

Dein KI- Werkzeugkasten für **Claude Code** und **Codex**

Was jedes Plugin kann, mit Befehl und Beispiel zum Einfügen. Auf den ersten Blick verständlich, mit Hinweisen zu Codex.

15 Plugins

3 Skill-Pakete

Stand 9. Oktober 2026

So startest du in 30 Sekunden

1. **Terminal** öffnen (nicht die Claude-App).
2. In dein Projekt wechseln: `cd` tippen, einmal Leertaste, den Projektordner aus dem Finder ins Terminal ziehen, Enter.
3. `claude` eingeben und Enter.
4. Befehl mit `/` tippen oder einfach in normalen Worten schreiben. Claude wählt den passenden Skill meist selbst.

Tipp: `/` plus die ersten Buchstaben zeigt dir alle passenden Befehle. Bei Plugins steht manchmal der Plugin-Name davor, zum Beispiel `/marketing-skills:copywriting` . Beides führt zum selben Skill.

Für den Maps-Scraper startest du immer so: `cd ~/google-maps-scraper-kit && claude` (Docker Desktop muss offen sein).

Schnellwahl: Ich will ...

Ich will ...	Nimm	Erster Befehl
eine neue Funktion im Dashboard sauber bauen	agent-skills	<code>/spec</code>
einen Fehler finden und beheben	agent-skills oder superpowers	<code>/test</code>
eine Seite vor dem Livegang prüfen	agent-skills	<code>/ship</code>
Ladezeit und Core Web Vitals prüfen	agent-skills, chrome-devtools-mcp	<code>/webperf</code>
eine Landingpage, die nicht nach KI aussieht	frontend-design, impeccable	<code>/impeccable craft</code>
Texte und Conversion einer Seite verbessern	marketing-skills	<code>/copywriting</code>
ein Angebot schärfen	marketing-skills	<code>/offers</code>
SEO-Check, Keywords, Konkurrenz	openseo, marketing-skills	<code>/seo-audit</code>
Instagram Reels, Captions, Wochenplan	instagram-agent	<code>/ig-reel</code>
ein Video für Social Media bauen	remotion	<code>/remotion-create</code>
Word, Excel, PDF oder PowerPoint erstellen	document-skills	in normalen Worten
Animationen und UI-Feinschliff	emil	<code>/animate</code>
Betriebe aus Google Maps sammeln	google-maps-scraper-kit	<code>/scrape</code>
kürzere Antworten und weniger Code	ponytail	<code>/ponytail</code>
ein fremdes Projekt schnell verstehen	graphify	<code>/graphify</code> .
echte App-Designs als Vorlage ansehen	refero	in normalen Worten

Bauen und Qualität

agent-skills

Plugin

Wofür: Ein fester Ablauf für jede Aufgabe: erst klären, dann planen, bauen, testen, prüfen, live gehen. Weniger Fehler, weniger Nacharbeit.

Quelle: github.com/addyosmani/agent-skills

Befehl	Was passiert	Beispiel
<code>/spec</code>	Schreibt erst eine Beschreibung, bevor Code entsteht	<code>/spec</code> Im Akquise-Dashboard soll man Leads nach Status filtern können
<code>/plan</code>	Teilt die Arbeit in kleine, prüfbare Schritte	<code>/plan</code>
<code>/build</code>	Setzt Schritt für Schritt um, testet, speichert	<code>/build</code>
<code>/test</code>	Test zuerst, dann die Lösung. Bei Fehlern wird der Fehler erst nachgestellt	<code>/test</code> Der Login schlägt bei Umlauten im Namen fehl
<code>/review</code>	Prüft Korrektheit, Lesbarkeit, Aufbau, Sicherheit, Tempo	<code>/review</code>
<code>/code-simplify</code>	Macht Code übersichtlicher, ohne das Verhalten zu ändern	<code>/code-simplify</code>
<code>/webperf</code>	Misst Ladezeit und Core Web Vitals	<code>/webperf</code>
<code>/ship</code>	Startklar-Check vor dem Livegang, am Ende Go oder No-Go	<code>/ship</code>
<code>/constraints</code>	Legt Qualitätsgrenzen fest, zum Beispiel maximale Ladezeit	<code>/constraints</code>

Dahinter liegen 25 Skills, die Claude selbst zieht (Sicherheit, Barrierefreiheit, Fehlersuche, Datenbank-Migration). Dazu vier Prüf-Agenten: code-reviewer, security-auditor, test-engineer, web-performance-auditor.

superpowers

Plugin

Wofür: Eine zweite, strengere Arbeitsweise: erst Ideen klären, dann Plan, dann Umsetzung mit Kontrolle. Läuft meist von allein.

Quelle: github.com/obra/superpowers

Skill	Was passiert	Beispiel
brainstorming	Fragt nach, bevor etwas gebaut wird	Lass uns erst brainstormen: Kunden-Login fürs Dashboard
writing-plans	Schreibt einen Plan aus einer Beschreibung	Schreibe einen Plan dafür
systematic-debugging	Sucht die Ursache, bevor etwas repariert wird	Finde die Ursache, warum der Export leer ist
test-driven-development	Tests zuerst	Baue das testgetrieben
requesting-code-review	Prüfung nach größeren Änderungen	Lass das reviewen, bevor wir fertig sind
verification-before-completion	Sagt erst "fertig", wenn es bewiesen ist	läuft automatisch

ponytail Plugin

Wofür: Claude baut die kleinste Lösung, die das Problem wirklich löst, und antwortet kurz. Spart Code und Tokens.

Quelle: github.com/DietrichGebert/ponytail

Befehl	Was passiert	Beispiel
<code>/ponytail</code>	Einschalten oder Stufe anzeigen (Standard: full)	<code>/ponytail</code>
<code>/ponytail lite</code>	Baut, was verlangt ist, nennt die kleinere Alternative	<code>/ponytail lite</code>
<code>/ponytail ultra</code>	Hinterfragt die Aufgabe, bevor gebaut wird	<code>/ponytail ultra</code>
<code>/ponytail off</code>	Ausschalten	<code>/ponytail off</code>
<code>/ponytail-review</code>	Prüft die aktuelle Änderung auf Fehler, Risiko, Überflüssiges	<code>/ponytail-review uncommitted</code>
<code>/ponytail-audit</code>	Prüft das ganze Projekt, Wichtigstes zuerst	<code>/ponytail-audit</code>
<code>/ponytail-debt</code>	Listet bewusst aufgeschobene Abkürzungen	<code>/ponytail-debt</code>

graphify Skill

Wofür: Baut eine Wissenskarte deines Projekts. Claude muss nicht jedes Mal alle Dateien neu lesen.

Quelle: graphify.com

Befehl	Was passiert	Beispiel
<code>/graphify .</code>	Erstellt die Karte für den aktuellen Ordner (einmal pro Projekt)	<code>/graphify .</code>

chrome-devtools-mcp Plugin

Wofür: Claude schaut in einen echten Chrome und prüft Seiten wie ein Entwickler: Fehler, Tempo, Barrierefreiheit.

Quelle: github.com/ChromeDevTools/chrome-devtools-mcp

Skill	Was passiert	Beispiel
debug-optimize-lcp	Findet, warum die Seite langsam lädt	Warum ist der LCP auf meiner Startseite so schlecht?
a11y-debugging	Prüft Barrierefreiheit und Tastatur-Bedienung	Prüfe die Seite auf Barrierefreiheit
cookie-debugging	Prüft Cookies, Login-Probleme, Cookie-Banner	Warum werde ich ständig ausgeloggt?
memory-leak-debugging	Sucht Speicherlecks	Die App wird nach einer Stunde langsam

Design

Wichtig: Fünf bis sechs Design-Pakete liegen jetzt parallel. Teste sie an einer echten Landingpage und behalte zwei.

frontend-design Plugin, offiziell von Anthropic

Wofür: Claude gestaltet bewusster und fällt nicht in die typische KI-Optik. Wirkt automatisch bei Design-Aufgaben.

Quelle: github.com/anthropics/skills/tree/main/skills/frontend-design

Befehl	Was passiert	Beispiel
<code>/frontend-design</code>	Gestaltung mit klarer Richtung, Schrift, Raster	Baue eine Landingpage für einen Dachdecker in Steyr

impeccable Plugin

Wofür: Ein ganzes Design-Werkzeug mit vielen Handgriffen: planen, bauen, prüfen, verbessern.

Quelle: github.com/pbakaus/impeccable

Befehl	Was passiert	Beispiel
<code>/impeccable init</code>	Einmaliges Setup mit Produkt- und Markenwissen	<code>/impeccable init</code>
<code>/impeccable craft</code>	Plant und baut mit Sichtkontrolle	<code>/impeccable craft Hero-Bereich mit Anfrage-Formular</code>
<code>/impeccable critique</code>	Design-Kritik zu Klarheit und Hierarchie	<code>/impeccable critique</code>
<code>/impeccable audit</code>	Technische Prüfung: Barrierefreiheit, Tempo, Mobil	<code>/impeccable audit</code>
<code>/impeccable polish</code>	Letzter Feinschliff vor dem Livegang	<code>/impeccable polish</code>
<code>/impeccable typeset</code>	Schriften, Größen und Hierarchie korrigieren	<code>/impeccable typeset</code>
<code>/impeccable layout</code>	Abstände und Rhythmus korrigieren	<code>/impeccable layout</code>
<code>/impeccable bolder</code> / <code>quieter</code>	Mutiger oder ruhiger machen	<code>/impeccable quieter</code>
<code>/impeccable harden</code>	Fehlerfälle, lange Texte, Randfälle absichern	<code>/impeccable harden</code>

taste-skill Plugin

Wofür: Design-Stile als Vorgabe: minimalistisch, weich, brutalistisch, hochwertig. Gegen Standard-Optik.

Quelle: github.com/Leonxlnx/taste-skill

Skill	Was passiert	Beispiel
design-taste-frontend	Baut Landingpages mit eigener Designrichtung	Baue eine Landingpage im Stil design-taste-frontend
minimalist-ui	Ruhige, redaktionelle Oberflächen	Gestalte das Dashboard minimalistisch
high-end-visual-design	Wirkt wie von einer teuren Agentur	Gib dem Hero ein hochwertiges Finish
redesign-existing-projects	Wertet bestehende Seiten auf	Überarbeite die Startseite mit redesign-existing-projects

ui-ux-pro-max Plugin

Wofür: Eine Nachschlage-Datenbank mit 79 Stilen, 192 Farbpaletten und 74 Schriftpaarungen. Gut, wenn du Farben und Schriften wählen willst.

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

Skill	Was passiert	Beispiel
/ui-ux-pro-max	Wählt Stil, Farben, Schriften passend zur Branche	Gib mir eine Palette und Schriften für einen Handwerksbetrieb
/design-system	Baut Design-Tokens und Komponenten-Regeln	/design-system
/banner-design	Banner für Social und Anzeigen	/banner-design Meta-Ad Banner für Recruiting
/slides	Präsentationen als HTML	/slides

emil

Skills

Wofür: Feinschliff bei Bewegung und Details. Macht Oberflächen hochwertig und flüssig.

Quelle: github.com/emilkowalski/skills

Befehl	Was passiert	Beispiel
<code>/animate</code>	Baut eine Animation von Grund auf	<code>/animate Menü, das sanft aufklappt</code>
<code>/improve-animations</code>	Prüft alle Animationen im Projekt	<code>/improve-animations</code>
<code>/find-animation-opportunities</code>	Zeigt Stellen, die Bewegung brauchen	<code>/find-animation-opportunities</code>
<code>/break-ui</code>	Testet die Oberfläche mit Extremdaten	<code>/break-ui</code>
<code>/mobile-native</code>	Macht eine Web-App am Handy wie eine App	<code>/mobile-native</code>
<code>/pick-ui-library</code>	Empfiehl passende Bibliotheken	<code>/pick-ui-library</code>

refero

Plugin, Konto nötig

Wofür: Design-Beispiele aus echten Apps und Seiten als Vorlage. Die Live-Recherche ist laut Hersteller kostenpflichtig.

Quelle: github.com/referodesign/refero_skill

Skill	Was passiert	Beispiel
refero-design	Sucht Referenzen und baut danach	<code>Zeig mir Pricing-Seiten als Vorlage</code>

Marketing und Inhalte

marketing-skills

Plugin

Wofür: 50 Marketing-Skills für Texte, Angebote, Conversion, SEO, Anzeigen, Mails. Der wichtigste für deine Arbeit.

Quelle: github.com/coreyhaines31/marketingskills

Befehl	Was passiert	Beispiel
<code>/copywriting</code>	Schreibt oder überarbeitet Seitentexte	<code>/copywriting</code> Startseite für einen Dachdecker, Ziel: Anfragen
<code>/cro</code>	Mehr Anfragen aus bestehenden Seiten	<code>/cro</code> Prüfe diese Landingpage
<code>/offers</code>	Angebot, Garantie, Bonus, Knappheit	<code>/offers</code> Social-Recruiting-Paket schärfen
<code>/ad-creative</code>	Anzeigentexte in Varianten	<code>/ad-creative</code> Meta Ads für Handwerker-Recruiting
<code>/lead-magnets</code>	Ideen für Freebies	<code>/lead-magnets</code>
<code>/seo-audit</code>	SEO-Probleme finden	<code>/seo-audit</code> digitalgeschick.at
<code>/prospecting</code>	Zielkunden finden und bewerten	<code>/prospecting</code> Dachdecker in Oberösterreich
<code>/marketing-plan</code>	Komplette Marketing-Planung	<code>/marketing-plan</code>
<code>/marketing-psychology</code>	Verhaltenspsychologie im Marketing	<code>/marketing-psychology</code>
<code>/pricing</code>	Preise und Pakete	<code>/pricing</code>

openseo

Plugin, Konto nötig

Wofür: SEO mit echten Daten: Keywords, Konkurrenz, Local SEO, Sichtbarkeit in KI-Antworten. Beim ersten Einsatz meldest du dich im Browser an.

Quelle: github.com/every-app/open-seo

Skill	Was passiert	Beispiel
keyword-research	Findet und bewertet Suchbegriffe	Keywords für Dachdecker in Steyr
competitor-analysis	Analysiert einen Konkurrenten	Analysiere faatz.at
local-seo	Prüft Google-Unternehmensprofil und Maps-Sichtbarkeit	Prüfe das Maps-Profil dieses Betriebs
seo-audit	Kurzer, datenbasierter Bericht	Auditire meine Website
ai-visibility-audit	Wie sichtbar bist du in KI-Antworten	Wie taucht Digitalgeschick in ChatGPT auf?

instagram-agent

Plugin

Wofür: Instagram-Inhalte von der Idee bis zum Plan. Texte, Skripte, Wochenplanung.

Quelle: github.com/Jakeschincariol/instagram-agent-skill

Befehl	Was passiert	Beispiel
/ig-reel	Reel-Skript mit Hook und Szenenplan	/ig-reel Idee: 3 Fehler auf Handwerker-Websites
/ig-caption	Caption mit Hook, Ask, Hashtags	/ig-caption
/ig-carousel	Karussell mit Folientexten	/ig-carousel
/ig-plan	Wochenplan für Posts	/ig-plan
/ig-profile	Profil bewerten und neu schreiben	/ig-profile @digi.faki
/ig-repurpose	Aus einem langen Inhalt eine Woche Beiträge	/ig-repurpose
/ig-dm	Direktnachrichten, die Antworten bekommen	/ig-dm
/ig-human	Entfernt KI-Spuren aus Texten	/ig-human

Dateien und Video

document-skills

Plugin

Wofür: Claude erstellt und bearbeitet Word, Excel, PDF und PowerPoint.

Quelle: github.com/anthropics/skills

Skill	Was passiert	Beispiel
docx	Word-Dokumente	Erstelle ein Angebot als Word-Datei
xlsx	Tabellen mit Formeln	Baue eine Kalkulation als Excel-Datei
pdf	PDFs lesen, erstellen, zusammenfügen	Füge diese PDFs zusammen
pptx	Präsentationen	Erstelle ein Pitch-Deck als PowerPoint

example-skills

Plugin

Wofür: Zusatzwerkzeuge von Anthropic. Brauchbar sind vor allem diese:

Quelle: github.com/anthropics/skills

Skill	Was passiert	Beispiel
canvas-design	Poster und Grafiken als PNG oder PDF	Gestalte ein Plakat für ein Event
doc-coauthoring	Geführtes Schreiben von Konzepten	Wir schreiben zusammen ein Konzept
webapp-testing	Testet lokale Web-Apps im Browser	Teste das Formular im Dashboard
theme-factory	10 fertige Designthemen für Dokumente und Seiten	Wende ein Theme auf die Folien an
skill-creator	Eigene Skills bauen	Baue einen Skill für mein Angebots-Skript

remotion

Skills

Wofür: Videos aus Code. Claude baut Animationen, Erklärvideos, Untertitel und exportiert eine MP4-Datei für Social Media.

Quelle: github.com/remotion-dev/skills

Befehl	Was passiert	Beispiel
<code>/remotion-create</code>	Neues Video-Projekt anlegen	<code>/remotion-create 15 Sekunden</code> Instagram-Video mit Logo und drei Aussagen
<code>/remotion-captions</code>	Untertitel erzeugen und animieren	<code>/remotion-captions</code>
<code>/remotion-studio</code>	Vorschau im Browser	<code>/remotion-studio</code>
<code>/remotion-render</code>	Fertiges Video exportieren	<code>/remotion-render</code>
<code>/remotion-best-practices</code>	Weiß alles, wenn du unsicher bist	<code>/remotion-best-practices</code>

Sonstiges

google-maps-scraper-kit

Ordner, nicht global

Wofür: Betriebe aus Google Maps in eine Liste mit Telefon, E-Mail, Website und Social-Links.

Quelle: github.com/mahanaicoach/google-maps-scraper-kit

Befehl	Was passiert	Beispiel
<code>/scrape-setup</code>	Startet den Scraper (Docker muss laufen)	<code>/scrape-setup</code>
<code>/scrape</code>	Eine Suche	<code>/scrape Dachdecker in Steyr</code>
<code>/scrape-batch</code>	Mehrere Begriffe und Orte in einer Liste	Dachdecker, Spengler, Zimmerei in Steyr, Garsten, Sierning
<code>/scrape-jobs</code>	Laufende und fertige Aufträge ansehen	<code>/scrape-jobs</code>

Starten: `cd ~/google-maps-scraper-kit && claude` . Die Liste landet als CSV im selben Ordner.

caveman Plugin

Wofür: Antworten im Telegrammstil, das spart Tokens. Überschneidet sich mit ponytail.

Quelle: github.com/JuliusBrussee/caveman

Befehl	Was passiert	Beispiel
<code>/caveman</code>	Kurzmodus an	<code>/caveman</code>
<code>/caveman-review</code>	Code-Review mit einer Zeile pro Fund	<code>/caveman-review</code>
<code>/caveman-commit</code>	Kurze Commit-Nachricht	<code>/caveman-commit</code>
<code>stop caveman</code>	Kurzmodus aus	<code>stop caveman</code>

Auch mit Codex nutzen

Die meisten dieser Pakete folgen dem offenen Skill-Format und laufen auch in Codex von OpenAI. Drei Unterschiede: In Codex rufst du Skills mit `$` statt `/` auf (zum Beispiel `$ponytail`). Die Installation läuft über `codex plugin ...` oder `npx skills`. Befehle aus den `.claude/commands`-Ordnern sowie Claude-spezifische Hooks und Agenten gehen nicht automatisch mit.

Die Zeilen stammen aus den README-Dateien der Projekte. Ich habe sie nicht auf einem Mac mit Codex getestet.

Paket	In Codex	Installation laut README
agent-skills	ja	<code>codex plugin marketplace add addyosmani/agent-skills</code> , danach <code>codex plugin add agent-skills@agent-skills</code>
ponytail	ja	<code>codex plugin marketplace add DietrichGebert/ponytail</code> , Aufruf <code>\$ponytail:ponytail-review</code>
superpowers	ja	über den offiziellen Codex-Marktplatz, in der Codex-App unter Plugins
marketing-skills	ja	<code>codex plugin marketplace add coreyhaines31/marketingskills</code>
impeccable	ja	<code>npx impeccable install</code> , danach in Codex <code>/hooks</code> öffnen und freigeben

Paket	In Codex	Installation laut README
remotion	ja	<code>npx skills add remotion-dev/skills</code>
taste-skill	ja	<code>npx skills add https://github.com/Leonxlnx/taste-skill</code>
ui-ux-pro-max	ja	<code>uipro init --ai codex</code> (nach <code>npm install -g ui-ux-pro-max-cli</code>)
chrome-devtools-mcp	ja	als MCP-Server in Codex eintragen, siehe <code>docs/client-configurations.md</code> im Projekt
refero	ja	Plugin bringt MCP-Konfiguration für Codex mit, Konto nötig
emil, instagram-agent, openseo, caveman, document-skills	nicht geprüft	in den READMEs steht nichts zu Codex

Gut zu wissen

Thema	Was du wissen musst
Konto nötig	openseo und refero brauchen ein eigenes Konto. Bei refero kostet die Live-Recherche.
Caveman Cloud	Die Skills caveman-setup, caveman-discover, caveman-manage, caveman-optimize und caveman-learn hängen an einem Dienst des Herstellers. Nur nutzen, wenn du es bewusst willst.
Doppelungen	Workflow: agent-skills oder superpowers, nicht beides gleichzeitig. Kürzen: ponytail oder caveman. Design: höchstens zwei Pakete.
Kosten	Jedes Plugin belegt Kontext zu Beginn jeder Sitzung. Abschalten ohne Löschen: <code>claude plugin disable name@marktplatz</code>
Auto-Modus	Claude Code entscheidet bei vielen Rückfragen selbst. Ändern mit <code>/permissions</code> .
Scraper rechtlich	Google Maps zu scrapen verstößt gegen die Google-Bedingungen. Kaltakquise per Mail braucht in Österreich vorherige Einwilligung (§ 174 TKG). Kein Rechtsrat.
Remotion Lizenz	Für kleine Firmen soweit bekannt kostenlos, größere Teams brauchen eine Lizenz. Vor Kundenvideos auf remotion.dev prüfen.
Neustart	Nach jeder Installation Claude Code neu starten, sonst fehlen die neuen Skills.

Fertige Abläufe für deine Arbeit

Neue Dashboard-Funktion

1. `/spec` mit einem Satz zur Funktion
2. `/plan`
3. `/build`
4. `/review`
5. `/ship`

Landingpage für einen Handwerker-Kunden

1. `/copywriting` für Text und Aufbau
2. `/impeccable craft` oder `/frontend-design` für die Gestaltung
3. `/cro` zur Prüfung der Anfrage-Strecke
4. `/webperf` für Ladezeit
5. `/ship` vor dem Livegang

Social-Video aus einer Idee

1. `/ig-reel` für Hook und Skript
2. `/remotion-create` mit dem Skript, Format 9:16
3. `/remotion-captions` für Untertitel
4. `/remotion-render` für die MP4
5. `/ig-caption` für den Begleittext

Akquise vorbereiten

1. `/scrape-batch` für die Betriebsliste
2. `/prospecting` zum Bewerten der Liste
3. `/ig-dm` für die Erstnachricht
4. Vor jedem Kontakt die rechtliche Grundlage prüfen

Empfehlung nach einer Testwoche

Behalte als Kern: agent-skills, ponytail, marketing-skills, document-skills, instagram-agent, remotion, chrome-devtools-mcp und ein bis zwei Design-Pakete.

Abschalten, was du nicht brauchst:

```
claude plugin disable taste-skill@taste-skill
```

Als Nächstes lohnt sich eine CLAUDE.md pro Kundenprojekt mit deinen Regeln (Landingpage-Bauplan, Design-System, Messaging). Das hebt die Ergebnisse stärker als jedes weitere Plugin.