

Nizi

- Nizi so zaporedja znakov. Zapišemo jih v enojnih ali dvojnih narekovajih: `'beseda'`, `"beseda"`
- Posebni znaki: `\n` (nova vrsta), `\t` (tabulator), `\\` (nagibnica), `\'` (enojni narekovaj), `\"` (dvojni narekovaj), ...
- Prazen niz: `' '`, `""`

Operacije na nizih

Nizi se obnašajo podobno kot sezname znakov, le da moramo paziti na nekatere omejitve. Niza ne moremo spremeniti. To pomeni, da ne moremo spremeniti ali odstraniti posameznega znaka ali podniza.

Znaki in podnizi

- `s[i]`
 - o `i`-ti znak niza `s`.
 - o Znaki so indeksirani od 0 naprej.
 - o Če je `i` negativen, potem štejemo od konca niza proti začetku.
- `s[i:j]`
 - o Podniz od `i`-tega do `j`-tega znaka (`j`-ti znak ni več zraven). Če spustimo `i`, se podniz prične na začetku, če spustimo `j`, pa konča na koncu.
 - o `s[i:j]` je podniz, sestavljen iz znakov `si, si+1, ..., sj-1`
 - o `s[:j]` je podniz, sestavljen iz znakov `s0, s1, ..., sj-1`
 - o `s[i:]` je podniz, sestavljen iz znakov `si, si+1, ..., sn-1`, kjer je `n` število znakov v nizu `s`
 - o `s[:]` je podniz, sestavljen iz znakov `s0, s1, ..., sn-1`, kjer je `n` število znakov v nizu `s`
- `a[i:j:k]`
 - o Podniz od `i`-tega do `j`-tega znaka (`j`-ti znak ni več zraven), pri čemer vzamemo vsak `k`-ti znak
 - o `s[i:j:k]` je podniz, sestavljen iz znakov `si, si+k, si+2k, ...` do `j`-tega znaka

Stikanje

- `s + t`
 - o Sestavi nov niz, sestavljen iz znakov niza `s`, ki jim sledijo znaki niza `t`.
- `s += t`
 - o Podobno kot prej, le da novi niz shrani nazaj v `s`.
- `n * s`
 - o Sestavi nov niz, ki je enak stiku `n` kopij niza `s` (`s + s + ... + s`).
- `s *= n`
 - o Podobno kot prej, le da novi niz shrani nazaj v `s`.

Primerjanje

- operatorji `<`, `>`, `<=`, `>=`, `==` in `!=`
 - o Dva niza primerjamo leksikografsko (kot so urejene besede v leksikonih).
- Operatorja `in` in `not in`
 - o Preverimo, ali se prvi niz nahaja kot podniz v drugem nizu.

Zanka for

Podobno kot smo zanko for uporabili za pregled vseh elementov v seznamu, jo lahko uporabimo tudi za pregled vseh znakov v nizu.

Funkcije na nizih

- `len(s)` ... število znakov niza `s`
- `list(s)` ... vrne seznam znakov niza `s`
- `int(s)` ... vrne celo število, zapisano v nizu `s`

Nekatere metode na nizih

- `s.isdigit()` ... vrne `True`, če niz `s` vsebuje vsaj en znak in so vsi znaki števke
- `s.isalpha()` ... vrne `True`, če niz `s` vsebuje vsaj en znak in so vsi znaki črke
- `s.islower()` ... vrne `True`, če niz `s` vsebuje vsaj en znak in so vsi znaki male črke
- `s.isupper()` ... vrne `True`, če niz `s` vsebuje vsaj en znak in so vsi znaki velike črke
- `s.isalnum()` ... vrne `True`, če niz `s` vsebuje vsaj en znak in so vsi znaki alfanumerični (črke ali števke)
- `s.isspace()` ... vrne `True`, če niz `s` vsebuje vsaj en znak in so vsi znaki beli (presledki, tabulatorji, skoki v novo vrsto)
- `s.lower()` ... vrne kopijo niza `s`, kjer so vse črke predelane v male črke
- `s.upper()` ... vrne kopijo niza `s`, kjer so vse črke predelane v velike črke

Primeri

1. Sestavi funkcijo, ki prešteje, koliko samoglasnikov je v nizu.
2. Sestavi funkcijo, ki prešteje, koliko besed je v nizu. Besede so zaporedja poljubnih znakov, ki so med seboj ločena z enim ali več presledki.
3. Sestavi funkcijo, ki preveri, ali je dani niz palindrom.