

CLAUDE SKILLS

Die 5 Skills, die was bringen

Was jeder davon macht, wie du ihn installierst und wann du ihn wirklich brauchst

5 Skills

handverlesen

10 Min

alle installiert

Claude Code

läuft überall

Skills sind einfach Ordner mit Anweisungen, die Claude sich selbst zieht, sobald sie zum Thema passen.

Du musst sie also nicht aufrufen und nicht daran denken. Du installierst sie einmal und ab da arbeitet Claude anders, und genau das ist der Punkt, warum die meisten Leute mit demselben Claude viel schlechtere Ergebnisse bekommen.

Alle fünf sind Community-Skills auf GitHub, also nichts Offizielles von Anthropic. Ich habe die Quellen am 27. August 2026 geprüft und jeweils unten auf der Seite verlinkt. Schau vor dem Installieren halt kurz selber rein, weil sich so ein Repo natürlich ändern kann.

NUMMER 1

Stop Slop

Damit dein Text nicht mehr nach KI klingt

01 Was er macht

Der Skill kennt die typischen KI-Muster und wirft sie raus. Also Gedankenstriche, Floskel-Einstiege, Business-Jargon und diese Gegensatz-Sätze nach dem Schema "nicht X, sondern Y". Am Ende bewertet er den Text auf fünf Achsen von 1 bis 10, und unter 35 von 50 Punkten sagt er dir, dass du nochmal ranmusst.

INSTALLATION

```
git clone https://github.com/hardikpandya/stop-slop \
~/.claude/skills/stop-slop
```

02 Wann du ihn brauchst

Immer wenn Text rausgeht, den eine echte Person liest. Also Kundenmails, Angebotstexte, Webseitentexte, Captions. Du schreibst einfach "schreib mir die Mail an den Kunden" und der Skill greift von alleine, du musst ihn nicht erwähnen.

03 Worauf du achten solltest

Es gibt mehrere Kopien von dem Repo unter anderen Namen. Nimm das Original von hardikpandya und lass den Ordner komplett, also mit dem Unterordner references, sonst fehlt dem Skill die halbe Regelliste.

Quelle: github.com/hardikpandya/stop-slop (MIT). Das README nennt keinen fertigen Installbefehl, sondern nur "Ordner als Skill hinzufügen". Die Zeile oben ist daraus abgeleitet.

NUMMER 2

Claude Mem

Gedächtnis über alle Sitzungen hinweg

01 Was er macht

Er zeichnet auf, was in einer Sitzung passiert ist, also welche Dateien angefasst wurden und welche Entscheidungen gefallen sind, fasst das zusammen und legt es durchsuchbar ab. Beim nächsten Start spielt er den passenden Teil davon wieder ein.

INSTALLATION

```
npx claude-mem install

# oder als Plugin
/plugin marketplace add thedotmack/claude-mem
/plugin install claude-mem
```

02 Wann du ihn brauchst

Wenn du an einem Projekt über Wochen arbeitest. Du fängst eine neue Sitzung an und musst nicht nochmal erklären, wie das Projekt aufgebaut ist und was ihr letztes Mal verworfen habt.

03 Worauf du achten solltest

Installier ihn nicht mit `npm install -g claude-mem`. Das schreibt die Doku selber, weil dabei nur die Library landet und weder die Hooks registriert noch der Worker gestartet werden. Dann liegt der Skill da und tut halt nichts.

Quelle: github.com/thedotmack/claude-mem, [Doku docs.claude-mem.ai/installation](https://docs.claude-mem.ai/installation)

NUMMER 3

UI UX Pro Max

Damit deine Oberflächen nicht nach KI aussehen

01 Was er macht

Das ist eine Design-Wissensbasis, die anspringt, sobald du an einer Oberfläche baust. Laut Repo stecken da 67 UI-Stile drin, 161 Farbpaletten, 57 Font-Paare und 99 UX-Regeln, dazu Vorgaben für Layout, Typo, Barrierefreiheit und Responsive.

INSTALLATION

```
/plugin marketplace add nextlevelbuilder/ui-ux-pro-max-skill  
/plugin install ui-ux-pro-max@ui-ux-pro-max-skill
```

```
# oder über die CLI  
npx ui-ux-pro-max-cli init --ai claude
```

02 Wann du ihn brauchst

Bei Landingpages, Dashboards und kleinen Tools. Ohne den Skill kommt bei jedem Prompt derselbe lila Verlauf mit denselben abgerundeten Kacheln raus, und das sieht man einem Projekt sofort an.

03 Worauf du achten solltest

Sag Claude vorher, in welcher Richtung du unterwegs bist, also welche Marke und welche Farben. Sonst sucht er sich aus den 161 Paletten einfach eine aus und du bekommst ein sauberes Design, das aber nicht nach dir aussieht.

Quelle: github.com/nextlevelbuilder/ui-ux-pro-max-skill

NUMMER 4

Task Observer

Verbessert deine anderen Skills von alleine

01 Was er macht

Er schaut der laufenden Sitzung zu und schreibt mit, wo du korrigierst, was du immer wieder anders haben willst und welche Abläufe sich wiederholen. Das landet in einer Log-Datei, und daraus schlägt er dir neue Skills vor oder Verbesserungen an deinen bestehenden.

INSTALLATION

```
git clone https://github.com/rebelytics/one-skill-to-rule-them-all /tmp/osra  
cp -r /tmp/osra/task-observer ~/.claude/skills/
```

02 Wann du ihn brauchst

Wenn du merkst, dass du Claude jedes Mal dasselbe hinterherrufst. Statt dir das selber zu notieren, lässt du ihn eine Woche mitlaufen und schaust dann in die Log-Datei, was sich da gesammelt hat.

03 Worauf du achten solltest

Der Skill gehört zum Projekt One Skill to Rule Them All, du klonst also das ganze Repo und kopierst nur den Unterordner raus. Prüf vorher kurz, wie der Ordner im Repo heißt. Es gibt außerdem gleichnamige Kopien in anderen Skill-Sammlungen, das Original liegt bei rebelytics.

Quelle: github.com/rebelytics/one-skill-to-rule-them-all (CC BY 4.0). Auch hier gibt es keinen offiziellen Ein-Zeilen-Befehl, die Zeilen oben sind aus der Doku abgeleitet.

NUMMER 5

Find Skills

Der Skill, der dir die anderen Skills sucht

01 Was er macht

Du sagst einfach, was du bauen willst, und er durchsucht das offene Skill-Verzeichnis danach, schaut sich Quelle und Installationszahlen an und installiert dir die passenden direkt.

INSTALLATION

```
npx skills add vercel-labs/skills/find-skills --agent claude-code
```

02 Wann du ihn brauchst

Immer dann, wenn du ein neues Thema anfängst. Du schreibst zum Beispiel "ich will eine Rechnungs-App bauen" und er sucht dir raus, was es dafür schon gibt, statt dass du dich durch GitHub klickst.

03 Worauf du achten solltest

Er installiert fremden Code auf deinem Rechner. Lass dir also kurz zeigen, was er gefunden hat, bevor du ihn installieren lässt, und schau bei jedem Vorschlag auf die Quelle.

Quelle: github.com/vercel-labs/skills, Unterordner `skills/find-skills`

SELBER SUCHEN

So findest du weitere

Es gibt drei Wege und du brauchst eigentlich nur den ersten

01 Claude suchen lassen

Wenn Find Skills installiert ist, sagst du einfach, was du vorhast. Er sucht, zeigt dir die Treffer und installiert auf Zuruf. Das ist der bequemste Weg und deshalb steht der Skill auch mit auf der Liste.

02 Marketplace direkt anzapfen

Jedes GitHub-Repo mit Skills kannst du als Marketplace einhängen und dir dann anschauen, was drin ist. Danach installierst du gezielt einzelne Sachen daraus.

IN CLAUDE CODE EINTIPPEN

```
/plugin marketplace add <benutzer>/<repo>
/plugin

# Verzeichnis durchsuchen, ohne Claude zu fragen
npx skills find "<wonach du suchst>"
```

03 Von Hand ablegen

Am Ende ist ein Skill nur ein Ordner mit einer SKILL.md. Du kopierst ihn nach ~/.claude/skills/ und Claude findet ihn beim nächsten Start. Wenn er nur für ein Projekt gelten soll, legst du ihn stattdessen in .claude/skills/ im Projektordner ab.

Installier nicht alles, was gut klingt.

Jeder Skill, der anspringt, kostet Kontext, und wenn zwanzig Stück gleichzeitig mitreden, wird Claude eher schlechter. Ich fahre selber mit einer Handvoll, die ich wirklich benutze, und schmeiße den Rest wieder raus.

NÄCHSTER SCHRITT

Direkt umsetzen

Fang mit zweien an, nicht mit allen fünf

01 Find Skills zuerst

Installier den einen Befehl und lass Claude ab da selber suchen. Damit sparst du dir das Durchklicken bei allem, was danach kommt.

02 Dann den, der zu deiner Arbeit passt

Du schreibst viel Text, dann nimm Stop Slop. Du baust Oberflächen, dann UI UX Pro Max. Du arbeitest lange am selben Projekt, dann Claude Mem.

03 Eine Woche laufen lassen

Setz Task Observer dazu und schau nach ein paar Tagen in die Log-Datei. Da steht dann schwarz auf weiß, wo du Claude ständig korrigierst, und daraus baust du dir deinen eigenen Skill.

Du willst die Skills sehen, die es wirklich in mein Setup schaffen?

Auf Instagram zeige ich, was bei Claude Code neu ist und was ich davon behalte, sobald ich es getestet habe.

→ [@timhuck.ai auf Instagram](#)

→ www.shortmedia.agency

Alle Quellen geprüft am 27. August 2026. Es sind Community-Repos, keine offiziellen Anthropic-Skills, und Installationswege können sich ändern. Schau vor dem Installieren einmal ins jeweilige README.