

# Claude Code:

## *Slash-Befehle & Shortcuts*

Die Befehle und Tastenkürzel, die im Terminal-Alltag den Unterschied machen. Stand: Claude Opus 4.8 (Mai 2026).

**tbai**  
Stand Mai 2026

### Slash-Befehle

<b>/help</b>	Hilfe und Befehls-Übersicht innerhalb der Session.
<b>/clear</b>	Konversation komplett zurücksetzen — neuer Kontext.
<b>/compact</b>	Bisherigen Kontext zusammenfassen, Platz für Folgearbeit schaffen.
<b>/init</b>	CLAUDE.md im aktuellen Projekt initialisieren (Bootstrap).
<b>/model</b>	Modell wechseln (Haiku 4.5, Sonnet 4.6, Opus 4.8).
<b>/fast</b>	Fast Mode an/aus — 2,5× Throughput zum Premium-Preis (Opus 4.6+).
<b>/cost</b>	Aktuelle Token-/Kosten-Übersicht der Session.
<b>/agents</b>	Verfügbare Subagents listen, anlegen, wechseln.
<b>/plugin</b>	Plugin-Marketplace öffnen (Discover · Installed).
<b>/permissions</b>	Tool-Berechtigungen prüfen und anpassen.
<b>/diff</b>	Aktuellen Git-Diff der Session anzeigen.

### Tastenkürzel

<b>Tab</b>	Auto-Vervollständigung für Dateipfade und Befehle.
<b>Ctrl + K</b>	Such-Dialog innerhalb der Session.
<b>Esc</b>	Laufende Aktion oder Tool-Aufruf abbrechen.
<b>! Prefix</b>	Inline-Shell-Befehl: <code>! npm test</code> führt direkt aus.
<b>@ Prefix</b>	Datei-Referenz: <code>@src/auth.ts</code> hängt Datei an den Prompt.
<b>↑ / ↓</b>	Durch Prompt-History blättern.

### Headless & CI

#### Mini-Beispiel

#### PR-Auto-Review

```
claude -p "Review changes
since main. Max 5 findings." \
  --allowedTools "Read,Grep,
  Bash(git diff:*,git log:*)" \
  --bare \
  --output-format json
```

#### Faustregel

#### Sichere Whitelist

In CI niemals `--allowedTools "*" .` Bash-Befehle per Glob-Pattern einschränken: `Bash(npm test:*)` erlaubt nur Test-Varianten, nicht Deploys.