

Foundations

AI *n.*

"AI" is a catch-all term used to describe the latest smart program — chess, then LLMs, now "Agents". Try to name the technical piece you mean when relevant: the model, the harness, the agent, the context.

Model (LLM) *n.*

A large language model trained on internet-scale text. Stateless: does next-token prediction and nothing else. "Claude Opus" and "GPT-5.x" are models. On its own a model can't do anything agentic; it has to be harnessed.

Agent *n.*

A model harnessed with tools, a system prompt, and useful context, so it can run autonomously. Agents run multi-step workflows; chatbots just return responses.

Harness *n.*

Everything around the model that turns it into a powerful agent: tools, system prompt, context management, skills, memories, files. ChatGPT in the browser and Codex on your computer are using the same model, but have different harnesses.

Environment *n.*

The world the agent acts on — anything outside the harness that it perceives through tool results and changes through tool calls. Usually a filesystem; can be a database, an API, a browser.

Context window *n.*

The finite, model-specific span the model can see on one request, measured in tokens. Everything — system prompt, conversation history, tool results — must fit inside it. Lately, context windows are around 1M tokens.

Personal OS *n.*

A mega folder-of-folders holding all your notes, projects, data, and everything you want agents to be able to see or work on with you, with an AGENTS.md explaining the structure so agents can navigate it easily. Your personal-os is your personal 'harness' for AI, and lets you compound your progress, knowledge, and tools over time.

File system *n.*

The tree of files and directories everything on your computer lives in, which you explore using Finder or your Terminal. Your agent runs inside the file system!!

Prompting *n.*

The art and science of communicating with AI to get the outcomes you want. The way you talk to the model materially changes performance.

System prompt *n.*

The standing instructions the harness prepends to every request: who the agent is, how to behave, what tools it has.

Memory system *n.*

Anything that makes an agent stateful across sessions: persist information into files during a session, reload it at the start of the next. Memory compounds; sessions don't.

Files *n.*

The universal container of computing: documents, data, code, and configuration are all just files. Agents are better with files than with anything else — when in doubt, write it to a file.

AGENTS.md *file*

The file at the root of a project that agents read at the start of every session: who you are, what this project is, and the rules, preferences, and conventions to work by here. The open cross-harness convention — Claude Code also reads its own variant, CLAUDE.md.

Notes & reference as files *n.*

Your knowledge kept in plain files the agent can read — notes, todo lists, reference material, project docs. The substance of your Personal OS: your agent knows your world because your world is written down.

Design concept *n.*

The shared understanding of what's being built, held between you and the agent. The conversation, plans, and code all try to capture it; none of them are actually it.

Context engineering *method*

Deliberately structuring what goes into the context window using file systems, skills, prompt engineering, and other tactics to get the outcomes you want.