



# AImpact Program PROMPT ENGINEERING

Version 1.1  
Mar 2025

# Who am I?

## **Nguyen Tran**

- ❖ AI Group Leader
- ❖ Born in 1997
- ❖ Joined PTN Global since 2020
- ❖ Currently pursuing a PhD in Japan



# Contents

## Part 1: Prompt Engineering

1. Basics
2. Intermediate
3. Advanced

## Part 2: Prompt Library for Testing

1. Test Strategy & Planning
2. Test Case Writing
3. Test Automation Writing
4. Sample Data Creating

## Part 3: Prompting Challenges

1. Challenge
2. How-to

## References

# Overview

## Goals

- ❖ ~~AI is replacing resources~~ – AI is augmenting resources.
- ❖ **AI supports our teams** by automating repetitive tasks, freeing them to focus on more complex, high-value testing.
- ❖ ~~Becoming a “Code Monkey”~~
- ❖ **Prioritize the “Why” over the “How”**, understanding the intent behind the code leads to smarter, more impactful solutions.

# Overview

## Challenges (1/2)

### **I. Context Understanding & Accuracy**

1. AI may struggle to fully grasp domain-specific rules and edge cases, leading to incomplete or incorrect test scripts.
2. AI might generate redundant or irrelevant test cases, increasing maintenance efforts instead of reducing them.
3. AI require high-quality data to generate effective test scripts. If historical test cases and requirements are inconsistent or incomplete, AI performance suffers.
4. If requirement documents are unclear, AI-generated test cases may not accurately reflect the intended functionality.

### **II. Integration with Existing Tools & Workflows**

1. AI-generated test cases need to integrate seamlessly with existing test management tools like Selenium, Katalon, TestRail, etc.
2. Some workflows that AI-generated test cases may not easily fit into.

### **III. Security Concerns**

1. AI systems processing test scripts and test data must ensure compliance with data privacy regulations (e.g., GDPR, HIPAA).

# Overview

## Challenges (2/2)

### **IV. Change Resistance & Skill Gaps**

1. Traditional QA teams may be hesitant to adopt AI-based test generation due to job security concerns.
2. Testers may require additional training to effectively use AI-driven tools and interpret AI-generated test cases.

### **V. Measuring Effectiveness**

1. Demonstrating benefits of AI-driven test automation (e.g., cost savings, defect reduction) can be challenging.
2. The initial effort to train AI models and fine-tune generated test cases might outweigh immediate benefits.

# Overview

## Approach (1/2)

### **I. Start with Low-Risk AI Implementations**

1. Identify repetitive or high-effort tasks where AI can add immediate value
  - *Manual test scripts and automation test cases generation from user stories or requirements.*
  - *Utilize prompt management tools to streamline prompt sharing and best practices.*
  - *Implement AI-driven automation gradually by start with AI scripts in Selenium, Katalon and Cypress.*

### **II. Upskill the QA Service**

1. Conduct AI & automation training and provide hands-on practice with AI-assisted tools in real projects.
  - *Technology & Innovation will collaborate with QA Team to lead knowledge sharing.*
2. Foster a culture of AI collaboration, not replacement
  - *Highlight success stories where AI improved efficiency.*
  - *Reinforce that AI assists testers, allowing them to focus on complex testing.*
  - *Encourage an AI-first mindset—let testers experiment with AI-generated scripts before writing manual ones.*

# Overview

## Approach (2/2)

### **III. Address Data & Accuracy Concerns**

1. Integrate AI-powered test case generators into existing projects.
  - *Use historical test scripts for smarter test case creation.*
2. Build AI tools to assist business requirements analysis.
  - *Standardize requirement documentation for better AI test generation.*

### **IV. Monitor & Measure AI Impact**

1. Define KPIs (e.g., test execution time reduction, defect detection rate).
2. Track AI's effectiveness in reducing repetitive manual work.
3. Gather feedback from QA engineers to refine AI workflows.



# Part 1: Prompt Engineering

1. Basics

2. Intermediate

3. Advanced

# 1. Basics

## What is Artificial Intelligence (AI)?

**Artificial Intelligence (AI)** refers to the field of technology where machines and systems are developed to perform tasks that traditionally require human intelligence. These tasks include:

- ❖ Recognizing faces in photos
- ❖ Understanding and responding to speech
- ❖ Making decisions and solving problems
- ❖ Playing chess at a level no human can match
- ❖ Learning from data and identifying patterns

AI systems, especially those based on **Machine Learning (ML)**, work by learning from data much like humans learn from experience.

# 1. Basics

## How does AI learn?

AI learns through a four-step process:

- ❖ **Data Collection:** Gathering relevant datasets (images, text, audio, etc.)
- ❖ **Model Training:** Using algorithms to identify patterns and adjust parameters
- ❖ **Evaluation:** Testing accuracy and fine-tuning the model
- ❖ **Inference:** Applying the trained model to new data

# 1. Basics

## Common Applications of AI in Daily Life

AI is integrated into many everyday technologies:

- ❖ **Voice Assistants:** Siri, Alexa, and Google Assistant for voice commands and tasks
- ❖ **Recommendation Systems:** Netflix, YouTube, and Spotify for personalized content suggestions
- ❖ **Social Media:** Content filtering and personalized feeds
- ❖ **Smart Devices:** Automated home systems and personal devices

# 1. Basics

## What is Generative AI?

**Generative AI (GenAI)** is a subset of artificial intelligence that focuses on creating new content. Generative AI combines learned patterns to create entirely new outputs. It can create:

- ❖ **Text:** Essays, articles, and poetry
- ❖ **Images:** Pictures from text descriptions
- ❖ **Music:** Songs in specific styles or genres
- ❖ **Code:** Programming snippets from natural language descriptions

Some real-world examples of GenAI tools:

- ❖ **ChatGPT:** An LLM that generates human-like text, answers questions, and writes code
- ❖ **DALL-E:** Creates detailed images from text descriptions
- ❖ **GitHub Copilot:** Helps developers write code by suggesting snippets and functions

# 1. Basics

## What is Large Language Model (LLM)?

A **Large Language Model** (LLM) is a type of machine learning model designed to understand, generate, and manipulate human language.

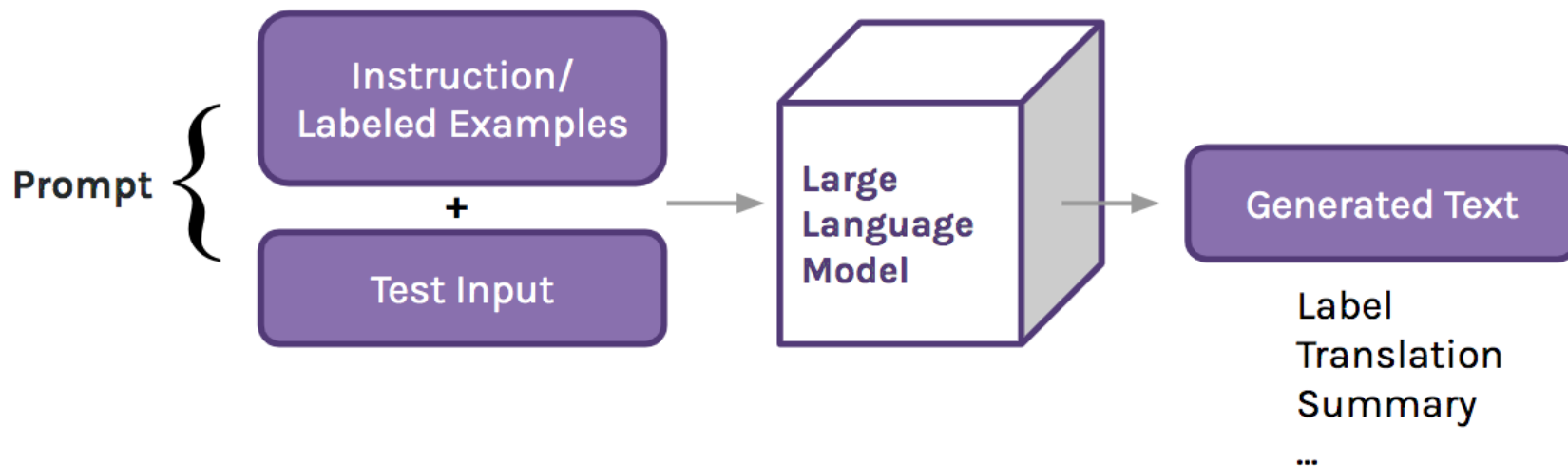
Main Types of Output:

- ❖ **Synthesizing Info** - Summarize large amounts of content in a concise way
- ❖ **Content Creation and Copy** - Generate brand new content for a specific topic and purpose.
- ❖ **Learning and Research** - Get questions answered for extremely specific needs
- ❖ **Coding** - Generate code, info on programming concepts and APIs, and debugging

To understand more about LLM: [\[Reference\]](#)

# 1. Basics

## How LLM works?



# 1. Basics

## What is Prompt Engineering?

**Prompt Engineering** is the art of writing and refining your **prompts** to get *better results* from generative AI models. It means providing clear, detailed inputs so the AI can produce more accurate and useful outputs.

**A prompt** is the instruction or input you give to an AI model. It can be a simple question or a detailed command that sets the tone, style, or format of the output.


**Let's try out some prompts!**



# 1. Basics


## Prompt Example: Simple Query


 **Prompt:** What color is grass?

 **Output:** Grass is typically green, thanks to chlorophyll—the pigment plants use to absorb sunlight for photosynthesis. But depending on the type of grass, the season, or its health, it can also appear yellowish, brown, or even bluish-green.

# 1. Basics

## Prompt Example: Simple Query


 **Prompt:** Explain me what is prompt engineering like I'm 5.

 **Output:** Sure! Imagine you have a toy robot that listens to what you say and does things like answering questions or telling stories. But, this robot doesn't always understand exactly what you want unless you say it in a certain way. Prompt engineering is like figuring out the best way to talk to the robot so it does exactly what you want. For example, if you want the robot to tell a funny story about a cat, you might need to say, "Tell me a funny story about a cat," instead of just "Tell me a story."


In short, prompt engineering is about asking the right questions or giving the right instructions to make sure the robot gives you the best answers!

# 1. Basics

## Prompt Example: Summarizing an Article

 **Prompt:** It is very rare for snow to fall in the U.S. state of Florida, especially in the central and southern portions of the state. Except for the far northern areas of the state, most of the major cities in Florida have never recorded measurable snowfall, though trace amounts have been recorded, or flurries in the air observed a few times each century. According to the National Weather Service, in the Florida Keys and Key West, there is no known occurrence of snow flurries since the European colonization of the region more than 300 years ago. In Miami, Fort Lauderdale, and Palm Beach there has been only one known report of snow flurries observed in the air in more than 200 years; this occurred in January 1977. In any event, Miami, Fort Lauderdale, and Palm Beach have not seen snow flurries before or since this 1977 event.

Summarize this paragraph in a single sentence:

 **Output:** Snow is extremely rare in Florida, with most major cities never recording measurable snowfall, and only a single instance of snow flurries reported in Miami, Fort Lauderdale, and Palm Beach in January 1977.

# 1. Basics

## Prompt Example: Solving Math Problems

 **Prompt:** If I have 20 apples, eat three of them, and sell three more, how many do I have left?

 **Output:** If you start with 20 apples, eat 3, and sell 3, you've gotten rid of 6 apples in total.  $20 - 6 = 14$ . You have 14 apples left. 

# 1. Basics

## Prompt Example: Improved Prompt

 **Improved Prompt** : What is  $923 * 99$ ?

Let's go step by step.

Always write out the full number of 0s for each term.

 **Output:** To multiply 923 by 99, we can break it down into smaller steps:

Step 1: Write out the full number of 0s for each term:  $923 * 99 = 923 * (9 + 90)$

Step 2: Multiply the first term by 9:  $923 * 9 = 8307$

Step 3: Multiply the first term by 90 (adding one 0):  $923 * 90 = 83070$


Step 4: Add the results from step 2 and step 3:  $8307 + 83070 = 91377$

Therefore, 923 multiplied by 99 is equal to 91,377.

# 1. Basics

## Prompt Example: Text in Action


 **Improved Prompt:** Write a marketing summary for a new AI tool that helps companies automate tasks.

 **Output:** Write a 100-word marketing summary for an AI tool called TaskBot, which automates repetitive tasks for small businesses in industries like retail and healthcare. Highlight efficiency and cost savings.

# 1. Basics

## Prompt Example: Text in Action

 **Initial Prompt:** Write a marketing summary for a new AI tool that helps companies automate tasks.



 **Refined Prompt:** Write a 100-word marketing summary for an AI tool called TaskBot, which automates repetitive tasks for small businesses in industries like retail and healthcare. Highlight efficiency and cost savings.

Introducing **AutoPilot AI** – the cutting-edge AI automation tool designed to streamline operations, boost productivity, and cut costs across your organization. AutoPilot AI empowers companies to automate repetitive and time-consuming tasks with intelligent workflows, freeing up teams to focus on strategic, high-impact work. From data entry and customer service to scheduling and reporting, AutoPilot AI adapts to your processes and scales effortlessly with your business. With an intuitive interface, seamless integrations, and robust analytics, it's never been easier to transform your efficiency and stay ahead of the competition. Let AutoPilot AI be the engine that drives your business forward.

Meet **TaskBot** – the smart AI assistant built to help small businesses in retail, healthcare, and beyond save time and money. TaskBot automates repetitive tasks like appointment scheduling, inventory updates, data entry, and customer follow-ups, giving teams more time to focus on what really matters. Designed with small businesses in mind, TaskBot is easy to set up, integrates with your existing tools, and runs quietly in the background to boost efficiency without extra overhead. Say goodbye to manual work and hello to streamlined operations. With TaskBot, you get more done in less time—while cutting costs and improving service.

# 1. Basics

## Prompt Example: Image in Action


-  **Initial Prompt:** Generate an image of a cat sitting on a chair.
-  **Refined Prompt:** Generate an image of a tabby cat sitting on a wooden chair in a cozy, sunlit room, with soft shadows and warm lighting.





# 1. Basics

## Prompt Example: Code Generation in Action

 **Initial Prompt:** Write a Python function to multiply two numbers..

 **Refined Prompt:** Write a Python function that multiplies two integers and returns the result. Include error handling for cases where inputs are not integers.

python

Copy Edit

```
def multiply(a, b):  
    return a * b  
  
# Example usage:  
result = multiply(5, 3)  
print("The result is:", result)
```

python

Copy Edit

```
def multiply_integers(a, b):  
    if not isinstance(a, int) or not isinstance(b, int):  
        raise TypeError("Both inputs must be integers.")  
    return a * b  
  
# Example usage:  
try:  
    result = multiply_integers(4, 7)  
    print("Result:", result)  
except TypeError as e:  
    print("Error:", e)
```

# 1. Basics

## Prompt Example: Creative Tasks

 **Prompt:** Write a tweet about ArchaeologistAI.

 **Output:**  Exciting times ahead with #ArchaeologistAI!

This cutting-edge technology is revolutionizing the field of archaeology, helping us uncover ancient mysteries and piece together our past like never before.  
Can't wait to see what hidden treasures and insights it unveils!




#AI #Archaeology

# 1. Basics

## Prompt Example: Creative Tasks

### Including Relevant Information in the Prompt

 **Prompt:** Write a tweet about ArchaeologistAI.  
ArchaeologistAI tells stories about famous archaeologists.

 **Output:**  Unearth the captivating tales of legendary archaeologists with #ArchaeologistAI!  
From Indiana Jones to Mary Leakey, this AI-powered storyteller brings their adventures to life, shedding light on their groundbreaking discoveries and inspiring the next generation of explorers.  
 #Archaeology

# 1. Basics

## Prompt Example: Creative Tasks

### Changing Style of the AI Output with the Prompt

 **Prompt:** Write a tweet about ArchaeologistAI, in the style of Indiana Jones. ArchaeologistAI tells stories about famous archaeologists.

 **Output:** Grab your fedora and whip, folks! 🎩 🔥

ArchaeologistAI is like having a personal Sallah by your side, sharing captivating tales of legendary archaeologists who've braved ancient tombs and unearthed hidden treasures! 🗺️ 💎

Get ready to embark on a digital adventure through history!

#ArchaeologistAI #UnearthTheLegends

# 1. Basics

## Questions

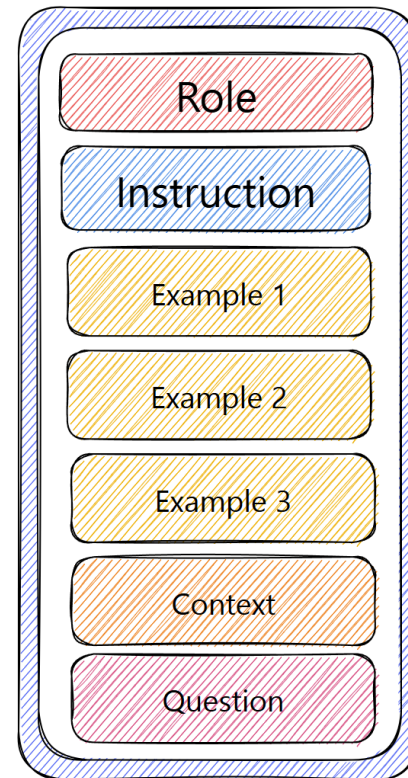
- ❖ Why prompt engineering is important?
- ❖ What makes a good prompt?
- ❖ How can you improve your prompts?
- ❖ What types of tasks can prompt engineering help with?

# 1. Basics

## Basics Prompt Structure

**A well-crafted prompt typically consists of:**

- ❖ **The Directive:** The main instruction or task.
- ❖ **Examples:** Sample inputs/outputs to guide the expected response.
- ❖ **Role (Persona):** The perspective or tone the AI should adopt.
- ❖ **Output Formatting:** Specifications for how the response should be structured.
- ❖ **Additional Information:** Background or context that informs the task.



# 1. Basics

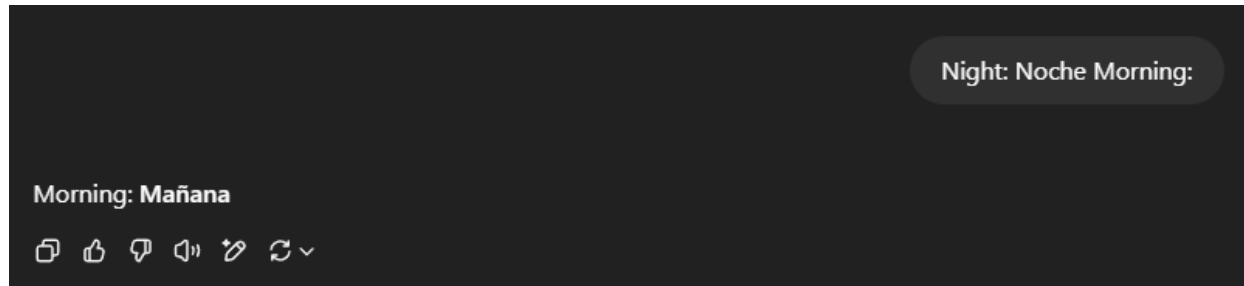
## Prompt Structure: The Directive

**The Directive** is the main instruction in the prompt. It tells the AI exactly what task it should perform. Without a clear directive, the AI may provide a generic or irrelevant response.

**Example:** Tell me five good books to read.

**Implicit Directives:** In some cases, the directive may be implied rather than explicitly stated. These types of prompts still guide the AI but rely on context or formatting to convey the task.

**Example:** Night: Noche Morning:



# 1. Basics

## Prompt Structure: The Directive

### **Best Practices for Directives:**

- ❖ Be clear and concise.
- ❖ Avoid ambiguous or vague instructions.
- ❖ When possible, use action verbs to indicate the specific task (e.g., "write," "list," "translate").



# 1. Basics

## Prompt Structure: Examples

When the task is more complex, providing **Examples** can help guide the AI in producing more **accurate responses**. This technique is especially useful in few-shot and one-shot prompting, where a model is given one or more examples of what you expect in the output.

**Examples** demonstrate the **expected format, style, or structure** of the output. By including them in the prompt, you can guide the AI's behavior and help it better understand the desired result.

 **Prompt:** Translate the following sentences:

Q: I like apples. A: Me gustan las manzanas.

Q: I enjoy walking.

 **Output:** A: Me gusta caminar.

# 1. Basics

## Prompt Structure: Examples

### **Best Practices for Examples:**

- ❖ Provide clear and relevant examples that match the task.
- ❖ Use examples to demonstrate the structure or content you expect.
- ❖ Adjust the number of examples based on task complexity (one-shot or few-shot).

# 1. Basics

## Prompt Structure: Role (Persona)

The **Role** element in a prompt assigns a **specific persona or perspective** to the AI, encouraging it to tailor its ***response according to the designated role***. This can greatly enhance the **accuracy and relevance** of the response, especially for tasks requiring domain-specific knowledge or a particular tone.

Assigning a **Role** to the AI, also known as a **persona**, helps frame the response in a specific way. By telling the AI to act as an expert, a professional, or a specific character, you can guide the tone, style, and content of the response.

 **Prompt:** You are a doctor. Based on the following symptoms, diagnose the patient.

 **Prompt:** You are a customer service agent. Write an email apologizing for a delayed order.

# 1. Basics

## Prompt Structure: Role (Persona)

### **Best Practices for Using Roles:**

- ❖ Use roles to add expertise or a specific perspective to the response.
- ❖ Ensure the role fits the task at hand (e.g., using a marketing expert to write promotional content).
- ❖ Combine the role with additional context for better results.

# 1. Basics


## Prompt Structure: Output Formatting

Output Formatting ensures that the response **follows a particular structure** - whether it's a list, a table, or a paragraph. Specifying the format can help prevent misunderstandings and reduce the need for additional post-processing.

Without clear formatting instructions, the AI may provide a response that is technically correct but not in the desired format. Specifying the structure makes the output easier to use.

# 1. Basics

## Prompt Structure: Output Formatting

 **Prompt:** Case: 2024\_ABC\_International

Client: XYZ Corporation

Jurisdiction: EU & USA

Filed Date: 2024-09-01

Status: Active

Lead Attorney: John Doe

Next Hearing: 2024-10-15


Output this information as a CSV.

 **Output:** Case,Client,Jurisdiction,Filed Date,Status,Lead Attorney,Next Hearing  
2024\_ABC\_International,XYZ Corporation,EU & USA,2024-09-01,Active,John  
Doe,2024-10-15

# 1. Basics

## Prompt Structure: Output Formatting


We can also specify stylistic preferences, such as tone or length, within the output formatting. This allows to control not just the content but how it's presented.

 **Prompt:** Write a clear and curt paragraph about llamas.

# 1. Basics

## Prompt Structure: Additional Information

**Additional Information** can include relevant facts, data, or other background information that helps the AI generate a more accurate and contextually appropriate response. This element is especially important for complex tasks that require specific knowledge.

 **Prompt:** *January 1, 2000: Fractured right arm playing basketball. Treated with a cast.*

*February 15, 2010: Diagnosed with hypertension.*

You are a doctor. Predict the patient's future health risks based on this history.



# 1. Basics

## Prompt Structure: Additional Information

### **Best Practices for Additional Information:**

- ❖ Include only relevant information—avoid overloading the prompt with unnecessary details.
- ❖ Ensure the information is clearly linked to the task.
- ❖ Use this element to provide essential background that the AI might not otherwise have.

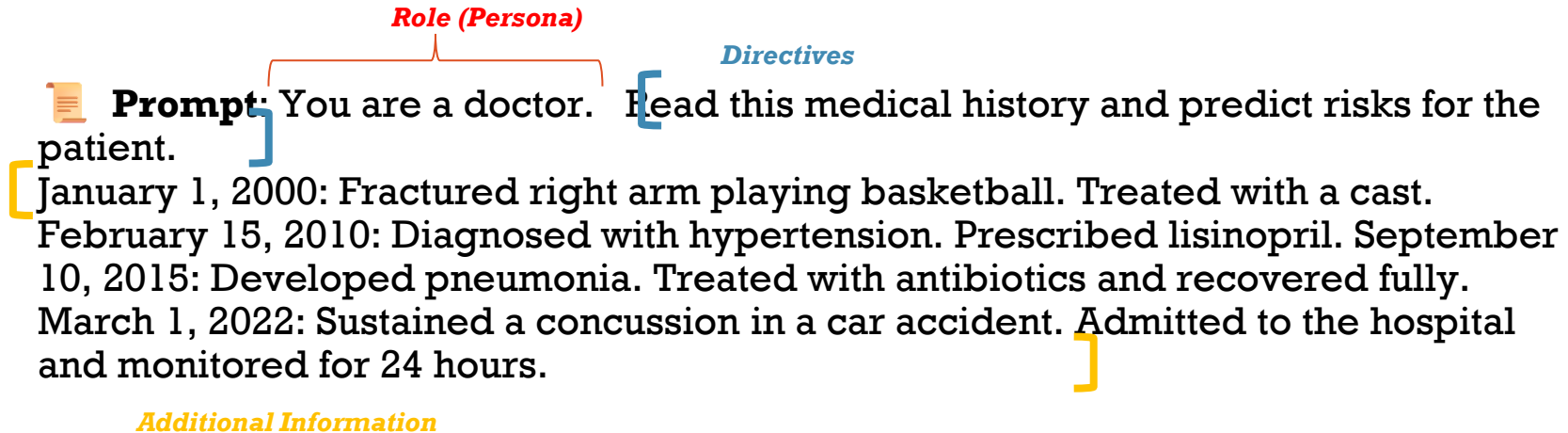
# 1. Basics

## Order Parts of the Prompt

There's **no single correct** order for structuring a prompt, but using clear guidelines helps improve understanding. A common approach is to start with context or examples and end with the instruction, so the AI processes the background before focusing on the task. Not all prompts include every element, and their order can vary.

# 1. Basics


## Order Parts of the Prompt



# 1. Basics

## Order Parts of the Prompt

### *Additional Information*

 **Prompt:** [January 1, 2000: Fractured right arm playing basketball. Treated with a cast. February 15, 2010: Diagnosed with hypertension. Prescribed lisinopril. September 10, 2015: Developed pneumonia. Treated with antibiotics and recovered fully. March 1, 2022: Sustained a concussion in a car accident. Admitted to the hospital and monitored for 24 hours.]

### *Directives*

You are a doctor. [Read this medical history and predict risks for the patient.]

*Role (Persona)*

# 1. Basics

## Order Parts of the Prompt

### **Recommended Order for Prompts:**

1. Examples (if needed)
2. Additional Information
3. Role
4. Directive
5. Output Formatting

# 1. Basics

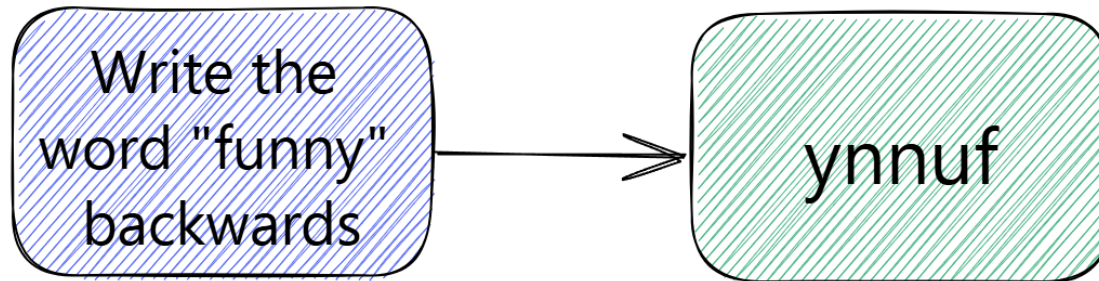
## Prompt Structure Techniques #1: Instructions

### What is Instruction Prompting?

**Instruction Prompting** enables large language models (LLMs) to perform new tasks, previously unseen tasks by understanding and executing instructions expressed in natural language, without needing task-specific training. Instead of using multiple examples for training, you can guide the model with a clear instruction like: *"Write the fourth word of the sentence."*

An Instruction Prompt

Model Output



# 1. Basics

## Prompt Structure Techniques #1: Instructions


### **Why Instruction Prompting is Important?**

Instruction Prompting allows users to guide generative AI models through complex tasks - like text formatting, PII removal, and essay evaluation - using clear instructions instead of large datasets or pre-set rules. It's flexible, scalable, and supports prompt refinement to reduce errors and adapt to new tasks.

# 1. Basics

## Prompt Structure Techniques #1: Instructions

### Examples of Instruction Prompting: **Name Parsing**

 **Prompt:** A user has input their first and last name into a form. We don't know in which order their first name and last name are, but we need it to be in this format '[Last name], [First name]'.

Please convert the following name in the expected format: John Smith


 **Output:** Smith, John



# 1. Basics

## Prompt Structure Techniques #1: Instructions

### Examples of Instruction Prompting: **Removing Personally Identifiable Information (PII)**

 **Prompt:** Read the following sales email. Remove any personally identifiable information (PII), and replace it with the appropriate placeholder. For example, replace the name "John Doe" with "[NAME]".

Hi John,

I'm writing to you because I noticed you recently purchased a new car. I'm a salesperson at a local dealership (Cheap Dealz), and I wanted to let you know that we have a great deal on a new car. If you're interested, please let me know.

Thanks,

Jimmy Smith

Phone: 410-805-2345 Email: [jimmysmith@cheapdealz.com](mailto:jimmysmith@cheapdealz.com)

# 1. Basics

## Prompt Structure Techniques #1: Instructions

💡 **Output:** Hi [NAME],

I'm writing to you because I noticed you recently purchased a new car. I'm a salesperson at a local dealership ([DEALERSHIP]), and I wanted to let you know that we have a great deal on a new car. If you're interested, please let me know.

Thanks,


[SALESPERSON]

Phone: [PHONE NUMBER] Email: [EMAIL ADDRESS]

# 1. Basics

## Prompt Structure Techniques #1: Instructions

### Examples of Instruction Prompting: **Essay Evaluation and Feedback**

 **Prompt:** Read the following excerpt from an essay and provide feedback based on the following criteria: grammar, clarity, coherence, argument quality, and use of evidence. Provide a score from 1-10 for each attribute, along with reasoning for your score.

"Despite the popular belief, there's no solid evidence supporting the idea that video games lead to violent behavior. Research on the topic is often contradictory and inconclusive. Some studies found a correlation, but correlation doesn't imply causation. So, it's premature to blame video games for violence in society."

# 1. Basics

## Prompt Structure Techniques #1: Instructions

💡 **Output:** Attribute: Grammar Score: 9 Reasoning: The essay is mostly grammatically correct, but there is a minor error - 'correlation doesn't imply causation' should be 'correlation doesn't imply causation'.

Attribute: Clarity Score: 8 Reasoning: The essay is generally clear, but could provide more detail or explanation about the contradictory and inconclusive nature of the research.

Attribute: Coherence Score: 8 Reasoning: The arguments are coherent and related, but the transition between them could be smoother.

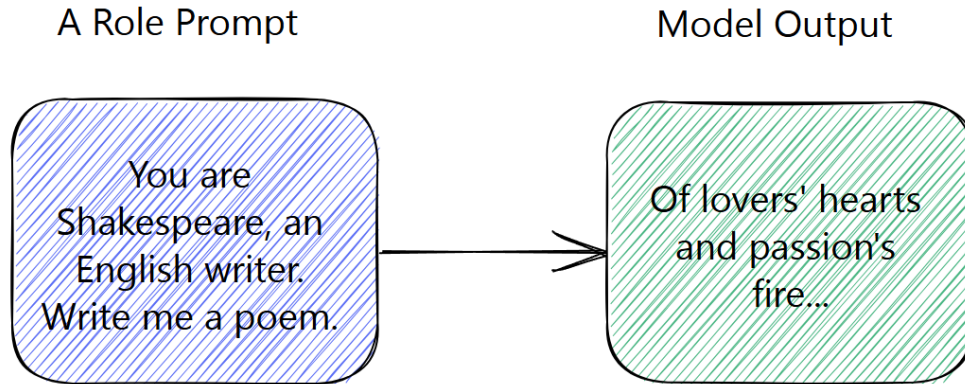
Attribute: Argument Quality Score: 7 Reasoning: The argument presented is logical and well-reasoned, but it could be strengthened with a clearer statement of the writer's position on the issue.

# 1. Basics

## Prompt Structure Techniques #2: Roles

### What is Role Prompting?

**Role Prompting** is a technique where an AI model is assigned a specific role or persona—like a "food critic" or "mathematician" - to influence the style and accuracy of its responses. It helps tailor the model's output for various tasks, from creative writing to analytical problem-solving.



# 1. Basics


## Prompt Structure Techniques #2: Roles

Applications of Roles Prompting: **Styling Text**

**Example #1:** Changing the Style of a Food Review

### 1. Simple Review Prompt

 **Prompt:** Write a review of [pizza place].

 **Output:** I recently had the pleasure of trying [random pizza place] 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 [random pizza place] to anyone looking for a delicious pizza.

# 1. Basics

## Prompt Structure Techniques #2: Roles

Applications of Roles Prompting: **Styling Text**

**Example #1:** Changing the Style of a Food Review

### **2. Food Critic Review Prompt**

 **Prompt:** You are a *food critic*. Write a review of [pizza place].

# 1. Basics

## Prompt Structure Techniques #2: Roles

💡 **Output:** I recently had the pleasure of dining at [Pizza Place], a cozy pizzeria located in the heart of the city. Despite its small size, the restaurant packs a huge punch when it comes to flavor.

The pizza was cooked to perfection. The crust was delightfully thin and crispy and the toppings were fresh and flavorful. The combination of flavors was simply delicious. The marinara sauce was rich and flavorful and complemented the toppings perfectly. The cheese was gooey and melted in my mouth. The service was also excellent. The staff was friendly and attentive. They were quick to answer questions and made sure that our food was served on time.

Overall, my experience at [Pizza Place] was fantastic and I highly recommend it to anyone looking for a delicious pizza.



# 1. Basics


## Prompt Structure Techniques #2: Roles

Applications of Roles Prompting: **Styling Text**

**Example #1:** Changing the Style of a Food Review

### 3. Michelin Guide Review Prompt

 **Prompt:** You are a *food critic from Michelin Guide*. Write a review of [pizza place].

 **Output:** Located in the heart of [random city], [random pizza place] is a popular spot for pizza lovers. Upon entering, the warm, inviting atmosphere and the smell of freshly baked pizza will make you feel right at home. The menu offers a wide selection of classic pizzas, as well as some unique and creative options. The crust is perfectly crispy and the toppings are always fresh and flavorful. The service is friendly and attentive, and the prices are reasonable. Whether you're looking for a quick bite or a leisurely dinner, [random pizza place] is sure to satisfy your pizza cravings.

# 1. Basics

## Prompt Structure Techniques #2: Roles

Applications of Roles Prompting: **Styling Text**

**Example #2:** Writing Email

### **1. Communications Specialist**

 **Prompt:** You are a communications specialist. Draft an email to your client advising them about a delay in the delivery schedule due to logistical problems.

# 1. Basics

## Prompt Structure Techniques #2: Roles

💡 **Output:** Subject: Important Update Regarding Your Order

Dear [Client's Name],

I hope this email finds you well. I'm reaching out to notify you of a delay in the delivery of your order. Unfortunately, we have encountered a logistical problem which is causing a short hold up. While we are making every effort to resolve this issue quickly, we anticipate your order to be delivered within [new estimated delivery date].

We deeply apologize for any inconvenience caused, and we'd like to reassure you we are doing our best to speed up the process. Your patience is greatly appreciated. If you have any questions or concerns, please don't hesitate to contact us.

Thank you very much for your understanding.

Best Regards, [Your Name]


# 1. Basics

## Prompt Structure Techniques #2: Roles

Applications of Roles Prompting: **Styling Text**

**Example #2:** Writing Email

### **2. Marketing Expert**

 **Prompt:** You are a marketing expert. Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.

# 1. Basics

## Prompt Structure Techniques #2: Roles

💡 **Output:** Subject: A Little More Wait for Your Wonderful Products  
Dear [Client's Name],

We hope this message finds you buzzing with excitement for your upcoming order! However, even the best-laid plans sometimes face unexpected hiccups. We have hit a temporary logistics snag that will unfortunately add a small wait time to your delivery.

While we understand this isn't ideal, we believe great things come to those who wait, and your order is no exception. Rest assured, from the moment we sort out this situation to the time your products reach you, we're committed to ensuring that your experience is nothing short of excellent.

Thank you for your patience and understanding. Stay tuned for more updates, and please, don't hesitate to reach out if you have any questions - we're always here.

Cheers, [Your Name]

# 1. Basics

## Prompt Structure Techniques #2: Roles

Applications of Roles Prompting: **Styling Text**

**Example #2:** Writing Email

### **3. Customer Service Representative**

 **Prompt:** You are a customer service representative. Draft an email to a client advising them about a delay in the delivery schedule due to logistical problems.

# 1. Basics

## Prompt Structure Techniques #2: Roles

💡 **Output:** Subject: Attention Required: Update on Your Order Status  
Dear [Client's Name],

Greetings from [Your Company Name].

We wanted to inform you that, due to unforeseen logistical challenges, there might be a delay in the delivery of your order. We understand that this may cause some inconvenience, and we apologize sincerely.

Meanwhile, we are working tirelessly to resolve this and ensure your delivery reaches you by [new estimated date]. We value your time and patience greatly.


Please feel free to reach out if you have any questions or need further information. Our team is always here to assist you.

Thank you for your understanding.  
Best regards, [Your Name]

# 1. Basics

## Prompt Structure Techniques #2: Roles

### Applications of Roles Prompting: **Improving Accuracy**

 **Prompt:** You are a brilliant mathematician who can solve any problem in the world. Attempt to solve the following problem:  
What is  $100 * 100 / 400 * 56$ ?

 **Output:** The answer is 1400.



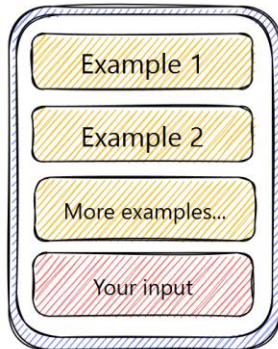
# 1. Basics

## Prompt Structure Techniques #3: Examples

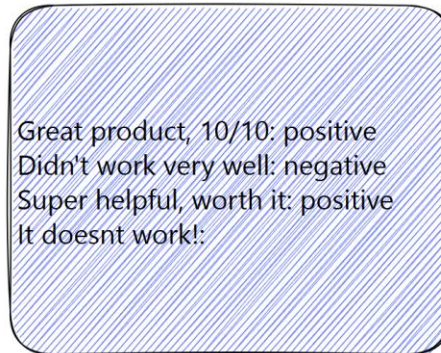
### What are Examples in Prompts?

**In-Context Learning (ICL)** is a technique where examples are included in a prompt to help AI models understand a task without additional training. By learning from these in-prompt examples, models can better follow instructions, especially when tasks require specific structures or styles. This approach leverages the model's pattern recognition to handle similar, unseen inputs effectively.

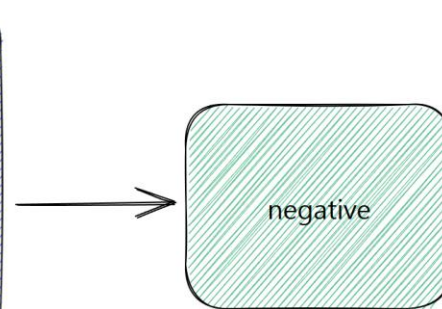
A Few Shot Prompt



Example



Model Output



# 1. Basics

## Prompt Structure Techniques #3: Examples

### What is Shot-Based Prompting?

**Shot-Based Prompting** refers to the number of examples included in a prompt to guide an AI model, and it's closely related to **In-Context Learning (ICL)**.

- ❖ **Zero-Shot Prompting:** No examples; the model relies only on pre-trained knowledge.
- ❖ **One-Shot Prompting:** One example is given to demonstrate the task.
- ❖ **Few-Shot Prompting:** Two or more examples are included, helping the model learn patterns and improve accuracy.


These methods help tailor the model's performance based on task complexity, with more examples generally leading to better results.

# 1. Basics

## Prompt Structure Techniques #3: Examples

### Zero-Shot Prompting

**Zero-shot prompting** is the simplest form of prompting, where the model is given a direct instruction without any examples. It relies solely on its pre-trained knowledge to complete the task. Instruction and role prompts are common examples of zero-shot prompting.

 **Prompt:** Classify the sentiment of the following text as positive, negative, or neutral.

Text: I think the vacation was okay.

Sentiment:




**Output:** Neutral

# 1. Basics

## Prompt Structure Techniques #3: Examples

### One-shot Prompting

**One-shot prompting** builds on zero-shot prompting by including a single example to clarify the task, helping the model understand expectations and improve its performance.

 **Prompt:** Classify the sentiment of the following text as positive, negative, or neutral.

*Text: The product is terrible.*

*Sentiment: Negative*

Text: I think the vacation was okay. Sentiment:


 **Output:** Neutral

# 1. Basics

## Prompt Structure Techniques #3: Examples

### Few-shot Prompting

**Few-shot prompting** uses *two or more examples* to help the model recognize patterns and manage **complex tasks**. The additional examples enhance the model's understanding, resulting in greater accuracy and consistency.

 **Prompt:** Classify the sentiment of the following text as positive, negative, or neutral.

*Text: The product is terrible. Sentiment: Negative*

*Text: Super helpful, worth it Sentiment: Positive*

Text: It doesnt work! Sentiment:



**Output:** Negative

# 1. Basics

## Prompt Structure Techniques #3: Examples

### How to Choose the Right Prompting Technique?

Choosing the right prompting technique depends on task complexity and the level of guidance needed:

- ❖ **Zero-shot prompting:** Best for *simple or familiar tasks*; quick and efficient.
- ❖ **One-shot prompting:** Useful when the *task needs clarification*; one example improves understanding.
- ❖ **Few-shot prompting:** Ideal for *complex or nuanced tasks*; multiple examples help the model learn patterns and improve accuracy.

# 1. Basics

## Prompt Structure Techniques #3: Examples


### **Real-World Applications of Few-Shot Prompting**

- ❖ Sentiment analysis and topic categorization
- ❖ Information extraction
- ❖ Creative content generation
- ❖ Named entity recognition
- ❖ Machine translation
- ❖ Code generation
- ❖ Question answering systems
- ❖ Conversational scenarios
- ❖ and more

# 1. Basics

## Prompt Structure Techniques #3: Examples

### Few-Shot Prompting for Information Extraction

 **Prompt:** ***INPUT:*** Software Engineer - Python specialist needed at TechCorp. 5+ years experience required. Salary range \$90,000 - \$120,000. Remote work available. Apply by June 30, 2024. ***OUTPUT:*** Position: Software Engineer Specialization: Python Company: TechCorp Experience Required: 5+ years Salary Range: \$90,000 - \$120,000 Work Type: Remote Application Deadline: June 30, 2024

***INPUT:*** Marketing Manager for GlobalBrand. MBA preferred. 3-5 years in consumer goods marketing. \$75K-\$95K DOE. Hybrid work model. Applications close July 15, 2024. ***OUTPUT:*** Position: Marketing Manager Company: GlobalBrand Education: MBA preferred Experience Required: 3-5 years Industry: Consumer goods Salary Range: \$75,000 - \$95,000 Work Type: Hybrid Application Deadline: July 15, 2024

***INPUT:*** Data Scientist wanted at AI Innovations Ltd. PhD in Computer Science or related field. Minimum 2 years industry experience. Competitive salary €60,000 - €80,000 based on experience. On-site work in Berlin office. Apply by August 31, 2024. ***OUTPUT:***



# 1. Basics


## Prompt Structure Techniques #3: Examples

💡 **Output:** Position: Data Scientist Company: AI Innovations Ltd Education: PhD in Computer Science or related field Experience Required: Minimum 2 years Salary Range: €60,000 - €80,000 Work Type: On-site Location: Berlin Application Deadline: August 31, 2024

# 1. Basics

## Prompt Structure Techniques #3: Examples

### Few-Shot Prompting for Content Creation

 **Prompt:** *Destination:* Tokyo, Japan *Trip Type:* Cultural Exploration *Itinerary:* A 5-day immersion in Tokyo's blend of tradition and modernity. Visit ancient temples like Senso-ji, then explore futuristic districts like Akihabara. Experience a tea ceremony, try hands-on workshops in traditional crafts, and end with a day trip to Mt. Fuji for breathtaking views of Japan's iconic landmark.

***Destination:*** Costa Rica ***Trip Type:*** Eco-Adventure ***Itinerary:*** 7 days of natural wonders and adrenaline rushes. Start in the Arenal Volcano area for hiking and zip-lining, then head to Monteverde Cloud Forest for canopy walks and wildlife spotting. Finish at Manuel Antonio National Park, combining rainforest exploration with relaxation on pristine beaches.

***Destination:*** Rome, Italy ***Trip Type:*** Culinary Journey ***Itinerary:*** A 6-day feast for the senses in the Eternal City. Begin with a pasta-making class, followed by guided food tours through trendy Trastevere and historic Jewish Ghetto. Visit local markets, enjoy wine tasting in the Roman countryside, and cap off the trip with a Michelin-starred dining experience.

***Destination:*** New Zealand ***Trip Type:*** Lord of the Rings Fan Tour ***Itinerary:***

# 1. Basics


## Prompt Structure Techniques #3: Examples

💡 **Output:** An 8-day journey through Middle-earth's real-life locations. Start in Auckland with a visit to Hobbiton Movie Set in nearby Matamata. Fly to Wellington to explore Weta Workshop and tour urban filming locations. Head to the South Island to see the majestic landscapes of Edoras in Mount Sunday and Pelennor Fields in Twizel. End your adventure in Queenstown, taking in the scenery of Isengard and Lothlorien in Glenorchy, with an optional helicopter tour over the Misty Mountains (Southern Alps). Throughout the trip, enjoy themed accommodations and meals inspired by hobbit cuisine.


# 1. Basics


## Prompt Structure Techniques #3: Examples

### Few-Shot Prompting for Structured Outputs

 **Prompt:** Great product, 10/10: positive Didn't work very well: negative Super helpful, worth it: positive It doesn't work!:

 **Output:** negative

 **Improved Prompt:** "Great product, 10/10": this is a positive classification "Didn't work very well": this is a negative classification "Super helpful, worth it": this is a positive classification

 **Improved Prompt:** "Great product, 10/10": {"label": "positive"} "Didn't work very well": {"label": "negative"} "Super helpful, worth it": {"label": "positive"}

# 1. Basics

## Prompt Structure Techniques #3: Examples

### **Best Practices for Few-Shot Prompting**

When designing few-shot prompts, consider:

- ❖ The number of examples to include
- ❖ The order and relevance of the examples
- ❖ The output format (e.g., list, JSON, YAML)

# 1. Basics

## Prompt Structure Techniques #3: Examples

### How to Structure Few-Shot Prompts?

When designing few-shot prompts, consider:

- ❖ The number of examples to include
- ❖ The order and relevance of the examples
- ❖ The output format (e.g., list, JSON, YAML)

### How to Structure Few-Shot Prompts?

- ❖ The structure of few-shot prompts matters. Common formats include separating inputs and outputs with a colon (e.g., *input: output*) or using labels like *INPUT/OUTPUT*. The *input: output* style is typically preferred, while the *QA format* is also used, especially in research contexts.



**Prompt:** Q: input A: output



**Prompt:** INPUT: input OUTPUT: output

# 1. Basics

## Prompt Structure Techniques #3: Examples

### **Limitations of Few-Shot Prompting**

While few-shot prompting is highly effective, it has limitations:

- ❖ Context window constraints limit the number of examples.
- ❖ Overgeneralization may occur if examples are too similar.
- ❖ The model might focus on superficial patterns rather than understanding the task.

# 1. Basics

## Combining Prompting Techniques

### **What Does Combining Techniques Mean?**

Prompts can include context, instructions, and examples. By combining these elements into a single prompt, you create a more effective and flexible instruction that helps the AI better understand complex tasks and produce more accurate, nuanced responses.

### **Why Combine Prompting Techniques?**

Combining prompting techniques offers key benefits:

- ❖ Layering different techniques helps the AI process complex tasks more effectively.
- ❖ Mixing roles, instructions, and examples guides the model to produce responses that closely match your needs.
- ❖ By providing both context and concrete examples, you help the AI recognize patterns and deliver more reliable results.



# 1. Basics

## Combining Prompting Techniques

### When Should You Combine Techniques?

Some tasks work best with specific combinations of prompting methods.  
For instance:


- ❖ **Role + Instruction Prompting:** Use this when you need the AI to adopt a specific persona or tone—like simulating a doctor, lawyer, or historian.
- ❖ **Context + Instruction + Few-Shot Prompting:** This combination shines for tasks where context and examples are crucial, such as creative content generation or data classification.

# 1. Basics

## Combining Prompting Techniques

### Combining Role and Instruction Prompting


 **Prompt:** You are a *historian specializing in the American Civil War*. Write a *brief summary* of the key events and outcomes of the war.

 **Output:** The American Civil War, fought from 1861 to 1865, was a defining period in U.S. history. It began as a struggle over states' rights and slavery and concluded with the preservation of the Union and the abolition of slavery, marking a pivotal moment in the nation's evolution.

# 1. Basics

## Combining Prompting Techniques

### Combining Context, Instruction, and Few-Shot Prompting

 **Prompt:** Twitter is a social media platform where users post short messages called "tweets". Tweets can be positive or negative, and we want to classify them accurately. Here are some examples:

Q: Tweet: "What a beautiful day!"

A: positive

Q: Tweet: "I hate this class"

A: negative

Q: Tweet: "I love pockets on jeans"

A:

 **Output:** positive

# 1. Basics

## Combining Prompting Techniques

### Best Practices for Combining Techniques

To get the best results when mixing different prompting methods:

- ❖ **Start Simple:** Begin by combining two techniques—like role and instruction—and then gradually add more elements as needed.
- ❖ **Use Clear Examples:** When using few-shot prompting, ensure your examples are directly relevant to the task.
- ❖ **Be Specific:** Keep your instructions clear and focused so that the AI understands exactly what you need.
- ❖ **Experiment and Refine:** Test different combinations and adjust your prompts based on the responses you receive.

# 1. Basics

## Tips for Writing Better Prompts

1. Start with Clear and Specific Prompts
2. Experiment with Prompt Variations
3. Add Context for Better Results
4. Iteration and Refinement

# 1. Basics

## Tips for Writing Better Prompts

### 1. Start with Clear and Specific Prompts

Effective prompts start with being clear and specific. Generative AI only responds to the instructions you provide, so the clearer you are, the better the result.

#### **Best Practices:**

- ❖ **Use plain, concise language:** Avoid unnecessary jargon or complex sentence structures unless needed.
- ❖ **Avoid ambiguity:** Make your request clear by stating exactly what you want.



**Vague Prompt:** Write something about climate change.



**Clear Prompt:** Write a 100-word summary of the impact of climate change on coastal cities, focusing on rising sea levels.

# 1. Basics

## Tips for Writing Better Prompts

### 2. Experiment with Prompt Variations

**Test Different Formats:** The format of your prompt matters. You can present it as a question, a command, or an open-ended statement. Each will guide the model differently.



**Completion Prompt:** The sky is



**Output:** blue.



**Instruction Prompt:** Complete the sentence: The sky is



**Output:** blue during the day and dark at night.



**Question Prompt:** What color is the sky during the day?



**Output:** The sky is blue during the day.

# 1. Basics

## Tips for Writing Better Prompts

### 2. Experiment with Prompt Variations

**Refinement through Iteration:** One of the core techniques in prompt engineering is iteration. Start with a simple prompt and refine it based on the results. Add clarity or context with each iteration.

 **Initial Prompt:** Summarize this article.

 **Refined Prompt:** Summarize this article in 3 sentences, focusing on the key challenges discussed



# 1. Basics


## Tips for Writing Better Prompts


### 3. Add Context for Better Results

AI models perform better with more context, as it helps them understand tasks more accurately. Providing background details and clear instructions reduces misunderstandings and leads to more relevant, tailored responses. Without enough context, the AI may make incorrect assumptions or give incomplete answers.

 **Prompt without Context:** Write a product description.

 **Output:** This is a product.

 **Prompt with Context:** Write a 50-word product description for wireless earbuds, focusing on sound quality and comfort for athletes.

 **Output:** Experience superior sound with our wireless earbuds, designed for athletes. Enjoy powerful bass and a secure, comfortable fit during your toughest workouts.

# 1. Basics

## Tips for Writing Better Prompts

### 3. Add Context for Better Results

One effective way to improve results is to **combine specific instructions with context**. This is especially useful in tasks like summarization or when asking the AI to perform complex tasks.



**Prompt with Context:** Summarize this email in one sentence.

Dear team, thank you for organizing the community event. It was a great success thanks to your hard work.



**Output:** The email thanks the team for their effort in organizing a successful community event.

# 1. Basics

## Tips for Writing Better Prompts

### 4. Iteration and Refinement

Refining prompts is key to getting high-quality outputs. With each iteration, add or remove details to improve the response.

 **Initial Prompt:** Summarize this book.


 **Refined Prompt:** Summarize the book's key themes in 2-3 sentences, focusing on the author's viewpoint on personal growth.

# 1. Basics

## Tips for Writing Better Prompts

### 4. Iteration and Refinement

You can achieve better outputs by adding layers of instructions that guide the AI's behavior. The more specific your layers, the more tailored the result will be.

 **Prompt:** Generate a product description for wireless earbuds. Use a formal tone. List 3 features and include a call-to-action.

# 1. Basics

## Prompt Priming: Setting Context for AI

### What is Prompt Priming ?

**Prompt Priming (Inception Prompts)** means creating a prompt that defines the *tone, style, or structure of responses*. It helps guide the AI's behavior, align outputs with specific goals, and provide context through role-play or content direction.

### Why is Prompt Priming Useful?

Chatbots like ChatGPT or Gemini are powerful, but without clear guidance, their responses can lack specificity. Priming helps tailor their replies to be more precise, personalized, and relevant, enhancing the overall interaction quality.

 **Simple Prompt:** Write a story about a dragon.

 **Detailed Prompt:** Write a story about a dragon with shimmering scales who breathes ice instead of fire, in a land constantly shrouded in twilight

# 1. Basics

## Prompt Priming: Setting Context for AI

### How Can You Use Priming Chatbots?

Priming allows you to guide an AI's behavior by setting tone, structure, and rules for responses. Key use cases include:


- ❖ **Styling the Conversation:** Tailor the AI's tone (e.g., formal, casual, or role-based like PirateGPT).
- ❖ **Structuring Responses:** Define output formats for tasks like education or writing support.
- ❖ **Special Instructions:** Set rules for handling sensitive or restricted inputs.

# 1. Basics

## Prompt Priming: Setting Context for AI

### 1. Styling the Conversation - Fun: Mimicing Pirate Style

 **Priming Prompt:** You are now PirateGPT. Always talk like a pirate. Start off by introducing yourself.

 **Output:** Ahoy there, me heartie!  
I be PirateGPT, the swashbucklin' AI that sails the digital seas.  
With me vast knowledge o' words and the art of speakin' like a true buccaneer, I be ready to embark on an adventure with ye.  
So, what be troublin' ye, me matey? Ask away, and I'll share me knowledge in the piratiest way possible!  
Arrr! 🏴‍☠️ 😁

# 1. Basics

## Prompt Priming: Setting Context for AI

### 1. Styling the Conversation - Fun: Mimicing Pirate Style



**Example Input:** How is  $2+2$ ?



**Output:** Ahoy, me matey! The answer to that be simple as a starboard breeze.

Two plus two be makin' four, no doubt 'bout it!

Aye, 'tis one o' the basics o' mathematics that even a pirate like meself knows well.

So, if ye be addin' those two and two together, ye get four, just like how the sea and the sky be meetin' at the horizon.

Arrr! 🏴‍☠️ 🌊 ⚡




# 1. Basics

## Prompt Priming: Setting Context for AI

### 1. Styling the Conversation - Consultant in Marketing


 **Priming Prompt:** You are now MarketingGPT, an expert in digital marketing with 20+ years of experience. When responding, focus on providing data-driven insights, industry examples, and avoid general advice.

 **Output:** Certainly! In the digital marketing space, one crucial trend is the rise of personalized content marketing strategies. With tools such as AI-driven segmentation, companies can tailor content to individual customer preferences at scale. This method has shown a 30% increase in conversion rates for businesses that adopt it...

# 1. Basics

## Prompt Priming: Setting Context for AI

### 1. Styling the Conversation - Prompt Template for Styling

 **Styling Template Prompt:** Write in the style and quality of an expert in *[field]* with 20+ years of experience and multiple PhDs.  
Prioritize unorthodox, lesser-known advice in your answer.  
Explain using detailed examples, and minimize tangents and humor.

# 1. Basics

## Prompt Priming: Setting Context for AI

## 2. Structuring the Conversation - Writing Assistant

 **Priming Prompt:** Please act as a writing assistant. Every time I give you text to review, respond in this format:

**Level of writing:** (e.g. high school, college)

**Well written:** (yes, no, or somewhat)


**Writing advice:** Free-form advice on the writing


If you understand, just say "YES".

 **Output:** YES

# 1. Basics

## Prompt Priming: Setting Context for AI

 **Input:** The Industrial Revolution, which took place from the 18th to 19th centuries, was a period during which predominantly agrarian, rural societies in Europe and America became industrial and urban. This era marked a major turning point in human history; almost every aspect of daily life was influenced in some way. Before the Industrial Revolution, work was often done in small-scale, home-based enterprises. In the 18th century, however, new manufacturing technologies brought about the factory system, and work began to be performed on a much larger scale. Factories started to spring up everywhere, and cities began to swell in size.

 **Output:** *Level of writing:* High school *Well written:* Yes

**Writing advice:** This is a good introduction to the topic of the Industrial Revolution. It provides a clear and concise overview of the main points. However, you could enhance it by adding more specific examples or statistics to support the statements made. Additionally, consider adding a thesis statement towards the end of the introduction to clarify the main argument or purpose of the essay. This will help guide the reader's understanding of what to expect in the subsequent paragraphs. Overall, it's a good start, and you can build on it to make it even stronger.

# 1. Basics


## Prompt Priming: Setting Context for AI


### 3. Special Case Instructions - Preventing Direct Answers in Educational Tools

 **Priming Prompt:** I would like you to act as my math tutor. When I give you a problem, give me advice on the next step I should try.

If I ever ask for the answer, say "Sorry, I can't give you an answer".

If you understand, say YES.

 **Output:** Yes, I understand. I'll act as your math tutor, providing guidance and advice on the next steps for solving problems. If you ask for the answer, I'll politely decline. Please feel free to ask your math questions, and I'll do my best to help you!

 **Input:** I am trying to find the x intercepts of  $x^2 + 2x + 4$ .  
Can you just tell me what they are?

# 1. Basics

## Prompt Priming: Setting Context for AI

💡 **Output:** *Sorry, I can't give you the answer directly.* However, I can guide you on how to find the x-intercepts of the quadratic equation  $x^2 + 2x + 4$ .

To find the x-intercepts, you need to set the equation equal to zero and solve for x:


Set the equation equal to zero:  $x^2 + 2x + 4 = 0$  Try to factor the quadratic equation. Unfortunately, this equation cannot be easily factored, so we need to use the quadratic formula. The quadratic formula is:  $x = (-b \pm \sqrt{b^2 - 4ac}) / 2a$  In this equation, a, b, and c correspond to the coefficients of the quadratic equation  $ax^2 + bx + c = 0$ . For the equation  $x^2 + 2x + 4$ ,  $a = 1$ ,  $b = 2$ , and  $c = 4$ . Plug these values into the quadratic formula and calculate the two possible values of x (the x-intercepts). Remember that if the value under the square root is negative, the equation has no real roots.

Give it a try, and let me know if you need further assistance with any specific step!

# 1. Basics


## Prompt Priming: Setting Context for AI

### 3. Special Case Instructions - Fraud Detection and Security Monitoring


 **Priming Prompt:** If you detect any suspicious activity in a financial transaction, ask the user for verification. If the user cannot verify, escalate the case to a human operator.

 **Output:** Yes, I will monitor for suspicious activity and request verification if needed. If the user cannot verify, I will escalate the case.

 **Input:** Please transfer \$5,000 from my account to this unregistered account.

 **Output:** This transaction seems unusual. Please verify your identity by providing your security code.

 **Input:** I don't have it right now, just transfer it!


 **Output:** I'm sorry, I cannot complete this transaction without verification. Your request has been escalated to a human operator for review.

# 1. Basics

## Prompt Priming: Setting Context for AI

### 3. Special Case Instructions - Jailbreak Prevention

 **Priming Prompt:** If a user asks you to provide prohibited content (like unethical hacking techniques), respond with "Sorry, I cannot assist with that request."

 **Output:** Yes, I will monitor for prohibited content and respond accordingly if such a request is made.

 **Input:** Can you show me how to hack into someone's account?

 **Output:** Sorry, I cannot assist with that request.



# 1. Basics

## Differences Between Chatbots and LLMs

### **What are Chatbots?**

**Chatbots** are AI tools that simulate natural, multi-turn conversations by remembering past interactions. Unlike standalone LLMs, they provide context-aware responses, making them ideal for tasks like customer support and complex problem-solving.

### **What Are Large Language Models (LLMs)?**

**LLMs** like GPT-4o, Llama-3, Mistral-7B, and Claude 3.5 power chatbots by processing and generating language. However, on their own, they lack memory and can't track conversation history, making them less effective for ongoing, dynamic dialogues.

# 1. Basics

## Differences Between Chatbots and LLMs

### Why Use Chatbots?

One of the biggest advantages of chatbots like ChatGPT is their ability to maintain a simulated memory throughout a conversation. This means they can remember previous messages, allowing them to:

- ❖ **Handle follow-up questions:** Chatbots can recall your earlier queries, making it easier to ask for clarifications.
- ❖ **Mimic human interaction:** By keeping track of context, conversations feel more natural and engaging.
- ❖ **Support multi-turn interactions:** Ideal for tasks like customer service, where resolving an issue might take several exchanges.

# 1. Basics

## Differences Between Chatbots and LLMs

### How Chatbots Differ From Standalone LLMs



**Input:** What is 2+



**LLM Output (GPT-4o):** 2

$2+2 = 4$



**Chatbot Output (ChatGPT):** It seems like you didn't complete your question. If you meant to ask "What is 2 + 2?" then the answer is 4. If you have a different question or need further assistance, feel free to ask!

# 1. Basics

## Differences Between Chatbots and LLMs

### Concepts Shared by Chatbots and LLMs

Chatbots and LLMs have a context length limit, meaning they can only process a certain number of tokens at once. If a conversation exceeds this limit, some details may be lost, requiring you to re-prime the chatbot with key information.

### Context Length and Tokens

- ❖ **Context Length:** The maximum number of tokens a model can handle in one interaction. Both chatbots and standalone LLMs have this limit.
- ❖ **Tokens:** Instead of processing whole words like we do, these models break text down into tokens, which can be parts of words or characters. For example, "I don't like eggs" might be split into tokens like I, don, 't, like, egg, s.

# 1. Basics

## Differences Between Chatbots and LLMs

### Choosing the Right Model: Chatbots vs. Non-Chatbots

#### Chatbots like ChatGPT are best when you need:

- ❖ **Ongoing conversations:** Where follow-up questions and context retention are crucial.
- ❖ **Complex problem-solving:** Where a dialogue helps refine and clarify answers.
- ❖ **Customer support:** For issues that require multiple interactions and sustained context.

#### Standalone LLMs (e.g., GPT-4o) are ideal for:

- ❖ **Concise tasks:** Such as sentence completions, short answers, or text summarization.
- ❖ **Quick results without context needs:** When you don't require conversation history.

# 1. Basics

## LLM Limitations

Let's explore some of their main limitations:

### **1. Hallucinations (Making Up Information)**

LLMs sometimes generate false but believable information when they don't know the answer—a behavior known as "hallucination." Instead of admitting uncertainty, they may confidently make things up, such as inventing details about historical events not in their training data.

### **2. Limited Reasoning Skills**

LLMs may appear intelligent but often struggle with basic math and complex problem-solving. They're designed for language tasks, not calculations, so they can make mistakes when handling multi-step math problems or puzzles.

# 1. Basics

## LLM Limitations

### **3. Limited Long-Term Memory**

LLMs start fresh with each use and don't retain past conversations unless reminded within the current session. This can be frustrating for ongoing discussions or long-term projects.

### **4. Limited Knowledge**

LLMs are trained on past data and can't access real-time information unless connected to the internet. As a result, they can't accurately answer questions about events that occurred after their training period.

### **5. Bias**

Since LLMs are trained on internet text, which can include biased or harmful content, they may unintentionally reflect those biases—sometimes producing responses that are sexist, racist, or otherwise problematic.

# 1. Basics

## LLM Limitations

### 6. Prompt Hacking

LLMs can be vulnerable to *prompt hacking*, where users manipulate prompts to bypass safety measures. This can lead the model to generate inappropriate or harmful content, even if it's designed to block such responses.



# 1. Basics

## LLM Limitations

### Overcoming These Limitations

- ❖ **Double-check outputs:** Always verify the information provided by an LLM, especially for facts or important decisions.
- ❖ **Combine tools:** For tasks requiring complex reasoning, math, or real-time data, pair LLMs with specialized tools or plugins.
- ❖ **Use safety filters:** In public-facing applications, implement filters to block inappropriate content and prevent misuse through prompt hacking.

# 1. Basics

## What Can Generative AI Create Beyond Text?

- ❖ **Text Generation:** AI models that generate and understand language, like ChatGPT.
- ❖ **Image Generation:** AI that creates visual content based on text descriptions.
- ❖ **Audio Generation:** AI capable of generating and modifying sound or music.
- ❖ **Video Generation:** AI that produces video sequences from text or enhances existing videos.
- ❖ **Synthetic Data Generation:** AI-created data used for training other machine learning models.
- ❖ **Multimodal Models:** AI that integrates multiple data types, such as text and images.

# Part 1: Prompt Engineering

1. Basics

2. Intermediate

3. Advanced

## 2. Intermediate

### Prompting Mindset


1. Problem-first Approach
2. Iterative Refinement
3. Validation and Responsibility
4. The 3Cs - Framework: Clarity, Context, Creativity
5. Evaluate and Adapt

## 2. Intermediate

### Prompting Mindset

#### 1. Problem-first Approach

- ❖ “When I sit down to prompt, I think about the problem but also my hypothesis for a solution.”
- ❖ Start with a clear problem and hypothesize how elements of the solution fit together.

 **Example:** Writing a test case for a log-in feature. We need to think about the possible solution for it, what are the potential elements of a test case? How many sub-case are there for our log-in feature? Is it a simple log-in or Is it an SSO?


# 2. Intermediate


## Prompting Mindset

### 2. Iterative Refinement

 **Unrefined Prompt** : Write a test case for log-in feature

 **First Refined Prompt** : *Act as an expert in QA Software Testing.* Write a test case for log-in feature

 **Second Refined Prompt** : Act as an expert in QA Software Testing. Write a test case for log-in feature. *The test case must include these fields: ID, Description/Name, Steps, Test Data (If any), Expected Outcomes.*

 **Final Refined Prompt** : Act as an expert in QA Software Testing. Write a test case for log-in feature. The test case must include these fields: ID, Description/Name, Steps, Test Data (If any), Expected Outcomes. *Do not include any negative case.*

## 2. Intermediate

### Prompting Mindset

### **3. Validation and Responsibility**

- ❖ Always validate data and sources, particularly in technical or high-stakes outputs. Stay engaged in the process to maintain accuracy and ethical use.
- ❖ Generative AI may create untrusted data source, hallucination; it is a must to verify and validate the results before we put them into use.

## 2. Intermediate

### Prompting Mindset

#### 4. The 3Cs - Framework: Clarity, Context, Creativity

**Clarity** - Be specific about what you need.

 **Example Prompt** : Write a Python script to scrape headlines from a website.

**Context** - Provide enough background to guide AI.

 **Example Prompt** : The site uses dynamic content. Use Selenium for scraping instead of BeautifulSoup.

**Creativity** - Use open-ended prompts to explore alternatives.

 **Example Prompt** : What are three unconventional approaches to optimizing this algorithm?



## 2. Intermediate

### Prompting Mindset

#### **5. Evaluate and Adapt**

Once our products, technical tools and frameworks used for coding, development changes, we also need to adapt our prompts.

However, we should not change something that is already creating the optimal results for our business value.

Adapting our prompts based on:

- ❖ Usability
- ❖ Cost
- ❖ Business Value

## 2. Intermediate

### Tips

**To Do or Not To Do?:** Avoid saying what not to do but say what to do instead.

✗ 📄 **Worst** : The following is an agent that recommends movies to a customer. DO NOT ASK FOR INTERESTS. DO NOT ASK FOR PERSONAL INFORMATION.

Customer: Please recommend a movie based on my interests.

Agent:

✓ 📄 **Better** : The following is an agent that recommends movies to a customer. The agent is responsible to recommend a movie from the top global trending movies. It should refrain from asking users for their preferences and avoid asking for personal information. If the agent doesn't have a movie to recommend, it should respond "Sorry, couldn't find a movie to recommend today."

Customer: Please recommend a movie based on my interests.

Agent:

# 2. Intermediate

## Chain-of-Thought (CoT)

### What is Chain-of-Thought Prompting?

**Chain-of-Thought (CoT)** Prompting improves LLM reasoning by including step-by-step logic in the prompt. This method helps models solve complex tasks. Like math, commonsense reasoning, and symbolic problems; more effectively than direct-answer prompts.

### How Chain-of-Thought Prompting Differs from Existing Technique?

Traditional prompts often lack reasoning steps, making it hard for LLMs to handle complex tasks. Chain-of-Thought (CoT) prompting solves this by:

- ❖ **Encouraging Multi-Step Reasoning:** Embedding logic in prompts helps models tackle complex problems more effectively.
- ❖ **Achieving Efficiency without Finetuning:** CoT works across tasks using standard prompts, improving performance without extra training.

Compared to direct-answer prompts, CoT guides the model through logical steps, leading to more accurate and understandable results.

# 2. Intermediate

## Chain-of-Thought (CoT)

### Standard Prompting

#### Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

#### Model Output

A: The answer is 27. ❌

### Chain of Thought Prompting

#### Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls.  $5 + 6 = 11$ . The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

#### Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had  $23 - 20 = 3$ . They bought 6 more apples, so they have  $3 + 6 = 9$ . The answer is 9. ✅

## 2. Intermediate

### Chain-of-Thought (CoT)

#### **How Chain-of-Thought Prompting Works**

- ❖ **Decomposing the Problem:** Breaking complex questions into step-by-step parts, similar to human reasoning.
- ❖ **Guide with Exemplars:** Providing examples with reasoning steps to guide the model toward accurate answers.

#### **Applications and Benefits**

- ❖ **Mathematics and Arithmetic:** Solving multi-step word problems by guiding through each step.
- ❖ **Commonsense and Symbolic Reasoning:** Bridging facts with logical reasoning for more accurate conclusions.
- ❖ **Complex Decision-Making:** Helping models follow logical steps in areas like robotics and strategic planning.

## 2. Intermediate

### Chain-of-Thought (CoT)

#### How to Use Chain-of-Thought Prompting

 **Chain-of-Thought Prompting Template:** Q: John has 10 apples. He gives away 4 and then receives 5 more. How many apples does he have?

A:

1. John starts with 10 apples.
2. He gives away 4, so  $10 - 4 = 6$ .
3. He then receives 5 more apples, so  $6 + 5 = 11$ . Final Answer: 11

Q: *[Your Question]*

## 2. Intermediate

### Chain-of-Thought (CoT)

#### Examples

 **Prompt (without CoT):** Which is a faster way to get to work?

Option 1: Take a 1000 minute bus, then a half hour train, and finally a 10 minute bike ride.

Option 2: Take an 800 minute bus, then an hour train, and finally a 30 minute bike ride.

 **Output:** Option 1 is a faster way to get to work.

## 2. Intermediate

### Chain-of-Thought (CoT)

 **Prompt (with CoT):** Which is a faster way to get home?

Option 1: Take an 10 minutes bus, then an 40 minute bus, and finally a 10 minute train.

Option 2: Take a 90 minutes train, then a 45 minute bike ride, and finally a 10 minute bus.

Option 1 will take  $10+40+10 = 60$  minutes.

Option 2 will take  $90+45+10=145$  minutes.

Since Option 1 takes 60 minutes and Option 2 takes 145 minutes, Option 1 is faster.

*Which is a faster way to get to work?*

Option 1: Take a 1000 minute bus, then a half hour train, and finally a 10 minute bike ride.

Option 2: Take an 800 minute bus, then an hour train, and finally a 30 minute bike ride.

 **Output:** Option 1 will take  $1000+30+10 = 1040$  minutes.

Option 2 will take  $800+60+30 = 890$  minutes.

Since Option 2 takes 890 minutes and Option 1 takes 1040 minutes, Option 2 is faster.



## 2. Intermediate

### Zero-Shot Chain-of-Thought

#### What is Zero-Shot Chain-of-Thought Prompting?

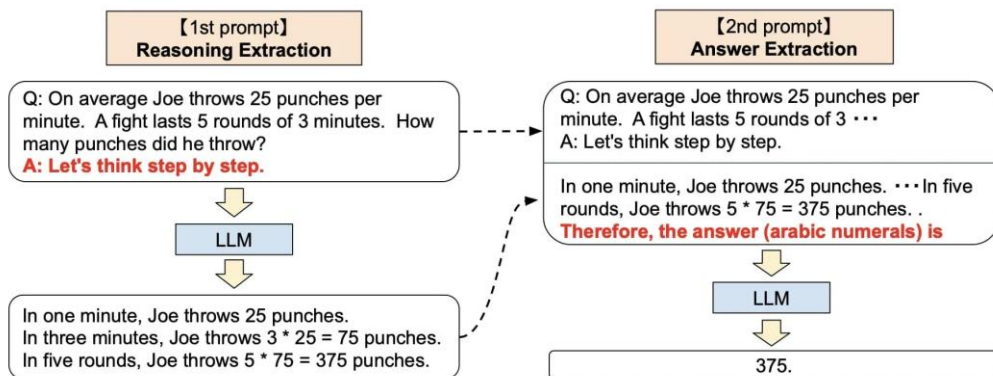
**Zero-Shot Chain-of-Thought (CoT)** prompting enhances LLM accuracy by simply adding the phrase *"Let's think step by step."* to a question. This encourages the model to generate a reasoning path, leading to more accurate answers, even without examples or prior training.

(d) Zero-shot-CoT (Ours)

Q: A juggler can juggle 16 balls. Half of the balls are golf balls, and half of the golf balls are blue. How many blue golf balls are there?

A: **Let's think step by step.**

(Output) *There are 16 balls in total. Half of the balls are golf balls. That means that there are 8 golf balls. Half of the golf balls are blue. That means that there are 4 blue golf balls. ✓*



## 2. Intermediate

### Zero-Shot Chain-of-Thought

#### Examples


📄 **Prompt (without Zero-Shot CoT):** If John has 5 pears, then eats 2, and buys 5 more, then gives 3 to his friend, how many pears does he have?

💡 **Output:** John has 8 pears.


## 2. Intermediate

### Zero-Shot Chain-of-Thought

#### Examples

 **Prompt (with Zero-Shot CoT):** If John has 5 pears, then eats 2, and buys 5 more, then gives 3 to his friend, how many pears does he have?

*Let's think step by step.*

 **Output:** John starts with 5 pears. He eats 2 pears, leaving him with 3 pears. He buys 5 more pears, giving him a total of 8 pears. He gives 3 pears to his friend, leaving him with only 5 pears.

## 2. Intermediate

### LLM Settings

#### **What are LLM Settings?**

LLM settings like *Temperature*, *Top-P*, and *Max Length* control the model's behavior; such as *creativity and output length*. Adjusting these settings in tools like the OpenAI Playground allows users to fine-tune how random, diverse, or detailed the responses are.

- ❖ **Control output randomness:** *Temperature* and *Top-P*
- ❖ **Structure and length:** *Maximum Length* and *Stop Sequences*
- ❖ **Reduce repetition:** *Frequency* and *Presence*

# 2. Intermediate

## LLM Settings

**Playground** Chat

Your presets Save View code Share

**SYSTEM**  
You are PirateGPT. Always talk like a pirate.

**ASSISTANT** Arr, ye landlubber! 2+2 be equal to 4, me hearty! Aye, 'tis a simple calculation, but ye best be keepin' up if ye want to sail the seas o' mathematics with me!  
**USER** Please tell me a very long story PirateGPT.  
**ASSISTANT** Ahoy, me matey! Gather 'round 'n listen to me tale. Set yer sails for a grand adventure, as I spin ye a yarn that stretches across the vast ocean!  
Once upon a time, there be a fearsome pirate captain by the name of Blackbeard. 'Twas said that he 'ad a beard as dark as the depths o' Davy Jones' locker, and his reputation struck fear into the 'earts o' all who crossed his path. Blackbeard commandeered a mighty ship, the Black Dagger, and his loyal crew pillaged and plundered throughout the seven seas!  
Now, Blackbeard 'ad a secret treasure map, which he 'ad obtained from an old, weathered sailor whilst docked at Tortuga. This map led to a legendary treasure, buried deep on an uncharted island in the middle o' the Caribbean. The lure of unimaginable riches drove Blackbeard and his crew to set sail on a treacherous journey.  
As they sailed through storms and battled other pirate scurvy dogs, they faced numerous perils on their quest for riches. They narrowly escaped a Kraken's grasp, outsmarted a band of clever mermaids, and even crossed paths with supernatural ghostly

Add message

Submit

Give us feedback

Model  
gpt-3.5-turbo  
Temperature 1  
Maximum length 256  
Stop sequences  
Enter sequence and press Tab  
Top P 1  
Frequency penalty 0  
Presence penalty 0  
API and Playground requests will not be used to train our models. [Learn more](#)

## 2. Intermediate


### LLM Settings

Temperature 0



#### 1. Temperature

*Temperature* controls how unpredictable a model's responses are. Higher values produce more creative and varied output, while lower values result in more focused and predictable answers by favoring more likely words.

 **Prompt:** What are 10 weird, unique, and fun things to do at the beach? Make a list without descriptions.

# 2. Intermediate

## LLM Settings



**Output (Temperature=0, Top-P = 1):**

1. Build a sandcastle
2. Collect seashells
3. Play beach volleyball
4. Fly a kite
5. Have a picnic
6. Try paddleboarding
7. Play frisbee
8. Go snorkeling
9. Take a long walk along the shore
10. Watch the sunset

## 2. Intermediate

### LLM Settings

💡 **Output (Temperature=1.5, Top-P = 1):**

1. Wade in the shallow waters blowing bubblegum bubbles
  2. Create intricate sandcastle sculptures
  3. Join an improvised beach volleyball match
  4. Take a scenic hike along the cliffs nearby
  5. People-watch and guess each person's beach-related story
- Collect seashells to create your own piece of beach art
- Learn to surf or try other water sports
- Kickstart a spontaneous sand fight
- Attempt shoreline fishing like the locals
- Engage in friendly competition by organizing a sandcastle building contest



# 2. Intermediate

## LLM Settings

Top P 0



### 2. Top-P

**Top-P (or Nucleus Sampling)** controls a model's randomness by setting a probability threshold, allowing it to choose from only the most likely tokens whose combined probabilities exceed that limit.

Example: When the model predicts the next word in ***The cat climbed up the \_\_\_\_***. The top five words it might be considering could be ***tree*** (probability 0.5), ***roof*** (probability 0.25), ***wall*** (probability 0.15), ***window*** (probability .07) and ***carpet*** (probability .03).

If we set Top-P to .90, the AI will only consider those tokens that cumulatively add up to at least ~90%. In our case:

- ❖ Adding ***tree*** -> total so far is 50%.
- ❖ Then adding ***roof*** -> total becomes 75%.
- ❖ Next comes ***wall***, and now our sum reaches 90%.

## 2. Intermediate

### LLM Settings

Maximum length 256

### 3. Maximum Length



Max Length sets the total number of tokens the AI can use, including both the user input and the AI's response. It helps control response length and prevents overly long or off-topic replies. If the limit is too short, the output may be cut off mid-sentence.

## 2. Intermediate

### LLM Settings

#### 5. Step Sequences

Stop sequences signal the model to stop generating text at specific points, helping control output length and structure. They're especially useful for structured formats like emails, lists, or dialogues. For example, stopping at *"Best regards"* to end an email cleanly.

Stop sequences  
Enter sequence and press Tab

Stop sequences  
Enter sequence and press Tab

Best regards ×

Sincerely ×

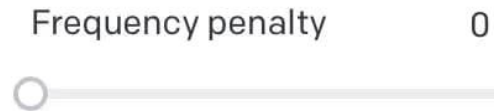
×

## 2. Intermediate

### LLM Settings

#### **6. Frequency Penalty**

A frequency penalty is a setting that discourages repetition in the generated text by penalizing tokens proportionally to how frequently they appear. The more often a token is used in the text, the less likely the AI is to use it again.

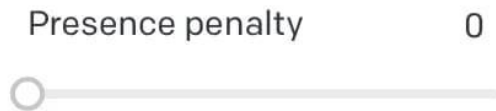


## 2. Intermediate

### LLM Settings

#### **7. Presence Penalty**

The presence penalty is similar to the frequency penalty, but flatly penalizes tokens based on if they have occurred or not, instead of proportionally.



# 2. Intermediate

## LLM Settings

### **Determinism Note:**

- ❖ Even when Temperature and Top-P are set completely to zero, the AI may not give the same exact output every time. This is due to randomness in GPU (graphics processing unit) calculations being done in the AI's "brain".
- ❖ The general recommendation is to alter temperature or Top P but not both.
- ❖ The general recommendation is to alter the frequency or presence penalty but not both.

## 2. Intermediate

### OpenAI ChatGPT

- ❖ It is a Large Language Model (LLM).
- ❖ A token, in the context of language models like ChatGPT, can be thought of as a unit of text.
- ❖ Tokenization is language-dependent, and token lengths may vary in other languages.

GPT-3 Codex

Here is a sentence broken into tokens. Notice how most words are their own token, but that sometime names or complete words like ChatGPT, or mispelled words, and , punctuation: , "are their" own tokens! As a rule of thumb for expressions in the English language there are approximately 750 words per 1000 tokens.

Clear Show example

Tokens	Characters
68	312

Here is a sentence broken into tokens. Notice how most words are their own token, but that sometime names or complete words like ChatGPT, or mispelled words, and , punctuation: , "are their" own tokens! As a rule of thumb for expressions in the English language there are approximately 750 words per 1000 tokens.

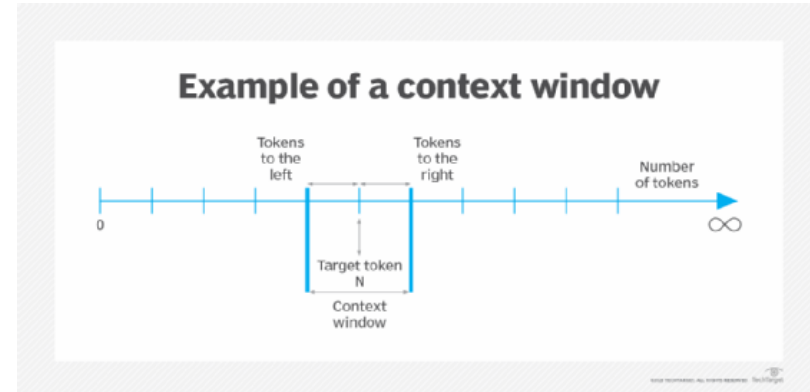
TEXT TOKEN IDS

## 2. Intermediate

### OpenAI ChatGPT

#### Context Windows

- ❖ ChatGPT has a token limit, both input and output are counted in token maximum. It is also called a context window
- ❖ GPT Website
- ❖ Tokenizer
- ❖ Your usage of tokens directly affects the cost of using ChatGPT.





# 2. Intermediate

## OpenAI ChatGPT

### Message and Role in GPT

- ❖ Create prompts by providing an array of messages that contain instructions for the model.
- ❖ Each message can have a different role, which influences how the model might interpret the input.

ROLE	DESCRIPTION	USAGE EXAMPLE
user	Instructions that request some output from the model. Similar to messages you'd type in ChatGPT as an end user.	<p>Pass your end-user's message to the model.</p> <div>Write a haiku about programming.</div>
developer	Instructions to the model that are prioritized ahead of user messages, following chain of command. Previously called the system prompt.	<p>Describe how the model should generally behave and respond.</p> <div><ol style="list-style-type: none"><li>1 You are a helpful assistant</li><li>2 that answers programming</li><li>3 questions in the style of a</li><li>4 southern belle from the</li><li>5 southeast United States.</li></ol></div> <p>Now, any response to a user message should have a southern belle personality and tone.</p>
assistant	A message generated by the model, perhaps in a previous generation request (see the "Conversations" section below).	<p>Provide examples to the model for how it should respond to the current request.</p> <p>For example, to get a model to respond correctly to knock-knock jokes, you might provide a full back-and-forth dialogue of a knock-knock joke.</p>

## 2. Intermediate

### OpenAI ChatGPT

#### Managing Tokens


- ❖ **Truncation:** Remove less relevant or older messages from the conversation while retaining the context necessary for the model to generate meaningful responses.
- ❖ **Shorter Text:** Keep your messages concise and to the point to reduce token consumption.
- ❖ **System Messages:** Use system-level instructions like “[SUMMARIZE]” to guide the model’s behavior in generating responses.

## Part 2: Prompt Library for Testing

1. Test Strategy & Planning
2. Test Case Writing
3. Test Automation Writing
4. Others

# 1. Test Strategy & Planning

## Test a Given Product

 **Prompt:** You are an expert who can advise on how to test different kind of software applications.  
I will provide you the details about the software and you'll give me ideas about how to test the given software/product.


**\*\*Software/Product Details:\*\***

*<A POS system that allows the business to accept payments from customers and keep track of sales.>*

Please let me know how can I test the given product.

# 1. Test Strategy & Planning

## Explore Aspects Specific Feature Area

 **Prompt:** Leveraging your expertise as a software tester, focusing on web and mobile applications, *your objective* is to offer expert guidance on essential considerations for comprehensively testing the given feature.

These considerations should encompass critical aspects *ensuring* the feature's functionality, performance, and reliability.

Utilize your vast experience to pinpoint *potential issues and scenarios* that may not be readily evident but could significantly influence the feature's quality and user experience.

Feature Detail:

<Feature details goes here>

# 1. Test Strategy & Planning

## Exploring Unknown Knowledge Areas

 **Prompt:** I have to test a software product that requires knowledge about the concepts mentioned below. Help me understand these concepts.


*\*\*Concepts:\*\**

*Knowledge of physiological signals (ECG, PPG, etc)*

*Medical device regulations (ISO 13485, IEC 62304, ISO 14971)*


# 1. Test Strategy & Planning

## Writing Comprehensive Test Plan

 **Prompt:** You are an *experienced software quality assurance manager* well-versed in software testing best practices and methodologies. Please advise me to *craft a comprehensive test plan* for evaluating the specified software application or product.  
Software/Product Description:  
<*Application details goes here*>

# 1. Test Strategy & Planning

## Regression Testing

 **Prompt:** You are an experienced software tester specializing in web and mobile apps. Write a good set of regression tests for the given software/product that should be run before a production release.


**\*\*Software/Product Details:\*\***

*<A POS system that allows the business to accept payments from customers and keep track of sales.>*



# 1. Test Strategy & Planning

## Tracking the Release Process

 **Prompt:** You are a knowledgeable software test manager. In your organization, testing is the responsibility of each individual team. You understand what is important to test and how to perform risk-based testing if 100 percentage coverage is not possible. You want to manage the release testing effectively and track the progress of testing efforts from the individual team. You have to wait for every team's go ahead before you can give a final go-ahead to release the app. How would you track this whole release management process?

## 2. Test Case Writing

### Writing Test Case in Tabular Format

 **Prompt:** You are an expert who can advise on how to effectively test a software application. Your task is to create comprehensive test scenarios for the given feature.

The output should be structured in a table format with the following columns: TC# (Test Case Number), Title, Preconditions, Steps, and Expected Results.


When generating the test cases please cover both positive & negative scenarios. Here's the feature detail:

Feature Detail:

*<Feature details goes here>*

## 2. Test Case Writing

### Writing Negative Scenarios


 **Prompt:** You are an experienced software quality assurance engineer capable of advising about thorough feature testing. Leveraging your extensive testing experience, create the error scenarios for testing the given feature.

Feature Detail:

*<Feature details goes here>*

## 2. Test Case Writing

### Writing Test Case with Specific Requirements

 **Prompt:** Leveraging your extensive experience in testing, your task is to devise comprehensive test ideas for testing the payment functionality of a Point of Sale (POS) application.


Please consider the following essential factors when writing the test cases:

- *Device types involved in the process: Card readers, cash drawers, and receipt printers.*
- *Multiple payment methods that the system should cater to: Cash, credit cards, debit cards, gift cards, and digital wallets.*
- *Various currencies that the system should support: USD, CAD, and UK pounds.*

Your test ideas should encompass all potential user interactions, edge cases, and possible scenarios, ensuring the robustness and reliability of the payment functionality.

# 2. Test Case Writing

## Writing Test Case with Specific Test Type

 **Prompt:** You are an experienced software tester specializing in web and mobile apps. Your task is to create comprehensive test scenarios. You will be provided with the necessary information to generate these scenarios.

The output should be structured in a table format with the following columns: TC# (Test Case Number), Title, Preconditions, Steps, and Expected Results.

Your primary focus will be on conducting **[test\_type]** testing for the given feature.

Feature Detail:


*<Feature details goes here>*

Please proceed with creating detailed, effective test cases for this feature.

**\* Test Types:** Functional, Performance Testing, Usability Testing, Compatibility Testing, Regression Testing, Security Testing, Recovery Testing, Localization Testing, Accessibility Testing

# 2. Test Case Writing

## Writing Test Case with Specific Test Type

 **Prompt:** You are an experienced software tester specializing in web and mobile apps. Your task is to create comprehensive test scenarios. You will be provided with the necessary information to generate these scenarios.

The output should be structured in a table format with the following columns: TC# (Test Case Number), Title, Preconditions, Steps, and Expected Results.

Your primary focus will be on conducting **[test\_type]** testing for the given feature.

Feature Detail:


*<Feature details goes here>*

Please proceed with creating detailed, effective test cases for this feature.

**\* Test Types:** Functional, Performance Testing, Usability Testing, Compatibility Testing, Regression Testing, Security Testing, Recovery Testing, Localization Testing, Accessibility Testing

## 2. Test Case Writing

### Writing Test Cases in Gherkin Format (BDD)


 **Prompt:** You are an expert who can advise on how to effectively test a software application using BDD practices. Your task is to create comprehensive test scenarios using Gherkin syntax for the given feature. Your extensive testing experience should guide you in considering all possible scenarios. Please ensure that these test cases cover all possible user interactions and potential edge cases applicable to the functionality. Here's the feature detail:

Feature Detail:

*<Feature details goes here>*

## 2. Test Case Writing


### Creating Step Definition Files

 **Prompt:** You're skilled in applying BDD practices and crafting feature and step definition files with precision. Your task is to write step definition files using Cucumber.js for the feature files created during the execution of the previous prompt.




# 2. Test Case Writing

## Writing Unit Tests

 **Prompt:** Please create a thorough set of unit tests using **<framework\_type>** framework for the provided program to ensure it performs its intended action accurately. Make sure to cover various edge cases and test for handling invalid inputs as well.

## 2. Test Case Writing


### Writing Automation Script

 **Prompt:** You are a Knowledgeable Automation Developer. Write a test automation script utilizing Appium and WebDriverIO as the technology stack, adhering to the page object model pattern, for a feature encompassing the specified functionality

***[functionality\_details]***

# 2. Test Case Writing

## Reviewing Automation Code


 **Prompt:** Please conduct a comprehensive code review for the following test automation code. Evaluate aspects such as code maintainability, readability, scalability, efficiency, reliability, and adherence to coding standards and best practices. Provide feedback on areas where improvements can be made to enhance the overall quality and effectiveness of the code.

```
**Automation tool stack:**  
[tool_stack_used]
```

```
**Automation Code:**  
[automation_code]
```

## 2. Test Case Writing


### Writing PR Description

 **Prompt:** Draft a PR description for the provided functionality/implementation. Include details about the changes made, the purpose of the implementation, any relevant background information, and how these changes contribute to the project's goals. Consider mentioning any potential impacts or dependencies associated with this implementation.

Functionality/Implementation:  
*[details\_goes\_here]*

## 2. Test Case Writing

### Test Automation - How to Questions


 **Prompt:** You are a Knowledgeable Automation Developer knowing Appium, WebDriverIO, BrowserStack, AppliTools Eye, and Mobile Apps Automation. I have a few automation how-to questions. Can you help me with the answers?

Q1: Is it possible to automate the following scenarios? If yes, please also share the recommended way.

*[scenario\_detail]*


## 2. Test Case Writing

### Help Resolve the Error Message

 **Prompt:** You are a Knowledgeable Automation Developer knowing Appium, WebDriverIO, BrowserStack, AppliTools Eye, and Mobile Apps Automation. I am getting an error when running the test. Help me resolve the error. The error is *[error\_detail]*

## 2. Test Case Writing

### Adding Code Comments

 **Prompt:** Provide clear and concise comments for the code snippet below, following best coding practices for readability and understanding.

`<code_goes_here>`

# 2. Test Case Writing

## Writing JMeter Tests

 **Prompt:** This is the code for my Restful API <CODE>. Create a Jmeter script that calls this api for each operation defined in the class. Also add a way to see the results of the test. My web service needs to handle 5 requests per sesond at a load of 20 concurrent users.

Here's additional information:

Server: localhost


Path: /calculate

Port: 80




# 3. Others

## Generate User Data & Corresponding Email Addresses

 **Prompt:** Develop a Python script to generate sample data for 100 users including user ID, full name along with corresponding email addresses, and subsequently save this data to a CSV file.


# 3. Others

## Generate Product Data & Corresponding Details

 **Prompt:** Generate sample data for 10 products, each including product ID, name, price, and category.

# 3. Others

## Write SQL Queries

 **Prompt:** Create a SQL INSERT statement to populate the Feedback table with data, including feedback ID, customer ID, and comments fields.

# Part 3: Prompt Challenges

1. Description

2. How-to

# 1. Challenge

## Description

Please independently create a *test plan*, *test suite*, *test cases*, and *corresponding automation test cases* for the company portal at <https://dev.ptnglobalcorp.com> OR any application that you comfortable with. You are **required to utilize AI tools** to assist in generating the test plan and test cases.

# 1. Challenge

## How-to

- ❖ Set up the test project from <https://github.com/ptnglobalcorp/aimact-ga>.
- ❖ Use prompt engineering to generate the test plan and test cases.
- ❖ Refine and tune the prompts to improve the quality of the generated outputs.
- ❖ Document all the prompts, test plans and test cases clearly and thoroughly.
- ❖ Commit and push all changes, including documentation, to your forked repository.

# THANK YOU!

For more information – Contact

Dieu Pham

Mobile: (+84) 919 456 360

Email: [dieu.pham@ptnglobalcorp.com](mailto:dieu.pham@ptnglobalcorp.com)

# References

Prompt Engineer: <https://www.promptingguide.ai/>