20211005_Informationsmanagement_MV1

October 5, 2021

[1]:
```
# Tuples: geordnet, nicht veränderbare, Duplikaten erlauben

thistuple = ("apple", "banana", "cherry")

print(thistuple)
```

```
('apple', 'banana', 'cherry')
```

[2]:
```
thistuple = ("apple", "banana", "cherry", "apple", "cherry")
print(thistuple)
```

```
('apple', 'banana', 'cherry', 'apple', 'cherry')
```

[3]:
```
print(len(thistuple))
```

```
5
```

[5]:
```
# unterschiedliche Datatypen in Tuples

tuple1 = ("apple", "banana", "cherry")
tuple2 = (1, 5, 7, 9, 3)
tuple3 = (True, False, False)

tuple4 = ("abc", 34, True, 40, "male")
```

[6]:
```
# daten type von einem Tuple?

mytuple = ("apple", "banana", "cherry")
print(type(mytuple))
```

```
<class 'tuple'>
```

[7]:
```
# tuples update

x = ('apple', 'banana', 'cherry')
y = list(x)

print(type(y))
```

```
<class 'list'>
```

[9]:
```
y[1] = 'kiwi'
x = tuple(y)

print(y)
print(type(x))
```

```
['apple', 'kiwi', 'cherry']
<class 'tuple'>
```

[13]:
```
# elementen einfügen in dem Tuple dadurch, dass wir Tuple
# als Liste umwandeln und dann als Tuple wieder wechseln

thistuple = ("apple", "banana", "cherry")
y = list(thistuple) #verändern wir die Datei auf einer Liste
y.append("orange") # fügen wir einen neuen Element mit "append"
thistuple = tuple(y) # wechseln wir wieder auf tuple

print(thistuple)
```

```
('apple', 'banana', 'cherry', 'orange')
```

[19]:
```
# Übung.

# Bitte definiere einen Tuple mit 5 Elementen,
# addiere einen Element "append" und durcke das Ergebnis aus.

Dies_ist_ein_Tuple = ('Element0', 'Element1', 'Element2', 'Element3',␣
 ↪'Element4')
Dies_ist_eine_Liste_aus_dem_Tuple = list(Dies_ist_ein_Tuple)
Dies_ist_eine_Liste_aus_dem_Tuple.append('ElementX')
Dies_ist_ein_Tuple = tuple(Dies_ist_eine_Liste_aus_dem_Tuple)

print(Dies_ist_ein_Tuple)
```

```
('Element0', 'Element1', 'Element2', 'Element3', 'Element4', 'ElementX')
```

[18]:
```
# Elementen entfernen aus dem Tuple
# als Liste umwandeln, dann "remove" und dann wieder zurück als Tuple

thistuple = ('apple', 'banana', 'cherry')
y = list(thistuple)
y.remove('apple')
thistuple=tuple(y)

print(thistuple)
```

```
('banana', 'cherry')
```

[28]:
```
# Set: nicht geordnete, nicht veränderbare Datensätze, keine Duplikaten erlauben

thisset = {'apple', 'banana', 'cherry', 'apple'}

print(thisset)
print(type(thisset))
```

```
{'apple', 'cherry', 'banana'}
<class 'set'>
```

[21]:
```
print(len(thisset))
```

```
3
```

[23]:
```
# Loop durch die Elemente vom Set um die Elemente herauszufinden

thisset = {'apple', 'banana', 'cherry'}

for x in thisset:
    print(x)
```

```
apple
cherry
banana
```

[27]:
```
# addieren von elementen

thisset = {'apple', 'banana', 'cherry'}
tropical = {'ananas', 'mango', 'papaya'}

thisset.update(tropical)

for x in thisset:
    print(x)
```

```
apple
papaya
ananas
cherry
mango
banana
```

[32]:
```
# Dictionaries: veränderbar und keine Duplikaten erlaubt

thisdict = {
    "brand": "Ford",
```

```
    "model": "Mustang",
    "year" : 1964
}

print(thisdict)
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964}
```

[30]:
```
# elemente aus einem Dict darstellen

print(thisdict["brand"])
```

```
Ford
```

[34]:
```
# es werden keine Duplikaten erlaubt

thisdict = {
    "brand": "Ford",
    "model": "Mustang",
    "year" : 1964,
    "year" : 2000
}

print(thisdict)
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 2000}
```

[35]:
```
print(len(thisdict))
```

```
3
```

[44]:
```
thisdict = {
    "brand": "Ford",
    "model": "Mustang",               # String
    "year" : 1964,                    # Integer
    "color": ['red', 'white', 'blue'], # Liste
    "Heckleuchten": ('sport', 'class', 'holzklasse')
}
```

[40]:
```
print(thisdict)
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 1964, 'color': ['red', 'white',
'blue']}
```

[42]:
```
print(type(thisdict['year']))
```

```
<class 'int'>
```

[45]:
```
print(type(thisdict['Heckleuchten']))
```

```
<class 'tuple'>
```

[38]:
```
# elementen von Dictionaries wechseln

thisdict = {
    "brand": "Ford",
    "model": "Mustang",                # String
    "year" : 1964,                     # Integer
    "color": ['red', 'white', 'blue'] # Liste
}

thisdict['year'] = 2018

print(thisdict)
```

```
{'brand': 'Ford', 'model': 'Mustang', 'year': 2018, 'color': ['red', 'white',
'blue']}
```

[47]:
```
# If... Else...

#
#   Equals: a == b
#   Not Equals: a != b
#   Less than: a < b
#   Less than or equal to: a <= b
#   Greater than: a > b
#   Greater than or equal to: a >= b
```

[48]:
```
a = 33
b = 200

if b > a:
    print("b is grösser als a")
```

```
b is grösser als a
```

[50]:
```
a = 33
b = 200

if b > a:
    print("b is grösser als a")
elif a == b: # bezogen auf die vorherige Kondition
    print('a und b sind gleich')
else: # bezogen auf alle vorherige Konditionen
    print('a is grösser als b')
```

```
b is grösser als a
```

[51]:
```python
# ich kann mehrere Konditionen darstellen "and"

a = 200
b = 33
c = 500

if a>b and c>a:
    print("Beide Konditionen sind wahr")
```

```
Beide Konditionen sind wahr
```

[52]:
```python
# "oder" geht auch...

if a>b or a>c:
    print("Mindestens ist eine Kondition wahr.")
```

```
Mindestens ist eine Kondition wahr.
```

[61]:
```python
# mehrere "if" Konditionen ineinander darstellen:

x = 9

if x>10:
    print("x ist grösser 10",)
    if x>20:
        print("auch grösser 20!")
    else: # weil es eine Indentation (-->|)hat, bezieht sich auf "if x>20"
        print("aber nicht über 20.")
else:  # weil es keine Indentation hat, bezieht sich auf "if x>10"
    print("x ist kleiner oder gleich 10.")
```

```
x ist kleiner oder gleich 10.
```

[62]:
```python
price = 50

if price >= 100:
    print("price is greater than 100")
else:
    print("price is less than 100")
```

```
price is less than 100
```

[64]:
```python
price = 100

if price > 100:
    print("price is greater than 100")
```

```
elif price == 100:
    print("price is 100")
elif price < 100:
    print("price is less than 100")
```

```
price is 100
```

[65]:
```
# while loops "während" eine Kondtion wahr ist, dann tue etwas

i = 1
while i < 6:
    print(i)
    i += 1
```

```
1
2
3
4
5
```

[67]:
```
# while und if können kombiniert werden

i = 1
while i < 6:
    print(i)
    if i == 3:
        break # while loop wird gestopt
    i += 1
```

```
1
2
3
```

[69]:
```
i = 0
while i < 6:
    i += 1
    if i == 3:
        continue # while wird weiter geführt
    print(i)
```

```
1
2
4
5
6
```

[70]:
```
i = 1
while i < 6:
    print(i)
    i += 1
else:
    print("i is no longer less than 6")
```

```
1
2
3
4
5
i is no longer less than 6
```

[73]:
```
a = ['foo', 'bar', 'baz', 'qux']
s = 'H4'
i = 0 # stellt die Position in der LISTE []
while i < len(a):
    if a[i] == s:     # Processing for item found
        print(s, 'found in list.')
        break
    i += 1
else: # Processing for item not found
    print(s, 'not found in list.')
```

```
H4 not found in list.
```

[74]:
```
# Übung. Erstelle eine Liste mit 6 Elementen. Definiere ein neues Element.
# Nutze "format", "while", und "if" um herauszufinden
# ob ein Wort der gleichen Länge wie ein neues Element in der Liste hat.
```

[ ]:
```

```