```
####### list methods #######

>>> newstr = '***foobar***'
>>> komy = 'lala'
>>> mylist = [newstr, komy, 3, 12, 'a', 'z', 'c']
>>> mylist[1] = 'b'
>>> mylist
['**foobar**', 'b', 3, 12, 'a', 'z', 'c']
>>> mylist.reverse()
>>> mylist
['c', 'z', 'a', 12, 3, 'b', '**foobar**']
>>> mylist.sort()
>>> mylist
[3, 12, '**foobar**', 'a', 'b', 'c', 'z']
>>> len(mylist)
7
>>> mylist[0:1] = ['x', 'r', 't']
>>> mylist
['x', 'r', 't', 12, '**foobar**', 'a', 'b', 'c', 'z']
>>> mylist.count('t')
1
>>> mylist.sort()
>>> mylist
[12, '**foobar**', 'a', 'b', 'c', 'r', 't', 'x', 'z']
>>> mylist.pop()
'z'
>>> mylist
[12, '**foobar**', 'a', 'b', 'c', 'r', 't', 'x']
>>> first = mylist.pop(0)
>>> first
12
>>> mylist
['**foobar**', 'a', 'b', 'c', 'r', 't', 'x']
>>> del mylist[0:2]
>>> mylist
['b', 'c', 'r', 't', 'x']
>>> mylist.append('h')
>>> mylist
['b', 'c', 'r', 't', 'x', 'h']
>>> mylist.extend(['y', 'x', 'c'])
>>> mylist
['b', 'c', 'r', 't', 'x', 'h', 'y', 'x', 'c']
>>> mylist.sort()
>>> mylist
['b', 'c', 'c', 'h', 'r', 't', 'x', 'x', 'y']
```

```
####### list comprehension and generators #######

>>> squares = []
>>> for x in range(5):
...     squares.append(x ** 2)
...
>>> squares
[0, 1, 4, 9, 16]
>>> powlist = [i ** 2 for i in range(5)]
>>> powlist
[0, 1, 4, 9, 16]
>>>
>>> comp_list = [j ** 2 for j in range(7) if j % 2 == 0]
>>> comp_list
[0, 4, 16, 36]
>>>
>>> gen_list = (j ** 2 for j in range(7) if j % 2 == 0)
>>> gen_list
<generator object <genexpr> at 0x10bd2cf00>
>>>
>>> for x in gen_list:
...     print(x)
...
0
4
16
36
>>>
!!!! note generator uses less memory but is slower in execution !!!


>>> mystr = "suzanne berger is trying"
>>> listfromstr = [x for x in mystr if x != " "]
>>> listfromstr
['s', 'u', 'z', 'a', 'n', 'n', 'e', 'b', 'e', 'r', 'g', 'e', 'r', 'i',
's', 't', 'r', 'y', 'i', 'n', 'g']
>>>
>>> my_iterator = iter(comp_list)
>>> next(my_iterator)
0
>>> next(my_iterator)
4
>>> next(my_iterator)
16
```