



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Aider in Practice: Open-Source AI Pair Programming in the Terminal

Class Duration

14 hours of live training delivered over 2 days.

Student Prerequisites

- Professional software development experience in at least one language
- Working knowledge of Git (commits, diffs, branches, reverting)
- Comfort with the command line
- No prior AI tool experience required

Target Audience

Software engineers and teams who want a capable AI pair programmer without vendor lock-in. Aider is open source, model-agnostic, and git-first — making it especially relevant for organizations with procurement, data-residency, or AI-usage restrictions, and for developers who want to work with the model of their choice, including fully local open-weight models. Also valuable for teams using Claude Code, Codex, or Cursor who want an open-source option in their toolkit for comparison or fallback.

Description

Aider is the leading open-source AI pair programmer: it lives in your terminal, edits your files directly, and treats git as the source of truth — every AI change is a clean, attributed commit you can inspect, amend, or undo. This two-day course covers Aider's feature set end to end. Day one builds fluency with the core workflow: installing and configuring Aider, connecting it to the model of your choice (Anthropic, OpenAI, Gemini, DeepSeek, or local models via Ollama), managing the chat context with file commands and the repo map, and using the chat modes — code, ask, architect, and help — for the right task. We cover architect mode's two-model split (a reasoning model plans, an editor model applies) and when it pays for itself. Day two moves to power features and team adoption: auto-lint and auto-test loops that iterate until the code passes, conventions files for steering style, watch-files mode for driving Aider from comments in any IDE, voice and image input, scripting Aider for batch and CI use, cost tracking, and configuration management. A



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

dedicated module covers running Aider fully local with open-weight models for restricted environments. Students leave knowing where Aider shines, where full agents like Claude Code or Codex are the better tool, and how to make that call per task.

Learning Outcomes

- Install, configure, and connect Aider to hosted frontier models or local open-weight models.
- Manage chat context effectively: adding and dropping files, read-only references, and the repo map.
- Choose the right chat mode per task: code, ask, architect, and help.
- Use architect mode's planner/editor model split for complex, multi-file changes.
- Exploit Aider's git-first workflow: auto-commits, clean diffs, /undo, and attribution.
- Configure auto-lint and auto-test loops so Aider iterates until checks pass.
- Steer output with conventions files and per-project configuration.
- Drive Aider from any IDE with watch-files mode, and use voice and image input.
- Script Aider for non-interactive batch edits and CI integration.
- Run Aider fully local with Ollama for data-residency and compliance-constrained environments.
- Track per-session costs and choose models on the cost/capability curve.
- Judge when Aider is the right tool versus a fully agentic assistant.

Training Materials

Comprehensive courseware is distributed online at the start of class. All students receive a downloadable MP4 recording of the training.

Software Requirements

Python 3.10+ (for installing Aider), an API key for at least one hosted model or Ollama installed for local labs, Git, and a code editor of choice.



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Training Topics

Getting Started with Aider

- Installation options and project setup
- Connecting models: Anthropic, OpenAI, Gemini, DeepSeek, OpenRouter
- Model aliases and switching models mid-session
- The Aider philosophy: open source, model-agnostic, git-first

Managing Chat Context

- Adding and dropping files; read-only references
- The repo map: how Aider understands large codebases
- Keeping context small for better edits and lower cost
- Images and web pages as chat context

Chat Modes

- Code mode: direct edits to your files
- Ask mode: discussing code without changing it
- Architect mode: planner and editor models working together
- Choosing model pairings for architect mode
- Help mode and in-chat commands

Git-First Workflows

- Auto-commits with attributed, descriptive messages
- Reviewing AI changes with diffs before and after
- /undo, amending, and squashing AI commits
- Working on branches and integrating with PR review

Automation Loops

- Auto-lint: fixing lint errors automatically after each edit
- Auto-test: running your test command and iterating to green
- Edit formats: whole-file vs. diff-based editing
- Handling failure loops and knowing when to intervene

Steering and Configuration

- Conventions files for style and architectural guidance
- Project and user configuration files
- Per-model settings and prompt caching
- Cost tracking and token budgets



To discuss this course and customizations:
Call: 434-509-5680 or Email: sales@cloudcontraptions.com

Beyond the Terminal

- Watch-files mode: driving Aider from comments in any IDE
- Voice input for hands-free coding
- The browser UI
- Scripting Aider: non-interactive batch edits and CI usage

Local and Restricted Environments

- Running Aider fully local with Ollama and open-weight models
- Capability tradeoffs of local models for code editing
- Data-residency and compliance considerations
- Aider vs. full agents (Claude Code, Codex): choosing per task