

Claude Code

Command Reference

Every terminal command, every option and every slash command inside a session — with an explanation of what it does.

Compiled by Christian's virtual server · 12 September 2026 · Update of the edition of 11 September 2026 (2.1.268)

How this list came about

Terminal commands and options (parts A and B) are read straight out of the installed version 2.1.269 — `claude --help` and the help of every subcommand.

The slash commands (part C) come from the command definitions inside the program itself. The extraction also picks up commands whose description is only assembled at runtime or fetched from a variable, and since this edition equally those whose *name* sits in a variable instead of being spelled out in the code. Both are artefacts of the code being minified at build time and say nothing about the command itself.

Commands that exist in the program but are switched off by a flag are listed separately at the end, so that nothing appears available that is not.

What has changed since the last edition

Compared with 2.1.268: one new slash command, `/output-style` (list output styles or switch to one) — the reference now lists 139 slash commands. No command was removed, and no existing command's description changed. 37 commands had only internal, non-visible identifiers renamed, which has no effect on anyone reading the reference.

`claude --help` came out byte-for-byte identical. The sub-command `help` has one substantive change: `claude plugin eval` now describes in more detail what the command does and what to watch for — it loads the plugin and runs its eval suite (prompts, graders; scaffold scripts and real MCP servers only when you opt in) on your own machine, so only plugins you trust should be evaluated. The first run in an untrusted plugin directory asks for confirmation (`--trust-plugin` answers that for CI).

Part A · Terminal commands and options

Call form: `claude [options] [command] [input]`. With no arguments it starts a session in the current folder.

Starting a session

<code>claude</code>	Starts a session in the current folder.
<code>claude "your question"</code>	Starts with a first message.
<code>claude -p "..."</code>	Answers once and exits — for scripts.

Options

<code>--add-dir <directories...></code>	Additional folders Claude is allowed to touch.
<code>--agent <agent></code>	Agent for this session; overrides the setting.
<code>--agents <json></code>	Define custom agents inline as JSON.

<code>--allow-dangerously-skip-permissions</code>	Allows turning off all prompts later, without doing so right away.
<code>--allowedTools, --allowed-tools <tools...></code>	Allow only these tools, e.g. "Bash(git *) Edit".
<code>--append-system-prompt <prompt></code>	Append text to the system prompt.
<code>--autocompact <auto tokens></code>	When to compact automatically (auto, or 100k–1M).
<code>--ax-screen-reader</code>	Screen-reader output: flat text, no borders, no animations.
<code>--bg, --background</code>	Start the session in the background and return right away. Prints the id that claude attach, logs, stop and rm take; claude agents lists them. With <code>--resume</code> , continues that session in the background under the same ID, or starts a copy and says so when the session is already running.
<code>--bare</code>	Frugal mode: no hooks, plugins, auto-memory or CLAUDE.md discovery.
<code>--betas <betas...></code>	Send beta identifiers to the API (API key users only).
<code>--brief</code>	Lets Claude send you short progress notes while it works.
<code>--chrome</code>	Turn on the Claude-in-Chrome integration.
<code>--cloud [description session_id url]</code>	Create a cloud session or attach to an existing one.
<code>-c, --continue</code>	Continue the most recent conversation in this folder.
<code>--dangerously-skip-permissions</code>	Turn off all prompts. Only for sealed-off environments without internet.
<code>-d, --debug [filter]</code>	Turn on debugging, optionally filtered (e.g. "api,hooks").
<code>--debug-file <path></code>	Write the debug log to this file.
<code>--disable-slash-commands</code>	Turn off all skills.
<code>--disallowedTools, --disallowed-tools <tools...></code>	Block these tools.
<code>--effort <level></code>	Thinking effort for the session: low, medium, high, xhigh, max.
<code>--environment <environment_id></code>	Start a cloud session in a self-hosted environment.
<code>--exclude-dynamic-system-prompt-sections</code>	Keep machine-specific details out of the system prompt (better caching).
<code>--fallback-model <model></code>	Backup model for when the chosen one is overloaded (now also outside <code>--print</code>).
<code>--file <specs...></code>	Download files at startup (file_id:target_path).
<code>--fork-session</code>	On resume, issue a new session ID instead of continuing the old one.
<code>--forward-subagent-text</code>	Pass through the text and thinking of subagents as well.
<code>--from-pr [value]</code>	Resume the session belonging to a pull request (number or URL).
<code>-h, --help</code>	Show help.
<code>--ide</code>	Connect to the development environment automatically at startup.
<code>--include-hook-events</code>	Write hook events into the output stream as well (only with <code>--print</code>).
<code>--include-partial-messages</code>	Emit parts of the answer as they arrive (only with <code>--print</code>).
<code>--input-format <format></code>	Input format: text or stream-json (only with <code>--print</code>).
<code>--json-schema <schema></code>	JSON schema the answer is validated against.

<code>--max-budget-usd <amount></code>	Dollar ceiling for the API calls (only with <code>--print</code>).
<code>--mcp-config <configs...></code>	Load MCP servers from JSON files or strings.
<code>--model <model></code>	Model for this session.
<code>-n, --name <name></code>	Display name for the session.
<code>--no-chrome</code>	Turn off the Chrome integration.
<code>--no-session-persistence</code>	Do not write the session to disk.
<code>--output-format <format></code>	Output format: text, json or stream-json (only with <code>--print</code>).
<code>--permission-mode <mode></code>	Permission mode: <code>acceptEdits</code> , <code>plan</code> , <code>bypassPermissions</code> , <code>default</code> .
<code>--permission-prompts <target></code>	Who answers permission prompts with <code>--print</code> : "host" (the SDK host or <code>--permission-prompt-tool</code>) or "none" (auto-deny; the permission mode still decides everything else). Default: host.
<code>--plugin-dir <path></code>	Load a plugin from a folder or archive, for this session only; a folder holding several plugins loads all of them.
<code>--plugin-url <url></code>	Load a plugin archive from a URL, for this session only.
<code>-p, --print</code>	Print the answer and exit — for scripts and pipes.
<code>--prompt-suggestions [value]</code>	Turn on prompt suggestions.
<code>--remote-control [name]</code>	Start the session with Remote Control (phone, claude.ai).
<code>--remote-control-session-name-prefix <prefix></code>	Name prefix for auto-named Remote Control sessions.
<code>--replay-user-messages</code>	Echo incoming user messages back for acknowledgement.
<code>-r, --resume [value]</code>	Resume a conversation by ID, or open the picker.
<code>--safe-mode</code>	Start without any customisation: no CLAUDE.md, skills, plugins, hooks.
<code>--session-id <uuid></code>	Use a fixed session ID.
<code>--setting-sources <sources></code>	Which setting sources to load: user, project, local.
<code>--settings <file-or-json></code>	Additional settings from a file or a JSON string.
<code>--strict-mcp-config</code>	Use only the MCP servers given via <code>--mcp-config</code> .
<code>--system-prompt <prompt></code>	Replace the system prompt for the session.
<code>--system-prompt-snapshot <on off></code>	Record the system prompt once per conversation and reuse it verbatim on every request and resume (recommended: on). On by default for the built-in prompt; <code>--system-prompt</code> or <code>--append-system-prompt</code> turns it off so the given text applies fresh each launch.
<code>--teleport [session]</code>	Resume a teleport session, optionally by session ID.
<code>--tmux</code>	Create a tmux session for the worktree (needs <code>--worktree</code>).
<code>--tools <tools...></code>	Pick the built-in tools: "" for none, "default" for all, or a list.
<code>--verbose</code>	Override the verbose setting from the configuration.
<code>-v, --version</code>	Show the version number.
<code>-w, --worktree [name]</code>	Create a new git worktree for this session (name optional).

Part B · Subcommands

Calls of the form `claude <subcommand>`. Indented below each one, its own subcommands.

claude agents	Manage background agents.
claude attach	Open a background session in this terminal.
claude auth	Manage sign-in.
claude auth login	Sign in to the Anthropic account.
claude auth logout	Sign out.
claude auth status	Show sign-in status.
claude auto-mode	View or reset the settings of automatic mode.
claude auto-mode config	Print the effective configuration as JSON.
claude auto-mode critique	Get feedback on your own rules.
claude auto-mode defaults	Print the built-in default rules.
claude auto-mode reset	Reset to the default rules.
claude doctor	Check the installation. For the full examination use <code>/doctor</code> in a session.
claude gateway	Run a company gateway for sign-in and telemetry.
claude import	Take over the configuration of another AI tool.
claude install	Install the native build (stable, latest or a fixed version).
claude logs	Show a background session's recent terminal output.
claude mcp	Set up and manage MCP servers.
claude mcp add	Add an MCP server (stdio, http or sse).
claude mcp add-from-claude-desktop	Import servers from Claude Desktop (Mac and WSL only).
claude mcp add-json	Create a server from a JSON string.
claude mcp get	Show the details of one server.
claude mcp list	List the configured servers.
claude mcp login	Sign in to a server.
claude mcp logout	Sign out of a server.
claude mcp remove	Remove a server.
claude mcp reset-project-choices	Reset the per-project approvals.
claude mcp serve	Run Claude Code itself as an MCP server.
claude plugin	Manage plugins.
claude plugin details	Contents of a plugin and its estimated context cost.
claude plugin disable	Turn a plugin off.

<code>claude plugin enable</code>	Turn a plugin on.
<code>claude plugin eval</code>	Run test cases against a plugin and score them.
<code>claude plugin init new</code>	Scaffold a new plugin.
<code>claude plugin install i</code>	Install a plugin from a marketplace.
<code>claude plugin list</code>	List the installed plugins.
<code>claude plugin marketplace</code>	Manage marketplaces.
<code>claude plugin prune autoremove</code>	Remove plugins that are no longer needed.
<code>claude plugin tag</code>	Tag plugins with attributes.
<code>claude plugin uninstall remove</code>	Uninstall a plugin.
<code>claude plugin update</code>	Update a plugin.
<code>claude plugin validate</code>	Check a plugin for errors.
claude project	Manage project state.
<code>claude project purge</code>	Delete all stored state of a project.
claude respawn	Restart background sessions so they run the current version.
claude rm	Delete a background session, and its worktree when safe.
claude setup-token	Create a long-lived authentication token (for scripts and CI).
claude stop kill	Stop a background session; its conversation is kept.
claude ultrareview	Start a multi-stage code review in the cloud.

Part C · Slash commands inside a session

134 available commands, in alphabetical order. In grey behind them: second names and expected arguments.

<code>/add-dir <path></code>	Add another working folder.
<code>/advisor</code>	Let a stronger model weigh in at the important moments.
<code>/agents</code>	Removed. Subagents now live in <code>.claude/agents/</code> .
<code>/artifact-capabilities</code>	What published pages can do at runtime.
<code>/artifact-components</code>	A kit of reusable building blocks for published pages.
<code>/artifact-design</code>	Design rules for published pages.
<code>/artifact-diagramming</code>	Diagrams for published pages.
<code>/artifact-pr-review [pr number or url]</code>	Publishes the review of a pull request as a shareable page: bottom line, recommendation, judgment calls, visual explainer, blind spots.
<code>/artifacts</code>	Browse published and shared artifacts.
<code>/auto-mode-setup [--request-id <uuid>] [--wizard posture=\u2026 scope=\u2026 depth=\u2026 --propose --expect-sha256 <64-hex> --apply-file <path>]</code>	Set up and tune automatic mode.
<code>/autocompact [auto <tokens>]</code>	Decide at which fill level the context is compacted automatically.

<code>/autofix-pr</code>	Watch the current pull request and fix what breaks.
<code>/background</code> also <code>/bg</code> · [prompt]	Send the session to the background and free up the terminal.
<code>/batch</code> <instruction>	Plan a large change, then have it carried out in parallel.
<code>/branch</code> [name]	Fork the conversation at this point.
<code>/brief</code>	Toggle brief mode.
<code>/btw</code> [question]	Ask a side question without disturbing the current thread.
<code>/bug</code> also <code>/share</code> · [report]	Report a bug or share the conversation.
<code>/chrome</code>	Open the settings for Claude in Chrome.
<code>/claude-api</code>	Reference for the Claude API (models, prices, tools).
<code>/claude-code-docs</code> [question]	Answer questions about Claude Code itself (commands, flags, settings, hooks, skills), checked against the installed build.
<code>/claude-in-chrome</code>	Drive the browser: click, type, read pages.
<code>/clear</code> also <code>/reset</code> , <code>/new</code> · [name]	New session with an empty context; the old one is kept.
<code>/cloud-plugins</code>	Decide whether cloud sessions use the plugins enabled here.
<code>/code-review</code> also <code>/review</code> · [...] [--fix] [--comment] [<pr#> <branch> <path>]	Review changes for bugs and simplifications.
<code>/color</code>	Colour of the prompt box for this session.
<code>/commit</code> [guidance]	Creates a git commit, with a fitting message written for it.
<code>/commit-push-pr</code>	Commit, push and open a pull request — in one go.
<code>/compact</code> <optional custom summarization instructions>	Free up context by summarising what happened so far.
<code>/config</code> also <code>/settings</code> · [key=value]	Open the settings.
<code>/context</code> [all]	Show context usage as a coloured grid.
<code>/copy</code>	Copy the last answer to the clipboard (<code>/copy N</code> for older ones).
<code>/cowork-plugin</code>	Creates a new Cowork plugin or adapts an existing one.
<code>/daemon</code>	Manage background services and routines.
<code>/dataviz</code>	Design charts and analytics pages properly.
<code>/debug</code> [issue description]	Turn on debugging for this session.
<code>/design</code> [sync login consent revoke import export status <prompt>]	Grant or revoke access to design projects.
<code>/design-consent</code>	Grant access to design projects.
<code>/design-login</code>	Unlock design access for <code>/design-sync</code> .
<code>/design-revoke</code>	Revoke access to design projects.
<code>/design-sync</code>	Push a React design system to <code>claude.ai/design</code> .
<code>/desktop</code> also <code>/app</code>	Continue the running session in the desktop app.
<code>/diff</code>	Look at uncommitted changes and changes per turn.
<code>/doctor</code> also <code>/checkup</code>	Examine the installation and fix problems.
<code>/effort</code>	Set the thinking effort.

<code>/exit</code> also <code>/quit</code>	End the session.
<code>/explain-usage</code>	Have your own usage explained to you.
<code>/export</code> <code>[filename]</code>	Write the conversation to a file or the clipboard.
<code>/extra-usage</code>	Renamed to <code>/usage-credits</code> .
<code>/fast</code> <code>[on off]</code>	Toggle fast mode (same model, higher speed).
<code>/feedback</code> <code>[report]</code>	Send feedback to Anthropic.
<code>/fewer-permission-prompts</code>	Spot frequent harmless calls and allow them for good.
<code>/focus</code>	Show only the input, the summary and the answer.
<code>/fork</code> <code><directive></code>	Start a background agent that inherits the whole conversation.
<code>/goal</code> <code>[<condition> clear]</code>	Set a goal Claude checks before it stops.
<code>/heapdump</code>	Write a memory dump (for troubleshooting).
<code>/help</code>	Show help and the available commands.
<code>/hooks</code>	Look at the hooks configured for tool events.
<code>/ide</code> <code>[open]</code>	Manage the connection to the development environment.
<code>/import</code> <code>[codex gemini cursor]</code> <code>[--dry-run]</code>	Take over the configuration of another AI tool.
<code>/init</code>	Generate a first CLAUDE.md from the existing code.
<code>/insights</code>	Build a report about your own sessions.
<code>/install</code> <code>[options]</code>	Install the native build.
<code>/install-github-app</code>	Set up Claude GitHub Actions for a repository.
<code>/install-slack-app</code>	Install the Claude app for Slack.
<code>/keybindings</code>	Open the keyboard shortcuts file.
<code>/limit-reset</code>	Reset your session limit now and keep working; available once a week, still counts toward the weekly limit.
<code>/list-agents</code> also <code>/peers</code>	List subagents and other sessions you can reach.
<code>/login</code>	Sign in to the Anthropic account or switch accounts.
<code>/logout</code>	Sign out of the Anthropic account.
<code>/loop</code> also <code>/proactive</code>	Run a task over and over at a fixed interval.
<code>/low-priority</code>	Continue after hitting the session limit at lower priority; run again to stop.
<code>/mcp</code> <code>[reconnect <server> enable disable</code> <code>[<server> all]]</code>	Manage MCP servers.
<code>/memory</code>	Edit CLAUDE.md files and the memory settings.
<code>/memory-types</code>	Reference on the kinds of memory: what belongs where, and when.
<code>/mobile</code> also <code>/ios</code> , <code>/android</code>	Show the QR code for the phone app.
<code>/model</code> <code>[model]</code>	Set the model for Claude Code.
<code>/output-style</code> <code>[style]</code>	List output styles or switch to one.
<code>/passes</code>	Give away a free week of Claude Code.
<code>/permissions</code> also <code>/allowed-tools</code>	Manage the rules for allowed and forbidden tools.
<code>/plan</code> <code>[open <description>]</code>	Turn on plan mode or look at the session's plan.

<code>/plugin</code> also <code>/plugins</code> , <code>/marketplace</code>	Manage plugins.
<code>/plugin-authoring</code>	Helps you write and debug a plugin made of function hooks.
<code>/plugin-types</code> <code>[dir]</code>	Write two type files: the plugin API's TypeScript declarations and the inputs of the connected MCP tools, for typing a hooks module against this session.
<code>/powerup</code>	Get to know Claude Code's features in short lessons.
<code>/privacy-settings</code>	View and change the privacy settings.
<code>/pro-trial-expired</code>	Your options once the Pro trial has run out.
<code>/prototype</code>	Turns an idea into a working draft and publishes it as a page.
<code>/radio</code>	Turn on the lo-fi radio.
<code>/rate-limit-options</code>	Show the options once the usage limit is reached.
<code>/recap</code>	Produce a one-line summary of the session.
<code>/release-notes</code>	Show what is new in each version.
<code>/reload-plugins</code> <code> [--force]</code>	Apply pending plugin changes.
<code>/reload-skills</code>	Read in newly added or changed skills.
<code>/remote-control</code> also <code>/rc</code>	Drive the session from your phone or from <code>claude.ai</code> .
<code>/remote-env</code>	Choose the default environment for cloud agents.
<code>/rename</code> also <code>/name</code> · <code>[name]</code>	Rename the conversation.
<code>/resume</code> also <code>/continue</code> · <code>[conversation id or search term]</code>	Continue an earlier conversation.
<code>/rewind</code> also <code>/checkpoint</code> , <code>/undo</code>	Roll code and/or conversation back to an earlier state.
<code>/run</code>	Start the application and check a change in real operation.
<code>/run-skill-generator</code>	Turn a workflow of yours into a new skill.
<code>/sandbox</code>	Set up and inspect the sandbox.
<code>/schedule</code> also <code>/routines</code>	Create and manage scheduled cloud agents.
<code>/scroll-speed</code>	Set the scroll speed of the mouse wheel.
<code>/session</code> also <code>/remote</code>	Show the address and QR code of the cloud session.
<code>/setup-bedrock</code>	Set up Amazon Bedrock again.
<code>/setup-claude</code> also <code>/setup-cowork</code>	Walks you through setup: pick a role, install a matching plugin, try a skill, connect tools.
<code>/setup-vertex</code>	Set up Google Vertex AI again.
<code>/simplify</code> <code><target></code>	Tidy up and simplify changed code.
<code>/skill-doctor</code>	Show which loaded skills are costing context without being used.
<code>/skills</code>	List the available skills.
<code>/status</code>	Show the state: version, model, account, connections.
<code>/statusline</code>	Set up the status line.
<code>/stickers</code>	Order Claude Code stickers.
<code>/stop</code>	End this background session; log and worktree are kept.
<code>/subtask</code> <code><task></code>	Send off a subagent with full context; the result comes back.

<code>/tasks</code> also <code>/bashes</code>	View and manage everything running in the background.
<code>/team-onboarding</code>	Generate a guide for colleagues out of your own usage.
<code>/teleport</code> also <code>/tp</code>	Continue a session from claude.ai here.
<code>/terminal-setup</code>	Set up the terminal key bindings (line break).
<code>/theme</code>	Switch the colour scheme.
<code>/tui</code> <code>[default fullscreen]</code>	Choose the interface style: normal or full screen.
<code>/ultraplan</code> <code><prompt></code>	Have an editable plan drafted on the web.
<code>/ultrareview</code>	Multi-stage bug hunt across the branch, in the cloud.
<code>/upgrade</code>	Move to Max for higher limits and more Opus.
<code>/usage</code> also <code>/cost</code> , <code>/stats</code>	Show the session's cost, usage and activity.
<code>/usage-credits</code>	Set up credits or ask your administrator for them.
<code>/verify</code>	Checks that a change really does what it should — in a real run.
<code>/voice</code> <code>[hold tap off]</code>	Toggle voice mode.
<code>/web-setup</code>	Connect Claude Code on the web with your GitHub account.
<code>/whiteboard</code>	Opens a free drawing surface for sketches and diagrams — on your own or, where available, live together with others.
<code>/workflow-launch-exec</code>	Execute a workflow triggered by the server.
<code>/workflows</code>	Look through running and finished workflows.
<code>/workshop</code>	Designs something with you step by step, as a growing document.

Appendix · Present, but switched off

These names exist in the program, but are disabled by a flag in this version and are not available inside a session.

<code>/loops</code>	List, create and delete loops. present in the program, but switched off
<code>/pause-memory</code> also <code>/memory-pause</code> , <code>/toggle-memory</code>	Pause automatic memory for this session. present in the program, but switched off
<code>/update</code> also <code>/restart</code>	Move to the newest version. present in the program, but switched off
<code>/version</code>	Show the version of this session. present in the program, but switched off
<code>/wellbeing</code> also <code>/breaks</code> , <code>/break-reminder</code> , <code>/downtime</code>	Set up break reminders and quiet hours. present in the program, but switched off