

LU12.L01: Collections

lists.py

```
def add_cities(cities):
    cities.append('Biel') # add Biel at the end of the list
    cities.insert(2, 'Basel') # add Basel at Position 2
    print('add_cities:\n', cities)

def remove_cities(cities):
    cities.pop() # remove the last city from the list
    cities.pop(1) # remove the city at position 1
    cities.remove('Bern') # remove Bern from the list
    print('remove_cities:\n', cities)

def find_cities(cities):
    print('find_city:')
    print(cities[4]) # print the city at position 4
    print(cities.index('Genf')) # print the position of Genf

def loop_cities(cities):
    print('loop_cities:')
    # print all cities in the list. output must be 'Nr. x: cityname', i.e.
    # 'Nr. 1: Zürich'
    number = 1
    for city in cities:
        print(f'Nr. {number}: {city}')
        number += 1

def sort_cities(cities):
    print('sort_cities:')
    # print all cities ordered by Name (descending). output must be 'Nr. x:
    # cityname', i.e. 'Nr. 8: Zürich'
    number = len(cities)
    for city in sorted(cities, reverse=True):
        print(f'Nr. {number}: {city}')
        number -= 1
```

dicts.py

```
def add_cities(cities):
    cities['2500'] = 'Biel' # add 2500 Biel
    cities['4000'] = 'Basel' # add 4000 Basel
```

```
print('add_cities:\n', cities)

def remove_cities(cities):
    del cities['8400'] # remove the city with the zip-code 8400
    print('remove_cities:\n', cities)

def find_cities(cities):
    print('find_city:')
    print(cities['6000']) # print the name of the city with the zip-code
6000

    # print the zip-code of Genf
    zipcodes = list(cities.keys())
    names = list(cities.values())
    index = names.index('Genf')
    print(zipcodes[index])

def loop_cities(cities):
    print('loop_cities:')
    # print all cities in the list. output should be 'zip-code: name', i.e.
'3000: Bern'
    for zipcode, name in cities.items():
        print(f'{zipcode}: {name}')

def sort_cities(cities):
    print('sort_cities:')
    # print all cities ordered by zipcode (descending). output should be
'name: zip-code', i.e. 'Bern: 3000'
    for zipcode, name in sorted(cities.items(), reverse=True):
        print(f'{name}: {zipcode}')
```

Bonus: Dieses Beispiel sortiert die Städte nach ihrem Namen.



```
def sort_cities(cities):
    print('sort_cities:')
    # print all cities ordered by name
(descending). output should be 'name: zip-
code', i.e. 'Bern: 3000'
    for zipcode, name in
sorted(cities.items(), key=lambda item:
item[1], reverse=True):
        print(f'{name}: {zipcode}')
```



Marcel Suter

From:

<https://wiki.bzz.ch/> - **BZZ - Modulwiki**

Permanent link:

<https://wiki.bzz.ch/modul/archiv/m319python/learningunits/lu14/loesungen/collection>Last update: **2024/03/28 14:07**