

# **Programmazione in Python**

## **strutture dati multidimensionali**

Dario Pescini - Mirko Cesarini

# liste bidimensionali

Una lista bidimensionale può essere considerata una **lista di**  
**liste** organizzate sequenzialmente:

**M** = **A**  
      **B**  
      **C**  
      **D**  
      ...  
      **Z**

# liste bidimensionali

Una lista bidimensionale può essere considerata una **lista di**  
**liste** organizzate sequenzialmente:

```
M = M[0]  
    M[1]  
    M[2]  
    M[3]  
    ...  
    M[m-1]
```

# liste bidimensionali

$M[0] = A[0] A[1] A[2] A[3] \dots A[n-1]$

$M[1] = B[0] B[1] B[2] B[3] \dots B[n-1]$

...

$M[m-1] = Z[0] Z[1] Z[2] Z[3] \dots Z[n-1]$

ovvero:

$M[0] = M[0][0] M[0][1] M[0][2] M[0][3] \dots M[0][n-1]$

$M[1] = M[1][0] M[1][1] M[1][2] M[1][3] \dots M[1][n-1]$

...

$M[m-1] = M[m-1][0] M[m-1][1] \dots M[m-1][n-1]$

## lista di liste: esempio

Durante una votazione è stato eseguito lo spoglio delle schede in più seggi e se ne sono raccolti i dati.

I dati hanno la seguente struttura:

Seggio	Strumento	Bianche	Nulle	Bush	Gore	Browne	Nader	Harris	Hagelin	Buchanan
Alachua	Optical	1	217	105	34124	47365	658	3226	6	42
Baker	Optical	1	79	46	5610	2392	17	53	0	3
Bay	Optical	1	541	141	38637	18850	171	828	5	18
Bradford	Optical	2	41	695	5414	3075	28	84	0	2
Brevard	Optical	1	277	136	115185	97318	643	4470	11	39
Broward	Votomatic	1	4946	7826	177902	387703	1217	7104	54	135
...	...	...	...	...	...	...	...	...	...	...

Che può essere immagazinata in una lista di liste: `[[ ], ..., [ ] ]`

# lista di liste: esempio

Ovvero:

```
votiSeggi = [['Alachua', 'Optical', 1, 217, 105, 34124, 47365,
              658, 3226, 6, 42, 263, 4, 20, 21, 0, 0],
             ['Baker', 'Optical', 1, 79, 46, 5610, 2392, 17,
              53, 0, 3, 73, 0, 3, 3, 0, 0],
             ['Bay', 'Optical', 1, 541, 141, 38637, 18850,
              171, 828, 5, 18, 248, 3, 18, 27, 0, 0],
             ['Bradford', 'Optical', 2, 41, 695, 5414,
              3075, 28, 84, 0, 2, 65, 0, 2, 3, 0, 0],
             ['Brevard', 'Optical', 1, 277, 136, 115185,
              97318, 643, 4470, 11, 39, 570, 11, 72,
              76, 0, 0],
             ['Broward', 'Votomatic', 1, 4946, 7826, 177902,
              387703, 1217, 7104, 54, 135, 795, 37, 74, 122,
              0, 0],
             ....
             ]
```

## lista di liste: esempio

Per controllo, viene richiesto di effettuare il conteggio delle schede consegnate per ciascun seggio:

Seggio	Strumento	Bianche	Nulle	Bush	Gore	...
Alachua	Optical	1	217	105	34124	...
Baker	Optical	1	79	46	5610	...
Bay	Optical	1	541	141	38637	...
Bradford	Optical	2	41	695	5414	...
Brevard	Optical	1	277	136	115185	...
Broward	Votomatic	1	4946	7826	177902	...
...	...	...	...	...	...	...

## lista di liste: esempio

Per controllo, viene richiesto di effettuare il conteggio delle schede consegnate per ciascun seggio:

Seggio	Strumento	Bianche	Nulle	Bush	Gore	...
Alachua	Optical	1	217	105	34124	...
Baker	Optical	1	79	46	5610	...
Bay	Optical	1	541	141	38637	...
Bradford	Optical	2	41	695	5414	...
Brevard	Optical	1	277	136	115185	...
Broward	Votomatic	1	4946	7826	177902	...
...	...	...	...	...	...	...

che si ottiene sommando gli elementi opportuni per ciascuna riga.



## lista di liste: esempio

Le liste di liste permettono di automatizzare il processo tramite gli indici:

$$\begin{aligned} M[0] &= M[0][0] M[0][1] M[0][2] M[0][3] \dots M[0][n-1] \\ M[1] &= M[1][0] M[1][1] M[1][2] M[1][3] \dots M[1][n-1] \\ M[2] &= M[2][0] M[2][1] M[2][2] M[2][3] \dots M[2][n-1] \\ M[3] &= M[3][0] M[3][1] M[3][2] M[3][3] \dots M[3][n-1] \\ M[4] &= M[4][0] M[4][1] M[4][2] M[4][3] \dots M[4][n-1] \\ M[5] &= M[5][0] M[5][1] M[5][2] M[5][3] \dots M[5][n-1] \\ &\dots \end{aligned}$$

## lista di liste: esempio

Le liste di liste permettono di automatizzare il processo tramite gli indici:

$$\begin{aligned} M[0] &= M[0][0] \ M[0][1] \ M[0][2] \ M[0][3] \ \dots \ M[0][n-1] \\ M[1] &= M[1][0] \ M[1][1] \ M[1][2] \ M[1][3] \ \dots \ M[1][n-1] \\ M[2] &= M[2][0] \ M[2][1] \ M[2][2] \ M[2][3] \ \dots \ M[2][n-1] \\ M[3] &= M[3][0] \ M[3][1] \ M[3][2] \ M[3][3] \ \dots \ M[3][n-1] \\ M[4] &= M[4][0] \ M[4][1] \ M[4][2] \ M[4][3] \ \dots \ M[4][n-1] \\ M[5] &= M[5][0] \ M[5][1] \ M[5][2] \ M[5][3] \ \dots \ M[5][n-1] \end{aligned}$$

...

il primo per le righe

## lista di liste: esempio

Le liste di liste permettono di automatizzare il processo tramite gli indici:

$$\begin{aligned} M[0] &= M[0][0] M[0][1] M[0][2] M[0][3] \dots M[0][n-1] \\ M[1] &= M[1][0] M[1][1] M[1][2] M[1][3] \dots M[1][n-1] \\ M[2] &= M[2][0] M[2][1] M[2][2] M[2][3] \dots M[2][n-1] \\ M[3] &= M[3][0] M[3][1] M[3][2] M[3][3] \dots M[3][n-1] \\ M[4] &= M[4][0] M[4][1] M[4][2] M[4][3] \dots M[4][n-1] \\ M[5] &= M[5][0] M[5][1] M[5][2] M[5][3] \dots M[5][n-1] \end{aligned}$$

...

## lista di liste: esempio

Le liste di liste permettono di automatizzare il processo tramite gli indici:

```
M[0] = M[0][0] M[0][1] M[0][2] M[0][3] ... M[0][n-1]
M[1] = M[1][0] M[1][1] M[1][2] M[1][3] ... M[1][n-1]
M[2] = M[2][0] M[2][1] M[2][2] M[2][3] ... M[2][n-1]
M[3] = M[3][0] M[3][1] M[3][2] M[3][3] ... M[3][n-1]
M[4] = M[4][0] M[4][1] M[4][2] M[4][3] ... M[4][n-1]
M[5] = M[5][0] M[5][1] M[5][2] M[5][3] ... M[5][n-1]
...
```

il secondo per le colonne

## lista di liste: esempio

Le liste di liste permettono di automatizzare il processo tramite gli indici:

$$\begin{aligned} M[0] &= M[0][0] \ M[0][1] \ M[0][2] \ M[0][3] \ \dots \ M[0][n-1] \\ M[1] &= M[1][0] \ M[1][1] \ M[1][2] \ M[1][3] \ \dots \ M[1][n-1] \\ M[2] &= M[2][0] \ M[2][1] \ M[2][2] \ M[2][3] \ \dots \ M[2][n-1] \\ M[3] &= M[3][0] \ M[3][1] \ M[3][2] \ M[3][3] \ \dots \ M[3][n-1] \\ M[4] &= M[4][0] \ M[4][1] \ M[4][2] \ M[4][3] \ \dots \ M[4][n-1] \\ M[5] &= M[5][0] \ M[5][1] \ M[5][2] \ M[5][3] \ \dots \ M[5][n-1] \end{aligned}$$

...

## lista di liste: esempio

Le liste di liste permettono di automatizzare il processo tramite gli indici:

$$\begin{aligned} M[0] &= M[0][0] \ M[0][1] \ M[0][2] \ M[0][3] \ \dots \ M[0][n-1] \\ M[1] &= M[1][0] \ M[1][1] \ M[1][2] \ M[1][3] \ \dots \ M[1][n-1] \\ M[2] &= M[2][0] \ M[2][1] \ M[2][2] \ M[2][3] \ \dots \ M[2][n-1] \\ M[3] &= M[3][0] \ M[3][1] \ M[3][2] \ M[3][3] \ \dots \ M[3][n-1] \\ M[4] &= M[4][0] \ M[4][1] \ M[4][2] \ M[4][3] \ \dots \ M[4][n-1] \\ M[5] &= M[5][0] \ M[5][1] \ M[5][2] \ M[5][3] \ \dots \ M[5][n-1] \\ &\dots \end{aligned}$$

$M[i][j]$ : riga  $i$ , colonna  $j$

## lista di liste: esempio

“effettuare il conteggio delle schede consegnate per ciascun seggio”:

è necessaria una *somma per ogni colonna* di ciascuna *riga*

$$\text{somma} = M[0][0] + M[0][1] + M[0][2] + M[0][3] + \dots$$

## lista di liste: esempio

`somma = M[0][0] + M[0][1] + M[0][2] + M[0][3] + ...`

ovvero

```
sommaRiga = 0
colonna = 0
while colonna < len(M[0]):
    sommaRiga += M[0][colonna]
    colonna += 1
```



## lista di liste: esempio

$M[0] = M[0][0] \ M[0][1] \ M[0][2] \ M[0][3] \ \dots \ M[0][n-1]$   
 $M[1] = M[1][0] \ M[1][1] \ M[1][2] \ M[1][3] \ \dots \ M[1][n-1]$   
 $M[2] = M[2][0] \ M[2][1] \ M[2][2] \ M[2][3] \ \dots \ M[2][n-1]$   
...

ma abbiamo più righe:

```
riga = 0
while riga < len(M):
    sommaRiga = 0
    colonna = 0
    while colonna < len(M[riga]):
        sommaRiga += M[riga][colonna]
        colonna += 1
    riga += 1
```

# Il codice

Lo schema appena visto applicato al caso di studio genera il seguente codice:

```
# votiSeggi = [[], ... ,[]]
candidati = ['Bianche', 'Nulle', 'Bush', 'Gore', 'Browne',
             'Nader',
             'Harris', 'Hagelin', 'Buchanan', 'McReynolds',
             'Phylips',
             'Moorehead', 'Chote', 'McCharthy']

nCandidati = len(candidati)
nSeggi = len(votiSeggi)

seggio = 0
while seggio < nSeggi:
    candidato = 2
    schede = 0
    while candidato < nCandidati:
        schede += votiSeggi[seggio][candidato]
        candidato += 1
    print 'totale schede seggio ', votiSeggi[seggio][0], schede
    seggio += 1
```

## lista di liste: esempio

Per lo spoglio, viene richiesto di effettuare il conteggio delle preferenze ottenute per ciascun candidato:

Seggio	Strumento	Bianche	Nulle	Bush	Gore	...
Alachua	Optical	1	217	105	34124	...
Baker	Optical	1	79	46	5610	...
Bay	Optical	1	541	141	38637	...
Bradford	Optical	2	41	695	5414	...
Brevard	Optical	1	277	136	115185	...
Broward	Votomatic	1	4946	7826	177902	...
...	...	...	...	...	...	...

## lista di liste: esempio

Per lo spoglio, viene richiesto di effettuare il conteggio delle preferenze ottenute per ciascun candidato:

Seggio	Strumento	Bianche	Nulle	Bush	Gore	...
Alachua	Optical	1	217	105	34124	...
Baker	Optical	1	79	46	5610	...
Bay	Optical	1	541	141	38637	...
Bradford	Optical	2	41	695	5414	...
Brevard	Optical	1	277	136	115185	...
Broward	Votomatic	1	4946	7826	177902	...
...	...	...	...	...	...	...

che si ottiene sommando gli elementi opportuni per ciascuna colonna.

## lista di liste: esempio

Le liste di liste permettono di automatizzare il processo tramite gli indici:

$$\begin{aligned} M[0] &= M[0][0] \ M[0][1] \ M[0][2] \ M[0][3] \ \dots \ M[0][n-1] \\ M[1] &= M[1][0] \ M[1][1] \ M[1][2] \ M[1][3] \ \dots \ M[1][n-1] \\ M[2] &= M[2][0] \ M[2][1] \ M[2][2] \ M[2][3] \ \dots \ M[2][n-1] \\ M[3] &= M[3][0] \ M[3][1] \ M[3][2] \ M[3][3] \ \dots \ M[3][n-1] \\ M[4] &= M[4][0] \ M[4][1] \ M[4][2] \ M[4][3] \ \dots \ M[4][n-1] \\ M[5] &= M[5][0] \ M[5][1] \ M[5][2] \ M[5][3] \ \dots \ M[5][n-1] \\ &\dots \end{aligned}$$

$M[i][j]$ : riga  $i$ , colonna  $j$

## lista di liste: esempio

“effettuare il conteggio delle preferenze ottenute per ciascun candidato”:

è necessaria una **somma per ogni riga** di ciascuna **colonna**

```
somma = M[0][0] +  
         M[1][0] +  
         M[2][0] +  
         M[3][0] +  
         ...
```

## lista di liste: esempio

```
somma = M[0][0] +  
        M[1][0] +  
        M[2][0] +  
        M[3][0] +  
        ...
```

ovvero

```
sommaColonna = 0  
riga = 0  
while riga < len(M):  
    sommaColonna += M[riga][0]  
    riga += 1
```

## lista di liste: esempio

$M[0] = M[0][0] \ M[0][1] \ M[0][2] \ M[0][3] \ \dots \ M[0][n-1]$   
 $M[1] = M[1][0] \ M[1][1] \ M[1][2] \ M[1][3] \ \dots \ M[1][n-1]$   
 $M[2] = M[2][0] \ M[2][1] \ M[2][2] \ M[2][3] \ \dots \ M[2][n-1]$   
...

ma abbiamo più colonne:

```
colonna = 0
while colonna < len(M[0]):
    sommaColonna = 0
    riga = 0
    while riga < len(M):
        sommaColonna += M[riga][colonna]
        riga += 1
    colonna += 1
```



# Il codice

Lo schema appena visto applicato al caso di studio genera il seguente codice:

```
# votiSeggi = [[], ... ,[]]
candidati = ['Bianche', 'Nulle', 'Bush', 'Gore', 'Browne',
             'Nader',
             'Harris', 'Hagelin', 'Buchanan', 'McReynolds',
             'Phylips',
             'Moorehead', 'Chote', 'McCharthy']

nCandidati = len(candidati)
nSeggi = len(votiSeggi)

candidato = 0
while candidato < nCandidati:
    seggio = 0
    preferenze = 0
    while seggio < nSeggi:
        preferenze += votiSeggi[seggio][candidato + 2]
        seggio += 1
    print 'totale preferenze candidato ',
    candidati[candidato], preferenze
    candidato += 1
```

# Strutture di strutture

Nulla vieta di mescolare le strutture dati da comporre:

- **Lista** di **liste**:  

```
lista = [lista1, lista2, ...]  
      = [[ , , ...], [ , , ...], ...]
```
- **Liste** di **tuple**:  

```
lista = [tupla1, tupla2, ...]  
      = [ ( , , ...), ( , , ...), ...]
```
- **Dizionari** di **tuple**:  

```
diz = {tupla1, tupla2, ...}  
     = { t1:( , , ...), t2:( , , ...), ...}
```

# Esempio

```
trms=[ #(ingrediente, grammi, calorie_ogni_100_grammi)
      ('savoardi',300,391),
      ('zucchero',100,392),
      ('uova',220,130),
      ('caffè',300,6),
      ('mascarpone',500,460),
      ('cacao',3,320) ]
```

```
i=0
totCal=0
while i<len(trms):
    grammi=trms[i][1]
    cal100=trms[i][2]
    totCal+=grammi*cal100 / 100.0
    i+=1
print('Calorie totali tiramisu', totCal)
```

```
('Calorie totali tiramisu', 4178.6)
```

# Esempio

```
trms=[ #(ingrediente, grammi, calorie_ogni_100_grammi)
      ('savoardi',300,391),
      ('zucchero',100,392),
      ('uova',220,130),
      ('caffè',300,6),
      ('mascarpone',500,460),
      ('cacao',3,320) ]

totCal=0
totGram=0
for el in trms:
    grammi=el[1]
    cal100=el[2]
    totCal+=grammi*cal100 / 100.0
    totGram+=grammi
print('Calorie ogni 100 gr di tiramisu', int(totCal /
    float(totGram) * 100 + 0.5))
```

```
('Calorie ogni 100 gr di tiramisu', 294)
```