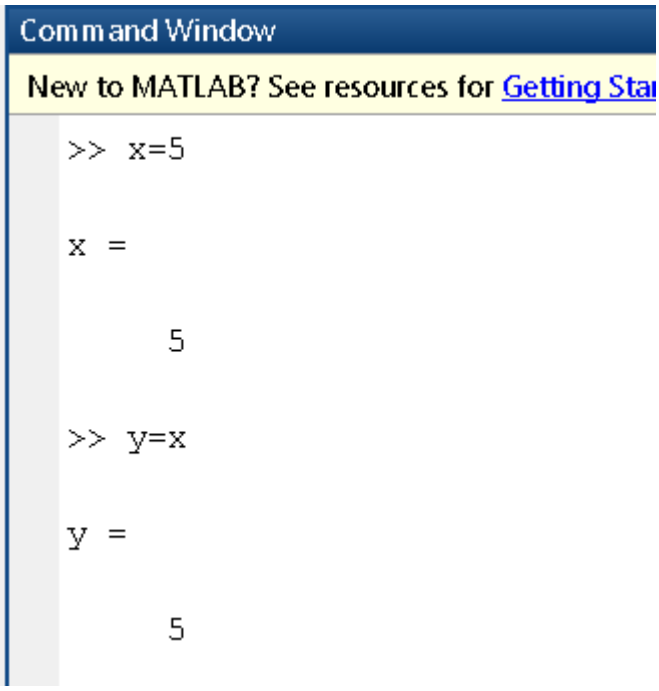
Execute a command

1. Try writing a command in command window
2. Assign a variable : Assign a constant to variable
3. Assignment operator : The operator(=) can be used to assign value




```
x=5
```

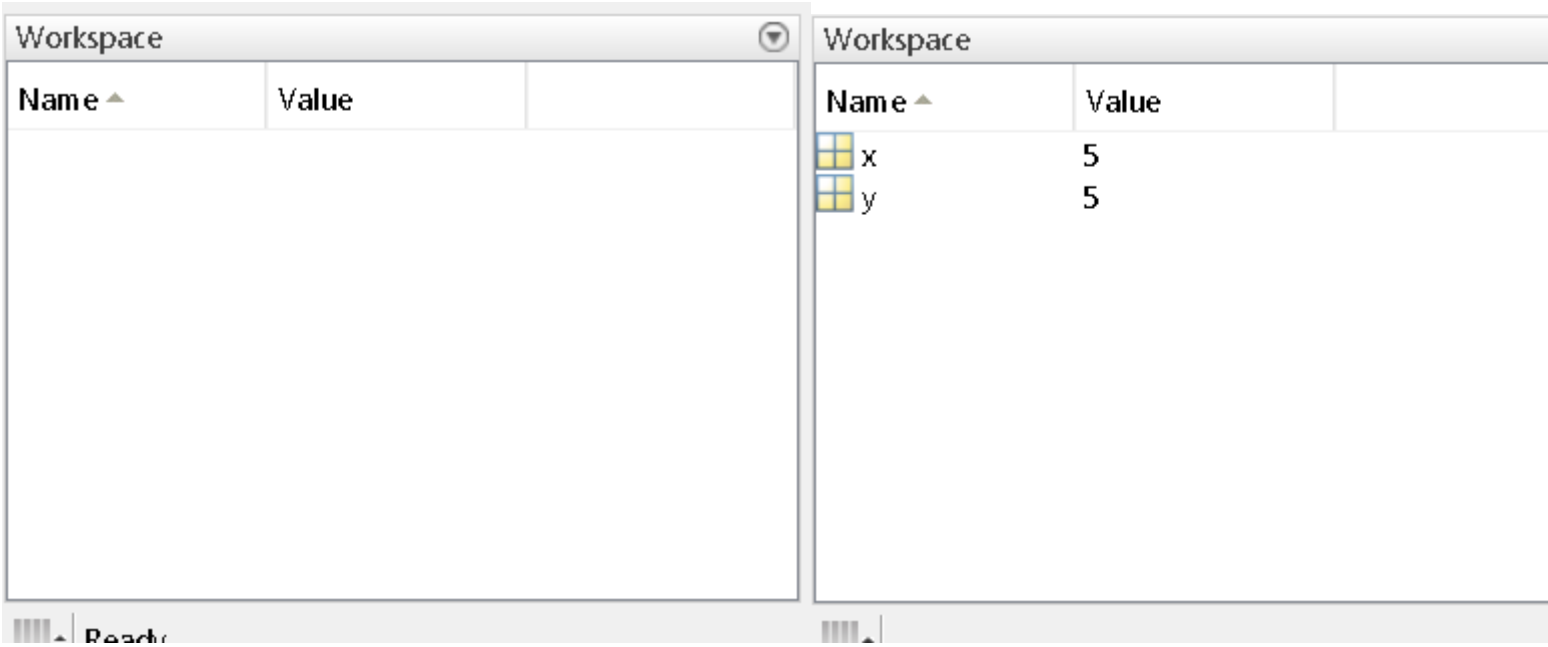
```
x =
     5
```

```
y=x
```

```
y =
     5
```

Workspace in matlab

It contains the current variables that are stored

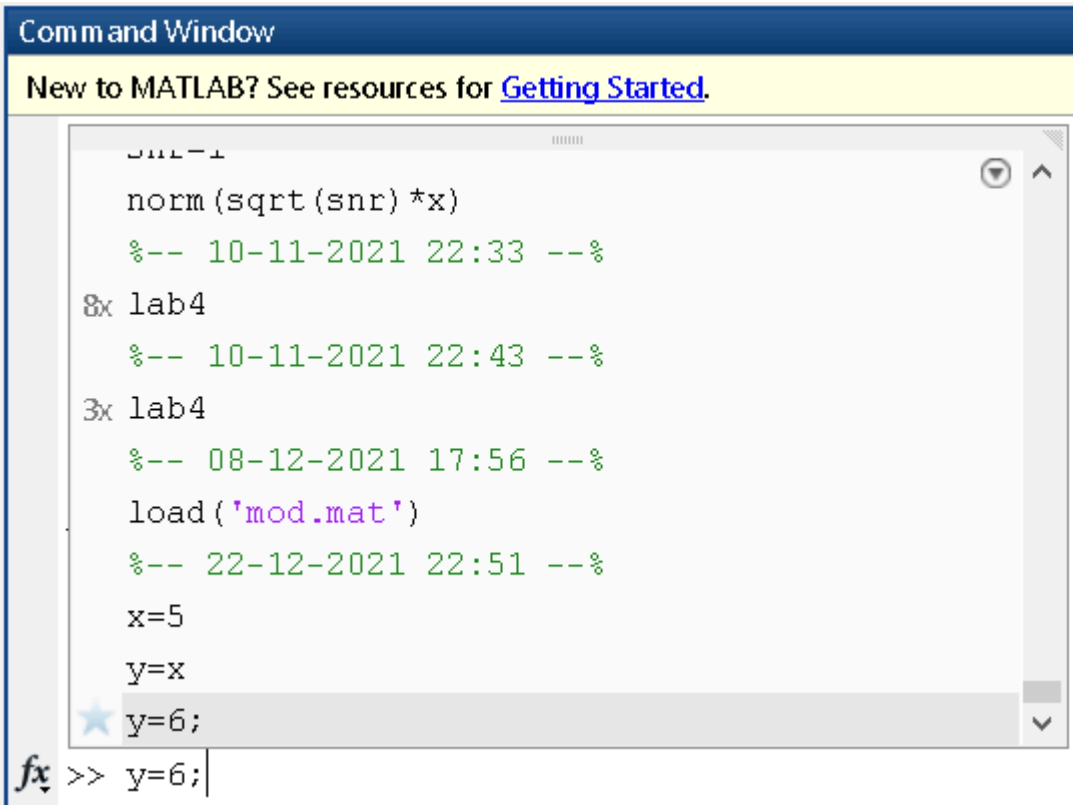


Semicolon

Using a semicolon after any operation does not print the output

Recall previous commands

By using arrow keys we can also refer to previous commands entered in command window



Enter just a variable

writing a varibale and simply pressing enter prints its value

```
>> y

y =

     6

>>
```

Naming a Variable

You can name your Matlab Variables anything start wih a letter and contain only letters , numbers and underscores (_)

```
y=6;
y
```

```
y =
     6
```

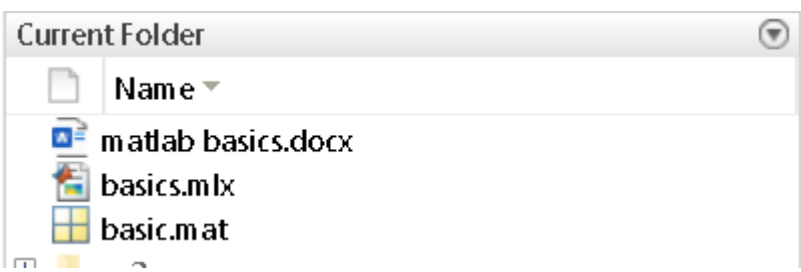
Save the workspace to a mat file

We can save the current workspace to be loaded again in future . This saves the data to .mat file in the current directory .

Simply,

save filename

load filename



Side tip : To remove previous clear the command window use "clc"

```
save basic.mat
load basic.mat
```

Using built in functions

Matlab has a lot of built in functions that can be used to perform operations

abs , sin , cos , sqrt

```
Command Window
New to MATLAB? See resources for Getting Started.
>> abs(-6)

ans =

     6

>> sqrt(-6)

ans =

   0.0000 + 2.4495i

>>
```

While performing an operation you might have noticed that the output shows only 4 decimal digits to increase/decrease the accuracy of operations we use

format long , format short

```
Command Window
New to MATLAB? See resources for Getting Started.
>> sqrt(-6)

ans =

   0.0000 + 2.4495i

>> format long
>> sqrt(-6)

ans =

  0.000000000000000 + 2.449489742783178i

>>
```

```
abs(-6)
```

```
ans =
     6
```

```
sqrt(-6)
```

```
ans =
  0.000000000000000 + 2.449489742783178i
```

```
format long
sqrt(-6)
```

```
ans =
  0.000000000000000 + 2.449489742783178i
```

Array

Collection of data is called an array . It has two constituents rows and columns

array=[(row1,col1) (row1,col2) (row1,col3);

(row2,col1) (row2,col2) (row2,col3);

(row2,col1) (row3,col2) (row3,col3)]

A column array of size [rows cols]=[1 2]

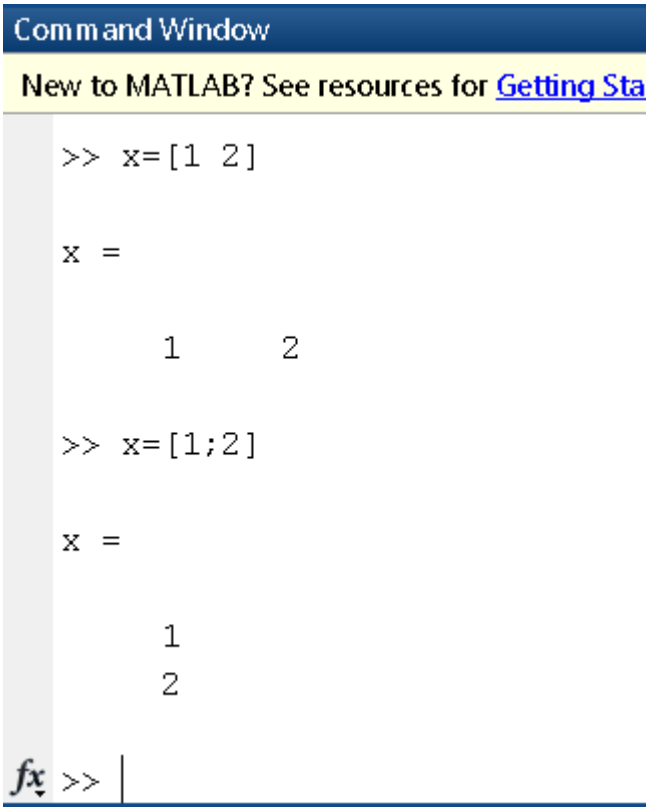
x=[1 2]

To make rows semicolon is used.[rows cols]=[2 1]

x=[1;2]

Example

x = [3 4 5;6 7 8]




```
x=[1 2]
```

```
x = 1×2
     1     2
```

```
x=[1;2]
```

```
x = 2×1
     1
     2
```

```
x=[3 4 5;6 7 8]
```

```
x = 2×3
     3     4     5
     6     7     8
```

Vector

A **vector** is a list of numbers (can be in a row or column)

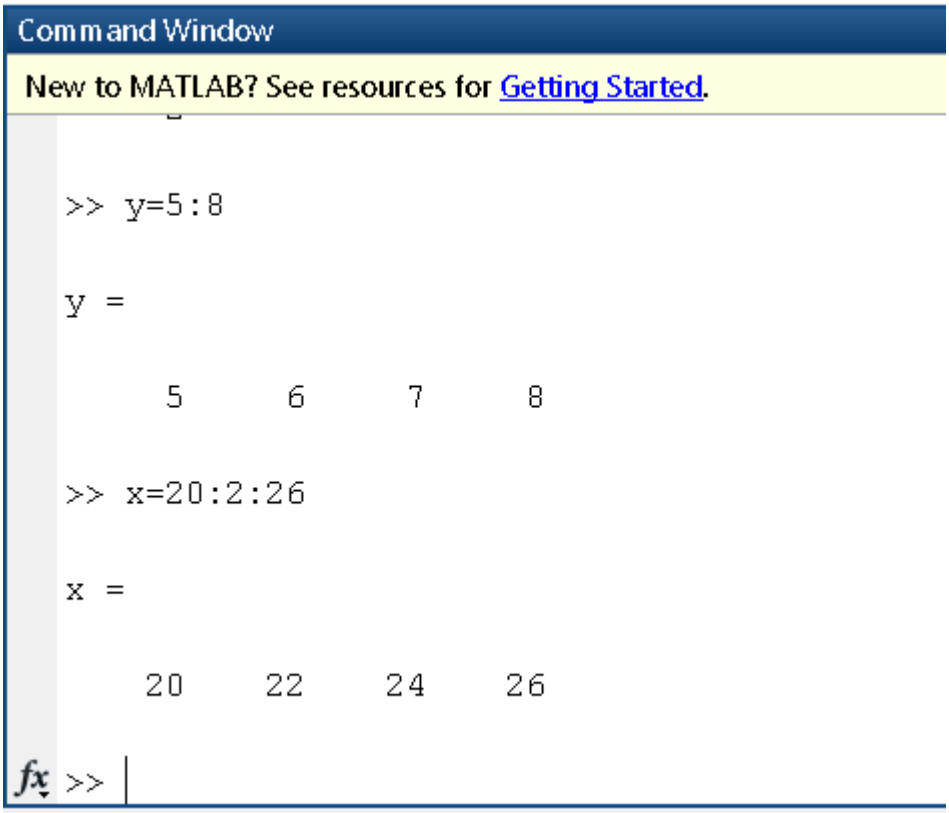
Evently Spaced vectors

To make a list of vectors we use a **first:last** , by default the difference is 1.

y=5:8

We can also use the following format **first:difference:last**

x=20:2:26



If we know the number of elements we require we can use the following command **linspace(*first*,*last*,*number_of_elements*)**

linspace(1,10,100)

```
Command Window
New to MATLAB? See resources for Getting Started.
>> linspace(1,10,100)

ans =

  Columns 1 through 7

   1.000000000000000   1.090909090909091   1.181818181818182   1.272727272

  Columns 8 through 14

   1.636363636363636   1.727272727272727   1.818181818181818   1.909090909

  Columns 15 through 21
```

Exercise : If we want to make a 100 elements between 1 and 2*pi what command should we use

Note : The output of these commands results in a column vector , to convert it into a row vector we can do Transpose by

x=x'

Side tip: If you dont assign a variable to the output it is stored in ans

```
y=5:8
x=20:2:26
linspace(1,10,100)
x=x'
```

Array Creation Functions

Some arrays which are widely used during calculations have pre defined functions

Make a 2x2 random number array between 0 and 1

x = rand(2)

Make a 2x3 random number array between 0 and 1

x = rand(2,3)

```
Command Window
New to MATLAB? See resources for Getting Started.

>> x=rand(2)

x =

   0.814723686393179   0.126986816293506
   0.905791937075619   0.913375856139019

>> x=rand(2,3)

x =

   0.632359246225410   0.278498218867048   0.957506835434298
   0.097540404999410   0.546881519204984   0.964888535199277

fx >>
```

Make an array of ones

ones(2,3)

Similarily try zeros,eye

Tip : size , can be used to get the size of an array

```
x=rand(2)
x=rand(2,3)
size(x)
```

Indexing an array

To refer to an element in array indexing is used

Index an element : x(1)

For a 2D it reads column wise i.e for a 2x3 array

1 3 5

2 4 6

Index series of element : x(2:4) , elemts from index 2 to 4 are given as output

Index 2D array : x(1,:) , all elements in row=1 and all columns are given as output

```
Command Window
New to MATLAB? See resources for Getting Started.
>> x(1)

ans =

   0.632359246225410

>> x(2:4)

ans =

   0.097540404999410   0.278498218867048   0.546881519204984

>> x(1,:)

ans =

   0.632359246225410   0.278498218867048   0.957506835434298
```

So, x(idx) , x(row,col)

Indexing the last element : z=x(end,2) or x(end-1,2)

Exercise : Try extracting the non consecutive numbers in array

```
x(1)
x(2:4)
x(1,:)
z=x(end,2)
y(1)=x(end,2)
```

Arithmetic Operations

x=[1 2 3 4]

y=x+2

z=x+y

```
x =

     1     2     3     4


y =

     3     4     5     6


z =

     4     6     8    10

fx >>
```

basic statistic max,sqrt,matrix multiplication are all applicable on arrays

For Matrix multiplication , z=[3 4] *[10;20]

For element wise multiplication , z=[3 4] .*[10 20]

```
>> z=[3 4] *[10;20]

z =

   110

>> z=[3 4].*[10 20]

z =

    30    80

fx >>
```

The dot before a mathematical opration implies element wise operation

Example: x=x.^2

x=x^2

```
x=[1 2 3 4]
```

```
y=x+2
z=x+y
z=[3 4] *[10;20]
z=[3 4].*[10 20]
```

Function calls

x=[1 2 3 4 5]

[xrow,xcol] = size(x)

The output of size is assigned to xrow and xcol variables

[xMax,idx] = max(x)

Use tilde to ignore specific values

[~,ivMax] = max(x)

Live Editor - C:\Users\prana\Downloads\basics.mlx

Command Window

New to MATLAB? See resources for Getting Started.

```
x =

     1     2     3     4     5


xrow =

     1


xcol =

     5


xMax =

     5


idx =

     5


ivMax =

     5
```

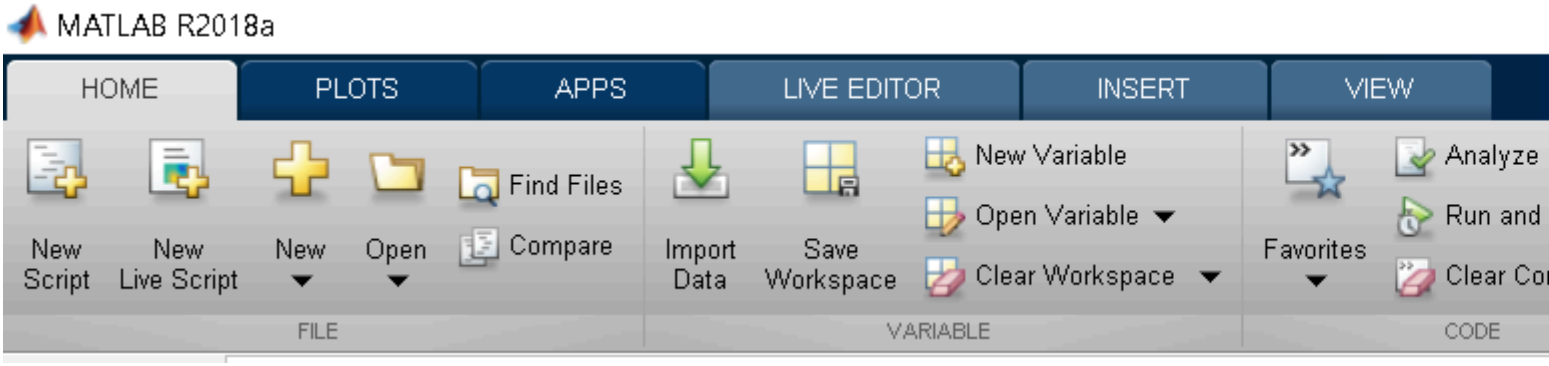
Exercise : Check the documentation for creating a custom function https://in.mathworks.com/help/matlab/ref/function.html

```
x=[1 2 3 4 5]
[xrow,xcol] = size(x)
[xMax,idx] = max(x)
```

```
[~,ivMax] = max(x)
```

Obtaing Help from matlab documentation

We cant remember everything so matlab has offline documentation of everything that is required . Click on Help or search in documentation.



Or Help and search for function in command window

doc randi

```
doc randi
```

Plotting

It is important to plot the data we have for better visualization

x are the points on x axis

y are the points on y axis

plot(x,y)

plot(x,y,"r--o")

The command above plots a red (r) dashed (--) line with a circle (o) as a marker. You can learn more about the symbols available in the documentation for Line Specification.

hold on command to hold the previous plot while you add another line

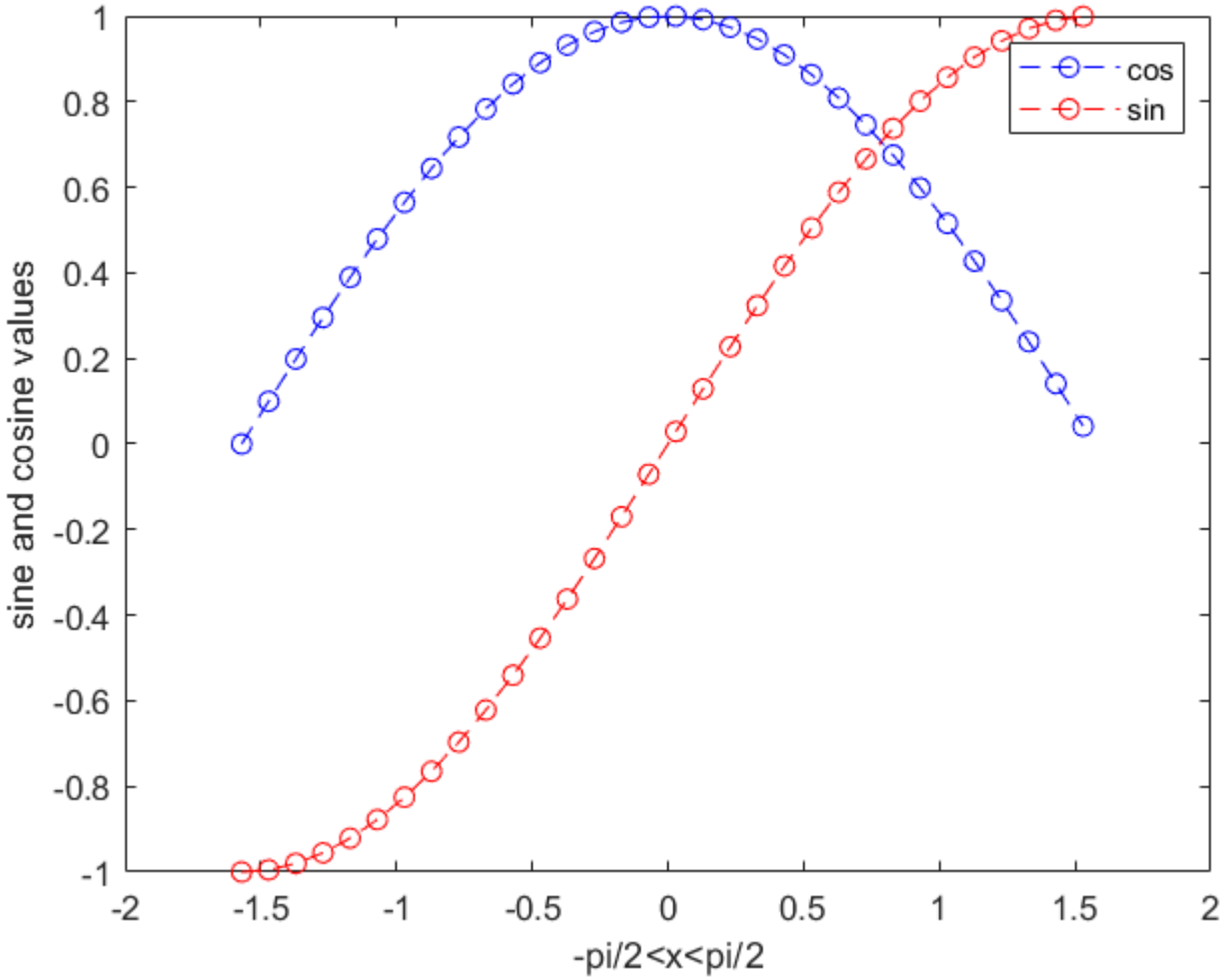
Check out : https://in.mathworks.com/products/matlab/plot-gallery.html

Some more commands .

xlabel('-pi/2 < x < pi/2')

ylabel('Sine and Cosine Values')

legend(["cos" "sin"])




```
x=-pi/2:0.1:pi/2
```

```
x = 1×32
  -1.570796326794897  -1.470796326794896  -1.370796326794897  -1.270796326794897 ···
```

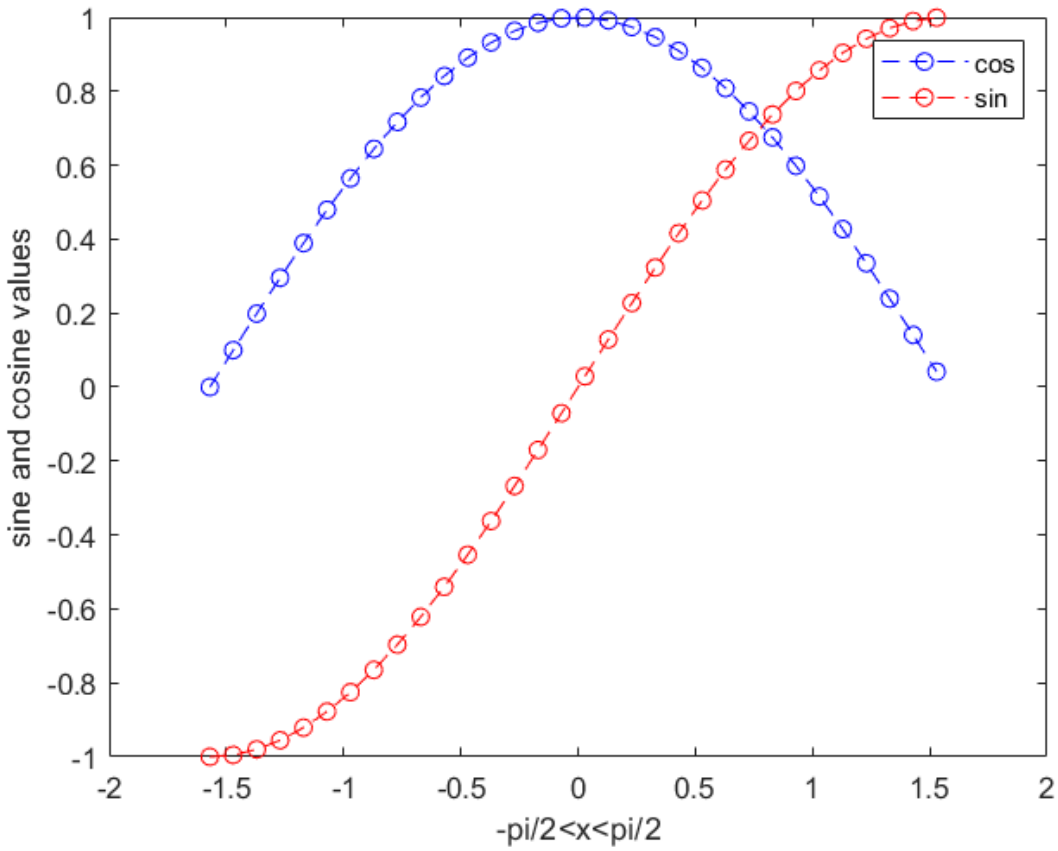
```
y=cos(x)
```

```
y = 1×32
   0.000000000000000   0.099833416646828   0.198669330795061   0.295520206661340 ···
```

```
z=sin(x)
```

```
z = 1×32
  -1.000000000000000  -0.995004165278026  -0.980066577841242  -0.955336489125606 ···
```

```
plot(x,y,"b--o")
hold on
plot(x,z,"r--o")
hold off
xlabel('-pi/2<x<pi/2')
ylabel('sine and cosine values')
legend(["cos" "sin"])
```



Relational operators, such as >, <, ==, and ~= perform comparisons between two values. The outcome of a comparison for equality or inequality is either 1 (true) or 0 (false).

Command Window

New to MATLAB? See resources for Getting Started.

```
>> x=-pi/2:0.1:pi/2;
x<0

ans =

  1×32 logical array

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

>>
```

You can use a logical array as an array index, in which case MATLAB extracts the array elements where the index is true.

x(x<0)

```
Command Window
New to MATLAB? See resources for Getting Started.

>> x(x<0)

ans =

  Columns 1 through 7

  -1.570796326794897  -1.470796326794896  -1.370796326794897  -1.270796326

  Columns 8 through 14

  -0.870796326794896  -0.770796326794897  -0.670796326794897  -0.570796326

  Columns 15 through 16

  -0.170796326794896  -0.070796326794897

fx >>
```

To reassign value

x(x==0) = 1

```
x=-pi/2:0.1:pi/2
x<0
x(x<0)
```

IF/ELSE

Conditional statements are required to skip a section of code if some contion is true or false

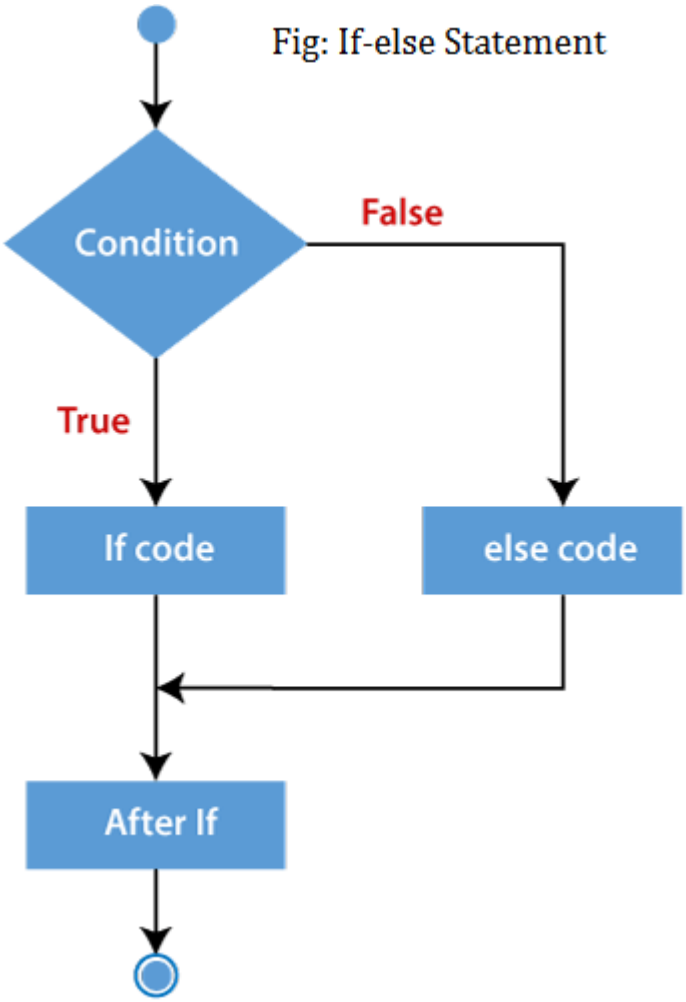


Fig: If-else Statement

```
xxx
if condition
    xxx
elseif condition
   xxx
else
xxxxxx
end
xxx
```

Tip: No need to worry about indentation but it is good to have

Command Window

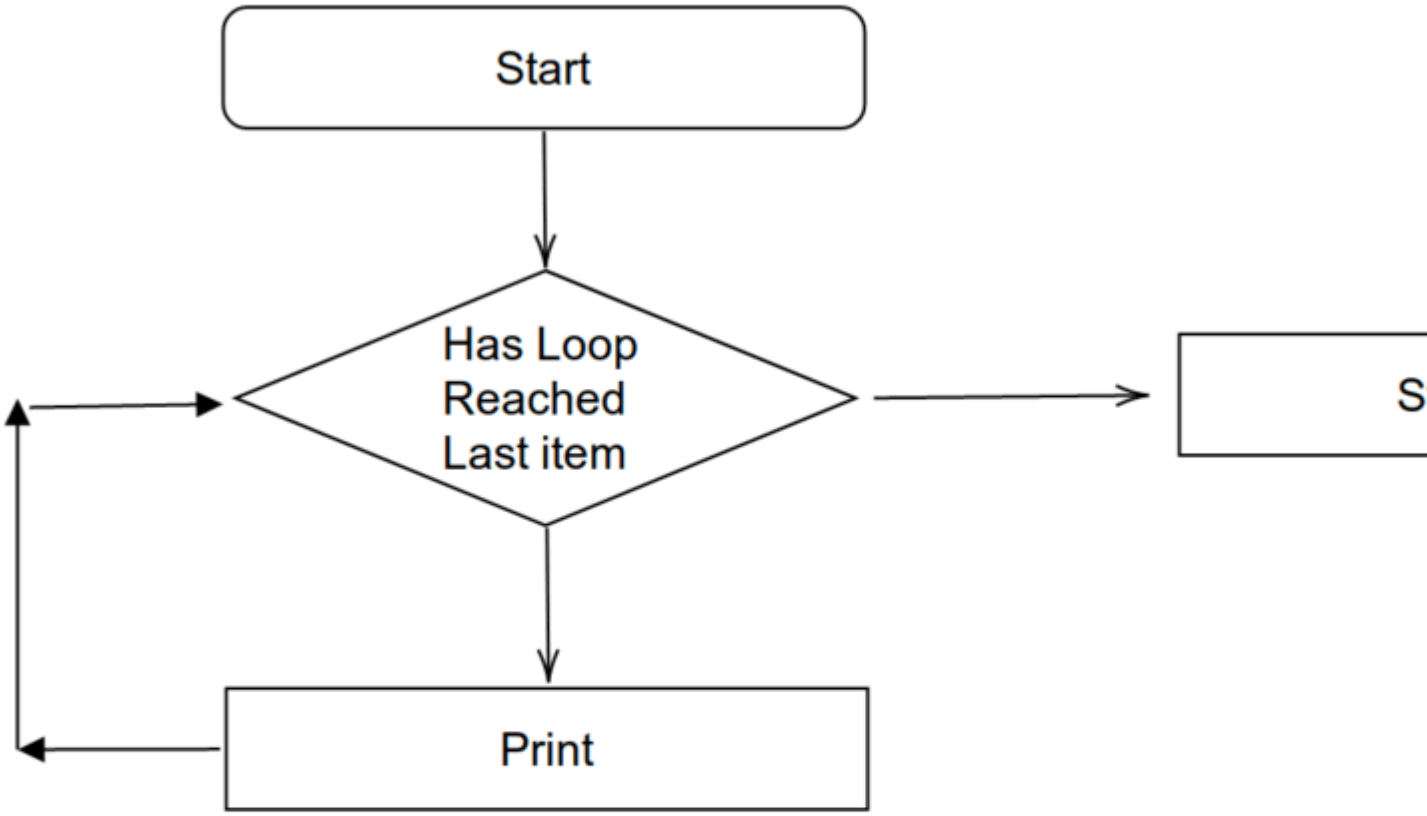
New to MATLAB? See resources for Getting Started.

```
>> if(1<5)
disp("1<5")
elseif(1>5)
disp("1>5")
else
disp("1==5")
end
1<5
>>
```

```
if(1<5)
disp("1<5")
elseif(1>5)
disp("1>5")
else
disp("1==5")
end
```

LOOP

looops are used if we want ot execute same command again and again




```
for x=1:5
xxxxx
end
```

Command Window

New to MATLAB? See resources for Getting Started.

```
>> z=[]
for x=1:5
    z=[z x];
end
z

z =

     []


z =

     1     2     3     4     5

>>
```

```
z=[]
for x=1:5
    z=[z x];
end
z
```