

pythonTM

Wer von euch hat schon einmal
eine Deadline verpasst, weil ihr
sie vergessen habt?

Studien zeigen: 70% der Studierenden verlieren Aufgaben aus den Augen, wenn sie mehrere parallel haben.

Wir bieten euch die Lösung...

To-Do-List Manager mit Python

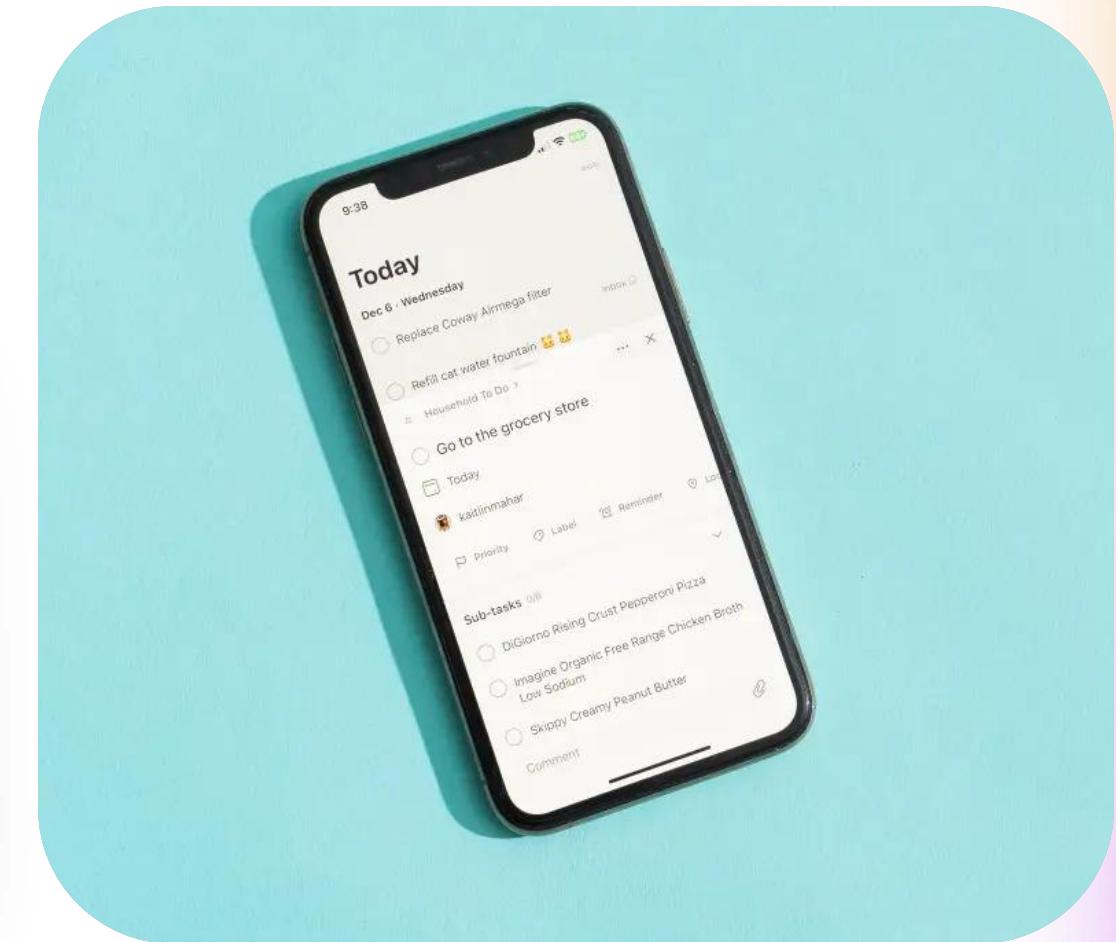
Wir wollten verstehen, was ein wirklich «*hilfreicher*» To-Do-Manager braucht
und haben unseren eigenen entwickelt.

Inhaltsverzeichniss

- Begründung des gewählten Themas
- Projektziele und -funktionen
- Projektmanagement
- Highlights und Herausforderungen
- Live-Demo der App-Funktionen
- Fragerunde

Warum ein Todo Manager?

- **Praxisbezug**
- **Umsetzbarkeit:**
z.B. mit Listen, Dictionaries und Dateioperationen.
- **Erweiterbarkeit:**
z.B. durch Sortierung, Prioritäten oder einfache Benutzeroberflächen.



Projektziele und -funktionen

1. Aufgaben können **hinzugefügt, entfernt, geändert und als erledigt markiert** werden.
2. Aufgaben werden in einer **Datei gespeichert** und beim Neustart wieder geladen.
3. Aufgaben können nach **Datum, Status oder Priorität gefiltert** werden.
4. Aufgaben können mit einer **Sortierfunktion** sortiert werden und/oder mit einer **Textsuche** gefunden werden.
5. Beim **Start des Programms** werden schon **3 Beispiel als Todo's** gezeigt.

Funktionen

1. Start der Datei – in VS –Code
2. CSV wird generiert
3. Beispieldateien werden geladen
4. Der Todo-Manager ist einsatzbereit

```
# Menü Anzeige
def show_menu() -> None:
    print(
        """
=====
===== To-Do Manager (CSV) =====
1) Aufgaben anzeigen
2) Aufgabe hinzufügen
3) Aufgabe bearbeiten
4) Aufgabe als erledigt/unerledigt markieren
5) Aufgabe löschen
6) Filtern/Sortieren
7) Speicher zurücksetzen (Backup + Neu)
0) Beenden
"""
    )
```

Projektmanagement

Planung

Gemeinsame Planung der Datenstruktur und Programmarchitektur.

Arbeitsverteilung im Team

Wir haben die Arbeiten untereinander aufgeteilt

Matthias: task_menu / reset / input_validation

Joël: todo_manager_v1 / hilfsfunktionen

Gemeinsame Absprache

Regelmässiges absprechen und Code-Reviews

Highlights

- Das zusammen arbeiten im Team
- Benutzerfreundliche Bedienung
- Aufteilen des Codes in Dateien
- Möglichkeiten Python (Programme)

Herausforderung

- Problem: Wenn die CSV-Datei manuell verändert oder sonst ein Fehler passierte, gingen die gespeicherten Daten verloren
- Lösung: Implementierung einer Backup-Logik, die den aktuellen Stand dauerhaft in einer Datei sichert.

ZUSÄTZLICH:

Speicher
zurücksetzen
(Backup + Neu)

Name abfragen,
Persönlicher

Live-Demo

ID	Beschreibung	Prio	Fällig
1	README der Aufgabe lesen	high	2025-12-10
2	Funktionen add/edit/delete implementiere	medium	2025-12-13
3	Tests & Doku schreiben	low	2025-12-15

===== To-Do Manager (CSV) =====

- 1) Aufgaben anzeigen
- 2) Aufgabe hinzufügen
- 3) Aufgabe bearbeiten
- 4) Aufgabe als erledigt/unerledigt markieren
- 5) Aufgabe löschen
- 6) Filtern/Sortieren
- 7) Speicher zurücksetzen (Backup + Neu)
- 0) Beenden

Auswahl: █

Fragen?



Danke fürs Zuhören

Projekt made by Matthias and Joël

Ende