lektion5

January 29, 2018

Table of Contents

1 Lektion 5

```
In [1]: a=1
        b=a
        id(a),id(b)
```

```
Out[1]: (94182036167808, 94182036167808)
```

```
In [2]: b=2
        id(b)
```

```
Out[2]: 94182036167840
```

```
In [3]: a
```

```
Out[3]: 1
```

1.1 Listen und Tupel

```
In [4]: al = [1,2,3]
        at = (1,2,4)
        am = {1,2,2,5}
```

```
In [5]: bl = al
        id(bl), id(al)
```

```
Out[5]: (140087346931592, 140087346931592)
```

```
In [6]: bl[2]=55
        bl

Out[6]: [1, 2, 55]

In [7]: al

Out[7]: [1, 2, 55]

In [8]: cl = al[:]
        id(cl)

Out[8]: 140087287268424

In [9]: al

Out[9]: [1, 2, 55]

In [10]: cl[0]=22
         cl

Out[10]: [22, 2, 55]

In [11]: al

Out[11]: [1, 2, 55]

In [12]: l = [1,2,[3,'a']]
         bl = l
         blc = l[:]
         bl[0] = 4
         blc[0] =5
         bl[2][0] = 'c'
         blc[2][0] ='aa'

In [13]: l

Out[13]: [4, 2, ['aa', 'a']]

In [14]: bt = at
         bt[0] =2


        ---------------------------------------------------------------------------

        TypeError                                 Traceback (most recent call last)

        <ipython-input-14-8afd7a8bee71> in <module>()
          1 bt = at
    ----> 2 bt[0] =2


        TypeError: 'tuple' object does not support item assignment
```

```
In [15]: am[0]
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)

<ipython-input-15-88a47290f1cd> in <module>()
----> 1 am[0]


TypeError: 'set' object does not support indexing
```

1.2 Umwandlung

```
In [16]: aLt = list(at)
         aLm = list(am)
         aLt, aLm
```

```
Out[16]: ([1, 2, 4], [1, 2, 5])
```

```
In [17]: aTl = tuple(al)
         aTm = tuple(am)
         aTl, aTm
```

```
Out[17]: ((1, 2, 55), (1, 2, 5))
```

```
In [18]: aMt = set(at)
         aMl = set(al)
         aMt, aMl
```

```
Out[18]: ({1, 2, 4}, {1, 2, 55})
```

1.3 Slicing

```
In [19]: a = [1,2,[3,4],[5,6],7]
         a
```

```
Out[19]: [1, 2, [3, 4], [5, 6], 7]
```

```
In [20]: a[0::2]
```

```
Out[20]: [1, [3, 4], 7]
```

```
In [21]: a[0:-1:]
```

```
Out[21]: [1, 2, [3, 4], [5, 6]]
```

```
In [22]: a[0:-1:2]
```

```
Out[22]: [1, [3, 4]]
```

1.4 True, False

```
In [23]: True and True
```

```
Out[23]: True
```

```
In [24]: False or True
```

```
Out[24]: True
```

```
In [25]: not False
```

```
Out[25]: True
```

1.5 Vergleiche

```
In [26]: 2 <=2, 2<2, 1<2, 1==2, 'a'=='a', (1,2)==(1,2)
```

```
Out[26]: (True, False, True, False, True, True)
```

1.6 if Anweisung

```
In [27]: def f (c):
             if c > 1:
                 c= 1
             elif c <-1:
                 c =-1
             else:
                 c = c**3
             return c
         f(2.3)
```

```
Out[27]: 1
```

```
In [28]: f(-2), f(-1), f(0), f(0.5), f(1), f(2)
```

```
Out[28]: (-1, -1, 0, 0.125, 1, 1)
```

1.7 for / while Schleifen

```
In [29]: a= [[1]]*5
         a[0][0]=11
```

```
In [30]: a
```

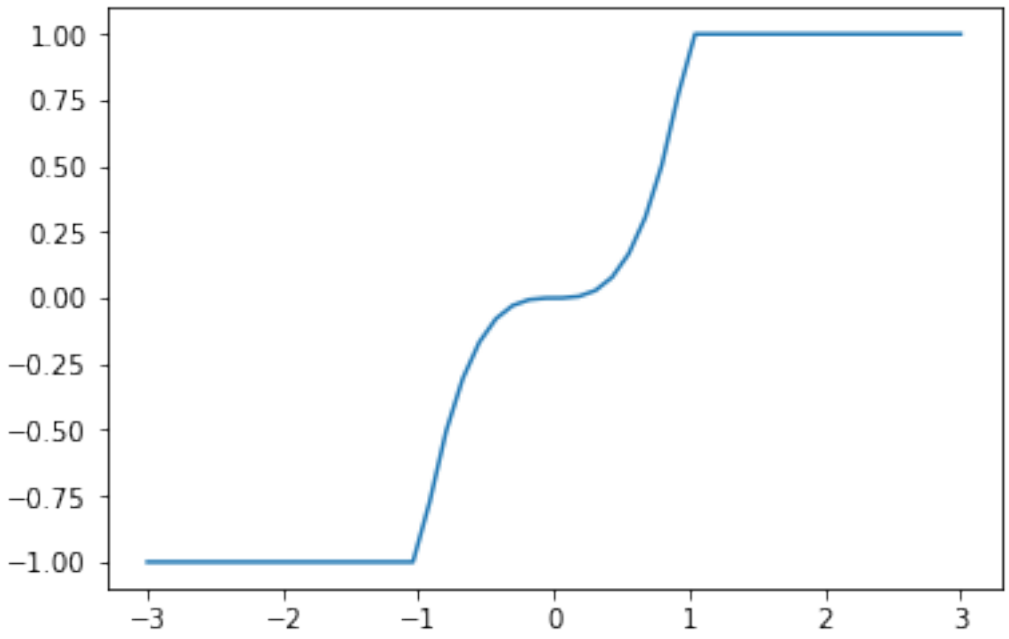
```
Out[30]: [[11], [11], [11], [11], [11]]
```

```
In [31]: (1,2)*5
```

```
Out[31]: (1, 2, 1, 2, 1, 2, 1, 2, 1, 2)
```

```
In [32]: for i in [1,2,4]:
             print(i)

1
2
4


In [33]: for i in (1,2,3):
             print(i)

1
2
3


In [34]: a= 1
         while True:
             if a >10:
                 break
             a = a+1
         a

Out[34]: 11

In [35]: l = range(5)
         for i in l:
             print(i)


0
1
2
3
4


In [36]: l2 = range(2,7)
         l2

Out[36]: range(2, 7)

In [37]: l3 = range(2,7,2)
         l3

Out[37]: range(2, 7, 2)

In [38]: a = []
         for _ in range(5):
             a.append([])
         a
```

```
Out[38]: [[], [], [], [], []]

In [39]: [[] for _ in range(5)]

Out[39]: [[], [], [], [], []]

In [40]: [( ) for _ in range(5)]

Out[40]: [(), (), (), (), ()]

In [41]: [i**2 for i in range(5)]

Out[41]: [0, 1, 4, 9, 16]

In [42]: import matplotlib.pyplot as plt
         import numpy as np
         xn = np.linspace(-3,3)
         fig = plt.figure()
         plt.plot(xn,[f(xi) for xi in xn])
         plt.show()
```




```
In [43]: al = [1,21,[3,4]]

In [44]: al1 = al.copy()
         al1[0]='a'
         al1[2][0]=22

In [45]: al

Out[45]: [1, 21, [22, 4]]
```

1.8 copy / deepcopy

```
In [46]: from copy import deepcopy

In [47]: al2 = deepcopy(al)

In [48]: al2[0] = 'aa'
         al2[2][0] = 55

In [49]: al

Out[49]: [1, 21, [22, 4]]

In [50]: al2

Out[50]: ['aa', 21, [55, 4]]

In [51]: al.insert(1,6)

In [52]: al

Out[52]: [1, 6, 21, [22, 4]]

In [53]: [al.pop(i) for i in (1,2)]

Out[53]: [6, [22, 4]]

In [54]: al

Out[54]: [1, 21]

In [55]: from sympy import *
         init_printing()

In [56]: x,y = symbols('x y')

In [57]: from IPython.display import display

In [58]: display(x**2)
```

$$x^2$$

```
In [59]: print(x**2)

x**2
```