

# Claude Code Built-in Tools Mastery

Learn every Claude Code tool: file ops, search, bash, web, agents, permissions, and hooks for predictable workflows.

**Artificial Intelligence** Instructor: Chris Carter • 10 sections • 57 lessons • 27.00h total

[View this course online](#) [Browse all courses](#)

## About this course

---

When you ask Claude to perform a task, you are hoping it figures out the right approach. But Claude Code provides a specific set of built-in tools with well-defined inputs, outputs, and behaviors. Understanding these tools is like learning a standard library or framework primitives. It gives you fine-grained control over what Claude can do.

This course provides a detailed overview of every built-in tool. You will learn exactly how each tool works, what it can and cannot do, and how to combine them into reliable workflows. Through video lessons and hands-on examples, you will learn file operations, search patterns, bash execution, web tools, agent orchestration, permission systems, and hook-based interception to create custom automation primitives.

## Course curriculum

---

### Foundations: Understanding the Tool System

- **How Built-in Tools Work as Primitives (18m)**  
Learn the philosophy behind Claude Code's tool system, why explicit tools provide better control than inference, and how to think about them like standard library functions.
- **Tool Execution Lifecycle and Context (22m)**  
Understand what happens when a tool is called: context injection, parameter validation, execution, and result handling. Includes real examples of execution flow.
- **Settings Precedence Hierarchy Explained (25m)**  
Master how user, project, local, and enterprise settings combine and override each other. Learn which settings take priority and how to debug configuration conflicts.
- **Environment Variables and Configuration Scope (15m)**  
Use the env key effectively, understand variable scope, and learn best practices for managing environment-specific configurations across different machines.
- **Model Configuration and Overrides (12m)**  
Configure default models for main agents and subagents, understand when to override models, and learn how model selection affects tool behavior and costs.
- **Setting Up Your Development Environment (20m)**  
Step-by-step guide to configuring Claude Code with proper tool permissions, hook directories, and settings.json structure. Includes downloadable configuration templates.

### Core File Tools: Read, Write, and Edit Operations

- **Read Tool: File Reading Behavior and Limits (28m)**  
Master the Read tool's exact behavior: line offsets, character limits, permission requirements, and when Read is preferred over Bash cat commands.
- **Write Tool: Creating New Files Safely (24m)**  
Learn Write tool parameters, file creation behavior, encoding handling, and best practices for creating new files without accidental overwrites.
- **Edit Tool: Modifying Existing Files (32m)**  
Understand the critical Read-before-Write requirement, how Edit handles different file types, and strategies for making precise modifications to existing code.
- **File Permissions and Access Patterns (18m)**  
Learn how Claude Code handles file permissions, what happens with read-only files, and how to configure tools to work with restricted filesystems.
- **Common File Operation Patterns and Pitfalls (20m)**  
Real-world examples of file operations gone wrong and how to avoid them. Learn when to use Read vs Edit, and how to handle large files efficiently.
- **Hands-on: Building a File Processing Workflow (35m)**  
Practical exercise: Create a skill that reads configuration files, modifies specific sections, and writes updates safely. Includes error handling and validation.

## Search Tools: Glob and Grep Mastery

- **Glob Patterns: Syntax and Performance (30m)**  
Master Glob pattern syntax including wildcards, character classes, and directory matching. Learn performance characteristics and when Glob is the right choice.
- **Advanced Glob Techniques and Use Cases (22m)**  
Learn negation patterns, directory exclusions, and complex matching scenarios. Real examples of finding files across nested directory structures.
- **Grep Tool: Regex Patterns and Output Modes (35m)**  
Understand Grep's regex engine, output modes (line numbers, context lines), and how to construct effective search patterns for code and text files.
- **Grep vs Glob: Choosing the Right Search Tool (18m)**  
Decision framework for when to use Grep vs Glob based on task requirements, performance needs, and result formatting. Includes performance benchmarks.
- **Combining Search Tools with File Operations (25m)**  
Learn to chain Glob and Grep with Read/Edit operations to build powerful file discovery and modification workflows. Real example: bulk code refactoring.
- **Hands-on: Building a Project Search Skill (40m)**  
Create a custom skill that searches across your codebase, finds patterns, and provides context-aware results with file previews and navigation.

## Bash Tool: Command Execution and Sandboxing

- **Bash Tool Execution Model and Timeouts (28m)**  
Understand how Bash commands execute, timeout behavior, background processes, and how output is captured and formatted for Claude's context.
- **Sandboxing Behavior and Security Boundaries (32m)**  
Learn about sandbox modes, excluded commands, network restrictions, and how Claude Code prevents potentially harmful operations while enabling legitimate automation.
- **Working with BashOutput and Exit Codes (20m)**  
Master handling command output, interpreting exit codes, and building robust workflows that respond appropriately to success and failure conditions.
- **Background Processes and Long-Running Commands (25m)**  
Learn techniques for managing long-running processes, background tasks, and how to interact with services that continue after command completion.
- **Debugging Bash Tool Issues (18m)**  
Common problems with Bash tool execution and how to solve them: permission denied errors, timeout issues, and output parsing problems.
- **Hands-on: Building a Deployment Automation Skill (45m)**  
Create a skill that runs build commands, handles errors gracefully, and provides real-time feedback on deployment status using Bash tool capabilities.

## Web Tools: Fetch and Search Capabilities

- **WebFetch Tool: URL Fetching and Content Extraction (30m)**  
Learn WebFetch's capabilities, how it handles different content types, prompt-based extraction, and permission patterns for accessing external resources.
- **WebSearch Tool: Query Construction and Results (28m)**  
Master search query syntax, domain filtering, result pagination, and how to work with search results effectively in your automation workflows.
- **Web Tool Limitations and Permission Models (22m)**  
Understand what WebFetch and WebSearch cannot do, network restrictions, rate limiting, and how to configure permissions for web access.
- **Combining Web Tools with File Operations (25m)**  
Learn to fetch web content and save it to files, extract structured data, and build workflows that combine online research with local processing.
- **Hands-on: Building a Research Assistant Skill (38m)**  
Create a skill that searches for information, fetches relevant pages, extracts key insights, and compiles research reports automatically.

## Agent Tools: Task and TodoWrite Orchestration

- **Task Tool: Subagent Execution and Context Passing (35m)**  
Understand when to use subagents, how to select models for tasks, and best practices for passing context between main agents and subagents.
- **Subagent Communication and Result Handling (28m)**  
Learn how subagents communicate results back to the main agent, handle errors in sub-tasks, and coordinate multiple parallel subagent executions.
- **TodoWrite Tool: Programmatic Task Management (20m)**  
Master using TodoWrite to create, update, and track tasks programmatically. Learn how to integrate todos into skills for complex multi-step workflows.
- **When to Use Subagents vs Direct Tool Calls (18m)**  
Decision framework for choosing between subagents and direct tool execution based on task complexity, isolation needs, and resource constraints.
- **Hands-on: Building a Multi-Agent Review System (42m)**  
Create a workflow that uses subagents for code review, testing, and documentation generation, then aggregates results into a comprehensive report.

## Permission System: Security and Access Control

- **Permission Rules: Allow, Ask, and Deny Evaluation (32m)**  
Master the exact evaluation order of permission rules, how allow/ask/deny patterns interact, and how to debug permission conflicts effectively.
- **Working Directories and Filesystem Boundaries (25m)**  
Understand how additionalDirectories works, filesystem access boundaries, and how to configure Claude Code to work across multiple project directories.

- **Deny Patterns: Protecting Sensitive Files (28m)**  
Learn to use glob patterns in deny rules to protect credentials, secrets, and sensitive configuration files from accidental or malicious access.
- **Sandbox Configuration and Security Modes (30m)**  
Understand different sandbox modes, how to configure security levels for different use cases, and balance safety with functionality requirements.
- **Hands-on: Implementing Enterprise Security Policies (35m)**  
Create a comprehensive permission configuration for a team environment that protects sensitive data while enabling productive automation workflows.

## **Hook System: Intercepting and Modifying Tool Behavior**

- **PreToolUse Hooks: Intercepting Tool Calls (35m)**  
Learn to intercept tool calls before execution, read tool context and parameters, and block execution with custom exit codes and error messages.
- **PostToolUse Hooks: Reacting to Tool Results (28m)**  
Master reacting to tool execution results, triggering follow-up actions, logging, and modifying behavior based on tool outcomes.
- **Hook Matchers: Targeting Specific Tools and Patterns (30m)**  
Understand hook matcher syntax for targeting specific tools, file patterns, and command types. Learn to create precise hook triggers.
- **Hook Security: Avoiding Infinite Loops and Recursion (22m)**  
Learn critical techniques for preventing hook recursion, managing settings inheritance in hooks, and ensuring hooks don't interfere with each other.
- **Advanced Hook Patterns and Use Cases (32m)**  
Explore advanced hook techniques: validation pipelines, automated testing triggers, compliance checks, and custom logging systems.
- **Hands-on: Building a Custom Validation System (40m)**  
Create a hook system that validates file modifications, enforces coding standards, and prevents certain operations based on your team's requirements.

## **Advanced Integration and Workflow Design**

- **Building Predictable Multi-Tool Workflows (38m)**  
Learn to chain tools together with confidence, handle errors at each step, and create workflows that produce consistent, reproducible results.
- **Creating Precise Skills with Explicit Tool References (32m)**  
Move beyond relying on agent inference, learn to reference specific tools in skill definitions for predictable, controlled automation behavior.
- **Debugging Tool Execution and Configuration Issues (25m)**  
Systematic approach to debugging: which tool is being called, why it's failing, and how to trace problems through the execution stack.
- **Performance Optimization and Tool Selection (28m)**  
Learn to choose the most efficient tool for each task, optimize execution order, and reduce latency in complex multi-step workflows.

- **Hands-on: Building a Complete CI/CD Integration (50m)**  
Create a comprehensive automation that uses file operations, search tools, bash commands, and hooks to implement a full CI/CD pipeline for a sample project.
- **Testing and Validating Tool-Based Workflows (35m)**  
Learn strategies for testing automation workflows, validating outputs, and ensuring your skills work correctly across different environments and conditions.

## Enterprise Configuration and Team Deployment

- **Enterprise Settings and Policy Management (30m)**  
Configure enterprise-wide settings, manage team permissions, and implement organizational policies that scale across multiple users and projects.
- **Centralized Hook and Skill Repositories (28m)**  
Set up shared repositories for hooks and skills, manage versioning, and enable team collaboration on automation tools and workflows.
- **Monitoring and Logging Tool Usage (25m)**  
Implement logging and monitoring for tool execution, track usage patterns, and gather insights for optimizing team workflows.
- **Security Auditing and Compliance (32m)**  
Learn to audit tool permissions, review hook configurations, and ensure your Claude Code setup meets organizational security and compliance requirements.
- **Hands-on: Deploying Team-Wide Automation Standards (45m)**  
Create a complete team deployment package with standardized skills, hooks, and permissions that can be distributed and maintained across your organization.
- **Course Conclusion and Next Steps (15m)**  
Review key concepts, explore advanced topics for further learning, and learn how to stay updated with new tool capabilities and best practices.

---

## Discover our blog

Learn more with free articles written by our professors and guest experts.

Latest article: **[Flight Attendant Preparation: What to Study Before You Apply](#)**

[Visit the blog](#)   [All courses](#)   [This course](#)