

7. Zanka for za nize

Pojem zanke `for` je Pythonu novost, ki je večina starejših programskih jezikov ne pozna. Vpeljali jo bomo v najpreprostejši različici, pri nizih. Kasneje bomo spoznali zanko `for` v drugih, bolj zapletenih okoljih. Zato je pomembno, da zanko `for` a nize dobro razumemo.

Pri pregledovanju nizov namesto znake `while` raje uporabimo zanko `for`, ki je v jeziku Python mogočen konstrukt s semantiko, ki je precej drugačna od semantike zanke `for` v starejših programskih jezikih.

7.1. Zanka for

Če je `s` niz, potem je stavek

```
for c in s:  
    A
```

zanka, v kateri spremenljivka `c` vsakič dobi vrednost naslednjega znaka v nizu, potem pa se izvede stavek `A`. Zanka se torej prviloma izvede `len(s)` - krat. V marsičem spominja na univerzalni kvantifikator: "za vsak `c` iz `s` izvedi `A`". Z njo lahko tudi realiziramo univerzalni kvantifikator. Naj bo `P` enomestni predikat, definiran nad znaki niza `s`. Naslednji del programa preveri, ali vsi znaki niza `s` zadoščajo pogoju `P(s)`.

```
odg = True  
for c in s:  
    if not P(c):  
        odg = False  
print(odg)
```

V zanki lahko uporabljamo ukaze **`pass`**, **`continue`** in **`break`**, podobno kot pri stavku `while`. V zgornjem primeru uporaba stavka `break` naredi program hitrejši.

Tako naslednji del programa preveri, ali niz `s` vsebuje le števke in velike ter male tiskane črke od "a" do "f".

```
odg = True  
for c in s:  
    if not c in "1234567890abcdefABCDEF":  
        odg = False  
        break  
print(odg)
```

7.2. Razširjeni stavek for

Z zanko `for` pregledamo vse znake izbranega niza. V njej običajno izvajamo zaporedje stavkov.

```
for znak in niz:  
    stavek1  
    stavek2
```

Podobno kot pri stavku `while` obstaja razširjeni stavek `for` obilke

```
for znak in niz:  
    stavek1  
    stavek2  
  
else:  
    stavek3  
    stavek4  
    stavek5
```

7.3. Funkcija enumerate.

Funkcija `enumerate` nam vrne čuden objekt, ki omogoča, da ob tekočem znaku niza dobimo hkrati tudi njegov indeks, torej njegov položaj v nizu.

Če potrebujemo položaj `i` znaka `znak` v nizu `niz`, ga lahko dobimo z naslednjim ukazom:

```
for i, znak in enumerate(niz):  
    print(i, z)
```

Zgornja zanka nam v vsaki vrstici izpiše tekoči indeks in znak niza. Čeprav je `enumerate` funkcija, ni čisto običajna funkcija. Vne namreč objekt, ki zna šteti. Kaj točno to je, bomo spoznali kasneje. Za zdaj zadošča, da razumemo kakšen učinek dosežemo, če `enumerate` uporabljamo v stavku `for`.

Uporabimo ga kot koristno okrajšavo. Brez njega bi enak učinek dosegli takole:

```
i = 0  
for znak in niz:  
    print(i, z)  
    i += 1
```

7.4. Naloge

1. Sestavite funkcijo, ki za dani niz `w` vrne število različnih znakov niza `w`.
2. Sestavite funkcijo, ki obrne vrstni red znakov v danem nizu.
3. Sestavite funkcijo, ki dani niz `w` dopolne z najkrajšim nizom `u`, tako da je dobljeni niz `wu` palindrom.
4. Sestavite funkcijo, ki vrne frekvence znakov danega niza.
5. Sestavite funkcijo, ki vrne z znakoma "o" in "*" oblikovano šahovnico dimenzij `n x m`.