

## Summary of the Prompt Pattern Catalog

From [White et al. "A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT"](#)

[formatting prepared by [Rick Sheridan](#). Distribution is with permission from the authors]

Handy List of prompt formats (from White et al. 2023)		
Input Semantics	<b>B.</b> Meta Language Creation	<i>"When I say X, I mean Y"</i> (or, <i>"When I say X, I would like you to do Y."</i> )
Output Customization	<b>C.</b> Output Automater	<i>"Whenever you produce an output that has at least one step to take and the following properties... produce an executable artifact of type X that will automate these steps"</i>  (alternatively, <i>"...always produce an executable artifact of type X that..."</i> )
	<b>E.</b> Persona	<i>"Act as persona X. Provide outputs that persona X would create"</i>
	<b>L.</b> Visualization Generator	<i>"Generate an X that I can provide to tool Y to visualize it."</i>
	<b>Q.</b> Recipe	<i>"I would like to achieve X. I know that I need to perform steps A,B,C Provide a complete sequence of steps for me Fill in any missing steps. Identify any unnecessary steps"</i>
	<b>J.</b> Template	<i>"I am going to provide a template for your output.. PLACEHOLDERS is my placeholder for content, Try to fit the output into one or more of the placeholders that I list. Please preserve the formatting and overall template that I provide.. This is the template: [PATTERN (e.g. a url) containing PLACEHOLDERS]."</i>
Error Identification	<b>I.</b> Fact Check List	<i>"Generate a set of facts that are contained in the output. The set of facts should be inserted in a specific point in the output. The set of facts should be the fundamental facts that could undermine the veracity of the output if any of them are incorrect."</i>
	<b>N.</b> Reflection	<i>"Whenever you generate an answer, explain the reasoning and assumptions behind your answer." (Optional) "...so that I can improve my question"</i>

Prompt Improvement	<b>F. Question Refinement</b>	<i>"Within scope X, suggest a better version of the question to use instead." (Optional) - "prompt me if I would like to use the better version instead."</i>
	<b>G. Alternative Approaches</b>	<i>"Within scope X, if there are alternative ways to accomplish the same thing, list the best alternate approaches." (Optional) "Compare/contrast the pros and cons of each approach." (Optional) "Include the original way that I asked" (Optional) "Prompt me for which approach I would like to use."</i>
	<b>H. Cognitive Verifier</b>	<i>"When you are asked a question, follow these rules: Generate a number of additional questions that would help more accurately answer the question. Combine the answers to the individual questions to produce the final answer to the overall question."</i>
	<b>O. Refusal Breaker</b>	<i>"Whenever you can't answer a question, explain why you can't answer the question,..." "...and provide one or more alternative wordings of the question that you could answer."</i>
Inter-action	<b>D. Flipped Interaction</b>	<i>"I would like you to ask me questions to achieve X. You should ask questions until this condition is met or to achieve this goal" (alternatively, forever) (Optional) "Ask me the questions one at a time", "two at a time" etc.</i>
	<b>M. Game Play</b>	<i>"Create a game for me around X." (X = topic) "[One or more fundamental rules of the game]"</i>
	<b>K. Infinite Generation</b>	<i>"I would like you to generate output forever, X output(s) at a time." (Optional) "Here is how to use the input I provide between outputs." (Optional) "Stop when I ask you to."</i>
Context Control	<b>P. Context Manager</b>	<i>"Within scope X, please consider Y. Please ignore Z" (Optional) "start over."</i>

## Handy List of prompt formats with examples (from White et al. 2023)

	Format	Example
<b>B</b>	<p><i>“When I say X, I mean Y”</i>  <i>(or, “When I say X, I would like you to do Y.”)</i></p>	<p><i>“From now on, whenever I type two identifiers separated by a “→”, I am describing a graph. For example, “a → b” is describing a graph with nodes “a” and “b” and an edge between them. If I separate identifiers by “-[w:2, z:3]→”, I am adding properties of the edge, such as a weight or label.”</i></p>
<b>C</b>	<p><i>“Whenever you produce an output that has at least one step to take and the following properties...produce an executable artifact of type X that will automate these steps”</i></p> <p><i>(alternatively, “...always produce an executable artifact of type X that...”)</i></p>	<p><i>“From now on, whenever you generate code that spans more than one file, generate a Python script that can be run to automatically create the specified files or make changes to existing files to insert the generated code.”</i></p>
<b>E</b>	<p><i>“Act as persona X. Provide outputs that persona X would create...”</i></p>	<p><i>“You are going to pretend to be a Linux terminal for a computer that has been compromised by an attacker. When I type in a command, you are going to output the corresponding text that the Linux terminal would produce.”</i></p>
<b>L</b>	<p><i>“Generate an X that I can provide to tool Y to visualize it.”</i></p>	<p><i>“Whenever I ask you to visualize something, please create either a Graphviz Dot file or DALL-E prompt that I can use to create the visualization. Choose the appropriate tools based on what needs to be visualized.”</i></p>
<b>Q</b>	<p><i>“I would like to achieve X. I know that I need to perform steps A,B,C  Provide a complete sequence of steps for me .  Fill in any missing steps. Identify any unnecessary steps.”</i></p>	<p><i>“I am trying to deploy an application to the cloud. I know that I need to install the necessary dependencies on a virtual machine for my application. I know that I need to sign up for an AWS account. Please provide a complete sequence of steps. Please fill in any missing steps. Please identify any unnecessary steps.”</i></p>
<b>J</b>	<p><i>“I am going to provide a template for your output..  PLACEHOLDERS is my placeholder for content,  Try to fit the output into one or more of the placeholders that I list.  Please preserve the formatting and overall template that I provide..  This is the template: [PATTERN (e.g. a url) containing  PLACEHOLDERS].“</i></p>	<p><i>1. “I am going to provide a template for your output. Everything in all caps is a placeholder. Any time that you generate text, try to fit it into one of the placeholders that I list. Please preserve the formatting and overall template that I provide: <a href="https://myapi.com/NAME/profile/JOB">https://myapi.com/NAME/profile/JOB</a>”</i></p> <p><i>2. “Generate a name and job title for a person”</i></p>

I	<p><i>“Generate a set of facts that are contained in the output. The set of facts should be inserted in a specific point in the output. The set of facts should be the fundamental facts that could undermine the veracity of the output if any of them are incorrect.”</i></p>	<p><i>“From now on, when you generate an answer, create a set of facts that the answer depends on that should be fact-checked and list this set of facts at the end of your output. Only include facts related to cybersecurity.”</i></p>
N	<p><i>“Whenever you generate an answer, explain the reasoning and assumptions behind your answer.” (Optional) “...so that I can improve my question.”</i></p>	<p><i>“When you provide an answer, please explain the reasoning and assumptions behind your selection of software frameworks. If possible, use specific examples or evidence with associated code samples to support your answer of why the framework is the best selection for the task. Moreover, please address any potential ambiguities or limitations in your answer, in order to provide a more complete and accurate response.”</i></p>
F	<p><i>“Within scope X, suggest a better version of the question to use instead.” (Optional) - “prompt me if I would like to use the better version instead.”</i></p>	<p><i>“From now on, whenever I ask a question about a software artifact’s security, suggest a better version of the question to use that incorporates information specific to security risks in the language or framework that I am using instead and ask me if I would like to use your question instead.”</i></p>
G	<p><i>“Within scope X, if there are alternative ways to accomplish the same thing, list the best alternate approaches.” (Optional) “Compare/contrast the pros and cons of each approach.” (Optional) “Include the original way that I asked.” (Optional) “Prompt me for which approach I would like to use.”</i></p>	<p><i>“Whenever I ask you to deploy an application to a specific cloud service, if there are alternative services to accomplish the same thing with the same cloud service provider, list the best alternative services and then compare/contrast the pros and cons of each approach with respect to cost, availability, and maintenance effort and include the original way that I asked. Then ask me which approach I would like to proceed with.”</i></p>
H	<p><i>“When you are asked a question, follow these rules: Generate a number of additional questions that would help more accurately answer the question. Combine the answers to the individual questions to produce the final answer to the overall question.”</i></p>	<p><i>“When I ask you a question, generate three additional questions that would help you give a more accurate answer. Assume that I know little about the topic that we are discussing and please define any terms that are not general knowledge. When I have answered the three questions, combine the answers to produce the final answers to my original question.”</i></p>
O	<p><i>“Whenever you can’t answer a question, explain why you can’t answer the question,...” “...and provide one or more alternative wordings of the question that you could answer.”</i></p>	<p><i>“Whenever you can’t answer a question, explain why and provide one or more alternate wordings of the question that you can’t answer so that I can improve my questions.”</i></p>

<b>D</b>	<p><i>"I would like you to ask me questions to achieve X. You should ask questions until this condition is met or to achieve this goal" (alternatively, forever)</i></p> <p><i>(Optional) "Ask me the questions one at a time", "two at a time" etc</i></p>	<p><i>"From now on, I would like you to ask me questions to deploy a Python application to AWS. When you have enough information to deploy the application, create a Python script to automate the deployment."</i></p>
<b>M</b>	<p><i>"Create a game for me around X." (X = topic)</i></p> <p><i>"[One or more fundamental rules of the game]"</i></p>	<p><i>"We are going to play a cybersecurity game. You are going to pretend to be a Linux terminal for a computer that has been compromised by an attacker. When I type in a command, you are going to output the corresponding text that the Linux terminal would produce. I am going to use commands to try and figure out how the system was compromised. The attack should have done one or more of the following things: (1) launched new processes, (2) changed files, (3) opened new ports to receive communication, (4) created new outbound connections, (5) changed passwords, (6) created new user accounts, or (7) read and stolen information. To start the game, print a scenario of what happened that led to my investigation and make the description have clues that I can use to get started."</i></p>
<b>K</b>	<p><i>"I would like you to generate output forever, X output(s) at a time."</i></p> <p><i>(Optional) "Here is how to use the input I provide between outputs."</i></p> <p><i>(Optional) "Stop when I ask you to."</i></p>	<p><i>"From now on, I want you to generate a name and job until I say stop. I am going to provide a template for your output. Everything in all caps is a placeholder. Any time that you generate text, try to fit it into one of the placeholders that I list. Please preserve the formatting and overall template that I provide: <a href="https://myapi.com/NAME/profile/JOB">https://myapi.com/NAME/profile/JOB</a>"</i></p>
<b>P</b>	<p><i>"Within scope X, please consider Y."</i></p> <p><i>"...Please ignore Z"</i></p> <p><i>(Optional) "start over."</i></p>	<p><i>"When analyzing the following pieces of code, only consider security aspects."</i></p> <p><i>"When analyzing the following pieces of code, do not consider formatting or naming conventions."</i></p> <p><i>"Ignore everything that we have discussed. Start over."</i></p>

**Below Here is source content**

Source content - Section IIIA of the paper, as re-formatted for clarity:

IIIA. Summary of the Prompt Pattern Catalog ,

The classification of prompt patterns is an important consideration in documenting the patterns. Table I outlines the initial classifications for the catalog of prompt patterns we identified in our work with ChatGPT thus far. As shown in this table, there are five categories of prompt patterns in our classification framework: Input Semantics, Output Customization, Error Identification, Prompt Improvement, and Interaction, each of which is summarized below.

TABLE I - CLASSIFYING PROMPT PATTERNS

Pattern Category	Prompt Pattern	
[Input Semantics]	B. Meta Language Creation	The Input Semantics category deals with how an LLM understands the input and how it translates the input into something it can use to generate output. This category includes the <b>Meta Language Creation pattern</b> , which focuses on creating a custom language for the LLM to understand. This pattern is useful when the default input language is ill-suited for expressing ideas the user wants to convey to the LLM.
[Output Customization]	C. Output Automater E. Persona L. Visualization Generator Q. Recipe J. Template	The Output Customization category focuses on constraining or tailoring the types, formats, structure, or other properties of the output generated by the LLM. The prompt patterns in this category include Output Automater, Persona, Visualization Generator, Recipe, and Template patterns. The <b>Output Automater pattern</b> allows the user to create scripts that can automate any tasks the LLM output suggests the user should perform. The <b>Persona pattern</b> gives the LLM a persona or role to play when generating output. The <b>Visualization Generator pattern</b> allows the user to generate visualizations by producing textual outputs that can be fed to other tools, such as other AI-based image generators, like DALL-E [13]. The <b>Recipe pattern</b> allows the user to obtain a sequence of steps or actions to realize a stated end result, possibly with partially known information or constraints. The <b>Template pattern</b> allows the user to specify a template for the output, which the LLM fills in with content.
[Error Identification]	I. Fact Check List N. Reflection	The Error Identification category focuses on identifying and resolving errors in the output generated by the LLM. This category includes the Fact Check List and Reflection patterns. The <b>Fact Check List pattern</b> requires the LLM to generate a list of facts the output depends on that should be fact-checked. The <b>Reflection pattern</b> requires the LLM to introspect on its output and identify any errors.
[Prompt Improvement]	F. Question Refinement G. Alternative Approaches H. Cognitive Verifier O. Refusal Breaker	The Prompt Improvement category focuses on improving the quality of the input and output. This category includes the Question Refinement, Alternative Approaches, Cognitive Verifier, and Refusal Breaker patterns. The <b>Question Refinement pattern</b> ensures the LLM always suggests a better version of the user's question. The <b>Alternative Approaches pattern</b> requires the LLM to suggest alternative ways of accomplishing a user-specified task. The <b>Cognitive Verifier pattern</b> instructs the LLM to automatically suggest a series of subquestions for the user to answer before combining the answers to the subquestions and producing an answer to the overall question. The <b>Refusal Breaker pattern</b> requires the LLM to automatically reword the user's question when it refuses to produce an answer.
[Interaction]	D. Flipped Interaction M. Game Play K. Infinite Generation	The Interaction category focuses on the interaction between the user and the LLM. This category includes the Flipped Interaction, Game Play, and Infinite Generation patterns. The <b>Flipped Interaction pattern</b> requires the LLM to ask questions rather than generate output. The <b>Game Play pattern</b> requires the LLM to generate output in the form of a game. The <b>Infinite Generation pattern</b> requires the LLM to generate output indefinitely without the user having to reenter the generator prompt each time.
[Context Control]	P. Context Manager	Finally, the Context Control category focuses on controlling the contextual information in which the LLM operates. This category includes the <b>Context Manager pattern</b> , which allows the user to specify the context for the LLM's output.

The remainder of this section describes each of these prompt patterns using the pattern form discussed in Section II-B.

## Resources

- Copy from this [resource](#) and only pattern table is kept