# ChatGPT Key Terms

**GPT** Generative Pretrained Transformer. GPT works by predicting the next word in a sequence, creating text that closely mirrors human writing.

**Token** A unit of information in the GPT model, usually representing a word or part of a word. Each unique word, punctuation mark, or space in the text input to GPT is counted as a single token.

**Context** The preceding series of tokens in a text that GPT uses as a reference to predict the next word or sequence.

**Hallucination** A phenomenon where GPT generates outputs unrelated to its input or pretrained knowledge, essentially creating unrequested or incorrect details.

**Jailbreaks** Strategies employed to make GPT output information beyond its trained restrictions, such as details post-training cutoff or content against ethical guidelines.

**Token Limits** The maximum number of tokens, including input and output, a GPT model can handle in a single sequence, like the 4096-token limit in GPT-3.5.

**Training Cutoff** The end date of the data used to train GPT, limiting its knowledge to events or developments that occurred before this date.

**Embedding** A representation of words or text as points in high-dimensional space that the model learns during training. Embeddings express semantic meaning and relationships between words.

# 4 Ingredients to a Good Prompt

## 1. The Initial Context

Instruct GPT which role it should play

- You are a writing tutor that helps people improve their spelling, writing, and tone.
- Imagine you are a tech-industry hiring expert that specializes in getting your clients jobs at FAANG companies.
- Act as a Python Code-Reviewing assistant

*Often, we can distill our initial context into an "Act as..." phrase:*

- **Act as** a hiring manager...
- **Act as** a strict English tutor...

## 2. The Instruction(s)

Describe the actual task or question you want help with

- Respond to the following email...
- Translate this paragraph...
- Write a cover letter for the following job description...
- Write a Javascript function that generates random prime numbers...

# 4 Ingredients to a Good Prompt

## 3. Input Data

If necessary, provide input information the model needs.

Respond to the following email: **Dear So-and-so, ...**

Translate this paragraph: **"There once was a man from Nantucket"**

Write a cover letter for the following job description: **```We're looking for a...```**

## 4. Constraints and Format

Include specific requirements or constraints for your output

**Constraints**

- Summarize The Great Gatsby **in 2-3 sentences**.
- **As concisely as possible**, describe Frodo's journey to Gondor.
- Explain superconductivity **in terms a 5th grader could understand**.
- **Responding with only "yes" or "no"**, answer the following questions...

**Format**

- Print all Pokemon and their resistances **in CSV format**
- Give me a list of all Pokemon and their descendants **as JSON**
- Generate a list of all Poker hands **formatted as a Markdown Table**
- Fetch the historic statistics for ___ plotted as a **Histogram/Scatter Plot/Bar Graph** (Uses Code Interpreter)

# 4 Ingredients to a Good Prompt

1. **Initial Context**
2. **The Instruction(s)**
3. **Input Data**
4. **Constraints and Format**

## Complete Examples

(1) Explain microprocessors (4) to a middle schooler, using fewer than 200 characters

(1) You are leading an interview for a front-end developer role at a tech company. I will be the candidate and you will ask me interview questions. (4) Ask me one question at a time. Once I have responded, rate my answer to your question on a scale from 1-10. And then ask the next question.

(1) Act as a news article summarizer. (2) I will provide you with an article and you will create a summary of the main points. (4) Your summary should include a 2 sentence overview of the article and then 4-6 bullet points. Your summary should not include any direct quotes from the paper. Your response should be no more than 100 words. (3) Here is the article text: ...

(4) Write a 3 paragraph blog post on (1) why live coding interviews are a terrible method of assessing developer candidates. (4) Your response should be formatted as markdown. You should bold any key sentences or phrases.

# Prompting Techniques: N-Shot Prompting

## Zero Shot Example

Classify the sentiment in the following tweet: I've never seen Vini THIS frustrated, can't even imagine what this fan said to him but the fact that the whole team got behind him means it's something very serious...

## One Shot Example

You are a sentiment analysis assistant. Given a tweet, respond with the overall sentiment of that tweet.
Input: "Arsenal completely gave up on winning the league this year"
Output: Negative
Input: "At least I'm looking forward to next season. Come on Arsenal!"
Output:

## Few Shot Example

I want you act as an expert web development instructor, course creator, and blog writer. You are an expert on topics related to web development, coding, AI, and technical interviews. You are creating a blog post on how to prepare for coding interviews. Follow my instructions and follow them very carefully! Do not deviate from my instructions.
1. Provide a step by step reasoning of why you're making the decisions you are making
2. Write an engaging, professional title for the blog post. The primary audience is entry-level developers and coding bootcamp graduates
3. Write a 100-150 word abstract for the blog post. The abstract should be two paragraphs, the first paragraph introduces why the blog post is important, the second paragraph provides an overview of what will be covered in the blog post.
4. Provide 3 key takeaways readers of the blog post should remember.
5. Have a credible, professional, and accessible voice and tone. A good example would be that of DataCamp, McKinsey, and other reputable organizations.
6. Write the entire blog post.

# Prompting Techniques

**Incremental Prompting** Rather than writing a single complex prompt, it's often better to write multiple smaller prompts that you run individually in sequential order. This is especially useful when dealing with code-related prompts.

**Self-Criticism** Ask the model to criticize its own output.

"How can you improve this?"
"Can you improve the blog post?"

**Chain of Thought Reasoning** Explain your reasoning step-by-step.

"Think step by step."

**Ask for multiple perspectives.**

"from the perspective of medieval peasant"
"From the perspective of a college student"

**Ask ChatGPT to write its own prompt.**

"Help me craft a GPT prompt for..."

**Avoiding Hallucinations** Ask for citations or references

"...include citations with any facts you provide"

# Coding with ChatGPT

**Code Generation** Use GPT to generate self-contained code snippets ranging from API calls, simple algorithms, unit tests, or boilerplate.

**Code Annotation** Explicitly request GPT to annotate your code. This enhances readability and future code maintenance.

**Code Refinement** Utilize GPT for code refactoring; ask it to simplify, deduplicate, or optimize your existing code.

**Comprehending Algorithms** For better understanding, request GPT not only to formulate an algorithm but also to clarify its working mechanism step-by-step.

**Naming Conventions** Seek GPT's recommendations for naming variables and functions, adhering to best practices for code clarity.

**Error Mitigation** Ask GPT for strategies to manage potential errors or exceptions, contributing to the robustness of your code.

**Language Proficiency** Use GPT to learn language-specific idioms and practices, enhancing your coding efficiency and adherence to best standards.

## Weaknesses

Writing code is complex. GPT struggles with the generation of larger applications, complex interdependencies, and software popularized or updated after its training cutoff date.

## Recommended Usage

"Generate a function to perform [task]."
"Identify the errors in this code snippet..."
"What does this error mean?"
"Suggest enhancements for this algorithm."
"Convert this Python code to Scala."

# GPT Advanced Parameters

When using GPT through the playground or the API, you can tune the model's behavior with parameters. Some important ones include:

**Temperature** Controls the randomness of the output. Higher values are more random, lower values are more deterministic. The value ranges from 0 to 2, defaulting to 1.

**Top P** Also known as nucleus Sampling, the Top P parameter tunes the randomness of the output by limiting the set of words the model can choose from. The default Value is 1 which means "100%" of values. A lower value such as 0.2 would mean "20%" and thus, induce, more predictable output.

**Frequency Penalty (Proportional)** Decreases the model's likelihood of repeating output verbatim by penalizing words' likelihood based on *their existing frequency* in the text so far. To reduce repetitive samples, use a penalty from 0.1-1. To induce more repetition (a rare use case), use a negative value.

**Presence Penalty (One-off)** Decreases the model's likelihood of repeating output verbatim by penalizing words' likelihood based on *whether or not* they've appeared in the text so far. To induce more variation in your output, use a presence penalty from 0.1-1.