

## CLAUDE CODE

# Best Practices abholen

Was da drin liegt und wie du es nutzt, ohne dass du es lesen musst

**83 Tipps**

an einem Ort

**65.000+**

Sterne auf GitHub

**0 Seiten**

musst du lesen

## DER LINK

<https://github.com/shanraishan/claude-code-best-practice>

**Die Sammlung heißt claude-code-best-practice und da liegt einfach alles an einem Ort, was gerade mit Claude Code funktioniert.**

Also Subagents, Commands, Skills, Hooks, MCP-Server und Workflows, und zu jedem Punkt gibt es die passenden Beispiele dazu. Das kostet nichts und steht unter MIT-Lizenz. Die Sternzahl habe ich am 27.08.2026 über die GitHub-API abgefragt.

## WAS DRIN STECKT

# Die Bausteine

Zu jedem Punkt gibt es die Erklärung und einmal die fertige Umsetzung

## 01 Die Grundlagen

Subagents, Commands, Skills, Workflows, Hooks, MCP-Server, Plugins, Settings, Status Line, Memory, Checkpointing, Sessions und Context Window. Zu jedem Punkt liegt eine Best-Practice-Seite im Repo und dazu einmal die fertige Umsetzung, die du dir anschauen kannst.

## 02 Die neueren Sachen

In der Hot-Tabelle stehen die neueren Sachen, also Agent Teams, Auto Mode, Fast Mode, Code Review, Computer Use, Agent SDK, GitHub Actions, Scheduled Tasks, Routines und Voice Dictation.

## 03 83 Tipps und Tricks

Die sind sortiert nach Prompting, Planning, Context, Sessions, CLAUDE.md, Agents, Commands, Skills, Hooks, Workflows, Git und Debugging. Jeder Tipp kommt aus dem Claude-Team oder aus der Community und ist verlinkt, du kannst also immer nachschauen, wo er herkommt.

## 04 Tipps von Boris Cherny

Boris Cherny wird im Repo als Creator of Claude Code geführt, und seine Posts sind da einzeln eingebunden. Er zeigt darin halt einfach, wie er selber mit Claude Code arbeitet.

## 05 Fertige Sammlungen

Dazu kommen noch kuratierte Skill-Sammlungen und Agent-Sammlungen von anderen Entwicklern und von Anthropic. Und es liegt ein kompletter Orchestration-Workflow drin, den du dir als Vorlage nehmen kannst.

## SCHRITT FÜR SCHRITT

# In drei Schritten nutzen

Du liest das nicht durch, du legst es Claude einfach hin

## 01 Repo runterladen

Du gehst im Repo oben rechts auf Code und dann auf Download ZIP, oder du klonst es dir im Terminal. Beides kostet nichts und du brauchst auch keinen Account dafür.

## 02 Neben dein Projekt legen

Den Ordner legst du einfach neben dein Projekt oder ziehst ihn direkt in dein Projekt rein. Wichtig ist nur, dass Claude Code ihn von deinem Arbeitsverzeichnis aus lesen kann.

## 03 Claude sagen, was du bauen willst

Du startest Claude Code in deinem Projekt und sagst ihm in einem Satz, wo die Sammlung liegt und was du bauen willst. Er sucht sich dann die passenden Stellen raus und setzt das für dich um.

## SO KLONST DU ES

```
git clone https://github.com/shanraishan/claude-code-best-practice
cd dein-projekt
claude
```

## BEISPIEL

# Der Prompt, den du reingibst

Du kopierst ihn dir einmal und passt den Pfad und dein Ziel an

## PROMPT ZUM KOPIEREN

```
Im Ordner ../claude-code-best-practice liegt eine Sammlung zu Claude Code. Lies dir daraus die Stellen zu Subagents, Skills und CLAUDE.md durch und bau mir für dieses Projekt einen Subagent, der meine Kundenmails in meinem Ton beantwortet. Halt dich an die Best Practices aus der Sammlung und sag mir vorher, welche Dateien du anlegst.
```

**Der Trick ist einfach, dass du Claude den Pfad nennst und ihm dann sagst, was du haben willst.**

Er liest sich dann nur die Stellen raus, die er dafür braucht, und du musst dich vorher nicht durch die ganze Sammlung arbeiten. Und wenn du was anderes bauen willst, tauschst du halt einfach den letzten Teil vom Prompt aus.

## ÜBERSICHT

# Was **wo** liegt

Die Ablageorte, die in der Sammlung stehen

| ORT                                | WOFÜR                                                                     |
|------------------------------------|---------------------------------------------------------------------------|
| <code>.claude/agents/</code>       | Subagents, also spezialisierte Agents für einzelne Aufgaben               |
| <code>.claude/commands/</code>     | Eigene Slash-Commands, die du selber aufrufst                             |
| <code>.claude/skills/</code>       | Skills als SKILL.md, die Claude bei Bedarf selber lädt                    |
| <code>.claude/hooks/</code>        | Hooks, die bei bestimmten Ereignissen automatisch laufen                  |
| <code>.claude/settings.json</code> | Einstellungen, Rechte, Modellwahl und Status Line                         |
| <code>.mcp.json</code>             | MCP-Server, also die Anbindung an externe Tools                           |
| <code>CLAUDE.md</code>             | Das Gedächtnis fürs Projekt, Regeln liegen in <code>.claude/rules/</code> |

**Im Repo steht selber drin, dass man das als Kurs lesen soll und nicht als fertigen Workflow.**

Es liegt aber ein kompletter Beispiel-Workflow drin, der heißt weather-orchestrator und spielt einmal den ganzen Weg von Command über Agent bis Skill durch. Den kannst du dir einfach nehmen und daraus deinen eigenen Ablauf bauen.

## NÄCHSTER SCHRITT

# Direkt umsetzen

Zehn Minuten, dann hast du es liegen

## 01 Repo holen

Du machst den Link auf, gehst auf Code und lädst dir das ZIP runter oder klonst es. Danach legst du den Ordner neben dein aktuelles Projekt.

## 02 Eine Sache aussuchen

Nimm dir einfach eine Aufgabe, die du sowieso ständig machst, und lass Claude daraus einen Subagent oder einen Command bauen.

## 03 CLAUDE.md aufräumen lassen

Zeig Claude die Tipps-Sektion aus der Sammlung und lass ihn deine CLAUDE.md danach umbauen. Das ist der Unterschied, den du am schnellsten merkst.

## Du willst solche Sachen sehen, bevor sie überall rumgehen?

Dann folg mir einfach auf Instagram, da kommen neue Claude-Sachen und KI-Tools rein, sobald sie draußen sind.

→ [@timhuck.ai](#) auf Instagram

→ [www.shortmedia.agency](https://www.shortmedia.agency)

Quelle: [github.com/shanraishan/claude-code-best-practice](https://github.com/shanraishan/claude-code-best-practice) (README und GitHub-API, abgerufen am 27.08.2026). Die Inhalte und die Sternzahl können sich natürlich ändern, schau im Zweifel einfach direkt ins Repo.