

Slovarji

Poišči

- Zelo pogosto
- Seznam parov
 - Za dano vrednost k iščemo v
 - Vrednosti k so enolične (so vse med seboj različne)
- Naj bo hitro!

Slovar

- Hitra realizacija
- Sezname:
 - Za dan indeks vrni vrednost
- Slovar
 - "seznam", kjer je indeks poljuben (no, skoraj)
 - "indeks" pri slovarju imenujemo ključ
 - Ključ je lahko int, nabor, str, ...
 - Enoličnost ključev
 - Dictionary (tip je dict)
 - `stevilke = {"Ana": "041 310239", "Berta": "040 318319", "Cilka": "041 103194", "Dani": "040 193831", "Eva": "051 123123", "Fanči": "040 135367", "Helga": "+49 175 4728 475"}`

Operacije

- Je prazen?
 - if stevilke:
 - V splošnem:
<http://www.pythoncentral.io/how-to-check-if-a-list-tuple-or-dictionary-is-empty-in-python/>
- Dostop
 - stevilke["ana"]
 - Če ključa "ana" ni, dobimo napako KeyError
- Dodajanje
 - stevilke["mojca"] = "041 2222222"
 - Če že obstaja ključ "mojca", prekrijemo staro vrednost
- Brisanje
 - del stevilke["mojca"]
 - Če ključ ne obstaja, izjema
- Kopiraj
 - novSl = stevilke.copy()
- Sprazni
 - stevilke.clear()

Sprehodi po slovarjih

- `for t in nekSlovar:`
 - `print(t)`
 - Izpis ključev, t torej zavzame vrednosti ključev
 - `print(nekSlovar[t])`
 - Izpis vrednosti
- `for t in nekSlovar.items():`
 - `print(t)`
 - Izpis parov
 - `ime, štev = t # 'razpakiranje'`
 - `print(ime, štev)`
- `for ime, št in nekSlovar.items():`
 - `print(ime, štev)`
 - Razpakiranje že v glavi zanke

Sprehodi po slovarjih

- **POZOR!**
 - Vrstni red, v katerem `t` (oz. ime v 3. zgledu) dobi vrednosti, je "naključen"
 - Tudi v zgledu spodaj
- `for t in nekSlovar.values():`
 - `print(t)`
 - Izpis vrednosti, `t` torej zavzame vse vrednosti
 - Kaj, če se kaka vrednost ponovi?
 - Do ključev ne moremo
 - kot pri `for x in sez` ne do indeksov

Še nekaj funkcij

- `len(slovar)` # dolžina (število "parov")
- `slovar.keys()` # **skoraj seznam** ključev
- `slovar.values()` # **skoraj seznam** vrednosti
- `slovar.items()`
skoraj seznam parov (ključ, vrednost)
- "**skoraj seznam**" – nekaj, po čemer gremo lahko z zanko `for`
- Vrstni red je "naključen"
- `list(slovar.keys())` # **seznam** ključev
- `kl in slovar` # ali ima slovar ključ `kl`
- `slovar.update(sl2)`
doda pare (ključ, vrednost) iz `sl2` v slovar

Dejansko gre za preslikavo: ključ → vrednost

Kaj je SLOVAR

Pogostost

- Beleženje metov kocke
- meti = [4, 4, 4, 3, 2, 3, 5, 3, 3, 4, 6, 6, 6, 1, 3, 6, 6, 4, 1, 4, 6, 1, 4, 4, 4, 6, 4, 6, 5, 5, 6, 6, 2, 4, 4, 6, 3, 2, 6, 1, 3, 6, 3, 2, 6, 6, 4, 6, 4, 2, 4, 4, 1, 1, 6, 2, 6, 6, 4, 3, 4, 2, 6, 5, 6, 3, 2, 5, 1, 5, 3, 6, 4, 6, 2, 2, 4, 1, 4, 4, 3, 1, 4, 2, 1, 3, 1, 4, 6, 1, 1, 3, 4, 1, 4, 3, 2, 4, 6, 6]
- Zanima nas, kolikokrat je padla katera številka

S seznamom

- DEMO
- Kaj pa, če imamo seznam klicanih števil
- klici = ['041 103194', '040 193831', '040 318319', '040 193831', '041 310239', '040 318319', '040 318319', '040 318319', '040 193831', '040 193831', '040 193831', '040 193831', '040 318319', '040 318319', '040 318319', '040 193831', '040 318319', '041 103194', '041 103194', '041 310239', '040 193831', '041 103194', '041 310239', '041 310239', '040 193831', '041 310239', '041 103194', '040 193831', '040 318319']
- Ali pa seznam imen
- klici = ['Cilka', 'Dani', 'Berta', 'Dani', 'Ana', 'Berta', 'Berta', 'Berta', 'Dani', 'Dani', 'Dani', 'Dani', 'Dani', 'Dani', 'Berta', 'Berta', 'Berta', 'Dani', 'Berta', 'Cilka', 'Cilka', 'Ana', 'Dani', 'Cilka', 'Ana', 'Ana', 'Dani', 'Ana', 'Cilka', 'Dani', 'Berta']

Rešitev s seznamom

- Ideja: "števence" vodimo kot sezname s pari (ključ, kolikokrat)
- Vedno pogledamo seznam vseh števencev, če ključ najdemo, povečamo ustrezní element v paru

```
pogostosti = []
for ts in klici:
    for žeTs in pogostosti:
        if žeTs[0] == ts:
            žeTs[1] += 1
            break
    else: # k FOR!!
        pogostosti.append((ts, 1)) # to ne bo šlo, zakaj ne?
print(pogostosti)
```

S slovarjem

```
pogostosti = {}
for ts in klici:
    # smo na klic ts že naleteli?
    if ts in pogostosti:
        pogostosti[ts] += 1
    else: #nov klic, dodamo v slovar
        pogostosti[ts] = 1
print(pogostosti)
```

Gre še lažje

```
pogostosti = {}  
for ts in klici:  
    pogostosti[ts] =  
        1 + pogostosti.get(ts, 0)  
print(pogostosti)
```

- `slovar.get(ključ, privzetaVrednost)`
- Z `get` dobimo `slovar[ključ]`, če je `ključ` v slovarju, oziroma `privzetaVrednost` drugače.

In ker smo vraževerni ...



CHUCK NORRIS DOESN'T WRITE CODE

He stares at a computer screen until he gets the program he wants.