

AUFGABEN SESSION 4:

AUFGABE 1: LISTE

Erzeuge eine Liste, die drei int-Werte speichern kann. Fülle diese mit Zahlen und gib die Werte in die Konsole aus.

Hinweis: Versuche nicht die ganze Liste auf einmal auszugeben sondern die einzelnen Werte (siehe `range()` Funktion), aber so, dass du das Programm nur einmal ausführen musst.

AUFGABE 2: TUPEL

Nutze für diese Aufgabe deine Liste aus Aufgabe 1. Es soll nun zuerst ein leeres Tupel erstellt werden, um dann die Werte der Liste in das Tupel zu kopieren. Prüfe, ob auch wirklich die Liste eine Liste und das Tupel ein Tupel ist, indem du beide jeweils komplett ausgibst. Die Werte in der Liste sollten von eckigen Klammern `[]` und die des Tupels von runden Klammern `()` umschlossen sein

Hinweis: informiere dich über das Schlüsselwort „tupel“ und dessen Verwendung. Vielleicht kannst du es nutzen!

AUFGABE 3: KONKATENATION

Erstelle einen Code Teil1 und Code Teil2 (jeweils in einer Variablen gespeichert) in der kurzmöglichsten Variante.

Gib diese zwei Codes zusammengefügt aus.

Code_Teil1: kurs190bbbbbbbfunky1pythhhhhh

Code_Teil2: sssss12abcabcabcabcd

AUFGABE 4: UMKEHRUNG

Erzeuge eine Liste für 3 Zahlen. Lass deinen Benutzer die Zahlen eingeben. Gib die Zahlen in umgekehrter Reihenfolge in der Konsole aus.

Hinweis: Informiere dich darüber, mit welcher Funktion eine Liste umgedreht werden kann.

AUFGABE 5: LÖSCHEN UND ERWEITERN



Erzeuge eine Liste mit den Werten 20,30,40,50. Erweitere die Liste mit einer beliebigen Zahl. Lösche anschließend die Zahl 30 und gebe die Werte der Liste aus

Tipp: Nutze zum Erweitern die Methode `append()`, zum Löschen entweder die Methode `pop()` oder `remove()`.

AUFGABE 6: STRINGS



Erstelle parallel zu dem Beispiel aus dem Kurs einen zwei Tupel Wochentag und Wochentag Kürzel. Befülle beide mit Strings wie Montag, Dienstag, und MO, DI,... Gebe am Ende folgendes mit Hilfe des Formatstrings aus:

Mittwoch = MI

AUFGABE 7: NESTED LISTS - AUFLÖSEN



Löse folgende verschachtelte Liste so auf, dass sie nunmehr als einfache Liste angezeigt wird. Lass die Liste anschließend in der Konsole ausgeben. (Tipp: nutze die „extend“ Funktion)

```
l1 = [[111, 49, 77], [21, 22, 23], [50, 65, 3]]
```

AUFGABE 8: COPY/PASTE



Erzeuge eine Liste mit [1, 2, 3, 4, 5]. Erzeuge eine zweite Liste, ebenfalls mit 5 Stellen und kopiere die Daten aus der ersten in die zweite Liste. Außer einer for-Schleife und dem Kopieren der Daten ist nichts erlaubt. Gib die Werte, welche sich nun in der Kopie befinden mithilfe einer weiteren for-Schleife aus.

Tipp: wenn du am Anfang die zweite Liste mit „0“ befüllst, kannst du nachprüfen, ob auch wirklich jeder Wert einzeln von der ersten Liste in die zweite kopiert wird, da die noch nicht kopierten Stellen noch „0“ sein müssten. Dazu musst du nach dem Kopieren innerhalb der for-Schleife die ganze zweite Liste ausgeben.

AUFGABE 9: NESTED LISTS - DURCHLAUF



Lass auf dem Bildschirm die numerischen Listen zeilenweise ausgeben und trenne die Zahlen jeweils mit einem Leerzeichen. Versuche dies ohne die „range“ Funktion.

```
l2 = [[1, 2, 3, 4], [5, 6], [7, 8, 9]]
```

AUFGABE 10: NESTED LISTS - AUFSUMMIEREN



Berechne die Summe aller Elemente der in Aufgabe 6 genannten verschachtelten Liste mit Hilfe von verschachtelten for-Schleifen. **(Mit und ohne „range“ Funktion)**

AUFGABE 11: GRÖßTE UND KLEINSTE ZAHL



Gegeben ist folgende Liste: zahlenliste = [78,12,85,39,39,25,93,25,10,18] Es soll nun eine Kontrollflussanweisung geschrieben werden, welche den größten sowie den kleinsten Wert der Liste ermittelt und in der Konsole ausgibt.

AUFGABE 12: PYRAMIDE



Gib eine Zahlenpyramide in die Konsole aus:

1

22

333

4444

55555

.....n

Lass deinen Benutzer die Anzahl der Reihen bestimmen!