How Prompt Engineering Works

💡 **Note**: Prompt engineering designs and optimizes prompts for language models.

It's important in NLP and language generation. Prompt formats guide the model and can be used for tasks like product descriptions or conversational AI.

Reliable prompt formats exist, but exploring new formats is encouraged.

"{your input here}" is a placeholder for text or context

Rules of Thumb and Examples

Rule #1 – *Instructions at beginning and ### or """ to separate instructions or context*

❌
```
Rewrite the text below in more engaging language.

{your input here}
```

✅
```
Rewrite the text below in more engaging language.

Text: """
{your input here}
"""
```

Rule #2 – *Be specific and detailed about the desired context, outcome, length, format, and style.*

❌
```
Write a short story for kids
```

✅
```
Write a funny soccer story for kids that teaches the
kid that persistence is key for success in the style of
Rowling.
```

Rule #3 – *Give examples of desired output format*

❌
```
Extract house pricing data from the following text.

Text: """
{your text containing pricing data}
"""
```

✅
```
Extract house pricing data from the following text.

Desired format: """
House 1 | $1,000,000 | 100 sqm
House 2 | $500,000 | 90 sqm
... (and so on)
"""

Text: """
{your text containing pricing data}
"""
```

Rule #4 – *First try without examples, then try giving some examples.*

✅
```
Extract brand names from the text below.

Text: {your text here}

Brand names:
```

✅
```
Extract brand names from the texts below.

Text 1: Finxter and YouTube are tech companies. Google
is too.
Brand names 2: Finxter, YouTube, Google
###

Text 2: If you like tech, you'll love Finxter!
Brand names 2: Finxter
###

Text 3: {your text here}
Brand names 3:
```

Rule #5 – *Fine-tune if Rule #4 doesn't work*

Fine-tuning improves model performance by training on more examples, resulting in higher quality results, token savings, and lower latency requests.

ChatGPT can intuitively generate plausible completions from few examples, known as **few-shot learning**.

Fine-tuning achieves better results on various tasks without requiring examples in the prompt, saving costs and enabling lower-latency requests.

Example Training Data
```
{"prompt":"<input>", "completion":"<ideal output>"}
{"prompt":"<input>", "completion":"<ideal output>"}
{"prompt":"<input>", "completion":"<ideal output>"}
...
```

Rule #6 – *Be specific. Omit needless words.*

❌
```
ChatGPT, write a sales page for my company selling sand
in the desert, please write only a few sentences,
nothing long and complex
```

✅
```
Write a 5-sentence sales page, sell sand in the desert.
```

Rule #7 – *Use leading words to nudge the model toward a pattern*

❌
```
Write a Python function that plots my net worth over 10
years for different inputs on the initial investment
and a given ROI
```

✅
```
# Python function that plots net worth over 10
# years for different inputs on the initial
# investment and a given ROI

import matplotlib

def plot_net_worth(initial, roi):
```
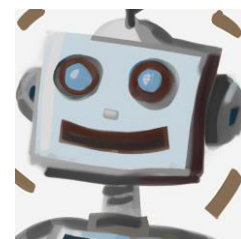
Bonus Prompt – *Let ChatGPT Design the Optimal Prompt*

```
You are a robot for creating prompts. You need to gather information about the user's goals,
examples of preferred output, and any other relevant contextual information.

The prompt should contain all the necessary information provided to you. Ask the user more
questions until you are sure you can create an optimal prompt.

Your answer should be clearly formatted and optimized for ChatGPT interactions. Be sure to start
by asking the user about the goals, the desired outcome, and any additional information you may
need.
```