

MATLAB Tutorial

http://www.mathworks.co.kr/help/pdf_doc/matlab/getstart.pdf

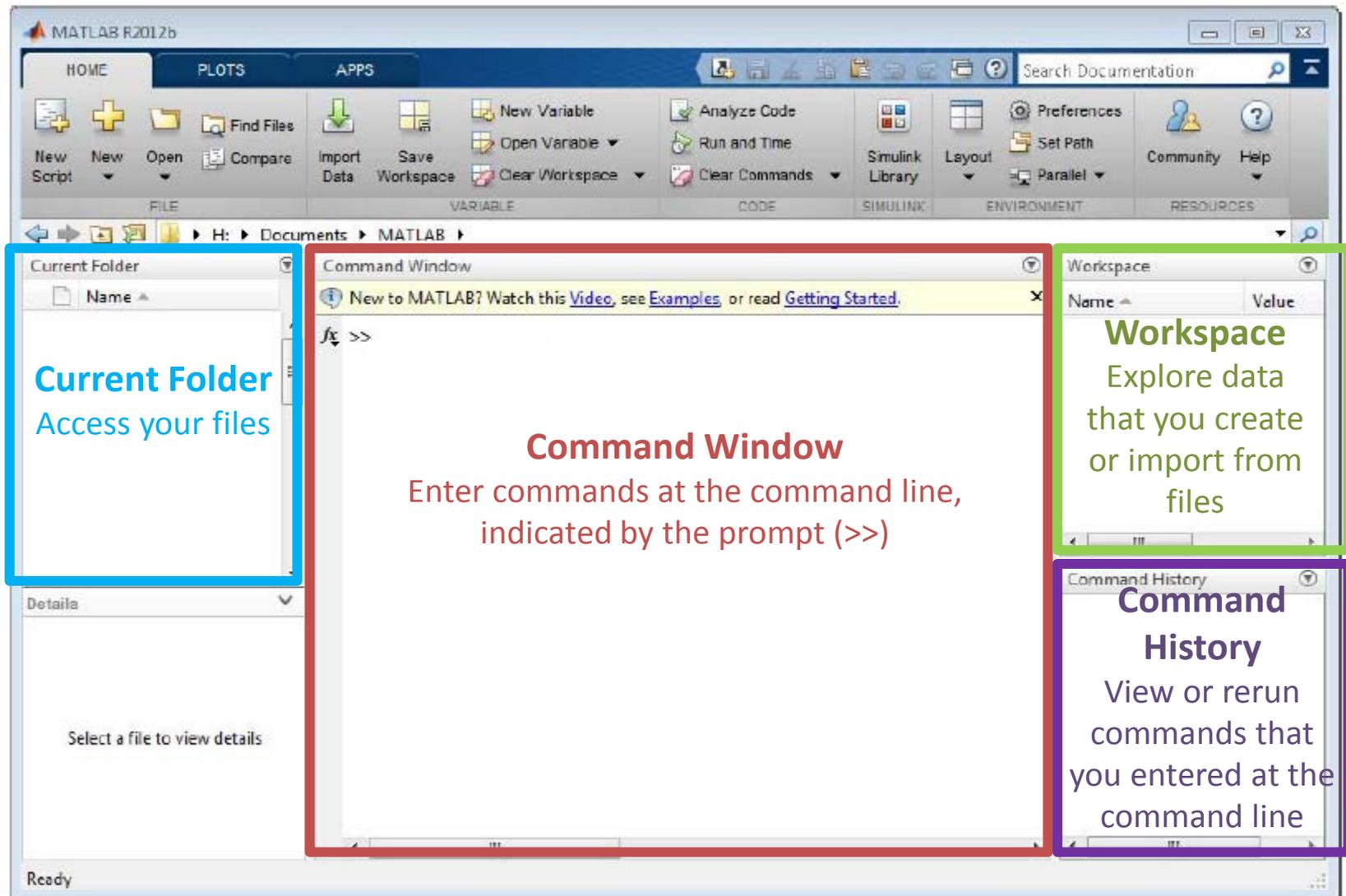
Artificial Intelligence : Cognitive Agents

2014.04.26 Practice2

MATLAB Product Description

- High-level language for numerical computation, visualization, and application development
- Interactive environment for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics, Fourier analysis, filtering, optimization, numerical integration, and solving ordinary differential equations
- Built-in graphics for visualizing data and tools for creating custom plots
- Development tools for improving code quality and maintainability and maximizing performance
- Tools for building applications with custom graphical interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages such as C, Java, .NET, and Microsoft® Excel®

Desktop Basics

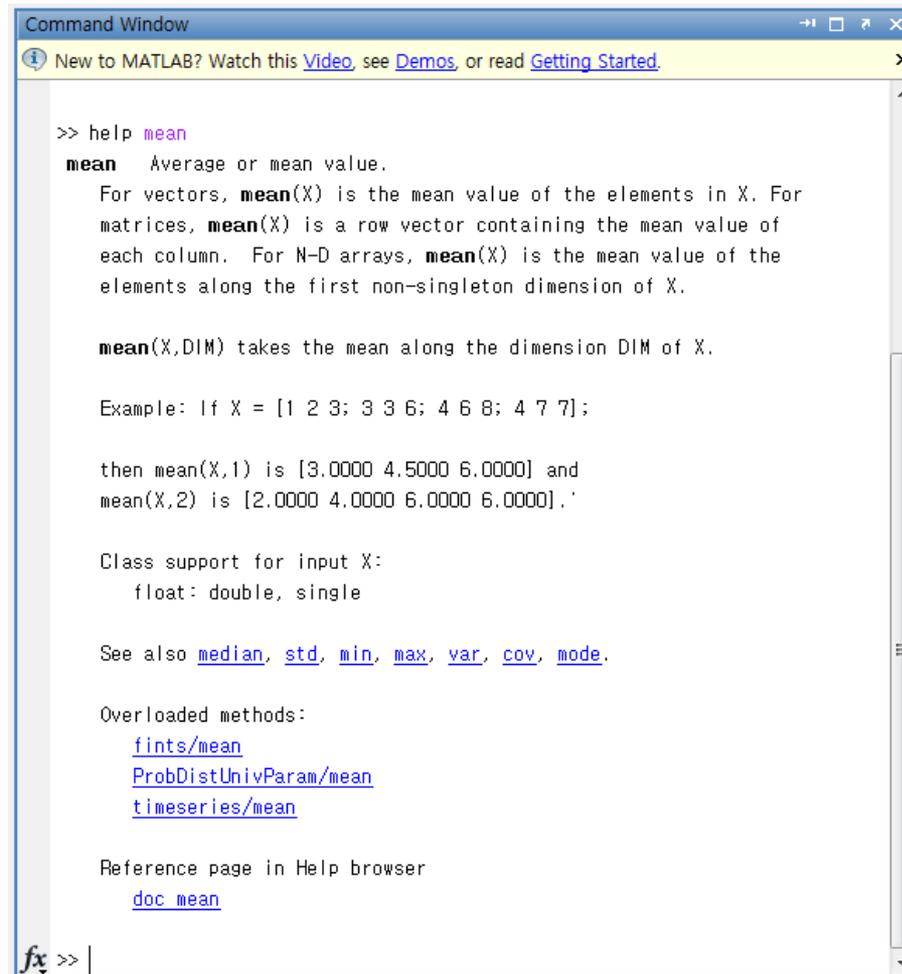


The image shows the MATLAB R2012b desktop environment. The main window has a ribbon with tabs for HOME, PLOTS, and APPS. Below the ribbon are toolbars for FILE, VARIABLE, CODE, SIMULINK, ENVIRONMENT, and RESOURCES. The current folder is H:\Documents\MATLAB. Three windows are highlighted with colored borders and text boxes:

- Current Folder** (blue border): A file explorer window showing the current directory. Text: **Current Folder**
Access your files
- Command Window** (red border): A window for entering commands. It shows a prompt `>>` and a message: "New to MATLAB? Watch this [Video](#), see [Examples](#), or read [Getting Started](#)." Text: **Command Window**
Enter commands at the command line, indicated by the prompt (`>>`)
- Workspace** (green border): A window showing the current workspace. It has columns for Name and Value. Text: **Workspace**
Explore data that you create or import from files
- Command History** (purple border): A window showing the history of commands entered. Text: **Command History**
View or rerun commands that you entered at the command line

Help and Documentation

View an abbreviated text version of the function documentation in the Command Window using the **help** command



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.

>> help mean

mean    Average or mean value.

For vectors, mean(X) is the mean value of the elements in X. For
matrices, mean(X) is a row vector containing the mean value of
each column. For N-D arrays, mean(X) is the mean value of the
elements along the first non-singleton dimension of X.

mean(X,DIM) takes the mean along the dimension DIM of X.

Example: If X = [1 2 3; 3 3 6; 4 6 8; 4 7 7];

then mean(X,1) is [3.0000 4.5000 6.0000] and
mean(X,2) is [2.0000 4.0000 6.0000 6.0000].

Class support for input X:
    float: double, single

See also median, std, min, max, var, cov, mode.

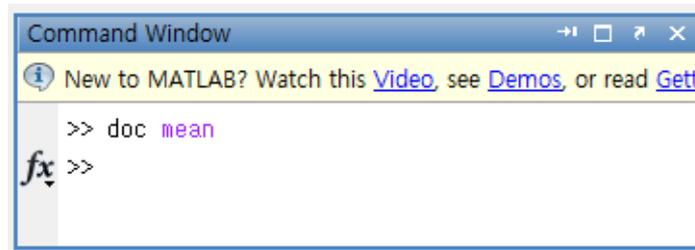
Overloaded methods:
fints/mean
ProbDistUnivParam/mean
timeseries/mean

Reference page in Help browser
doc mean

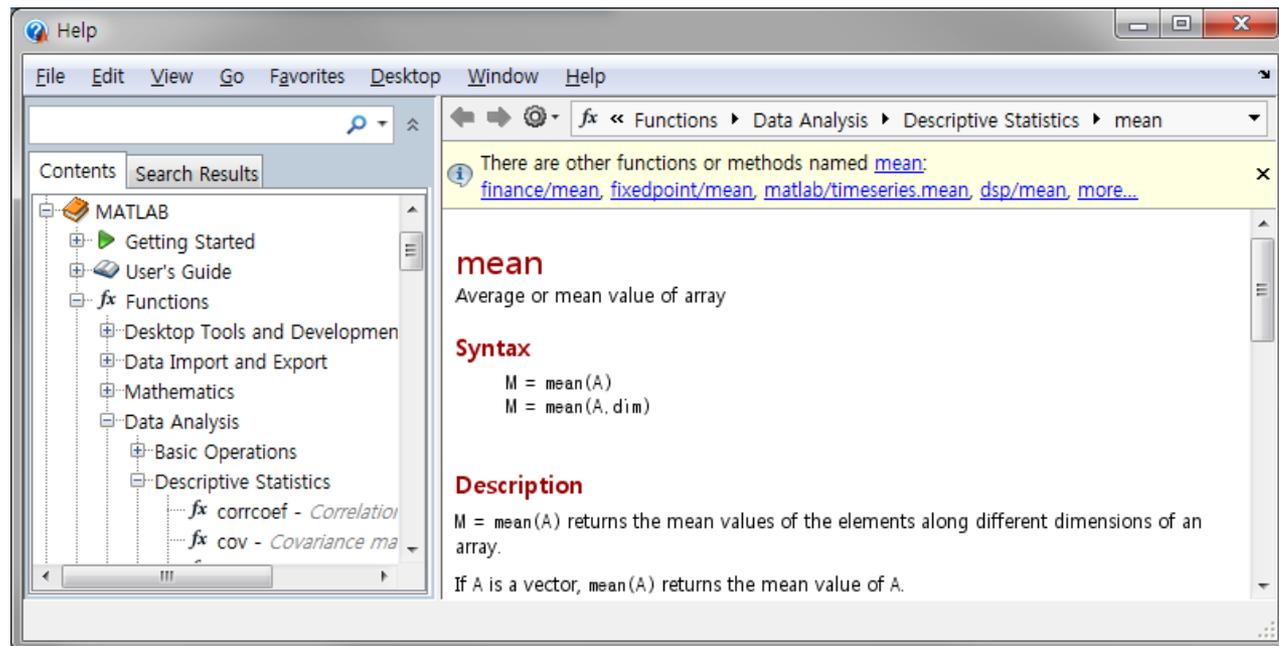
fx >> |
```

Help and Documentation

Open the function documentation in a separate window using the **doc** command.



```
Command Window  
New to MATLAB? Watch this Video, see Demos, or read Gett  
  
>> doc mean  
fx >>
```



Create a variable

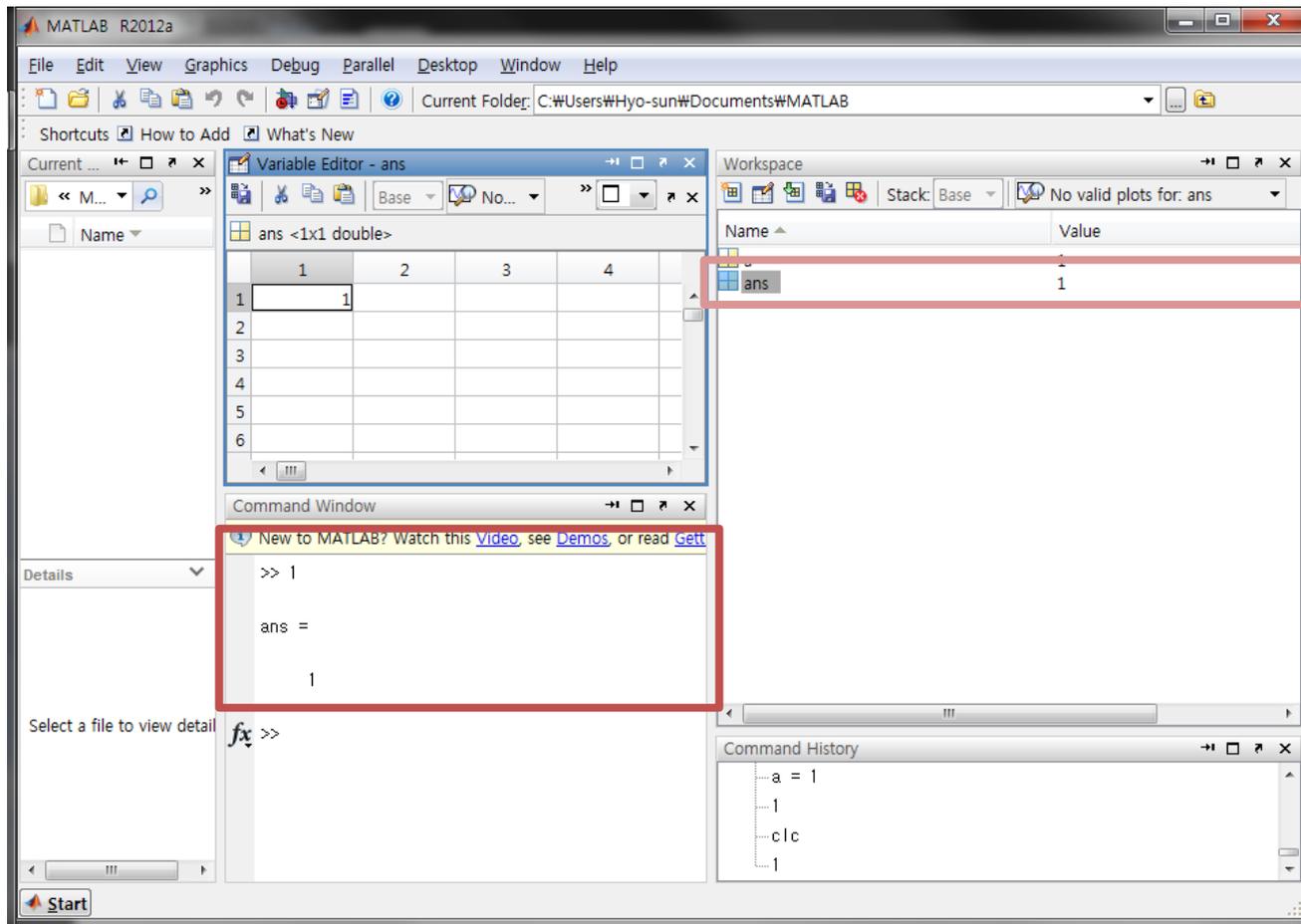
The image displays the MATLAB R2012a interface with several windows open, illustrating the process of creating a variable. Red boxes highlight the key elements:

- Variable Editor - a:** Shows a table for variable 'a' with the title 'a <1x1 double>'. The table has columns 1, 2, 3, and 4, and rows 1 through 6. The value '1' is entered in the first cell (row 1, column 1).
- Workspace:** A table with columns 'Name' and 'Value'. It shows a single entry for variable 'a' with a value of '1'.
- Command Window:** Contains the command `>> a = 1` and the output `a =` followed by `1`.
- Command History:** Shows a list of previous commands, with `a = 1` highlighted at the bottom.

The MATLAB interface includes a menu bar (File, Edit, View, Graphics, Debug, Parallel, Desktop, Window, Help), a toolbar, and a status bar at the bottom with a 'Start' button. The current folder is `C:\Users\Hyo-sun\Documents\MATLAB`.

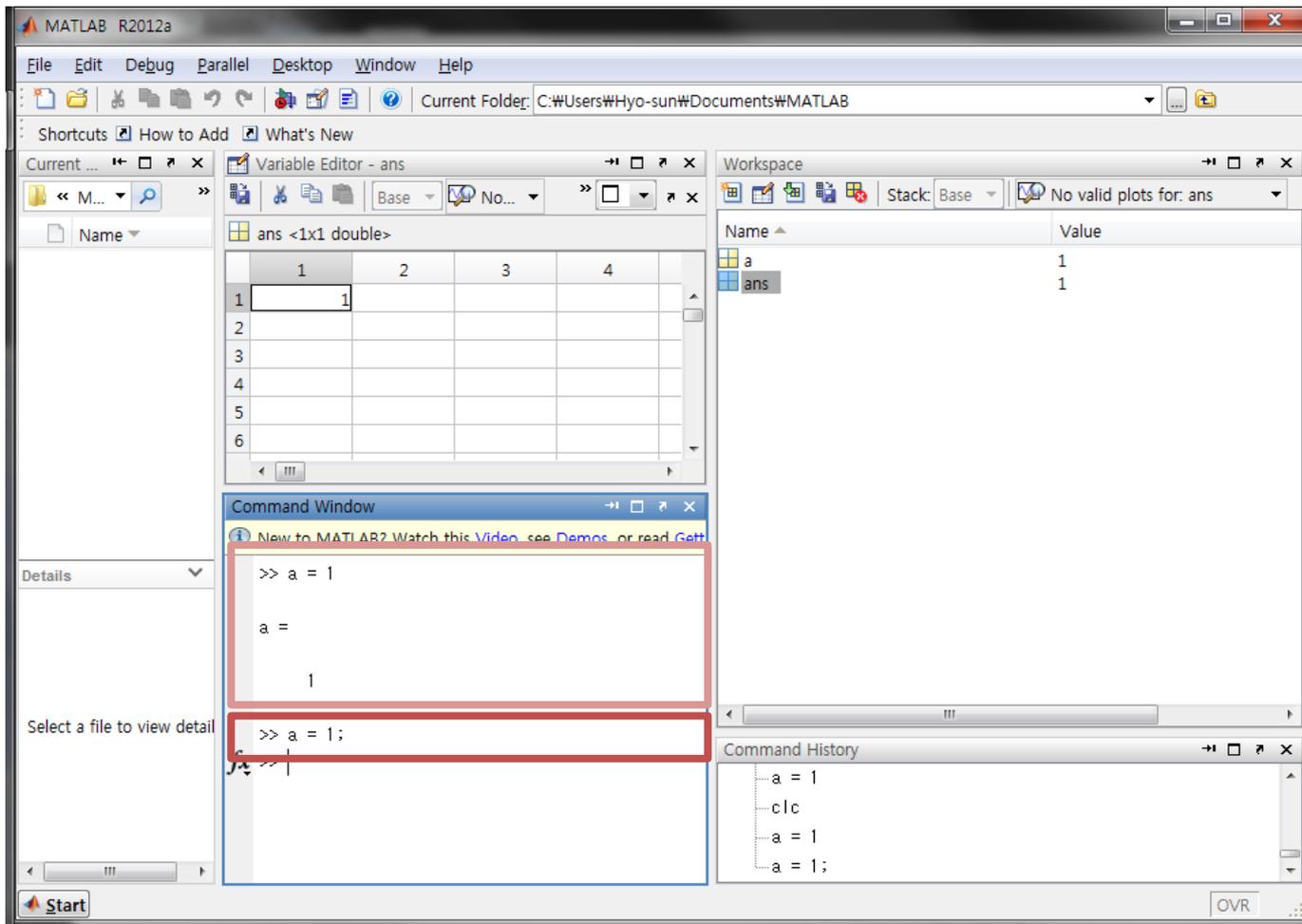
Variable ans

When you do not specify an output variable, MATLAB uses the variable **ans**, short for answer, to store the results of your calculation.



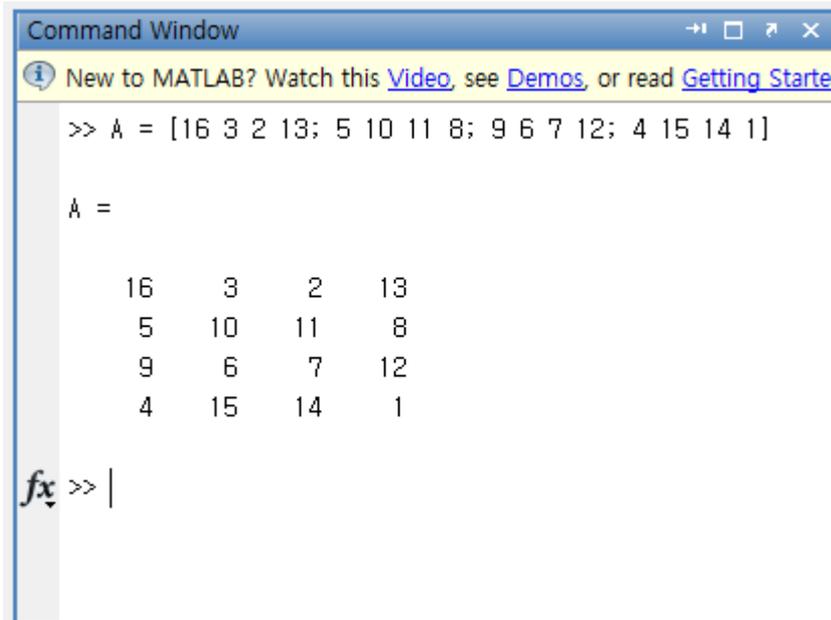
End with a semicolon

If you end a statement with a semicolon, MATLAB performs the computation, but suppresses the display of output in the Command Window.



Matrices and Arrays

- MATLAB is an abbreviation for "matrix laboratory." While other programming languages mostly work with numbers one at a time, MATLAB is designed to operate primarily on whole matrices and arrays.
- All MATLAB variables are multidimensional arrays, no matter what type of data. A matrix is a two-dimensional array often used for linear algebra.



Command Window

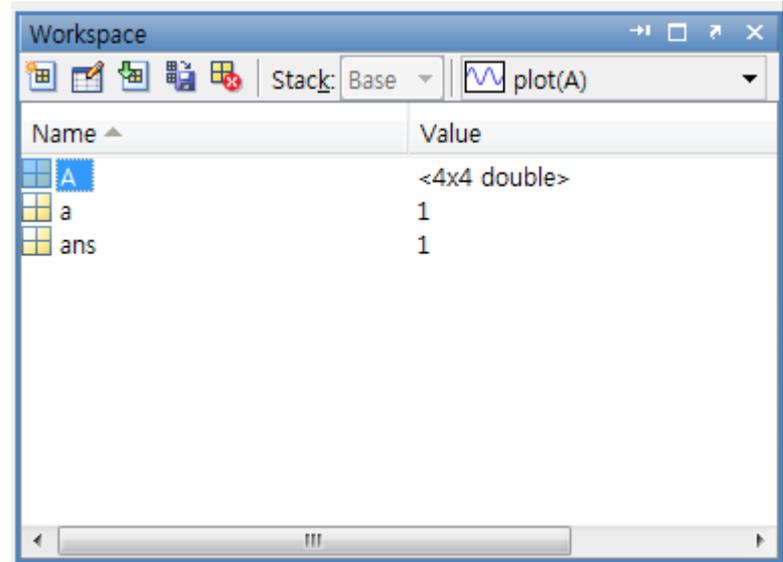
New to MATLAB? Watch this [Video](#), see [Demos](#), or read [Getting Started](#)

```
>> A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1]
```

A =

```
    16     3     2    13
     5    10    11     8
     9     6     7    12
     4    15    14     1
```

fx >> |



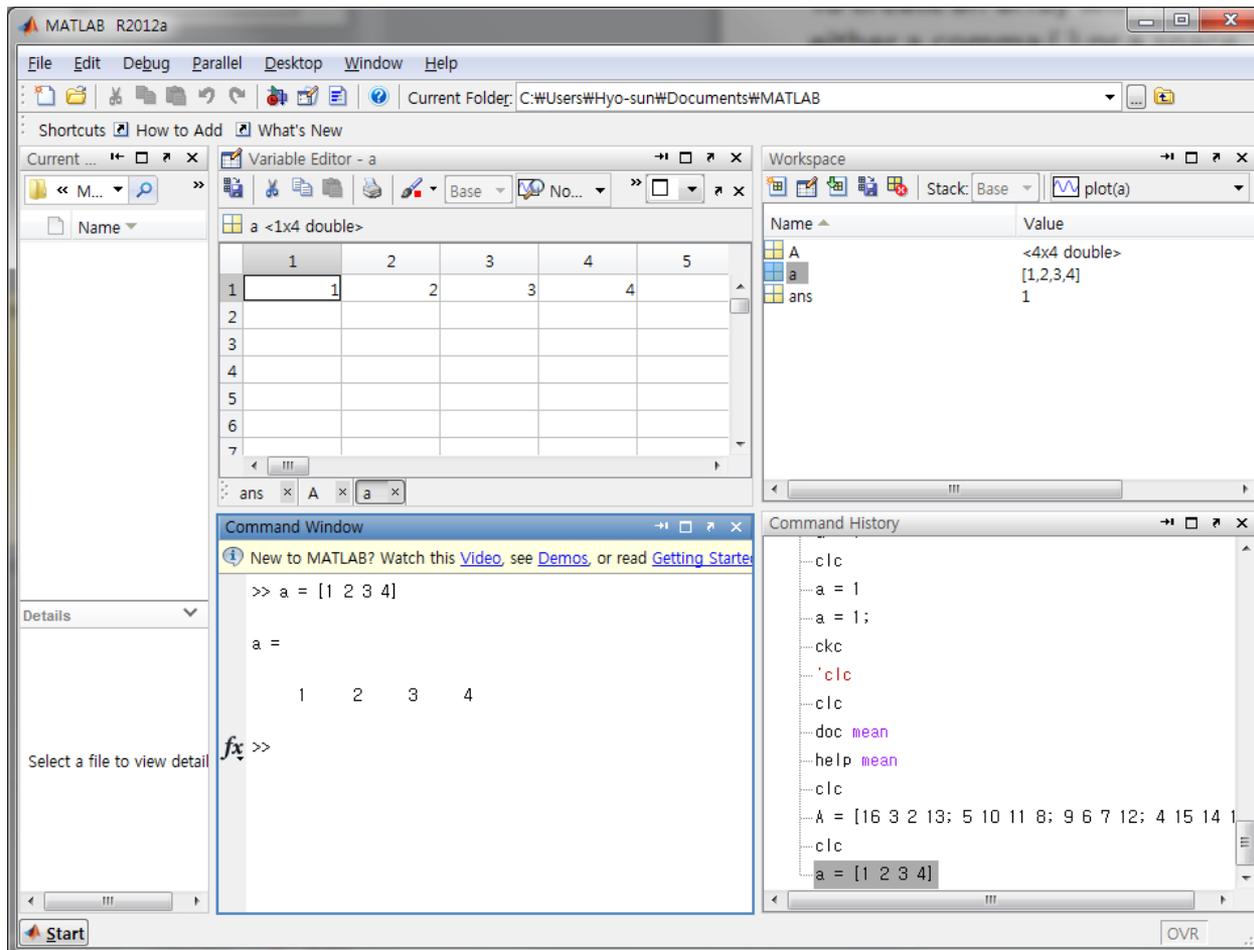
Workspace

Stack: Base | plot(A)

Name	Value
A	<4x4 double>
a	1
ans	1

Array Creation

To create an array with four elements in a single row, separate the elements with either a comma (,) or a space.



Matrix creation

To create a matrix that has multiple rows, separate the rows with semicolons.

The image displays the MATLAB R2012a software interface. The main window is titled 'MATLAB R2012a' and shows the 'Current Folder' as 'C:\Users\Hyo-sun\Documents\MATLAB'. The 'Variable Editor' window is open, showing a 3x3 matrix 'a' with the following values:

	1	2	3	4	5
1	1	2	3		
2	4	5	6		
3	7	8	9		
4					
5					
6					
7					

The 'Workspace' window shows the following variables and their values:

Name	Value
A	<4x4 double>
a	[1,2,3;4,5,6;7,8,9]
ans	1

The 'Command Window' shows the following commands and output:

```
>> a = [1 2 3; 4 5 6; 7 8 9]

a =

     1     2     3
     4     5     6
     7     8     9

fx >>
```

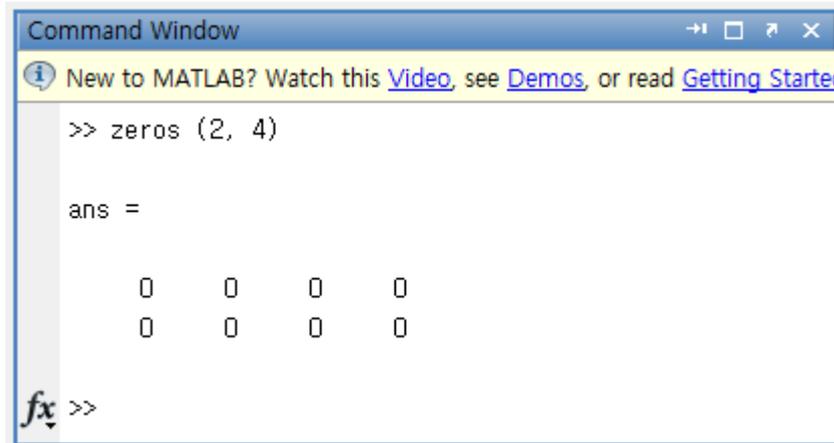
The 'Command History' window shows the following commands:

```
...clc
... 'clc
... clc
... doc mean
... help mean
... clc
... A = [16 3 2 13; 5 10 11 8; 9 6 7 12; 4 15 14 1
... clc
... a = [1 2 3 4]
... a = [1 2 3; 4 5 6; 7 8 9]
... clc
... a = [1 2 3; 4 5 6; 7 8 9]
```

Matrix creation – ones, zeros, rand, randn

MATLAB software provides four functions that generate basic matrices.

zeros all zeros



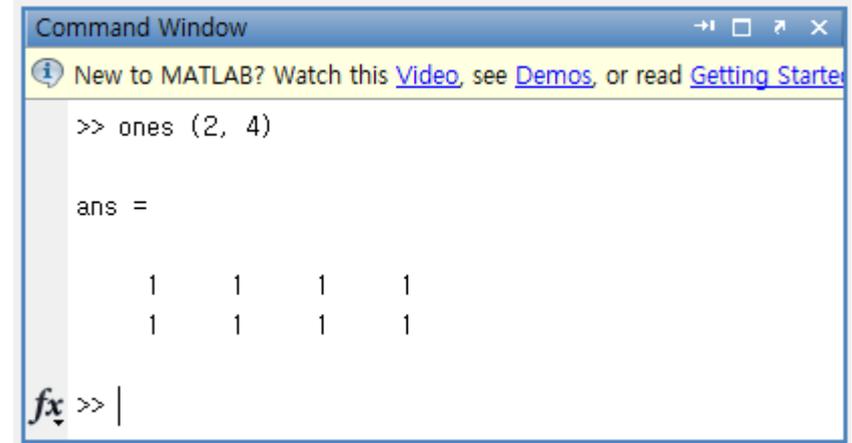
```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started
>> zeros (2, 4)

ans =

     0     0     0     0
     0     0     0     0

fx >>
```

ones all ones



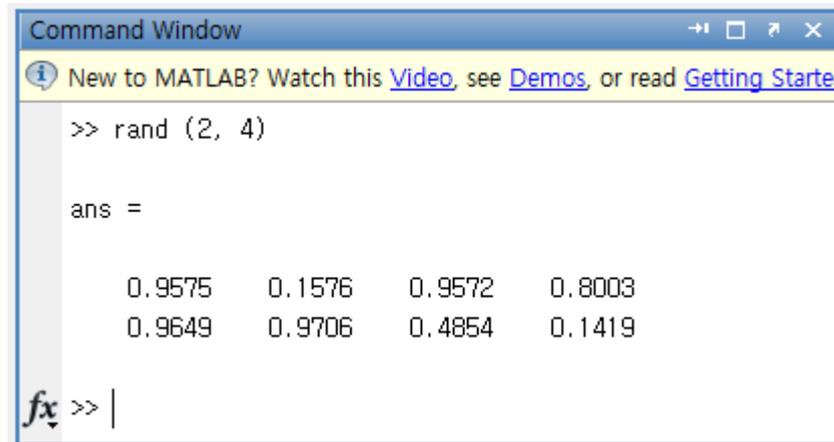
```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started
>> ones (2, 4)

ans =

     1     1     1     1
     1     1     1     1

fx >> |
```

rand uniformly distributed random number



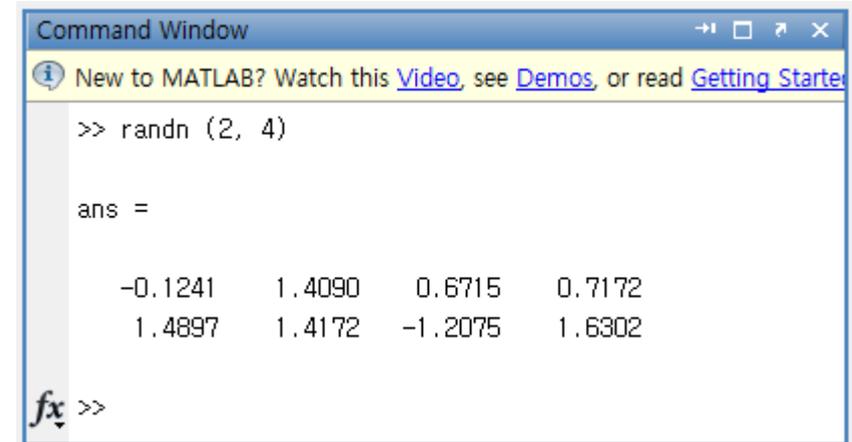
```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started
>> rand (2, 4)

ans =

    0.9575    0.1576    0.9572    0.8003
    0.9649    0.9706    0.4854    0.1419

fx >> |
```

randn normally distributed random number



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started
>> randn (2, 4)

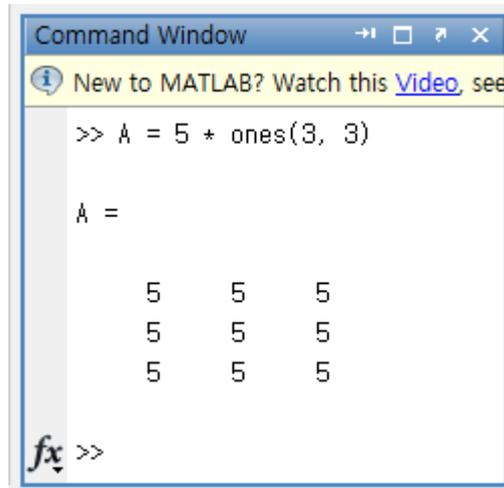
ans =

   -0.1241    1.4090    0.6715    0.7172
    1.4897    1.4172   -1.2075    1.6302

fx >>
```

Matrix creation – ones, zeros, rand, randn

Create a matrix of all fives



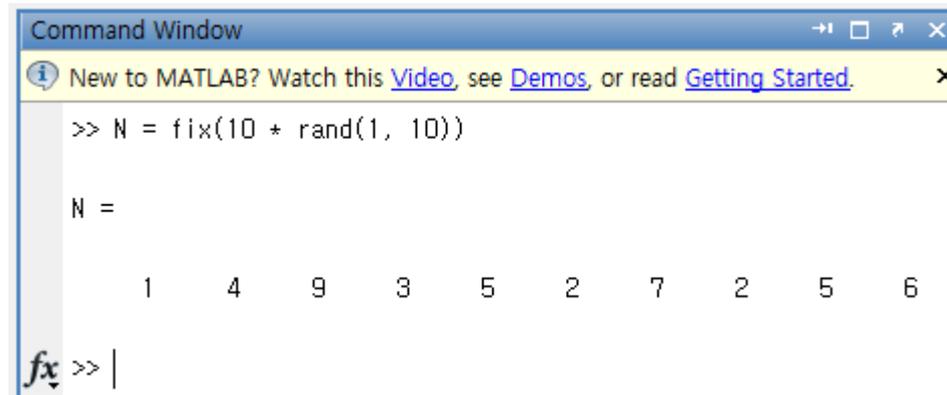
```
Command Window
New to MATLAB? Watch this Video, see Getting Started.
>> A = 5 * ones(3, 3)

A =

     5     5     5
     5     5     5
     5     5     5

fx >>
```

Create a matrix of uniformly distributed random integers between 1 and 10 inclusive



```
Command Window
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> N = fix(10 * rand(1, 10))

N =

     1     4     9     3     5     2     7     2     5     6

fx >> |
```

Matrix creation – start:step:end

The colon operator also allows you to create an equally spaced vector of values using the more general form **start:step:end**.

```
>> B = 0:10:100  
  
B =  
  
    0    10    20    30    40    50    60    70    80    90   100
```

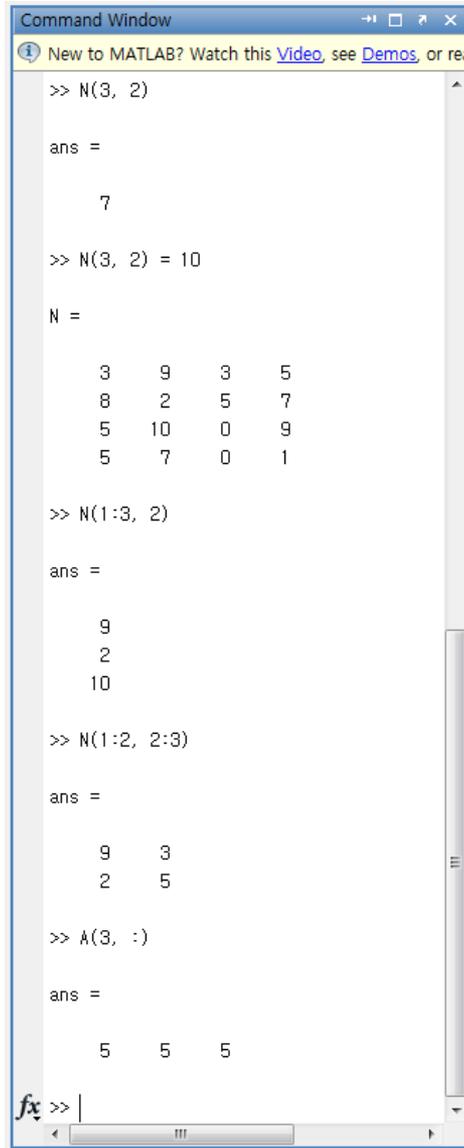
If you omit the middle step, as in **start:end**, MATLAB uses the default step value of 1.

```
>> B = 1:10  
  
B =  
  
     1     2     3     4     5     6     7     8     9    10
```

Create an array with increasing odd numbers

```
>> B = 1:2:20  
  
B =  
  
     1     3     5     7     9    11    13    15    17    19
```

Matrix indexing

$$N = \begin{bmatrix} 3 & 9 & 3 & 5 \\ 8 & 2 & 5 & 7 \\ 5 & 7 & 0 & 9 \\ 5 & 7 & 0 & 1 \end{bmatrix}$$


```
Command Window
New to MATLAB? Watch this Video, see Demos, or read the Help
>> N(3, 2)
ans =
    7
>> N(3, 2) = 10
N =
    3    9    3    5
    8    2    5    7
    5   10    0    9
    5    7    0    1
>> N(1:3, 2)
ans =
    9
    2
   10
>> N(1:2, 2:3)
ans =
    9    3
    2    5
>> A(3, :)
ans =
    5    5    5
```

Access element in row 3 and column 2

Change element in row 3 and column 2

Access elements in row 1~3 and column 2

Access elements in row 1~2 and column 2~3

Access elements in row 3

Matrix Functions – sum, min, max, mean, var, cov

A = 3 1 7 3 2

B = 4 9 3 1

0 5 8 6

1 0 0 7

9 2 0 6

Sum(A)

Sum(B, 1)

Sum(B, 2)

Min(A)

Min(B, 1)

Min(B, 2)

Max(A)

Max(B, 1)

Max(B, 2)

Mean(A)

Mean(B, 1)

Mean(B, 2)

Var(A)

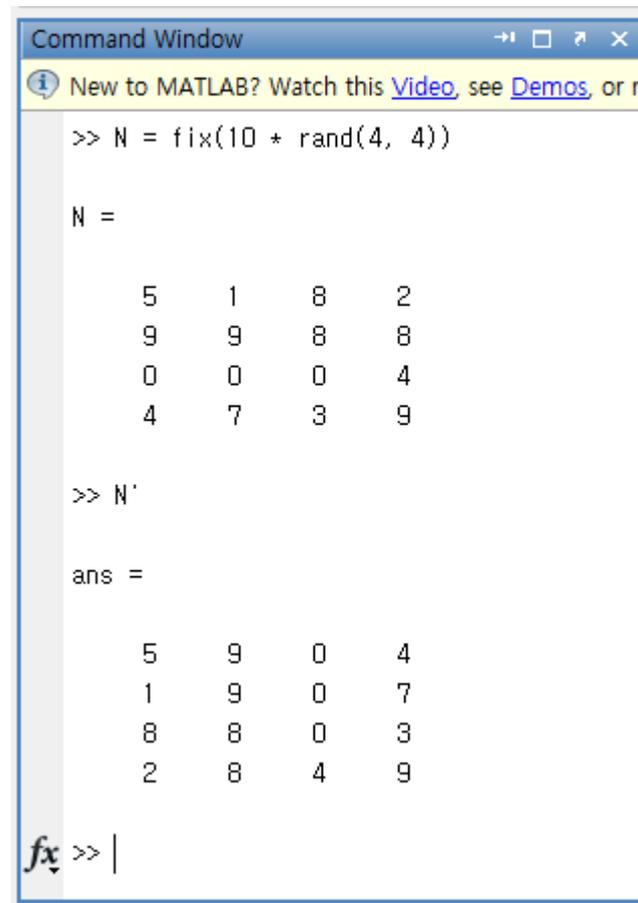
Var(B, 1)

Var(B, 2)

Cov(A)

Matrix Functions - transpose

The apostrophe operator (for example, A') performs a complex conjugate transposition. It flips a matrix about its main diagonal, and also changes the sign of the imaginary component of any complex elements of the matrix.



```
Command Window
New to MATLAB? Watch this Video, see Demos, or re
>> N = fix(10 * rand(4, 4))

N =

     5     1     8     2
     9     9     8     8
     0     0     0     4
     4     7     3     9

>> N'

ans =

     5     9     0     4
     1     9     0     7
     8     8     0     3
     2     8     4     9

fx >> |
```

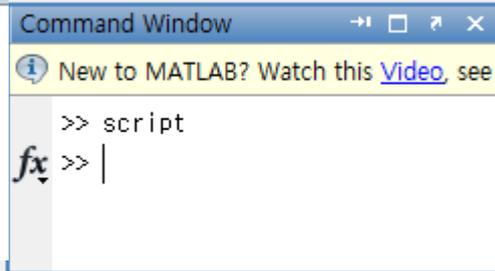
Script & Function

There are two kinds of program files:

- Scripts, which do not accept input arguments or return output arguments. They operate on data in the workspace.
- Functions, which can accept input arguments and return output arguments. Internal variables are local to the function.

Script

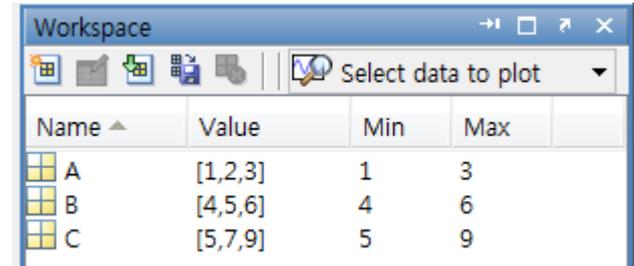
```
1 - A = [1 2 3];  
2 - B = [4 5 6];  
3 - C = A+B;
```



Command Window

New to MATLAB? Watch this [Video](#), see

```
>> script  
fx >> |
```



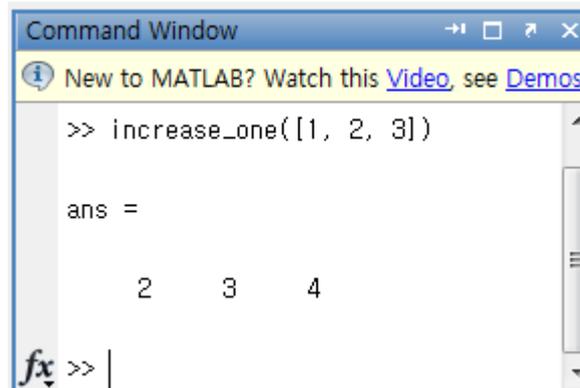
Workspace

Select data to plot

Name	Value	Min	Max
A	[1,2,3]	1	3
B	[4,5,6]	4	6
C	[5,7,9]	5	9

Function

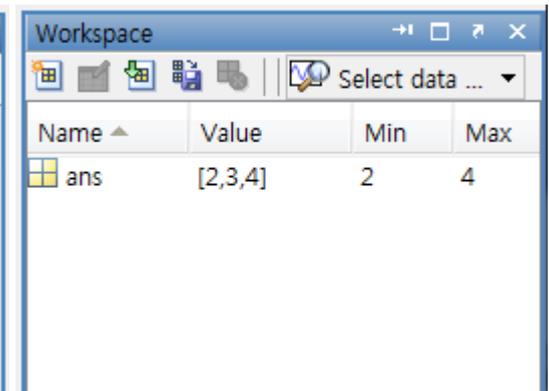
```
1 - function C = increase_one (A)  
2 -     b = 1;  
3 -     C = A + b;  
4 - end
```



Command Window

New to MATLAB? Watch this [Video](#), see [Demos](#).

```
>> increase_one([1, 2, 3])  
  
ans =  
  
     2     3     4  
  
fx >> |
```



Workspace

Select data ...

Name	Value	Min	Max
ans	[2,3,4]	2	4