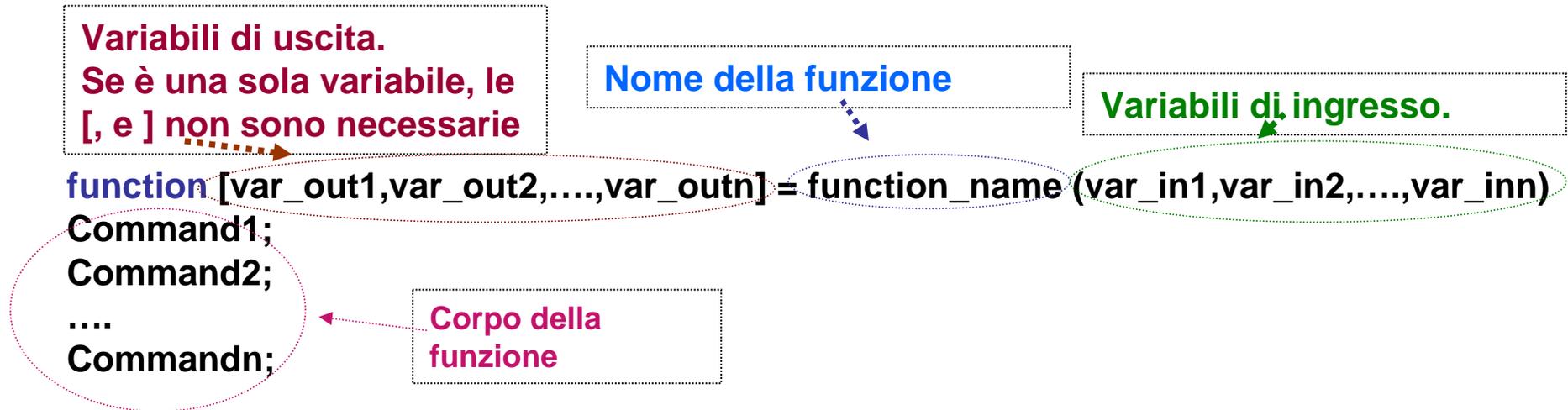


Funzioni definite dall'utente: sintassi

Definizione di funzione:

In un M-file, il cui nome è (bene che sia...vedi diapo successive): “**function_name.m**”



Chiamata di funzione:

Nell'M-file con il programma principale

```
[v1,v2,....,vn] = function_name (in1, in2,...., inn)
```

N.B.: Nome dell'M-file in cui è definita la funzione

Funzioni definite dall'utente: esempio

Definizione di funzione:

```
function y =seno_in_gradi(x)
% seno_in_gradi calcola il seno di un angolo in gradi
tmp = x*pi/180; %N.B.: tmp è una variabile LOCALE
y = sin(tmp);
```

Il nome dell'M-file con la definizione, è (bene che sia... vedi diapo successive):
seno_in_gradi.m

Chiamata di funzione:

```
% script che richiama la funzione seno_in_gradi, per diversi valori
% dell'angolo in gradi
x = input('valore dell'angolo in gradi:');
→ s = seno_in_gradi(x);
disp(['il seno di ' num2str(x) '° e': ' num2str(s)])
```

Funzioni definite dall'utente: regole

1. Chiamata di funzione:

Una funzione definita dall'utente è chiamata utilizzando **il nome dell'M-file in cui questa è definita non il nome della funzione, dato nella prima riga di definizione**

Se la funzione `seno_in_gradi` dell'esempio precedente venisse salvata in un M_file con nome "**`sin_in_degree.m`**", lasciando invariate le linee di codice,(e quindi anche il nome della funzione: **`seno_in_gradi`**), nell'M-file principale (per esempio "main.m") dovremo fare la seguente chiamata di funzione:

`y = sin_in_degree(x);`



Per comodità, e per evitare errori, è conveniente dare lo stesso nome alla funzione e all'M-file!

2. Commenti:

Nella definizione di funzione, le righe dopo la prima, dovrebbero essere commenti, che spiegano cosa fa la funzione che si sta definendo.

```
function y =seno_in_gradi(x)
```



```
% seno_in_gradi calcola il seno di un angolo in gradi
```

```
tmp = x*pi/180; %N.B.: tmp è una variabile LOCALE  
y = sin(tmp);
```

Tale commento può essere visto, utilizzando il comando "help nome_funzione":

```
>> help seno_in_gradi  
seno_in_gradi calcola il seno di un angolo in gradi
```



nella Command
Window

3. Informazioni in uscita:

La sola informazione in uscita alla chiamata di funzione, sono i valori delle variabili di uscita.



Nel corpo della funzione deve sempre esserci almeno un comando che assegna un valore alle variabili di uscita specificate nella prima linea di definizione.

Le variabili di uscita sono opzionali: si possono definire funzioni che eseguono operazioni, ma non restituiscono informazioni (per esempio “plot” o “disp”)

4. “comunicazione”:

Una funzione comunica con il workspace del Matlab solo attraverso le variabili di ingresso e di uscita.

Le variabili ‘intermedie’, cioè definite ed utilizzate nel corpo della funzione, non appaiono e non interagiscono con il Workspace principale del Matlab:

*ogni funzione ha un suo Workspace,
separato dal Workspace principale.*

Le variabili presenti nel Workspace locale
si dicono, appunto **Variabili Locali**

```
function y = seno_in_gradi(x)
```

```
% seno_in_gradi calcola il seno di un angolo in gradi
```

```
→ tmp = x*pi/180; %N.B.: tmp è una variabile LOCALE
```

```
y = sin(tmp);
```

N.B.: molti comandi Matlab sono scritti come M-files funzione.

Si può conoscere la cartella in cui tali M-files sono salvati mediante il comando:

which:

```
>> which linspace % comando per conoscere in quale cartella è l'M-file linspace.m  
C:\Programmi\MATLAB701\toolbox\matlab\elmat\linspace.m
```

Si può vedere il contenuto dell'M-file funzione con il comando ***type***

```
>>type linspace  
function y = linspace(d1, d2, d3,n)  
.....
```