```
>> diary laborario.txt
>> % questo e' un commento
>> % definisco la variabile x e le assegno il valore 10
>> x=10

x =

    10

>> % definisco y e gli assegno il valore 20 senza eco
>> y = 20;
>> % per vedere le variabili disponibili uso "whos"
>> whos
  Name      Size            Bytes  Class     Attributes

  x         1x1                 8  double
  y         1x1                 8  double

>> % dimensioni sono 1x1 perche' ogni variabile Matlab
>> % e' una matrice con n righe e m colonne
>> % n x m
>>
>>
>>
>>
>> % voglio vedere il contenuto di y
>> y

y =

    20

>> % assgno a x il contenuto di y:  x <-- y
>> x = y;
>> x

x =

    20

>> % definisco un vettore riga di elementi 1, 2, -1, 0
>> v = [1 2 -1 0]

v =

     1     2    -1     0

>> % definisco un vettore colonna di elementi -1 -2 -3
>> vc = [-1;-2;-3]

vc =

    -1
    -2
    -3

>> % voglio assegnare alla seconda posizione di vc il contenuto
>> % della terza poszione di v:  vc(2) <-- v(3)
>> vc(2)=v(3)

vc =

    -1
```

```
   -1
   -3

>> % altri modi per definire vettori particolari
>> % voglio il vettore 1 2 3 4 5 6 7 8 9 10
>> % GLI ELEMENTI DEL VETTORE SONO EQUISPAZIATI
>> w = 1:1:10

w =

    1    2    3    4    5    6    7    8    9   10

>> % voglio creare 3 5 7 9 11
>> w1 = 3:2:11

w1 =

    3    5    7    9   11

>> % posso crearli decrescenti: 9 6 3 0 -3
>> w2 = 9:-3:-3

w2 =

    9    6    3    0   -3

>> % cosa succede se prendo 2:4:20?
>> w3 = 2:4:20

w3 =

    2    6   10   14   18

>> % e se voglio crearli colonna?
>> % uso l'operator di trasposizione ' (trasposto coniugato)
>> % volgio creare 3:5:25 colonna
>> w4 = (3:5:25)'

w4 =

    3
    8
   13
   18
   23

>> % voglio creare un vettore che ha un certo numero di
>> % elementi di cui specifico il piu' piccolo ed
>> % il piu' grande e che siano equispaziati
>> % ad esempio, voglio 5 elementi equispaziati nell'intervallo
>> % [2, 20]
>> w5 = linspace(2,20,5)

w5 =

   2.0000    6.5000   11.0000   15.5000   20.0000

>> % cancelliamo la terza componente di w5
>> w(3)=[]

w =

    1    2    4    5    6    7    8    9   10
```

```
>> w5(3)=[]

w5 =

    2.0000    6.5000   15.5000   20.0000

>> % tolgo da w5 la 1 e la 3 componente
>> w5([1 3])=[]

w5 =

    6.5000   20.0000

>>
>>
>>
>>
>> % costruiamo il grafico di sin(x) nell'intervallo [a b]
>>
>>
>>
>>
>>
>> % vediamo le variabili presenti in memoria (nel Workspace
>> % di Matlab)
>> whos
  Name      Size            Bytes  Class     Attributes

  v         1x4                32  double
  vc        3x1                24  double
  w         1x9                72  double
  w1        1x5                40  double
  w2        1x5                40  double
  w3        1x5                40  double
  w4        5x1                40  double
  w5        1x2                16  double
  x         1x1                 8  double
  y         1x1                 8  double

>> % cancello le variabili col comando clear
>> clear
>> whos
>> % creiamo l'intervallo per il grafico
>> a = -pi; % estremo inferiore
>> b =  pi; % estremo superiore
>> % creo il vettore x che contiene le ascisse dei punti
>> % (x(i),y(i)) che verranno utilizzati per creare il grafico
>> % Brutalmente, Matlab congiunge con segmenti di retta i
>> % punti (x(i), y(i)) adiacenti
>>
>> x = linspace(a,b,200);
>> y = sin(x);
>> whos
  Name      Size            Bytes  Class     Attributes

  a         1x1                 8  double
  b         1x1                 8  double
  x         1x200            1600  double
  y         1x200            1600  double

>> % creo il grafico y=sin(x)
>> plot(x,y)
```

```
>> grid on
>> grid off
>> % aggiungo y=cos(x)
>> yc = cos(x);
>> plot(x,yc)
>> % per tenere i grafici gia' presenti
>> % devo, prima di scrivere plot,
>> % aggiungere il comando hold on
>> % proviamo ad aggiungere il sin(x) al
>> % cos(x)
>> hold on
>> plot(x,y)
>> plot(x,y,'k')
>> % aggiungiamo le legende degli assi
>> xlabel('questa e'' l''asse x')
>> ylabel('asse y')
>> ylabel('asse y','fontSize',18)
>>
>>
>>
>> % COMANDO FONDAMENTALE!!!!!
>> % help nomeFunzione
>> % Ad esempio, help plot
>> % per avere informazioni su cosa fa e come
>> % lo fa la funzione plot
>> help plot
 PLOT   Linear plot.
    PLOT(X,Y) plots vector Y versus vector X. If X or Y is a matrix,
    then the vector is plotted versus the rows or columns of the matrix,
    whichever line up.  If X is a scalar and Y is a vector, disconnected
    line objects are created and plotted as discrete points vertically at
    X.

    PLOT(Y) plots the columns of Y versus their index.
    If Y is complex, PLOT(Y) is equivalent to PLOT(real(Y),imag(Y)).
    In all other uses of PLOT, the imaginary part is ignored.

    Various line types, plot symbols and colors may be obtained with
    PLOT(X,Y,S) where S is a character string made from one element
    from any or all the following 3 columns:

        b     blue          .     point              -     solid
        g     green         o     circle             :     dotted
        r     red           x     x-mark             -.    dashdot
        c     cyan          +     plus               --    dashed
        m     magenta       *     star             (none)  no line
        y     yellow        s     square
        k     black         d     diamond
        w     white         v     triangle (down)
                            ^     triangle (up)
                            <     triangle (left)
                            >     triangle (right)
                            p     pentagram
                            h     hexagram

    For example, PLOT(X,Y,'c+:') plots a cyan dotted line with a plus
    at each data point; PLOT(X,Y,'bd') plots blue diamond at each data
    point but does not draw any line.

    PLOT(X1,Y1,S1,X2,Y2,S2,X3,Y3,S3,...) combines the plots defined by
    the (X,Y,S) triples, where the X's and Y's are vectors or matrices
    and the S's are strings.
```

```
    For example, PLOT(X,Y,'y-',X,Y,'go') plots the data twice, with a
    solid yellow line interpolating green circles at the data points.

    The PLOT command, if no color is specified, makes automatic use of
    the colors specified by the axes ColorOrder property.  The default
    ColorOrder is listed in the table above for color systems where the
    default is blue for one line, and for multiple lines, to cycle
    through the first six colors in the table.  For monochrome systems,
    PLOT cycles over the axes LineStyleOrder property.

    If you do not specify a marker type, PLOT uses no marker.
    If you do not specify a line style, PLOT uses a solid line.

    PLOT(AX,...) plots into the axes with handle AX.

    PLOT returns a column vector of handles to lineseries objects, one
    handle per plotted line.

    The X,Y pairs, or X,Y,S triples, can be followed by
    parameter/value pairs to specify additional properties
    of the lines. For example, PLOT(X,Y,'LineWidth',2,'Color',[.6 0 0])
    will create a plot with a dark red line width of 2 points.

    Example
      x = -pi:pi/10:pi;
      y = tan(sin(x)) - sin(tan(x));
      plot(x,y,'--rs','LineWidth',2,...
                  'MarkerEdgeColor','k',...
                  'MarkerFaceColor','g',...
                  'MarkerSize',10)

    See also plottools, semilogx, semilogy, loglog, plotyy, plot3, grid,
    title, xlabel, ylabel, axis, axes, hold, legend, subplot, scatter.

    Overloaded methods:
      timeseries/plot
      phytree/plot
      clustergram/plot
      dspdata.plot

    Reference page in Help browser
      doc plot

>>
>>
>>
>>
>>
>>
>> % voglio il grafico di f(x)=0.1x^3-x^2
>>
>>
>>
>>
>> % impariamo come eseguire la stessa operazione
>> % su tutti gli elementi di un vettore: operatori
>> % punto
>>
>> clear
>>
>>
>> % voglio elevare al quadrato tutti gli elementi
>> % del vettore 1 2 3 5
```

```
>> v = [1 2 3 5]

v =

    1     2     3     5

>> v2 = v^2
??? Error using ==> mpower
Matrix must be square.

>> v2 = v.^2

v2 =

    1     4     9    25

>> % creo il grafico di 0.1x^3-x^2
>> a =-1;
>> b = 1;
>> x = linspace(a,b,100);
>> y = 0.1*x.^3-x.^2;
>> plot(x,y,'r')
>> grid on
>>
>>
>> % ci sono altri operatori di tipo punto
>> % .*, ./ che eseguono le operazioni
>> % componente a componente di due vettori
>> %(o matrici) delle STESSE DIMENSIONI
>>
>> v1 = [2 4 6 8]

v1 =

    2     4     6     8

>> v2 = [1 2 3 4]

v2 =

    1     2     3     4

>> v1./v2

ans =

    2     2     2     2

>> v1.*v2

ans =

    2     8    18    32

>> v1.^v2

ans =

  Columns 1 through 3

         2          16         216

  Column 4
```

```
      4096

>> help diary
 DIARY Save text of MATLAB session.
    DIARY FILENAME causes a copy of all subsequent command window input
    and most of the resulting command window output to be appended to the
    named file.  If no file is specified, the file 'diary' is used.

    DIARY OFF suspends it.
    DIARY ON turns it back on.
    DIARY, by itself, toggles the diary state.

    Use the functional form of DIARY, such as DIARY('file'),
    when the file name is stored in a string.

    See also save.

    Reference page in Help browser
       doc diary

>>
```