

The Claude Code Power User's Journey

From Interactive Chat to Fully Automated Co-pilot



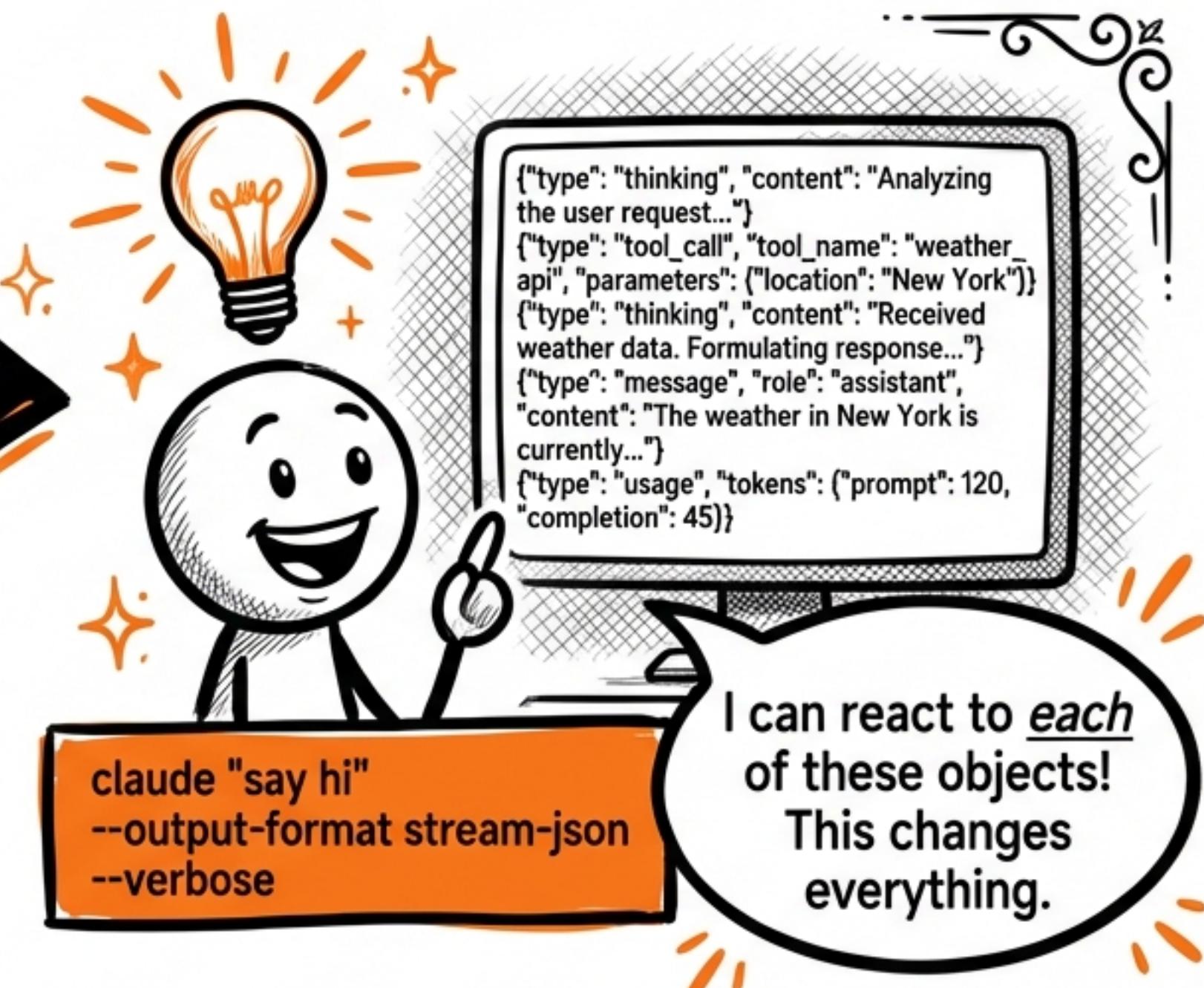
This journey follows a developer's progression through distinct stages of mastery, transforming Claude Code from a simple tool into a powerful, automated partner. The key isn't just learning features; it's a fundamental mindset shift.

The Threshold: It's Not a Chat, It's a JSONL Stream

Most users interact with Claude Code as a conversational partner. This is just the surface.

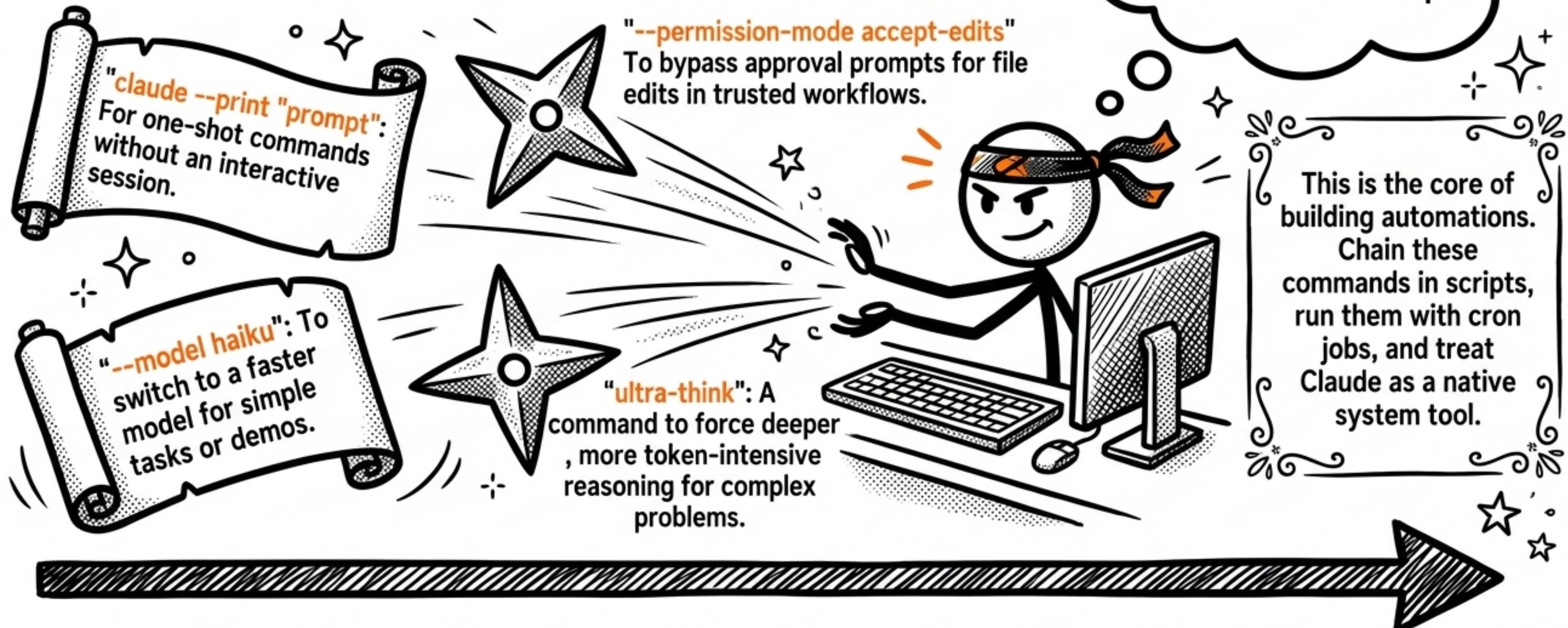


Power users understand the truth: Claude is a tool that takes a prompt and outputs a stream of individual JSON objects (JSONL). Every action—thinking, calling a tool, responding—is a structured object.



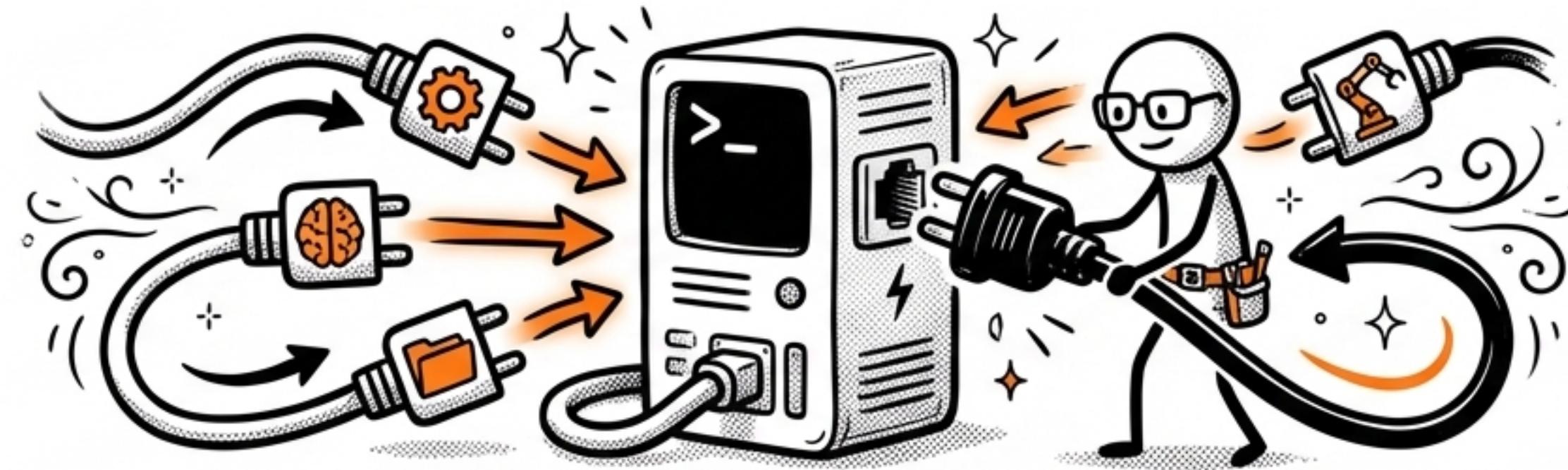
Stage 1: Master the Command Line

The first step to power usage is leaving the interactive UI behind. Direct CLI commands unlock scripting, automation, and precise control over Claude's behavior.



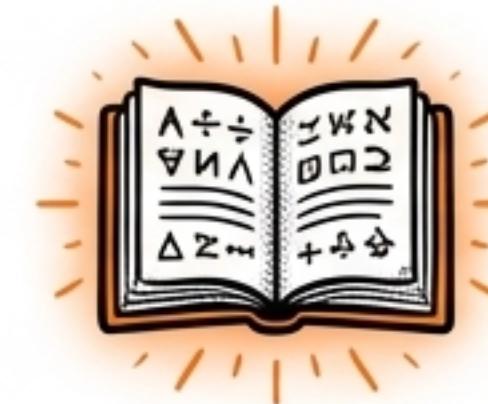
Stage 2: Expand the Toolbox with MCPs

MCPs are specialized, external tools that give Claude superpowers. By connecting them, you grant it the ability to browse the web, analyze code repositories with expert knowledge, and even run sandboxed development environments.



Playwright MCP

Gives Claude a fully controllable browser. Perfect for automating UI tasks, scraping data, or running integration tests.



Deep Wiki MCP

Provides AI-optimized documentation for open-source projects, based on the *actual source code*. Unlocks expert-level understanding of libraries.



RepoMix MCP

Packs an entire project into a single, context-friendly file. Ideal for feeding large codebases to models with massive context windows for high-level analysis or refactoring.



ContainerUse MCP

Runs tasks in a completely sandboxed Docker environment. Manages commits and branches automatically for safe, parallel experimentation.

Power in Practice: Automating the Web with Playwright

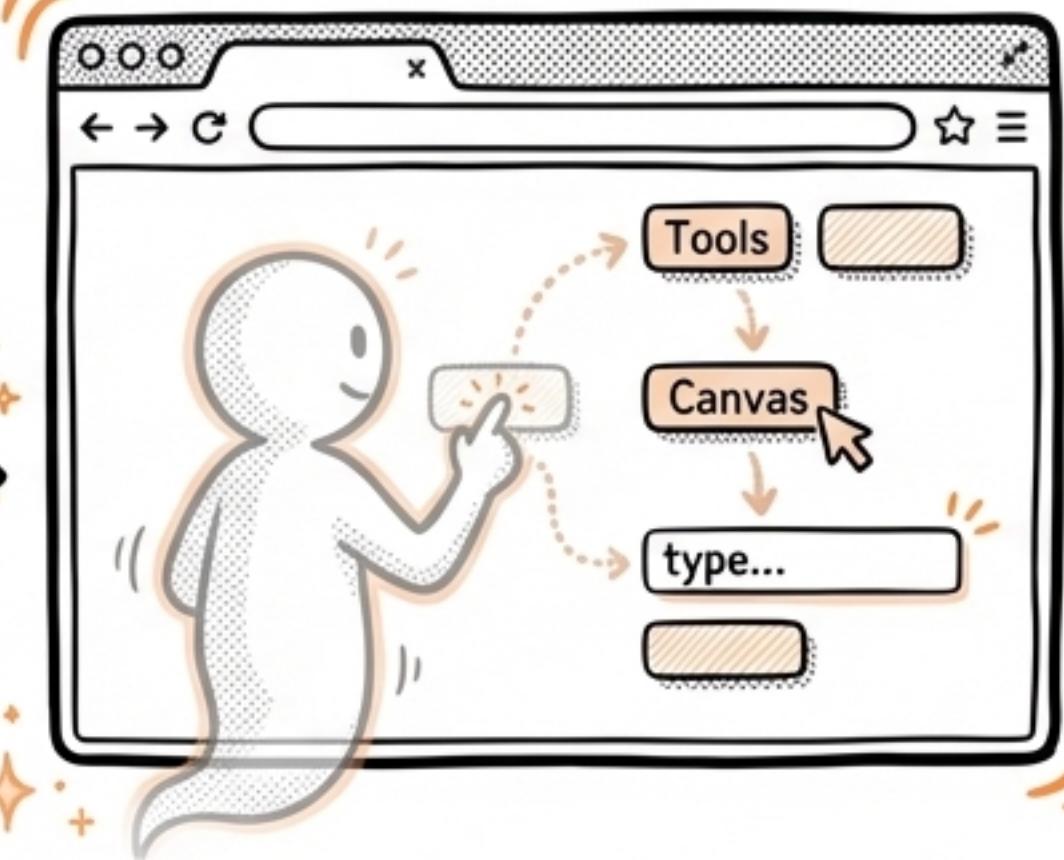
Go beyond simple web searches. The Playwright MCP allows Claude to orchestrate complex sequences of actions on any website. This is invaluable for automating repetitive tasks, generating variations of web content, or testing user flows.

Step 1

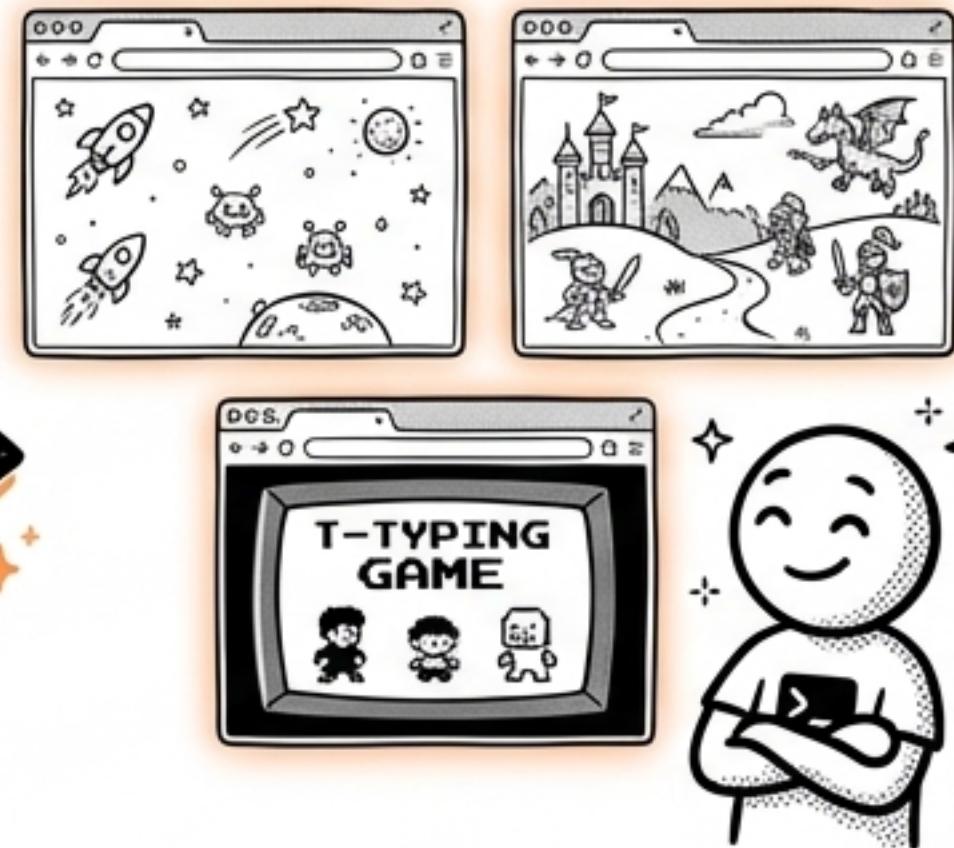
Navigate to the site, click "Tools", then "Canvas", enter a prompt to create a typing game, and hit enter.



Step 2



Step 3



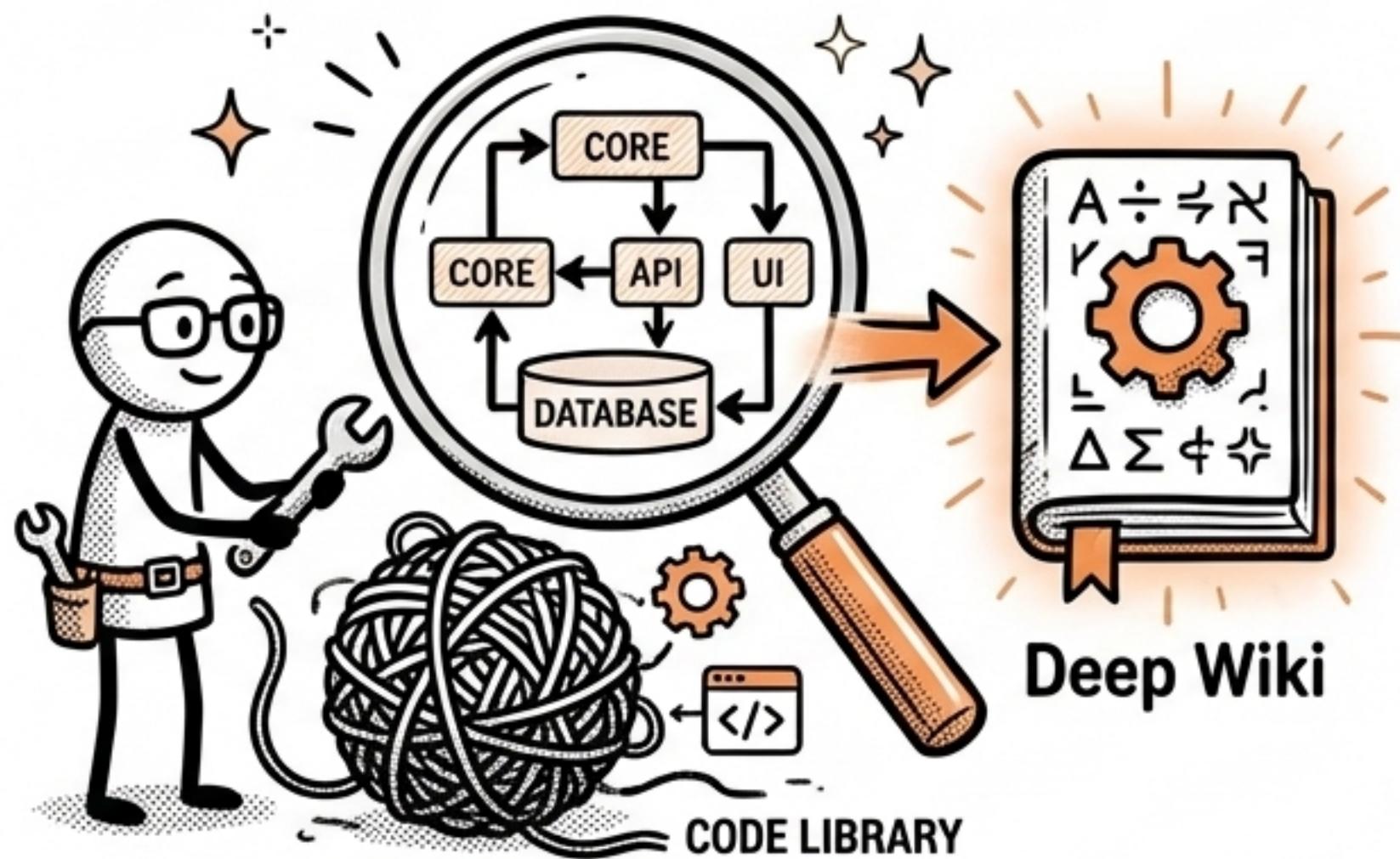
Orchestrate entire sessions.

By telling Claude to 'open three more tabs and do the same thing with different styles,' you can spin up parallel agents to explore multiple variations of a task, maximizing the value of your other AI accounts.

Engineering Perfect Context

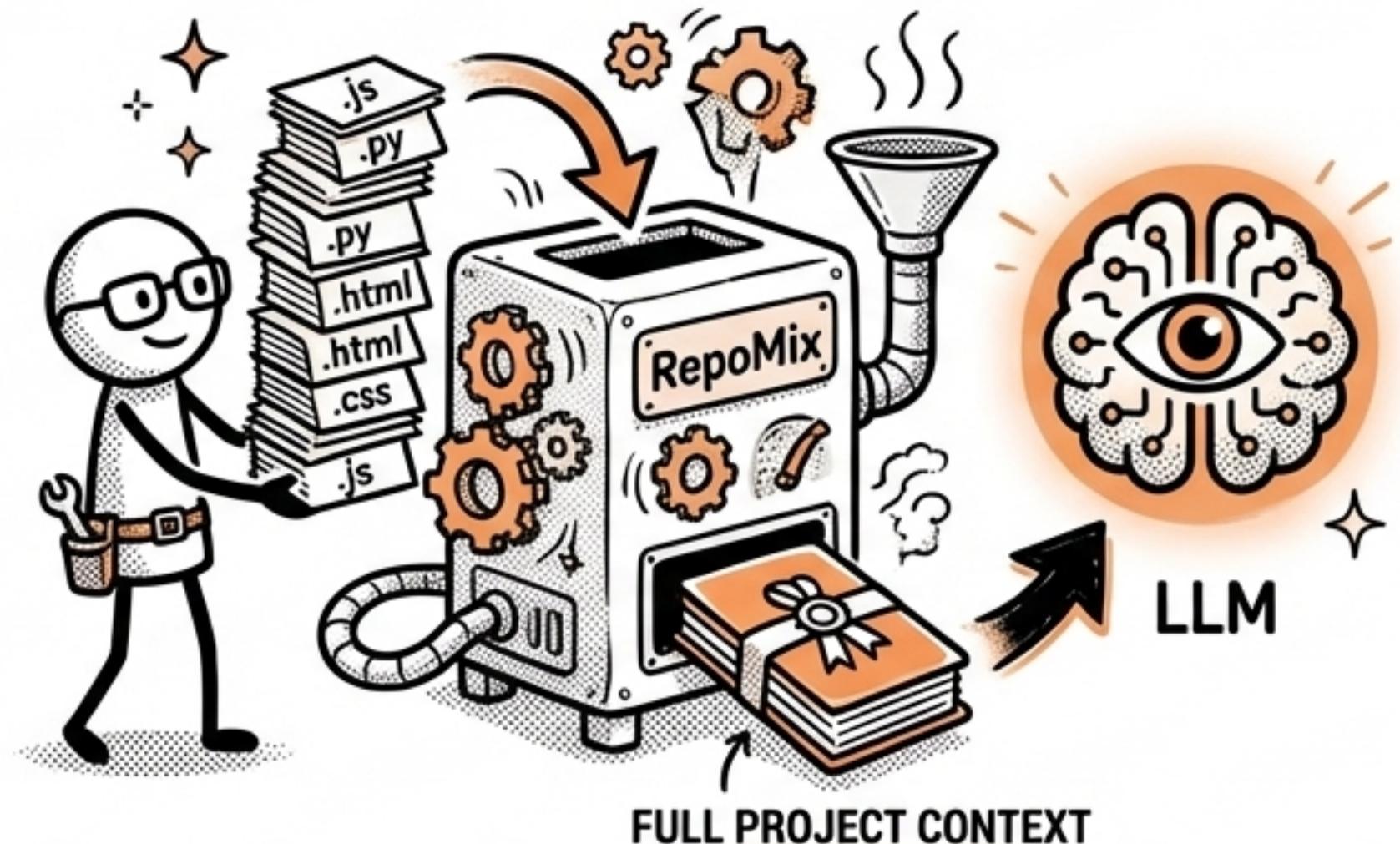
Better context will always win over better thinking. Power users spend time preparing the perfect context before starting a complex task.

Deep Wiki:



Provides diagrams and knowledge based on the code's source of truth, not outdated docs. Use it to give Claude an expert's understanding of how a library *actually* works.

RepoMix:



Packages an entire project—code, file paths, and all—into a single file. Perfect for analysis, cross-language ports, or finding gnarly bugs that span your entire application.

Stage 3: Building Your Own Custom Tools

True power comes from building your own tools. Custom agents and hooks allow you to encapsulate complex logic into simple commands and enforce consistent, project-specific rules.



Custom Agents (Sub-Agents):

Create reusable, specialized assistants.

A single prompt can "spin up three sub-agents" to tackle a "spin up three sub-agents" to tackle a problem from different perspectives (performance, security, project management) in parallel, each with its own isolated context.

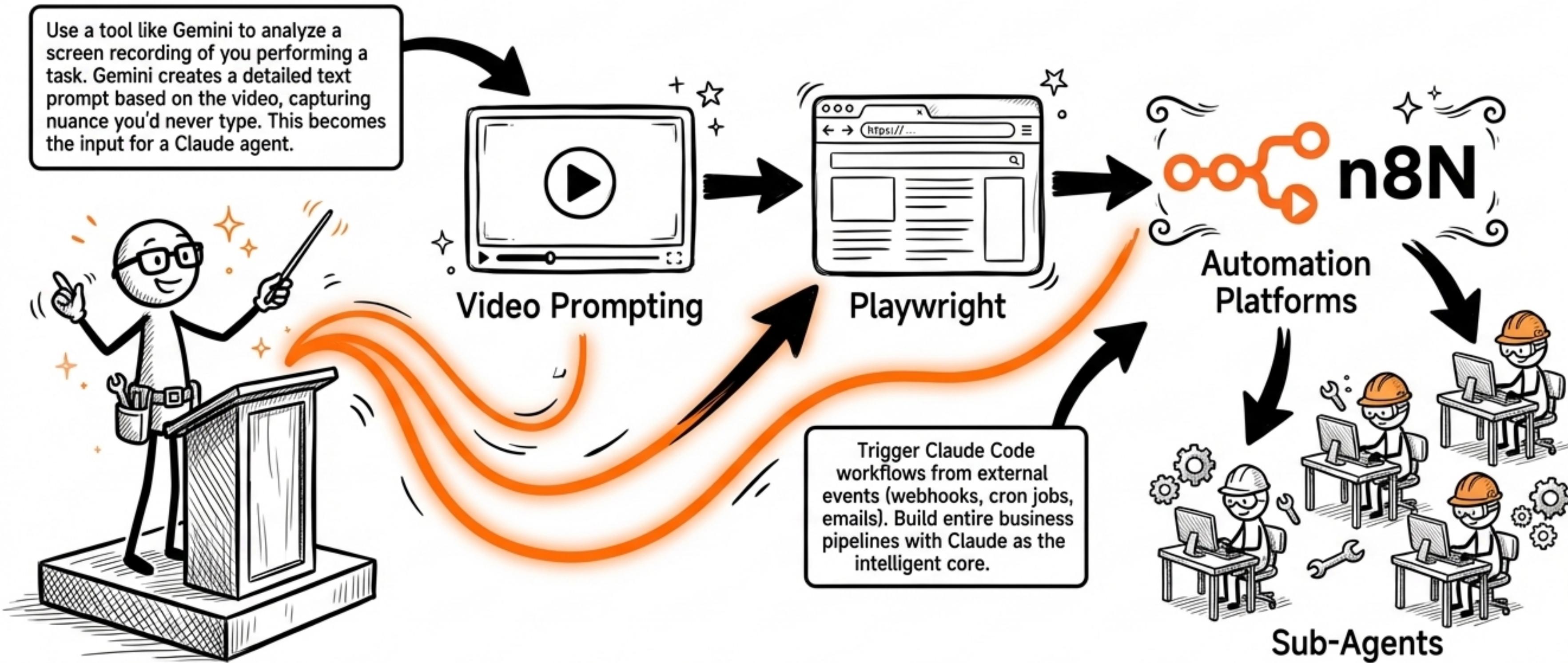
Hooks:

Intercept the JSONL stream to enforce rules.

A `pre-tool-use` hook can check a command before it runs. For example, it can block `npm install` and respond with "Use `bun` instead," ensuring project conventions are always followed.

Stage 4: Becoming the Orchestrator

The final stage is **orchestration**: combining all your tools into seamless, complex workflows. The power user becomes a conductor, guiding multiple agents and services to achieve a single, ambitious goal.



The Journey Never Ends

Becoming a Claude Code power user is not a one-time achievement. It's a continuous cycle of personalizing your tools to match the way *you* think and work.

- Automate Repetition:** If you do something more than twice, build a skill or custom agent for it.
- Control the Context:** Curate high-quality context with tools like RepoMix, Deep Wiki, and the search editor.
- Enforce Your Rules:** Use hooks as programmatic guardrails to ensure consistency and best practices.
- Combine Your Tools:** Orchestrate multi-tool, multi-agent workflows to tackle your most ambitious projects.

