

Slovníky a operace se slovníky

Slovník

Operace se slovníky

Metody pro slovníky

Slovník

Zatímco seznamy jsou uspořádané kolekce, slovníky jsou kolekce neuspořádané. Slovníky:

- **Podporují přístup k prvkům přes klíč** – říká se jim též **asociativní pole** (*associative array*) nebo **hashe** (*hashes*). Jednotlivým klíčům přiřazují hodnoty, takže k hodnotám přistupujeme přes příslušný klíč.
- **Jsou kolekcemi bez definovaného pořadí libovolných objektů** – klíče určují logickou (nikoli fyzickou) polohu prvku ve slovníku.
- **Mohou být heterogenní, vnořené a mohou mít proměnlivou délku.**
- **Jsou nekonstantními asociativními datovými typy**, jednotlivé prvky slovníků můžeme měnit přímo.
- **Jsou (hashovacími) tabulkami na objekty.**

Operace se slovníky

Zápis konstanty či operace	Význam
<code>d1 = { }</code>	Prázdný slovník
<code>d2 = {'ham' : 2, 'egg' : 3}</code>	Slovník o dvou prvcích
<code>d3={'food':{'ham': 2, 'egg':3}}</code>	Vnořený slovník
<code>d2['egg'] d3['food']['ham']</code>	Výběr prvků přes klíč
<code>len(d2)</code>	Počet prvků
<code>d2[klic] = novahodn</code>	Přidání (změna)
<code>del d2[klic]</code>	Vymazání prvku

Tab. 1 – Operace se slovníky

Příklady:

vytvoření slovníku:

```
d1 = {'pes' : 'dog', 'kocka' : 'cat', 'ryba' : 'fish'}
```

```
d1
```

```
{'pes': 'dog', 'kocka': 'cat', 'ryba': 'fish'}
```

výběr pomocí klíče:

```
d1['kocka']
```

```
'cat'
```

délka slovníku

```
len(d1)
```

```
3
```

přidání prvku do slovníku:

```
d1['kun'] = 'horse'
d1
{'pes': 'dog', 'kocka': 'cat', 'kun': 'horse', 'ryba': 'fish'}
```

odstranění prvku ze slovníku:

```
del d1['ryba']
d1
{'pes': 'dog', 'kocka': 'cat', 'kun': 'horse'}
```

výpis klíčů:

```
d1.keys()
['pes', 'kocka', 'kun']
```

výpis hodnot:

```
d1.values()
['dog', 'cat', 'horse']
```

“překlad”:

```
cesky = 'pes'
anglicky = d1[cesky]
anglicky
'dog'
```

opis slovníku:

```
for j in d1.keys(): print j, '\t', d1[j]
pes      dog
kocka    cat
kun      horse
```

Poznámky ke slovníkům:

- **Operace nad posloupnostmi nemohou fungovat** – nelze je např. skládat, třídít apod. protože jsou asociativní pole a jako takové nemá definované pořadí prvků.
- **Prvky do slovníku se přidávají přiřazením zatím nedefinovanému klíči**
- **Klíčem nemusí být jen řetězce** – můžeme použít další konstantní datové typy (nelze použít seznamy či slovníky, lze však použít n-tice). Třídění lze za jistých podmínek též používat. Slovníky, kde jsou klíče celá čísla, jsou vlastně seznamy bez definovaného pořadí prvků. Ukázka:

```
d2 = {1 : 'a', 2 : 'b', 3 : 'c', 4 : 'd'}
d2
{4: 'd', 3: 'c', 2: 'b', 1: 'a'}
d2[4]
'd'
d2.values()
['d', 'c', 'b', 'a']
```

Poznámka k ukázce: Python si prvky ve slovníku přehazuje tak, aby zajistil rychlý přístup.

Metody pro slovníky

Zápis konstanty či operace	Význam
<code>d.clear()</code>	Odstraňuje všechny prvky ze slovníku <code>d</code>
<code>d.copy()</code>	Vrací kopii slovníku <code>d</code>
<code>d.has_key(k)</code>	Vrací 1, když <code>d</code> má klíč <code>k</code> , jinak vrací 0
<code>d.items()</code>	Vrací seznam dvojic (<code>key, value</code>)
<code>d.keys()</code>	Vrací seznam klíčových hodnot
<code>d.update(b)</code>	Přidává všechny objekty ze slovníku <code>b</code> do <code>d</code>
<code>d.values()</code>	Vrací seznam všech hodnot <code>v</code> <code>m</code>
<code>d.get(k[,v])</code>	Vrací <code>d[k]</code> , pokud jej nalezne, jinak vrací <code>v</code>
<code>d.setdefault(k[,v])</code>	Vrací <code>d[k]</code> , pokud jej nalezne; jinak vrací <code>v</code> a provádí přiřadí <code>d[k]=v</code>
<code>d.popitem()</code>	Odstraňuje náhodnou dvojici (<code>key, value</code>) z <code>d</code> a vrací ji jako n-tici

Tab. 2 – Metody pro slovníky

Metoda `m.copy()` provádí „mělkou“ kopii položek ve slovníku a umísťuje ji do jiného slovníku. „Mělká“ kopie znamená, že se vytváří nový slovník, ale vkládají se do něj jen na odkazy na vložené objekty.

Rozdíl mezi `d.get(k[,v])` a `d.setdefault(k[,v])` je v tom, že druhá metoda, kromě toho, že vrací `v`, jestliže nenalezne objekt, provádí navíc přiřazení `d[k]=m`. Pokud je `v` vypuštěno, vrací standardně `none`.

Metoda `d.popitem()` se užívá pro iterační mazání slovníku.

Literatura:

- [1] Rubeš, J.: Nebojte se programovat, ComputerMedia, Bedihošť 2001
- [2] Lutz, M., Ascher, D.: Naučte se Python, Grada, Praha 2003
- [3] Beazley, D. M.: Python, Neocortex, Praha 2002
- [4] Python Reference Manual
- [5] Švec, J.: Létající cirkus, Python tutoriál, 2003
- [6] Harms, D., McDonald K.: Začínáme programovat v jazyce Python, Computer Press, Brno 2003