

(Die nachfolgenden Aufgaben wurden mit Hilfe von ChatGPT erstellt, überprüft und ggf. angepasst)

Übungsaufgaben zur 2. Vorlesung: for-Schleifen, Funktionen, positional und key-word Argumente, Funktionen in Variablen speichern, import

1. Aufgabenstellung: Summiere die Zahlen einer Liste

Gegeben ist eine Liste von Zahlen. Schreibe ein Python-Programm, das mit einer **for-Schleife** die **Summe aller Zahlen** berechnet und ausgibt.

Anforderungen:

- Definiere eine Liste mit Zahlen, zum Beispiel: zahlen = [3, 7, 1, 8, 4]
- Verwende eine **for-Schleife**, um durch die Liste zu gehen und die Zahlen aufzusummieren.
- Gib am Ende die Gesamtsumme aus.

Beispielausgabe:

Summe der Zahlen: 23

2. Aufgabenstellung: Ein kleines Einmaleins

Schreibe ein Python-Programm, das das kleine Einmaleins (1 bis 10) als Tabelle ausgibt.

Anforderungen:

- Verwende zwei verschachtelte for-Schleifen:
- Die äußere Schleife läuft über die Zahlen 1 bis 10 (Zeilen).
- Die innere Schleife läuft ebenfalls über 1 bis 10 (Spalten).
- Gib das Produkt von Zeilenzahl × Spaltenzahl aus.

3. Aufgabenstellung: Rechteck-Muster mit Rand

Schreibe ein Python-Programm, das ein **Rechteck aus Sternchen** zeichnet, bei dem **nur der Rand** aus Sternchen besteht und die Mitte leer ist (gefüllt mit Leerzeichen).

Anforderungen:

- Der Benutzer soll die **Breite** und **Höhe** des Rechtecks eingeben.
- Verwende zwei **verschachtelte for-Schleifen**:
- Eine Schleife für die Zeilen (Höhe),
- Eine Schleife für die Spalten (Breite).
- Drucke:
- Ein *, wenn du am Rand bist,

- Ein Leerzeichen , wenn du in der Mitte bist.

Beispiel bei Breite 8 und Höhe 5:

```

* * * * *
*           *
*           *
*           *
*           *
* * * * *

```

Hinweis: Am Rand bist du, wenn du in der **ersten oder letzten Zeile** bist oder in der **ersten oder letzten Spalte**.

4. Aufgabenstellung: Wiederholtes Spielen

Stell dir vor, du hast bereits mehrere kleine Spiele geschrieben (z.B. eine zufällige Zahl erraten lassen, ein Quiz, etc.). Diese Spiele sind als Funktionen geschrieben. Nun möchtest du für alle Spiele eine Funktion schreiben, die dem Benutzer die Möglichkeit gibt, mehrfach hintereinander zu spielen.

Nimm an, du hast eine Funktion `spiel()`, die das eigentliche Spiel ausführt (wie oben beispielhaft erwähnt: eine zufällige Zahl erraten lassen, ein Quiz, etc.). Du sollst nun eine weitere Funktion `wiederholtes_spielen()` erstellen, die das Spiel wiederholt, solange der Benutzer weiterhin spielen möchte.

Anforderungen:

1. Erstelle eine Funktion `wiederholtes_spielen(spiel)`, die:
 - Die Funktion `spiel()` ausführt.
 - Den Benutzer fragt, ob er nochmal spielen möchte (`input("Möchtest du nochmal spielen? (ja/nein)")`).
 - Falls der Benutzer "ja" eingibt, wird `spiel()` erneut aufgerufen.
 - Falls der Benutzer "nein" eingibt, gibt die Funktion "Tschüss" aus und beendet das Programm.

Die Funktion "spiel" selbst muss nicht implementiert werden. Wenn du dein Programm testen möchtest, kannst du etwas einfaches verwenden, z.B.

```
def errate_die_zahl():
```

```
    zahl = int(input("Errate die Zahl zwischen 1 und 10: "))
```

```
if zahl == 5:  
    print("Richtig!")  
else:  
    print("Falsch, die Zahl war 5.")
```

Rufe zum Test wiederholtes_spielen(errate_die_zahl) auf!

5. Aufgabenstellung: Aktionen auf einer Liste

Schreibe eine Funktion `verarbeite_liste()`, die eine andere Funktion als Eingabe erhält und diese auf jedes Element einer Liste anwendet.

Anforderungen:

1. **Erstelle eine Funktion** `verarbeite_liste(liste, funktion)`, die:
 - Eine Liste von Zahlen (z.B. [1, 2, 3, 4, 5]) als Eingabe erhält.
 - Eine Funktion (`funktion`), die eine Zahl entgegennimmt und etwas damit macht (z.B. verdoppeln, quadrieren) als zweites Eingabeargument erhält.
 - Die Funktion `funktion` wird auf jedes Element der Liste angewendet.
 - Gib die bearbeitete Liste zurück.
2. **Erstelle zwei Funktionen**, die du an `verarbeite_liste()` übergeben kannst:
 - Eine Funktion `verdoppeln(x)`, die eine Zahl verdoppelt.
 - Eine Funktion `quadrieren(x)`, die eine Zahl quadriert.
3. **Teste die `verarbeite_liste()` Funktion** mit der Liste [1, 2, 3, 4, 5] und den beiden Funktionen `verdoppeln` und `quadrieren`. Gib die Ergebnisse aus.