

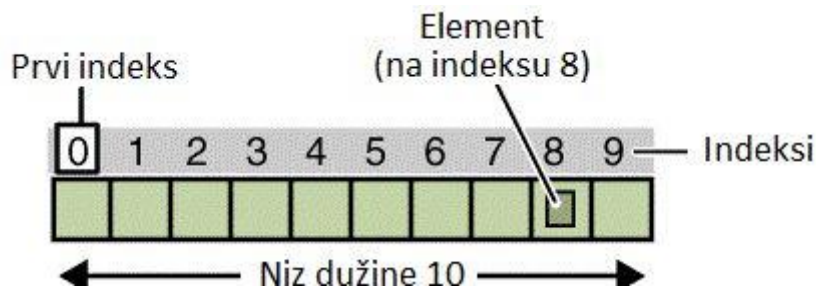
JEDNODIMENZIONALNI NIZOVI

Jednodimenzionalni niz je programska struktura koja omogućava unos više podataka spremljenih na nekoj memorijskoj lokaciji. Možemo ih zamisliti kao varijablu koja omogućuje istovremen unos više brojeva na jednu memorijsku lokaciju u memoriji računala. Primjerice deklaracijom varijable alociramo neki memorijski prostor u koji smještamo podatke u brojčanom ili tekstualnom obliku. Naravno unutar varijable smije se nalaziti samo jedan podatak. To znači da svakoj varijabli pripada samo jedna podatak. Unutar jednodimenzionalnog niza, može se smjestiti više podataka na jednu memorijsku lokaciju.

Primjerice deklaracija bvarijable u Pythonu: `a=15`. No unutar niza, moguće je smjestiti više podataka. Deklaracija jednodimenzionalnog niza izgleda ovako:

```
>>> a=[1,2,3,4,5,6,7,8]
>>> a
[1, 2, 3, 4, 5, 6, 7, 8]
```

Unutar uglatih zagrada, možemo nizati podatke; u ovom slučaju brojeve od 1 do 8. Jednodimenzionalni niz možemo zamisliti kao izduženi pravokutnik s kvadratićima koji označavaju pozicije za zapisivanje podataka niza.



Svaki kvadratić se puni odgovarajućom vrijednosti onim redoslijedom kojim korisnik puni taj niz. Pozicije za zapisivanje podataka niza nazivamo **elementima**. Svaki element ima pridružen **indeks** koji služi za **dohvaćanje elemenata niza**. Indekse uvijek započinjemo brojati od nule. Dužinu niza određujemo sami pisanjem programskog koda.

Niz možemo deklarirati na više načina:

1. način: deklaracija određivanjem fiksnih vrijednosti niza.

```
>>> a=[1,2,3,4,5,6,7,8]
>>> a
[1, 2, 3, 4, 5, 6, 7, 8]
```

2.način: deklaracija niza određivanjem duljine niza i unosom elemenata preko tipkovnice

```
niz=[0]*10
for i in range (0,10):
    niz[i]=int(input("unesite broj: " ))

for i in range (0,10):
    print(niz[i])
```

Dakle kada niz deklariramo na način da unosimo vrijednosti preko tipkovnice, potrebno je početnu vrijednost niza postaviti na **0** zatim odrediti njegovu duljinu na određeni broj elemenata niza. U prethodnom primjeru to je 10 elemenata. Zatim su nam potrebne dvije for petlje; jedna za punjenje niza i jedna za ispis elemenata niza. Svaku do petlji ograničimo na broj elemenata na koji smo postavili niz. U našem slučaju to je 10 elemenata po nizu. Unutar petlje za unos trebamo naredbu za unos elemenata niza **input**. Istu for petlju upotrijebimo za ispis elemenata niza samo usmjesto naredbe input stavimo naredbu **print** koja će omogućiti ispis elemenata niza.

Grafički primjer niza s elementima:

a	1	2	3	4	5	6
indeks	0	1	2	3	4	5

NAPOMENA:

Ukoliko želimo ispisati specifični element niza kao na primjer peti element, onda moramo prije svega dobro odbrojati pozicije indeksa niza. Indeks započinje brojati s nulom. Dakle ako tražimo 5. element niza, moramo obavezno navesti da želimo ispisati element pod indeksom 4 jer je to brojavši od nule peta pozicija indeksa. Dakle s tim treba biti vrlo oprezan.

ZADACI

Zadatak1:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Omogućite unos elemenata preko tipkovnice. Ispišite elemente niza.

Primjer ispisa:

```
unesite broj: 1
unesite broj: 2
unesite broj: 3
unesite broj: 4
unesite broj: 5
unesite broj: 6
unesite broj: 7
unesite broj: 8
unesite broj: 9
unesite broj: 10
1
2
3
4
5
6
7
8
9
10
```

Rješenje:

```
niz=[0]*10 #deklaracija niza

#unos podataka u niz
for i in range (0,10):
    niz[i]=int(input("unesite broj: " ))

for i in range (0,10):
    print(niz[i])
```

Zadatak2:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Omogućite unos elemenata preko tipkovnice. Ispišite samo parne elemente niza.

Primjer ispisa:

```
unesite broj: 1
unesite broj: 2
unesite broj: 3
unesite broj: 4
unesite broj: 5
unesite broj: 6
unesite broj: 7
unesite broj: 8
unesite broj: 9
unesite broj: 10
2
4
6
8
10
```

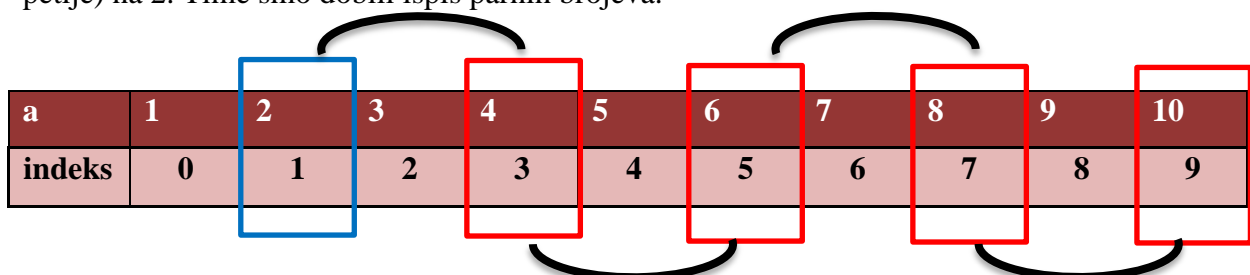
Rješenje:

```
niz=[0]*10 #deklaracija niza

#unos podataka u niz
for i in range (0,10):
    niz[i]=int(input("unesite broj: " ))

for i in range (1,10,2):
    print(niz[i])
```

Za ispis parnih članova niza, bilo je potrebno u for petlji započeti s vrijednošću 1 i ograničiti se na završnu vrijednost 10. Isto tako potrebno je staviti treći element for uvjeta (korak for petlje) na 2. Time smo dobili ispis parnih brojeva.



Dakle prvi element petlje smo postavili na jeda **for(1,10,2)** .To znači da smo kao početnu vrijednost ispisa stavili **indeks 1**. Na indeksu 1 nalazi se prvi parni element niza odnosno **broj 2**. Treći element for petlje **for(1,10,2)** postavljen je na dva čime se nizu zadalo da ispisuje svaki drugi element niza.

Zadatak3:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Omogućite unos elemenata preko tipkovnice. Ispišite svaki četvrti element niza.

Primjer ispisa:

```
unesite broj: 1
unesite broj: 2
unesite broj: 3
unesite broj: 4
unesite broj: 5
unesite broj: 6
unesite broj: 7
unesite broj: 8
unesite broj: 9
unesite broj: 10
1
4
7
10
```

Rješenje:

```
niz=[0]*10 #deklaracija niza

#unos podataka u niz
for i in range (0,10):
    niz[i]=int(input("unesite broj: " ))

for i in range (0,10,3):
    print (niz[i])
.
```

Dakle sada smo samo kod petlje za ispis početnu vrijednost postavili na nulu što znači da se ispisuje element s indeksom 0 kao početni . Broj koraka je stavljen na 3 da se omogući ispisivanje elemenata na svakom drugom indeksu.

a	1	2	3	4	5	6	7	8	9	10
indeks	0	1	2	3	4	5	6	7	8	9

Zadatak4:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Omogućite unos elemenata preko tipkovnice. Ispišite neparne elemente niza.

Primjer ispisa:

```
unesite broj: 1
unesite broj: 2
unesite broj: 3
unesite broj: 4
unesite broj: 5
unesite broj: 6
unesite broj: 7
unesite broj: 8
unesite broj: 9
unesite broj: 10
1
3
5
7
9
```

Rješenje:

```
niz=[0]*10 #deklaracija niza

#unos podataka u niz
for i in range (0,10):
    niz[i]=int(input("unesite broj: " ))

for i in range (0,10,2):
    print (niz[i])
```

a	1	2	3	4	5	6	7	8	9	10
indeks	0	1	2	3	4	5	6	7	8	9

Zadatak5:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Napunite niz bez unosa preko tipkovnice. Ispišite niz u obrnutom redoslijedu od 10 prema 1..

Primjer ispisa:

```
ispis niza
10
9
8
7
6
5
4
3
2
1
```

Rješenje:

```
niz=[0]*10

#Automatsko popunjavanje niza
for i in range (0,10):
    niz[i]=i+1

print("ispis niza")

for i in range (9,-1,-1):
    print(niz[i])
|
```

Za početka da bi kreirali niz bez da unosimo njegove elemente preko tipkovnice, napisali smo **niz[i]=i+1**. Taj i+1 znači da se svaki idući element niza puni preko **for petlje i povećava se za 1**. Nadalje kod ispisa u petlji, morali smo postaviti početnu vrijednost na **9** to jest petlja će ispisivati od zadnjeg elementa na **indeksu 9** prema prvom s indeksom 0. Završna vrijednost je -1 što znači da niz mora ići u obrnutom smjeru. A broj koraka je isto tako -1 iz razloga što se svaki indeks smanjuje kako se kreće od desno prema lijevo.

a	1	2	3	4	5	6	7	8	9	10
indeks	0	1	2	3	4	5	6	7	8	9



Od desna ulijevo

Početni indeks se umanjuje za jedan i spušta se prema lijevo strani točno prema indeksu 0
Za svaki -1 korak.

Zadatak6:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Napunite unosom preko tipkovnice. Zbrojite elemente niza i ispišite njihov zbroj.

Primjer ispisa:

```
unesite broj: 1
unesite broj: 2
unesite broj: 3
unesite broj: 4
unesite broj: 5
unesite broj: 6
unesite broj: 7
unesite broj: 8
unesite broj: 9
unesite broj: 10
ispis niza
1
2
3
4
5
6
7
8
9
10
()
('suma niza:', 55)
```

Rješenje:

```
niz=[0]*10 #deklaracija niza
suma=0

#unos podataka u niz
for i in range (0,10):
    niz[i]=int(input("unesite broj: "))

#suma niza
for i in range (0,10):
    suma=suma+niz[i]

#ispis niza
print("ispis niza")
for i in range (0,10):
    print(niz[i])
print()

print("suma niza:",suma)
```

Ključan dio ovog programa je linija koda **suma=suma+niz[i]** što znači da se u varijablu suma sprema zbroj svakog člana niza. Na primjer suma je na početku deklarirana na vrijednost 0 ->**suma=0**. Zatim u Sumu ulazi prvi broj niza a to je 1. Zatim se unutar te iste

varijable **suma** koja sad trenutno sadržava samo vrijednost 1 vrši opet zbrajanje s idućim elementom niza. Dakle sad je **suma= 1+iduci element niza**. I tako dalje do zadnjeg elementa niza.

Zadatak7:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Napunite niz bez unosa preko tipkovnice. Ispišite sumu **parnih** elemenata niza.

Primjer ispisa:

```
ispis niza
1
2
3
4
5
6
7
8
9
10
('Zbroj parnih brojeva:', 30)
```

Rješenje:

```
niz=[0]*10
zbroj=0

#Automatsko popunjavanje podacima
for i in range (0,10):
    niz[i]=i+1

#zbroj parnih brojeva u nizu
for i in range (0,10):
    if niz[i]%2==0:
        zbroj=zbroj+niz[i]

#ispis niza
print("ispis niza")
for i in range (0,10):
    print(niz[i])

print("Zbroj parnih brojeva:",zbroj)
```

Da bi u petlji provejrali koji su elementi parni koji nisu, prije nego uđu u zbroj, ispitali smo svaki element pomoću if-else odluke i ispitali svaki element niza modularnim dijeljenjem s brojem 2 (**%-modularno dijeljenje**) Ukoliko zadani član niza nije nakon modularnog dijeljenja nije imao ostatak to jest ostatak je bio **==0**, pribrojen je varijablu zbroj i tako sve do zadnjeg elementa niza.

Zadatak8:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Napunite niz bez unosa preko tipkovnice. Pomnožite svaki element niza **brojem 3**.

Primjer ispisa:

```
ispis niza
3
6
9
12
15
18
21
24
27
30
```

Rješenje:

```
niz=[0]*10

#Automatsko popunjavanje niza
for i in range (0,10):
    niz[i]=i+1

#Kvadriranje brojeva
for i in range (0,10):
    niz[i]=niz[i]*3

#ispis niza
print("ispis niza")
for i in range (0,10):
    print(niz[i])
print()
```

Da bi sve elemente nizova pomnožili s 3 ponovno smo koristili već prije korištenu metodu: **niz[i]=niz[i]*3**. Time smo dobili da se prvi element niza 1 smjesti u niz[i] i u njemu se izvrši operacija množenja. Nakon toga u isti niz za idući element se izvrši ista operacija i sve se spremi ponovno u niz: **niz[i]=niz[i]*3** Ovo izdvajanje niz[i]=niz[i]*3 znači da se operacijom množenja unutar niza prolazi svakim elementom niza.

Zadatak9:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Napunite niz bez unosa preko tipkovnice. Izvršite kvadriranje – operacija u pythonu je **a**2**.

Primjer ispisa:

```
ispis niza
1
4
9
16
25
36
49
64
81
100
```

Rješenje:

```
niz=[0]*10

#Automatsko popunjavanje niz:
for i in range (0,10):
    niz[i]=i+1

#Kvadriranje brojeva
for i in range (0,10):
    niz[i]=niz[i]**2

#ispis niza
print("ispis niza")
for i in range (0,10):
    print(niz[i])
print()
```

Zadatak10:

Kreirajte dva niza od po 5 elemenata. Zatim napunite prvi niz preko tipkovnice. Nakon toga premjestite elemente prvog niza u drugi kreirani niz i ispišite taj niz.

Primjer ispisa:

```
unesite broj: 1
unesite broj: 2
unesite broj: 3
unesite broj: 4
unesite broj: 5
ispis originalnog niza
1
2
3
4
5
ispis drugog niza
1
2
3
4
5
```

Rješenje:

```
niz1=[0]*5
niz2=[0]*5

#unos podataka u niz
for i in range (0,5):
    niz1[i]=int(input("unesite broj: "))
print("ispis originalnog niza")
for i in range (0,5):
    print(niz1[i])
#Prepisivanje parnih brojeva u drugi niz
j=0
for i in range (0,5):
    niz2[j]=niz1[i]
    j=j+1

#ispis niza
print("ispis drugog niza ")
for i in range (0,5):
    print(niz2[i])
print()
```

Zadatak11:

Deklarirajte niz naziva **niz** i ograničite ga na 10 elemenata. Napunite unosom preko tipkovnice. Koirajte u drugi niz samo parne elemente prvog niza.

Primjer ispisa:

```
unesite broj: 1
unesite broj: 2
unesite broj: 3
unesite broj: 4
unesite broj: 5
unesite broj: 6
unesite broj: 7
unesite broj: 8
unesite broj: 9
unesite broj: 10
ispis originalnog niza
1
2
3
4
5
6
7
8
9
10
ispis niza
2
4
6
8
10
0
0
0
0
0
```

Rješenje:

```
niz1=[0]*10
niz2=[0]*10

#unos podataka u niz
for i in range (0,10):
    niz1[i]=int(input("unesite broj: "))
print("ispis originalnog niza")
for i in range (0,10):
    print(niz1[i])
#Prepisivanje parnih brojeva u drugi niz
j=0
for i in range (0,10):
    if niz1[i]%2==0:
        niz2[j]=niz1[i]
        j=j+1

#ispis niza
print("ispis niza")
for i in range (0,10):
    print(niz2[i])
print()
```

Kod ispisa drugog niza, pojavile su se vrijednosti 0,0,0,0. To nije pogreška nego ukoliko unaprijed određeni niz kao u našem primjeru niz od 10 elemenata nije **cjelovito popunjen sa svim elementima, nedostajući elementi poprimaju vrijednost nula**. Dakle niz sam dodaje vrijednost 0 na prazna mjesta elemenata.

Zadatak12:

Napunite niz s 10 elemenata i ispišite najveći i najmanji element niza.(MIN/MAX)

Primjer ispisa:

```
unesite broj: 1
unesite broj: 2
unesite broj: 3
unesite broj: 4
unesite broj: 5
unesite broj: 6
unesite broj: 7
unesite broj: 8
unesite broj: 9
unesite broj: 10
Niz:
1
2
3
4
5
6
7
8
9
10
('Najveci broj u nizu:', 10)
('Najmanji broj u nizu:', 1)
```

Rješenje:

```
niz=[0]*10
min=0
max=0

#unos podataka u niz
for i in range (0,10):
    niz[i]=int(input("unesite broj: "))

#pocetna vrijednost minimuma i maksimuma
min=max=niz[0]

print("Niz:")
for i in range (1,10):
    if min>niz[i]:
        min=niz[i]
    elif max<niz[i]:
        max=niz[i]

for i in range (0,10):
    print(niz[i])

print("Najveci broj u nizu:",max)
print("Najmanji broj u nizu:",min)
```

Kod ovog zadatka najbitnija instanca su nam provjera uvjeta pomoću if-else odluke.

```
if min>niz[i]:  
    min=niz[i]  
elif max<niz[i]:  
    max=niz[i]
```

Koristili smo naredbe **min** i **max**. One automatski provjeravaju sve elemente niza uspoređujući ih i na kraju utvrđuju koji je od njih najveći odnosno najmanji. Naredbe min i max rade usporedbu s elementima niza. Kada utvrde najmanji ili najveći, dohvate tu vrijednost i smjeste u varijablu ili u našem slučaju niz. $Min > niz[i]$ i $Max < niz[i]$ znači da se uspoređuju elementi unutar raspona niza.