

Programmeringsuppgift: PyTrack – din Python-studie-TODO i terminalen

Bakgrund

Du ska bygga en enkel TODO-app i terminalen som hjälper dig att planera och följa upp ditt lärande i Python. Appen låter dig lägga till *koncept* (t.ex. **listor**, **tupler**, **dict**), och koppla *todos* till varje koncept (t.ex. ”läs noteringar”, ”gör 3 övningar”, ”skriv en mini-sammanfattning”). Du kan markera todos som klara, se status och få en liten ”studie-poäng” för motivation.

Syfte

- Träna **listor**, **tupler** och **dictionaries**.
- Repetera **variabler**, **if-satser** och **loopar**.
- Designa en enkel **datastruktur** och ett **menysystem** i terminalen.

Krav (MVP)

1. **Startdata:** Appen ska börja med några förifyllda koncept (minst 3) och minst en todo per koncept.
2. **Meny:** Programmet ska loopa tills användaren väljer att avsluta:
 - 1. Lista alla koncept
 - 2. Visa todos för valt koncept
 - 3. Lägg till nytt koncept
 - 4. Lägg till todo till ett koncept
 - 5. Markera todo som klar/inte klar
 - 6. Visa översikt/statistik

- 0. Avsluta

3. **Datastruktur:** Använd dict, list och tuple.

```
concepts: dict[str, list[tuple[str, bool]]]
# nyckel: koncept-namn, värde: lista av tupler (todo_text,
klar_flagga)
```

Listing 1: Förslag på datastruktur

4. **Felhantering:** Hantera ogiltiga menyval och indexfel utan att krascha.

5. **Uppdatering:** Ändringar ska synas direkt (ingen filhantering krävs).

Exempel på meny

```
=== PyTrack ===
1) Lista alla koncept
2) Visa todos för ett koncept
3) Lägg till nytt koncept
4) Lägg till todo till koncept
5) Markera todo klar/inte klar
6) Visa översikt/statistik
0) Avsluta
Val: _
```

Exempel på startdata

```
concepts = {
    "variabler": [("läs igenom lektionsanteckningar", True),
                  ("skriv 5 exempel i REPL", False)],
    "listor":    [("gör 3 övningar med slicing", False)],
    "dictionaries": [("skriv en fusklapp med metoder", False),
                     ("lös 2 uppgifter om nycklar/värden", False)]
}
```

Funktionella detaljer

- **Lista alla koncept:** skriv t.ex. "listor (2/5 klara)".

- **Visa todos:** numrera todos och visa status, t.ex. [] gör tre övningar, [x] läs dokumentation.
- **Lägg till koncept:** be om namn, undvik dubletter.
- **Lägg till todo:** välj koncept, skriv todo-text, status ska börja som False.
- **Markera todo:** välj koncept, välj todo-index, växla status.
- **Visa översikt/statistik:**
 - Totalt antal koncept.
 - Totalt antal todos och hur många som är klara.
 - Koncept med högst andel klara todos.

Extra utmaningar (valfria)

1. Lägg till prioritet (1–3) för todos.
2. Lägg till enkel deadline som sträng (t.ex. "2025-11-05").
3. Poängsystem (+10 poäng för varje klar todo).
4. Taggar (t.ex. #övning, #repetition) och menyval "Sök på tagg".
5. Spara/läs todos till/från textfil.
6. Lägg till historiklista med ändringar.
7. Menyval för slumpmässig oklar todo.

Bedömningskriterier

- Fungerande meny och loop.
- Korrekt användning av `list`, `tuple`, `dict`.
- Felhantering utan krascher.
- Begriplig kodstil och struktur.
- Bonus för extra utmaningar.

Tips

Använd gärna hjälpfunktioner som:

```
def print_menu(): ...
def list_concepts(concepts): ...
def list_todos(concepts, name): ...
def add_concept(concepts, name): ...
def add_todo(concepts, name, text): ...
def toggle_todo(concepts, name, index): ...
def stats(concepts): ...
```

Vanliga fallgropar

- Kontrollera att konceptet finns innan du visar/lägger till todos.
- Var försiktig med index (Python börjar på 0!).
- Tupler är oföränderliga – skapa en ny tuple vid statusändring.

Testfall

1. Starta programmet och välj **1** → alla koncept listas.
2. Välj **2**, skriv ett befintligt koncept → todos listas numrerade.
3. Välj **4**, lägg till todo → lista igen, ska synas.
4. Välj **5**, markera todo klar → status växlar.
5. Välj **6** → statistik visas.
6. Testa ogiltigt menyval → programmet kraschar inte.

Inlämning

- `pytrack.py` (+ ev. `data.txt` vid export/import)
- Körning: `python pytrack.py`
- README med kort beskrivning av datastruktur och extra utmaningar.

Lycka till och ha kul med **PyTrack**!