

# Prompt Formatting Best Practices

## Prompt Formatting Best Practices

Effective prompt formatting is crucial for clarity and for guiding the LLM to produce the desired output. While the specific formatting might vary slightly depending on the model and the task, several general best practices apply across the board.

### 1. Start with Clear Instructions

Always place your primary instructions at the beginning of the prompt. This ensures that the model immediately understands its task before processing any additional context or input data. This practice helps the model prioritize the core objective.

**Example:**

*Summarize the following article in three bullet points.*

*Article: [Insert article text/link here]*

### 2. Use Delimiters for Separation

When providing multiple pieces of information, such as instructions, context, and input data, use clear delimiters to separate them. This helps the model distinguish between different sections of the prompt and prevents misinterpretation. Common delimiters include triple backticks (`), triple dashes (—), or hashtags (###).

**Example:**

*Instruction: Summarize the following text for a 10-year-old.*

*Text:*

*""” The rapid advancement of artificial intelligence has led to significant breakthroughs in various fields, from natural language processing to computer vision. These innovations promise to revolutionize industries and improve daily life.  
""”*

### 3. Be Specific and Detailed

The more specific and detailed your prompt, the better the results. Vague or ambiguous language can lead to generic or irrelevant outputs. Clearly express the desired context, outcome, length, format, and style. This includes specifying the tone (e.g., formal, informal, humorous), the target audience, and any constraints on the output.

**Example (Less Effective):**

*Write a poem about AI.*

**Example (More Effective):**

*Write a short, inspiring poem about the future of artificial intelligence, focusing on its potential to solve global challenges, in the style of a hopeful visionary. The poem should have four stanzas, with an AABB rhyme scheme.*

### 4. Provide Examples (Few-Shot Prompting)

One of the most effective ways to guide an LLM is by providing examples of the desired input-output pairs. This technique, known as few-shot prompting, that allows the model to learn the pattern and format you expect. The examples act as in-context learning demonstrations, helps the model understand the task without explicit fine-tuning.

**Example (Sentiment Classification):**

*Classify the sentiment of the following movie reviews as Positive, Negative, or Neutral.*

*Review: "This movie was fantastic!" -> Sentiment: Positive*

*Review: "I found the plot quite boring." -> Sentiment: Negative*

*Review: "The acting was okay, but the story was weak." -> Sentiment: Neutral*

*Review: "Absolutely loved the special effects!" -> Sentiment:*

When providing examples, ensure consistent formatting across all examples for better pattern recognition by the model.

## 5. Define the Desired Output Format Explicitly

Beyond just providing examples, explicitly stating the desired output format can significantly improve the model's adherence to your requirements. This is particularly useful when you need structured data, such as lists, JSON, or tables.

### Example (Entity Extraction):

*Extract the company names, people names, and specific topics from the following text. Present the output in a JSON format.*

**Text:** "During the recent tech conference, Satya Nadella of Microsoft announced new partnerships with OpenAI to develop advanced AI solutions. The discussion focused on ethical AI and large language models."

**Output Format:**

```
{  
    "company_names": [],  
    "people_names": [],  
    "specific_topics": []  
}
```

## 6. Avoid Negative Constraints (Say What to Do, Not What Not to Do)

It is generally more effective to tell the model what you want it to do rather than what you don't want it to do. Negative constraints can sometimes confuse the model or lead to unintended outputs. Frame your instructions positively.

### Example (Less Effective):

*Do not include any personal opinions in the summary.*

**Example (More Effective):**

*Provide an objective summary of the text.*

## 7. Use Leading Words for Code Generation

When prompting for code generation, providing leading words or phrases can guide the model towards a particular programming language or structure. For instance, starting with import for Python or SELECT for SQL can guide the model effectively.

### Example (Python Function):

```
# Write a Python function that calculates the factorial of a number.  
import
```

## 8. Iterative Refinement

Prompt engineering is an iterative process. It's rare to get the perfect output on the first try. Start with a simple prompt and gradually add more details, constraints, and examples as you refine your desired output. Experimentation is key to discovering what works best for your specific use case and model.