The Ultimate

Claude Code Productivity Guide

Slash Commands · Keyboard Shortcuts · Power Workflows

For beginners & pros alike · 2026 Edition

CC

Welcome

Claude Code is more than a terminal AI assistant — it's a full agentic coding runtime. Most users only scratch the surface. This guide puts everything you need in one place: the slash commands that save minutes every session, the keyboard shortcuts that create real flow, and the workflows that 10× your output.

> ■ *Pro tip: You don't need to memorize everything at once. Start with Section 1 slash commands, then layer in the shortcuts as they become natural.*

What's Inside

01	Slash Commands — Session Management
02	Slash Commands — Context & Memory
03	Slash Commands — Code & Review
04	Slash Commands — Setup & Config
05	Keyboard Shortcuts
06	Power Workflows
07	Quick Reference Cheat Sheet

01 · SLASH COMMANDS

Session Management

These commands control the lifecycle of your Claude Code sessions — starting, stopping, naming, and revisiting them.

Command	What it does
/clear	Wipes the entire conversation context and starts fresh. Best when switching to a completely unrelated task. File edits made during the session are kept.
/resume [session]	Browse or jump back into a previous session. Use arrow keys to scroll history, or pass a session ID directly to resume a specific one.
/rename [name]	Give the current session a meaningful name so you can find it later with /resume. Especially useful when juggling multiple projects.
/compact [focus]	Compresses older conversation history into a summary while keeping the thread alive. Pass a focus hint like 'retain error handling patterns' to control what's preserved.
/context	Shows how much of the context window you've used and when auto-compaction will kick in. Run this when responses start feeling slower or incomplete.
/cost	Displays token usage and estimated cost for the current session. Good habit to run at the end of long sessions.

■ *Rule of thumb: use /compact when context hits ~80%, and /clear when switching to a completely different task.*

© 2026 Claude Productivity Guide

02 · SLASH COMMANDS

Context & Memory

Claude has no memory between sessions by default. These commands let you build persistent context so Claude always knows your project and preferences.

Command	What it does
/init	Creates a CLAUDE.md file in your project root. Claude reads this at the start of every session — use it to capture project goals, tech stack, coding conventions, and anything Claude should always know.
/btw [question]	Ask a quick side question without adding it to the conversation history. Works even while Claude is mid-response. Ideal for quick lookups during long tasks.
/usage	Shows a breakdown of token usage across tools and messages in the current session.
/agents	Lists and manages sub-agents that Claude has spawned during the current session to handle parallel tasks.

■ *CLAUDE.md is your secret weapon. The more context you put in it — your stack, your style, your conventions — the less you repeat yourself every session.*

© 2026 Claude Productivity Guide

03 · SLASH COMMANDS

Code & Review

These commands turn Claude into a proactive coding partner — reviewing your work, planning before acting, and running tests.

Command	What it does
/review	Triggers a full code review of recent changes. Claude checks for bugs, security issues, performance problems, and style. Works best after a chunk of edits.
/plan	Switches Claude into planning mode — it reads and reasons about your codebase without making any changes. Use this before big refactors to get a roadmap first.
/fast	Enables Fast Mode, which uses speed-optimized API settings for the same model. Great for rapid interactive iteration and live debugging. Disable it when cost matters more than speed.
/effort [level]	Sets the reasoning depth: low, medium, high, or xhigh. Use xhigh for complex architectural decisions. Use low for quick, simple tasks to save tokens.
/model [name]	Switch models mid-session. Options: opus, sonnet, haiku. Use Opus for deep reasoning tasks, Sonnet for balanced everyday work, Haiku for fast simple tasks.
/doctor	Runs a health check on your Claude Code setup — checks keybindings, MCP servers, permissions, and config for errors or warnings.

© 2026 Claude Productivity Guide

04 · SLASH COMMANDS

Setup & Config

Configure your environment, connect external tools, and manage permissions without ever leaving the terminal.

Command	What it does
/mcp	Opens the MCP (Model Context Protocol) server manager. Connect external tools like Notion, GitHub, Slack, databases, and APIs so Claude can interact with them directly.
/permissions	View and manage what Claude is and isn't allowed to do — which tools it can use, which files it can edit, and which Bash commands it can run.
/config	Opens the settings interface for Claude Code. Adjust model defaults, auto-compaction thresholds, telemetry, and more.
/ide	Connects Claude Code to your VS Code or JetBrains IDE. Once connected, diffs open in your editor with full syntax highlighting instead of the terminal.
/terminal-setup	Configures your terminal so Option+Enter (Alt+Enter on Windows) creates a new line instead of submitting. Essential for writing multi-line prompts comfortably.
/install-github-app	Sets up the official Claude GitHub Actions integration so Claude can review PRs, comment on issues, and trigger workflows from GitHub.
/hooks	Manage lifecycle hooks — shell commands that run automatically when Claude edits a file, completes a task, or uses a specific tool. Great for auto-formatting or running tests.
/help	Lists all available slash commands with short descriptions. Type / and keep typing to filter in real time.

© 2026 Claude Productivity Guide

05

Keyboard Shortcuts

These shortcuts are the difference between fighting Claude Code and flowing through it. Learn the first five — the rest will come naturally.

Essential Shortcuts

Shortcut	Action
Escape	Interrupt Claude mid-response. Type 'go' to resume from where it stopped, or press Escape again to cancel entirely.
Escape × 2	Enter Rewind Mode. Scroll back through conversation history with the up arrow. Press Enter to restore code + conversation to that point, or restore conversation only.
Ctrl + C	Cancel the current action or discard a stashed prompt.
Ctrl + S	Stash your current in-progress prompt so you can type and send something else first. Your original prompt auto-restores afterward — no more clipboard juggling.
Shift + Tab	Cycle through autonomy modes: Default → Accept Edits → Auto → Plan. Use Auto mode to let Claude work through multi-step tasks without asking for approval each time.

Power Shortcuts

Shortcut	Action
Ctrl + G	Open your system's default text editor ($EDITOR) to compose a complex multi-paragraph prompt. Save and close to send it. Game-changer for long prompts.
Ctrl + F × 2	Kill all background agents instantly. Press Ctrl+F twice within 3 seconds. Use this when Claude is running parallel tasks that have gone off the rails.
Ctrl + L	Clear the terminal screen without clearing Claude's context. Keeps the conversation going but gives you a clean visual slate.
Alt + T	Toggle extended thinking mode on/off mid-session. Turn it on for complex reasoning tasks; turn it off when you want faster, lighter responses.
Ctrl + R	Search through previous commands in bash mode using reverse history search.

© 2026 Claude Productivity Guide

Ctrl + O	Toggle between normal transcript view and verbose mode (shows all tool calls as they happen). Useful for debugging what Claude is actually doing.
Shift + Enter	Add a new line in your prompt without submitting. Write multi-step instructions in one go before hitting Enter.
Tab	Autocomplete file paths and slash command names. Double-tap Tab to see all available completions.
Up Arrow	Recall your previous prompt. Keep pressing to go further back in input history.

© 2026 Claude Productivity Guide

06

Power Workflows

Commands and shortcuts become truly powerful when combined into repeatable workflows. Here are the patterns that experienced Claude Code users rely on every day.

Workflow	How to do it
Start a new project	Run /init to create CLAUDE.md → fill it with your stack, conventions & goals → run /plan to let Claude map the codebase → then start your first task. This one-time setup pays dividends every session.
Context getting full?	Run /context to check usage → if above 80%, run /compact with a focus hint → continue working. Never lose context mid-task again.
Pre-PR code review	After finishing a feature, run /review → read Claude's feedback → address issues → run /review again to verify → commit. Catch bugs before your teammates do.
Long task, side question	Claude is working on a big refactor → you need a quick answer → type /btw your question → read the response in a popup → dismiss with Space → Claude keeps going. Zero context disruption.
Switching tasks cleanly	Finish task → run /cost to log usage → run /clear to wipe context → start next task fresh. Context bleed between tasks is a silent productivity killer.
Multi-step prompt writing	Press Ctrl+G to open your text editor → write a detailed, structured prompt with numbered steps → save and close → Claude gets the full picture. Better prompts = fewer back-and-forths.
Undo a bad refactor	Claude went in the wrong direction → press Escape to stop → press Escape × 2 for Rewind Mode → scroll to the last good point → press Enter → choose 'Restore code and conversation'. Instant undo.
Automate repetitive tasks	Create a .md file in ~/.claude/commands/ with your workflow as instructions → it becomes a custom slash command → invoke it any time with /your-command-name. Build your personal command library.
Speed vs depth trade-off	For quick, interactive work: /fast + /effort low. For deep architectural work: /model opus + /effort xhigh. Switch mid-session without restarting.
Connect external tools	Run /mcp → add a server (e.g., claude mcp add notion https://mcp.notion.com/mcp) → Claude can now read/write Notion pages, query databases, and more directly in your session.

© 2026 Claude Productivity Guide

07

Quick Reference Cheat Sheet

Cut this out. Stick it on your wall.

Slash Command	Action
/clear	Wipe context
/compact	Compress history
/resume	Return to session
/rename	Name a session
/context	Check token usage
/cost	Check cost
/init	Create CLAUDE.md
/btw	Side question
/review	Code review
/plan	Plan before coding
/fast	Speed mode on/off
/effort	Set reasoning depth
/model	Switch model
/mcp	Connect tools
/doctor	Health check
/help	List all commands

Shortcut	Action
Escape	Interrupt response
Esc × 2	Rewind mode
Ctrl+S	Stash prompt
Ctrl+C	Cancel / discard
Shift+Tab	Cycle auto mode
Ctrl+G	Open text editor
Ctrl+F × 2	Kill all agents
Ctrl+L	Clear screen
Alt+T	Toggle thinking
Ctrl+O	Toggle verbose
Shift+Enter	New line in prompt
Tab	Autocomplete
Up Arrow	Previous prompt
Ctrl+R	Search history

■ *You now have everything you need to work with Claude Code at full speed. Start with /init on your next project, add /compact to your muscle memory, and build your first custom command. The 10× is in the habits, not just the shortcuts.*

© 2026 Claude Productivity Guide