

Python Lernaufgaben – Nach Themen

Keine Lösungen enthalten. Erst selbst probieren – bei Bedarf gehen wir eine Aufgabe gemeinsam Schritt für Schritt durch.

1. Kleine Python-Programme

1. Schreibe ein Programm, das den Nutzer nach seinem Namen fragt und ihn begrüßt.

2. Schreibe ein Programm, das den Nutzer nach seinem Namen fragt und ihn begrüßt.
Ergebnis:

2. APIs

1. Nutze requests, um Daten von <https://jsonplaceholder.typicode.com/users> zu holen und die Namen aller Nutzer auszugeben.

2. Hole Daten von <https://jsonplaceholder.typicode.com/users> und gib die Namen aller Nutzer + Ort aus.

Setup: pip install requests

3. Daten verarbeiten

1. Gegeben eine Liste von Zahlen: berechne Summe, Durchschnitt, Minimum und Maximum – ohne die eingebauten Funktionen sum(), min(), max().

2. Gegeben eine Liste von Zahlen: berechne Summe, Durchschnitt, Minimum und Maximum – ohne die eingebauten Funktionen sum(), min(), max().
vorkommen:

Tipp für CSV: Modul csv oder pandas.

4. Dateien lesen/schreiben

1. Schreibe eine Funktion, die einen Text in eine Datei notizen.txt schreibt.

2. Schreibe eine Funktion, die einen Text in eine Datei notizen.txt schreibt.

5. Fehlerbehandlung

1. Schreibe ein Programm, das den Nutzer eine Zahl eingeben lässt. Fängt den Fehler ab, falls keine gültige Zahl eingegeben wird, und fragt erneut.

2. Baue
sinnvolle

6. Funktionen und Klassen

1. Schreibe eine Funktion `ist_palindrom(text)`, die prüft, ob ein Wort rückwärts gelesen gleich ist (z.B. "anna").

2. Erste
(aktuelle

7. Listen/Dictionaries

1. Entferne alle doppelten Werte aus einer Liste (ohne `set()` zu benutzen).

2. Drehe

8. Einfache Algorithmen

1. Implementiere Bubble Sort, um eine Liste von Zahlen zu sortieren.

2. Imple

9. Datenbanken

Setup: SQLite ist in Python bereits eingebaut (`import sqlite3`), kein separater Server nötig.

1. Erstelle eine SQLite-Datenbank mit einer Tabelle `personen` (`id`, `name`, `alter`).

2. Schre

10. Kleine Webanwendungen (FastAPI)

Setup: `pip install fastapi uvicorn`

1. Baue eine API mit einem Endpunkt `/hallo/{name}`, der `{"message": "Hallo, name!"}` zurückgibt.

2. Baue
erstmal

11. Debugging

1. Nimm eines deiner obigen Programme und baue absichtlich einen Fehler ein (z.B. falscher Variablenname)
– finde ihn anhand der Fehlermeldung im Terminal.

2. Nutze
Stelle im

Vorschlag für eine Lernreihenfolge

1. Kleine Python-Programme

2. Funkt

Hinweis

Bei "kle
Funktio
sitzt.

Sag mir, mit welchem Thema/welcher Aufgabe du starten willst – dann gehen wir sie zusammen Schritt für Schritt durch.