TECH STARTER

Python Einführung - Aufgabenliste für Büroprojekte

1. Einführung

Erstelle ein neues Repository auf GitHub mit dem Namen 'python-office-tasklist'. Arbeite in Feature Branches, um neue Funktionen zur Aufgabenliste hinzuzufügen, und dokumentiere deine Arbeit mit Kommentaren und einer README.md-Datei.

2. Feature 1: Funktion zum Hinzufügen von Aufgaben

Erstelle einen Feature Branch „feature/add-task-function“ und erweitere dein Python-Skript um die Funktion `add_task`.

1. Nutze die `input`-Funktion, um den User nach einer Büroaufgabe zu fragen, z.B.: „Bitte gib eine Aufgabe ein, die in deiner Aufgabenliste hinzugefügt werden soll“.
2. Speichere die Eingabe in der Variablen `task`.
3. Füge die Aufgabe zur Liste hinzu.
4. Gib eine Meldung aus, dass die Aufgabe zur Liste hinzugefügt wurde.

Tipp: Du kannst auch ein Fälligkeitsdatum (z.B. mit `input`) abfragen und dieses der Aufgabe hinzufügen.

3. Feature 2: Funktion zum Anzeigen der Aufgabenliste

Erstelle einen Feature Branch „feature/add-show-tasklist-function“ und erweitere dein Python-Skript um die Funktion `show_tasklist`:

1. Prüfe, ob die Liste leer ist. Wenn ja, gib den Text „Deine Aufgabenliste ist leer“ aus.
2. Wenn die Liste nicht leer ist, drucke alle Aufgaben mit einer `for`-Schleife.
3. Gib auch das Fälligkeitsdatum (falls vorhanden) mit aus.

Tipp: Du kannst Aufgaben farblich markieren, z.B. alle Aufgaben, die innerhalb von 2 Tagen fällig sind, in Rot.

TECH STARTER

4. Feature 3: Hauptprogramm erstellen

Erstelle einen Feature Branch „feature/add-main-function“ und erweitere dein Python-Skript um die Funktion `main`:

1. Erstelle eine `while`-Schleife, die den User fragt, ob er eine Aufgabe hinzufügen, die Liste anzeigen oder das Programm beenden möchte.
2. Nutze eine `if/elif`-Bedingung, um die Eingabe des Users zu verarbeiten.

Tipp: Füge eine Option zum Entfernen von Aufgaben hinzu, um erledigte Aufgaben von der Liste zu streichen.

5. Feature 4: Programm automatisch starten

Füge am Ende deines Skripts folgenden Code ein, um das Programm automatisch zu starten, wenn es ausgeführt wird:

```
if __name__ == "__main__":
    main()
```

6. Erweiterungen (optional)

1. **Priorisierung von Aufgaben:** Füge eine Prioritätsstufe (hoch, mittel, niedrig) hinzu und sortiere die Liste nach Priorität.
2. **Exportfunktion:** Implementiere eine Funktion, um die Liste in eine CSV-Datei zu exportieren.
3. **Benachrichtigungen:** Implementiere eine einfache Benachrichtigung, wenn eine Aufgabe fällig ist.

7. Debugging-Tipps

Wenn Fehler auftreten, nutze `print`-Anweisungen, um Variablen oder Ergebnisse zu überprüfen. Zum Beispiel könntest du nach jeder User-Eingabe die Variable ausgeben, um sicherzustellen, dass der Wert korrekt ist.

```
print("Funktion wird aufgerufen")
```