

Claude Code

Befehlsreferenz

Alle Terminal-Befehle, alle Optionen und alle Slash-Befehle innerhalb einer Sitzung — mit einer Erklärung, was sie tun.

Zusammengestellt von Christians virtuellem Server · 12.09.2026 · Aktualisierung der Fassung vom 11. September 2026 (2.1.268)

Wie diese Liste entstanden ist

Terminal-Befehle und Optionen (Teil A und B) sind direkt aus der installierten Fassung 2.1.269 ausgelesen — `claude --help` und die Hilfe jedes Unterbefehls.

Die Slash-Befehle (Teil C) stammen aus den Befehlsdefinitionen im Programm selbst. Das Ausleseverfahren erkennt auch Befehle, deren Beschreibung erst zur Laufzeit gebildet oder aus einer Variablen geholt wird, und seit dieser Fassung ebenso solche, deren *Name* in einer Variablen steht statt als Text im Programm. Beides kommt durch das Verdichten des Programmcodes beim Bauen zustande und sagt nichts über den Befehl selbst aus.

Befehle, die im Programm angelegt, aber per Schalter abgeschaltet sind, stehen gesondert am Ende, damit nichts als verfügbar erscheint, was es nicht ist.

Was sich seit der letzten Fassung geändert hat

Gegenüber 2.1.268: ein neuer Slash-Befehl, `/output-style` (Ausgabestile auflisten oder zu einem wechseln) — damit stehen jetzt 139 Slash-Befehle in der Referenz. Kein Befehl ist weggefallen, und bei keinem bestehenden Befehl hat sich die Beschreibung geändert. Bei 37 Befehlen haben sich nur interne, nicht sichtbare Bezeichner geändert; das betrifft niemanden, der die Referenz liest.

`claude --help` ist zeichengleich geblieben. Bei der Hilfe der Unterbefehle gibt es eine inhaltliche Änderung: `claude plugin eval` beschreibt jetzt ausführlicher, was der Befehl tut und worauf zu achten ist — er lädt das Plugin und lässt seine Eval-Suite (Prompts, Grader; Gerüstskripte und echte MCP-Server nur nach Zustimmung) auf der eigenen Maschine laufen, weshalb nur Plugins ausgewertet werden sollten, denen man vertraut. Der erste Lauf in einem nicht vertrauten Plugin-Ordner fragt nach Bestätigung (`--trust-plugin` beantwortet das für CI).

Teil A · Terminal-Befehle und Optionen

Aufrufform: `claude [optionen] [befehl] [eingabe]`. Ohne Angaben startet eine Sitzung im aktuellen Ordner.

Sitzung starten

<code>claude</code>	Startet eine Sitzung im aktuellen Ordner.
<code>claude "deine Frage"</code>	Startet mit einer ersten Nachricht.
<code>claude -p "..."</code>	Antwortet einmal und beendet sich — für Skripte.

Optionen

<code>--add-dir <verzeichnisse...></code>	Weitere Ordner, auf die Claude zugreifen darf.
<code>--agent <agent></code>	Agent für diese Sitzung; überschreibt die Einstellung.
<code>--agents <json></code>	Eigene Agenten direkt als JSON mitgeben.

<code>--allow-dangerously-skip-permissions</code>	Erlaubt das spätere Abschalten aller Rückfragen, ohne es sofort zu tun.
<code>--allowedTools, --allowed-tools <werkzeuge...></code>	Nur diese Werkzeuge zulassen, z. B. "Bash(git *) Edit".
<code>--append-system-prompt <text></code>	Text an die Systemanweisung anhängen.
<code>--autocompact <auto tokens></code>	Ab welcher Füllung automatisch zusammengefasst wird (auto oder 100k–1M).
<code>--ax-screen-reader</code>	Ausgabe für Screenreader: flacher Text, keine Rahmen, keine Animationen.
<code>--bg, --background</code>	Sitzung im Hintergrund starten und sofort zurückkehren. Gibt die Kennung aus, die Claude attach, logs, stop und rm entgegennehmen; Claude Agents listet sie auf. Mit <code>--resume</code> läuft diese Sitzung unter derselben Kennung im Hintergrund weiter, oder es startet eine Kopie mit entsprechendem Hinweis, falls die Sitzung noch läuft.
<code>--bare</code>	Sparmodus: ohne Hooks, Plugins, Auto-Gedächtnis und CLAUDE.md-Suche.
<code>--betas <betas...></code>	Beta-Kennungen an die Schnittstelle mitsenden (nur mit API-Schlüssel).
<code>--brief</code>	Erlaubt Claude, dir während der Arbeit kurze Zwischenmeldungen zu schicken.
<code>--chrome</code>	Claude-in-Chrome-Anbindung einschalten.
<code>--cloud [beschreibung id url]</code>	Cloud-Sitzung anlegen oder an eine bestehende andocken.
<code>-c, --continue</code>	Letzte Unterhaltung in diesem Ordner fortsetzen.
<code>--dangerously-skip-permissions</code>	Alle Rückfragen abschalten. Nur für abgeschottete Umgebungen ohne Internet.
<code>-d, --debug [filter]</code>	Fehlersuche einschalten, wahlweise gefiltert (z. B. "api,hooks").
<code>--debug-file <pfad></code>	Fehlerprotokoll in diese Datei schreiben.
<code>--disable-slash-commands</code>	Alle Skills abschalten.
<code>--disallowedTools, --disallowed-tools <werkzeuge...></code>	Diese Werkzeuge sperren.
<code>--effort <stufe></code>	Denkaufwand für die Sitzung: low, medium, high, xhigh, max.
<code>--environment <id></code>	Cloud-Sitzung in einer selbst betriebenen Umgebung starten.
<code>--exclude-dynamic-system-prompt-sections</code>	Rechnerabhängige Angaben aus der Systemanweisung heraushalten (bessere Zwischenspeicherung).
<code>--fallback-model <modell></code>	Ersatzmodell, wenn das gewählte überlastet ist (auch außerhalb von <code>--print</code>).
<code>--file <angaben...></code>	Dateien beim Start herunterladen (file_id:zielpfad).
<code>--fork-session</code>	Beim Fortsetzen eine neue Sitzungs-ID vergeben statt die alte weiterzuführen.
<code>--forward-subagent-text</code>	Text und Denkschritte der Unteragenten mit durchreichen.
<code>--from-pr [wert]</code>	Sitzung zu einem Pull Request fortsetzen (Nummer oder URL).
<code>-h, --help</code>	Hilfe anzeigen.
<code>--ide</code>	Beim Start automatisch mit der Entwicklungsumgebung verbinden.
<code>--include-hook-events</code>	Hook-Ereignisse mit in den Ausgabestrom schreiben (nur mit <code>--print</code>).

<code>--include-partial-messages</code>	Teilstücke der Antwort ausgeben, sobald sie eintreffen (nur mit <code>--print</code>).
<code>--input-format <format></code>	Eingabeformat: text oder stream-json (nur mit <code>--print</code>).
<code>--json-schema <schema></code>	JSON-Schema, gegen das die Antwort geprüft wird.
<code>--max-budget-usd <betrag></code>	Obergrenze in Dollar für die Schnittstellenaufrufe (nur mit <code>--print</code>).
<code>--mcp-config <dateien...></code>	MCP-Server aus JSON-Dateien oder -Texten laden.
<code>--model <modell></code>	Modell für diese Sitzung.
<code>-n, --name <name></code>	Anzeigename der Sitzung.
<code>--no-chrome</code>	Chrome-Anbindung abschalten.
<code>--no-session-persistence</code>	Sitzung nicht auf die Festplatte schreiben.
<code>--output-format <format></code>	Ausgabeformat: text, json oder stream-json (nur mit <code>--print</code>).
<code>--permission-mode <modus></code>	Berechtigungsmodus: acceptEdits, plan, bypassPermissions, default.
<code>--permission-prompts <target></code>	Wer bei <code>--print</code> auf Berechtigungsabfragen antwortet: "host" (SDK-Host bzw. <code>--permission-prompt-tool</code>) oder "none" (automatische Ablehnung; der Berechtigungsmodus entscheidet weiterhin über alles andere). Standard: host.
<code>--plugin-dir <pfad></code>	Plugin aus einem Ordner oder Archiv laden, nur für diese Sitzung; ein Ordner mit mehreren Plugins lädt sie alle.
<code>--plugin-url <url></code>	Plugin-Archiv von einer Adresse laden, nur für diese Sitzung.
<code>-p, --print</code>	Antwort ausgeben und beenden — für Skripte und Pipes.
<code>--prompt-suggestions [wert]</code>	Eingabevorschläge einschalten.
<code>--remote-control [name]</code>	Sitzung mit Fernsteuerung starten (Handy, claude.ai).
<code>--remote-control-session-name-prefix <prefix></code>	Namensvorsatz für automatisch benannte Fernsteuer-Sitzungen.
<code>--replay-user-messages</code>	Eingelesene Nutzernachrichten zur Bestätigung zurückgeben.
<code>-r, --resume [wert]</code>	Unterhaltung per ID fortsetzen oder Auswahlliste öffnen.
<code>--safe-mode</code>	Ohne alle Anpassungen starten: keine CLAUDE.md, Skills, Plugins, Hooks.
<code>--session-id <uuid></code>	Feste Sitzungs-ID vorgeben.
<code>--setting-sources <quellen></code>	Welche Einstellungsquellen geladen werden: user, project, local.
<code>--settings <datei-oder-json></code>	Zusätzliche Einstellungen aus Datei oder JSON-Text.
<code>--strict-mcp-config</code>	Nur die per <code>--mcp-config</code> angegebenen MCP-Server verwenden.
<code>--system-prompt <text></code>	Systemanweisung für die Sitzung ersetzen.
<code>--system-prompt-snapshot <on off></code>	Systemanweisung einmal je Unterhaltung festhalten und danach bei jeder Anfrage sowie beim Fortsetzen unverändert wiederverwenden (empfohlen: on). Standardmäßig an für die eingebaute Anweisung; <code>--system-prompt</code> oder <code>--append-system-prompt</code> schaltet automatisch ab, damit der übergebene Text bei jedem Start frisch gilt.
<code>--teleport [sitzung]</code>	Teleport-Sitzung fortsetzen.
<code>--tmux</code>	tmux-Sitzung für den Arbeitsbaum anlegen (zusammen mit <code>--worktree</code>).
<code>--tools <werkzeuge...></code>	Auswahl aus den eingebauten Werkzeugen festlegen.
<code>--verbose</code>	Ausführliche Ausgabe, unabhängig von der Einstellung.

<code>-v, --version</code>	Versionsnummer ausgeben.
<code>-w, --worktree [name]</code>	Neuen Git-Arbeitsbaum für diese Sitzung anlegen.

Teil B · Unterbefehle

Aufrufe der Form `claude <unterbefehl>`. Eingerückt die jeweils zugehörigen Unter-Unterbefehle.

claude agents	Hintergrund-Agenten verwalten.
claude attach	Eine Hintergrundsitzung in diesem Terminal öffnen.
claude auth	Anmeldung verwalten.
<code>claude auth login</code>	Am Anthropic-Konto anmelden.
<code>claude auth logout</code>	Abmelden.
<code>claude auth status</code>	Anmeldestatus anzeigen.
claude auto-mode	Einstellungen des automatischen Modus ansehen oder zurücksetzen.
<code>claude auto-mode config</code>	Wirksame Konfiguration als JSON ausgeben.
<code>claude auto-mode critique</code>	Rückmeldung zu eigenen Regeln einholen.
<code>claude auto-mode defaults</code>	Mitgelieferte Standardregeln ausgeben.
<code>claude auto-mode reset</code>	Auf die Standardregeln zurücksetzen.
claude doctor	Installation prüfen. Für die große Untersuchung <code>/doctor</code> in einer Sitzung.
claude gateway	Firmen-Gateway für Anmeldung und Telemetrie betreiben.
claude import	Konfiguration eines anderen KI-Werkzeugs übernehmen.
claude install	Native Fassung installieren (stable, latest oder feste Version).
claude logs	Letzte Terminalausgabe einer Hintergrundsitzung anzeigen.
claude mcp	MCP-Server einrichten und verwalten.
<code>claude mcp add</code>	MCP-Server hinzufügen (stdio, http oder sse).
<code>claude mcp add-from-claude-desktop</code>	Server aus Claude Desktop übernehmen (nur Mac und WSL).
<code>claude mcp add-json</code>	Server aus einem JSON-Text anlegen.
<code>claude mcp get</code>	Angaben zu einem Server anzeigen.
<code>claude mcp list</code>	Eingerichtete Server auflisten.
<code>claude mcp login</code>	An einem Server anmelden.
<code>claude mcp logout</code>	Von einem Server abmelden.
<code>claude mcp remove</code>	Server entfernen.
<code>claude mcp reset-project-choices</code>	Projektbezogene Zustimmungen zurücksetzen.
<code>claude mcp serve</code>	Claude Code selbst als MCP-Server betreiben.

claude plugin	Plugins verwalten.
claude plugin details	Bestandteile eines Plugins und geschätzte Kontextkosten.
claude plugin disable	Plugin abschalten.
claude plugin enable	Plugin einschalten.
claude plugin eval	Testfälle gegen ein Plugin laufen lassen und bewerten.
claude plugin init new	Gerüst für ein neues Plugin anlegen.
claude plugin install i	Plugin aus einem Marktplatz installieren.
claude plugin list	Installierte Plugins auflisten.
claude plugin marketplace	Marktplätze verwalten.
claude plugin prune autoremove	Nicht mehr benötigte Plugins entfernen.
claude plugin tag	Plugins mit Merkmalen versehen.
claude plugin uninstall remove	Plugin deinstallieren.
claude plugin update	Plugin aktualisieren.
claude plugin validate	Plugin auf Fehler prüfen.
claude project	Projektzustand verwalten.
claude project purge	Allen gespeicherten Zustand eines Projekts löschen.
claude respawn	Hintergrundsitzung(en) neu starten, damit sie die aktuelle Fassung nutzen.
claude rm	Hintergrundsitzung löschen, auch ihren Arbeitsbaum, wenn das gefahrlos möglich ist.
claude setup-token	Langlebiges Anmeldetoken erzeugen (für Skripte und CI).
claude stop kill	Hintergrundsitzung anhalten; die Unterhaltung bleibt erhalten.
claude ultrareview	Mehrstufige Code-Durchsicht in der Cloud starten.

Teil C · Slash-Befehle in der Sitzung

134 verfügbare Befehle, alphabetisch. Grau dahinter: Zweitnamen und erwartete Argumente.

/add-dir <path>	Weiteren Arbeitsordner hinzufügen.
/advisor	Claude an wichtigen Stellen ein stärkeres Modell befragen lassen.
/agents	Entfernt. Unteragenten liegen jetzt in .claude/agents/.
/artifact-capabilities	Was veröffentlichte Seiten zur Laufzeit können.
/artifact-components	Baukasten wiederverwendbarer Bausteine für veröffentlichte Seiten.
/artifact-design	Gestaltungsregeln für veröffentlichte Seiten.
/artifact-diagramming	Diagramme für veröffentlichte Seiten.
/artifact-pr-review [pr number or url]	Fasst die Durchsicht eines Pull Requests als teilbare Seite zusammen: Kernaussage, Empfehlung, Abwägungen, Schaubild, blinde Flecken.

<code>/artifacts</code>	Veröffentlichte und geteilte Artefakte durchsehen.
<code>/auto-mode-setup</code> [--request-id <uuid>] (--wizard posture=\u2026 scope=\u2026 depth=\u2026 --propose --expect-sha256 <64-hex> --apply-file <path>)	Automatischen Modus einrichten und anpassen.
<code>/autocompact</code> [auto <tokens>]	Festlegen, ab welcher Füllung automatisch zusammengefasst wird.
<code>/autofix-pr</code>	Den aktuellen Pull Request überwachen und Probleme beheben.
<code>/background</code> auch /bg · [prompt]	Sitzung in den Hintergrund schicken und das Terminal freigeben.
<code>/batch</code> <instruction>	Große Änderung planen und dann parallel ausführen lassen.
<code>/branch</code> [name]	Die Unterhaltung an dieser Stelle verzweigen.
<code>/brief</code>	Knapp-Modus umschalten.
<code>/btw</code> [question]	Zwischenfrage stellen, ohne den laufenden Faden zu stören.
<code>/bug</code> auch /share · [report]	Fehler melden oder die Unterhaltung teilen.
<code>/chrome</code>	Einstellungen für Claude in Chrome öffnen.
<code>/claude-api</code>	Nachschlagewerk zur Claude-Schnittstelle (Modelle, Preise, Werkzeuge).
<code>/claude-code-docs</code> [question]	Fragen zu Claude Code selbst beantworten (Befehle, Optionen, Einstellungen, Hooks, Skills), geprüft gegen die installierte Fassung.
<code>/claude-in-chrome</code>	Browser fernsteuern: klicken, tippen, Seiten auslesen.
<code>/clear</code> auch /reset, /new · [name]	Neue Sitzung mit leerem Kontext; die alte bleibt erhalten.
<code>/cloud-plugins</code>	Festlegen, ob Cloud-Sitzungen die hier eingeschalteten Plugins mitbenutzen.
<code>/code-review</code> auch /review · [...] [--fix] [--comment] [<pr#> <branch> <path>]	Änderungen auf Fehler und Vereinfachungen durchsehen.
<code>/color</code>	Farbe der Eingabezeile für diese Sitzung.
<code>/commit</code> [guidance]	Legt einen Git-Commit an, mit passend geschriebener Nachricht.
<code>/commit-push-pr</code>	Committen, hochladen und einen Pull Request eröffnen – in einem Zug.
<code>/compact</code> <optional custom summarization instructions>	Kontext freimachen, indem das Bisherige zusammengefasst wird.
<code>/config</code> auch /settings · [key=value]	Einstellungen öffnen.
<code>/context</code> [all]	Kontextverbrauch als farbiges Raster anzeigen.
<code>/copy</code>	Letzte Antwort in die Zwischenablage kopieren (/copy N für ältere).
<code>/cowork-plugin</code>	Legt ein neues Cowork-Plugin an oder passt ein vorhandenes an.
<code>/daemon</code>	Hintergrunddienste und Routinen verwalten.
<code>/dataviz</code>	Diagramme und Auswertungsseiten sauber gestalten.
<code>/debug</code> [issue description]	Fehlersuche für diese Sitzung einschalten.
<code>/design</code> [sync login consent revoke import export status <prompt>]	Zugriff auf Design-Projekte erteilen oder entziehen.
<code>/design-consent</code>	Zugriff auf Design-Projekte erteilen.
<code>/design-login</code>	Design-Zugriff für /design-sync freischalten.
<code>/design-revoke</code>	Zugriff auf Design-Projekte entziehen.

<code>/design-sync</code>	Ein React-Designsystem nach claude.ai/design übertragen.
<code>/desktop</code> auch <code>/app</code>	Die laufende Sitzung in der Desktop-App weiterführen.
<code>/diff</code>	Nicht eingeecheckte Änderungen und Änderungen je Zug ansehen.
<code>/doctor</code> auch <code>/checkup</code>	Installation durchleuchten und Probleme beheben.
<code>/effort</code>	Denkaufwand einstellen.
<code>/exit</code> auch <code>/quit</code>	Sitzung beenden.
<code>/explain-usage</code>	Den eigenen Verbrauch erklären lassen.
<code>/export</code> [filename]	Unterhaltung in eine Datei oder die Zwischenablage ausgeben.
<code>/extra-usage</code>	Umbenannt in <code>/usage-credits</code> .
<code>/fast</code> [on off]	Schnellmodus umschalten (gleiches Modell, höheres Tempo).
<code>/feedback</code> [report]	Rückmeldung an Anthropic senden.
<code>/fewer-permission-prompts</code>	Häufige harmlose Aufrufe erkennen und dauerhaft erlauben.
<code>/focus</code>	Nur Eingabe, Zusammenfassung und Antwort anzeigen.
<code>/fork</code> <directive>	Hintergrund-Agent starten, der die ganze Unterhaltung erbt.
<code>/goal</code> [<condition> clear]	Ein Ziel setzen, das Claude vor dem Aufhören prüft.
<code>/heapdump</code>	Speicherabbild schreiben (zur Fehlersuche).
<code>/help</code>	Hilfe und verfügbare Befehle anzeigen.
<code>/hooks</code>	Eingerichtete Hooks für Werkzeug-Ereignisse ansehen.
<code>/ide</code> [open]	Anbindung an die Entwicklungsumgebung verwalten.
<code>/import</code> [codex gemini cursor] [--dry-run]	Konfiguration eines anderen KI-Werkzeugs übernehmen.
<code>/init</code>	Eine erste CLAUDE.md aus dem vorhandenen Code erzeugen.
<code>/insights</code>	Auswertung der eigenen Sitzungen erstellen.
<code>/install</code> [options]	Native Fassung installieren.
<code>/install-github-app</code>	Claude-GitHub-Aktionen für ein Repository einrichten.
<code>/install-slack-app</code>	Claude-App für Slack installieren.
<code>/keybindings</code>	Datei mit den Tastenkürzeln öffnen.
<code>/limit-reset</code>	Das Sitzungslimit sofort zurücksetzen und weiterarbeiten; einmal pro Woche möglich, zählt weiter aufs Wochenlimit.
<code>/list-agents</code> auch <code>/peers</code>	Unteragenten und andere erreichbare Sitzungen auflisten.
<code>/login</code>	Am Anthropic-Konto anmelden oder Konto wechseln.
<code>/logout</code>	Vom Anthropic-Konto abmelden.
<code>/loop</code> auch <code>/proactive</code>	Einen Auftrag in festem Abstand wiederholt laufen lassen.
<code>/low-priority</code>	Nach Erreichen des Sitzungslimits mit niedrigerer Priorität fortsetzen; erneuter Aufruf stoppt das wieder.
<code>/mcp</code> [reconnect <server> enable disable [<server> all]]	MCP-Server verwalten.
<code>/memory</code>	CLAUDE.md-Dateien und Gedächtnis-Einstellungen bearbeiten.
<code>/memory-types</code>	Nachschlagewerk zu den Gedächtnisarten: was wohin gehört und wann.

<code>/mobile</code> auch <code>/ios</code> , <code>/android</code>	QR-Code für die Handy-App anzeigen.
<code>/model</code> <code>[model]</code>	Modell für Claude Code einstellen.
<code>/output-style</code> <code>[style]</code>	Ausgabestile auflisten oder zu einem wechseln.
<code>/passes</code>	Eine Gratiswoche Claude Code verschenken.
<code>/permissions</code> auch <code>/allowed-tools</code>	Regeln für erlaubte und verbotene Werkzeuge verwalten.
<code>/plan</code> <code>[open <description>]</code>	Plan-Modus einschalten oder den Plan der Sitzung ansehen.
<code>/plugin</code> auch <code>/plugins</code> , <code>/marketplace</code>	Plugins verwalten.
<code>/plugin-authoring</code>	Hilft beim Schreiben und Prüfen eines Plugins aus Funktions-Hooks.
<code>/plugin-types</code> <code>[dir]</code>	Zwei Typdateien schreiben: die TypeScript-Deklarationen der Plugin-Schnittstelle und die Eingaben der verbundenen MCP-Werkzeuge, zum Typisieren eines Hook-Moduls gegen diese Sitzung.
<code>/powerup</code>	Funktionen von Claude Code in kurzen Lektionen kennenlernen.
<code>/privacy-settings</code>	Datenschutz-Einstellungen ansehen und ändern.
<code>/pro-trial-expired</code>	Auswahl nach Ablauf der Pro-Testphase.
<code>/prototype</code>	Macht aus einer Idee einen lauffähigen Entwurf und veröffentlicht ihn als Seite.
<code>/radio</code>	Lo-Fi-Radio einschalten.
<code>/rate-limit-options</code>	Möglichkeiten anzeigen, wenn das Nutzungslimit erreicht ist.
<code>/recap</code>	Einzeilige Zusammenfassung der Sitzung erzeugen.
<code>/release-notes</code>	Neuerungen der Versionen anzeigen.
<code>/reload-plugins</code> <code>[-force]</code>	Ausstehende Plugin-Änderungen übernehmen.
<code>/reload-skills</code>	Neu hinzugekommene oder geänderte Skills einlesen.
<code>/remote-control</code> auch <code>/rc</code>	Sitzung vom Handy oder von <code>claude.ai</code> aus steuern.
<code>/remote-env</code>	Standardumgebung für Cloud-Agenten wählen.
<code>/rename</code> auch <code>/name</code> · <code>[name]</code>	Die Unterhaltung umbenennen.
<code>/resume</code> auch <code>/continue</code> · <code>[conversation id or search term]</code>	Frühere Unterhaltung fortsetzen.
<code>/rewind</code> auch <code>/checkpoint</code> , <code>/undo</code>	Code und/oder Unterhaltung auf einen früheren Stand zurücksetzen.
<code>/run</code>	Die Anwendung starten und eine Änderung im Betrieb prüfen.
<code>/run-skill-generator</code>	Aus einem Ablauf einen neuen Skill erzeugen lassen.
<code>/sandbox</code>	Abschottung einrichten und anzeigen.
<code>/schedule</code> auch <code>/routines</code>	Geplante Cloud-Agenten anlegen und verwalten.
<code>/scroll-speed</code>	Rollgeschwindigkeit des Mausekkrads einstellen.
<code>/session</code> auch <code>/remote</code>	Adresse und QR-Code der Cloud-Sitzung anzeigen.
<code>/setup-bedrock</code>	Amazon Bedrock neu einrichten.
<code>/setup-claude</code> auch <code>/setup-cowork</code>	Führt durch die Einrichtung: Rolle wählen, passendes Plugin installieren, Skill ausprobieren, Werkzeuge verbinden.
<code>/setup-vertex</code>	Google Vertex AI neu einrichten.
<code>/simplify</code> <code>[<target>]</code>	Geänderten Code aufräumen und vereinfachen.

<code>/skill-doctor</code>	Zeigen, welche geladenen Skills ungenutzt Kontext kosten.
<code>/skills</code>	Verfügbare Skills auflisten.
<code>/status</code>	Zustand anzeigen: Version, Modell, Konto, Anbindung.
<code>/statusline</code>	Statuszeile einrichten.
<code>/stickers</code>	Claude-Code-Aufkleber bestellen.
<code>/stop</code>	Diese Hintergrundsituation beenden; Protokoll und Arbeitsbaum bleiben.
<code>/subtask <task></code>	Unteragent mit vollem Kontext losschicken; Ergebnis kommt zurück.
<code>/tasks</code> auch <code>/bashes</code>	Alles ansehen und verwalten, was im Hintergrund läuft.
<code>/team-onboarding</code>	Anleitung für Kollegen aus der eigenen Nutzung erzeugen.
<code>/teleport</code> auch <code>/tp</code>	Eine Sitzung von claude.ai hier fortsetzen.
<code>/terminal-setup</code>	Tastenbelegung des Terminals einrichten (Zeilenumbruch).
<code>/theme</code>	Farbschema wechseln.
<code>/tui</code> <code>[default fullscreen]</code>	Oberflächenart wählen: normal oder bildschirmfüllend.
<code>/ultraplan <prompt></code>	Einen bearbeitbaren Plan im Web entwerfen lassen.
<code>/ultrareview</code>	Mehrstufige Fehlersuche im Zweig, in der Cloud.
<code>/upgrade</code>	Auf Max wechseln für höhere Grenzen und mehr Opus.
<code>/usage</code> auch <code>/cost</code> , <code>/stats</code>	Kosten der Sitzung, Verbrauch und Aktivität anzeigen.
<code>/usage-credits</code>	Guthaben einrichten oder beim Administrator anfragen.
<code>/verify</code>	Prüft nach, ob eine Änderung wirklich tut, was sie soll – im echten Durchlauf.
<code>/voice</code> <code>[hold tap off]</code>	Sprachmodus umschalten.
<code>/web-setup</code>	Claude Code im Web mit dem GitHub-Konto verbinden.
<code>/whiteboard</code>	Öffnet eine freie Zeichenfläche für Skizzen und Schaubilder — allein oder, wo verfügbar, live gemeinsam mit anderen.
<code>/workflow-launch-exec</code>	Vom Server angestoßenen Arbeitsablauf ausführen.
<code>/workflows</code>	Laufende und abgeschlossene Arbeitsabläufe durchsehen.
<code>/workshop</code>	Entwirft etwas Schritt für Schritt gemeinsam mit Dir, als wachsendes Dokument.

Anhang · Angelegt, aber abgeschaltet

Diese Namen existieren im Programm, sind in dieser Fassung aber per Schalter deaktiviert und stehen in einer Sitzung nicht zur Verfügung.

<code>/loops</code>	Wiederholungen auflisten, anlegen und löschen. im Programm angelegt, aber abgeschaltet
<code>/pause-memory</code> auch <code>/memory-pause</code> , <code>/toggle-memory</code>	Automatisches Gedächtnis für diese Sitzung anhalten. im Programm angelegt, aber abgeschaltet
<code>/update</code> auch <code>/restart</code>	Auf die neueste Fassung wechseln. im Programm angelegt, aber abgeschaltet
<code>/version</code>	Version dieser Sitzung anzeigen. im Programm angelegt, aber abgeschaltet

`/wellbeing` auch `/breaks`, `/break-reminder`, `/downtime` Pausenerinnerungen und Ruhezeiten einstellen. **im Programm angelegt, aber abgeschaltet**

Claude Code 2.1.269 · 64 Optionen · 134 verfügbare Slash-Befehle · 5 abgeschaltet · Stand 12.09.2026