LU11.L01 - Dictionaries

dicts.py

```
def main():
    """
    main program - creates some dictionaries and calls the functions
    """
    cities = {'8000': 'Zürich', '1200': 'Genf', '1000': 'Lausanne',
'3000': 'Bern', '8400': 'Winterthur',
              '6000': 'Luzern', '9000': 'St. Gallen', '6900': 'Lugano'}
    add_cities(cities)

    cities = {'8000': 'Zürich', '1200': 'Genf', '1000': 'Lausanne',
'3000': 'Bern', '8400': 'Winterthur',
              '6000': 'Luzern', '9000': 'St. Gallen', '6900': 'Lugano'}
    remove_cities(cities)

    cities = {'8000': 'Zürich', '1200': 'Genf', '1000': 'Lausanne',
'3000': 'Bern', '8400': 'Winterthur',
              '6000': 'Luzern', '9000': 'St. Gallen', '6900': 'Lugano'}
    find_cities(cities)
    loop_cities(cities)
    sort_cities(cities)

def add_cities(cities):
    """
    adds two cities
    :param cities: dictionary of cities
    :return: None
    """
    cities['2500'] = 'Biel'  # add 2500 Biel
    cities['4000'] = 'Basel'  # add 4000 Basel
    print('add_cities:\n', cities)

def remove_cities(cities):
    """
    removes a city
    :param cities: dictionary of cities
    :return: None
    """
    del cities['8400']  # remove the city with the zip-code 8400
    print('remove_cities:\n', cities)


def find_cities(cities):
    """
    finds a city
    :param cities: dictionary of cities
```

```
    :return: None
    """
    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):
    """
    prints all cities using a loop
    :param cities: dictionary of cities
    :return: None
    """
    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):
    """
    sorts the cities by zip-code
    :param cities: dictionary of cities
    :return: None
    """
    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}')

if __name__ == '__main__':
    main()
```

M319-LU11