

Claude Code Commands Cheat Sheet

Slash commands, CLI commands, flags, MCP patterns, and practical workflows - condensed from the full ScriptByAI reference.

Always check the latest edition

Claude Code changes quickly. Commands, defaults, flags, and availability can change after this PDF is published. Follow the main post to download the newest PDF whenever Claude Code updates.

PROJECT SETUP

/init · /status · /model

CONTEXT CONTROL

/context · /compact · /clear

CODE REVIEW

/review · /security-review · /diff

DEBUG & PLAN

/debug · /plan · /diff

EXTENSIONS

/mcp · /plugin · /skills · /agents

USAGE & HEALTH

/usage · /doctor · /release-notes

Public command

Built-in workflow / skill

MCP-generated

Community, internal, or removed: verify first



Download the latest PDF

Scan or open the main post for the complete A-Z index, longer notes, and future revisions.

scriptbyai.com/claude-code-commands-cheat-sheet/

Project setup and session control

Start with `/status` when a session behaves differently than expected. Use `/context` before compacting and `/clear` only when the next task no longer needs prior conversation context.

PROJECT AND ENVIRONMENT

Start and configure

<code>/init</code>	Create CLAUDE.md project instructions.
<code>/status</code>	Show the current model, environment, and session state.
<code>/model [model]</code>	Switch the active model.
<code>/config</code>	Open settings or set a supported key/value.
<code>/memory</code>	Open or edit Claude memory files.
<code>/permissions</code>	Manage permission mode and recent denials.
<code>/sandbox</code>	Open sandbox controls.
<code>/add-dir <path></code>	Add another directory to the working scope.
<code>/hooks</code>	Manage hook scripts and lifecycle events.
<code>/keybindings</code>	Customize keyboard shortcuts.
<code>/theme</code>	Change or create a terminal theme.
<code>/terminal-setup</code>	Configure terminal integration.
<code>/doctor</code>	Run installation and environment diagnostics.
<code>/help</code>	Show commands available in the installed build.

CONVERSATION STATE

Manage context and sessions

<code>/context</code>	Inspect context-window usage.
<code>/compact [focus]</code>	Compress the session while preserving key context.
<code>/clear</code>	Clear conversation context for a new task.
<code>/cd <path></code>	Move the session to another working directory.
<code>/rename [name]</code>	Name the current session.
<code>/resume [session]</code>	Reopen an earlier or background session.
<code>/branch [name]</code>	Fork the current conversation or workflow.
<code>/rewind</code>	Return to an earlier checkpoint.
<code>/copy [N]</code>	Copy the latest or selected response.
<code>/export [filename]</code>	Export the conversation.
<code>/goal <condition></code>	Keep working until a clear completion condition is met.
<code>/btw <question></code>	Ask a side question with minimal task context.
<code>/usage</code>	View plan limits, quota, cost, and statistics.

Review, extend, and work remotely

The commands below cover everyday development work, extension management, remote sessions, and usage checks. Availability can vary by Claude Code build and account configuration.

DEVELOPMENT WORKFLOW

Review and change code

<code>/plan [desc]</code>	Enter plan mode before a larger change.
<code>/debug [desc]</code>	Run a guided debugging workflow.
<code>/review</code>	Review current code changes.
<code>/code-review [effort]</code>	Review correctness issues; add --fix to apply findings.
<code>/security-review</code>	Run a security-focused review.
<code>/simplify</code>	Apply cleanup, reuse, efficiency, and structure fixes.
<code>/diff</code>	Open the interactive diff viewer.
<code>/batch</code>	Apply one change across many files or worktrees.
<code>/pr-comments [PR]</code>	Fetch GitHub pull-request comments.
<code>/ultrareview [PR#]</code>	Run a deeper cloud review.
<code>/install-github-app</code>	Set up GitHub App integration.
<code>/less-permission-prompts</code>	Propose a safer read-only allowlist.

EXTENSIONS

MCP, plugins, skills, agents

<code>/mcp</code>	Manage MCP servers and authentication.
<code>/mcp__[server]__[prompt]</code>	Run a prompt exposed by a connected MCP server.
<code>/plugin</code>	Open plugin management.
<code>/plugin list</code>	List installed plugins.
<code>/reload-plugins</code>	Hot-reload installed plugins.
<code>/skills</code>	List available skills.
<code>/reload-skills</code>	Re-scan skill directories.
<code>/agents</code>	Manage subagents.
<code>/workflows</code>	View dynamic workflow runs.
<code>/claude-api</code>	Load the Claude API or SDK helper workflow.

CONNECTED WORKFLOWS

Remote work and utilities

<code>/remote-control</code>	Start or manage Remote Control.
<code>/remote-env</code>	Configure remote environments.
<code>/teleport</code>	Transfer or bridge sessions.
<code>/desktop</code>	Hand off to the desktop app.
<code>/ide</code>	Open or manage IDE integration.
<code>/chrome</code>	Open Chrome integration when available.
<code>/schedule</code>	Manage cloud scheduled tasks.
<code>/voice</code>	Enable push-to-talk voice mode.
<code>/release-notes</code>	Review changes after an update.
<code>/usage-credits</code>	View usage-credit information.

CLI commands, flags, and environment variables

Shell commands start or automate Claude Code. Slash commands control an active session. Treat permission-bypass options as high-risk and restrict them to isolated, trusted environments.

RUN FROM YOUR SHELL

Core CLI commands

claude	Start interactive Claude Code.
claude "prompt"	Start interactively with an initial prompt.
claude -p "prompt"	Run one non-interactive prompt and exit.
claude -c	Continue the last conversation.
claude -r "name"	Resume a named session.
claude agents	Open the unified agent/session view.
claude agents --json	List active sessions as JSON.
claude --bg --exec '<command>'	Run a shell command as a background session.
claude update	Update Claude Code.
claude mcp list	List configured MCP servers.
claude mcp login <name>	Authenticate an MCP server.
claude mcp serve	Run Claude Code as an MCP server.
claude plugin init <name>	Scaffold a plugin in .claude/skills.

AUTOMATION AND CONTROL

Important flags

--model	Set the model.
--fallback-model	Use a fallback when the primary model is unavailable.
-w, --worktree	Run in a git worktree.
-n, --name	Name the session.
--add-dir	Add a directory to scope.
--agent	Select an agent.
--allowedTools	Pre-approve tools.
--output-format	Return machine-readable output such as JSON.
--json-schema	Request structured output.
--max-turns	Limit agentic turns.
--max-budget-usd	Cap spend for a run.
--effort	Set reasoning effort.
--permission-mode	Set permission behavior such as plan.
--remote	Start a remote or web-backed session.
--safe-mode	Disable customizations for troubleshooting.
--dangerously-skip-permissions	Skip prompts; use only in controlled environments.

SELECTED OPERATIONAL SETTINGS

Useful environment variables

CLAUDE_CODE_SAFE_MODE	Start with customizations disabled.
CLAUDE_CODE_ENABLE_GATEWAY_MODEL_DISCOVERY=1	Enable gateway model discovery in /model.
CLAUDE_CODE_DISABLE_ALTERNATE_SCREEN=1	Keep output in native terminal scrollbar.
CLAUDE_CODE_FORCE_SYNC_OUTPUT=1	Force synchronized terminal output.
CLAUDE_CODE_DISABLE_BUNDLED_SKILLS	Hide bundled skills and workflows from the model.
CLAUDE_CODE_MCP_TOOL_IDLE_TIMEOUT	Override idle timeout for remote MCP tool calls.
DISABLE_UPDATES	Block automatic and manual update paths.
OTEL_LOG_ASSISTANT_RESPONSES	Control assistant response text in telemetry logs.
OTEL_LOG_TOOL_DETAILS=1	Add tool parameters to telemetry events.

Practical workflow recipes

Run these sequences as starting points. Review each proposed tool action, diff, and permission request before applying changes to production code.

Start a new repository

```
/init  
/status  
/model
```

Understand an unfamiliar codebase

```
/status  
/context  
/plan map the project structure and main entry  
points
```

Review a pull request

```
/review  
/security-review  
/pr-comments  
/diff
```

Prepare a safe refactor

```
/review  
/plan refactor the auth flow without changing  
behavior  
/diff
```

Manage a long session

```
/context  
/compact keep decisions only  
/usage
```

Fix a bug in stages

```
/debug trace the failing checkout flow  
/review  
/diff
```

Apply a multi-file change

```
/review  
/code-review --fix  
/batch rename OldThing NewThing
```

Set up MCP and use it

```
/mcp  
/mcp_github__list_prs  
/mcp_jira__create_issue
```

Work across devices

```
/remote-control  
/remote-env  
/teleport
```

Run a headless task

```
claude -p "review this diff and list risks"
```

Run a background shell session

```
claude --bg --exec 'npm test'
```

Work toward a clear goal

```
/goal fix the failing checkout test and stop  
after tests pass  
/context  
/review
```

Public command index: A-L

Fast alphabetical lookup for public commands, built-in skills, public workflow commands, shortcuts, and MCP-generated command syntax found in the supplied article.

/add-dir <path>

Add another directory to scope.

/agents

Manage subagents.

/batch

Apply one change across many files.

/branch [name]

Branch or fork the conversation.

/btw <question>

Ask a side question with minimal context.

/cd <path>

Move the session to a new working directory.

/chrome

Open Chrome integration.

/claude-api

Load Claude API or SDK helper workflow.

/clear

Clear conversation context.

/code-review [effort]

Review correctness issues.

/color [color]

Change session accent color.

/compact [focus]

Compact context with optional focus.

/config

Open settings; supports /config key=value .

/context

Show context usage.

/copy [N]

Copy latest or selected response.

/cost

Open cost information inside /usage .

/debug [desc]

Run a debugging workflow.

/desktop

Hand off to desktop app.

/diff

Open diff viewer.

/doctor

Run diagnostics.

/effort

[low|medium|high|xhigh|max|auto]

Set reasoning effort.

/exit

Exit Claude Code.

/export [filename]

Export conversation.

/fast [on|off]

Toggle fast mode.

/feedback

Send feedback.

/focus

Toggle Focus view.

/goal

Set a completion condition.

/help

Show help and commands.

/hooks

Manage hook scripts and events.

/ide

Open or manage IDE integration.

/init

Create CLAUDE.md .

/install-github-app

Set up GitHub App integration.

/insights

Show usage or session insights.

/keybindings

Edit keyboard shortcuts.

/less-permission-prompts

Propose safe read-only allowlist entries.

/login

Sign in.

/logout

Sign out.

/loop [interval]

Run a recurring workflow.

Public command index: M-Z

Use `/help` to confirm which commands are available in your installed build. The main post carries the longer notes and update history.

/mcp

Manage MCP servers and authentication.

/mcp__[server]__[prompt] [args]

Dynamic MCP prompt command.

/memory

Open memory files.

/model [model]

Switch active model.

/output-style [style]

Change response style.

/permissions

Manage permission rules and recent denials.

/plan [desc]

Enter plan mode.

/plugin

Manage plugins.

/plugin list

List installed plugins.

/powerup

Open interactive lessons.

/pr-comments [PR]

Fetch PR review comments.

/rc

Start Remote Control.

/release-notes

View release notes.

/reload-plugins

Reload plugins.

/reload-skills

Re-scan skill directories.

/remote-control

Start or manage Remote Control.

/remote-env

Configure remote environments.

/rename [name]

Rename current session.

/resume [session]

Resume previous session.

/review

Review current code changes; PR reviews use the `/code-review` medium engine.

/rewind

Rewind to an earlier checkpoint, including before `/clear` .

/sandbox

Open sandbox controls and credential/network protections.

/schedule

Manage scheduled tasks.

/scroll-speed

Tune mouse wheel scroll speed.

/security-review

Run security-focused review.

/simplify

Run cleanup-only review and apply fixes.

/skills

List skills.

/stats

Open stats tab inside `/usage` .

/status

Show session status.

/stickers

Easter egg or promo command.

/team-onboarding

Generate teammate onboarding guide.

/teleport

Bridge or transfer sessions.

/terminal-setup

Configure terminal integration.

/theme

Change or create themes.

/tui

Switch terminal UI mode.

/ultrareview [PR#]

Comprehensive cloud code review.

/usage

Show limits, usage, and costs.

/usage-credits

Show usage credits.

/voice

Enable push-to-talk voice mode.

/workflows

View dynamic workflow runs.

Compatibility notes and custom commands

A command found in a blog post, repository, source leak, or older screenshot is not automatically available in a current Claude Code installation. Use `/help` and the release notes to verify support.

Common community commands

<code>/commit</code>	Community command: generate a commit message and commit changes.
<code>/pr</code>	Community command: create a pull request.
<code>/push</code>	Community command: push the current branch.
<code>/lint</code>	Community command: run linting.
<code>/fix-pipeline</code>	Community command: repair a failing CI pipeline.
<code>/merge-to-main</code>	Community command: merge to main.
<code>/vitest</code>	Community command: run Vitest workflows.

Removed or legacy names

<code>/vim</code>	Removed in v2.1.92; enable Vim mode through <code>/config</code> .
<code>/tag</code>	Legacy command removed in v2.1.92.

Internal and leak-based examples: `/advisor` · `/brief` · `/bughunter` · `/env` · `/files` · `/session` · `/share` · `/tasks` · `/upgrade` · `/version`. These names can disappear, remain inaccessible, or behave differently from public commands.

Project command

```
.claude/commands/name.md
```

Store a command with the repository so teammates can use the same workflow.

Personal command

```
~/.claude/commands/name.md
```

Make a custom command available across your local projects.

Arguments and controls

```
$ARGUMENTS
```

Capture text passed after the command name. Frontmatter can define the model, description, allowed tools, or disallowed tools.

This PDF will change as Claude Code changes

Keep the main post bookmarked. ScriptByAI will revise the downloadable cheat sheet when commands, flags, workflows, defaults, or compatibility notes change.

Get the latest PDF: scriptbyai.com/claude-code-commands-cheat-sheet/