

New to MATLAB/Octave?

Matlab isn't hard; it's just a bunch of variable assignments and functions on variables, vectors, or matrices.

Rules of the game:

- 1) Putting a semicolon after a statement does not print anything, leaving it off prints the value.**
- 2) Assignments are made variable=value**

```
>> a=3
```

```
a =
```

```
3
```

```
>> a
```

```
a =
```

```
3
```

```
>> a=4;
```

```
>> a
```

```
a =
```

```
4
```

- 3) Vectors can be column or row; you use brackets to type in vectors and matrices**

```
>> row=[1 2]
```

```
row =
```

```
1    2
```

```
>> col=[1;2]
```

```
col =
```

```
1
```

```
2
```

```
>> matrix=[1 2; 3 4]
```

```
matrix =
```

```
1    2
```

```
3    4
```

#### 4) You can get a sequence using start:stop, or start:step:stop

```
>> c=1:5
```

```
c =
```

```
    1    2    3    4    5
```

```
>> d=0:0.2:1
```

```
d =
```

```
    0    0.2000    0.4000    0.6000    0.8000    1.0000
```

#### 5) for loops are done this way

```
>> for i=1:5
```

```
    i*3
```

```
end
```

```
ans =
```

```
    3
```

```
ans =
```

```
    6
```

```
ans =
```

```
    9
```

```
ans =
```

```
   12
```

```
ans =
```

```
   15
```

#### 6) Many functions take a vector

```
>> sum(c)
```

```
ans =
```

```
   15
```

```
>> prod(c)
```

```
ans =
```

```
  120
```

### 7) Matrix multiplication versus pointwise multiplication of matrices

```
>> matrix*matrix
```

```
ans =
```

```
     7     10
    15     22
```

```
>> matrix.*matrix
```

```
ans =
```

```
     1     4
     9    16
```

### 8) apostrophe marks transposition

```
>> col'
```

```
ans =
```

```
     1     2
```

```
>> row'
```

```
ans =
```

```
     1
     2
```

```
>> matrix'
```

```
ans =
```

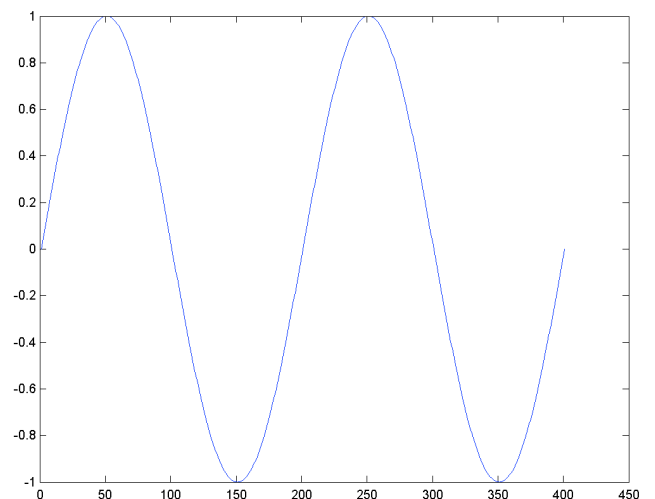
```
     1     3
     2     4
```

### 9) plot can be used to graph functions, print will create graphics files from plots

```
>> x=sin(0:pi/100:4*pi);
```

```
>> plot(x)
```

```
>> print -dtiff sinplot.tif
```



## 10) lookfor will help find functions you want

## 11) help will give manual definitions of functions

```
>> lookfor fft
DETREND Remove a linear trend from a vector, usually for FFT
processing.
FFT Discrete Fourier transform.
FFT2 Two-dimensional discrete Fourier Transform.
FFTN N-dimensional discrete Fourier Transform.
FFTSHIFT Shift zero-frequency component to center of spectrum.
FFTW Interface to FFTW library run-time algorithm tuning control.
IFFT Inverse discrete Fourier transform.
IFFT2 Two-dimensional inverse discrete Fourier transform.
IFFTN N-dimensional inverse discrete Fourier transform.
IFFTSHIFT Inverse FFT shift.
INTERPFT 1-D interpolation using FFT method.
fftdemo.m: %% FFT for Spectral Analysis
sunspots.m: %% Using FFT in MATLAB
FFT Overloaded function for UINT16 input.
FFTN Overloaded function for UINT16 input.
IFFT Overloaded function for UINT16 input.
IFFTN Overloaded function for UINT16 input.
FFT Overloaded function for UINT8 input.
FFTN Overloaded function for UINT8 input.
IFFT Overloaded function for UINT8 input.
IFFTN Overloaded function for UINT8 input.
>> help fft
FFT Discrete Fourier transform.
    FFT(X) is the discrete Fourier transform (DFT) of vector X. For
    matrices, the FFT operation is applied to each column. For N-D
    arrays, the FFT operation operates on the first non-singleton
    dimension.

    FFT(X,N) is the N-point FFT, padded with zeros if X has less
    than N points and truncated if it has more.

    FFT(X,[],DIM) or FFT(X,N,DIM) applies the FFT operation across the
    dimension DIM.

    For length N input vector x, the DFT is a length N vector X,
    with elements
        
$$X(k) = \sum_{n=1}^N x(n) \exp(-j*2*\pi*(k-1)*(n-1)/N), \quad 1 \leq k \leq N.$$

    The inverse DFT (computed by IFFT) is given by
        
$$x(n) = (1/N) \sum_{k=1}^N X(k) \exp(j*2*\pi*(k-1)*(n-1)/N), \quad 1 \leq n \leq N.$$


    See also fft2, fftn, fftshift, fftw, ifft, ifft2, ifftn.

    Overloaded functions or methods (ones with the same name in other
    directories)
        help uint8/fft.m
        help uint16/fft.m

    Reference page in Help browser
        doc fft
```

**12) Comments start with %**

**13) You can define a function by putting it into a file with the same name and .m following. For example, if you wanted to add two numbers/vectors/matrices together and you didn't want to use +, create the file "add.m" with the following:**

```
function output=add(a,b)
```

```
output=a+b;
```

**The variable before the function name is the return variable. You can call the function as you would any other function:**

```
>> add(3,4)
```

```
ans =
```

```
7
```