

Variable tipa niz - array

Kada i zašto koristiti varijablu tipa niza?

```
a = 2
b = 4
c = 6
d = 8
e = 10
Ispis a, b, c, d, e
```

2 4 6 8 10

```
za i = 0 do 4
    unesi vars(i)

za i = 0 do 4
    ispis vars(i)
```

Oznaka varijable	Vrijednost varijable	Oznaka varijable
a	2	<u>vars(0)</u>
b	4	<u>vars(1)</u>
c	6	<u>vars(2)</u>
d	8	<u>vars(3)</u>
e	10	<u>vars(4)</u>

Nizovi i/ili liste

- Programski jezici koriste naziv:
 - lista (Python, LOGO)
 - niz (eng. array) (Qbasic, C)
- Namjena je ista
- Nizovi su općenito definirane strukture (nakon što se definira broj elemenata, onda to ne možemo naknadno mijenjati tijekom izvođenja programa).
- Moramo unaprijed znati koliko ima elemenata niza (ili predvidjeti maksimalan broj) te rezervirati prostor u memoriji za tu namjenu.
- Nizovi sadrže isti tip podataka

QBasic

Deklaracija niza

```
n=7
```

```
DIM a(n)
```

```
For i = 1 to 7
```

```
    Input a(i)
```

```
Next i
```

Ne koristi se niz

```
Input A
```

```
Input B
```

```
Input C
```

```
Input D
```

```
Input E
```

```
Input F
```

```
Input G
```

Python

```
n=7
```

```
broj=[0]*n
```

```
for i in range(n):
```

```
    broj[i]=int(input('Unesi broj: '))
```

Kako spremiti podatke?

- Prosjek brojeva

```
suma = 0
n = int(input("Koliko ima brojeva: "))

for i in range(n):
    broj = eval(input("Unesi broj: "))
    suma = suma + broj

p = suma / n
print("Prosjek:", p)
```

- Ispišite sve brojeve koji su manji od prosjeka

Kako kad ih nismo spremili!?

Niz – uređena struktura podataka

- **Niz** je uređena struktura podataka gdje isto ime dijeli više podataka
- S nizovima radimo slično kao u matematici
 - U matematici se obično niz označi slovom (kao što to radimo s varijablama), a elementi niza indeksima:
 - Niz **s** u matematici: s_0, s_1, \dots, s_{n-1}
 - Niz **s** u programiranju: $s(0), s(1), \dots, s(n-1)$
- Elementi (članovi) niza se **identificiraju** indeksom
- **Indeks** može biti nenegativni cijeli broj (konstanta, varijabla, cjelobrojni izraz)
 - $x(0)$ $x(9)$ $x(n)$ $x(\text{MAX})$ $x(n+1)$ $x(k/m+5)$
- Svi podaci u nekom nizu moraju biti **istog tipa**.

Tipovi podataka u nizu

- Podatak u nizu može biti:
 - Cjelobrojni
 - $A(1) = 2$; $b(5) = 3$
 - Realni
 - $A(11) = 3.333$; $d(14) = 4.4444$
 - String – niz znakova
 - $C(1) = \text{„Ivan”}$; $d(2) = \text{„kuća”}$

Vrste nizova

- Jednodimenzionalni niz veličine n

$x(0)$	$x(1)$	$x(2)$...	$x(n-2)$	$x(n-1)$
--------	--------	--------	-----	----------	----------

- Dvodimenzionalni niz (matrica, tablica) veličine $m \times n$

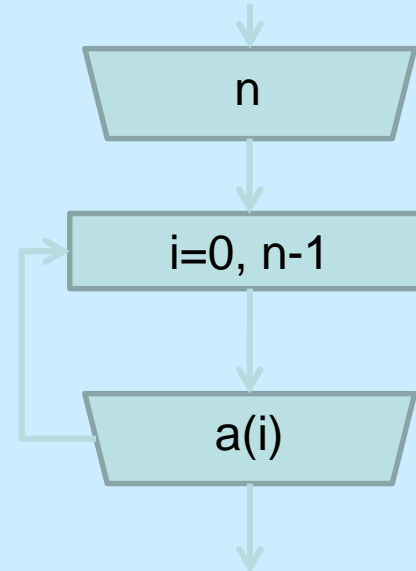
$x(0,0)$	$x(0,1)$	$x(0,2)$...	$x(0,n-2)$	$x(0,n-1)$
$x(1,0)$	$x(1,1)$	$x(1,2)$...	$x(1,n-2)$	$x(1,n-1)$
...					
$x(m-2,0)$	$x(m-2,1)$	$x(m-2,2)$...	$x(m-2,n-2)$	$x(m-2,n-1)$
$x(m-1,0)$	$x(m-1,1)$	$x(m-1,2)$...	$x(m-1,n-2)$	$x(m-1,n-1)$

Unos niza

- Pseudo kod

```
unesi (n)  
za i=0 do n-1  
    unesi i-ti član niza a(i)
```

Dijagram toka

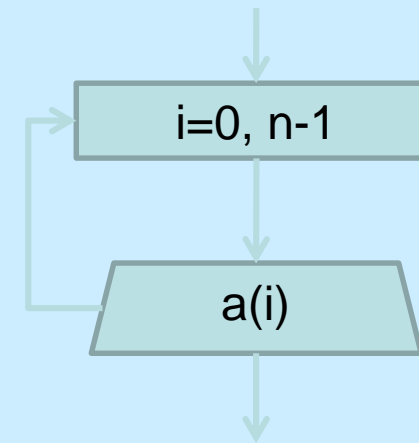


Ispis niza

- Pseudokod

```
za i=0 do n-1  
    ispiši i-ti član niza a(i)
```

Dijagram toka



Nizovi u Pythonu

- Niz istovrsnih podataka se može kreirati operatorom ponavljanja: *
- Instrukcijom: `a = [0]*5`
stvaramo niz: `a = [0, 0, 0, 0, 0]`
- ----
- Niz `s` u matematici: `s0, s1, ..., sn-1`
- Niz `s` u Pythonu: `s[0], s[1], ..., s[n-1]`

```
n = int(input("Koliko ima brojeva? "))
#definiranje niza
broj = [0]*n

#unos elemenata niza
for i in range(n):
    broj[i] = int(input("Unesi broj: "))
```

Stvaranje niza – statički

Niz – matematika - programiranje

- Matematičari mogu lako napisati i formulu za zbroj članova niza:

$$\sum_{i=0}^{n-1} s_i$$

- U programiranju je slično, koristimo jednu varijablu i indekse:

Zbroj = 0

for i in range(n):

 zbroj = zbroj + s[i]

Niz i/ili lista



- Python liste su dinamičke, mogu se mijenjati tijekom izvođenja programa (rasti i smanjivati se).
- U liste možemo spremati različite tipove podataka (stringove, brojeve, ...).
- Elementi liste se nalaze unutar uglatih zagrada
- **U kolegiju koristimo naziv - NIZ.**

```
br = [1,3,5,7]
print (br)
```

```
vozilo = ['osobno', 'teretno', 'autobus']
print (vozilo)
```

```
razno = [1,2, 'brod', '123']
print (razno)
```

```
prazno = []
print (prazno)
```

```
*** Remote Interpreter Reinitialized ***
>>>
[1, 3, 5, 7]
['osobno', 'teretno', 'autobus']
[1, 2, 'brod', '123']
[]
>>>
```

Stvaranje, unos i ispis elemenata niza u Python-u

```
n = int(input("koliko ima elemenata niza?"))
# -- definira se niz x od n elemenata
x = [0] * n
# -- unos elementa niza
for i in range(n):
    x[i] = int(input("unesi broj: "))

# -- ispis elemenata niza I.
for i in range(n):
    print(x[i])
# -- ispis elementa niza II.
print(x)
# -- ispis elemenata niza III.
for i in x:
    print(i)
```

```
# ispis elemenata niza u jednom retku
for i in range(n):
    print(x[i], end=' ')
```

```
>>>
*** Remote Interpreter Reinitialized ***
>>>
Ispis 1
2
3
4
Ispis 2
[2, 3, 4]
Ispis 3
2
3
4
Ispis 4
2 3 4
>>>
```

Oblikovanje niza - statički

- Niz istovrsnih podataka se oblikuje primjerom operatora ponavljanja: *
- Instrukcijom: `a = [0]*5` - stvorimo niz: `a = [0, 0, 0, 0, 0]`
- Rekapitulacija:
 - Inicijalizacija niza, **punjenje** nulama,
 - Punjenje niza brojevima koje unosi korisnik,
 - Ispis niza.

```
n = int(input("koliko ima elemenata?"))
niz = [0] * n
for i in range(n):
    niz[i] = int(input("unesi broj: "))

for element in niz:
    print(element)
```

Oblikovanje niza - dinamički

- Broj elemenata niza nije poznat unaprijed
- Inicijalizirati niz, kao **prazan**: `brojevi = []`
- **primjer**: `Vrste_Niza.py`

```
#elementi niza se dodaju dok se ne unese nula
```

```
brojevi = []  
x = int(input("Unesi broj: "))  
while x != 0:  
    brojevi.append(x) #dodaj x u niz  
    x = int(input("Unesi broj: "))  
  
for br in brojevi:  
    print(br)
```