

Pogostost

- Danj je 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 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', 'Berta', 'Berta', 'Berta', 'Dani', 'Berta', 'Cilka', 'Cilka', 'Ana', 'Dani', 'Cilka', 'Ana', 'Ana', 'Dani', 'Ana', 'Cilka', 'Dani', 'Berta']
- Za vsako število (ime) nas zanima, kako pogosto se je pojavilo v seznamu

Rešitev s seznamom

- Zadnjič

```
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)
```

Testiranje

- Pogosto ko napišemo funkcijo, hočemo na isto datoteko dodati še testni program
- Glej: [pogostost1.py](#)
- Če potem zadevo uporabimo v drugem programu (z import)
 - [TestPogostost1.py](#)
- nam testno okolje dela "zgago"

testiranje, import, __name__, __main__

Mimogrede

__name__, __main__

- Python prebere datoteko s kodo in jo izvede
 - F5
 - `import dat`
- A pred izvajanjem najprej nastavi nekaj posebnih spremenljivk
- Nas zanima tista z imenom `__name__`
- Če je datoteka s kodo glavni program, potem `__name__` dobi vrednost `__main__`
- Če pa je datoteka s kodo "uvožena" v drugo (z **import**), potem pa `__name__` postane enak imenu modula
 - [name.py](#)
 - [uvozName.py](#)

Uporaba slovarja

Nazaj na zgled

S slovarjem smo to naredili takole

```
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.

A gre še lažje

- Preštevanje je tako pogosto, da obstaja poseben modul, ki vsebuje ustrezne ukaze za preštevanje.
- Torej – dan imamo nek seznam in nas zanima, kolikokrat posamezni element nastopa v njem.

- `>>>import collections`
- `>>>steviloKlicev = collections.Counter(klici)`
- `>>> steviloKlicev`
- `Counter({'Dani': 11, 'Berta': 9, 'Cilka': 5, 'Ana': 5})`
- `>>> slovarPog = dict(steviloKlicev)`
- `>>> slovarPog`
- `{'Cilka': 5, 'Berta': 9, 'Ana': 5, 'Dani': 11}`
- `>>>list(steviloKlicev)`
- `['Cilka', 'Berta', 'Ana', 'Dani']`
- Kako pa do seznama vrednosti?
 - `[5, 9, 5, 11]`