

VOM NICHT-PROGRAMMIERER ZUM BUILDER

Kurzreferenz-Spickzettel

DU + KI = BUILDER | Du: Richtung & Urteil | KI: Code & Tempo

DIE SCHLEIFE

BESCHREIBEN → ERHALTEN → AUSFÜHREN → BEWERTEN

1. Beschreibe, was du willst, in einfachen Worten
2. Erhalte Code von KI (keine Sorge, wenn du es nicht verstehst)
3. Führe es aus → 4. Bewerte Ergebnisse → Wiederhole bis es funktioniert

ZERLEGUNG

GROSSE AUFGABE → KLEINE AUFGABEN

Regel: Wenn es mehr als einen Satz braucht, teile es auf
Jedes Teil sollte unabhängig testbar sein
Baue ein Teil nach dem anderen, überprüfe, dann weiter

EFFEKTIVE PROMPTS

Schlecht: "Mach es fertig"

Gut: Diese 5 Elemente einschließen:

- WAS: "Erstelle eine Funktion, die berechnet..."
- WO: "In Python, zur bestehenden Datei hinzufügen..."
- WIE: "Benutze die Formel: Spannung = Kraft / Fläche"
- WARUM: "Das wird für Sicherheitsprüfungen verwendet"

FEEDBACK GEBEN

ERHALTEN X, ERWARTET Y, ÄNDERE Z

- "Ich habe [tatsächliches Ergebnis] bekommen"
- "Ich habe [was du wolltest] erwartet"
- "Bitte ändere [spezifische Sache]"

Fehlermeldungen, Screenshots, Beispiele einschließen

FEHLER DEBUGGEN

Syntaxfehler: Tippfehler - prüfe Rechtschreibung, Klammern, D

Namensfehler: Variable existiert nicht - prüfe Rechtschreibung

Typfehler: Falscher Datentyp - String vs Zahl

Indexfehler: Listenposition existiert nicht

Komplette Fehlermeldung kopieren → An KI einfügen → Nach Lö

DIE 5-VERSUCH-REGEL

NACH 5 VERSUCHEN FESTSTECKEN? NEU ANFANGEN

1. Neue Konversation starten (Kontext löschen)
2. Das Problem anders formulieren
3. In kleinere Teile aufteilen

Versunkene Kosten sind real - Verluste früh begrenzen

CODE LESEN

Variablen: Benannte Kisten, die Werte speichern (x = 5)

Funktionen: Wiederverwendbare Rezepte (def calculate():)

Bedingungen: Entscheidungen (if/else)

Schleifen: Aktionen wiederholen (for, while)

Listen: Sammlungen ([1, 2, 3]) - Dicts: Schlüssel-Wert-Paare

KONTEXT TEILEN

KI WEISS NUR, WAS DU IHR SAGST

Immer einschließen:

- Dein bestehender Code (einfügen)
- Vollständige Fehlermeldungen (exakt kopieren)
- Was du bereits versucht hast

PROJEKTSTRUKTUR

```
my_project/  
├── main.py  
├── requirements.txt  
├── data/  
└── tests/
```

KI-VERTRAUENSSTUFEN

Hohes Vertrauen: Standardcode

Mittel: Logik, Algorithmen

Überprüfen: Geschäftslogik, Grenzfälle

Immer testen: Sicherheit, Daten

WENN FESTSTECKEN

1. KI mit vollem Kontext fragen
2. Fehler online suchen
3. Die Frage umformulieren
4. In kleinere Teile aufteilen

Klares Ziel + Kleine Schritte + Iteration + Ausdauer = Funktionierender Code

SCHNELLE PYTHON-BEFEHLE

python script.py pip install package pip freeze > requirements.txt python -m venv venv

ZAUBERSPRÜCHE FÜR KI

"Erkläre es mir als Anfänger"

"Was bedeutet dieser Fehler?"

"Zeig mir Schritt für Schritt"

"Was fehlt mir?"

Du musst nicht alles wissen, um etwas Nützliches zu bauen. Fang an zu bauen!