

ARTIFICIAL INTELLIGENCE

INTERMEDIATE

3 DAYS

Agentic Workflows with Cursor

Move from using Cursor as a smarter editor to running it as an orchestration layer. Covers agent mode in depth, background and cloud agents, parallel execution across git worktrees, persistent rules, MCP integration, and event-triggered automations that run without you sitting there.

Why Agentic Workflows with Cursor?

Orchestration, not autocomplete

Run several agents across a codebase instead of driving one suggestion at a time.

Context that persists

Rules and project instructions so every agent run follows your team's standards.

Work that continues without you

Background and event-triggered agents that produce reviewable output while you do something else.

Honest about the ceiling

Clear on where parallelism helps and where review load becomes the real bottleneck.

What you'll be able to do

- ✓ Distinguish Tab, Composer, agent mode and background agents, and choose between them
- ✓ Operate the agent loop including terminal execution, guardrails and steering
- ✓ Write delegations with clear intent, scope and stop conditions
- ✓ Encode persistent context using project rules and agent instruction files
- ✓ Run background and cloud agents and review the pull requests they produce
- ✓ Execute parallel agents across git worktrees without collisions
- ✓ Extend Cursor with MCP servers connected to internal systems
- ✓ Configure event-triggered automations with appropriate guardrails

Who this is for

- Software developers already using AI coding tools
- Platform and DevOps engineers
- Tech leads and engineering managers
- Architects evaluating agentic development at team scale

Curriculum

01 The Cursor Agent Surface

- Tab, Composer, agent mode and background agents compared
- The agents view as a single place to see local, cloud and remote work
- Model selection per task and what changes when you switch
- Usage pools and credit consumption: what actually burns budget
- Choosing the lightest surface that solves the problem in front of you

02 Agent Mode in Depth

- How the agent reads the codebase, plans and iterates toward a result
- Terminal execution, test runs and self-correction loops
- Approving, interrupting and redirecting a run in progress
- Diff review discipline before anything lands
- **Hands-on:** deliver a multi-file feature end to end in agent mode

03 Writing Delegations That Complete

- Intent, scope and explicit stop conditions in a delegation
- Defining success criteria the agent can check for itself
- Decomposing large work into units an agent can finish
- Diagnosing a run that produced unusable output and correcting the brief
- **Hands-on:** rewrite a failed delegation until it succeeds

04 Rules and Persistent Context

- Project rules as the mechanism for encoding team standards
- Agent instruction files and how they layer with rules
- Scoping rules to directories, file types and specific work
- Keeping rules short enough that the important parts still land
- **Hands-on:** author rules for your own repository and test the effect

05 Background and Cloud Agents

- Sending work off-machine to run in an isolated sandbox
- What background agents can and cannot reach
- Receiving output as a pull request rather than a working directory
- Deciding what is worth delegating asynchronously
- **Hands-on:** dispatch a background agent and review what comes back

06 Parallel Agents and Git Worktrees

- Running multiple agents on separate parts of a codebase simultaneously
- Git worktrees for isolation and why they prevent most collisions
- Splitting work so parallel agents do not fight over the same files
- Merge strategy and handling conflicting agent output
- Review throughput as the real constraint on parallelism

07 Extending Cursor with MCP

- Model Context Protocol fundamentals in a Cursor context
- Connecting internal APIs, documentation and databases
- Credential handling, scoping and least privilege
- Interactive MCP surfaces inside agent conversations
- **Hands-on:** connect an MCP server and complete a task through it

08 Automations and Event-Triggered Agents

- Triggering agents from issue trackers, chat, version control and webhooks
- Designing always-on automations that do not create noise
- Guardrails, approval steps and human-in-the-loop checkpoints
- Monitoring automated agents and catching silent failure
- Knowing when an automation is costing more than it saves