```
>> A = [1 2 3 4; 2 3 4 5; 3 4 5 6; 4 5 6 7];
>> A

A =

     1     2     3     4
     2     3     4     5
     3     4     5     6
     4     5     6     7

>> A'

ans =

     1     2     3     4
     2     3     4     5
     3     4     5     6
     4     5     6     7

>> A(3,4)

ans =

     6

>> A(:,2)
```

```
ans =

     2
     3
     4
     5

>> A(3,:)

ans =

     3     4     5     6

>> size(A)

ans =

     4     4

>> B = ones(4,2)

B =

     1     1
     1     1
```

```
     1     1
     1     1

>> A*B

ans =

>>
>> A*B

ans =

    10    10
    14    14
    18    18
    22    22

>> B*A
??? Error using ==> mtimes
Inner matrix dimensions must agree.

>> A(:,2)*B(:,1)
??? Error using ==> mtimes
Inner matrix dimensions must agree.

>> A(:,2).*B(:,1)
```

```
ans =

     2
     3
     4
     5

>> A

A =

     1     2     3     4
     2     3     4     5
     3     4     5     6
     4     5     6     7

>> length(A)

ans =

     4

>> B

B =
```

```
     1     1
     1     1
     1     1
     1     1

>> size(B)

ans =

     4     2

>> size(B,1)

ans =

     4

>> size(B,2)

ans =

     2

>> help size
 SIZE   Size of array.
```

D = SIZE(X), for M-by-N matrix X, returns↙ the two-element row vector
D = [M,N] containing the number of rows and↙ columns in the matrix.
For N-D arrays, SIZE(X) returns a 1-by-N↙ vector of dimension lengths.
Trailing singleton dimensions are ignored.

[M,N] = SIZE(X) for matrix X, returns the↙ number of rows and columns in
X as separate output variables.

[M1,M2,M3,...,MN] = SIZE(X) for N>1 returns↙ the sizes of the first N
dimensions of the array X. If the number of↙ output arguments N does
not equal NDIMS(X), then for:

N > NDIMS(X), SIZE returns ones in the↙ "extra" variables, i.e., outputs
NDIMS(X)+1 through N.
N < NDIMS(X), MN contains the product of the↙ sizes of dimensions N
through NDIMS(X).

M = SIZE(X,DIM) returns the length of the↙

```
dimension specified
    by the scalar DIM.  For example, SIZE(X,1)↙ returns the number
    of rows. If DIM > NDIMS(X), M will be 1.

    When SIZE is applied to a Java array, the↙ number of rows
    returned is the length of the Java array and↙ the number of columns
    is always 1.  When SIZE is applied to a Java↙ array of arrays, the
    result describes only the top level array in↙ the array of arrays.

    Example:
    If
       X = rand(2,3,4);
    then
       d = size(X)              returns  d = [2↙ 3 4]
       [m1,m2,m3,m4] = size(X)  returns  m1 = 2,↙ m2 = 3, m3 = 4, m4 = 1
       [m,n] = size(X)          returns  m = 2,↙ n = 12
       m2 = size(X,2)           returns  m2 = 3
```

See also length, ndims, numel.

Overloaded methods:
TriRep/size
timer/size
serial/size
tscollection/size
gf/size
InputOutputModel/size
daqdevice/size
daqchild/size
distributed/size
codistributed/size
Composite/size
fints/size
idmodel/size
idfrd/size
iddata/size
idnlmodel/size
idnlgrey/size
idnlfun/size
videosource/size
videoinput/size
visa/size
udp/size
tcpip/size

```
       icgroup/size
       icdevice/size
       gpib/size
       mpc/size
       uss/size
       umat/size
       ufrd/size
       ndlft/size
       icsignal/size
       atom/size
       dataset/size
       categorical/size
       sym/size

    Reference page in Help browser
       doc size

Hello World
>> helloWorldScript
Hello World
>> x = pi/100:pi/100:10*pi;
>> y = sin(x)./x;
>> plot(x,y)
>> Grid
Warning: Could not find an exact
(case-sensitive) match for 'Grid'.
```

```
C:\Program
Files\MATLAB\R2010b\toolbox\matlab\graph2d\grid.↙
m
is a case-insensitive match and will be
used instead.
You can improve the performance of your
code by using exact
name matches and we therefore recommend
that you update your
usage accordingly. Alternatively, you can
disable this warning using
warning('off','MATLAB:dispatcher:↙
InexactCaseMatch').
This warning will become an error in
future releases.
>> exampleScript1
>> clear all
??? Undefined function or variable 'x'.

Error in ==> exampleScript1 at 5
y2 = sin(x)./x;

>> help soft

soft not found.
```

```
Use the Help browser search field to search the↙
documentation, or
type "help help" for help command options, such↙
as help for methods.

>> help sor

sor not found.

Use the Help browser search field to search the↙
documentation, or
type "help help" for help command options, such↙
as help for methods.

>> help sort
 SORT   Sort in ascending or descending order.
    For vectors, SORT(X) sorts the elements of X↙
in ascending order.
    For matrices, SORT(X) sorts each column of X↙
in ascending order.
    For N-D arrays, SORT(X) sorts the along the↙
first non-singleton
    dimension of X. When X is a cell array of↙
strings, SORT(X) sorts
    the strings in ASCII dictionary order.
```

```
    Y = SORT(X,DIM,MODE)
    has two optional parameters.
    DIM selects a dimension along which to sort.
    MODE selects the direction of the sort
       'ascend' results in ascending order
       'descend' results in descending order
    The result is in Y which has the same shape↙
and type as X.

    [Y,I] = SORT(X,DIM,MODE) also returns an↙
index matrix I.
    If X is a vector, then Y = X(I).
    If X is an m-by-n matrix and DIM=1, then
        for j = 1:n, Y(:,j) = X(I(:,j),j); end

    When X is complex, the elements are sorted↙
by ABS(X).  Complex
    matches are further sorted by ANGLE(X).

    When more than one element has the same↙
value, the order of the
    elements are preserved in the sorted result↙
and the indexes of
    equal elements will be ascending in any↙
index matrix.
```

```
    Example: If X = [3 7 5
                     0 4 2]

    then sort(X,1) is [0 4 2  and sort(X,2) is↙
[3 5 7
                       3 7 5]↙
0 2 4];

    See also issorted, sortrows, min, max, mean,↙
median, unique.

    Overloaded methods:
       ordinal/sort
       nominal/sort
       sym/sort

    Reference page in Help browser
       doc sort

>> inArray = [4 5 0 -1 10];
>> [outArray, outInd] = descsort(inArray);
>> outArray

outArray =

    10     5     4     0    -1
```

```
>> descsort(inArray)

ans =

    10     5     4     0    -1


k =

     1


k =

     2


k =

     3


k =

     4
```

```
k =

     5

>>
>> k

k =

     5

>> A

A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
     5    10    15


A =
```

```
     1     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
```

A =

```
     1     2     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0
```

A =

```
     1     2     3
     0     0     0
     0     0     0
     0     0     0
     0     0     0
```

A =

```
     1     2     3
     2     0     0
     0     0     0
     0     0     0
     0     0     0
```

A =

```
     1     2     3
     2     4     0
     0     0     0
     0     0     0
     0     0     0
```

A =

```
     1     2     3
     2     4     6
     0     0     0
     0     0     0
     0     0     0
```

```
A =

     1     2     3
     2     4     6
     3     0     0
     0     0     0
     0     0     0

A =

     1     2     3
     2     4     6
     3     6     0
     0     0     0
     0     0     0

A =

     1     2     3
     2     4     6
     3     6     9
     0     0     0
     0     0     0
```

```
A =

     1     2     3
     2     4     6
     3     6     9
     4     0     0
     0     0     0

A =

     1     2     3
     2     4     6
     3     6     9
     4     8     0
     0     0     0

A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
     0     0     0
```

```
A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
     5     0     0

A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
     5    10     0

A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
```

```
     5    10    15

>>
>> clear all

A =

     1     0     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0


A =

     1     2     0
     0     0     0
     0     0     0
     0     0     0
     0     0     0


A =

     1     2     3
```

```
     0     0     0
     0     0     0
     0     0     0
     0     0     0
```

A =

```
     1     2     3
     2     0     0
     0     0     0
     0     0     0
     0     0     0
```

A =

```
     1     2     3
     2     4     0
     0     0     0
     0     0     0
     0     0     0
```

A =

```
     1     2     3
     2     4     6
     0     0     0
     0     0     0
     0     0     0
```

```
A =

     1     2     3
     2     4     6
     3     0     0
     0     0     0
     0     0     0
```

```
A =

     1     2     3
     2     4     6
     3     6     0
     0     0     0
     0     0     0
```

```
A =
```

```
     1     2     3
     2     4     6
     3     6     9
     0     0     0
     0     0     0
```

A =

```
     1     2     3
     2     4     6
     3     6     9
     4     0     0
     0     0     0
```

A =

```
     1     2     3
     2     4     6
     3     6     9
     4     8     0
     0     0     0
```

```
A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
     0     0     0

A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
     5     0     0

A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
     5    10     0
```

```
A =

     1     2     3
     2     4     6
     3     6     9
     4     8    12
     5    10    15


ans =

     2

     2


ans =

     6

     4


ans =
```

```
    10

     6


ans =

    14

     8


ans =

    18

    10

i is odd

ans =

     2

     2
```

```
i is odd

ans =

     6

     4

i is odd

ans =

    10

     6

i is odd

ans =

    14

     8

i is odd
```

```
ans =

    18

    10

i is odd

ans =

     2

i is even

i =

     2

i is odd

ans =

     6

i is even
```

```
i =

     4

i is odd

ans =

    10

i is even

i =

     6

i is odd

ans =

    14

i is even

i =
```

```
     8

i is odd

ans =

    18

i is even

i =

    10

i is 1
i is 2
i is 3
i is 4
i is none of the cases
i is 6
i is none of the cases
i is none of the cases
i is none of the cases
i is none of the cases
>>
```