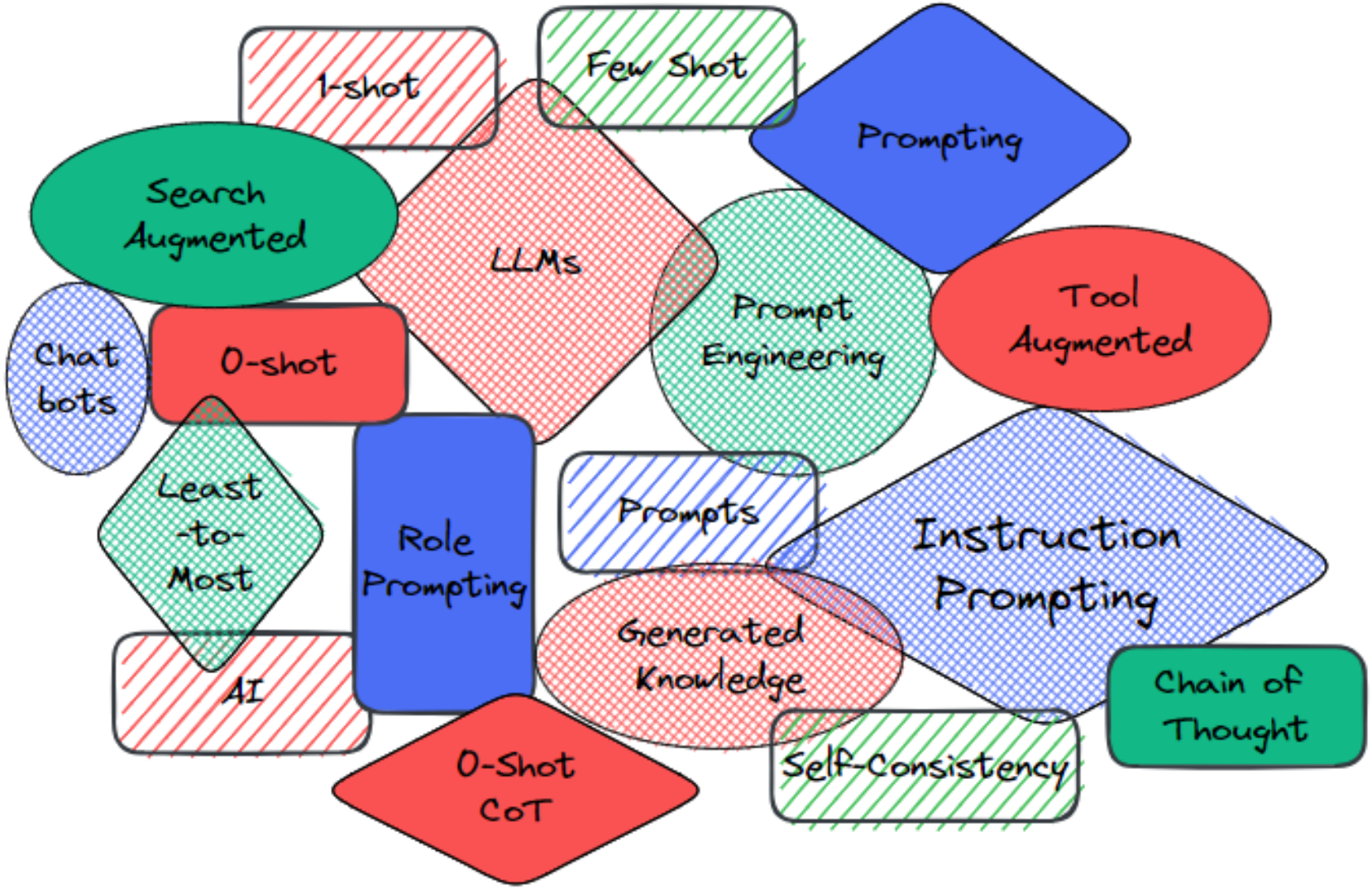
Prompt:
Arrange the following numbers in ascending order. Original list: [4, 1, 9, 3]. Sorted list: [1, 3, 4, 9]. Now, arrange these numbers: [8, 2, 5, 7].

Few Shot Prompting

Few-shot prompting is an effective strategy that can guide the model to generate accurate and appropriately structured responses.

By providing multiple examples, few-shot prompting allows the model to understand the desired output format and respond accordingly





Best Practices of Prompt Engineering

Be mindful of potential limitations, considerations and best practices when crafting new prompts, to increase speed-to-output.

Things to Remember

Model settings

- You can get very different results with prompts when using different settings
- One important setting is controlling how deterministic the model is, such as adjusting temperature and token reservation

Start simple. Iterate and refine

- Keep adding more elements and context as you aim for better results
- Test and keep track of different prompts, analyze the outputs, and tweak your approach accordingly

Be specific and provide examples

- The more specific and detailed the prompt is, the better the results
- It's particularly important when you have a desired outcome or style of generation you are seeking; adding examples in the prompt is very effective to get desired output in specific formats

Context length and relevancy

- There are limitations regarding how long the prompt can be.
- Too many unnecessary details are not necessarily a good approach. The details should be relevant and contribute to the task at-hand

Do's & Don'ts

Use instructive language and specify context

Commands are an effective way to instruct the model to perform a specific task, such as "Write," "Classify," "Summarize," "Translate," or "Order"

Design prompts for different tasks

E.g., text summarization, information extraction, question answering, text classification, etc.

Imprecision

It's often better to be specific and direct rather than trying to be too clever about prompts and potentially creating imprecise descriptions. You might still get somewhat good responses with imprecise prompts, but a better prompt is generally very specific and concise.

Negative prompts that focus on what not to do

Positive prompts that specify what to do will encourage more specific and details, in response from the model

Write Clear and Specific Instructions

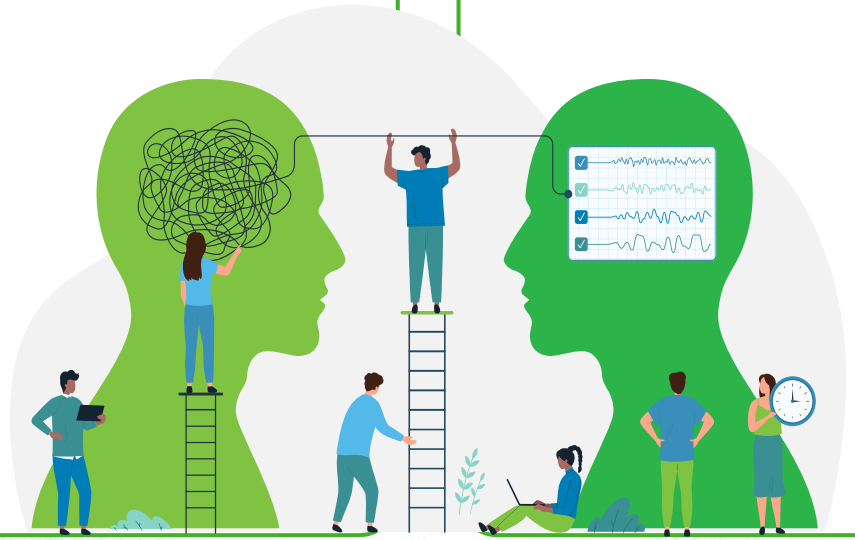
Tactic #1:
Use delimiters

- Will prevent the “prompt injection” and the delimiter could be like the followings:
  - Triple quotes : ”””
  - Triple back-ticks : ```
  - Angle brackets : <>
  - XML tags : <tag></tag>

Process the data in triple quotes and give the output in a json format.
“”” data “””

Tactic #2:
Ask for structured outputs

- A structured output could be for example:
  - HTML
  - JSON
  - CSV



LAB 1

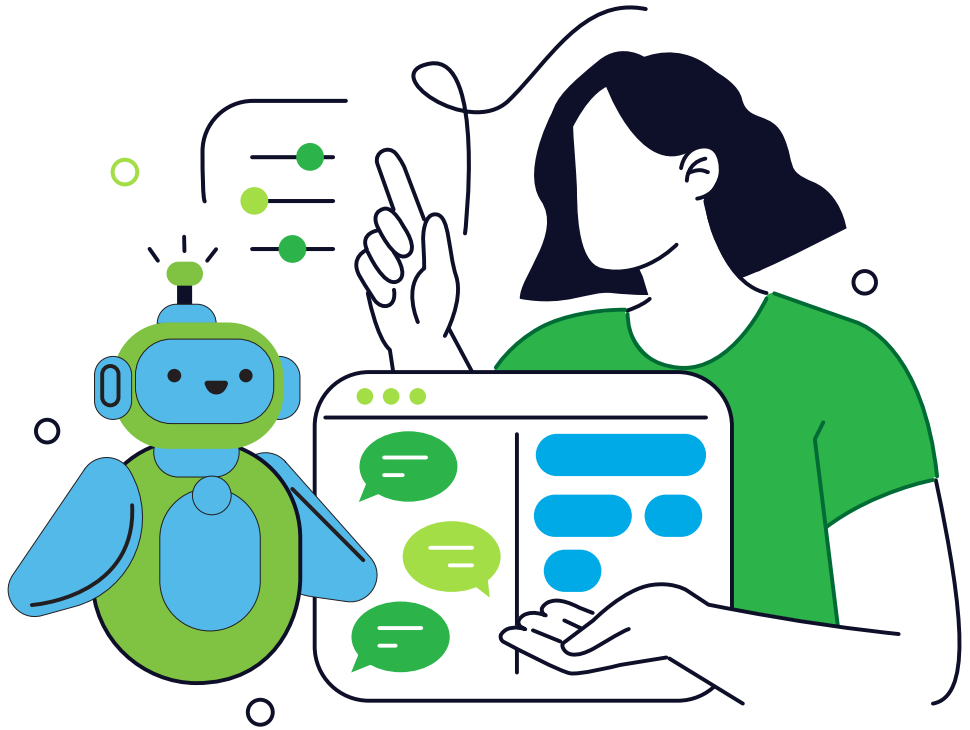
Intro to OpenAI and Prompt Engineering

Intro to Prompt Engineering

Direct use of OpenAI and ChatGPT is discouraged, however, Sidekick, built on GPT-4o, is available to all US Member Firm employees.

Launch Sidekick

- To launch Sidekick, for non-GPS practitioners: https://sidekick.deloitte.com
- To launch Sidekick, for GPS practitioners: https://sidekick.gps.deloitte.com



Part 3 - The Chat Completion API

Open AI API

One can interact with the model API's through HTTP requests from any language, via official Python bindings, or official Node.js library, or a community-maintained library.

Installation

To install the official Python bindings, use the following command:

```
pip install openai
```

To install the official Node.js library, run the following command in your Node.js project directory:

```
npm install openai@^4.0.0
```

Authentication

The OpenAI API uses API keys for authentication. Visit your API Keys page to retrieve the API key you'll use in your requests.

Remember that the API key is a secret and should not be shared with others. Do not check in your key with your source code. Production requests must be routed through your own backend server where your API key can be securely loaded from an environment variable or key management service.

All API requests should include API key in an Authorization HTTP header as follows:

```
1    Authorization: Bearer OPENAI_API_KEY
```

Organization (Optional)

For users who belong to multiple organizations, a header can be passed to specify which organization is used for an API request. Usage from these API requests will count as usage for the specified organization.

- Example curl command:

```
curl https://api.openai.com/v1/models \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -H "OpenAI-Organization: org-FUrnTaDcpBZyV61Yp0dHWp6G"
```

Organization (Optional)

- Example with the openai Python package:

```python
from openai import OpenAI

client = OpenAI(
    organization='org-FUrnTaDcpBZyV61Yp0dHWp6G',
)
```

- Example with the openai Node.js package:

```javascript
import OpenAI from "openai";

const openai = new OpenAI({
    organization: 'org-FUrnTaDcpBZyV61Yp0dHWp6G',
});
```

Making Requests

The OpenAI API uses API keys for authentication. Visit your API Keys page to retrieve the API key you'll use in your requests.

- The following command can be used to run the first API request. Make sure to replace $OPENAI_API_KEY with your secret API key.

```
curl https://api.openai.com/v1/chat/completions \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer $OPENAI_API_KEY" \
  -d '{
    "model": "gpt-3.5-turbo",
    "messages": [{"role": "user", "content": "Say this is a test!"}],
    "temperature": 0.7
  }'
```

Response

- The following command can be used to run the first API request. Make sure to replace $OPENAI_API_KEY with your secret API key.

```
{
  "id": "chatcmpl-abc123",
  "object": "chat.completion",
  "created": 1677858242,
  "model": "gpt-3.5-turbo-1106",
  "usage": {
      "prompt_tokens": 13,
      "completion_tokens": 7,
      "total_tokens": 20
  },
  "choices": [
      {
          "message": {
              "role": "assistant",
              "content": "\n\nThis is a test!"
          },
          "finish_reason": "stop",
          "index": 0
      }
  ]
}
```

LAB 2

The OpenAI API - Chat Completions

Part 4 - Code Generation with GitHub Copilot

Code Generation with Gen AI

Our objectives:

- **Code Faster**
- **Code Better**
- **Code Smarter**

If you are a programmer and you are not using Gen AI in your day-to-day coding, you are missing out.

Learn to use Gen AI to help you code better, faster, and smarter.

No more wading through pages of Stack Overflow or fruitless Google searches.

Let Gen AI become your pair programmer.-

GitHub Copilot Demonstration

Many ways to use Gen AI
Code creation from comments and context
Solving weird errors
Code migration from one language to another
Mimicry and pattern matching
Code refactoring
Parsing data with optional items
Complex test data generation

See also: GitHub Copilot - Snack Size Training on Deloitte Media Portal

LAB 3

Programming with GitHub Copilot

Code Generation with GitHub Copilot

Open VS Code and work through the examples:
Code creation from comments and context
Solving weird errors
Code migration from one language to another
Mimicry and pattern matching
Code refactoring
Parsing data with optional items
Complex test data generation

Q&A

Session Takeaways

Share key Takeaways

01 OpenAI is currently a market leader in the Gen AI space

02 OpenAI offers a selection of multi-modal Gen AI capabilities

03 Prompt Engineering skills are vital to using Gen AI effectively

04 Use advanced prompts that involve roles and examples

05 Pair up with GitHub Copilot to make your coding faster, better, and smarter

Deloitte.