



Introduction to the Python programming language

Laszlo SZATHMARY

University of Debrecen
Faculty of Informatics

Lab #5

- set
- dictionary

(last update: 2016-11-02 [yyyy-mm-dd])

2016-2017, 1st semester



set

<http://docs.python.org/library/stdtypes.html#set-types-set-frozenset>



```
>>> basket = ['apple', 'anas', 'banana', 'apple', 'orange', 'banana']
>>> fruits = set(basket)
>>> fruits
set(['orange', 'anas', 'apple', 'banana'])
>>> type(fruits)
<type 'set'>
>>> li = list(fruits)
>>> li
['orange', 'anas', 'apple', 'banana']
>>> type(li)
<type 'list'>
>>> sorted(li)
['anas', 'apple', 'banana', 'orange']
>>> fruits
set(['orange', 'anas', 'apple', 'banana'])
>>> 'kiwi' in fruits
False
>>> 'apple' in fruits
True
```

removing duplicates

Is an element in the set?

Exercise:

Consider the following elements: [5, 2, 3, 5, 1, 4, -200, 5, 1, 3, 2, 2, 5].
Remove the duplicates, i.e. one element can be present in the result at most once. The elements in the result should be sorted. ([set01](#))

```
>>> a = ['apple', 'banana', 'lemon']
>>> a = set(a)
>>> a
set(['lemon', 'apple', 'banana'])
>>> b = set()
>>> b.add('banana')
>>> b.add('orange')
>>> b
set(['orange', 'banana'])
>>> a.union(b)
set(['orange', 'lemon', 'apple', 'banana'])
>>> a.intersection(b)
set(['banana'])
>>> a.difference(b)
set(['lemon', 'apple'])
>>> a
set(['lemon', 'apple', 'banana'])
>>> a.remove('lemon')
>>> a
set(['apple', 'banana'])
```

list to set

empty set;
extending a set

classic
set operations

removing an element

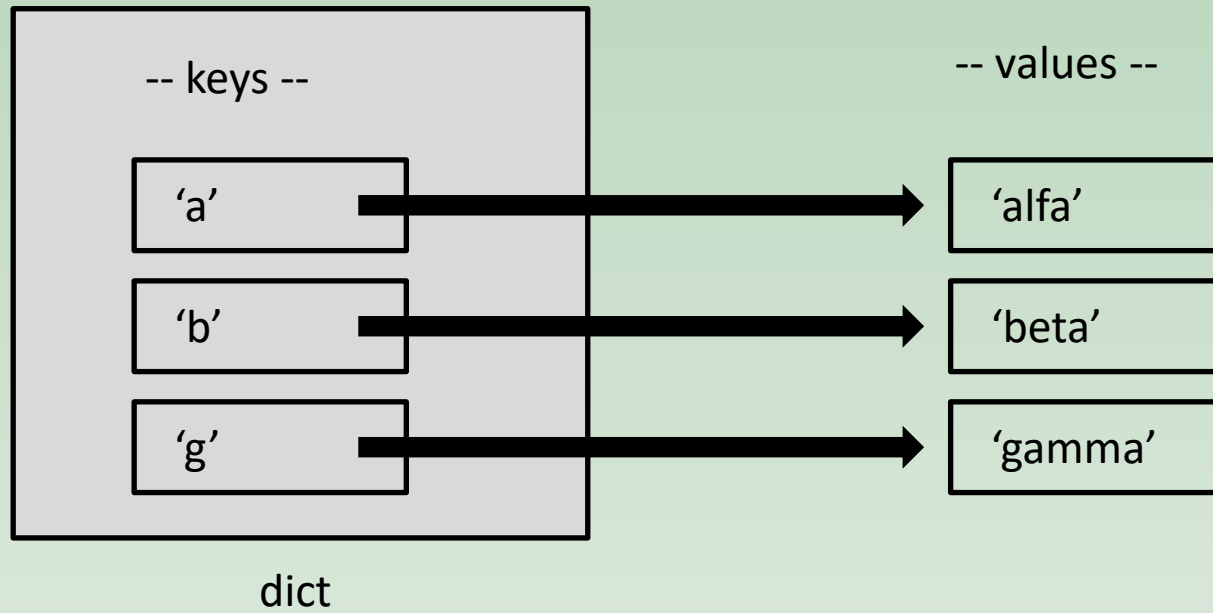
dictionary

for storing
key / value
pairs

```
1  >>> d = { }
2  >>> d['a'] = 'alfa'
3  >>> d['b'] = 'beta'
4  >>> d['g'] = 'gamma'
5  >>> d
6  {'a': 'alfa', 'b': 'beta', 'g': 'gamma'}
7  >>> d['a']
8  'alfa'
9  >>> d['o']
10 Traceback (most recent call last):
11   File "<stdin>", line 1, in <module>
12   KeyError: 'o'
13 >>> d.get('o')
14 >>> d.get('a')
15 'alfa'
16 >>> 'a' in d
17 True
18 >>> 'o' in d
19 False
20 >>> d['a'] = 'ALFA'
21 >>> d['a']
22 'ALFA'
23 >>> d
24 {'a': 'ALFA', 'b': 'beta', 'g': 'gamma'}
```

empty dictionary
(or: `d = dict()`)

Is an element with the
given key in the dictionary?



```
1 >>> d
2 {'a': 'alfa', 'b': 'beta', 'g': 'gamma'}
3 >>> d['o'] = 'omega'
4 >>> d
5 {'a': 'alfa', 'b': 'beta', 'o': 'omega', 'g': 'gamma'}
6 >>> d.keys()
7 ['a', 'b', 'o', 'g']
8 >>> d.values()
9 ['alfa', 'beta', 'omega', 'gamma']
10 >>> for k in sorted(d.keys()):
11 ...     print 'kulcs:', k, '->', d[k]
12 ...
13 kulcs: a -> alfa
14 kulcs: b -> beta
15 kulcs: g -> gamma
16 kulcs: o -> omega
17 >>> d.items()
18 [('a', 'alfa'), ('b', 'beta'), ('o', 'omega'), ('g', 'gamma')]
19 >>> for k, v in d.iteritems():
20 ...     print k, v
21 ...
22 a alfa
23 b beta
24 o omega
25 g gamma
```

the order of
elements is
arbitrary

list of keys;
list of values

use `iteritems()`
in a loop

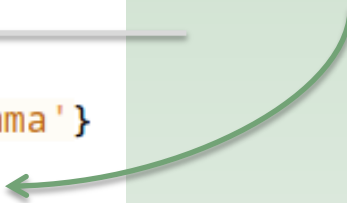
`d.items()` : list of tuples (like the `range` function)

`d.iteritems()` : generator, produces the tuples one by one (like `xrange`)

Exercise: [dict1.py](#)

removing an element from a dictionary

```
1  >>> a = 6
2  >>> a
3  6
4  >>> del a
5  >>> a
6  Traceback (most recent call last):
7    File "<stdin>", line 1, in <module>
8  NameError: name 'a' is not defined
9  >>>
10 >>> li = range(5)
11 >>> li
12 [0, 1, 2, 3, 4]
13 >>> del li[-1]
14 >>> li
15 [0, 1, 2, 3]
16 >>>
17 >>> d
18 {'a': 'alfa', 'b': 'beta', 'o': 'omega', 'g': 'gamma'}
19 >>> del d['b']
20 >>> d
21 {'a': 'alfa', 'o': 'omega', 'g': 'gamma'}
```





Exercises

1. [[20120904a](#)] removing duplicates (set)
2. [[20120905a](#)] dictionary #1
3. [[20120921a](#)] accent removal
4. [[20130218b](#)] certain characters
5. [[20120818h](#)] one hundred 50-digit long numbers (PE #13)
[version **A** only]
6. [[20120816a](#)] 8 queens