

# Slovníky

# Zoznam vs. slovník

## ► zoznam

```
zoznam = ['a', 'b', 'c']  
print(zoznam[1])  
zoznam[1]='d'  
print(zoznam)  
zoznam[3]='z'
```

## ► slovník

```
slovník={'calle':'ulica', 'gato':'mačka', 'perro':'pes'}  
print(slovník['gato'])  
slovník['gato']='kocúr'  
print(slovník)  
slovník['casa']='dom'  
print(slovník)
```

Dokumentácia slovníkov

## keys, values, items

```
d = {'a' : 1, 'b' : 2, 'c' : 3 }  
  
print(list(d.keys()))    # heslá  
print(list(d.values())) # hodnoty  
  
print(list(d.items()))  # položky = (heslo, hodnota)
```

## keys, values, items

- ▶ Heslá (keys) môžu byť len nemeniteľné typy (t.j. list ani dict **nie**).
- ▶ Ak potrebujete mať kolekciu prvkov ako heslo, použite tuple:

```
d = dict()  
d[(1,2,3)] = 1
```

- ▶ Hodnoty (values) môžu byť ľubovoľného typu

## Iterácia cez slovník

- ▶ Implicitne sa iteruje sa cez heslá

```
d = {'a' : 1, 'b' : 2, 'c' : 3 }
```

```
for i in d:  
    print(i)           # 'a', 'b', 'c'
```

- ▶ Ak treba iterovať cez hodnoty, použijeme d.values()

```
for i in d.values():  
    print(i)          # 1, 2, 3
```

- ▶ Dá sa iterovať aj cez dvojice:

```
for i, j in d.items():  
    print(i,j)        # 1, 2, 3
```

## Ako vytvoriť slovník

```
a = {}  
b = dict()  
a==b
```

```
a = {'one': 1, 'two': 2, 'three': 3}  
b = dict(one=1, two=2, three=3)  
c = dict([('two', 2), ('one', 1), ('three', 3)])  
d = dict({'three': 3, 'one': 1, 'two': 2})  
e = dict({'one': 1, 'three': 3}, two=2)  
f = dict(zip(['one', 'two', 'three'], [1, 2, 3]))  
g = dict(a)  
a == b == c == d == e == f == g      # True
```

## Ak nepotrebuje hodnoty ...

- ▶ set (množina) je slovník bez hodnôt (len heslá).
- ▶ hodí sa pokiaľ chcete zoznam, kde každý prvok je len raz a nezáleží na poradí prvkov.
- ▶ set **je nezoradený** (na rozdiel od moderného slovníku)

```
l = [3,4,2,3,1,1]
unique_l = list(set(l)) # poradie sa nezachova!
```

## Viac o set

- ▶ set má metódy (a operátory) na prácu s množinami, napr. intersection (&), difference (-), union (|).

```
s1 = {1, 2, 3}
s2 = {2, 5}
print(s1 - s2)    # {1, 3}
print(s1 & s2)   # {2}
print(s1 | s2)   # {1, 2, 3, 5}
```

- ▶ Podobne ako v prípade list, tuple a dict, pre set existuje zápis s vymenovaním prvkov:

```
s = {1, 2, 3}
s == set([1,2,3])
```

**Pozor ale:**

```
s = {}
```

je dict!!!



# Ilustračný problém

Máte nasledujúce data:

```
data = [  
    ("542213313", "19.2.1973"),  
    ("984343130", "20.4.2001"),  
    ...]
```

čo je zoznam párov, kde prvý člen páru je identifikátor osoby (číslo OP / tel. číslo), druhý je dátum narodenia.

Chcete posilať SMS ľuďom, ktorí majú narodeniny, t.j. data zorganizovať podľa dňa a mesiaca.

## Naivne riešenie so zoznamami

```
def naive_organize(data):
    dates = []
    ids = []
    for item in data:
        id_num = item[0]
        day_mon = ".".join(item[1].split('.')[ :2])
        if day_mon in dates:
            i = dates.index(day_mon)
            ids[i].append(id_num) # add to existing
        else:
            dates.append(day_mon)
            ids.append([id_num]) # create a new list

    return dates, ids # return pair of lists
```

## Problémy s riešením

- ▶ Nepohodlné na používanie. Napr., ak chceme nájsť kto má narodeniny 15.6:

```
dates, ids = naive_organize(data)
day = '15.6'
if day in dates:
    i = day.index(day)
    print(ids[i])
```

- ▶ To čo chceme je mapovanie: deň -> identifikátor:

```
organized = organize(data)
print(organized.get('15.6'))
```

- ▶ Naivne prehľadávanie zoznamu môže byť neefektívne. Samozrejme, asi by ste vedeli vymyslieť efektívnejšie varianty (napr. pomocou sorted).

## Ilustračný problém so slovníkmi

```
def organize_dict(data):
    organized = dict()

    for item in data:
        id_num = item[0]
        # get only day and month
        day_mon = ".".join(item[1].split('.')[0:2])

        if day_mon in organized:
            organized[day_mon].append(id_num)
        else:
            organized[day_mon] = [id_num]

    return organized
```

- ▶ beží rýchlejšie, kód je čitateľnejší

## Alternatívne riešenie

- Možno plánujete rozširovať slovník o nové data:

```
def add_item(d, identifier, date):
    day_mon = ".".join(date.split('.')[ :2])
    if day_mon in d:
        d[day_mon].append(identifier)
    else:
        d[day_mon] = [identifier]

def organize_dict(data):
    organized = dict()
    for i in data:
        add_item(organized, i[0], i[1])
    return organized

org = organize_dict(data)
add_item(org, '723299801', '19.02.2013')
print(org['19.02'])
```