

Übungsblatt 6: Python

VORBESPRECHUNG: 31.03.2023

ABGABE: 21.04.2023

Alle Aufgaben sind vollständig zu lösen.

Geben Sie die Übung bitte in Gruppen von 3 oder 4 Personen ab. Erstellen sie dazu ein Team auf ILIAS. Für die Abgabe der Übung selbst laden Sie bitte die Python-Dateien (*.py) hoch. Geben Sie die schriftlichen Aufgaben bitte in Form einer PDF- oder Textdatei (*.pdf oder *.txt) ab. Diese Abgabe wird dann für alle Gruppenmitglieder übernommen.

1 VARIABLEN UND LISTEN

Aufgabe 1

Betrachten Sie den untenstehenden Python-Code und geben Sie an, welche Werte in den Variablen a, b, c, d, e und f gespeichert sind, wenn der Code vollständig durchlaufen wurde. Erwähnen Sie jeweils auch den Datentyp der jeweiligen Variable. Versuchen Sie die Aufgabe ohne einen Computer zu lösen.

```
1 a, b = 2, 2
2 c = 4
3
4 d = a + b * c
5 c *= c % a
6 e = d / a + c
7 f = e // b
```

Aufgabe 2

Schreiben Sie ein Programm, welches vom Benutzer die maximal möglichen und die tatsächlich erreichten Punkte einer Prüfung einliest (m und e). Danach berechnen Sie anhand der Formel

$$Note = \frac{e}{m} \cdot 5 + 1$$

die *Note* und geben diese auf zwei Nachkommastellen gerunden aus (hierzu können Sie die eingebaute Funktion `round` verwenden).

Aufgabe 3

Gegeben sei eine Liste `numbers`, welche ganze Zahlen enthält. Schreiben Sie je ein Code-Fragment, das Folgendes erledigt:

- (a) Fügen Sie an das Ende der Liste `numbers` den Wert 17 hinzu.
- (b) Fügen Sie der Liste `numbers` den Wert 99 an Position 0 hinzu.
- (c) Addieren Sie zum Element an Position 4 in `numbers` den Wert 18.
- (d) Lesen Sie das Element an Position 3 aus `numbers` aus und weisen Sie dieses der Variable `value` zu.
- (e) Geben Sie die Grösse der Liste `numbers` aus.
- (f) Ersetzen Sie das Element an Position 1 mit dem Wert 17.

Aufgabe 4

Betrachten Sie die Liste `values` in folgender Abbildung:

values	-1	99	7	7	4	3	4	5	8
	0	1	2	3	4	5	6	7	8

Was ist:

- (a) `values[1]`
- (b) `values[2] + values[5]`
- (c) `values[2+5]`
- (d) `len(values)`
- (e) `values[len(values)-1]`
- (f) `values[1:3]`
- (g) `values[:4]`
- (h) `values[2:]`
- (i) `values[:2]`
- (j) `values[-1:1:-1]`