

```

1  # ----- Übungsblatt 2 -----
2
3  aufgabennummer = 0
4
5
6  def aufgabe(text=''):
7      global aufgabennummer
8      aufgabennummer += 1
9      print("\n----- Aufgabe", aufgabennummer, text, "-----")
10
11
12  # ----- Aufgabe 1 -----
13  aufgabe()
14
15  i = 28
16  f = 28.0
17  print("i=", i, "; f=", f, sep='')
18  print("id von i:", id(i))
19  print("id von f:", id(f))
20  print("Typ von i:", type(i))
21  print("Typ von f:", type(f))
22  # Dynamische Typisierung
23
24  # ----- Aufgabe 2 -----
25  aufgabe()
26
27  i = 28.0
28  print("id von i:", id(i))
29  print("Typ von i:", type(i))
30  # ID hat sich geändert
31  # Typ hat sich geändert => dynamische Typisierung
32
33
34  # ----- Aufgabe 3 -----
35  aufgabe()
36
37  s = "Hallo"
38  s2 = s
39  print("s = ", s, " (ID: ", id(s), ")", sep='')
40  print("s2 = ", s2, " (ID: ", id(s2), ")", sep='')
41  s += " Welt!"
42  print("s = ", s, " (ID: ", id(s), ")", sep='')
43  print("s2 = ", s2, " (ID: ", id(s2), ")", sep='')
44  # ID und Inhalt von s hat sich geändert; s2 bleibt davon unberührt
45  # => Datentyp ist immutable (Wert der Instanz ist nicht veränderbar)
46
47  # ----- Aufgabe 4 -----
48  aufgabe()
49
50  m = ['a', 'b', 'd', 'e', 'f']
51  print("m = ", m, " (ID: ", id(m), ")", sep='')
52  m[0] = 'A'
53  print("m = ", m, " (ID: ", id(m), ")", sep='')
54  # Wert von m hat sich geändert, während die ID gleich geblieben ist
55  # => Datentyp ist mutable (Wert der Instanz ist veränderbar)
56
57
58  # ----- Aufgabe 5 -----
59  aufgabe()
60
61  m2 = m
62  print("m2 = ", m2, " (ID: ", id(m2), ")", sep='')
63  m += ['g']
64  print("m = ", m, " (ID: ", id(m), ")", sep='')
65  print("m2 = ", m2, " (ID: ", id(m2), ")", sep='')
66  # der Inhalt von m UND m2 hat sich geändert; die ID ist unverändert
67  # Seiteneffekt!
68
69
70  # ----- Aufgabe 6 -----
71  aufgabe()

```

```

72
73 t = (1, 2, [31, 32], 4)
74 t2 = t
75 print("t = ", t, " (ID: ", id(t), ")", sep='')
76 print("t2 = ", t2, " (ID: ", id(t2), ")", sep='')
77
78 try:
79     t[0] = 'X'
80 except TypeError:
81     print("t[0] = 'X' <-- geht nicht")
82
83 # Änderung eines direkten Wertes innerhalb des Tupels nicht möglich.
84 # (immutable Datentyp; Wert der Instanz nicht änderbar)
85
86 t[2][0] = 'X'
87 # Änderung eines Wertes innerhalb einer Liste im Tupel möglich.
88 # (mutable Datentyp; Wert der Instanz änderbar)
89 print("t = ", t, " (ID: ", id(t), ")", sep='')
90 print("t2 = ", t2, " (ID: ", id(t2), ")", sep='')
91 # Die IDs bleiben bei der Änderung gleich (die ID des Elementes
92 # ändert sich hingegen; Test über Ausgabe von id(t[2][0]) vor und nach
93 # der Änderung möglich
94
95 # ----- Aufgabe 7 -----
96 aufgabe()
97
98 t += 5,
99 print("t = ", t, " (ID: ", id(t), ")", sep='')
100 print("t2 = ", t2, " (ID: ", id(t2), ")", sep='')
101 # t hat sich geändert, während t2 unverändert ist: kein Seiteneffekt!
102 # ID von t hat sich geändert; ID von t2 ist unverändert
103 # (Versuch das Tuple zu erweitern führt zur Erzeugung eines neuen Tupels
104 # mit kopiertem Inhalt und angehängter Erweiterung)
105
106 # ----- Aufgabe 8 -----
107 aufgabe()
108
109 del t
110
111 try:
112     print(t)
113 except NameError:
114     print("print(t) <-- geht nicht")
115
116 # t ist nicht mehr vorhanden; Zugriff erzeugt Fehler
117 print(t2)
118 del t2
119 # Das Objekt (die Instanz), auf das t2 referenziert hat, wird freigegeben.
120 # Der Garbage Collector kann den Speicherplatz nun wieder freigeben.
121
122
123 # ----- Aufgabe 9 -----
124
125 x = 5
126 y = '5'
127 try:
128     z = x + y
129 except TypeError as e:
130     print("Fehler: {0}".format(str(e)))
131
132 # Es wird ein Fehler erzeugt (Operation wird nicht unterstützt).
133 # Python nimmt keine automatische Typumwandlung vor: starke Typisierung!
134

```