

Claude Code Subagents - Cheat Sheet

Dein KI-Expertenteam - Komplette Anleitung



Was sind Claude Code Subagents?

Stell dir vor, du hättest nicht nur einen KI-Assistenten, sondern ein **ganzes Expertenteam**, das jeden Aspekt deiner Arbeit abdeckt! Genau das sind Claude Code Subagents - vorkonfigurierte KI-Persönlichkeiten mit spezifischen Fachgebieten und Zwecken.



Revolutionäre Vorteile:

- **Spezialisierte Experten:** Jeder Agent ist auf ein spezifisches Fachgebiet fokussiert
 - **Getrennte Kontextfenster:** Hauptkonversation bleibt sauber und fokussiert
 - **Längere Sitzungen:** Effizienter durch spezialisierte Agenten
 - **Automatische Delegation:** Claude wählt automatisch den richtigen Experten
 - **Flexible Tool-Berechtigungen:** Jeder Agent bekommt nur die Tools, die er braucht
-



Wie funktionieren Subagents?

Automatische Expertendelegation:

Performance-Problem? → Performance-Analyst springt ein

Bug entdeckt? → Debugging-Spezialist übernimmt

Code-Review nötig? → Code-Reviewer wird aktiviert

Dokumentation fehlt? → Dokumentations-Agent erstellt sie

Intelligente Arbeitsteilung:

1. **Du stellst deine Anfrage** in Claude Code
 2. **Claude analysiert automatisch** den Bedarf
 3. **Der passende Agent wird delegiert** basierend auf seiner Beschreibung
 4. **Spezialist arbeitet** in seinem eigenen Kontextfenster
 5. **Du bekommst Expertenantworten** ohne Aufwand
-



Agent-Struktur verstehen

Jeder Subagent besteht aus einer **Markdown-Datei mit YAML-Frontmatter** - das klingt technisch, ist aber sehr praktisch:

Beispiel-Agent-Struktur:

```
---
name: react-ui-developer
description: Use this agent when you need to create, modify, or debug React
user interface components. Examples include: building new React components,
fixing rendering issues, implementing responsive designs, or
troubleshooting component lifecycle issues.
tools: Task, Read, Edit, Write, WebSearch
color: cyan
---
```

You are a React Frontend Development Expert, specializing in creating robust, performant, and maintainable user interface components.

Your primary responsibilities include:

- Design and implement React components following best practices
- Debug rendering issues and performance bottlenecks
- Ensure components are accessible and responsive
- Provide clean, readable, and well-documented code

Wichtige Bestandteile:

Name: Eindeutige Bezeichnung des Agenten

Description: Definiert WANN Claude diesen Agenten einsetzt (präzise und aktionsorientiert formulieren)

Tools: Spezifische Werkzeuge für diesen Agenten (optional - ohne Angabe erbt er alle Tools)

Color: Visuelle Unterscheidung bei aktiven Agenten

System-Prompt: Das "Gehirn" des Agenten - definiert WIE er arbeitet



Speicherung & Organisation

Zwei Speicherorte verfügbar:

Projektspezifische Agenten (Höchste Priorität):

`.claude/agents/`

- Nur im aktuellen Projekt verfügbar • Perfekt für projektspezifische Experten • Überschreiben systemweite Agenten bei Namenskonflikt

Systemweite Agenten (Globale Verfügbarkeit):

`~/.claude/agents/`

- In allen Projekten verfügbar • Ideal für allgemeine Experten • Niedrigere Priorität bei Namenskonflikten

Portabilität & Team-Sharing:

• **GitHub-Integration:** Agenten-Dateien ins Repository hochladen • **Team-Konsistenz:** Alle nutzen dieselben Workflows • **Versionskontrolle:** Agenten gemeinsam verbessern und weiterentwickeln

Zugriff & Verwaltung

Einfacher Zugang über Kommando:

/agents

Öffnet interaktives Kontrollzentrum mit: • Alle verfügbaren Subagenten anzeigen • Neue Agenten erstellen • Bestehende Agenten bearbeiten
• Agenten löschen • Tool-Übersicht einsehen

Empfohlener Workflow:

1. **Mit Claude-generierten Agenten beginnen** (beste Ergebnisse)
 2. **Beschreibung des gewünschten Agenten** eingeben
 3. **Automatische Grundlage** wird erstellt
 4. **Nach Bedarf anpassen** und verfeinern
 5. **Über Zeit optimieren** basierend auf Verhalten
-

Tool-System verstehen

Flexible Berechtigungen:

Nur Lesezugriff? → Perfekt für Analyse-Agenten

Vollzugriff? → Ideal für Entwicklungs-Agenten

Spezifische Tools? → Maßgeschneiderte Experten

Tool-Vererbung:

Tools-Feld weglassen → Agent erbt ALLE Tools vom Haupt-Thread

Tools spezifizieren → Präzise Kontrolle über Berechtigungen

Weniger ist mehr → Zu viele Tools können Performance beeinträchtigen

MCP-Integration:

Externe Dienste verfügbar: • Gmail-Zugriff für E-Mail-Agenten • Linear-Integration für Projektmanagement

• Canva-Tools für Design-Agenten • Und viele weitere Services

Visuelle Organisation

Farb-Codierung für Übersicht:

Jeder Agent kann eine eigene **Farbe** bekommen: • **Cyan** für Frontend-Entwicklung • **Green** für Backend-Services

• **Red** für Debugging & Problemlösung • **Blue** für Dokumentation & Analyse

Vorteil: Bei mehreren aktiven Agenten siehst du sofort, wer gerade arbeitet!

Erweiterte Anwendungsfälle

Agent-Verkettung (Zukunftspotenzial):

Automatische Workflows:

1. **Performance-Analyst** findet Probleme
2. **Code-Optimierer** behebt sie
3. **Dokumentations-Agent** erstellt Release Notes
4. **Alles automatisch, nacheinander, perfekt koordiniert**

Team-Effizienz:

Konsistente Arbeitsweise durch geteilte Agenten
Spezialisierte Workflows für wiederkehrende Aufgaben
Wissensbewahrung in Agent-Konfigurationen

Warum Subagents revolutionär sind

Löst echte Probleme:

- ✗ **Vorher:** Kontext-Vermischung in langen Gesprächen
- ✗ **Vorher:** Generische Antworten für spezifische Probleme
- ✗ **Vorher:** Manuelle Koordination verschiedener Aufgaben

- ✓ **Jetzt:** Spezialisierte Experten mit fokussierten Kontexten
- ✓ **Jetzt:** Automatische Delegation an den richtigen Spezialisten
- ✓ **Jetzt:** Längere, effizientere Arbeitssitzungen

Formalisiert bewährte Praktiken:

Ersetzt manuelle Workarounds wie parallele Task-Spawns

Bietet echtes System statt improvisierter Lösungen

Skaliert professionell für Teams und komplexe Projekte

Perfekt für:



Entwickler: Spezialisierte Agenten für Frontend, Backend, Testing, Debugging



Projektmanager: Agenten für Planung, Dokumentation, Kommunikation



Designer: UI/UX-Experten, Design-System-Spezialisten



Content-Creator: Schreibagenten, SEO-Experten, Social Media-Spezialisten



DevOps: Infrastructure-Experten, Monitoring-Spezialisten



Schnellstart-Empfehlung

So startest du optimal:

1. **/agents eingeben** in Claude Code
2. **"Erstelle mir einen [Bereich]-Experten"** (z.B. Python-Entwickler)
3. **Beschreibung verfeinern** nach ersten Erfahrungen
4. **Weitere Agenten hinzufügen** je nach Bedarf
5. **Team-Agenten teilen** über GitHub/Versionskontrolle

Tipp: Beginne mit 2-3 spezialisierten Agenten und erweitere schrittweise!

Die Zukunft ist jetzt

Mit Claude Code Subagents hast du nicht mehr nur einen KI-Assistenten - **du hast ein ganzes Expertenteam!** Jeder Agent ist spezialisiert, fokussiert und arbeitet genau dann, wenn du ihn brauchst.

Deine Produktivität wird explodieren, weil:

- Jeder Agent ist ein Spezialist in seinem Bereich
- Automatische Delegation spart Zeit und Denkarbeit
- Getrennte Kontexte halten Gespräche fokussiert
- Flexible Tool-Berechtigungen geben präzise Kontrolle

Das ist nicht nur ein Feature - das ist eine Revolution in der Art, wie wir mit KI arbeiten!

Willkommen in der Zukunft der KI-gestützten Arbeit! 🤖✨