

DVODIMENZIONALNO POLJE

Dvodimenzionalno polje se može predložiti tablicom s zadanim brojem redaka i stupaca. Položaj člana unutar dvodimenzionalnog polja označen je sa dva cjelobrojna indeksa. Prvi indeks određuje redak, a drugi stupac.

Prvi član dvodimenzionalnog polja (prvi redak, prvi stupac) označen je indeksom: $[0,0]$, a posljednji (posljednji redak, posljednji stupac) indeksom: $[(\text{broj redaka} - 1), (\text{broj stupaca} - 1)]$.

$a_{0,0}$	$a_{0,1}$	$a_{0,2}$
$a_{1,0}$	$a_{1,1}$	$a_{1,2}$
$a_{2,0}$	$a_{2,1}$	$a_{2,2}$

Primjer deklaracije dvodimenzionalnog polja:

$$a = [[], []]$$

Prve zagrade označavaju broj retka, a druge broj stupca. Raspored elemenata unutar matrice određen je indeksima koji započinju sa $a_{0,0}$, a završavaju sa $a_{i,j}$ što u prijevodu znači da krajnji indeksi matrice završavaju s korisnikovim odabirom stupca i retka. Primjerice ako korisnik odabere 3x3 odnosno tri retka i tri stupca. U Python programskom jeziku postoji više načina kreiranja matrice. Najčešće je to način da se matrica kreira u Python Shell-u. No kako bi u skripti kreirali i napunili matricu elementima, moramo se poslužiti for petljom odnosno dvjema for petljama od kojih se jedna odnosi na redak, a druga na stupac.

```
a = []
for i in range(3):#odredjivanje retka
    a.append([])
    for j in range(3):#odredjivanje stupca
        a[i].append(i+j)
print(a)
```

Prva for petlja sa elementom **i** odnosi se na određivanje broja redaka. Druga petlja sa **j** elementom, odnosi se na određivanje broja stupaca. Izraz **a.append** označava u svakom slučaju da se vrijednost stupca pripaja to jest dodaje u matricu.

Kako bi lakše mogli predočiti matricu, sljedeći prikaz popisuje retke i stupce matrice 3x3.

```
matrica = [1, 2, 3]    -> 1. red (indeks 0)
          [4, 5, 6]    -> 2. red (indeks 1)
          [7, 8, 9]    -> 3. red (indeks 2)
          | | |
          | | -> 3. stupac (indeks 2)
          | -> 2. stupac (indeks 1)
          -> 1. stupac (indeks 0)
```

ZADACI

Zadatak1:

Napraviti fiksni unos stupaca i redaka

Ispis:

```
[[0, 1, 2], [1, 2, 3], [2, 3, 4]]  
>>> |
```

Rješenje:

```
a = []  
for i in range(3):#odredjivanje retka  
    a.append([])  
    for j in range(3):#odredjivanje stupca  
        a[i].append(i+j)  
print(a)
```

U ovom primjeru zadan je rang retka i stupca 3x3 (**range(3)**). Unutar tog ranga, određeno je automatsko popunjavanje elemenata matrice pa je tako dobiveno [0,1,2], [1,2,3], [2,3,4].

Zadatak2:

Napraviti unos redaka i stupaca matrice preko tipkovnice.

Ispis:

```
Broj redaka:2  
Broj stupaca:2  
[[0, 1], [1, 2]]  
\\ \ |
```

Rješenje:

```
m=input("Broj redaka:")  
n=input("Broj stupaca:")  
  
a = []  
for i in range(m):#odredjivanje retka  
    a.append([])  
    for j in range(n):#odredjivanje stupca  
        a[i].append(i+j)  
  
print(a)
```

U odnosu na prethodni zadatak, napravljena je mala preinaka. U for petljama na mjestu `range()` su sada upisane varijable retka i stupca odnosno u našem slučaju `m` i `n`. Naravno koliko god stupaca i redaka unijeli, python će automatski popuniti pozicije elemenata u rangu od nula do vrijednosti koju smo preko tipkovnice odredili kao krajnju vrijednost retka i stupca.

Zadatak3:

Omogućiti tipkovnički unos redaka i stupaca te elemenata matrice.

Ispis:

```
Broj redaka:2
Broj stupaca:3
unesite elemente matrice:1
unesite elemente matrice:2
unesite elemente matrice:3
unesite elemente matrice:12
unesite elemente matrice:3
unesite elemente matrice:4
[[1, 2, 3], [12, 3, 4]]
```

Rješenje:

```
m=input("Broj redaka:")
n=input("Broj stupaca:")

a=[]
for i in range(m):
    b=[]
    for j in range(n):
        b.append(input("unesite elemente matrice:"))
    a.append(b)

print(a)
```

Prethodni zadatak je nadopunjen naredbom unosa na mjestu for petlje. Na mjesto `append` umjesto `a.append(i+j)`, upisano je `input(„Unesite elemente matrice:“)`. Oznake `i+j` označavale su automatsko popunjavanje elemenata matrice. Sada na tome mjestu dodajemo ručno preko tipkovnice svoje elemente.

Zadatak4:

Pomnožiti svaki element matrice s brojem 3

Ispis:

```
1. matrica
[[0, 3, 6], [3, 6, 9], [6, 9, 12]]
>>> |
```

Rješenje:

```
a = []
print("1. matrica")
for i in range(3):#odredjivanje retka
    a.append([])
    for j in range(3):#odredjivanje stupca
        a[i].append((i+j)*3)
print(a)
```

Upotrijebljena je matrica s fiksnim retkom i stupce te automatskim popunjavanjem elemenata matrice. Da bi svaki element matrice pomnožili s tri, na mjestu **a[i].append((i+j)*3)** dodali smo operaciju množenja i pomnožili taj izraz s tri. Tako smo fiksnu elemente fiksne matrice pomnožili s tri.

Zadatak5:

Kreirati dvije matrice s fiksnim elementima i zbrojiti ih.

Ispis:

```
1. matrica
[[0, 1, 2], [1, 2, 3], [2, 3, 4]]
2. matrica
[[0, 1, 2], [1, 2, 3], [2, 3, 4]]
[[0, 1, 2], [1, 2, 3], [2, 3, 4], [0, 1, 2], [1, 2, 3], [2, 3, 4]]
```

Rješenje:

```
a = []
for i in range(3):#odredjivanje retka
    a.append([])

    for j in range(3):#odredjivanje stupca
        a[i].append(i+j)
print("1. matrica")
print(a)
```

```
b = []
for i in range(3):#odredjivanje retka
    b.append([])

    for j in range(3):#odredjivanje stupca
        b[i].append(i+j)
print("2. matrica")
print(b)
c=a+b
print(c)
```

Zadatak6:

Preko tipkovnice unesite množitelj matrice s fiksno određenim elementima.

Ispis:

```
unesite mnozitelj matrice:56
[[0, 56, 112], [56, 112, 168], [112, 168, 224]]
```

Rješenje:

```
mnozitelj=int(input("unesite mnozitelj| matrice:"))
a = []
for i in range(3):#odredjivanje retka
    a.append([])
    for j in range(3):#odredjivanje stupca
        a[i].append((i+j)*mnozitelj)
print(a)
```

Na mjestu `a[i].append((i+j)*mnozitelj)` dodali smo varijablu **mnozitelj** koja nam je omogućila tipkovnički unos broja s kojim ćemo pomnožiti elemente matrice.

Zadatak7:

U matrici fiksnih redaka i stupaca te automatskog unosa elemenata pomnožite redak i stupac.

Ispis:

```
[[0, 0, 0], [0, 1, 2], [0, 2, 4]]  
>>> |
```

Rješenje:

```
a = []  
for i in range(3):#odredjivanje retka  
    a.append([])  
    for j in range(3):#odredjivanje stupca  
        a[i].append((i*j))  
print(a)
```

U retku **a[i].append** gdje gdje inače kod automatskog unosa elemenata matrice stoji **(i+j)** , promijenilo se u **(i*j)** čime smo omogućili množenje retka i stupca. Dobivena vrijednost je zapravo rezultat množenja prvog retka s prvim stupcem.

```
#[0,1,2]          #0,1,2 * #0    = 0 0 0  
#[1,2,3]          #1      0 1 2  
#[2,3,4]    #i*j    #2      0 2 4  
|
```


PRIMJERI

Sljedeći primjeri pokazuju konkretnu primjenu matrica u programiranju.

Primjer 1:

```
matrica = []
matrica.append([1, 2, 3])
matrica.append([4, 5, 6])
matrica.append([7, 8, 9])
print "Ispis cijele liste s podlistama:"
print matrica
print "Ispis red po red:"
for red in matrica:
    print red
print "Čisti ispis svih elemenata - bez zagrada, zareza i sl.:"
for red in matrica:
    for element in red:
        print element,
    print
print "Zadnji element u zadnjem retku: ", matrica[2][2]
```

U primjeru 1 matrica treba ispisati posljednji element koji se nalazi pod sjecištem retka i stupca (2,2). Matrica ima 3 retka i 3 stupca. S obzirom da se retci i stupci broje od indeksa nule, krajnji element retka i stupca sadržava indeks 2.

```

          [1, 2, 3]    -> 1. red (indeks 0)
matrica = [4, 5, 6]    -> 2. red (indeks 1)
          [7, 8, 9]    -> 3. red (indeks 2)
          |  |  |
          |  |  -> 3. stupac (indeks 2)
          |  -> 2. stupac (indeks 1)
          -> 1. stupac (indeks 0)
```

Ono što program treba ispisati jest zadnji element na lokaciji retka i stupca (2,2).

```
Čisti ispis svih elemenata - bez zagrada, zareza i sl.:
1 2 3
4 5 6
7 8 9
Zadnji element u zadnjem retku: 9
```

Primjer 2:

Sljedeća matrica 3x3 treba ispisati elemente koji se nalaze na dijagonali.

```
matrica = []
matrica.append([1, 2, 3])
matrica.append([4, 5, 6])
matrica.append([7, 8, 9])
print "Ispis cijele liste s podlistama:"
print matrica
print "Ispis red po red:"
for red in matrica:
    print red
print "Čisti ispis svih elemenata - bez zagrada, zareza i sl.: "
for red in matrica:
    for element in red:
        print element,
    print
#ispis po glavnoj dijagonali
print ("Ispis po glavnoj dijagonali: ")
print(matrica[0][0])
print(matrica[1][1])
print(matrica[2][2])
```

Ono što je potrebno kod ispisa vrijednosti po dijagonali je točno odrediti indekse retka i stupca.

```
matrica = [
    [1, 2, 3]   -> 1. red (indeks 0)
    [4, 5, 6]   -> 2. red (indeks 1)
    [7, 8, 9]   -> 3. red (indeks 2)
    | | |
    | | -> 3. stupac (indeks 2)
    | -> 2. stupac (indeks 1)
    -> 1. stupac (indeks 0)
```

Dakle moramo ispisati elemente matrice 3x3 pod indeksima: [0][0], [1][1], [2][2]. Dio programskog koda koji se odnosi na ispis po dijagonali je:

```
#ispis po glavnoj dijagonali
print ("Ispis po glavnoj dijagonali: ")
print(matrica[0][0])
print(matrica[1][1])
print(matrica[2][2])
```

Kroz zadane indekse prolazi dijagonala.

```
Ispis po glavnoj dijagonali:
1
5
9
```


Primjer 3:

Sljedeći program je konkretan prikaz primjene matrica u svrhu tabličnog skladištenja podataka. Sljedeća matrica skladišti podatke kao što su jmbg, ime, prezime. Ujedno ispisuje posljednji element matrice na indeksu [2][2].

```
datoteka = []
datoteka.append(["1234567890", "Pero", "Peric"])
datoteka.append(["0987654321", "Jozo", "Jozic"])
datoteka.append(["5432167890", "Ante", "Antic"])
print "Ispis cijele liste s podlistama (tj. datoteke):"
print datoteka
print "Ispis red po red (tj. slog po slog):"
for slog in datoteka:
    print slog
print "Cisti ispis svih polja - bez zagrada, zareza i sl.: "
print "JMBAG, Ime, Prezime"
print "-----"
for slog in datoteka:
    for polje in slog:
        print polje,
    print
print
print "Ljepse oblikovan ispis sadržaja (po dva tabulatora medju stupcima): "
print "JMBAG\t\tIme\t\tPrezime"
print "-----\t\t---\t\t-----"
for slog in datoteka:
    for polje in slog:
        print polje, "\t\t",
    print

print "Zadnje polje u zadnjem slogu: ", datoteka[2][2]
```

U odnosu na prethodni primjer ovaj primjer je nadograđen na način da je u matricu dodan tekstualni zapis, a ne brojevi. Ispis bi trebao izgledati ovako:

```
Ispis cijele liste s podlistama (tj. datoteke):
[['1234567890', 'Pero', 'Peric'], ['0987654321', 'Jozo', 'Jozic'], ['5432167890', 'Ante', 'Antic']]
Ispis red po red (tj. slog po slog):
['1234567890', 'Pero', 'Peric']
['0987654321', 'Jozo', 'Jozic']
['5432167890', 'Ante', 'Antic']
Cisti ispis svih polja - bez zagrada, zareza i sl.:
JMBAG, Ime, Prezime
-----
1234567890 Pero Peric
0987654321 Jozo Jozic
5432167890 Ante Antic

Ljepse oblikovan ispis sadržaja (po dva tabulatora medju stupcima):
JMBAG      Ime      Prezime
-----
1234567890      Pero      Peric
0987654321      Jozo      Jozic
5432167890      Ante      Antic
Zadnje polje u zadnjem slogu: Antic
```

Ujedno ispisan je zadnji element matrice **Antic** na lokaciji retka i stupca [2] [2].