



Rappresentare i dati in Matlab

Informatica B

Creazione ed inizializzazione di una variabile

- ❑ Le variabili sono create al momento della loro **inizializzazione**
- ❑ L'inizializzazione consiste nell'attribuire un valore ad una variabile **per la prima volta**:

```
nome = valore
```

- ❑ Il *nome* di una variabile deve iniziare con una lettera e può contenere solo lettere, numeri e underscore (_)
- ❑ Altri metodi di inizializzazione
 - ▶ Lettura dati da tastiera
 - ▶ Lettura da file

Rappresentare i numeri

Variabili numeriche (double)

- ❑ In Matlab una variabile che contiene un valore numerico è definita **double**
- ❑ Una variabile double può contenere valori numerici
 - ▶ reali, e.g., `var1 = -10.7;`
 - ▶ immaginari, e.g., `var2 = 4i; var3 = 4j;`
 - ▶ complessi, e.g., `var3 = 10.3 + 10i.`
- ❑ Le parti reali e immaginarie possono essere positive e negative nell'intervallo di valori $[10^{-308}, 10^{308}]$, con accuratezza di 15-16 cifre decimali

- ❑ Matlab consente di effettuare sulle variabili tutte le basilari operazioni aritmetiche

```
a=5
```

```
b=a+3 %b <- 8
```

```
c=b*4 %c <- 32
```

```
b=(c-5)/10 %b <- 2.7
```

```
c=1.25e-7 %c <- 1.25x10-7
```

```
c=a^3 %c <- 125
```

- ❑ Un **array** è una variabile che contiene una sequenza di valori dello stesso tipo
- ❑ Un array è organizzato come una tabella: ciascun valore è contenuto in una **cella** con un indice di **riga** e uno di **colonna**.
- ❑ In particolare, gli array che hanno **una sola riga** (o una **sola colonna**) sono chiamati **vettori**, mentre quelli con più di una riga e più di una colonna sono chiamati **matrici**.
- ❑ In Matlab, tutte le variabili sono gestite come se fossero array: le variabili contenenti un solo valore (dette anche **scalari**) sono infatti considerati **array con una sola riga ed una sola colonna**.

Array: assegnamento

□ Scalari

nome = valore

▶ a = 3

▶ b = 55

□ Vettori

nome = [v₁ v₂ ... v_n]

▶ a=[3 4 2 3] %valido anche a=[3,4,2,3]

□ Matrici

nome = [v₁₁ v₁₂ ... v_{1n}; ...; v_{m1} v_{m2} ... v_{mn}]

▶ a = [3 4 2; 4 5 6]


$$\begin{bmatrix} 3 & 4 & 2 \\ 4 & 5 & 6 \end{bmatrix}$$

- È possibile accedere ad un elemento di un array così:

nome(*<pos>*)

- ▶ se la variabile è un vettore, *<pos>* è un numero intero (≥ 1) che indica la posizione dell'elemento voluto
- ▶ se la variabile è una matrice, *<pos>* è una coppia di interi *<i,j>* che indicano la riga e la colonna dell'elemento
- ▶ se si desidera accedere a più elementi di un array, *<pos>* può essere un array di posizioni (o due array contenenti gli indici di riga e colonna nel caso di una matrice)

□ Esempi

▶ $a = [0 \ 7+1] \quad \%a = [0 \ 8]$

▶ $b = [a(2) \ 5 \ a] \quad \%b = [8 \ 5 \ 0 \ 8]$

← contenuto di a

← secondo elemento di a

□ Non tutti gli elementi devono essere specificati alla creazione...

▶ $c(2, 3) = 5 \rightarrow \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 5 \end{bmatrix}$

□ L'array può essere esteso successivamente ...

▶ $d = [2 \ 5]$

▶ $d(4) = 2 \quad \%d = [2 \ 5 \ 0 \ 2]$

Trasposizione e operatore :

□ Operatore di trasposizione

nome'

▶ Esempi

d = [2 5 0 2]

g = d'

$\begin{bmatrix} 2 \\ 5 \\ 0 \\ 2 \end{bmatrix}$

□ Operatore :

min:passo:max

x = 1:2:10

→ x = [1 3 5 7 9]

l = 1:3

m = [l' l']

$\begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{bmatrix}$

Array: funzioni di libreria

Funzione	Significato
<code>zeros (n)</code>	Genera una matrice nxn di zeri
<code>zeros (m,n)</code>	Genera una matrice mxn di zeri
<code>zeros (size(arr))</code>	Genera una matrice di zeri della stessa dimensione di arr
<code>ones(n)</code>	Genera una matrice nxn di uno
<code>ones(m,n)</code>	Genera una matrice mxn di uno
<code>ones(size(arr))</code>	Genera una matrice di uno della stessa dimensione di arr
<code>eye(n)</code>	Genera la matrice identità nxn
<code>eye(m,n)</code>	Genera la matrice identità mxn
<code>length(arr)</code>	Ritorna la dimensione più lunga del vettore
<code>size(arr)</code>	Ritorna il numero di righe e colonne dell'array

Array: esempi uso funzioni

`a = zeros(2)` \longrightarrow $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

`b = zeros(2,3)` \longrightarrow $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$

`c = [1 2; 3 4]`

`d = zeros(size(c))` \longrightarrow $\begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix}$

Array: altri esempi

- Esempio

▶ $m(1:4, 1:3) = 3 \longrightarrow \begin{bmatrix} 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix}$

- Regola: il modo con cui uno scalare viene assegnato ad un array dipende dalla forma dell'array che viene specificata a sinistra dell'assegnamento

- Esempio

▶ $m(1:2, 1:2) = 4 \longrightarrow \begin{bmatrix} 4 & 4 & 3 \\ 4 & 4 & 3 \\ 3 & 3 & 3 \\ 3 & 3 & 3 \end{bmatrix}$

- ... la regola si applica anche ai sottoarray

Array: operazioni aritmetiche

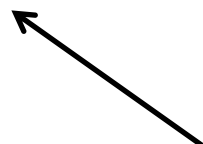
- ❑ Operazioni per gli scalari: + - * / ^
- ❑ Operazioni per gli array
 - ▶ Array operation: viene eseguita sugli elementi degli array coinvolti (devono avere lo stesso numero di righe e colonne)

$$a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad b = \begin{bmatrix} 2 & 3 \\ 5 & 7 \end{bmatrix} \quad a+b = \begin{bmatrix} 3 & 5 \\ 8 & 11 \end{bmatrix} \quad a.*b = \begin{bmatrix} 2 & 6 \\ 15 & 28 \end{bmatrix}$$

- ▶ Matrix operation: segue le regole dell'algebra lineare

$$a = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad b = \begin{bmatrix} 2 & 3 \\ 5 & 7 \end{bmatrix} \quad a*b = \begin{bmatrix} 12 & 17 \\ 26 & 37 \end{bmatrix}$$

$\sum_k a_{ik} * b_{kj}$



Array: operazioni aritmetiche (2)

Operazione	Sintassi Matlab	Commenti
Array addition	$a + b$	Array e matrix addition sono identiche
Array subtraction	$a - b$	Array e matrix subtraction sono identiche
Array multiplication	$a .* b$	Ciascun elemento del risultato è pari al prodotto degli elementi corrispondenti nei due operandi
Matrix multiplication	$a * b$	Prodotto di matrici
Array right division	$a ./ b$	$\text{risultato}(i,j) = a(i,j)/b(i,j)$
Array left division	$a .\ b$	$\text{risultato}(i,j) = b(i,j)/a(i,j)$
Matrix right division	a / b	$a * \text{inversa}(b)$
Matrix left division	$a \ b$	$\text{inversa}(a) * b$
Array exponentiation	$a .^ b$	$\text{risultato}(i,j) = a(i,j)^{b(i,j)}$

Array: altre funzioni

Funzione	Scopo
<code>ceil(x)</code>	approssima x all'intero immediatamente maggiore
<code>floor(x)</code>	approssima x all'intero immediatamente minore
<code>fix(x)</code>	approssima x all'intero più vicino verso lo zero
<code>max(x)</code>	ritorna il valore massimo nel vettore x e, opzionalmente, la collocazione di questo valore in x
<code>min(x)</code>	ritorna il valore minimo nel vettore x e, opzionalmente, la collocazione di questo valore nel vettore
<code>mod(m,n)</code>	$\text{mod}(x,y)$ è $x - n \cdot y$ dove $n = \text{floor}(x./y)$ se $y \neq 0$
<code>round(x)</code>	approssima x all'intero più vicino
<code>rand(N)</code>	genera una matrice di NxN numeri casuali

Rappresentare i caratteri

- ❑ Una variabile Matlab che contiene uno o più caratteri è chiamata **char**
- ❑ I valori caratteri in Matlab devono essere racchiusi tra `' '`
`v = 'a' %assegna alla variabile v il carattere 'a'`
`v = a %assegna a v il contenuto della variabile a`
- ❑ Un array di caratteri, chiamato **stringa**, in Matlab può essere rappresentato semplicemente come una sequenza di caratteri racchiusi fra `' '`
`s = ['c', 'i', 'a', 'o'] %s è un array di 4 caratteri`
`s = 'ciao' %notazione semplificata per le stringhe`

- ❑ Matlab definisce un insieme di variabili predefinite (e.g., pi)
- ❑ Queste variabili corrispondono in qualche caso a costanti

- ▶ **Attenzione!** Il valore di queste variabili può essere modificato, per esempio

```
circ1=2*pi*10
```

```
pi = 3
```

```
circ2=2*pi*10
```

- ▶ Il valore di circ2 non sarà più la circonferenza di un cerchio
- ❑ E` fortemente sconsigliato modificare il valore di una variabile predefinita

Variabili predefinite più comuni

Variabile	Scopo
pi	contiene 15 cifre significative di π
i, j	contiene il valore i ($\sqrt{-1}$)
inf (o Inf)	rappresentazione dell'infinito (ottenuto di solito come risultato di una divisione per 0)
nan	Not-A-Number è il risultato di una operazione matematica non definita, es 0/0
clock	contiene la data e l'orario corrente. E' un vettore di sei elementi (anno, mese, giorno, ora, minuti, secondi)
date	contiene la data corrente sotto forma di stringa
eps	epsilon: la più piccola differenza rappresentabile tra due numeri
ans	Variabile speciale usata per immagazzinare risultati non assegnati ad altre variabili