

Introduzione alle Variabili in MATLAB

Table of Contents

1. Variabili numeriche.....	1
2. Operazioni tra variabili.....	3
3. Funzioni built-in.....	4
4. Variabili di testo.....	5

1. Variabili numeriche

In MATLAB il tipo di una variabile viene definito contestualmente all'assegnazione del suo valore.

Esempio:

```
% Alloca la memoria per la variabile "a"  
% assegnandole il valore 6  
a = 6;  
disp(['a = ' num2str(a)])
```

```
a = 6
```

In MATLAB, ogni variabile è interpretata come una matrice.

Esempio:

```
% Si può vedere lo scalare "a", definito prima, come una matrice 1x1,  
% per cui si può accedere all'elemento di tale matrice come:  
b = a(1,1);  
disp(['b = ' num2str(b)])
```

```
b = 6
```

Le **matrici riga** sono matrici con 1 riga e N colonne, e si definiscono immettendo gli N numeri separati da spazio, dentro parentesi quadre.

Esempio:

```
% Alloca la variabile "r" come matrice riga di 4 elementi  
r = [1.0 4.3 0.4 -2.0];  
disp(['r = [' num2str(r) '']])
```

```
r = [1      4.3      0.4      -2]
```

Le **matrici colonna**, invece, sono costituite da una colonna e N righe; si definiscono immettendo i numeri tra parentesi quadre separati da "punto e virgola".

Esempio:

```
% Alloca la variabile "c" come matrice colonna di 4 elementi  
c = [1.0; 4.3; 0.4; -2.0];  
disp('c = '); disp(c);
```

```
c =  
1.0000
```

```
4.3000
0.4000
-2.0000
```

Le **matrici** hanno N righe e M colonne, e si definiscono scrivendo tra parentesi quadre le righe separate da "punto e virgola".

Esempio:

```
% Alloggia la variabile "A" come matrice di 2 righe e 3 colonne
A = [-1.0 1.0 2.4; 2.3 -0.4 9.9];
disp('A = '); disp(A)
```

```
A =
-1.0000    1.0000    2.4000
 2.3000   -0.4000    9.9000
```

Per accedere ad un elemento di una matrice si scrive il nome della variabile seguito da parentesi tonde nelle quali, separati da virgola, vanno indicati l'indice di riga e quello di colonna. Tali indici vanno da 1 al numero di righe/colonne.

Esempio:

```
% Si assegna alla variabile "a" l'elemento della matrice A
% nella seconda colonna della prima riga
a = A(1,2);
disp(['A(1,2) = ' num2str(a)])
```

```
A(1,2) = 1
```

La stessa scrittura si può utilizzare per accedere in scrittura agli elementi della matrice.

Esempio:

```
% Modificare l'elemento della terza colonna della seconda riga
% da 9.9 a -8.0:
A(2,3) = -8.0;
disp('Matrice modificata'); disp(' '); disp('A = '); disp(A)
```

```
Matrice modificata
```

```
A =
-1.0000    1.0000    2.4000
 2.3000   -0.4000   -8.0000
```

Si può accedere ad una porzione della matrice usando l'operatore che serve a definire gli intervalli, ossia "due punti". Inserire ":" al posto dell'indice di riga (o di colonna) significa prendere tutta la riga (o tutta la colonna). Alternativamente, al posto dell'indice di riga (o di colonna) indicare due numeri separati dai ":" significa prendere una porzione di riga (o di colonna) inclusa tra i due numeri.

Esempio:

```
% Copiare nella variabile "B" la parte di matrice "A" che prenda
% tutte le righe, ma solo le prime due colonne
B = A(:,1:2);
```

```
disp('B = '); disp(B)
```

```
B =  
-1.0000    1.0000  
 2.3000   -0.4000
```

2. Operazioni tra variabili

Quando si effettua un'operazione tra due variabili, il MATLAB riconosce se sono scalari o matrici. Nel secondo caso, applica le regole delle matrici per svolgere l'operazione richiesta.

Esempio:

```
% Decomentare il comando seguente ed eseguire questa cella. Il MATLAB darà  
% errore perché la somma tra matrici che hanno dimensioni differenti  
% non è definita. Per proseguire, commentare nuovamente il comando.  
  
%s = A + r
```

Attenzione, però, alle operazioni tra matrici riga e matrici colonna: il MATLAB può interpretare la moltiplicazione e la somma intese come diadiche.

Esempio:

```
% "r" è una matrice riga di 4 elementi, mentre "c" è una matrice colonna  
% di 4 elementi. Moltiplicando r per c, il MATLAB applica la regola di  
% moltiplicazione tra matrici, essendo "r" e "c" di dimensioni compatibili  
  
prc = r * c;  
disp(['prc = ' num2str(prc)])
```

```
prc = 23.65
```

```
% Scambiando l'ordine di moltiplicazione, però, essendo le matrici colonna  
% e riga, il MATLAB interpreta tale prodotto come diadico  
  
pcr = c * r;  
disp(['pcr =']); disp(pcr)
```

```
pcr =  
 1.0000    4.3000    0.4000   -2.0000  
 4.3000   18.4900    1.7200   -8.6000  
 0.4000    1.7200    0.1600   -0.8000  
-2.0000   -8.6000   -0.8000    4.0000
```

3. Funzioni *built-in*

Può tornare utile conoscere che esistono le funzioni: `max`, `min`, `sum`, `mean`, e `std` che calcolano (per colonna, a meno che non siano applicate a una matrice riga), rispettivamente, il massimo, il minimo, la somma, la media e la deviazione standard di ogni colonna della matrice data in ingresso.

Esempi:

```
% Le righe seguenti mostrano l'effetto delle funzioni sopra riportate
% sulla matrice A. L'operazione viene eseguita per ogni colonna.
```

```
disp('A = '); disp(A)
```

```
A =
-1.0000    1.0000    2.4000
 2.3000   -0.4000   -8.0000
```

```
ma = max(A);
disp('max(A) = '); disp(ma)
```

```
max(A) =
 2.3000    1.0000    2.4000
```

```
mi = min(A);
disp('min(A) = '); disp(mi)
```

```
min(A) =
-1.0000   -0.4000   -8.0000
```

```
s = sum(A);
disp('sum(A) = '); disp(s)
```

```
sum(A) =
 1.3000    0.6000   -5.6000
```

```
m = mean(A);
disp('mean(A) = '); disp(m)
```

```
mean(A) =
 0.6500    0.3000   -2.8000
```

```
sigma = std(A);
disp('std(A) = '); disp(sigma)
```

```
std(A) =
 2.3335    0.9899    7.3539
```

Si può trasporre una matrice con la funzione `transpose`.

Esempio:

```
disp('A = '); disp(A)
```

```
A =
-1.0000    1.0000    2.4000
 2.3000   -0.4000   -8.0000
```

```
At = transpose(A);
disp('A^tr = '); disp(At)
```

```
A^tr =
   -1.0000    2.3000
    1.0000   -0.4000
    2.4000   -8.0000
```

Si può calcolare l'aggiunta di una matrice con l'uso dell'apice "'"

```
% Alloca una matrice complessa (1j è l'unità complessa i)
B = [1j, 3-1j*2; 1-1j 4];
disp('B = '); disp(B)
```

```
B =
   0.0000 + 1.0000i   3.0000 - 2.0000i
   1.0000 - 1.0000i   4.0000 + 0.0000i
```

```
% Calcola l'aggiunta di B
Badj = B';
disp('B'' = '); disp(Badj)
```

```
B' =
   0.0000 - 1.0000i   1.0000 + 1.0000i
   3.0000 + 2.0000i   4.0000 + 0.0000i
```

4. Variabili di testo

Una variabile di testo viene definita se il valore assegnatole è incluso tra apici.

Esempio:

```
t = 't è variabile di testo';
disp(t)
```

```
t è variabile di testo
```

Si possono convertire le variabili numeriche in testo e viceversa.

Esempi:

```
% num2str() è una funzione che converte un numero in testo
a = 1.5;
ta = num2str(a);
disp(['a = ' ta])
```

```
a = 1.5
```

```
% str2num() è una funzione che converte un testo in numero
t = '2.0';
b = str2num(t);
c = 2 * b;
disp(['Il doppio di t vale ' num2str(c)])
```

```
Il doppio di t vale 4
```