

Sõnastik

Sõnastik (*dictionary*) on struktuurne andmetüüp. Ta on nagu list, kuid palju üldisem. Listis on indekseks täisarvud. Sõnastikus võivad indekseks olla mistahes tüüpi väärtused. Sõnastiku puhul ei räägta siiski indekist vaid **võtmest** (*key*) ja igale võtmele vastab mingi väärtus. Sõnastikus on väärtuste paarid, mis koosnevad **võtmest ja väärtusest** (*key-value pair*). Sõnastikku ei kutsuta ametlikult jadaks vaid **vastetüübiks** (*mapping type*). Suurem erinevus jadast on aga see, kuidas andmeid salvestatakse ja välja loetakse – nimelt on tegemist **paisktüübiga** (*hashable type*). Erinevalt jadast puudub sõnastikus loogiline järjestus. Mis on paiskmeetod, saavad ifikad teada algoritmide ja andmestruktuuride kursusel.

Sõnastiku tähistamiseks kasutatakse loogelisi sulge {}. Näiteks võib programmis teha omistamise:

```
koordinaadid = {'x': 1, 'y': 2, 'z': 3 }
```

Nüüd on sõnastikus koordinaadid 3 võtme-väärtuse paari: võtmed on stringid x, y ja z ning neile vastavad väärtused 1, 2 ja 3.'

```
print koordinaadid['x']
```

annab vastuseks 1 ja omistamine

```
koordinaadid['y'] = 4
```

muudab väärtuse 2 väärtuseks 4.

Sõnastikus olevad võtmed peavad olema unikaalsed, st kahte paari sama võtmega ei teki. Omistamise puhul kirjutatakse lihtsalt eelmine samale võtmele vastav väärtus üle. Kui aga võtit eelnevalt sõnastikus ei ole, lisatakse ta sinna.

Väärtustena võivad olla kasutusel igasugused Pythonis kasutatavad tüübid, kaasaarvatud listid (st ühele võtmele võib vastata terve väärtuste list).

Sõnastiku loomine

Kõigepealt tuleb moodustada tühi sõnastik (kui ei saa kohe väärtuseid sisse panna):

```
arvud = {}
```

Järgnevalt võib lisada suvaliste võtmetega väärtuseid:

```
arvud['zwei'] = 'kaks'
```

```
arvud['vier'] = 'neli'
```

Hilisemal sõnastiku trükkimisel ei teki sama järgnevust kui sõnastiku ehitamisel. See tuleneb tema tehnilisest ülesehitusest (*hash*).

Sõnastiku kasutamine

Sõnastiku saab läbida for-tsükli kasutades täitsa tavapärasel viisil:

```
for arv in arvud:
    print arv
```

in-operaatori abil on võimalik kontrollida võtme kuulumist sõnastikku. Oluline on meeles pidada, et sõnastikust saab otsida ainult võtme järgi.

Kasu võib olla meetoditest `keys` ja `values`: meetod `dict.keys()` tagastab listina kõik sõnastiku võtmed ning `dict.values()` tagastab listina sõnastiku väärtused.

Kindlasti ei saa sõnastikku sorteerida, kuid mõnigase pigutusega saab sõnastikust moodustada sorteeritud list.