

Übungsblatt_05z_Python_String_Funktionen_v1_2

```
1 # Übungen zu Zeichenketten
2 # M. Endejan
3
4 aufgabe = 0
5 def ueberschrift(anmerkung=''):
6     global aufgabe
7     aufgabe += 1
8     print('\n', 10*'- ', ' Aufgabe ', aufgabe, ' ', sep='', end='')
9     if anmerkung:
10         print('(', anmerkung, ') ', sep='', end='')
11     print(10*'- ', '\n', sep='')
12
13
14 # Gegeben sind die folgende Variable
15 titel = "Dunkel war's, der Mond schien helle"
16
17 gedicht = """
18 1 Dunkel war's, der Mond schien helle,
19 2 schneebedeckt die grüne Flur,
20 3 als ein Wagen blitzeschnelle,
21 4 langsam um die Ecke fuhr.
22
23 \t5 Drinnen saßen stehend Leute,
24 \t6 schweigend ins Gespräch vertieft,
25 \t7 Als ein totgeschoss'ner Hase
26 \t8 Auf der Sandbank Schlittschuh Lief.
27 """
28
29 # Aufgabe 1: Die einzelnen Strophen des Gedichts sollen separiert werden.
30 # Erzeugen Sie hierzu eine neue Liste mit dem Namen strophen, deren Elemente die
31 # beiden Strophen (jeweils als eine Zeichenkette) enthält. Geben Sie die Liste
32 # über print(strophen) aus.
33 ueberschrift('Strophen separieren')
34
35 strophen = gedicht.split(sep="\n\n")
36 print(strophen)
37
38 # Aufgabe 2
39 # Erzeugen Sie eine neue Variable strophe_1, und weisen sie ihr den Text der
40 # ersten Strophe zu. Geben Sie den Inhalt der Variablen über print aus.
41 ueberschrift('erste Strophe')
42
43 strophe_1 = strophen[0]
44 print(strophe_1)
45
46 # Aufgabe 3
47 # Separieren Sie die einzelnen Zeilen der 1. Strophe und speichern Sie
48 # die Zeilen als einzelne Elemente in einer neuen Liste zeilen_strophe_1.
49 # Entfernen Sie anschließend die Zeilennummern und das Leerzeichen am Anfang
50 # jeder Zeile.
51 # Geben Sie die Variable abschließend sowohl über print(zeilen_strophe_1)
52 # als auch Element für Element (Zeile für Zeile) aus.
53 ueberschrift('Zeilen separiert und Nummern entfernt')
54
55 zeilen_strophe_1 = strophe_1.splitlines()
56 for index, _ in enumerate(zeilen_strophe_1):
57     zeilen_strophe_1[index] = zeilen_strophe_1[index].lstrip('1234 ')
58
59 print(zeilen_strophe_1)
60
61 for zeile in zeilen_strophe_1:
```

```

62     print(zeile)
63
64 # Aufgabe 4
65 # Das Gedicht soll im Format cp1252 gespeichert werden. Sichern Sie den Text
66 # dazu in einer neuen Variablen gedicht_windows. Dekodieren Sie den Text
67 # anschließend und geben Sie ihn über print aus.
68 ueberschrift('cp1252-Codierung')
69
70 gedicht_windows = gedicht.encode(encoding='cp1252')
71 print(gedicht_windows)
72
73 gedicht_unicode = gedicht_windows.decode(encoding='cp1252')
74 print(gedicht_unicode)
75
76 # Aufgabe 5
77 # Das Gedicht soll im ASCII-Format gespeichert werden. Sichern Sie den Text
78 # dazu in einer neuen Variablen gedicht_ascii. Nehmen Sie dann die notwendigen
79 # Änderungen am Text vor. Geben Sie die Variable über print aus. Kodieren
80 # Sie den Text anschließend in das ASCII-Format (encode). Geben Sie den
81 # codierten Text abschließend über print aus.
82 ueberschrift('ASCII-Codierung')
83
84 # Ersetzung der nicht im ASCII vorhandenen Zeichen (Umlaute sowie 'ß')
85 gedicht_ascii = gedicht
86 gedicht_ascii = gedicht_ascii.replace('ä', 'ae')
87 gedicht_ascii = gedicht_ascii.replace('ö', 'oe')
88 gedicht_ascii = gedicht_ascii.replace('ü', 'ue')
89 gedicht_ascii = gedicht_ascii.replace('ß', 'ss') # alternativ: casefold()
90 gedicht_ascii = gedicht_ascii.replace("'", '"')
91 print(gedicht_ascii)
92
93 # Alternative Implementierung der Zeichenersetzung über for-Schleife
94 ersetzungen = (
95     ('ä', 'ae'),
96     ('ö', 'oe'),
97     ('ü', 'ue'),
98     ('ß', 'ss'),
99     (''', '''))
100
101 for von, nach in ersetzungen:
102     gedicht_ascii = gedicht_ascii.replace(von, nach)
103
104 # Umwandlung des Gedichts in das ASCII-Format
105 gedicht_ascii = gedicht_ascii.encode(encoding='ascii')
106 print(gedicht_ascii)
107
108 # Erneute Umwandlung zu einer Zeichenkette (Unicode)
109 gedicht_unicode = gedicht_ascii.decode(encoding='ascii')
110 print(gedicht_unicode)
111
112
113 # Aufgabe 6
114 # Geben Sie das Gedicht in zentrierter Form auf dem Bildschirm aus.
115 # Die längste Zeile des Gedichtes soll als Maß für die Weite der Zentrierung
116 # genutzt werden, wobei die längste Zeile mit einem Leerzeichen am Anfang
117 # ausgegeben werde soll. Der Titel des Gedichts soll, ebenfalls zentriert,
118 # als Überschrift in Großbuchstaben vor das Gedicht gestellt werden.
119 ueberschrift('zentrierte Ausgabe')
120
121 # Nummerierung und Tabulatoren für die Ausgabe entfernen
122 zeilen = gedicht.splitlines()

```

Übungsblatt_05z_Python_String_Funktionen_v1_2

```
123 for index, _ in enumerate(zeilen):
124     zeilen[index] = zeilen[index].strip(" 12345678\t")
125
126 # längste Zeile ermitteln
127 maximale_laenge = len(max(zeilen, key=len))
128
129 # Titel und Gedicht zentriert ausgeben
130 print("", titel.upper().center(maximale_laenge))
131 for zeile in zeilen:
132     print("", zeile.center(maximale_laenge))
133
```