

# AI Terminology

A cheat sheet. *The jargon you'll run into — in plain English, in one page.*

<p><b>LLM</b></p> <p>A text-prediction engine trained on a massive corpus. Given text, it predicts what comes next. "The model" usually means this. Examples: Claude, GPT, Gemini.</p>	<p><b>Token</b></p> <p>A chunk of text the model processes — often a word, sometimes less. Pricing, context size, and speed are all counted in tokens.</p>	<p><b>Context window</b></p> <p>The total tokens the model can consider in one go. When it fills up, older messages drop out silently. Start a fresh chat when it matters.</p>
<p><b>Prompt</b></p> <p>Whatever you type into the chat. Question, task, bug report. The quality of what you get back is directly tied to how clearly you write it.</p>	<p><b>System prompt</b></p> <p>Hidden instructions that sit above your chat — shaped by .cursor/rules and AGENTS.md. You don't write it directly; you configure it.</p>	<p><b>Parameter size</b></p> <p>How many learned connections a model has. More ≈ more capable, slower, pricier. "70B model" means 70 billion. Smaller models are often fine.</p>
<p><b>Model provider</b></p> <p>The company running the model. Anthropic (Claude), OpenAI (GPT), Google (Gemini), xAI (Grok). Cursor lets you switch between them.</p>	<p><b>Agent</b></p> <p>An AI mode with tools. It can read files, write files, run terminal commands, and iterate. It acts, not just talks. Creates checkpoints as it goes.</p>	<p><b>MCP</b></p> <p>Model Context Protocol. A standard connector for AI tools to talk to external systems (Jira, databases, custom services). Think USB for AI.</p>
<p><b>Reasoning / thinking</b></p> <p>Extra internal steps before answering. Helps on planning, debugging, multi-step analysis. Plan mode is this with a UI around it.</p>	<p><b>Hallucination</b></p> <p>Model is confidently wrong — invents a function, cites a missing file. It doesn't know it's wrong. Always verify references before you act.</p>	<p><b>Multimodality</b></p> <p>A model that takes images, audio, or video alongside text. Useful, but images and audio consume far more tokens than prose — watch the window.</p>

**Prompting styles, in 3 shapes** — *smaller, focused prompts tend to produce better outputs than broad, open-ended ones.*

<p>01 <b>Zero-shot</b></p> <p>Ask with no examples. Works for common tasks. Still review the output.</p> <p><i>"Write a function that validates email addresses."</i></p>	<p>02 <b>Few-shot</b></p> <p>Include a couple of input/output pairs. The model pattern-matches the shape. Good when output format matters.</p> <p><i>"Here are three input/output pairs. Now do the same for: ..."</i></p>	<p>03 <b>Chain of thought</b></p> <p>Ask the model to reason step by step. Better on debugging, planning, multi-step logic.</p> <p><i>"Before making changes, walk me through how token validation works. Think step by step."</i></p>
---	--	--