

Übungsblatt_03_Python_v1_1_2

```
1 # -----
2 # ---- Skriptsprachenorientierte Programmiertechnik
3 # ---- Übung #3 (Sequenzielle Datentypen) - Lösungen - v1.1.2
4 # ----
5 # ---- Beispiellösungen zu Übungsaufgaben; die Lösungen orientieren sich
6 # ---- an bereits besprochenen Sprachkonstrukten; andere (effizientere)
7 # ---- Lösungen sind ggf. möglich. Fragen und Anmerkungen bitte direkt an mich
8 # ----
9 # ----
10 # ---- Die doppelten Kommentarzeichen können über eine Selektion des Bereiches
11 # ---- und Alt + 4 entfernt werden (erneute Einkommentierung mit Alt + 3)
12 # -----
13
14 text_1 = "Hallo Welt!"
15 text_2 = "FOM"
16 tupel = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10
17 liste_1 = list(tupel)
18 liste_2 = [6, 7, 8, 9, 10, 11, 12, 14, 14, 15]
19 zahl_1 = 6
20 zahl_2 = 12
21
22 aufgabennummer = 0
23
24
25 def aufgabe(t=''):
26     global aufgabennummer
27     aufgabennummer += 1
28     print("\n----- Aufgabe", aufgabennummer, t, "-----")
29
30
31 # 1) Geben Sie aus, wie viele Elemente in text_1 und tupel enthalten sind.
32 aufgabe()
33
34 print("Elemente in text_1: ", len(text_1))
35 print("Elemente in folge_tupel: ", len(tupel))
36
37 # 2) Geben Sie jeweils das erste u. letzte Element von text_1, und tupel aus.
38 aufgabe()
39
40 print("das erste / letzte Element von")
41 print("- text_1: ", text_1[0], "/", text_1[-1])
42 print("- tupel: ", tupel[0], "/", tupel[-1])
43 # alternativ (zwei Elemente mit einem Zugriff)
44 print("- text_1: ", text_1[::len(text_1)-1])
45
46
47 # 3) Geben Sie die drei ersten und die drei letzten Elemente
48 # von text_1 und tupel aus.
49 aufgabe()
50
51 print("die ersten / letzten drei Element von")
52 print("- text_1: ", text_1[0:3], "/", text_1[-3:])
53 print("- tupel: ", tupel[:3], "/", tupel[-3:])
54
55 # 4) Geben Sie jedes zweite Element von liste_1 aus.
56 aufgabe()
57
58 print("jedes 2. Element von liste_1", liste_1[::2])
59 # bzw. (Startelement in Aufgabe nicht eindeutig definiert)
60 print("jedes 2. Element von liste_1", liste_1[1::2])
61
```

Übungsblatt_03_Python_v1_1_2

```
62
63 # 5) Geben Sie das größte Element von text_1 u. das kleinste von liste_2 aus.
64 aufgabe()
65
66 print("kleinstes Element in text_1: ", max(text_1))
67 print("kleinstes Element in liste_2: ", min(liste_2))
68
69 # 6) Fügen Sie die Elemente aus liste_1 zweimal an die ursprünglich liste_1 an
70 # und geben Sie die liste über print(liste_1) aus.
71 aufgabe()
72
73 liste_1 *= 3
74 print(liste_1)
75
76 # 7) Generieren Sie eine neue Liste (gesamtliste) mit allen Elementen aus
77 # tupel, liste_1 und liste_2. Geben Sie die Gesamtliste sowie die Anzahl
78 # der Elemente der Gesamtliste aus.
79 # Tipp: erzeugen Sie zunächst die Verbindung aus den beiden Listen liste_1 und
80 # liste_2 und fügen Sie anschließend die Elemente aus tupel hinzu
81 aufgabe()
82
83 gesamtliste = liste_1 + liste_2
84 gesamtliste += tupel
85
86 # Alternative
87 gesamtliste = liste_1 + liste_2 + list(tupel)
88 print(gesamtliste)
89
90 # 8) Geben Sie an, ob zahl_1 in der Gesamtliste vorkommt oder nicht
91 aufgabe()
92
93 if zahl_1 in gesamtliste:
94     print(zahl_1, "kommt in der Gesamtliste vor")
95 else:
96     print(zahl_1, "kommt nicht in der Gesamtliste vor")
97
98 # Kurzfassung
99 print(zahl_1, "kommt vor" if zahl_1 in gesamtliste else "kommt nicht vor")
100
101
102 # 9) Geben Sie die Position des ersten Auftretens von zahl_1 in der
103 # Gesamtliste an.
104 # Die Position soll dabei der tatsächlichen Position entsprechen (das erste
105 # Element der Liste hat die Position 1). Falls die zahl_1 nicht in der
106 # Gesamtliste vorkommt, soll eine entsprechende Meldung ausgegeben werden.
107 aufgabe()
108
109 # Lösung mit conditional expression
110 print(gesamtliste.index(zahl_1) + 1 if zahl_1 in gesamtliste
111       else "Zahl tritt nicht auf")
112
113 # Alternative über if/else-Konstrukt
114 if zahl_1 in gesamtliste:
115     print("erstes Auftreten von", zahl_1, "an Position",
116           gesamtliste.index(zahl_1) + 1)
117 else:
118     print("Die Zahl", zahl_1, "tritt nicht in der Gesamtliste auf.")
119
120
121 # 10) Geben Sie an, wie oft zahl_1 in der Gesamtliste vorkommt
122 aufgabe()
```

```

123
124 print("Die Zahl", zahl_1, "kommt", gesamtlste.count(zahl_1), " Mal vor")
125
126
127 # 11) Geben Sie den Index für jedes Element in Gesamtliste an, an der der Wert
128 # von zahl_1 steht.
129 aufgabe()
130
131 index = 0
132 for zahl in gesamtlste:
133     if zahl == zahl_1:
134         print("gefunden an Pos.", index)
135     index += 1
136
137 # Alternative mit index-Methode, while und try/except (effizientere
138 # Bearbeitung; bei einer gesamtlste mit vielen Elementen zu bevorzugen)
139 print("---- Alternative mit while & Exception")
140
141 position_index = -1
142 try:
143     while True:
144         position_index = gesamtlste.index(zahl_1, position_index+1)
145         print("gefunden an Pos.", position_index)
146 except ValueError:
147     pass
148
149
150 # 12) Erstellen Sie eine neue Variable (text) deren Inhalt sich aus dem ersten
151 # Wort von text_1, dem gesamten Inhalt von text_2 sowie einem abschließenden
152 # „!“ zusammensetzt. Geben Sie die Variable über print(text) aus.
153 aufgabe()
154
155 text = text_1[:text_1.index(' ')+1] + text_2 + '!'
156 print(text)
157
158 # Alternative: besser mit Abfrage, ob ein Leerzeichen in text_1 enthalten ist
159 if ' ' in text_1:
160     text = text_1[:text_1.index(' ')+1] + text_2 + '!'
161 else:
162     text = text_1 + ' ' + text_2 + '!'
163 print(text)
164

```