

CLAUDE SKILLS

Der komplette Guide

Was Skills sind, wie du sie installierst und

10 Skills die deinen Alltag sofort verändern

IN DIESEM GUIDE:

- Was ein Skill ist — einfach erklärt, kein Tech-Wissen nötig
- Schritt-für-Schritt Installation (alle Pläne, Desktop & Mobile)
- 10 fertige Skills mit konkreten Alltagsbeispielen
- Empfehlungen für den richtigen Einstieg

[Claude.ai](#)[Alle Pläne ✓](#)[Desktop & Mobile](#)[Kein Code nötig](#)

■ VIDEO: Skill in Claude einfügen — so geht's live

Ich zeige dir in meinem Reel Schritt für Schritt wie es funktioniert:

[instagram.com/reel/DZN4SKzRwUP](https://www.instagram.com/reel/DZN4SKzRwUP)

10

FERTIGE SKILLS

direkt einsatzbereit

#01–03

Produktivität
& Automation

#04–06

Design
& Kreativität

#07–10

Code
& Entwicklung

WAS IST EIN SKILL? — GANZ EINFACH ERKLÄRT

Skills und die Memory-Funktion von Claude lösen beide ein ähnliches Problem — aber auf völlig unterschiedlichen Wegen. Memory merkt sich Fakten über dich: "Marlon arbeitet im Marketing, er mag kurze Texte". Das ist nützlich.

Was Memory nicht kann: Claude beibringen WIE er eine komplexe Aufgabe genau ausführen soll — welche Schritte er geht, welches Format er nutzt, welche Standards er einhält. Genau das tun Skills.

MEMORY (eingebaut)

- "Marlon bevorzugt kurze Texte"
- "Er arbeitet im Marketing"
- Fakten & Präferenzen
- Keine Schritt-für-Schritt-Logik

SKILLS (du installierst sie)

- "Erstelle Reports immer in diesem Format"
- "Folge diesem 4-Phasen-Debug-Prozess"
- Workflows & Qualitätsstandards
- Exakte, wiederholbare Abläufe

Die Analogie: Dein neuer Mitarbeiter

Stell dir vor, du hast einen brillanten neuen Mitarbeiter. Am ersten Tag erklärst du ihm alles: wie Texte bei euch klingen sollen, welche Formatierung ihr nutzt, was euren Kunden wichtig ist. Ein guter Mitarbeiter merkt sich das — für immer. Ein Skill ist genau das: die Einarbeitung, die Claude einmal lernt und nie wieder vergisst.

3 DINGE, DIE DU ÜBER SKILLS WISSEN MUSST

Einmal einrichten — danach passiert es automatisch

Du schreibst einmal auf, wie Claude eine Aufgabe angehen soll. Danach erkennt er selbst wann der Skill relevant ist und wendet ihn an — ohne dass du jedes Mal daran denken oder ihn explizit aufrufen musst.

Je mehr du verfeinerst, desto besser wird Claude

Skills sind keine fertigen, starren Dateien. Du kannst sie jederzeit verbessern. Wenn Claude einen Report nicht ganz so erstellt wie du es willst, passt du den Skill an. Mit der Zeit wird er in dieser Aufgabe immer präziser.

Kein technisches Wissen nötig — nur eine Datei hochladen

Um fertige Community-Skills zu nutzen lädst du eine ZIP-Datei hoch — das war's. Um eigene Skills zu erstellen schreibst du eine Textdatei mit Anweisungen. Kein Code, keine Programmierung, keine Konfiguration.

WIE SIEHT DAS IM ALLTAG AUS? — 3 KONKRETE BEISPIELE

Für Marketing & Content Creators

DAS PROBLEM: Tanja arbeitet im Marketing und nutzt Claude täglich für Posts, Newsletter und Produkttexte. Das Problem: Jedes Mal erklärt sie ihm aufs Neue wie die Marke klingt, welche Worte vermieden werden sollen und wie lang Posts sein dürfen. Das kostet bei jedem Gespräch Zeit.

MIT SKILL: Mit einem Brand-Voice-Skill erklärt sie Claude einmal alles über ihre Marke. Danach liefert er jeden Text automatisch im richtigen Stil — ohne Briefing, ohne Wiederholung. Sie sagt "schreib einen Instagram-Post über unser neues Produkt" und bekommt direkt etwas das zur Marke passt.

Für Unternehmer & alle die Dokumente erstellen

DAS PROBLEM: Markus führt ein kleines Unternehmen. Er erstellt wöchentlich Reports, Angebote und Präsentationen — jedes Dokument nach einem eigenen Format das er immer mühsam erklärt. Manchmal vergisst Claude Details, manchmal sieht die Formatierung anders aus als erwartet.

MIT SKILL: Mit dem Document Suite Skill und einem eigenen Report-Skill erstellt Claude vollständige, professionell formatierte Dokumente in einem Bruchteil der Zeit. Tabellen mit Formeln, Inhaltsverzeichnis, Zusammenfassung — alles automatisch, immer im gleichen Format.

Für Entwickler

DAS PROBLEM: David entwickelt Software und nutzt Claude zum Debuggen. Das Problem: Claude probiert Lösungen aus ohne das eigentliche Problem zu verstehen. Lösung A klappt nicht, dann B, dann C. Jeder Versuch kostet Zeit und macht manchmal andere Dinge kaputt.

MIT SKILL: Mit dem Systematic-Debugging-Skill folgt Claude einem strikten 4-Phasen-Prozess: zuerst Beweise sammeln und die Ursache verstehen, dann Hypothesen aufstellen, dann und nur dann einen Fix implementieren. Studien zeigen: ~95% Erfolgsrate beim ersten Fix, statt ~40% ohne diesen Ansatz.

WAS BRAUCHST DU?

Plan	Gerät	Technisch
Alle Pläne ✓	Desktop / Web / Mobil	Kein Code nötig
Free, Pro, Max, Team, Enterprise	Installieren: Desktop Nutzen: überall	Nur eine Datei hochladen

WICHTIG zuerst: Code-Ausführung aktivieren

Skills funktionieren nicht ohne diese Einstellung:

Claude.ai → **Einstellungen** → **Capabilities** → "Code execution and file creation" einschalten

■ VIDEO: Ich zeige dir live wie ein Skill eingefügt wird

Schau dir das Reel an — damit ist alles sofort klar:

[instagram.com/reel/DZN4SKzRwUP](https://www.instagram.com/reel/DZN4SKzRwUP)

SKILL AKTIVIEREN — SCHRITT FÜR SCHRITT

1

Code-Ausführung einschalten

Einstellungen → Capabilities → Toggle "Code execution and file creation" einschalten. Das ist der einzige Setup-Schritt. Du machst das einmal und nie wieder.

2

Skill-Bereich öffnen

In Claude.ai auf "Customize" klicken → "Skills". Hier siehst du alle verfügbaren Skills und kannst eigene hochladen.

3a

Anthropic-Skills einschalten (z.B. Document Suite)

Diese Skills sind bereits integriert. Einfach den Toggle daneben einschalten — fertig. Kein Download, kein Upload, keine Einrichtung.

3b

Community-Skill als ZIP hochladen

Skill-Ordner von GitHub herunterladen → als ZIP komprimieren → "+" klicken → "Upload a skill" → ZIP auswählen → fertig.

DESKTOP VS. MOBILE — WAS GEHT WO?

AKTION	■ DESKTOP / WEB	■ MOBIL
Skills nutzen (im Chat)	✓ Ja	✓ Ja
Anthropic-Skills einschalten	✓ Ja	✗ Nur Desktop/Web
Community-Skills als ZIP hochladen	✓ Ja	✗ Nur Desktop/Web
Skills deaktivieren / verwalten	✓ Ja	✗ Nur Desktop/Web

Hinweis: Skills die du auf dem Desktop eingerichtet hast, funktionieren danach auch auf dem Handy. Du kannst sie mobil nutzen — nur das erstmalige Einrichten geht ausschließlich über Desktop oder Web.

WO FINDEST DU SKILLS?

Anthropic (offiziell)

github.com/anthropics/skills

Die offiziellen Anthropic-Skills. Qualitätsgeprüft und sicher. Empfohlen als Startpunkt.

Composio — Awesome Claude Skills

github.com/ComposioHQ/awesome-claude-skills

Große Community-Sammlung. 50+ Skills für verschiedene Anwendungsfälle. Hier kommen auch 8 der 10 Skills in diesem Guide her.

obra Superpowers

github.com/obra/superpowers

Framework für Entwickler. 20+ Skills speziell für professionelles Coding mit Claude Code.

#01 Rube MCP Connector

COMMUNITY

COMPOSIO

STELL DIR VOR...

Du nutzt täglich Slack, GitHub, Notion und Linear. Für jede dieser Apps müsstest du normalerweise eine separate Verbindung einrichten — eigene Zugangsdaten, eigene Authentifizierung, eigenes Setup. Das kostet Stunden, oft Nerven. Rube ist der Generalschlüssel: einmal einrichten, alle Apps laufen darüber.

WAS MACHT DIESER SKILL?

Rube ist ein MCP-Gateway-Server — eine universelle Verbindungsbrücke — der Claude mit über 500 Apps verbindet: Gmail, Slack, GitHub, Notion, Linear, Airtable und hunderte mehr. Du authentifizierst dich einmal bei Rube und alle Verbindungen laufen darüber. Keine separaten API-Keys, keine einzelnen Setups. Ideal für alle die Claude wirklich in ihren Arbeitsalltag integrieren wollen.

ANWENDUNGSBEISPIELE:

- "Erstelle einen GitHub-Issue und poste die Zusammenfassung automatisch in den #dev-channel auf Slack"
- "Überprüfe alle offenen Notion-Tasks dieser Woche und erstelle daraus einen priorisierten Tagesplan"
- "Schreib allen Gmail-Kontakten die seit 14 Tagen nicht geantwortet haben eine freundliche Follow-Up-Mail"

INSTALLATION: Terminal: `npx @composio/rube-mcp` | Claude Code: `/plugin install rube` | Einmalige App-Authentifizierung

#02 Superpowers

COMMUNITY

OBRA

STELL DIR VOR...

Du gibst Claude eine Coding-Aufgabe. Er fängt sofort an — ohne nachzufragen, ohne zu planen, ohne Alternativen zu überlegen. Manchmal klappt es, oft nicht auf Anhieb. Jeder Fehlversuch kostet Zeit und Frustration. Superpowers gibt Claude einen echten strukturierten Denkprozess.

WAS MACHT DIESER SKILL?

Superpowers ist obra's Developer-Toolkit für Claude Code. Es fügt drei mächtige Befehle hinzu: `/brainstorm` analysiert das Problem, stellt Fragen und durchdenkt Alternativen vor dem ersten Code. `/write-plan` erstellt einen glasklaren Implementierungsplan nach TDD-Prinzipien. `/execute-plan` arbeitet den Plan Schritt für Schritt ab, mit Validierung nach jedem Schritt. Das Framework enthält über 20 vorgefertigte Skills für professionelle Entwicklung.

ANWENDUNGSBEISPIELE:

- `/brainstorm`: "Ich will ein Login-System bauen" → Claude stellt 5 Fragen, bevor er eine Zeile schreibt
- `/write-plan` → vollständiger Plan mit Tests, Abhängigkeiten und Reihenfolge — bevor Code entsteht
- `/execute-plan` → strukturierte Ausführung mit Validierung und klaren Zwischenergebnissen

INSTALLATION: Claude Code: `/plugin marketplace add obra/superpowers-marketplace` → `/plugin install superpowers`

#03 Document Suite

OFFIZIELL

ANTHROPIC

STELL DIR VOR...

Du bittest Claude eine Excel-Tabelle für dein Budget zu erstellen. Ohne Skill bekommst du eine einfache Tabelle ohne echte Formatierung. Mit Document Suite bekommst du eine professionell formatierte Tabelle: korrekte Formeln, bedingte Farbgebung, Summenzeilen — wie von einem Office-Profi erstellt.

WAS MACHT DIESER SKILL?

Document Suite ist Anthrops eigene Skill-Suite und bereits in Claude integriert. Sie macht Claude wirklich kompetent in Word, Excel, PowerPoint und PDF — nicht nur im Lesen, sondern im tatsächlichen Erstellen mit echter Formatierung, echten Formeln und professionellen Layouts. Claude wendet die Skills automatisch an wenn du ein Dokument anforderst — du musst ihn nicht explizit daran erinnern.

ANWENDUNGSBEISPIELE:

- "Erstelle einen Q3-Report als Word-Dokument mit Inhaltsverzeichnis und Zusammenfassung"
- "Bau mir eine Excel-Tabelle für mein Monatsbudget mit automatischen Summen und Farbmarkierungen"
- "Erstelle eine 10-Slide PowerPoint Präsentation über KI-Trends 2026 mit professionellem Layout"

INSTALLATION: Bereits integriert! Claude.ai → Customize → Skills → Document Suite-Toggles einschalten. Fertig.

#04 Theme Factory

COMMUNITY

STELL DIR VOR...

Du erstellst regelmäßig Landingpages, Dashboards und Berichte mit Claude. Jedes Mal sagst du ihm: "Nutze diese Farben, diese Schrift, diesen Stil." Manchmal stimmt es, manchmal nicht. Theme Factory löst das dauerhaft: wähle ein Theme einmal — und jedes visuelle Artifact das Claude erstellt folgt diesem Design automatisch.

WAS MACHT DIESER SKILL?

Theme Factory gibt Claude 10 vorgefertigte Designthemes mit abgestimmten Farben und Fonts. Du aktivierst ein Theme mit einem einzigen Satz, und alle HTML-Artifacts, Präsentationen und visuellen Outputs dieser Unterhaltung folgen automatisch diesem Stil. Du kannst auch völlig eigene Themes basierend auf deinen Lieblingsfarben generieren lassen — Claude erstellt passende Kombinationen auf Knopfdruck.

ANWENDUNGSBEISPIELE:

- "Nutze das Dark Neon Theme für alle meine Dashboards diese Session"
- "Erstelle eine Landing Page im Arctic Frost Stil" → Ice Blue + Steel Blue Typografie automatisch
- "Generiere ein neues Theme in meinen Brand-Farben #FF6B35 und #2D3047"

INSTALLATION: GitHub: ComposioHQ/awesome-claude-skills → Ordner theme-factory → als ZIP komprimieren → Customize → Skills → Upload

#05 Algorithmic Art

COMMUNITY

STELL DIR VOR...

Du willst einen einzigartigen animierten Hintergrund für deine Website oder ein generatives Kunstwerk für Social Media. Normalerweise bräuchtest du Programmierkenntnisse in p5.js oder Processing. Mit diesem Skill beschreibst du es einfach auf Deutsch — und Claude erstellt den Code.

WAS MACHT DIESER SKILL?

Algorithmic Art generiert p5.js Generative Art aus Text-Beschreibungen. Das Besondere: durch Seed-Nummern sind Ergebnisse reproduzierbar — "Seed 42" ergibt immer exakt dasselbe Bild, egal wann du ihn ausführst. Ausgabe: interaktiver HTML-Viewer zum sofortigen Ansehen, PNG/SVG-Export und den vollständigen bearbeitbaren Code. Unterstützt Flow Fields, Partikel-Systeme, rekursive Muster und mehr.

ANWENDUNGSBEISPIELE:

- "Blau-lila Flow Field, 5000 Partikel, Seed 42" → reproduzierbares Kunstwerk als HTML + PNG-Export
- "Animierte geometrische Wellen in meinen Brand-Farben für meine Website als Loop"
- "Rekursive Blattstrukturen in Herbstfarben — 30 Sekunden animiert, für Instagram"

INSTALLATION: GitHub: [ComposioHQ/awesome-claude-skills](#) → Ordner algorithmic-art → ZIP → Customize → Skills → Upload

#06 Slack GIF Creator

COMMUNITY

COMPOSIO

STELL DIR VOR...

Dein Team deployed erfolgreich ein Feature. Du willst eine passende animierte Reaktion in Slack posten — aber nach 20 Minuten Giphy-Suche hast du immer noch nicht das was du dir vorstellst. Mit diesem Skill sagst du Claude: "Mach ein GIF von einem feiernden Roboter" — und bekommst es in Sekunden.

WAS MACHT DIESER SKILL?

Slack GIF Creator erstellt animierte GIFs die für Slack optimiert sind. Der Skill kennt Slacks Limit von 2MB und validiert automatisch ob das GIF darin passt — wenn nicht, optimiert er es selbstständig. Du beschreibst in normaler Sprache was du möchtest, Claude erstellt es mit Animations-Primitiven als herunterladbares GIF.

ANWENDUNGSBEISPIELE:

- "Sprint-Done-Celebration GIF für unseren #dev-channel mit tanzenden Code-Symbolen"
- "Animiertes Coffee-GIF für unseren Monday-Standup-Channel, warm und einladend"
- "Loading-Spinner als GIF in unseren Firmenfarben für Slack-Status-Updates"

INSTALLATION: GitHub: [ComposioHQ/awesome-claude-skills](#) → Ordner slack-gif-creator → ZIP → Customize → Skills → Upload

#07 Webapp Testing

COMMUNITY

STELL DIR VOR...

Du hast einen neuen Login-Flow entwickelt. Vor dem Deploy musst du testen: valide Credentials, falsche Passwörter, leere Felder, Rate-Limiting, Edge Cases. Normalerweise würde das Stunden dauern. Mit diesem Skill sagst du Claude was getestet werden soll — er schreibt den Code UND führt ihn direkt aus.

WAS MACHT DIESER SKILL?

Webapp Testing schreibt und führt Playwright-Automatisierungstests aus. Playwright ist Microsofts modernes Test-Framework für Web-Apps und kann Chrome, Firefox und Safari gleichzeitig testen. Du beschreibst in normaler Sprache was getestet werden soll, Claude schreibt den vollständigen Test-Code, führt ihn aus und liefert dir einen übersichtlichen Bericht mit Ergebnissen.

ANWENDUNGSBEISPIELE:

- "Teste den kompletten Login-Flow: richtige Credentials, falsches Passwort, leeres Feld, Rate-Limit"
- "Prüfe alle Formulare auf der Checkout-Seite auf korrekte Validierung und Fehlermeldungen"
- "Automatisierter Smoke-Test für das neue Dashboard-Feature vor dem nächsten Staging-Deploy"

INSTALLATION: GitHub: [ComposioHQ/awesome-claude-skills](#) → webapp-testing → ZIP → Upload (Playwright muss installiert sein)

#08 MCP Builder

COMMUNITY

COMPOSIO

STELL DIR VOR...

Du willst eine eigene Claude-Integration bauen — einen Connector zu deiner internen Datenbank oder einem proprietären API. Ohne diesen Skill startest du von Null: Projektstruktur, Error-Handling, Logging, Security, Input-Validierung — alles selbst. MCP Builder generiert den kompletten Grundaufbau in unter 15 Minuten.

WAS MACHT DIESER SKILL?

MCP Builder generiert produktionsreifen MCP-Server-Boilerplate in TypeScript oder Python. Er kennt alle Best Practices: korrekte Error-Handling-Pattern, Rate-Limiting, Input-Validierung, Logging-Setup und Security-Considerations — alles bereits eingebaut. Was normalerweise 2-4 Stunden reines Setup kostet, dauert damit unter 15 Minuten. TypeScript wird bevorzugt wegen besserem SDK-Support, Python (FastMCP) auch verfügbar.

ANWENDUNGSBEISPIELE:

- "Erstelle einen MCP-Server der auf unsere interne PostgreSQL-Datenbank zugreift"
- "TypeScript MCP-Connector für unsere REST-API mit OAuth2-Authentifizierung und Rate-Limiting"
- "Python FastMCP-Server der unsere proprietären Firmendokumente für Claude lesbar macht"

INSTALLATION: GitHub: ComposioHQ/awesome-claude-skills → Ordner mcp-builder → ZIP → Claude Code → Customize → Skills → Upload

#09 Brand Guidelines

COMMUNITY

COMPOSIO

STELL DIR VOR...

Du arbeitest für eine Agentur und betreust drei Kunden mit drei verschiedenen Brands. Für jeden brauchst du andere Farben, andere Fonts, andere Tonalität, andere Regeln. Mit Brand Guidelines speicherst du alle drei Identitäten einmal als Skill — und wechselst mit einem einzigen Satz zwischen ihnen.

WAS MACHT DIESER SKILL?

Brand Guidelines speichert vollständige Brand-Identitäten: Farben mit exakten Hex-Werten, Fonts, Tonalität der Kommunikation, Schreibregeln und was vermieden werden soll. Claude wendet sie auf alle Outputs an: Websites, Präsentationen, Texte, Dokumente. Im Gegensatz zu Theme Factory können mehrere komplette Marken verwaltet und jederzeit gewechselt werden — ideal für Agenturen oder Freelancer.

ANWENDUNGSBEISPIELE:

- "Erstelle eine Landing Page" → Claude nutzt automatisch Brand A's exakte Farben und Tonalität
- "Wechsle zu Brand B und schreib den gleichen Text für deren Newsletter-Stil um"
- "Erstelle eine Präsentation für Kunde C" → korrekte Fonts, Farben und Sprachregeln automatisch

INSTALLATION: GitHub: [ComposioHQ/awesome-claude-skills](#) → brand-guidelines → eigene Brand-Daten eintragen → ZIP
→ Upload

#10 Systematic Debugging

COMMUNITY

STELL DIR VOR...

Du hast einen Bug. Claude probiert Lösung A — klappt nicht. Dann B — klappt nicht. Dann C und D. Jeder Versuch macht manchmal andere Dinge kaputt. Das Grundproblem: Claude versteht die Ursache nicht, er rät. Systematic Debugging macht damit Schluss: kein einziger Fix ohne vollständig verstandene Ursache.

WAS MACHT DIESER SKILL?

Systematic Debugging erzwingt ein striktes 4-Phasen-Framework: 1. Beweise sammeln — Logs lesen, Datenfluss verstehen, Kontext aufnehmen. 2. Muster analysieren — mit funktionierendem Code vergleichen, Gemeinsamkeiten finden. 3. Hypothesen aufstellen und minimal testen — eine Variable nach der anderen. 4. Implementieren und dokumentieren — erst wenn die Ursache klar ist. Die eiserne Regel: KEIN Fix ohne Root Cause. Ergebnis: ~95% First-Fix-Rate gegenüber ~40% ohne diesen Ansatz.

ANWENDUNGSBEISPIELE:

- "Dieser API-Call schlägt manchmal fehl" → strukturierte Ursachenanalyse statt Random-Fixes
- "Unser Build bricht seit gestern" → Diffs, Logs, letzte Änderungen — erst dann ein Fix
- "Performance-Problem in Production" → Messung, Hypothesen, Lösung, Dokumentation für das Team

INSTALLATION: GitHub: obra/superpowers (bereits enthalten) oder ComposioHQ/awesome-claude-skills → systematic-debugging → ZIP

EMPFEHLUNGEN: SO FÄNGST DU AM BESTEN AN

Fang mit Document Suite an — er ist bereits eingebaut

Document Suite ist schon in Claude integriert, du musst ihn nur aktivieren: Customize → Skills → Toggles bei Word, Excel und PowerPoint einschalten. Das ist in unter 30 Sekunden erledigt — kein Download, kein Upload. Danach bitte Claude um einen Excel-Report oder eine PowerPoint-Präsentation und erlebe sofort den Unterschied. Das ist die beste Art um Skills kennenzulernen ohne sich mit Installationen zu befassen.

Rube + Superpowers sind "Daily Drivers"

"Daily Drivers" bedeutet: Skills die du jeden einzelnen Tag nutzt, nicht einmal die Woche — sondern bei jeder Session. Rube ist Daily Driver für alle die Claude mit ihren Apps verbinden wollen: Slack, Gmail, Notion, GitHub. Superpowers ist Daily Driver für Entwickler die täglich mit Claude Code arbeiten. Diese einmalige Einrichtung zahlt sich schon nach dem ersten Tag aus.

Maximal 20–50 Skills gleichzeitig aktiv halten

Hier steckt Technik dahinter: Claude lädt beim Start jeder Unterhaltung die Metadaten aller aktiven Skills — das sind die kurzen Beschreibungen die Claude sagen wann er welchen Skill nutzen soll. Jeder Skill kostet dabei etwa 30-50 "Token" (Verarbeitungseinheiten). Bei 5 Skills: kaum spürbar. Bei 100 Skills: tausende Token werden für Skills verschwendet die du gar nicht brauchst, Claude wird langsamer und manchmal weniger präzise. Faustregel: Nur aktiv halten was du regelmäßig nutzt.

Community-Skills kurz prüfen bevor du sie installierst

Skills können beliebigen Code enthalten — das klingt bedrohlicher als es ist, aber ein kurzer Blick lohnt sich. Öffne die SKILL.md Datei im ZIP bevor du hochlädst. Was steht darin? Macht das Sinn? Gibt es Script-Dateien im Ordner? Lies kurz rein. Die Repos von github.com/anthropics/skills und github.com/ComposioHQ/awesome-claude-skills werden community-geprüft und sind generell vertrauenswürdig. Bei unbekannten Quellen aus dem Internet kurz nachschauen.