

Programiranje za umetnike 1

~ 9 ~

Staša Vujičić Stanković

Teme

Složeni

tipovi podataka – liste.

Složeni tipovi podataka – liste

- Složeni tipovi predstavljaju kolekcije drugih objekata
- Lista predstavlja niz objekata proizvoljnih tipova
 - Dozvoljeno je da lista sadrži elemente različitog tipa
 - Jedna lista može biti element druge liste
- Elementi liste mogu da se menjaju, dodaju ili brišu.
- Svaki element u listi ima svoj indeks. Indeksiranje počinje od 0.
- Postoji veliki broj funkcija za manipulaciju listama.

Složeni tipovi podataka – liste

```
naziv_liste = [element_1, element_2, ..., element_n] #elementi se zadaju medju zagradama [ i ]  
oblici = ["trougao", "kruznicica", "kvadrat"]
```

```
prazna_lista = []  
print(prazna_lista)
```

```
lista_brojeva = [1, 2, 3]  
print(lista_brojeva)
```

```
mixed_lista = ["Programiranje za umetnike", 1]  
print(mixed_lista)
```

Pristup elementima liste

```
# Ispis prvog elementa liste  
print(mixed_lista[0])
```

- Elementima liste se pristupa preko indeksa koji se navodi iza naziva liste u zagradama [i].
- Indeksiranje počinje od 0, ali indeks može da bude i negativan
 - Pozitivni indeksi predstavljaju pomeraje u odnosu na početak liste
 - Negativni indeksi predstavljaju pomeraje u odnosu na kraj liste

Pristup elementima liste

- Moguć je pristup delu liste iz zadatog opsega
[donja_granica, gornja_granica]
(donja granica je obuhvaćena, gornja ne)
- Moguće je pravljenje liste iz zadatog opsega sa određenim korakom
[donja_granica, gornja_granica, korak]
- Moguće je izostaviti donju ili gornju granicu,
kada se ona automatski postavlja na početak, odnosno kraj liste.

Ispitivanje pripadnosti kolekciji

- Python poseduje operator `in` za ispitivanje pripadnosti elementa kolekciji

```
x in y # vraca True ako x pripada kolekciji y, a inace vraca False
```

```
x not in y # vraca False ako x pripada kolekciji y, a inace vraca True
```

Izmena elemenata liste

- Indeks elementa se može saznati funkcijom `index(object)`
- Izmena pojedinačnog elementa –
`naziv_liste[indeks_elementa] = nova_vrednost`
- Izmena grupe elemenata pristupom opsegu –
`naziv_liste[donja_granica, gornja_granica] = [nove_vrednosti]`

Izmena elemenata liste

```
lista_brojeva = [1, 2, 3, 4, 5]
print(lista_brojeva.index(2))
lista_brojeva[0] = "Ovo je lista koja nema 1 na pocetku"
print(lista_brojeva)
lista_brojeva[2:5] = [4, 6, "ni 5 na kraju"]
print(lista_brojeva)
```

Iteriranje kroz elemente liste

- Upotrebom for petlje
- Po indeksu ukoliko je potrebno menjati članove liste

```
lista_brojeva = [1, 2, 3, 4, 5]
for i in lista_brojeva:
    print(i, end=' ')
```

```
lista_brojeva = [1, 2, 3, 4, 5]
for i in range(len(lista_brojeva)):
    lista_brojeva[i] *= 100
print(lista_brojeva)
```

Funkcija `len` vraća dužinu liste

Dodavanje elemenata u listu

- Inicijalizacija putem funkcije `list()` na osnovu objekta kroz koji se može iterirati
- Množenje liste brojem prilikom čega se element liste umnožava određen broj puta, što je korisno za inicijalizaciju kada je broj elemenata unapred poznat

```
nova_lista = list("karakteristični elementi liste")  
print(nova_lista)
```

```
nova_lista = [5] * 10  
print(nova_lista)
```

Dodavanje elemenata u listu

- Dodavanje pojedinačnih elemenata na **kraj liste** funkcijom `append()`
- Dodavanje elementa na **zadatu poziciju** funkcijom `insert(pozicija, objekat)`

```
lista_brojeva = [1, 2, 3, 4, 5]
lista_brojeva.append(6)
for i in lista_brojeva:
    print(i, end=' ')

lista_brojeva.insert(0, "Ovo je nova lista: ")
print(lista_brojeva)
```

Dodavanje elemenata u listu

- Dodavanje više elemenata na **kraj liste**
 - Korišćenjem operatora +=
 - Korišćenjem funkcije extend()

```
lista_brojeva = [1, 2, 3, 4, 5]
print(lista_brojeva)

lista_brojeva += [6, 7, 8, 9, 10]
print(lista_brojeva)

lista_brojeva.extend([11, 12, 13, 14, 15])
print(lista_brojeva)
```

Uklanjanje elemenata iz liste

- Uklanjanje
 - pojedinačnog elementa ili opsega elemenata upotrebom naredbe `del`
 - elementa sa zadate pozicije funkcijom `pop(index)`
 - elementa po vrednosti funkcijom `remove(vrednost)`



Uklanjanje elemenata iz liste

```
lista_brojeva = [1, 2, 3, 3, 3, 3, 3, 3, 3, 3, 3, 4, 5, 6, 7, 8, 9, 10]
print(lista_brojeva)
del lista_brojeva[2]
print(lista_brojeva)
del lista_brojeva[10:]
print(lista_brojeva)
```

```
#podrazumevana vrednost index-a je indeks poslednjeg elementa liste
print(lista_brojeva.pop())
print(lista_brojeva)
print(lista_brojeva.pop(2))
print(lista_brojeva)
lista_brojeva.remove(3)
print(lista_brojeva)
```

Funkcije za rad sa listama

Funkcija	
<code>len(lista)</code>	vraća dužinu liste
<code>sum(lista)</code>	sabira elemente liste
<code>min(lista)</code>	vraća najmanji element u listi
<code>max(lista)</code>	vraća najveći element u listi
<code>reverse()</code>	obrće poredak elemenata liste
<code>sort()</code>	vrši sortiranje liste
<code>sorted(lista)</code>	formira novu listu koja predstavlja sortiranu zadatu listu
<code>count(vrednost)</code>	prebrojava broj pojavljivanja određene vrednosti

[lista](#)


```
brojevi = [1, 7, 5, 3, 4, 2, 6, 10, 8, 9]
print("Lista", brojevi, "ima", len(brojevi), "elemenata.")
brojevi.sort()
print("Lista sortirana rastuce:", brojevi)
brojevi.sort(reverse = True)
print("Lista sortirana opadajuće:", brojevi)

boje = ["crvena", "plava", "bela"]
boje.sort()
print("Lista boja sortirana rastuce:", boje)

oblici = ["kvadrat", "trougao", "Pravougaonik"]
oblici.sort()
#oblici.sort(key = str.lower)
print("Lista oblika sortirana rastuce:", oblici)
oblici.reverse()
print("Lista oblika u obrnutom poretku:", oblici)
```

```
brojevi = [1, 7, 10, 5, 3, 4, 2, 6, 10, 8, 9, 10]
print("Lista", brojevi, "ima", len(brojevi), "elemenata.")
print("Zbir elemenata liste:", sum(brojevi))
print("Najmanji element liste:", min(brojevi))
print("Najveci element liste:", max(brojevi))
print("Broj pojavljivanja broja 10:", brojevi.count(10))
```

Problem kopiranja liste

- Operator dodele vrednosti ne vrši kopiranje liste, već samo uvodi novi identifikator!
- Problem se rešava nekom od tehnika kopiranja liste



```
prva_lista_brojeva = [1, 2, 3, 4, 5]
print("Prva lista : ", prva_lista_brojeva)

druga_lista_brojeva = prva_lista_brojeva
print("Druga lista: ", druga_lista_brojeva)

druga_lista_brojeva[0] = 33
print("Druga lista: ", druga_lista_brojeva)
print("Prva lista : ", prva_lista_brojeva)

treca_lista_brojeva = prva_lista_brojeva.copy()
#treca_lista_brojeva = list(prva_lista_brojeva)
#treca_lista_brojeva = prva_lista_brojeva[:]
print("Treca lista: ", treca_lista_brojeva)
print("Prva lista : ", prva_lista_brojeva)

treca_lista_brojeva.append(44)
print("Treca lista: ", treca_lista_brojeva)
print("Prva lista : ", prva_lista_brojeva)
```

Literatura

- [Python 3.10.0 documentation](#)
- [Wentworth, Peter, Elkner, Jeffrey, Downey, Allen B. and Meyers, Chris. How to Think Like a Computer Scientist: Learning with Python 3. free online book](#)
- Lutz, Mark. Learning python: Powerful object-oriented programming. O'Reilly Media, Inc., 2013.
- Beazley, David, and Jones, Brian. Python Cookbook: Recipes for Mastering Python 3. O'Reilly Media, Inc., 2013.
- [Python Cheatsheet](#)
- [Website Setup Python cheat sheet](#)
- [Learn Python, basic tutorial](#)

Hvala



Staša Vujičić Stanković



stasa.vujicic.stankovic@math.rs



www.matf.bg.ac.rs/~stasa.vujicic.stankovic