MATLAB

ALIREZA GHORBI

دید کلی

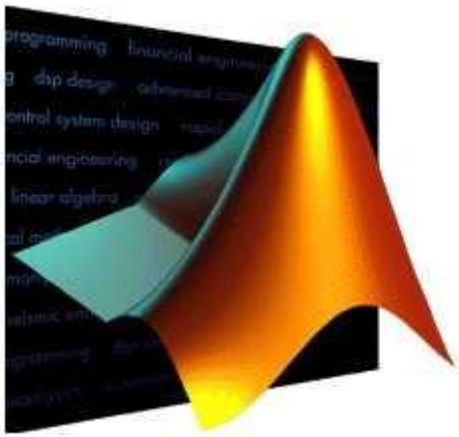
- معرفی Matlab
- محیط نرم افزار
- انواع داده ها
- مقدمات Matlab و ماتریس ها
- دستورات شرطی و حلقه های تکرار

معرفی MATLAB

MATLAB و کاربرد آن :

- MATLAB = MATrix laboratory

- متلب یک سیستم جامع نرم افزاری برای محاسبات ریاضی و محاسبات تکنیکی می باشد.
- نرم افزار متلب از زبان برنامه نویسی سطح بالا استفاده می کند که شامل صدها دستور برای محاسبات ریاضی می باشد.



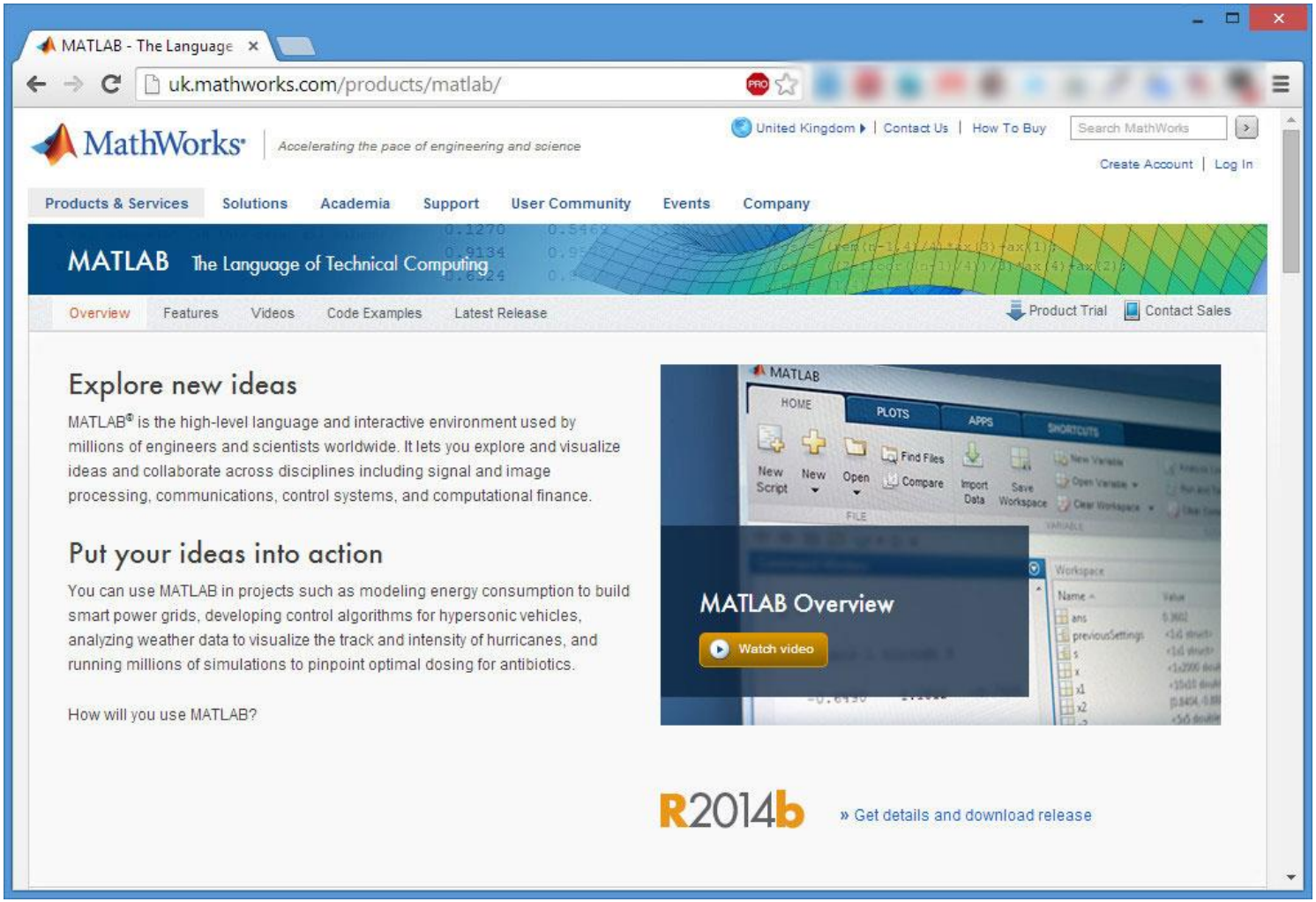
MATLAB®
The Language of Technical Computing
The MATH WORKS Inc.

MATLAB و کاربرد آن :

- نگران تعداد زیاد دستورها در متلب نباشید زیرا معمولا برای پروژه خود تنها به تعداد اندکی از آنها نیاز دارید و همچنین با Help قوی نرم افزار می توانید دستوراتی را که لازم دارید، بیابید.
- کاربردهای نرم افزار متلب بسیار وسیع می باشد. شما با متلب حتی می توانید صدا و یا انیمیشن بسازید.
- شرکت توسعه دهنده نرم افزار متلب، شرکت MathWorks می باشد.

MATLAB و کاربرد آن :

- http://www.mathworks.com/



تاریخچه نسخه های اخیر Matlab

R2015b (Version 8.6) - 3 Sep 2015	R2015a (Version 8.5) - 5 Mar 2015
R2014b (Version 8.4) - 2 Oct 2014	R2014a (Version 8.3) - 6 Mar 2014
R2013b (Version 8.2) - 5 Sep 2013	R2013a (Version 8.1) - 7 Mar 2013
R2012b (Version 8.0) - 11 Sep 2012	R2012a (Version 7.14) - 1 Mar 2012
R2010b (Version 7.11) - 3 Sep 2010	R2011a (Version 7.12) - 8 Apr 2011
R2011b (Version 7.13) - 1 Sep 2011	R2010a (Version 7.10) - 5 Mar 2010
R2009b (Version 7.9) - 4 Sep 2009	R2009a (Version 7.8) - 6 Mar 2009
R2008b (Version 7.7) - 9 Oct 2008	R2008a (Version 7.6) - 1 Mar 2008
R2007b (Version 7.5) - 1 Sep 2007	R2007a (Version 7.4) - 1 Mar 2007

دستور ver

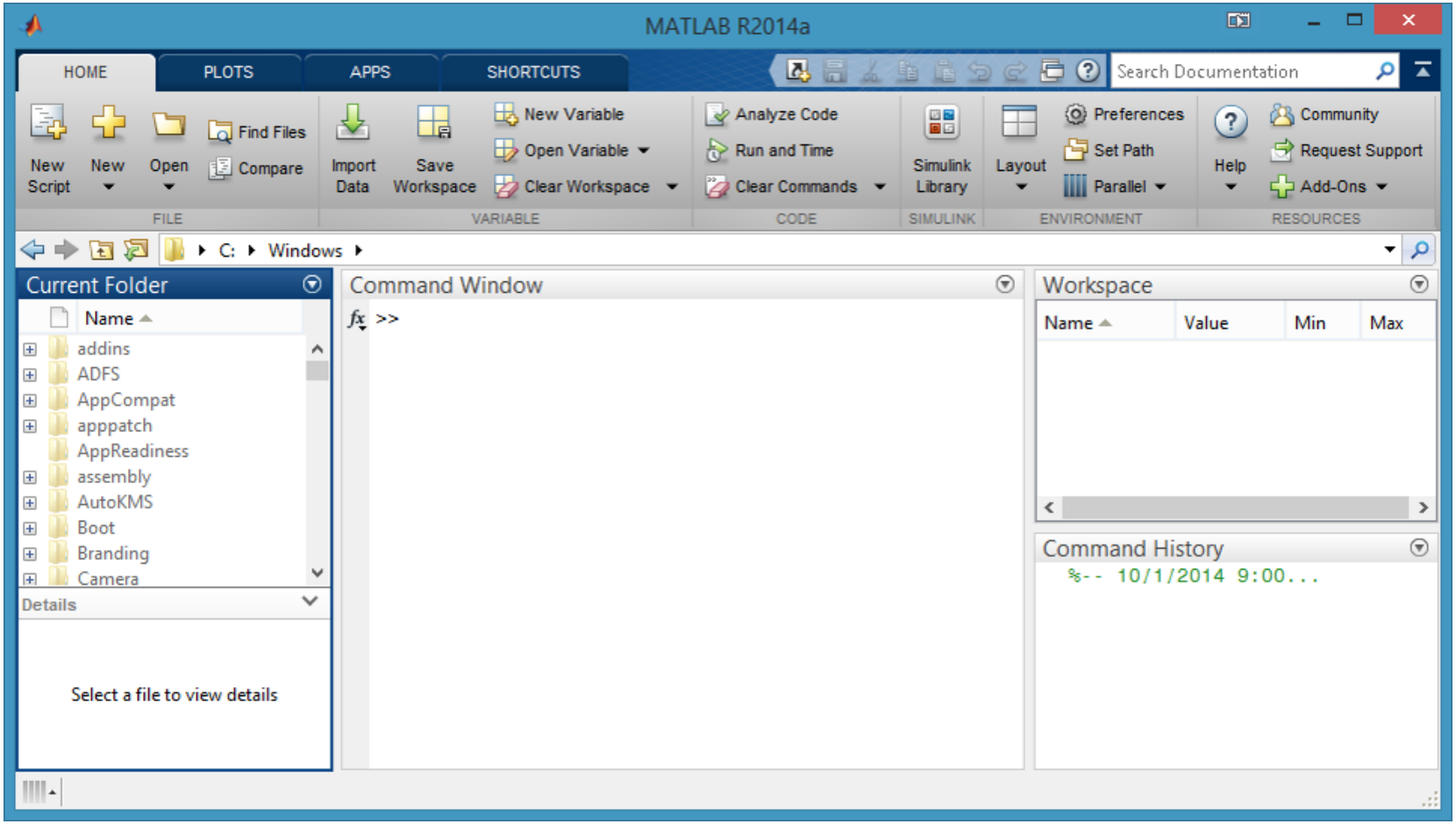
Command Window

```
>> ver
-----------------------------------------------------------------------------------------------
MATLAB Version: 8.4.0.150421 (R2014b)
MATLAB License Number: unknown
Operating System: Microsoft Windows 8.1 Pro Version 6.3 (Build 9600)
Java Version: Java 1.7.0_11-b21 with Oracle Corporation Java HotSpot(TM) 64-Bit Server VM mixed mode
-----------------------------------------------------------------------------------------------
MATLAB                                                Version 8.4        (R2014b)
Simulink                                              Version 8.4        (R2014b)
Aerospace Blockset                                    Version 3.14       (R2014b)
Aerospace Toolbox                                     Version 2.14       (R2014b)
Bioinformatics Toolbox                                Version 4.5        (R2014b)
Communications System Toolbox                         Version 5.7        (R2014b)
Computer Vision System Toolbox                        Version 6.1        (R2014b)
Control System Toolbox                                Version 9.8        (R2014b)
Curve Fitting Toolbox                                 Version 3.5        (R2014b)
DO Qualification Kit                                  Version 2.4        (R2014b)
DSP System Toolbox                                    Version 8.7        (R2014b)
Data Acquisition Toolbox                              Version 3.6        (R2014b)
Database Toolbox                                      Version 5.2        (R2014b)
Datafeed Toolbox                                      Version 5.0        (R2014b)
Econometrics Toolbox                                  Version 3.1        (R2014b)
Embedded Coder                                        Version 6.7        (R2014b)
Filter Design HDL Coder                               Version 2.9.6      (R2014b)
Financial Instruments Toolbox                         Version 2.0        (R2014b)
Financial Toolbox                                     Version 5.4        (R2014b)
Fixed-Point Designer                                  Version 4.3        (R2014b)
Fuzzy Logic Toolbox                                   Version 2.2.20     (R2014b)
Global Optimization Toolbox                           Version 3.3        (R2014b)
HDL Coder                                             Version 3.5        (R2014b)
HDL Verifier                                          Version 4.5        (R2014b)
IEC Certification Kit                                 Version 3.4        (R2014b)
Image Acquisition Toolbox                             Version 4.8        (R2014b)
Image Processing Toolbox                              Version 9.1        (R2014b)
Instrument Control Toolbox                            Version 3.6        (R2014b)
LTE System Toolbox                                    Version 1.2        (R2014b)
MATLAB Builder EX                                     Version 2.5.1      (R2014b)
MATLAB Builder JA                                     Version 2.3.2      (R2014b)
MATLAB Builder NE                                     Version 4.2.2      (R2014b)
MATLAB Coder                                          Version 2.7        (R2014b)
MATLAB Compiler                                       Version 5.2        (R2014b)
MATLAB Report Generator                               Version 4.0        (R2014b)
Mapping Toolbox                                       Version 4.0.2      (R2014b)
Model Predictive Control Toolbox                      Version 5.0        (R2014b)
Model-Based Calibration Toolbox                       Version 4.8        (R2014b)
Neural Network Toolbox                                Version 8.2.1      (R2014b)
OPC Toolbox                                           Version 3.3.2      (R2014b)
Optimization Toolbox                                  Version 7.1        (R2014b)
```

محیط متلب

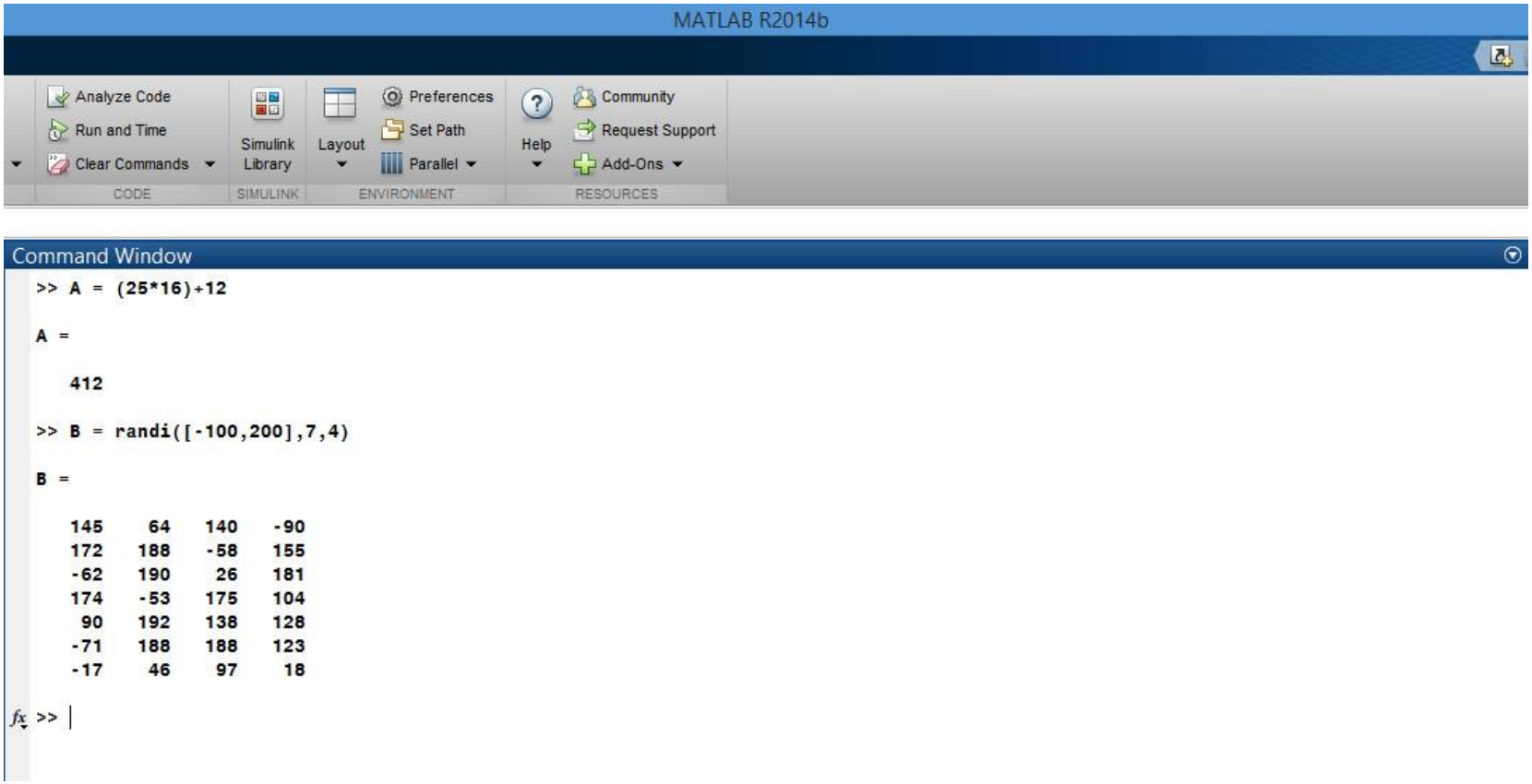
MATLAB
SIMULINK

محیط متلب :



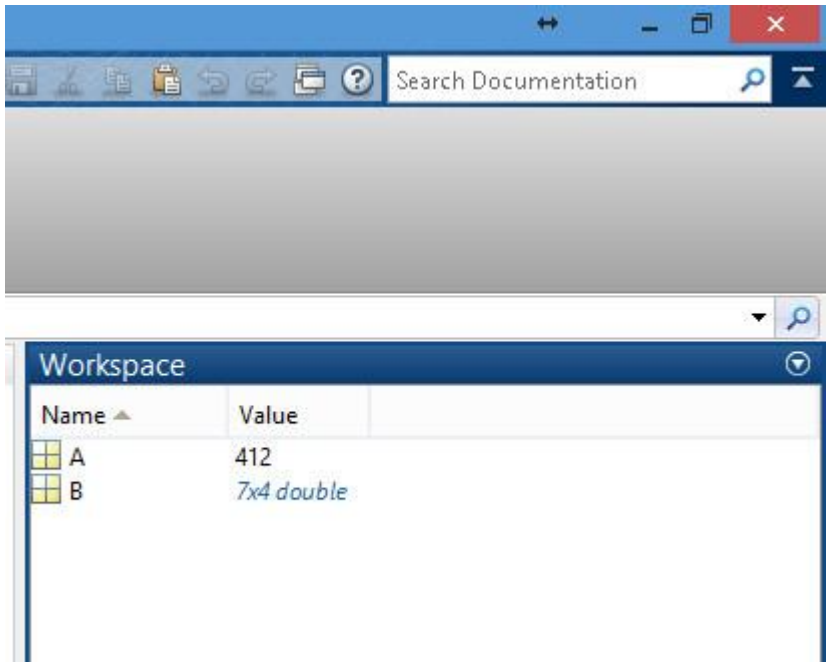
پنجره Command

- در پنجره Command می توانیم دستورات خود را نوشته و سپس با فشار دادن کلید Enter از کیبورد، نتایج اجرای دستورات را در همین پنجره مشاهده کنیم.



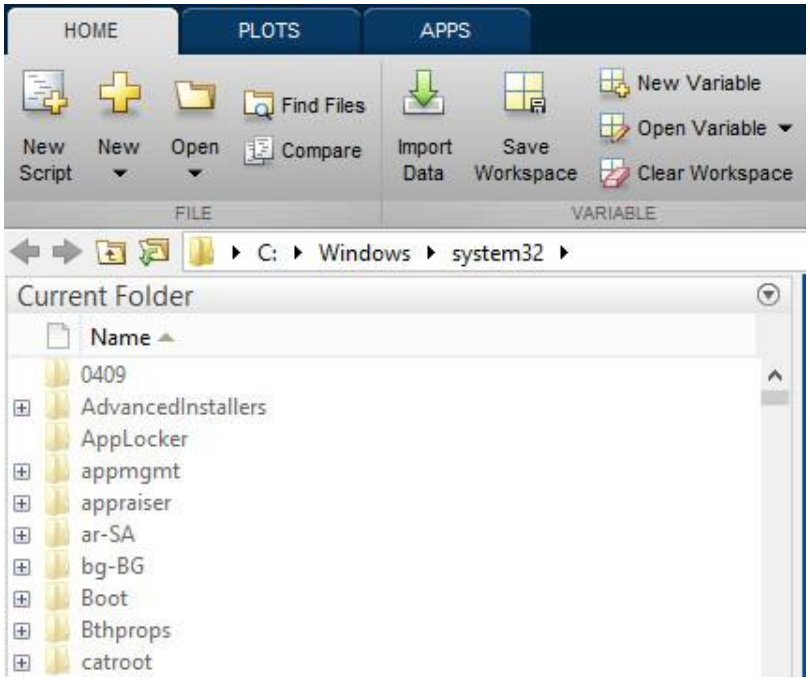
پنجره Workspace

- در پنجره Workspace لیستی از متغیرهایی که به وسیله دستورات در متلب تعریف شده است، نمایش داده می شود.



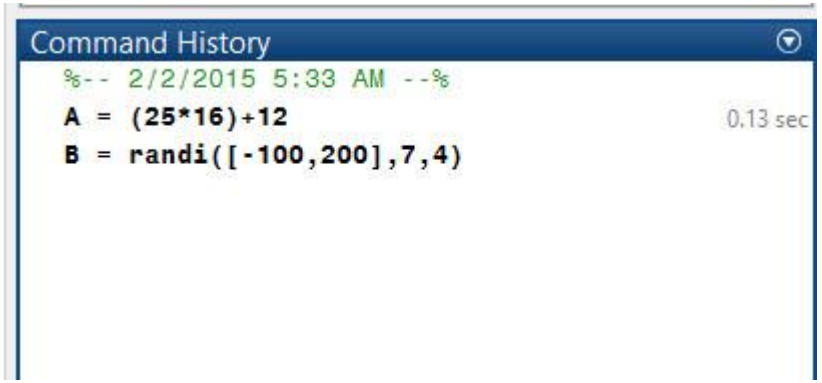
پنجره Current Folder

- در پنجره Current Folder می توانیم پوشه ای که در آن فایل های متلب مورد نظرمان وجود دارد را مشاهده کنیم و به آسانی پوشه و یا فایل های مورد نظرمان را بیابیم.



پنجره Command History

- در پنجره Command History لیستی از دستوراتی که در متلب اجرا کرده ایم، نمایش داده می شود.



انواع داده ها

انواع داده ها :

- داده های عددی (Numeric Data)
- NaN, INF و سایر ثابت های متلب
- فرمت خروجی های عددی (Format)
- حروف و جملات (Character Strings)
- Structures
- Cell Arrays

داده های عددی :

انواع داده های عددی به شرح زیر می باشند :

Numeric Types	Description
double	Convert to double precision
single	Convert to single precision
int8	Convert to 8-bit signed integer
int16	Convert to 16-bit signed integer
int32	Convert to 32-bit signed integer
int64	Convert to 64-bit signed integer
uint8	Convert to 8-bit unsigned integer
uint16	Convert to 16-bit unsigned integer
uint32	Convert to 32-bit unsigned integer
uint64	Convert to 64-bit unsigned integer

داده های عددی :

مقایسه داده های عددی مختلف با یکدیگر :

Class	Max value	Min Value	Bytes
logical	1	0	1
int8	127	-128	1
int16	32767	-32768	2
int32	2.14E+09	-2.14E+09	4
int64	9.22E+18	-9.22E+18	8
uint8	255	0	1
uint16	65535	0	2
uint32	4.29E+09	0	4
uint64	1.84E+19	0	8
single	3.4E+38	-3.4E+38	4
double	1.79e+308	-1.79e+308	8

داده های عددی :

```
Command Window
>> x = int8(32.6)

x =

    33

>> y = int8(3000)

y =

   127

>> whos x
  Name        Size                Bytes  Class     Attributes

  x           1x1                     1  int8
```

داده های عددی :

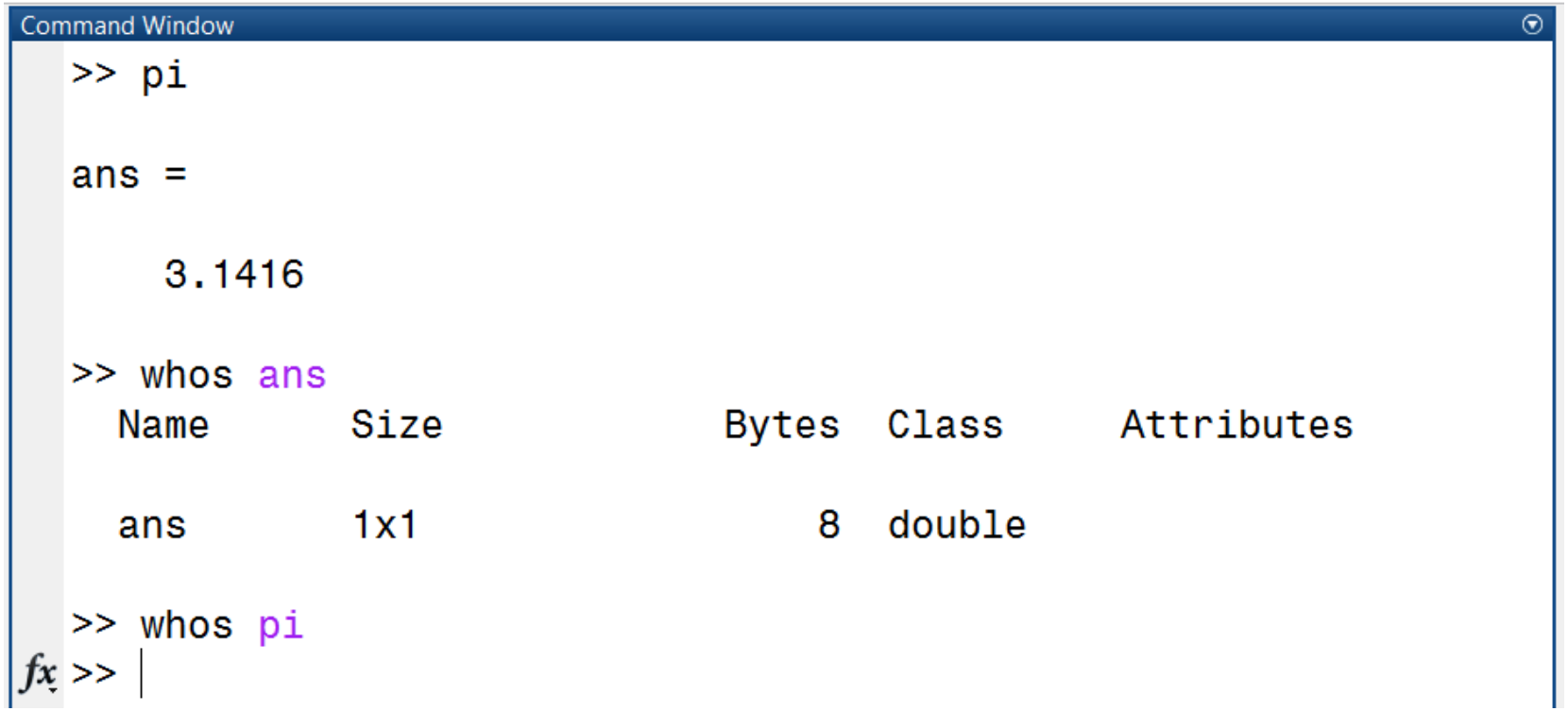
```
Command Window
>> intmax('int8')

ans =

  127

>> intmin('int8')

ans =

 -128

>> z = single(54)

z =

    54

>> whos z
  Name        Size                Bytes  Class     Attributes

  z           1x1                     4  single
```

داده های عددی :

```
>> realmax('double')

ans =

   1.7977e+308

>> realmax('single')

ans =

   3.4028e+38

fx >>
```

ثابت های متلب :

pi : عدد پی را با دقت بسیار بالا ارائه می دهد.



ثابت های متلب :

i : ثابت موهومی یا $\sqrt{-1}$ است که در اعداد مختلط ثابت موهومی را تشکیل می دهد. عدد مختلط را می توان بدین گونه تعریف کرد $x + iy$ که x قسمت حقیقی، y قسمت موهومی و i همان ثابت موهومی است.

```
Command Window
>> i

ans =

           0 +           1i

>> whos ans
  Name        Size                Bytes  Class     Attributes

  ans         1x1                    16  double    complex
```

ثابت های متلب :

```
Command Window
>> sqrt(-1)

ans =

           0 +          1i

>> whos ans
  Name       Size              Bytes  Class     Attributes

  ans        1x1                  16  double    complex

>> i^2

ans =

    -1

>> whos ans
  Name       Size              Bytes  Class     Attributes

  ans        1x1                   8  double
```

ثابت های متلب :

```
>> sqrt(-5:5)

ans =

  Columns 1 through 2

           0 +     2.2361i            0 +          2i

  Columns 3 through 4

           0 +     1.7321i            0 +     1.4142i

  Columns 5 through 6

           0 +          1i            0 +          0i

  Columns 7 through 8

           1 +          0i       1.4142 +          0i

  Columns 9 through 10

      1.7321 +          0i            2 +          0i

  Column 11

      2.2361 +          0i

>> whos ans
  Name      Size             Bytes  Class     Attributes

  ans       1x11               176  double    complex
```

ثابت های متلب :

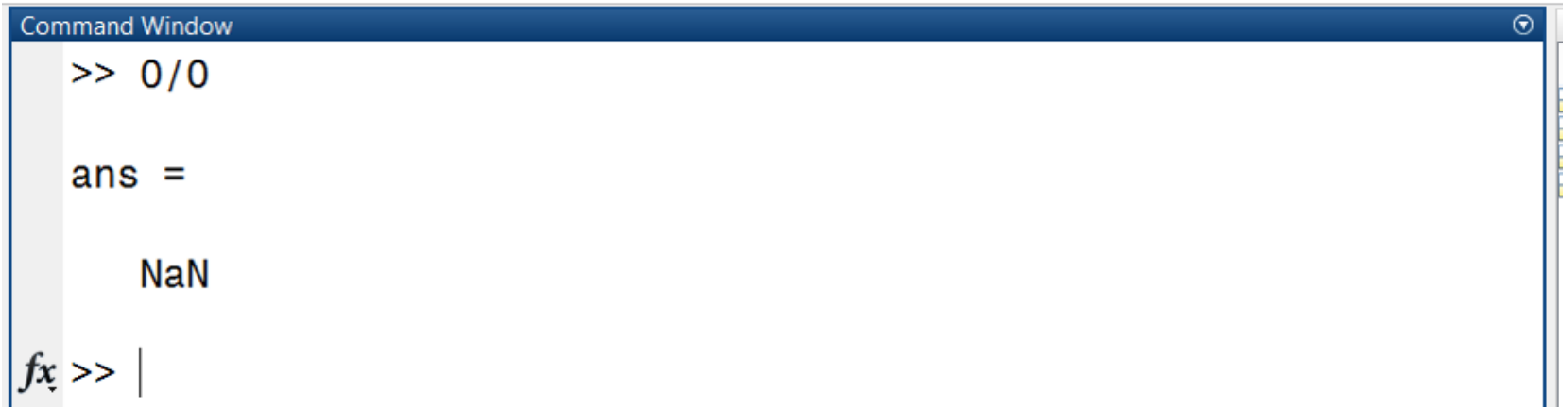
inf : بی نهایت، بیشتر در جواب های متلب دیده می شود به عبارتی جواب از محدوده شناخته شده متلب خارج است

```
Command Window
>> 250^250

ans =

   Inf

>> whos ans
  Name      Size              Bytes  Class     Attributes

  ans       1x1                   8  double

>> whos inf
>> 1/0

ans =

   Inf
```

ثابت های متلب :

NaN : مبهم، Not a Number

در صورت وقوع $\frac{0}{0}$ مشاهده می شود.



MATLAB SIMULINK

فرمت های رایج متلب :

برای مقایسه کامل در قسمت Help متلب واژه format را جستجو کنید.

Style	Result	Example
`short (default)`	Short fixed decimal format, with 4 digits after the decimal point. If you are displaying a matrix with a wide range of values, consider using `shortG`. See Display Large Data Range in short and shortg Formats	`3.1416`
`long`	Long fixed decimal format, with 15 digits after the decimal point for `double` values, and 7 digits after the decimal point for `single` values.	`3.141592653589793`
`shortE`	Short scientific notation, with 4 digits after the decimal point. Integer-valued floating-point numbers with a maximum of 9 digits do not display in scientific notation.	`3.1416e+00`
`longE`	Long scientific notation, with 15 digits after the decimal point for `double` values, and 7 digits after the decimal point for `single` values. Integer-valued floating-point numbers with a maximum of 9 digits do not display in scientific notation.	`3.141592653589793e+00`
`shortG`	The more compact of short fixed decimal or scientific notation, with 5 digits.	`3.1416`
`longG`	The more compact of long fixed decimal or scientific notation, with 15 digits for `double` values, and 7 digits for `single` values.	`3.14159265358979`
`shortEng`	Short engineering notation, with 4 digits after the decimal point, and an exponent that is a multiple of 3.	`3.1416e+000`
`longEng`	Long engineering notation, with 15 significant digits, and an exponent that is a multiple of 3.	`3.14159265358979e+000`

فرمت های رایج متلب :

برای مقایسه کامل در قسمت Help متلب واژه format را جستجو کنید.

Style	Result	Example
+	Positive/Negative format, with +, –, and blank characters displayed for positive, negative, and zero elements.	+
bank	Currency format, with 2 digits after the decimal point.	3.14
hex	Hexadecimal representation of a binary double-precision number.	400921fb54442d18
rat	Ratio of small integers.	355/113

DispType	Result	Example
compact	Suppresses excess line feeds to show more output in a single screen. Contrast with loose.	theta = pi/2 theta = 1.5708
loose	Adds linefeeds to make output more readable. Contrast with compact.	theta = pi/2 theta = 1.5708

فرمت های رایج متلب :

```
Command Window
>> format shorte
>> 10^10

ans =

   1.0000e+10

>> 6^42

ans =

   4.8123e+32
```

فرمت های رایج متلب :

```
Command Window
>> format bank
>> pi

ans =

         3.14

>> format rat
>> pi

ans =

     355/113
```

کاراکترها (Character Strings) :

```
Command Window
>> a='Civil'

a =

Civil

>> whos a
  Name        Size              Bytes  Class    Attributes

  a           1x5                  10  char

>> b='Engineering'

b =

Engineering

>> c=[a ' ' b '!']

c =

Civil Engineering!

>> whos c
  Name        Size              Bytes  Class    Attributes

  c           1x18                 36  char
```

کاراکترها (Character Strings) :

```
Command Window
>> age = 20

age =

    20

>> ['Hamid Keykavoosi is ' num2str(age) ' years old']

ans =

Hamid Keykavoosi is 20 years old

>> whos ans
  Name      Size              Bytes  Class    Attributes

  ans       1x32                 64  char

>> disp(ans)
Hamid Keykavoosi is 20 years old
>> disp('Hamid Keykavoosi is 20 years old')
Hamid Keykavoosi is 20 years old
fx >>
```

Structures :

برای مثال اگر بخواهیم اطلاعاتی را در مورد اسم، سن، آدرس و ... یک فرد یا موضوع مشابه این چنینی را در متلب ذخیره کنیم می توانیم از متغیرها استفاده کنیم و راه دیگر آن استفاده از Structure می باشد.

Command Window

```
>> person.name='Hamid'
person =
    name: 'Hamid'
>> person.age=20
person =
    name: 'Hamid'
     age: 20
>> person.age
ans =
    20
>> whos person
  Name          Size                Bytes  Class     Attributes

  person        1x1                   370  struct
```

Structures :

```
>> address.streetname='1001 Shab'
address =
    streetname: '1001 Shab'
>> address.streetnumber= 14
address =
      streetname: '1001 Shab'
    streetnumber: 14
>> person.address = address
person =
       name: 'Hamid'
        age: 20
    address: [1x1 struct]
>> whos person
  Name        Size              Bytes  Class     Attributes

  person      1x1                 924  struct

fx >>
```

Arrays :

در متلب در واقع هر داده ای نوعی Array (درایه) محسوب می شود.

```
>> x = 42
x =
    42
>> whos x
  Name        Size                Bytes  Class     Attributes

  x           1x1                     8  double

>> x(3) = 71
x =
    42     0    71
>> whos x
  Name        Size                Bytes  Class     Attributes

  x           1x3                    24  double

>> x(2,2) = 63
x =
    42     0    71
     0    63     0
>> whos x
  Name        Size                Bytes  Class     Attributes

  x           2x3                    48  double
```

Cell Arrays :

Cell ها کاری می کنند که بتوانیم داده های مختلف (عددی، کاراکترها، Structure ها) را کنار هم داشته باشیم. از { } برای سلولی کردن داده ها استفاده می شود.

Command Window

```
>> address = {14,'1001 Shab'}
address =
    [14]    '1001 Shab'
>> whos address
  Name          Size              Bytes  Class     Attributes

  address       1x2                 250  cell

>> person = {'Hamid',20,address}
person =
    'Hamid'    [20]    {1x2 cell}
>> whos person
  Name          Size              Bytes  Class     Attributes

  person        1x3                 604  cell
```

Cell Arrays :

```
>> person(1)
ans =
    'Hamid'
>> whos ans
  Name        Size                Bytes  Class     Attributes

  ans         1x1                   122  cell

>> person{1}
ans =
Hamid
>> whos ans
  Name        Size                Bytes  Class     Attributes

  ans         1x5                    10  char
```

بیشتر بدانید !

حال با استفاده از Help متلب موارد زیر را بررسی کنید :

- روش های دیگر تعریف کلمات و حروف (Strings)
- دستورات شامل num2...
- دستورات شامل cell2...
- دستور cellstr
- دستورات شامل struct2...

دستورات

دستورات :

- محاسبات اولیه و علائم
- متغیرها
- دستورات مقدماتی
- دستورات شرطی
- دستورات ورودی و خروجی

MATLAB SIMULINK

جمع و تفریق ، ضرب و تقسیم :

- + -
- * /

```
Command Window
>> 4+5
ans =
     9
>> 4*5
ans =
    20
>> 4/5
ans =
   0.800000000000000
>> 4-5
ans =
    -1
>> 4+5*9
ans =
    49
>> (4+5)*9
ans =
    81
```

توان :

- ^
- e

```
Command Window
>> 2*10^3
ans =
        2000
>> 2*10^-3
ans =
   0.002000000000000
>> 2e3
ans =
        2000
>> 2e-3
ans =
   0.002000000000000
>> 20^53
ans =
     9.007199254740992e+68
>> 300^260
ans =
   Inf
```

کار با کسرها و ساده سازی آن ها :

- sym( )
- pretty( )

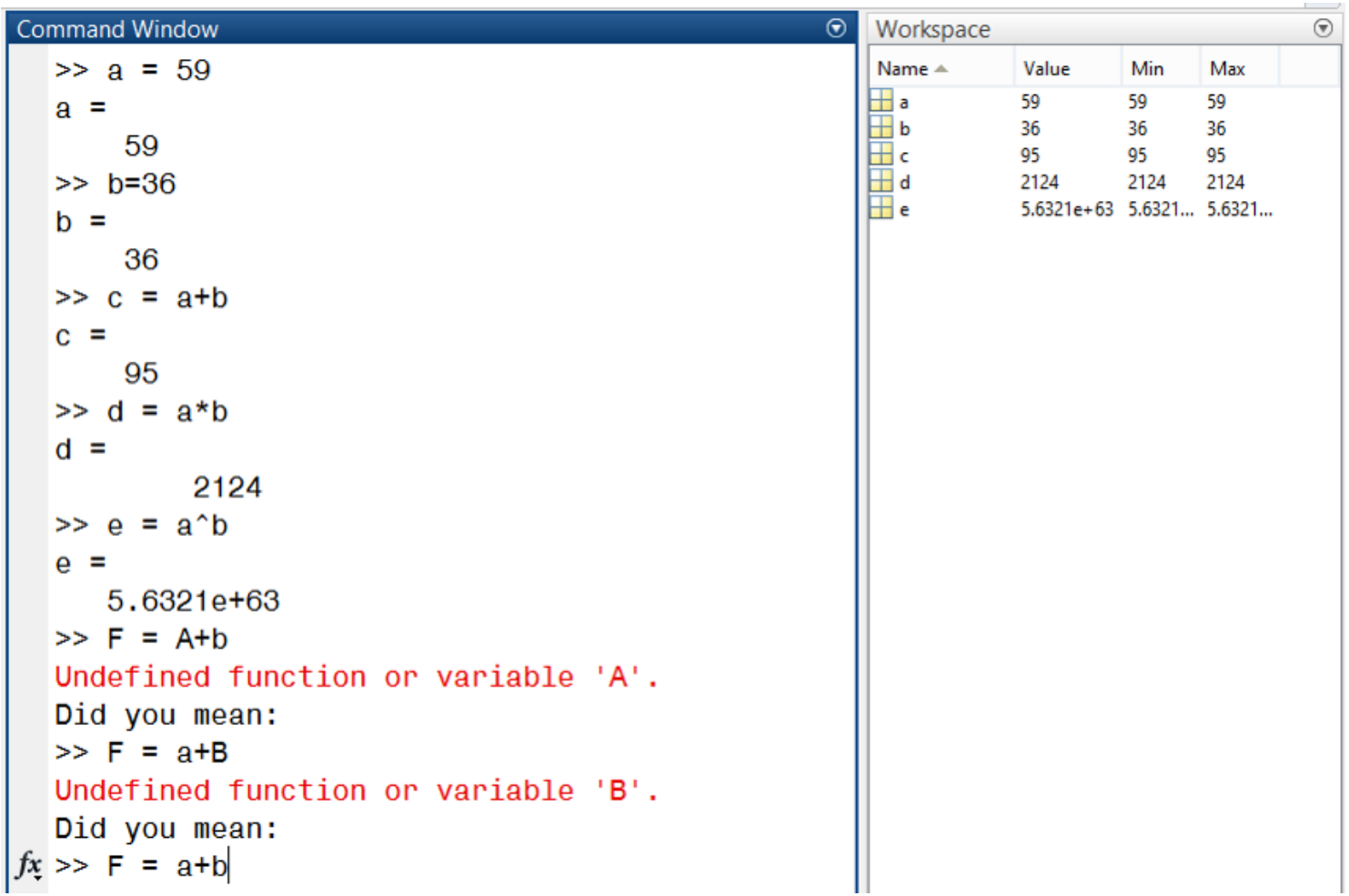
```
Command Window
>> 22/38
ans =
    0.5789
>> sym(ans)
ans =
11/19
>> sym(22/38)
ans =
11/19
>> sym(1/2 + 1/4)
ans =
3/4
>> sym(1/4 + 3/15)
ans =
9/20
>> sym(2^4 / 44)
ans =
4/11
>> pretty(ans)
 4
--
11
```

کار با کسرها و ساده سازی آن ها :

- sym( )
- pretty( )

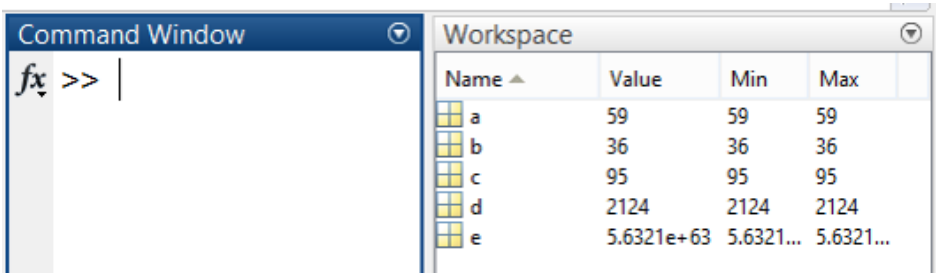
```
>> sym(105/403 + 5/8)
ans =
2855/3224
>> whos ans
  Name      Size            Bytes  Class     Attributes

  ans       1x1               112  sym

>> double(ans)
ans =
    0.8855
>> whos ans
  Name      Size            Bytes  Class     Attributes

  ans       1x1                 8  double
```

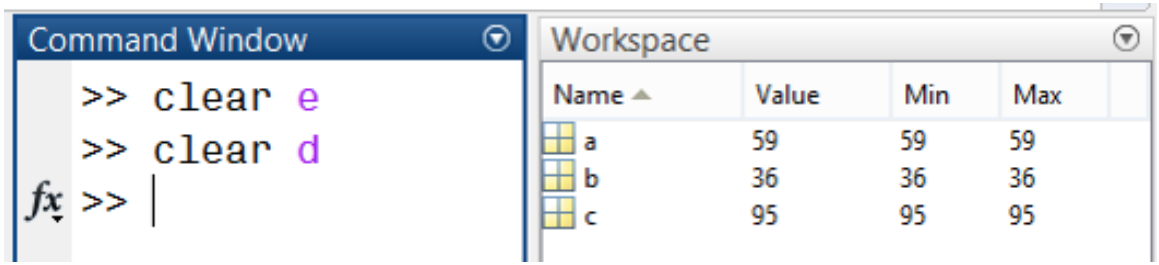
تعریف متغیرها و انجام محاسبات :

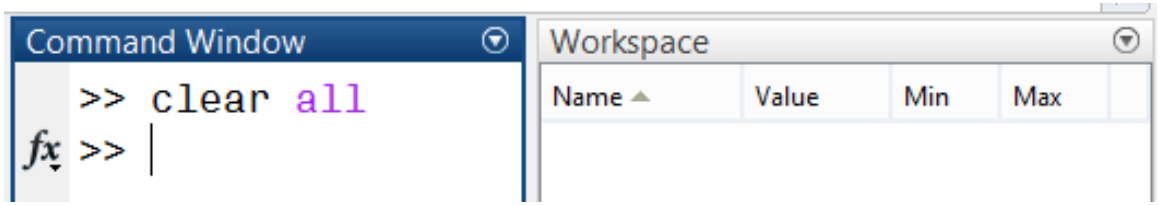


حذف متغیرها :

- clear
- clear all







فاکتور گیری ، جذر و ریشه n ام :

- factorial( )
- sqrt( )
- nthroot( )

Command Window

```
>> factorial (4)
ans =
    24
>> sqrt (81)
ans =
     9
>> sqrt (5)
ans =
    2.2361
>> nthroot(81,3)
ans =
    4.3267
>> nthroot(81,4)
ans =
     3
```

فاکتور گیری ، جذر و ریشه n ام :

- factorial( )
- sqrt( )
- nthroot( )

```
>> 81^(1/2)
ans =
     9
>> 81^(1/3)
ans =
    4.3267
>> 81^(1/4)
ans =
     3
>> sym(sqrt(40))
ans =
2*10^(1/2)
>> pretty(ans)
2 sqrt(10)
```

توابع مثلثاتی :

- sin( ) , cos( ) , tan( ) , cot( ) , sec( ) , csc( )
- asin( ) , acos( ) , atan( ) , ...
- sinh( ) , cosh( ) , tanh( ) , ...

```
Command Window
>> sin(30)

ans =

   -0.9880

>> sin(30*pi/180)

ans =

    0.5000

>> asin(0.5)*180/pi

ans =

   30.0000
```

توابع مثلثاتی :

- sin( ) , cos( ) , tan( ) , cot( ) , sec( ) , csc( )
- asin( ) , acos( ) , atan( ) , ...
- sinh( ) , cosh( ) , tanh( ) , ...

```
>> sym(sin(60*pi/180))

ans =

3^(1/2)/2

>> pretty(ans)
sqrt(3)
-------
   2
```

قدر مطلق :

- abs( )

```
Command Window
>> a = -9

a =

     -9

>> abs(a)

ans =

      9

>> abs(-9)

ans =

      9
```

تقسیم بندی بازه [a,b] :

- تقسیم بازه به گام های مساوی a:h:b
- تقسیم بازه به n نقطه با فاصله برابر linspace(a,b,n)

```
Command Window
>> a = 0:0.1:1

a =

  Columns 1 through 4

         0    0.1000    0.2000    0.3000

  Columns 5 through 8

    0.4000    0.5000    0.6000    0.7000

  Columns 9 through 11

    0.8000    0.9000    1.0000
```

تقسیم بندی بازه [a,b] :

- تقسیم بازه به گام های مساوی a:h:b
- تقسیم بازه به n نقطه با فاصله برابر linspace(a,b,n)

```
>> linspace(0,1,11)

ans =

  Columns 1 through 4

         0    0.1000    0.2000    0.3000

  Columns 5 through 8

    0.4000    0.5000    0.6000    0.7000

  Columns 9 through 11

    0.8000    0.9000    1.0000

>> linspace(0,100,4)

ans =

         0   33.3333   66.6667  100.0000
```

دستور input :

چنانچه در برنامه بخواهیم داده ای را از کاربر دریافت کنیم و به متغیری نسبت دهیم می توان از دستور input استفاده کرد.

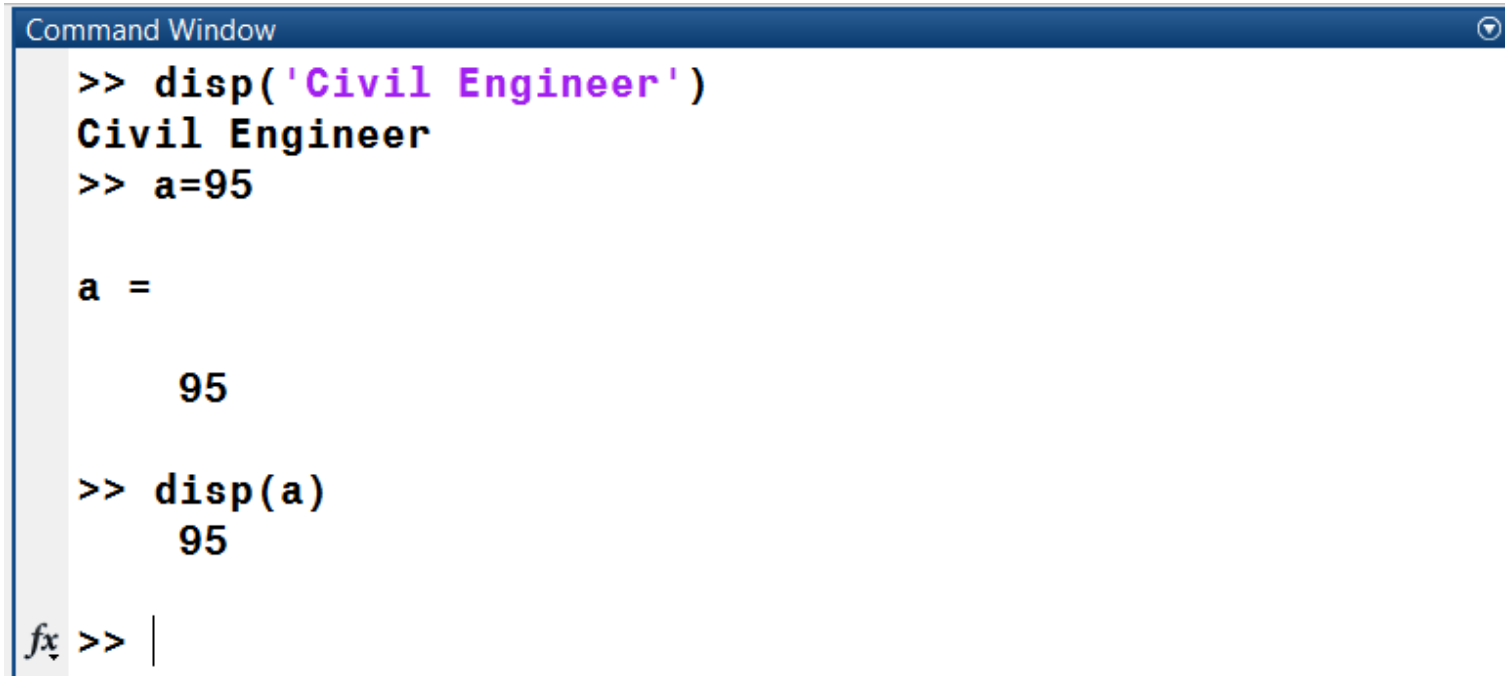
- a = input(‘text’)

```
Command Window
>> a = input('enter your number :    ')
enter your number :   57

a =

    57

fx >>
```

دستور disp :

برای نمایش مقدار یک متغیر یا نمایش یک متن در هنگام اجرای برنامه می توان از دستور disp استفاده کرد.

- disp('text')
- disp(x)



خواندن اطلاعات از Excel

- num = xlsread(filename,sheet,xlRange,'basic')
- filename : اسم فایل
- sheet : شماره صفحه اکسل
- xlRange : شماره سطر و ستون ابتدا و انتها
- 'basic‘ : با اضافه کردن این عبارت هنگام اجرای برنامه فایل اکسل از طریق خود متلب باز شده و نیازی به نصب برنامه اکسل نمی باشد، ضمن اینکه این کار باعث اجرای سریع کد می شود.

خواندن اطلاعات از Excel

```
Input = 'Input_Truss';                                          % Input file name
Sheet = 1;                                                % Input file sheet number
NoN = xlsread(Input,Sheet,'D2','basic');                        % Number of Nodes
NoE = xlsread(Input,Sheet,'D3','basic');                    % Number of Elements
NoNe = 2;                                             % Number of Nodes Per Element
NoDOFn = 2;                                 % Number of Degrees of Freedom Per Node
NoDOFe = NoNe*NoDOFn;                     % Number of Degrees of Freedom Per Element
%-------------------------------------------------------------------------
% NODES COORDINATES (x,y)
Geometry = xlsread(Input,Sheet,'C6:D500','basic');
%-------------------------------------------------------------------------
% ELEMENT CONoNeCTIVITY
Connectivity = xlsread(Input,Sheet,'O6:P1000','basic');
%-------------------------------------------------------------------------
% MATERIALS PROPERTIES
Material = xlsread(Input,Sheet,'Q6:R1000','basic');
%-------------------------------------------------------------------------
% BOUNDARY CONDITIONS
Nodal_Freedom = xlsread(Input,Sheet,'K6:L500','basic');
%-------------------------------------------------------------------------
% NODAL LOADS (FORCES)
Nodal_Loads = xlsread(Input,Sheet,'G6:H500','basic');
```

ماتریس ها

ماتریس ها :

- ساخت ماتریس
- کار با ماتریس ها
- مرتب کردن و شکل دهی ماتریس
- ترانهاده و معکوس کردن
- ماتریس های ۳ بعدی

MATLAB SIMULINK

ساخت ماتریس ۱ بعدی (بردار) :

```
Command Window
>> x = [1 2 3 4]
x =
     1     2     3     4
>> x = [1,2,3,4]
x =
     1     2     3     4
>> x = [1:4]
x =
     1     2     3     4
>> x = [1;2;3;4]
x =
     1
     2
     3
     4
>> x = [1
2
3
4]
x =
     1
     2
     3
     4
```

MATLAB SIMULINK

ساخت ماتریس ۱ بعدی (بردار) :

```
Command Window
>> x = 1:10
x =
     1     2     3     4     5     6     7     8     9    10
>> x = 1:2:10
x =
     1     3     5     7     9
>> x = 20:-4:0
x =
    20    16    12     8     4     0
>> [1:15]
ans =
  Columns 1 through 11
     1     2     3     4     5     6     7     8     9    10    11
  Columns 12 through 15
    12    13    14    15
```

MATLAB SIMULINK

ساخت ماتریس ۲ بعدی :

Command Window

```
>> y = [1 2;3 4]
y =
     1     2
     3     4
>> y = [1,2;3,4]
y =
     1     2
     3     4
>> y = [1:3;3:-1:1;4 8 9]
y =
     1     2     3
     3     2     1
     4     8     9
```

MATLAB SIMULINK

ماتریس واحد :

Command Window

```
>> eye(1,4)
ans =
     1     0     0     0
>> eye(4)
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
>> eye(4,4)
ans =
     1     0     0     0
     0     1     0     0
     0     0     1     0
     0     0     0     1
>> eye(3,5)
ans =
     1     0     0     0     0
     0     1     0     0     0
     0     0     1     0     0
```

MATLAB SIMULINK

ماتریس با درایه های واحد :

Command Window

```
>> ones(1,4)
ans =
     1     1     1     1
>> ones(4)
ans =
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
>> ones(4,4)
ans =
     1     1     1     1
     1     1     1     1
     1     1     1     1
     1     1     1     1
>> ones(4,3)
ans =
     1     1     1
     1     1     1
     1     1     1
     1     1     1
```

MATLAB SIMULINK

ماتریس با درایه های صفر :

Command Window

```
>> zeros(1,4)
ans =
     0     0     0     0
>> zeros(5)
ans =
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
     0     0     0     0     0
>> zeros(2,4)
ans =
     0     0     0     0
     0     0     0     0
>> zeros(4,4)
ans =
     0     0     0     0
     0     0     0     0
     0     0     0     0
     0     0     0     0
```

ماتریس قطری و ماتریس جادویی :

Command Window

```
>> a = [4 11 2 0 5]
a =
     4    11     2     0     5
>> b = diag(a)
b =
     4     0     0     0     0
     0    11     0     0     0
     0     0     2     0     0
     0     0     0     0     0
     0     0     0     0     5
>> c = magic(4)
c =
    16     2     3    13
     5    11    10     8
     9     7     6    12
     4    14    15     1
>> d = diag(c)
d =
    16
    11
     6
     1
```

نحوه سطر و ستون گیری :

```
Command Window
>> a = randi([-25,25],5,4)
a =
     8    25    -1    12
     7   -17    -8   -12
   -11   -20    23    -4
    -3    -7    21     2
   -25   -15   -23    23
>> b = a(2)
b =
     7
>> b = a(5)
b =
   -25
>> b = a(2,2)
b =
   -17
>> b = a(3,4)
b =
    -4
```

نحوه سطر و ستون گیری :

```
>> b = a(3,:)
b =
   -11   -20    23    -4
>> b = a(: , 2)
b =
    25
   -17
   -20
    -7
   -15
>> c = a ( : , : )
c =
     8    25    -1    12
     7   -17    -8   -12
   -11   -20    23    -4
    -3    -7    21     2
   -25   -15   -23    23
fx >>
```

MATLAB SIMULINK

عملیات ماتریسی :

```
Command Window
>> a= randi([-5,12],4,4)
a =
    -2    10    -1     2
     0    -2     2    -2
     3    -1     0    11
    -1    -2    11    12
>> b = randi([-5,12],4,4)
b =
     2     5    -2     2
    -3    -1    -3     4
    -1     5     0    -4
     2     7     0    -1
>> c = a + b
c =
     0    15    -3     4
    -3    -3    -1     2
     2     4     0     7
     1     5    11    11
>> d = a * b
d =
   -29   -11   -26    38
     0    -2     6   -14
    31    93    -3    -9
    17   136     8   -66
>> e = b - a
e =
     4    -5    -1     0
    -3     1    -5     6
    -4     6     0   -15
     3     9   -11   -13
```

MATLAB SIMULINK

عملیات درایه به درایه روی ماتریس ها :

- + -
- .* ./ .^

Command Window

```
>> f = a .* b
f =
    -4    50     2     4
     0     2    -6    -8
    -3    -5     0   -44
    -2   -14     0   -12
>> g = a.^ b
g =
   1.0e+05 *
    0.0000    1.0000    0.0000    0.0000
       Inf   -0.0000    0.0000    0.0002
    0.0000   -0.0000    0.0000    0.0000
    0.0000   -0.0013    0.0000    0.0000
>> h = b ./ a
h =
   -1.0000    0.5000    2.0000    1.0000
      -Inf    0.5000   -1.5000   -2.0000
   -0.3333   -5.0000       NaN   -0.3636
   -2.0000   -3.5000         0   -0.0833
```

ترانهاده ماتریس :

- علامت پریم

```
Command Window
>> a = 1:2:10
a =
     1     3     5     7     9
>> b = a'
b =
     1
     3
     5
     7
     9
>> b = (1:2:10)'
b =
     1
     3
     5
     7
     9
>> c = randi([-4,6],2,3)
c =
     4     6     1
    -4     4     2
>> d = c'
d =
     4    -4
     6     4
     1     2
>> d = (randi([-4,6],2,3))'
d =
    -2     1
     6     2
     1    -2
```

MATLAB SIMULINK

دترمینان ماتریس :

- det( )

```
Command Window
>> a = [5 1 3; 4 9 7; 2 0 6]
a =
     5     1     3
     4     9     7
     2     0     6
>> b = det(a)
b =
   206
>> b = det([5 1 3; 4 9 7; 2 0 6])
b =
   206
```

MATLAB SIMULINK

معکوس ماتریس :

- inv( )
- ^(-1)

```
Command Window
>> a = [5 1 3; 4 9 7; 2 0 6]
a =
     5     1     3
     4     9     7
     2     0     6
>> b = inv(a)
b =
    0.2621   -0.0291   -0.0971
   -0.0485    0.1165   -0.1117
   -0.0874    0.0097    0.1990
>> b = inv([5 1 3; 4 9 7; 2 0 6])
b =
    0.2621   -0.0291   -0.0971
   -0.0485    0.1165   -0.1117
   -0.0874    0.0097    0.1990
>> b = a^-1
b =
    0.2621   -0.0291   -0.0971
   -0.0485    0.1165   -0.1117
   -0.0874    0.0097    0.1990
```

MATLAB SIMULINK

مرتبه ماتریس :

- rank( )

```
Command Window
>> a = randi([-100,100],7,7)
a =
   -61    24   -95   -89    64    60   -22
   -94    72    -2    37    45    -9    67
    49    61   -67   -92   -70   -14    61
     0    15    96   -86    32    65   -88
    -4   -64    43     4     4   -84   -20
    81   -52     0   -81    95   -74     5
    22    78    -6    64    30   -66   -17
>> b = rank(a)
b =
     7
>> c = magic(5)
c =
    17    24     1     8    15
    23     5     7    14    16
     4     6    13    20    22
    10    12    19    21     3
    11    18    25     2     9
>> d = rank(c)
d =
     5
```

MATLAB SIMULINK

ابعاد ماتریس :

- size ( )

```
Command Window
>> a = [1 2 3;4 5 6]

a =

     1     2     3
     4     5     6

>> size(a)

ans =

     2     3

fx >>
```

طول ماتریس :

این دستور، تعداد ردیف ها و تعداد ستون های ماتریس را محاسبه می کند و هر کدام از این دو عدد که بزرگتر باشد را در خروجی نمایش خواهد داد.

- length ( )

Command Window

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

a =

     1     2     3
     4     5     6

>> length(a)

ans =

     3
```

تعداد کل عناصر ماتریس :

- numel ( )

```
Command Window
>> a = randi([-5,5],6,7)

a =

    -1    -5    -5     0     2    -2     2
     2    -4    -1    -1     2     1     4
    -4     4    -1     2    -4    -3     5
     2     2     3     2    -4     3     1
    -5    -2     3     3     0    -3    -4
    -2     5    -3    -2     5     0    -4

>> numel(a)

ans =

    42
```

MATLAB SIMULINK

مرتب کردن ماتریس :

- sort ( )

Command Window

```
>> a = randi([-20,20],4,7)

a =

   -10   -11   -10    14    -9     3    11
    14    18     5     3    11   -17    18
   -10    -6    -1     2    10   -18   -15
    13   -12    -6    17    -5     1     3

>> sort(a)

ans =

   -10   -12   -10     2    -9   -18   -15
   -10   -11    -6     3    -5   -17     3
    13    -6    -1    14    10     1    11
    14    18     5    17    11     3    18
```

ماکزیمم و مینیمم ماتریس :

- max( ) max(a,[ ],dim)
- min ( ) min(a,[ ],dim)

```
>> a = magic(4)

a =

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

>> max(a)

ans =

    16    14    15    13
```

ماکزیمم و مینیمم ماتریس :

- max( ) max(a,[ ],dim)
- min ( ) min(a,[ ],dim)

```
>> max(a,[],1)

ans =

    16    14    15    13

>> max(a,[],2)

ans =

    16
    11
    12
    15
```

ترکیب بردار یا ماتریس :

```
Command Window
>> a = [1 2 3]

a =

     1     2     3

>> b = [4 5 6]

b =

     4     5     6

>> c = [a;b]

c =

     1     2     3
     4     5     6
```

ترکیب بردار یا ماتریس :

- ترکیب افقی با horzcat
- ترکیب عمودی با vertcat

```
Command Window
>> c = horzcat(a,b)

c =

     1     2     3     4     5     6

>> d = vertcat(a,b)

d =

     1     2     3
     4     5     6

>> e = vertcat({'Sigma X','Sigma Y','Sigma Z'},num2cell(d))

e =

    'Sigma X'    'Sigma Y'    'Sigma Z'
    [      1]    [      2]    [      3]
    [      4]    [      5]    [      6]

>> f = vertcat({'Sigma X','Sigma Y','Sigma Z'},num2cell(a),num2cell(b))

f =

    'Sigma X'    'Sigma Y'    'Sigma Z'
    [      1]    [      2]    [      3]
    [      4]    [      5]    [      6]
```

مجموع درایه های بردار یا سطر و ستون ماتریس :

- sum( )

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

a =

     1     2     3
     4     5     6
     7     8     9

>> sum(a)

ans =

    12    15    18

>> sum(a,2)

ans =

     6
    15
    24
```

Trace ماتریس :

- برای جمع درایه های قطر اصلی از دستور trace استفاده می شود

- trace( )

Command Window

```
>> a = magic(3)

a =

     8     1     6
     3     5     7
     4     9     2

>> b = trace(a)

b =

    15
```

میانگین :

- mean( )

```
Command Window
>> a = 1:5

a =

     1     2     3     4     5

>> mean(a)

ans =

     3
```

MATLAB SIMULINK

ماتریس ۳ بعدی :

- a(m,n,x)

Command Window

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

a =

     1     2     3
     4     5     6

>> a(:,:,2) = [7 8 9;10 11 12]

a(:,:,1) =

     1     2     3
     4     5     6


a(:,:,2) =

     7     8     9
    10    11    12
```

ماتریس بالا مثلثی و پایین مثلثی

- triu(a) ماتریس بالا مثلثی
- tril(a) ماتریس پایین مثلثی

Command Window

```
>> a = magic(6)

a =

    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

>> triu(a)

ans =

    35     1     6    26    19    24
     0    32     7    21    23    25
     0     0     2    22    27    20
     0     0     0    17    10    15
     0     0     0     0    14    16
     0     0     0     0     0    11
```

ماتریس بالا مثلثی و پایین مثلثی

```
>> tril(a,1)

ans =

    35     1     0     0     0     0
     3    32     7     0     0     0
    31     9     2    22     0     0
     8    28    33    17    10     0
    30     5    34    12    14    16
     4    36    29    13    18    11

>> triu(a,1)

ans =

     0     1     6    26    19    24
     0     0     7    21    23    25
     0     0     0    22    27    20
     0     0     0     0    10    15
     0     0     0     0     0    16
     0     0     0     0     0     0
```

دستور find

- برای پیدا کردن سطر و ستون یک مقدار یا شرط خاص می توان از دستور find استفاده کرد.
- [row,col] = find(___)

MATLAB SIMULINK

دستور find

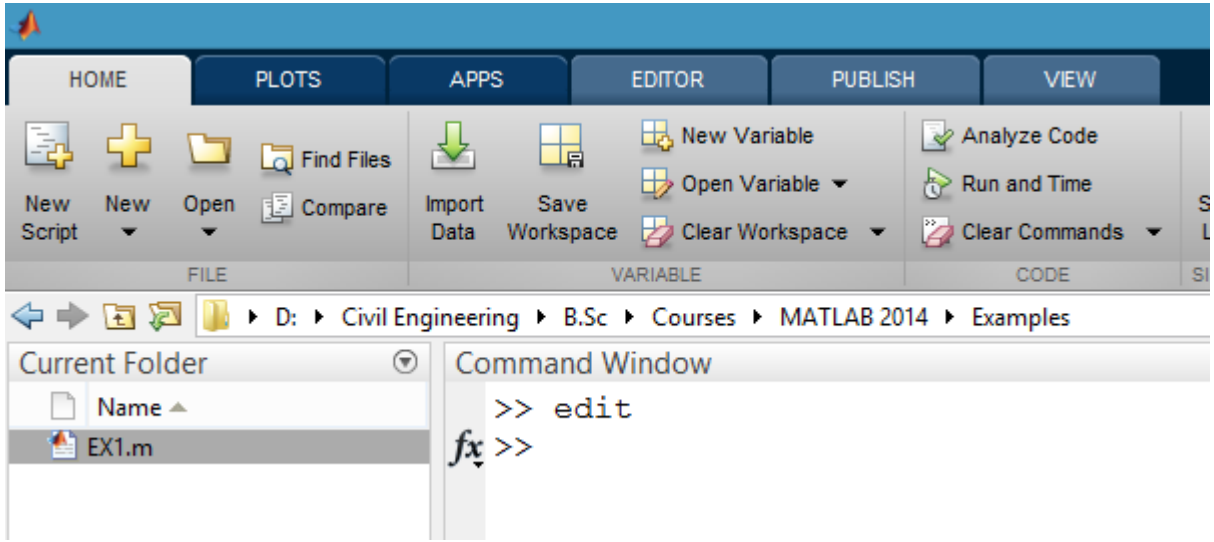
```
>> a = magic(7)

a =

    30    39    48     1    10    19    28
    38    47     7     9    18    27    29
    46     6     8    17    26    35    37
     5    14    16    25    34    36    45
    13    15    24    33    42    44     4
    21    23    32    41    43     3    12
    22    31    40    49     2    11    20

>> [m,n] = find(a>40)

m =

     3
     2
     1
     6
     7
     5
     6
     5
     4


n =

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

Scripts & Functions

نحوه ساخت یک اسکریپت :

هدف از ایجاد اسکریپت نوشتن برنامه پیچیده و نسبتا طولانی می باشد.

- گزینه new script در متلب (گزینه new و سپس گزینه script)
- دکمه های ترکیبی ctrl + n کیبورد
- تایپ عبارت edit در پنجره command window



شروع برنامه :

غالباً در ابتدای هر اسکریپت دو دستور زیر را به کار می برند :

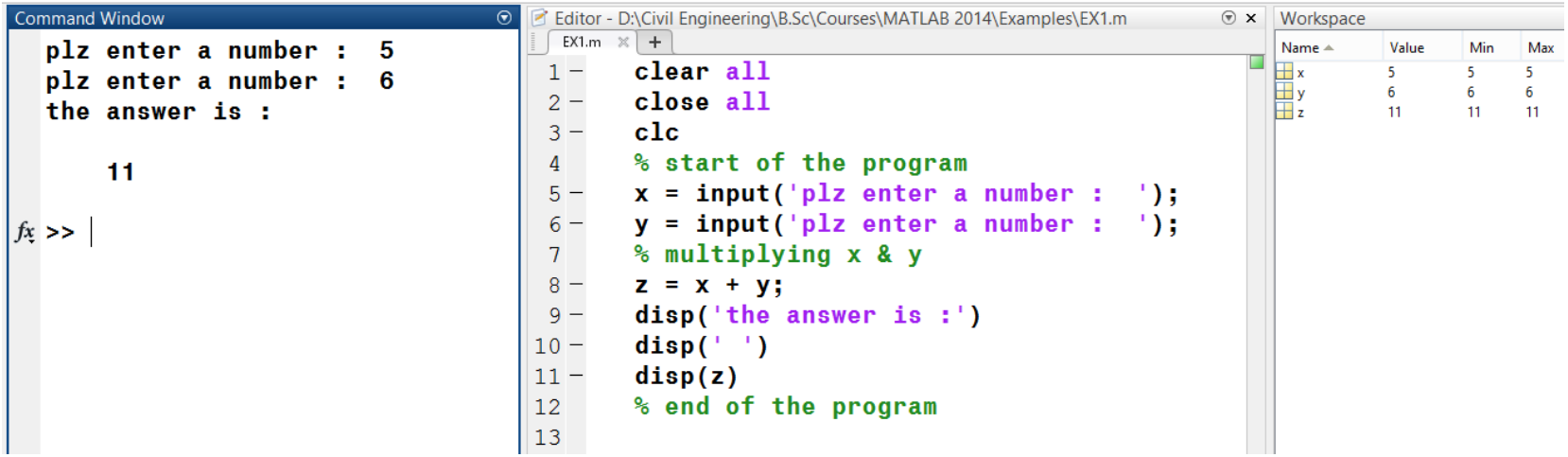
- **clear all :** برای حذف تمامی متغیرهایی که از قبل درون workspace یا به عبارتی حافظه متلب قرار گرفته است.
- **close all :** برای بستن تمامی پنجره های نموداری و شکلی استفاده می شود.
- **clc :** برای تمیز کردن (خالی کردن) محاسباتی که از قبل در پنجره command قرار گرفته است.

ارائه توضیحات در برنامه :

چنانچه از علامت درصد (٪) استفاده کنیم، تمامی نوشته های بعد از علامت درصد، به صورت توضیح در نظر گرفته می شوند.

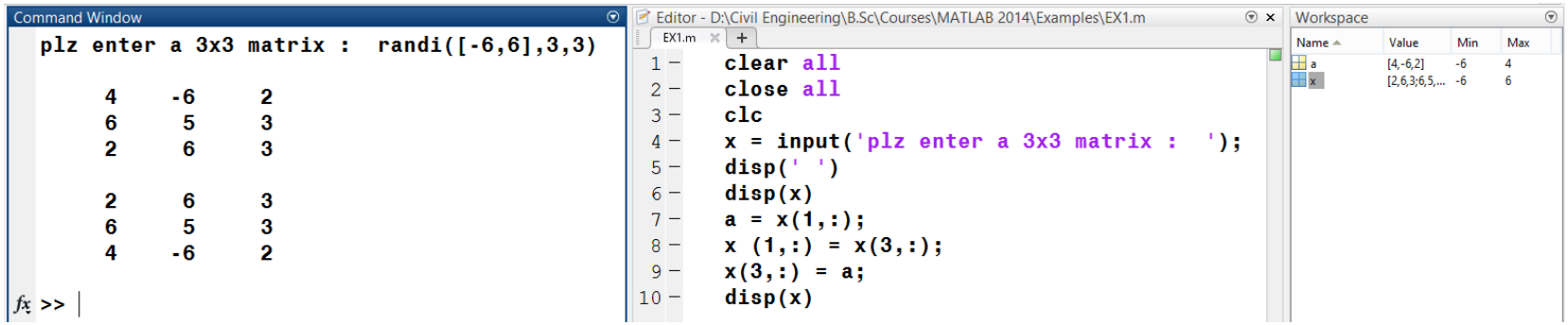
مثال ۱ :

برنامه ای بنویسید که ۲ عدد از کاربر گرفته و جمع آن ها با یکدیگر را محاسبه و نمایش دهد.



مثال ۲ :

برنامه ای بنویسید که یک ماتریس 3x3 از ورودی گرفته و سپس سطر اول با سوم آن را جا به جا کند.



دستور شرطی if...end :

دستور if برای اجرای دستورات شرطی استفاده می شود. یعنی در ابتدا شرط یا شرط هایی توسط متلب چک می شود و اگر آن شرط یا شرط ها برآورده شده باشد، آنگاه متلب دستورات مشخص شده را اجرا خواهد کرد.

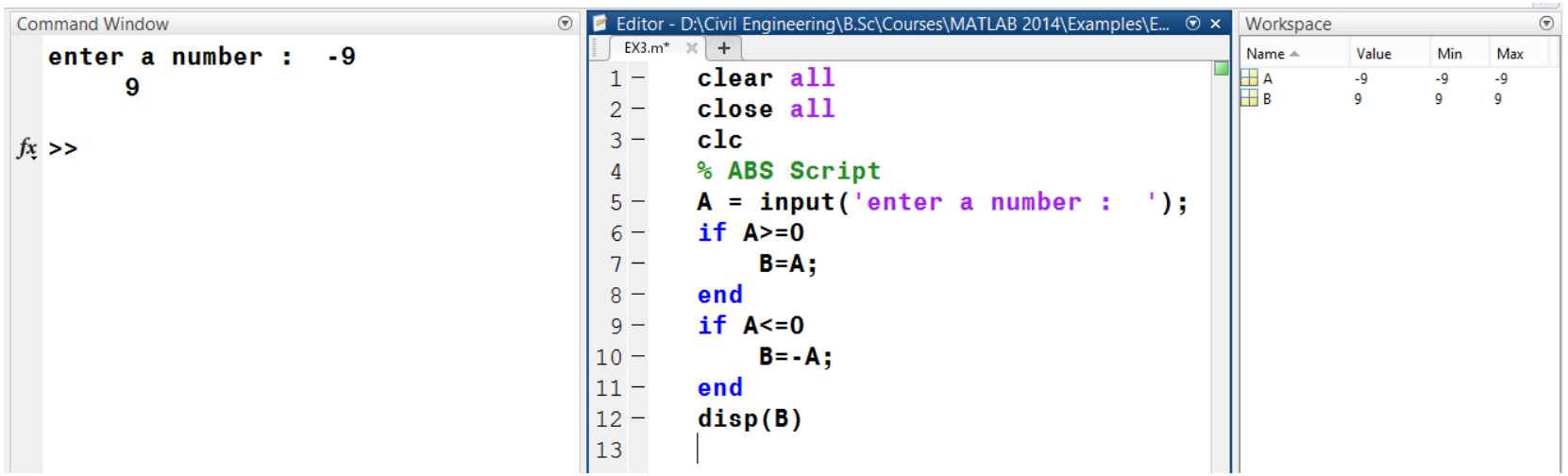
```
if <logical expression>
   <block of statements>
end
```

علائم مورد استفاده در جملات شرطی :

- علائم مورد استفاده در دستورات شرطی عبارتند از :
- < > <= >= == ~= && ||
- کوچکتر، بزرگتر، کوچکتر مساوی، بزرگتر مساوری، مساوی، نامساوی، و، یا

مثال ۳ :

برنامه قدر مطلق را با استفاده از if...end بنویسید.



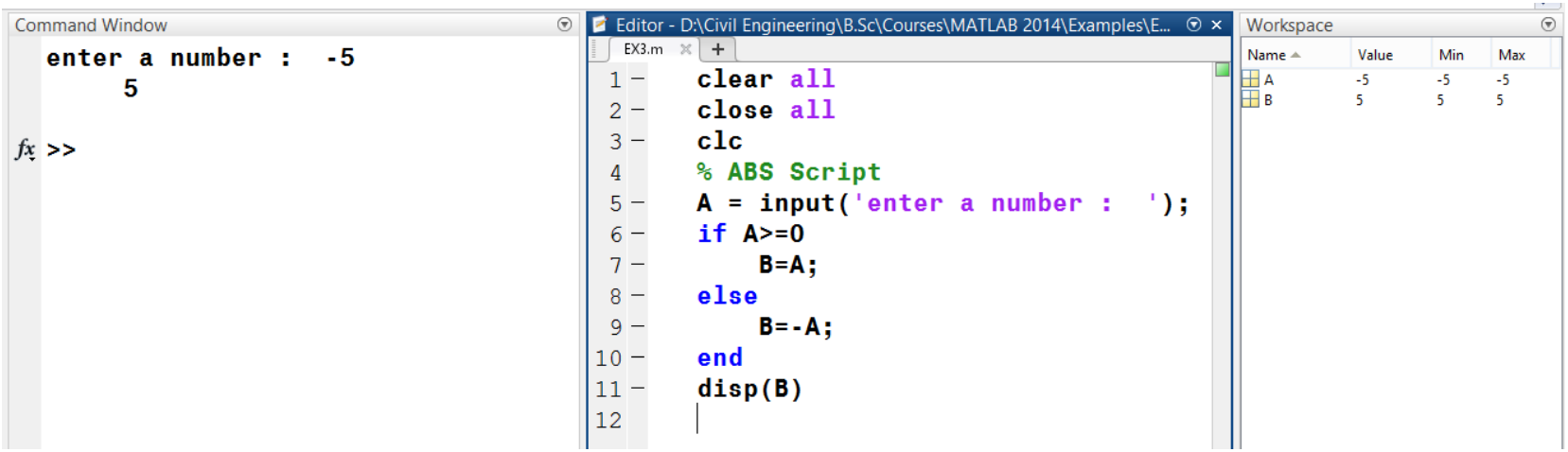
دستور شرطی if...else...end :

اگر بخواهیم به Matlab اعلام کنیم که اگر شرط یا شرط ها برآورده نشدند، آنگاه چه دستوراتی را اجرا شوند، در اینگونه موارد، دستور if را با else به کار می بریم.

```
if <logical expression>
   <block1>
else
   <block2>
end
```

مثال ٤ :

برنامه قدر مطلق را با استفاده از if...else...end بنویسید.



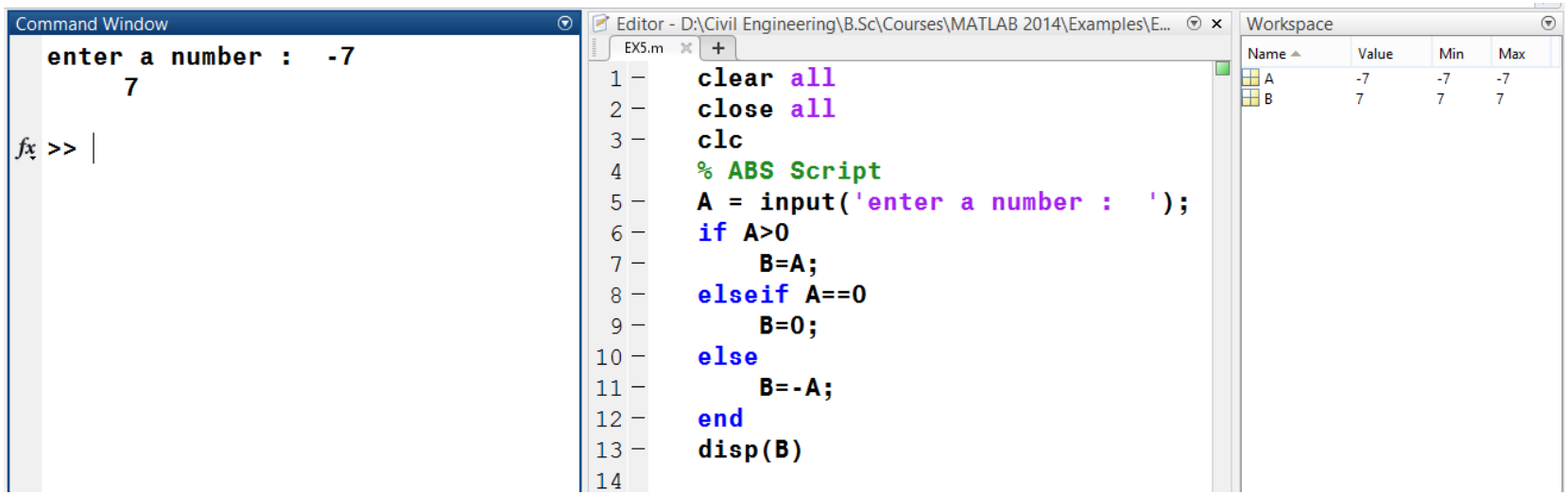
دستور شرطی if...elseif...else...end :

گاهی نیاز داریم که چندین شرط به صورت پی در پی چک شوند، در اینگونه موارد باید از دستور if به همراه elseif و else استفاده کنیم.

```
if <logical expression 1>
  <block1>
elseif <logical expression 2>
   <block2>
elseif <logical expression 3>
   <block3>
else
   <block5>
end
```

مثال ۵ :

برنامه قدر مطلق را با استفاده از if...elseif...else...end بنویسید.



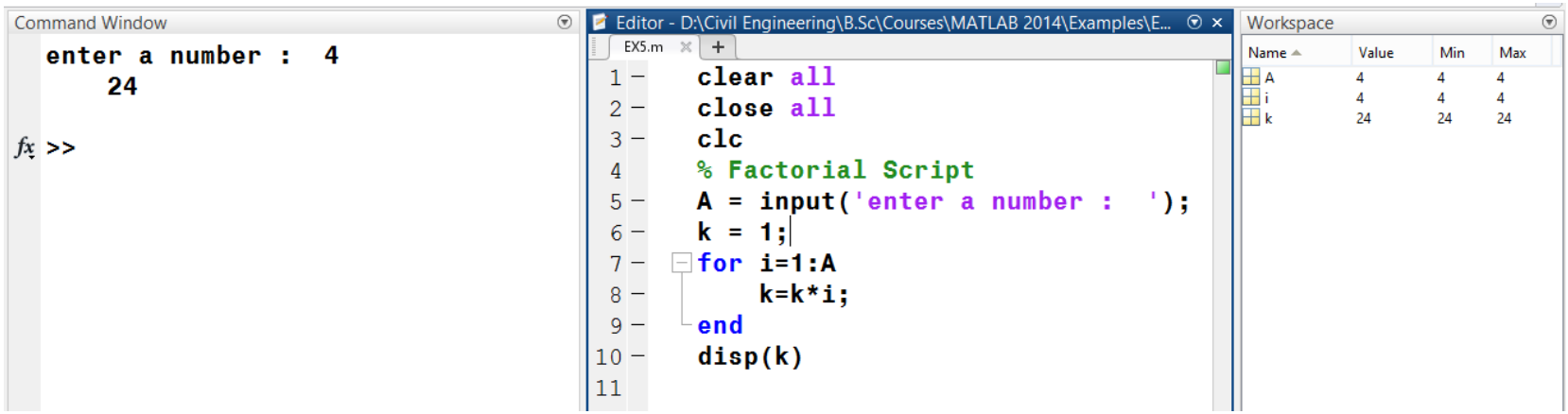
دستور for...end :

برای اجرای دستورات تکراری باید از حلقه ها استفاده کنیم، ساده ترین روش برای ساخت حلقه، استفاده از for می باشد.

```
for <index>=<initial value>:<increment>:<final value>
    <block>
end
```

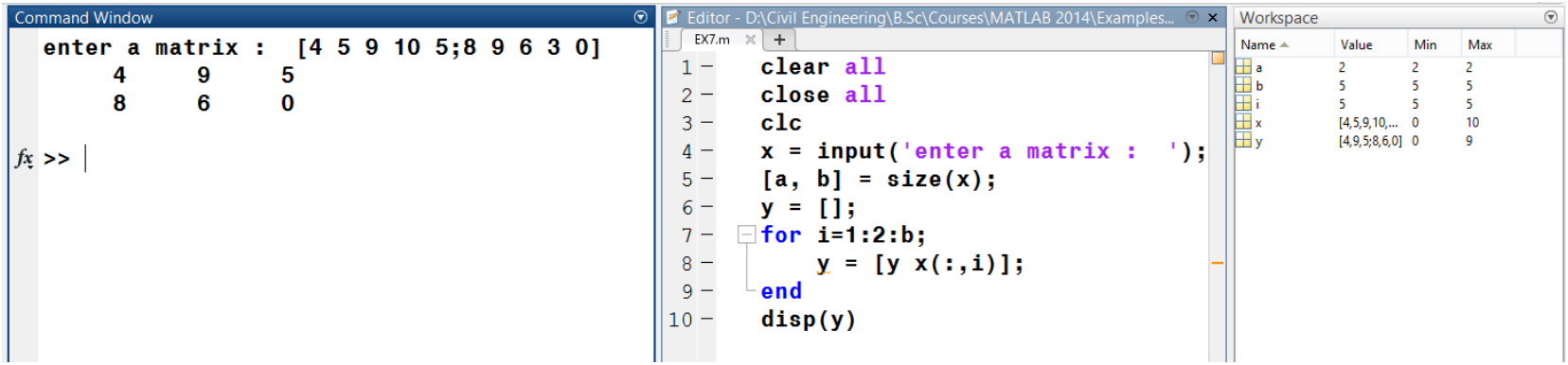
مثال ۶ :

برنامه ای بنویسید که یک عدد از ورودی گرفته و عملیات فاکتورگیری را بدون دستور factorial انجام دهد.



مثال ۷ :

برنامه ای بنویسید که یک ماتریس از ورودی بگیرد و فقط ستون های فرد را چاپ کند.



مثال ۸ :

برنامه ای بنویسید که از ۲ تاس به صورت رندوم و خودکار ۲ عدد بین ۱ تا ۶ گرفته و اعداد را چاپ کند، اگر هر ۲ تاس عدد ۱ بودند پیغام دلخواهی را چاپ کند، اگر جمع ۲ تاس ۷ بود پیغام دلخواه دیگری را چاپ کند و اگر جمع ۲ تاس ۳ یا ٤ یا ۱۰ بود پیغام دلخواه دیگری را چاپ کند، در غیر این صورت حاصل جمع دو عدد را چاپ کند.

مثال ۹ :

برنامه ای بنویسید که یک ماتریس از کاربر گرفته و آن را چاپ کند، سپس ابعاد آن را بررسی کند و اگر هر دو بعد آن از ۲ بیشتر بود سپس ۲ سطر و ۲ ستون آخر آن را حذف نموده و ماتریس نهایی را چاپ کند، در غیر این صورت پیغامی را مبنی بر کوچک بودن ابعاد ماتریس چاپ کند.

ساخت function

- معمولا برنامه نویسان حرفه ای، چارچوب برنامه را در یک فایل اصلی می نویسند و بخش های مختلف برنامه را به صورت تابع هایی می نویسند که یک یا چند ورودی را دریافت کرده و محاسبات لازم را انجام می دهند و سپس یک یا چند خروجی را بر می گردانند. برنامه نویس، عملکرد هر تابع را جداگانه چک می کند و سپس زمانی که از بابت آنها خیالش راحت شد، تمرکز اصلی خود را بر روی فایل اصلی که حاوی چارچوب برنامه می باشد، می گذارد و تنها در فایل اصلی، ارجاعاتی به توابع ساخته شده خواهد داد.
- چنانچه شما هم از این شیوه استفاده کنید، پس از مدتی صاحب بانکی از توابع خواهید شد که در نوشتن برنامه های جدید، بسیار به شما کمک می کند.

ساخت function

- output_args : خروجی تابع ساخته شده
- Untitled : عنوان تابع ساخته شده
- input_args : ورودی تابع ساخته شده

```
function [ output_args ] = Untitled( input_args )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here


end
```

مثال ۱۰ :

- Function (تابعی) بنویسید که یک ماتریس از ورودی بگیرد و دو سطر یا ستون مورد نظر آن را جا به جا کند.

Editor - D:\Civil Engineering\B.Sc\Courses\Analysis o

swap.m

```
1   function A = swap(A,Z,a,b)
2
3 -   if Z == 1
4 -   A([a b],:) = A([b a],:);
5 -   elseif Z == 2
6 -   A(:,[a b]) = A(:,[b a]);
7 -   end
```

مثال ۱۰ :

```
>> H = magic(6)

H =

    35     1     6    26    19    24
     3    32     7    21    23    25
    31     9     2    22    27    20
     8    28    33    17    10    15
    30     5    34    12    14    16
     4    36    29    13    18    11

>> F = swap(H,1,4,2)

F =

    35     1     6    26    19    24
     8    28    33    17    10    15
    31     9     2    22    27    20
     3    32     7    21    23    25
    30     5    34    12    14    16
     4    36    29    13    18    11
```

با تشکر از توجه شما