Python
...für Anfänger

Unit 11

by

Dr. Günter Kolousek

21. Juli 2015

Listen - 1

```
>>> buying_list = [”two oranges”, ”2l milk”]
>>> len(buying_list)
2
>>> buying_list[0]  # same as with tuples...
’two oranges’
>>> buying_list[0] = ”2 oranges”
>>> buying_list
[’2 oranges’, ’2l milk’]
>>> ”2 oranges” in buying_list
True
>>> buying_list.append(”1kg bread”)
>>> buying_list.insert(0, ”5 apples”)
>>> buying_list
[’5 apples’, ’2 oranges’, ’2l milk’, ’1kg bread’]
```

Listen - 2

```
>>> buying_list2 = buying_list
>>> buying_list2
['5 apples', '2 oranges', '2l milk', '1kg bread']
>>> buying_list2.clear()
>>> buying_list2
[]
>>> buying_list
[]
>>> buying_list2.extend(("5 apples", "2 oranges", "
>>> buying_list2
['5 apples', '2 oranges', '2l milk', '1kg bread']
>>> buying_list
['5 apples', '2 oranges', '2l milk', '1kg bread']
```

Listen - 3

```
>>> buying_list = buying_list2.copy()
>>> buying_list
['5 apples', '2 oranges', '2l milk', '1kg bread']
>>> del buying_list[2]
>>> buying_list
['5 apples', '2 oranges', '1kg bread']
>>> buying_list2
['5 apples', '2 oranges', '2l milk', '1kg bread']
>>> >>> buying_list2.sort()
>>> buying_list2
['1kg bread', '2 oranges', '2l milk', '5 apples']
>>> buying_list2.remove("2l milk")
>>> buying_list2
['1kg bread', '2 oranges', '5 apples']
```

Listen - 4

```
>>> buying_list.sort()
>>> buying_list == buying_list2
True
>>> buying_list.index("2 oranges")
1
>>> buying_list.reverse()
['5 apples', '2 oranges', '1kg bread']
>>> buying_list.extend(["5 apples"])
>>> buying_list.count("5 apples")
2
>>> buying_list.pop()
'5 apples'
>>> buying_list
['5 apples', '2 oranges', '1kg bread']
```

Listen - 5

```
>>> buying_list = [(5, ”apples”), (2, ”oranges”),\
... (2, ”apples”), (1, ”bread”)]
>>> first = buying_list[0]
>>> del first[0]
Traceback (most recent call last):
  File ”<stdin>”, line 1, in <module>
TypeError: ’tuple’ object doesn’t support item dele
>>> apples = 0
>>> for item in buying_list:
...     if item[1] == ”apples”:
...         apples += item[0]
...
>>> apples
7
```

Strings

```
>>> name = ”maxi Mustermann”
>>> name.lower()
’maxi mustermann’
>>> name
’maxi Mustermann’
>>> name.upper()
’MAXI MUSTERMANN’
>>> name.capitalize()
’Maxi mustermann’
>>> name.rjust(20)
’      maxi Mustermann’
>>> name.center(20)
’  maxi Mustermann   ’
```

Strings - 2

```
>>> number = ”  123   ”
>>> number.isdigit()
False
>>> number = ”  123   ”
>>> number.isdigit()
False
>>> number = number.strip()
>>> number
’123’
>>> number.isdigit()
True
```

Strings - 3

```
>>> address = ”1010 Wien”
>>> address.split()
[’1010’, ’Wien’]
>>> address = ”2700 Wiener Neustadt, Dr. Eckener Ga
>>> city, street = address.split(”,”)
>>> city
’2700 Wiener Neustadt’
>>> street
’ Dr. Eckener Gasse 2’
>>> int(city.split()[0])
2700
>>> street.lstrip()  # also: rstrip
’Dr. Eckener Gasse 2’
```

Sequenzen

```
>>> ”Wiener” in city
True
>>> 3 * ”Wiener”
’WienerWienerWiener’
>>> (1, 2) + (4,)
(1, 2, 3)
>>> (1, 2) * 2
(1, 2, 1, 2)
>>> (1, 2) + ”abc”
Traceback (most recent call last):
  File ”<stdin>”, line 1, in <module>
TypeError: can only concatenate tuple (not ”str”) t
>>>
```