AI Cheat Sheet The Essentials on One Page

MARC BAADEN
2026

The Ladder — Five Levels of AI Use

LEVEL	NAME	EXAMPLES
0	Plain chat	Claude.ai, ChatGPT, Gemini
1	Customized chat	Claude Projects, custom GPTs
1.5	Tools & extensions	MCP, skills, n8n, Fellou, Dia
2	Agentic coding	Claude Code, Cursor, Copilot
3	Wide-open agents	OpenClaw, Computer Use, Cowork
4	Agent swarms	AutoGen, multi-agent pipelines

Five Things Everyone Must Know

Hallucinations — AI confidently states things that are false. This is structural, not a fixable bug. Always verify what matters.

Tokens — The unit of everything (cost, length, limits). Roughly ¾ of a word. A "200K context" = 200,000 tokens.

Context window — The AI's working memory. What doesn't fit is forgotten.

Knowledge cutoff — The model doesn't know what happened after its training date.

Cost — Chat is cheap/free. Agents and coding can cost dollars per session.

Privacy Decision Framework

Before using any AI tool, ask three questions:

1. **What data am I sharing?** (Casual question vs. confidential document)
2. **Who can see it?** (The company, partners, anyone?)
3. **Can I keep it local?** (Ollama, LM Studio, Apple Intelligence)

If the data is sensitive and a local option exists — use the local option.

Beyond Text: The Modality Landscape

MODALITY	STATE OF PLAY	KEY TOOLS
Image	Mature, high quality	Nano Banana Pro, Midjourney, Flux, Firefly
Audio	Transcription: excellent. Voice cloning: powerful	Whisper, ElevenLabs
Video	Impressive but costly and inconsistent	Sora, Runway, Kling
Avatars	Accessible, built on existing tech	HeyGen

Risk / Privacy / Ease — Level 3 Tools

TOOL	PRIVACY	AUTONOMY	SETUP
Claude Computer Use	Medium	High	Easy
Claude Cowork	Medium	Medium	Easy
OpenClaw (self)	Low	Very High	Medium
OpenClaw (cloud)	Medium	Very High	Medium
Perplexity	Medium	Low	Very Easy

Key Terms — One-Liners

Generative AI — AI that creates new content (text, images, audio, video).

LLM — Large language model. The engine behind most AI chat tools.

RAG — Retrieval-augmented generation. Giving AI your own documents as context.

API — How software talks to software. The invisible glue behind every integration.

Fine-tuning — Permanently retraining a model on your data (unlike RAG, which is temporary).

MCP — Model Context Protocol. Connectors that let AI use external tools and services.

Prompt engineering — The craft of communicating clearly with AI to get better results.

Prompt injection — Malicious input that tricks an AI agent into unintended actions.

Vector database — Stores document embeddings for RAG systems (Pinecone, Chroma, etc.).

Open-source model — A model you can download, inspect, and run yourself (Llama, Mistral, etc.).

WAT framework — Workflows, Agents, Tools. A three-layer architecture for complex AI projects.

Vibe coding — Writing software by describing what you want in plain English (coined by Andrej Karpathy).

Power Tips

1. End prompts with: "*Ask me any clarifying questions and iterate until you are 95% sure.*"
2. Give the AI a role before giving it a task.
3. Show it what good looks like — one example beats fifty words of instruction.
4. Say what you want, not what you don't want.
5. When it goes wrong, correct it — don't start over.
6. Break big tasks into explicit steps.
7. Paste your own writing and say "Match my style."

Anatomy of a Good Prompt

Role + Context. Tell the AI who it is and what situation it's in. "You are an experienced grant reviewer…"

Format + Constraints. Be explicit about the shape. "Give me a bulleted list, exactly five items, each under 20 words."

Examples. Show, don't tell. Paste one good example and say "Follow this pattern."

Chain of thought. For reasoning tasks, add "think step by step" or "show your reasoning before answering."

Start Here — By Profile

Curious beginner: Level 0 — Claude.ai or ChatGPT. End prompts with "Ask me questions until 95% sure." Graduate to Level 1 (Claude Projects) once you want it to know your work.

Knowledge worker: Level 1 with RAG — feed it your documents. MCP integrations (1.5) wire AI into your daily tools. Keep the privacy framework in mind.

Developer: Level 2 — Claude Code or Cursor. Learn WAT for complex automations. Try Level 3 (OpenClaw, Computer Use). Use Ollama for sensitive projects.

Full document: "AI: What You Need to Know — A Map for the Overwhelmed" by Marc Baaden (2026)