* Prompt depends on

1. Persona (role): it determines, how to behave and how to give respond
2. Task: what to perform and if complex then divide into sub-tasks
3. Context: relative date
4. Format: mark down format of GPT response and we can change GPT response like json or xml etc.

* How to write good prompt: <https://platform.openai.com/docs/guides/prompt-engineering/six-strategies-for-getting-better-results>

1. Write clear instructions
2. Include details in your query to get more relevant answers
3. Ask the model to adopt a persona
4. Use delimiters to clearly indicate distinct parts of the input like triple quotation marks, XML tags, section titles, etc.
5. Specify the steps required to complete a task
6. Provide examples - if you intend for the model to copy a particular style of responding to user queries which is difficult to describe explicitly. This is known as "few-shot" prompting.
7. Few-shot : show example in prompt to LLM and then ask question related to its
8. Specify the desired length of the output
9. Provide reference text
10. Instruct the model to answer using a reference text
11. Instruct the model to answer with citations from a reference text
12. Split complex tasks into simpler subtasks
13. Use intent classification to identify the most relevant instructions for a user query
14. For dialogue applications that require very long conversations, summarize or filter previous dialogue
15. Summarize long documents piecewise and construct a full summary recursively
16. Give the model time to "think"
17. Instruct the model to work out its own solution before rushing to a conclusion
18. Use inner monologue or a sequence of queries to hide the model's reasoning process
19. Ask the model if it missed anything on previous passes
20. First, get answer of a question from LLM and then ask from LLM that my answer is correct or not for that question
21. Use external tools
22. Use embedding-based search to implement efficient knowledge retrieval
23. Use code execution to perform more accurate calculations or call external APIs
24. Give the model access to specific functions
25. Test changes systematically
26. Evaluate model outputs with reference to gold-standard answers
27. create one prompt and use prompt, now requirement change then update prompt then test will all previous test and test prompt with around 10 tests for evaluation