Prompt Name	How to Use it	Pattern	Prompt Example 1	Prompt Example 2	Best For	Use case 1	Use case 2	Use case 3
Zero-shot Prompting	Describe the task you want to complete Supply the single input to operate on	Perform task X on input Y	Summarize the following news article in one sentence: Input: The city council of Riverton voted 6-1 on Tuesday to …	Translate to Spanish: Input: Where is the nearest pharmacy?	Simple tasks, direct questions, common instructions where model has strong prior knowledge.	Classifying movie review sentiment directly.	Answering a simple factual question ("What is the capital of France?").	Summarizing a short, straightforward text.
Few-shot Prompting	Describe the task. Provide one or more EXAMPLE input → output pairs. End with the Actual Input followed by an arrow, signalling the model to continue.	Task X EXAMPLE: $A_1 \rightarrow B_1$ EXAMPLE: $A_2 \rightarrow B_2$ Y →	Convert movie titles to emoji. EXAMPLE: “Jaws” -> EXAMPLE: “Titanic” -> “The Matrix” -> ?	Categorize emails as “Work”, “Personal”, or “Spam” EXAMPLE: “50% off shoes this weekend only!” -> Spam EXAMPLE: “Can you send the Q2 budget file?” -> Work “Grandma’s apple pie recipe” ->	Guiding output format/structure, tasks needing specific patterns, improving accuracy over zero-shot, adapting to novel tasks.	Parsing pizza orders into a specific JSON format (PDF example).	Translating sentences using a very specific, less common style demonstrated in examples.	Performing a novel classification task based on provided examples.
System Prompting	Provide an overall instruction or constraint that governs every reply (often set in the system channel). Follow with the user’s actual prompt.	System instruction: Always apply rule X. User prompt: Y	System instruction: “You are a meticulous fact-checker who always cites sources.” User prompt: “List three surprising facts about honeybees.”	System instruction: “Always answer in Shakespearean English.” User prompt: “Explain photosynthesis.”	Setting overall model behavior/constraints, defining mandatory output formats (like JSON), enforcing safety/tone guidelines across interactions.	Specifying output must be returned in uppercase (PDF example).	Requiring all output to be valid JSON objects following a schema (PDF example).	Instructing the model to always answer respectfully.
Role Prompting	Act as a specific role or profession. Perform a task that naturally belongs to that role	Act as X Perform task Y	Act as a medieval blacksmith. Describe how you would forge a longsword	Act as a NASA flight director. Walk me through the launch go/no-go poll	Controlling output tone, style, persona; leveraging role-specific knowledge patterns; framing the interaction context.	Acting as a travel guide to suggest locations (PDF example).	Acting as an expert Python programmer to explain complex code.	Acting as a skeptical historian to analyze a document.
Contextual Prompting	Supply background information or context. Ask the model to perform a task that depends on that context.	Context: X Task: Y	Context: You are reviewing a grant proposal that seeks $50 000 to build a community garden. Task: Write a 200-word critique highlighting strengths and weaknesses.	Context: The user’s dietary restrictions: vegan, allergic to almonds. Task: Propose a three-course dinner menu.	Providing specific background for a task, tailoring responses to current situation/conversation, clarifying nuances based on provided info.	Suggesting blog post topics based on the blog's specific theme (PDF example).	Answering questions based on a provided document snippet within the prompt.	Summarizing a meeting based on previously supplied meeting notes.
Step-back Prompting	Ask for a high-level or abstract answer to a general version of the problem. Immediately use that general answer as context for solving the specific instance	Broadly, what is X? Using X, solve Y	1. In general, what factors determine whether a coastal city floods during a hurricane? 2. Using those factors, assess the flood risk for Wilmington, NC, given a Category-3 storm.	1. Broadly, what makes a job offer attractive to software engineers? 2. Apply those criteria to critique this offer from ByteForge Inc.	Improving reasoning on complex tasks, activating broader knowledge, reducing bias by focusing on principles first before specifics.	Generating a game storyline by first asking for key elements of the genre (PDF example).	Solving a physics problem by first asking for the underlying principles involved.	Evaluating a complex policy decision by first asking about general relevant criteria.
Chain of Thought (CoT)	Present the problem or question. Add an explicit nudge to reason step-by-step or supply few-shot demonstrations that include full reasoning traces.	Problem X Let’s think step by step to reach Y	If a train leaves Chicago at 60 mph and another leaves St Louis at 45 mph heading toward Chicago on the same track 300 miles apart, when will they meet? Let’s think step by step	Few-shot math word-problem prompt that shows worked solutions, then ends with a new problem and “Let’s think step by step	Arithmetic, commonsense reasoning, symbolic reasoning tasks where intermediate steps are crucial for accuracy. Improving interpretability.	Solving math word problems requiring intermediate calculations (PDF example).	Explaining the logical steps required to reach a specific conclusion.	Planning a sequence of actions to achieve a goal.
Self-consistency	Issue a Chain-of-Thought prompt. Internally (or via scripting) run it several times with higher randomness. Aggregate the diverse answers, choosing the one that appears most often or is best justified. (Implementation detail—aggregation—usually handled in code rather than in a single written prompt.)	Estimate X. Show your reasoning step by step to derive Y. (Run this prompt multiple times at higher temperature and aggregate the answers.)	Solve the puzzle below. Show your reasoning (Run 10 times at temperature = 1.2, then majority-vote the numeric answer.)	Estimate the monthly LinkedIn Ads budget required to generate 500 qualified leads for our B2B SaaS product Show your reasoning step by step (Run this prompt 8 times at temperature = 1.0 and report the median budget estimate.)	Improving accuracy and robustness of CoT results, especially for tasks with a single correct answer but multiple possible reasoning paths.	Getting a more reliable classification for ambiguous inputs (PDF email example).	Verifying the result of a multi-step mathematical calculation.	Increasing confidence in the answer to a complex reasoning question.
Tree of Thoughts (ToT)	Instruct the model to explore multiple reasoning branches, evaluating each intermediate “thought” before deciding to expand or prune. Often implemented with an external controller loop	Generate several reasoning branches to accomplish X. For each branch, evaluate Y; expand the best branch into a detailed plan	Generate three distinct high-level strategies for reducing urban traffic congestion. For each strategy, list pros and cons. After evaluating, choose the most promising strategy and elaborate a detailed 10-step action plan	Generate three distinct growth strategies for a bootstrapped e-commerce brand entering the EU market For each strategy, list key steps, required resources, risks, and projected ROI Evaluate all strategies and choose the one with the best ROI-to-risk ratio… then provide a detailed 90-day execution roadmap	Complex problem-solving requiring exploration and lookahead, planning, tasks where a single CoT path might be suboptimal.	Creative writing tasks exploring different plot developments.	Solving complex logical puzzles or planning problems (e.g., Game of 24).	Generating diverse potential solutions to an open-ended design problem.

Prompt Name	How to Use it	Pattern	Prompt Example 1	Prompt Example 2	Best For	Use case 1	Use case 2	Use case 3
ReAct (Reason & Act)	Alternate between Thought: (reflection) and Action: (call to a tool, search, calculation, etc.). Continue looping until the task is complete.	Thought: I need X to achieve Y Action: … Observation: … Thought: … Final Answer: …	Thought: I need the current weather in Paris to recommend attire. Action: weather_api("Paris") Observation: 18 °C, light rain. Thought: Light rain jacket is advisable. Final Answer: Pack a waterproof jacket and an umbrella	Thought: I need the current AWS price for t3. medium instances in us-east-1 to estimate hosting costs Action: aws_pricing_api("t3.medium","us-east-1") Observation: $0.0416 per hour Thought: Now calculate monthly cost at 70 % utilization across 4 instances Action: calculator("0.0416*24*30*0.7*4") Observation: 83.9 Thought: Add a 20 % buffer for bandwidth and storage. Final Answer: Budget about $100 per month for compute; with bandwidth and storage, plan for roughly $120	Tasks requiring external information retrieval, interaction with APIs/tools, grounding responses in real-time or external data. Agent-like behavior.	Answering questions needing current information via web search (PDF Metallica example).	Using a calculator tool for precise mathematical operations within a larger task.	Interacting with a calendar API to schedule an event based on natural language request.
Automatic Prompt Engineering (APE)	Ask the model to invent several candidate prompts for a task. Evaluate each candidate on sample inputs. Select (or ensemble) the highest-performing prompt.	Design N candidate prompts that perform task X on data Y; return them ranked by expected performance	Meta-Prompt: You are designing prompts that convert product reviews into concise pros/cons lists. Generate five diverse prompts that could accomplish this […model returns Prompt A … Prompt E…] Evaluate Prompt A-E on held-out reviews, pick the best F1, deploy.	Meta-Prompt: You are designing prompts that convert raw customer-support chat logs into a JSON object with 'issue_type', 'priority', and 'next_action' Generate six diverse candidate prompts suitable for busy SMB support teams	Automating prompt discovery/optimization, generating diverse phrasing for training data augmentation, finding effective instructions for complex tasks.	Generating various ways a user might phrase an e-commerce order (PDF example).	Creating diverse prompts for fine-tuning a sentiment analysis model.	Optimizing the instructional prompt for a complex data extraction task.

Code Prompting	State the code-related instruction (write, fix, explain, translate, etc.).
	
	Provide any relevant snippet or specification.
	
	Optionally include constraints (language, style, performance, libraries)
		Write X code that accomplishes Y	import requests
			
			def fetch(urls):
			    return [requests.get(u).text for u in urls]
			```
			”*

			*“Translate this Bash one-liner into Windows PowerShell.”*

			---

			Feel free to copy these “pattern sheets” into your own playbook or tweak the example prompts to suit your domain
			```
				Write a Python function that pulls the last 30 days of Stripe payments using the Stripe API
				
				… aggregates revenue by day
				
				… and returns a pandas DataFrame ready for plotting
					Code generation, code explanation, translation between programming languages, debugging errors, code review.	Generating a bash script based on requirements (PDF example).	Explaining what a specific Python function does (PDF example).	Debugging a Python script and suggesting fixes (PDF example).