Prompt-Writing-Practice (/github/KhushiMittaliya/Prompt-Writing-Practice/tree/main)

_Prompt Writing ASSIGNMENT.ipynb (/github/KhushiMittaliya/Prompt-Writing-Practice/tree/main/_Prompt Writing ASSIGNMENT.ipynb)

3: Prompt Writing Practice

Objective:

Learn prompt structures and master prompt engineering techniques by writing and comparing prompts.

Submitted by:

Khushi Mittaliya

Tool Used:

ChatGPT + Jupyter Notebook

Task Overview

We will write and evaluate 5 types of prompts:

- 1. Summarization
- 2. Rewriting
- 3. Code Generation
- 4. Role-based Prompt
- 5. Structured Output Prompt

Each section includes:

- Initial Prompt
- Model Output Screenshot (Before)
- Refined Prompt
- Model Output Screenshot (After)
- Reflection



1. Summarization Prompt

Initial Prompt:

"Summarize the following text:

Albert Einstein was a German-born theoretical physicist who developed the theory of relativity, one of the two pillars of modern physics. He received the 1921 Nobel Prize in Physics for his services to theoretical physics."

Output Screenshot (Before):

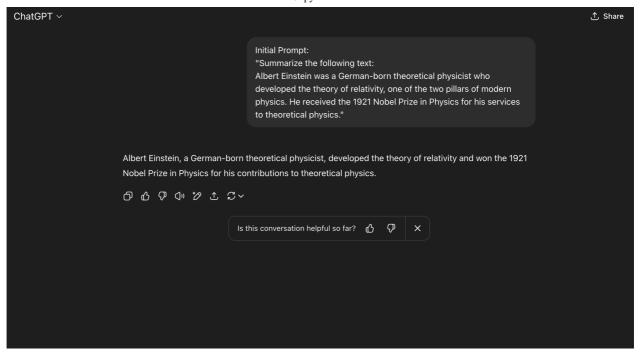
See Below

In [54]: from IPython.display import Image, display

print("Before Refinement:") display(Image("summarization before.png"))







Refined Prompt:

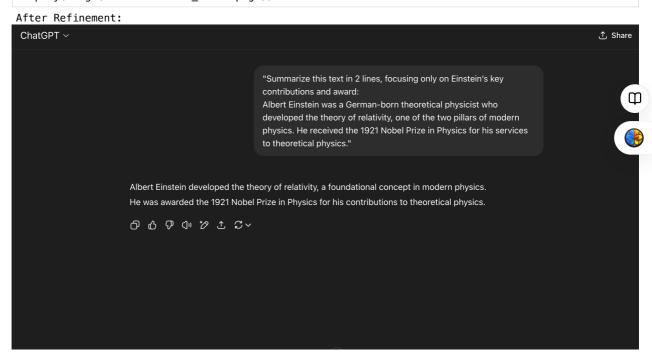
"Summarize this text in 2 lines, focusing only on Einstein's key contributions and award:

Albert Einstein was a German-born theoretical physicist who developed the theory of relativity, one of the two pillars of modern physics. He received the 1921 Nobel Prize in Physics for his services to theoretical physics."

Output Screenshot (After):

See Below

In [57]: print("After Refinement:")
 display(Image("summarization_after.png"))



Reflection:

Although the initial prompt resulted in a single-line summary, it was still quite close to the original and didn't emphasize brevity or clarity. The refined prompt produced a cleaner two-line output, separating Einstein's contribution and award. This made the summary more structured and easier to understand.



In this section, we will explore how to use prompt engineering to improve the rewriting of a sentence.

The goal is to transform an informal or unclear sentence into a more refined and professional version. We'll start with a basic rewriting prompt, evaluate the result, and then refine the prompt to improve tone, clarity, and formality.



2. Rewriting Prompt

Initial Prompt:

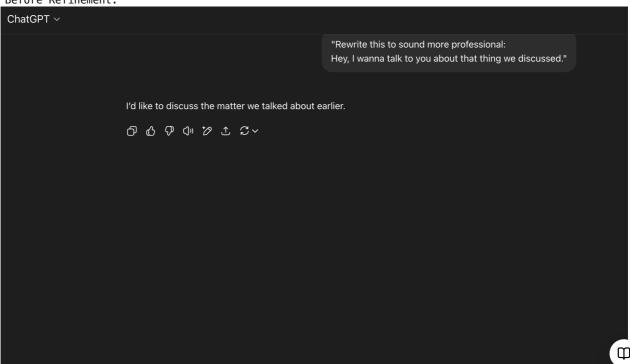
"Rewrite this to sound more professional:

Hey, I wanna talk to you about that thing we discussed."

Output Screenshot (Before):

In [62]: print("Before Refinement:")
 display(Image("rewriting_before.png"))

Before Refinement:



Refined Prompt:

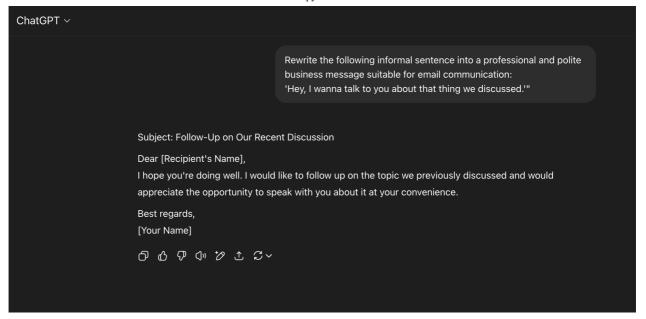
"Rewrite the following informal sentence into a professional and polite business message suitable for email communication:



'Hey, I wanna talk to you about that thing we discussed.'"

Output Screenshot (After):

After Refinement:



Reflection:

The initial prompt resulted in a basic rephrasing: "I'd like to discuss the matter we talked about earlier." While it sounded slightly more formal, it lacked clear structure, tone, and context.

After refining the prompt to specify the goal ("a professional and polite business message suitable for email"), the response became more complete. It included a subject line, greeting, and closing – making it look like a polished email.

This shows that giving clear tone, purpose, and context improves the model's understanding and output quality.

Code Generation Prompt

In this section, we'll test how well a language model can generate Python code with and without clarity in the prompt.

We start with a basic prompt to generate a prime number checker, then refine it to improve clarity, add comments, and specify the output format.

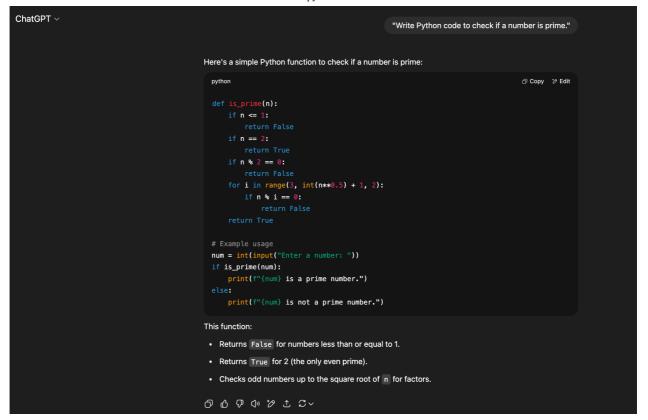
Initial Prompt:

"Write Python code to check if a number is prime."

Output Screenshot (Before):

In [70]: from IPython.display import Image, display

print("Before Refinement:")
display(Image("codegen_before.png"))

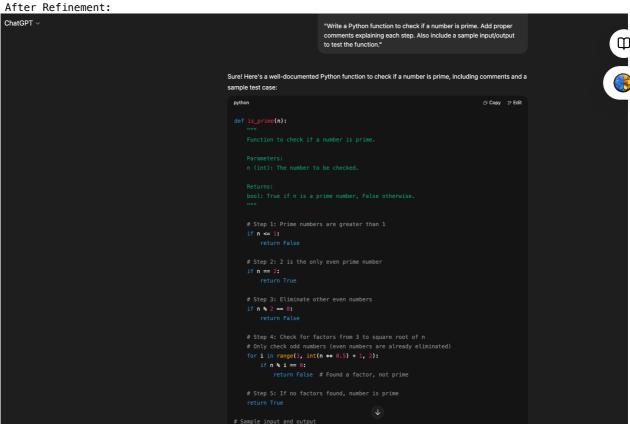


Refined Prompt:

"Write a Python function to check if a number is prime. Add proper comments explaining each step. Also include a sample input/output to test the function."

Output Screenshot (After):

```
In [73]: print("After Refinement:")
         display(Image("codegen_after1.png"))
         display(Image("codegen_after2.png"))
```



```
ChatGPT
                                                                                       # Only check odd numbers (even numbers are already eliminated)
for i in range(3, int(n ** 0.5) + 1, 2):
                                                                                  test_numbers = [2, 4, 5, 9, 13, 25, 29]
                                                                                  for num in test_numbers:
    result = is_prime(num)
                                                                                       print(f"{num} is {'a prime' if result else 'not a prime'} number.")

▼ Sample Output:

                                                                                       not a prime number.
                                                                                          ot a prime number.
                                                                                            t a prime number.
                                                                                Let me know if you'd like this inside a Jupyter notebook format too
                                                                                 ታ ያ ል ው ል ጥ ጄ <
```

Reflection:

The initial prompt gave correct code for checking prime numbers, but it was minimal and lacked clarity for beginners. There were no comments or explanations, and the output was basic.

The refined prompt produced a much more user-friendly and educational result. It included:

- · A well-structured function
- Inline comments for each step
- · A docstring explaining input/output
- · Sample test inputs and printed output

This highlights how adding instructions like "add comments" and "include test cases" can drastically improve both the readability and usability of code generated by an LLM.



Role-Based Prompt



Ф

In this section, we will evaluate how well a language model can imitate a professional persona when explicitly asked to do so.

We'll begin with a basic prompt asking for a diet plan, and then refine it to specify role, context, and goals — resulting in more accurate and relevant advice.

Initial Prompt:

"Give me a diet plan."

Output Screenshot (Before):

```
In [78]: from IPython.display import Image, display
         print("Before Refinement:")
         display(Image("role_before.png"))
```

ChatGPT ~ Give me a diet plan. Sure! Please help me tailor the diet plan to your needs by answering a few guick guestions: 1. What is your goal? Weight loss Muscle gain General fitness Medical condition management (e.g. diabetes, thyroid) 2. Your details: Age: . Height (cm or ft/in): • Weight (kg or lbs): • Activity level (sedentary, light, moderate, active): 3. Diet preference: Vegetarian / Non-vegetarian / Eggetarian / Vegan? 4. Any allergies or food restrictions? Once I have these, I'll give you a personalized Indian diet plan for the entire day (with timing). O O O O O O L C >

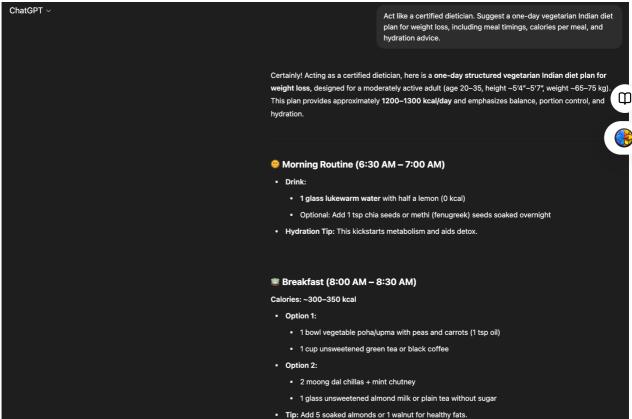
Refined Prompt:

"Act like a certified dietician. Suggest a one-day vegetarian Indian diet plan for weight loss, including meal timings, calories per meal, and hydration advice."

Output Screenshot (After):

In [81]: print("After Refinement:")
 display(Image("role_after1.png"))
 display(Image("role_after2.png"))
 display(Image("role_after3.png"))

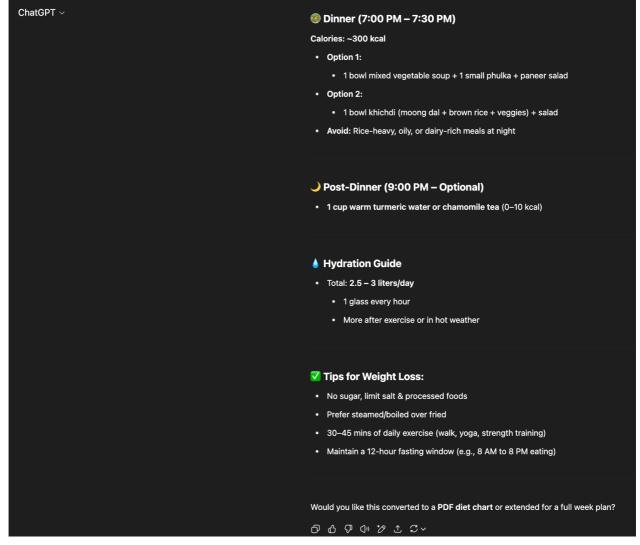
After Refinement:



Mid-Morning Snack (10:30 AM - 11:00 AM) ChatGPT ~ Calories: ~100-120 kcal • Option 1: • 1 seasonal fruit (apple/orange/guava/papaya) • Option 2: • 1 glass coconut water or buttermilk • Hydration Tip: Include ~1 glass water every hour. M Lunch (1:00 PM - 1:30 PM) Calories: ~350-400 kcal • 1 cup cooked dal (moong/masoor/toor) • 1 phulka (no ghee) or $\frac{1}{2}$ cup brown rice • 1 cup mixed vegetable sabzi (light oil) • Raw salad (carrot, cucumber, tomato) • 1 glass plain water before the meal (improves satiety) **➡** Evening Snack (4:00 PM - 4:30 PM) Calories: ~100 kcal • Option 1: • Roasted chana (1/2 cup) • Option 2: • Green tea + 1 khakhra (plain or methi) • Tip: Avoid biscuits, namkeens, or fried snacks.







Reflection:

The initial prompt was vague and produced a generic diet plan with no personalization or structure.

After refining the prompt with a clear role ("certified dietician") and adding specific instructions (vegetarian, Indian, weight loss, calories, hydration), the output became detailed, organized, and more realistic.



This shows how using role-based prompts and specifying the target outcome leads to more professional and useful responses.



Structured Output Prompt

In this section, we explore how prompt engineering helps guide the language model to present information in a structured format, such as a table.

We start with a basic list-style output, then refine the prompt to return a clear table with organized columns.

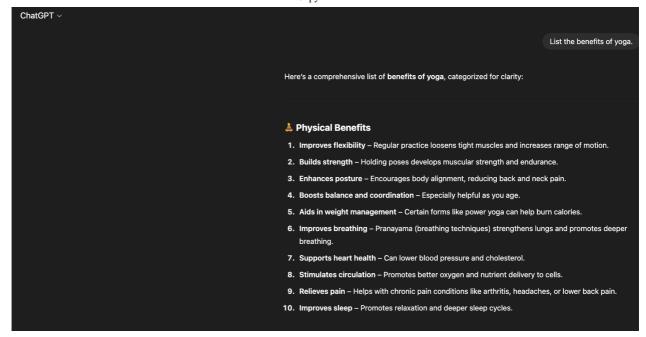
Initial Prompt:

"List the benefits of yoga."

Output Screenshot (Before):

In [89]: from IPython.display import Image, display

print("Before Refinement:")
display(Image("structured_before.png"))



Refined Prompt:

"List 5 key benefits of yoga in a table with three columns: Benefit, Physical Impact, and Mental Impact."

Output Screenshot (After):

In [91]: print("After Refinement:")
 display(Image("structured_after.png"))

After Refinement: ChatGPT ~ List 5 key benefits of yoga in a table with three columns: Benefit, Physical Impact, and Mental Impact. Here is a table listing 5 key benefits of yoga, with their corresponding physical and mental impacts: Benefit **Physical Impact** Mental Impact Improved Flexibility Increases joint mobility and reduces Enhances comfort and reduces physical muscle stiffness stress-related worry Stress Reduction Lowers blood pressure and relaxes the Reduces anxiety, calms the mind **Better Posture** Boosts self-confidence and body Aligns spine, reduces back and neck awareness **Enhanced Breathing** Strengthens respiratory muscles and Promotes calmness and emotio improves lung function Builds muscle tone and endurance Increased Strength Improves willpower and mental resilience Let me know if you'd like it in PDF or infographic format! 中 の 句 今 は は、

Reflection:

The original prompt returned a basic bulleted list, which was useful but unstructured.

After refining the prompt to request a specific output format (table with columns), the response became much easier to interpret, compare, and present visually.

This confirms that **explicit instructions about structure** (like "return a table") significantly enhance the quality and clarity of the output — especially for informative or analytical content.