

Day 16-20 Revision: OpenAI API & Prompt Engineering

Day 16 - LLM & GPT Basics

- LLM = Large Language Model trained on vast amounts of text data.
- GPT = Generative Pretrained Transformer (generates text based on prompts).
- Token = A small piece of word (GPT reads input/output as tokens).
- GPT works by predicting the next token given the context.

Day 17 - OpenAI API Setup & First Call

- Create account on <https://platform.openai.com> and get your API key.
- Install OpenAI Python library: `pip install openai`
- Use ChatCompletion API with messages in role-content format.
- Secure API key using `.env` file and load it using `dotenv`.

Day 18 - Prompt Engineering Techniques

- Zero-shot: Give direct instruction, no examples.
- Few-shot: Give 2-3 examples before asking the question.
- System role: Defines the assistant's behavior (e.g., friendly teacher).
- Better prompts = Better responses. GPT follows instruction quality closely.

Day 19 - Memory-Based Chatbot

- Use a chat history list with user/assistant messages.
- Append each message to maintain conversation context.
- Pass the full message list every time for GPT to stay in context.
- Good for assistant-style bots with multi-turn memory.

Day 20 - Prompt Template Mini Project

- Built a chatbot that gives topic-based structured introductions.

Day 16-20 Revision: OpenAI API & Prompt Engineering

- Used a system prompt to instruct GPT to reply with:
 1. Introduction
 2. Key Use-cases
 3. Why it's important
 4. Motivational closing
- Combined user input with prompt template.
- Fully functional CLI bot with OpenAI API + Prompt Engineering.