

Claude Code Spickzettel

Deine unverzichtbare Kurzreferenz für Claude Code

</> CLAUDECODE-LAB

1. Grundlegende Befehle

<code>claude</code>	Interaktiven Modus starten
<code>claude -p "..."</code>	Einmalige Ausführung (nicht-interaktiv)
<code>/help</code>	Hilfe anzeigen
<code>/clear</code>	Kontext löschen
<code>/compact</code>	Kontext komprimieren und fortfahren
<code>/cost</code>	Sitzungskosten prüfen
<code>/exit</code>	Claude Code beenden

2. Tastaturkurzel

<code>Ctrl + C</code>	Aktuelle Operation abbrechen
<code>Esc</code>	Eingabe abbrechen
<code>Esc</code> ×2	Zur vorherigen Nachricht zurückkehren
<code>Shift + Tab</code>	Nächster Vorschlag/Modus wechseln
<code>↑</code> / <code>↓</code>	Eingabeverlauf durchsuchen

3. CLAUDE.md Schreiben

- ▶ **Projektüberblick** — Was du baust in 1-2 Zeilen
- ▶ **Tech-Stack** — Sprache, Framework, DB, wichtige Bibliotheken
- ▶ **Coding-Standards** — Benennung, Formatter, Lint-Konfiguration
- ▶ **Häufige Befehle** — `build` / `test` / `dev` / `deploy`
- ▶ **Sperrgebiete** — Dateien/Verzeichnisse, die nicht verändert werden sollen

4. Effektive Prompts

- ▶ **Genauere Dateinamen angeben** — "Repariere `@src/api/user.ts`"
- ▶ **Erwartetes Ausgabeformat angeben** — JSON / Markdown / diff usw.
- ▶ **Vorher/Nachher-Beispiele zeigen** — E/A-Beispiele erhöhen die Genauigkeit
- ▶ **In Schritte aufteilen** — Entwerfen → Implementieren → Testen → Refaktorisieren
- ▶ **Verifizierung beschreiben** — "npm test muss bestehen"

5. Nützliche Pipe-Integrationen

```
git diff | claude -p "Prüfe diesen Diff"
Sofortiges Code-Review
```

```
cat error.log | claude -p "Was ist die Ursache?"
Ursachenanalyse aus Logs
```

```
cat package.json | claude -p "Raume die Deps auf"
package.json Abhängigkeiten prüfen
```

```
git log --oneline -20 | claude -p "Schreibe Release Notes"
CHANGELOG aus Commits generieren
```

Profi-Tipps

- ▶ **@dateiname** — Dateien direkt zum Kontext hinzufügen
- ▶ **/compact regelmäßig ausführen** — Genauigkeit in langen Sitzungen beibehalten
- ▶ **Planung und Umsetzung trennen** — Erst planen, dann handeln