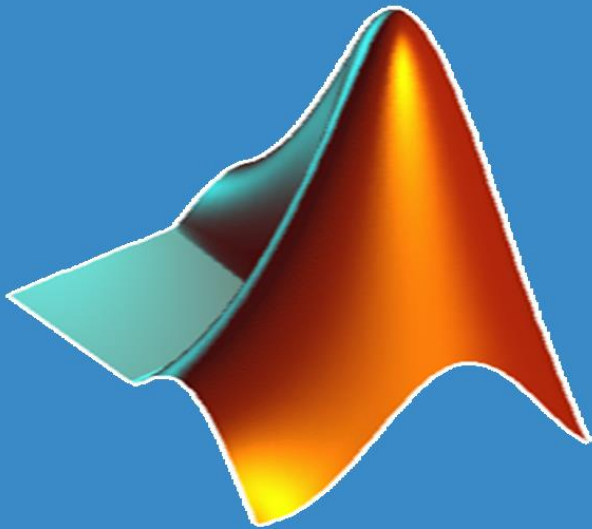


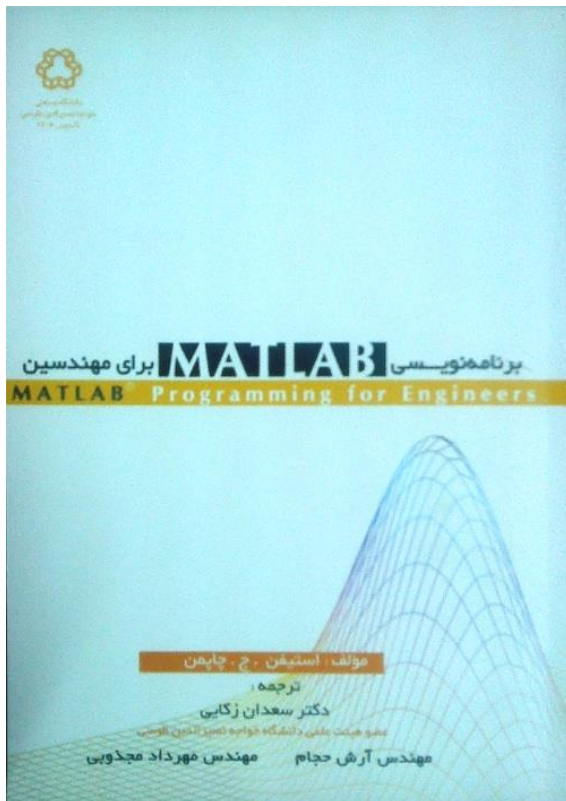
آشنایی با MATLAB

مدرس: خاطره قربانی مقدم



❖ یک منبع خوب برای یادگیری MATLAB:

- برنامه نویسی MATLAB برای مهندسين
- انتشارات دانشگاه خواجه نصیر الدین طوسی
- نویسنده استیفن چاپمن
- مترجم دکتر سعدان زکایی



❖ نرم افزار متلب را می‌توانید از لینک زیر دانلود کنید:

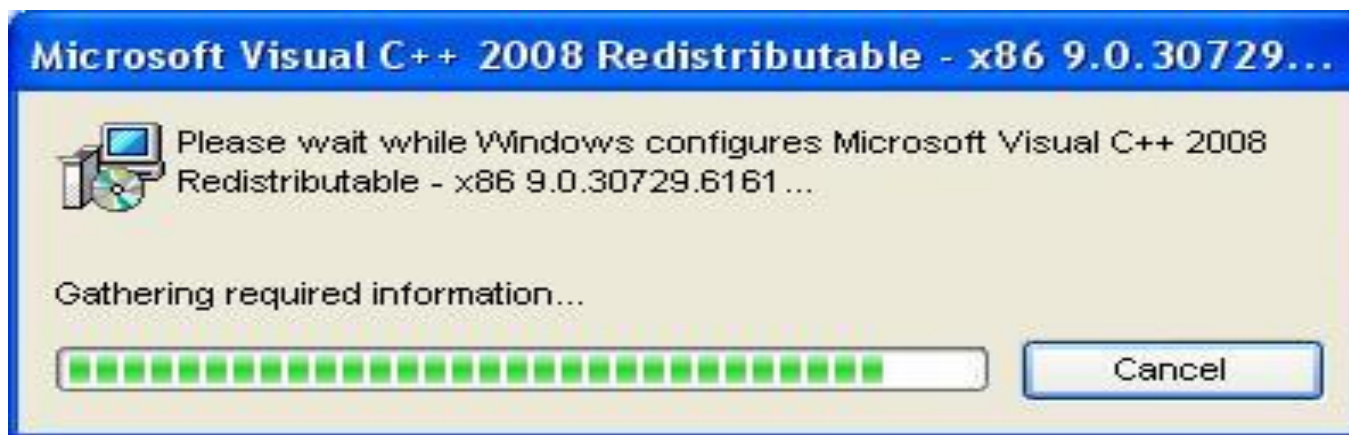
<ftp://ftp.sharif.ir/Engineering/Matlab/>

❖ MATLAB چیست ؟

- MATLAB = MATrix LABoratory
- MATLAB در ابتدا بعنوان محیطی برای کار با ماتریس‌ها بوجود آمد
- بتدریج نرم‌افزار MATLAB به یک زبان برنامه‌سازی برای انجام محاسبات علمی و فنی تبدیل شد

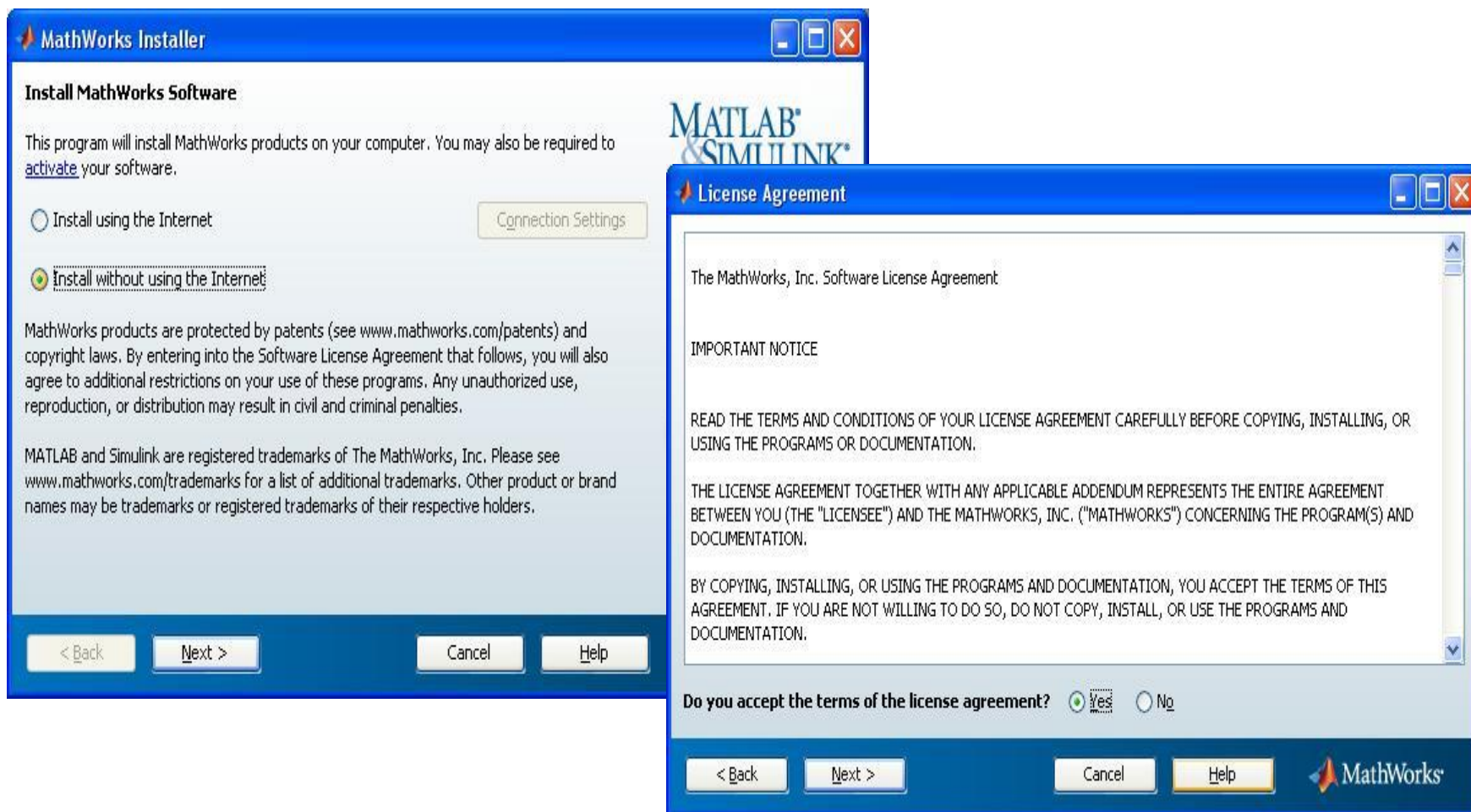
مراحل نصب نرم افزار :

بصورت تصویری نمایش داده می شود و در ابتدای اجرای فایل SETUP.exe تعدادی از کامپوننت های مورد نیاز برنامه از ویژوال C++ شروع به اکسترک کردن می شود:



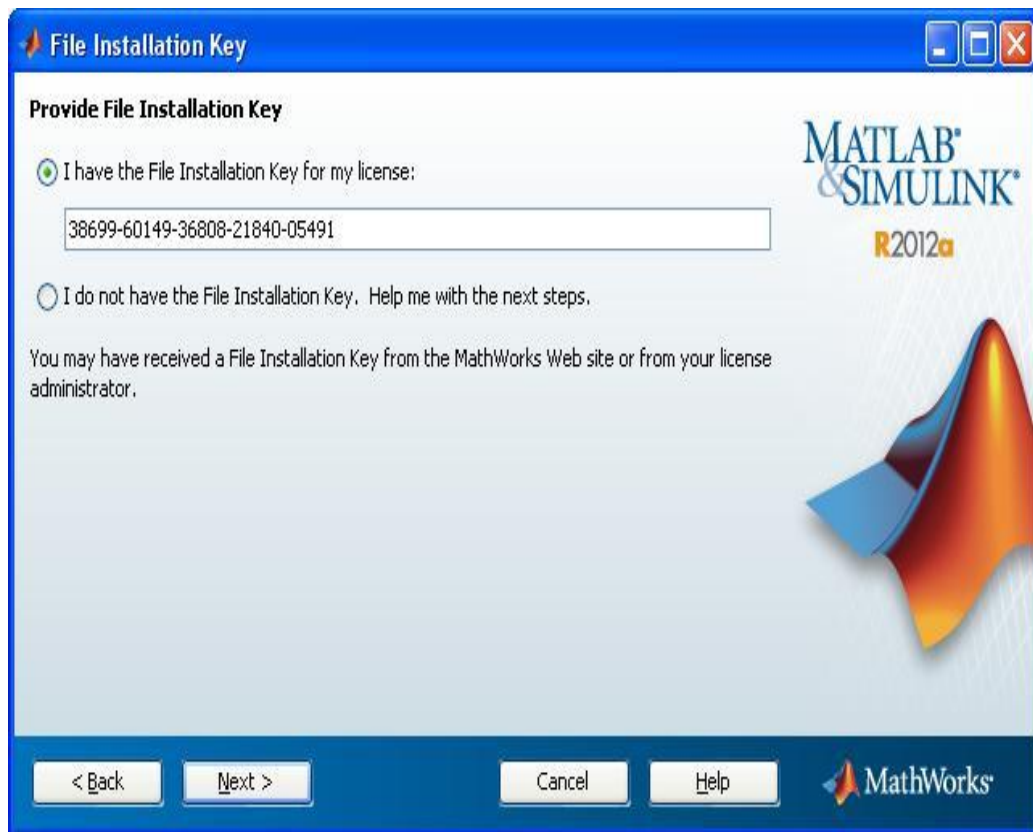
مراحل نصب نرم افزار:

گزینه **Install without using the Internet** انتخاب و در ادامه پذیرش موافقت نامه و تأیید آن:



مراحل نصب نرم افزار:

انتخاب گزینه I have the File Installation Key for my license و وارد کردن شماره سریال برنامه این نکته حائز اهمیت است که گاهی اوقات بعد از نصب Error هایی دارد که بهتر است همان ابتدا شماره سریال را بصورت دستی وارد کنید:



مراحل نصب نرم افزار:

در این قسمت اگر Custom انتخاب شود می توانید یکسری از محصولات برنامه را نصب نکنید و بهتر است همان Typical انتخاب تا کلیه کامپوننت های برنامه نصب شود:



مراحل نصب نرم افزار:

آدرس محل نصب برنامه که اگر قبلاً نصب نشده باشد سؤال می پرسد آیا دایرکتوری ایجاد کند؟ آنرا تأیید کنید:



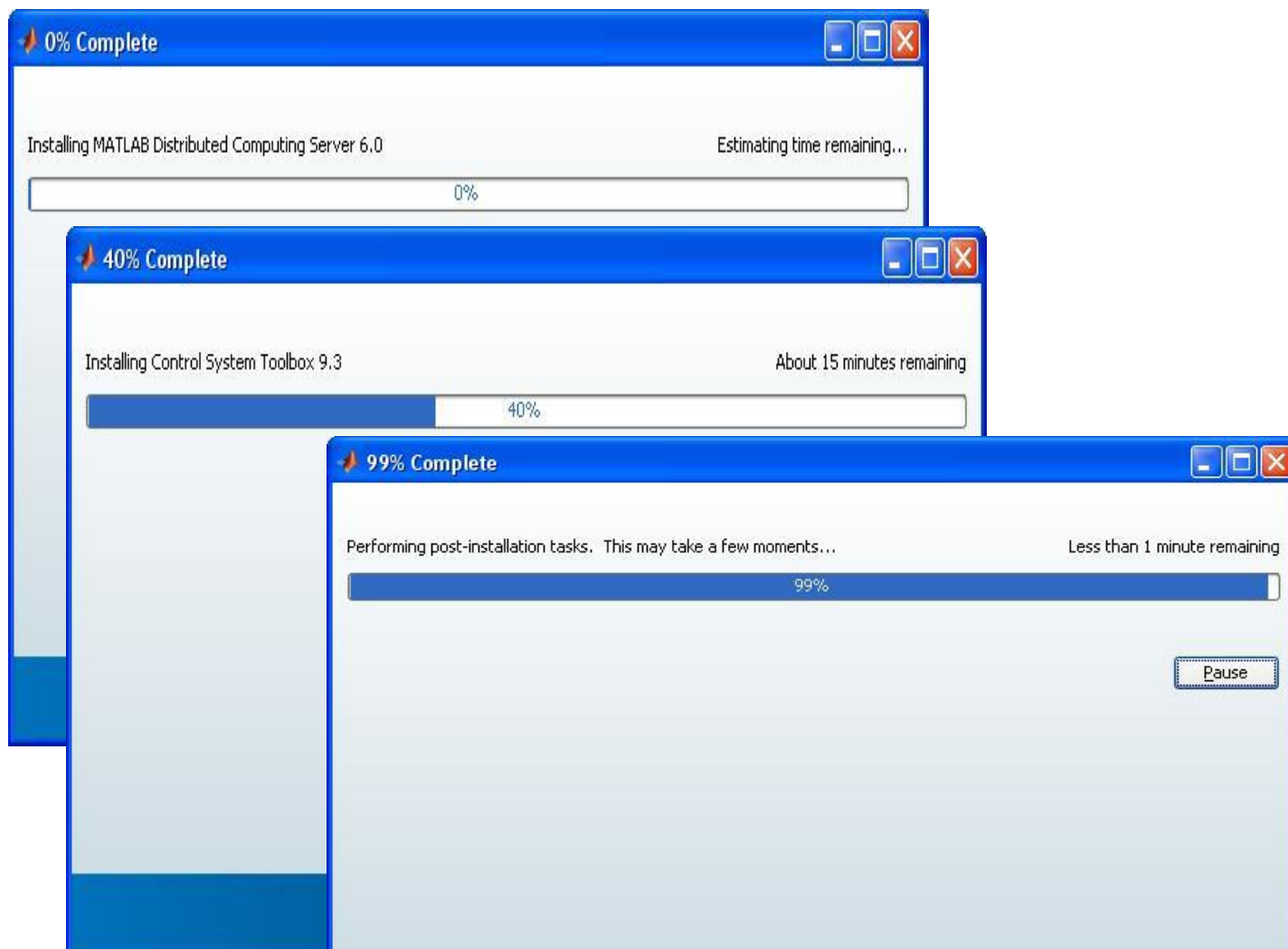
مراحل نصب نرم افزار:

توضیحاتی درخصوص تنظیمات انجام شده ، لیست برنامه های انتخاب شده ، مسیر برنامه و کلیه برنامه هایی که قرار است نصب شوند:



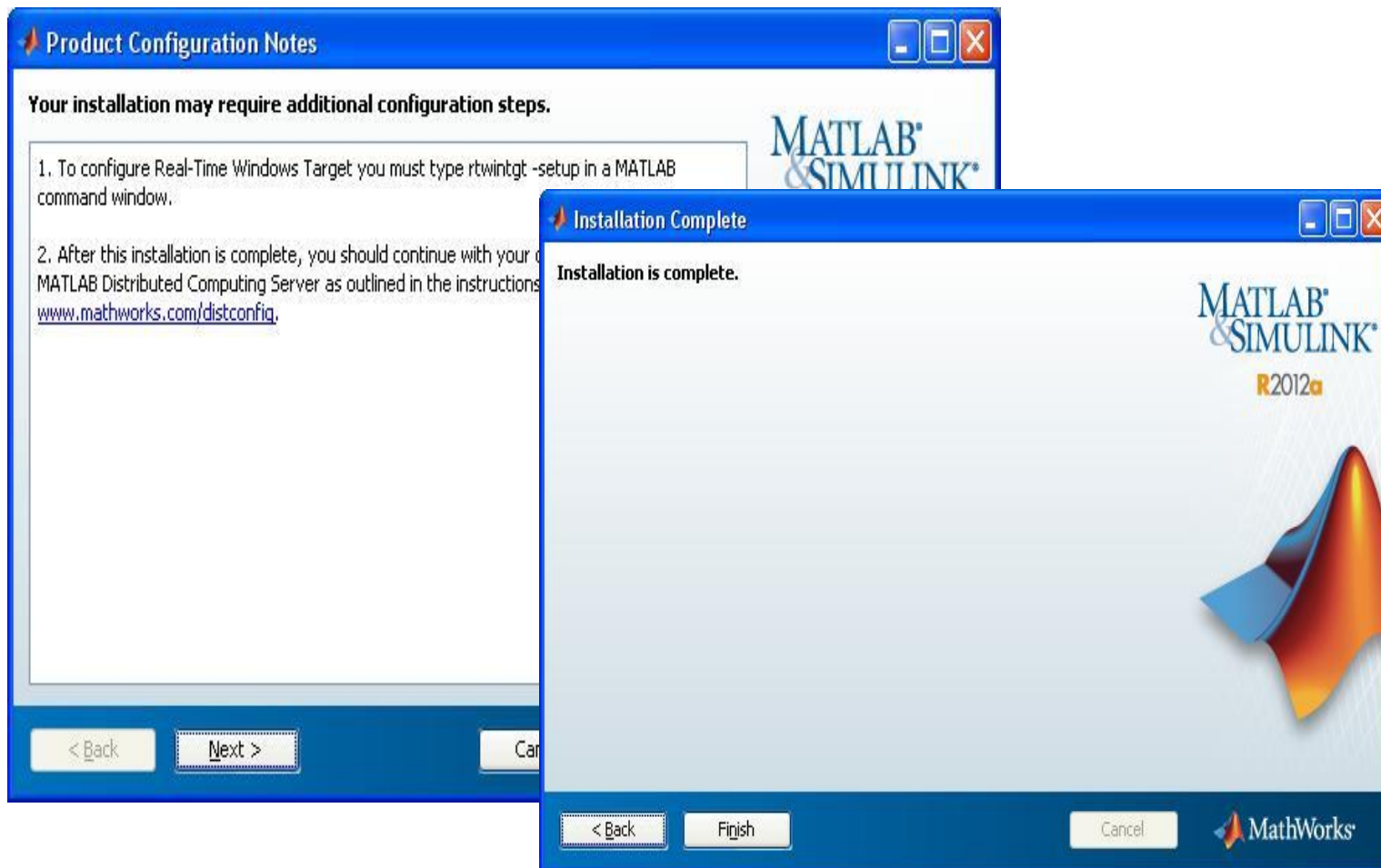
مراحل نصب نرم افزار:

شروع نصب برنامه و پایان آن و زمانی حدود 20 تا 30 دقیقه (بسته به قدرت سیستم):



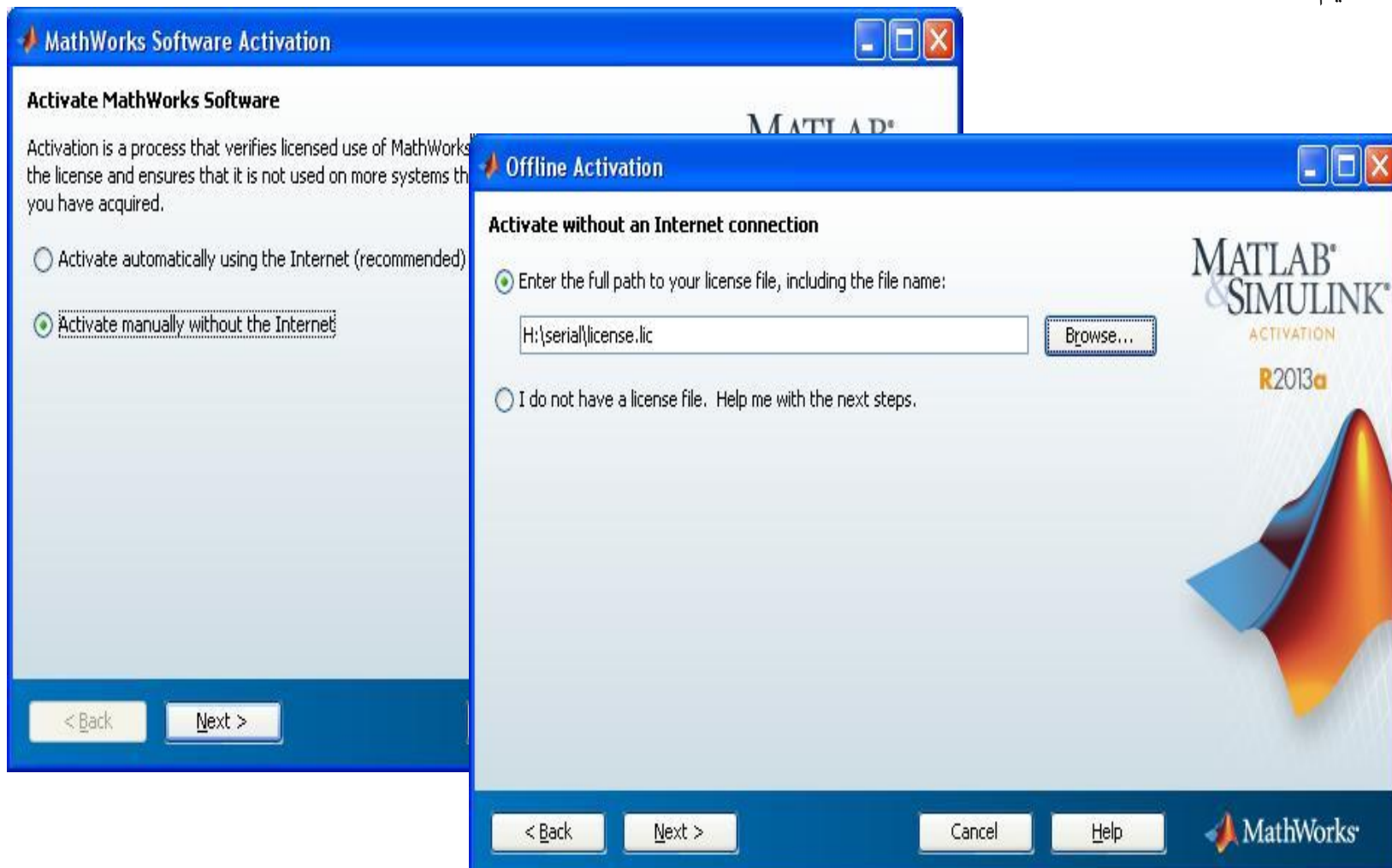
مراحل نصب نرم افزار :

پایان نصب برنامه :



مراحل نصب نرم افزار:

پس از نصب برنامه نوبت به رجیستر کردن آن است که **Active manually without the Internet** انتخاب کنیم



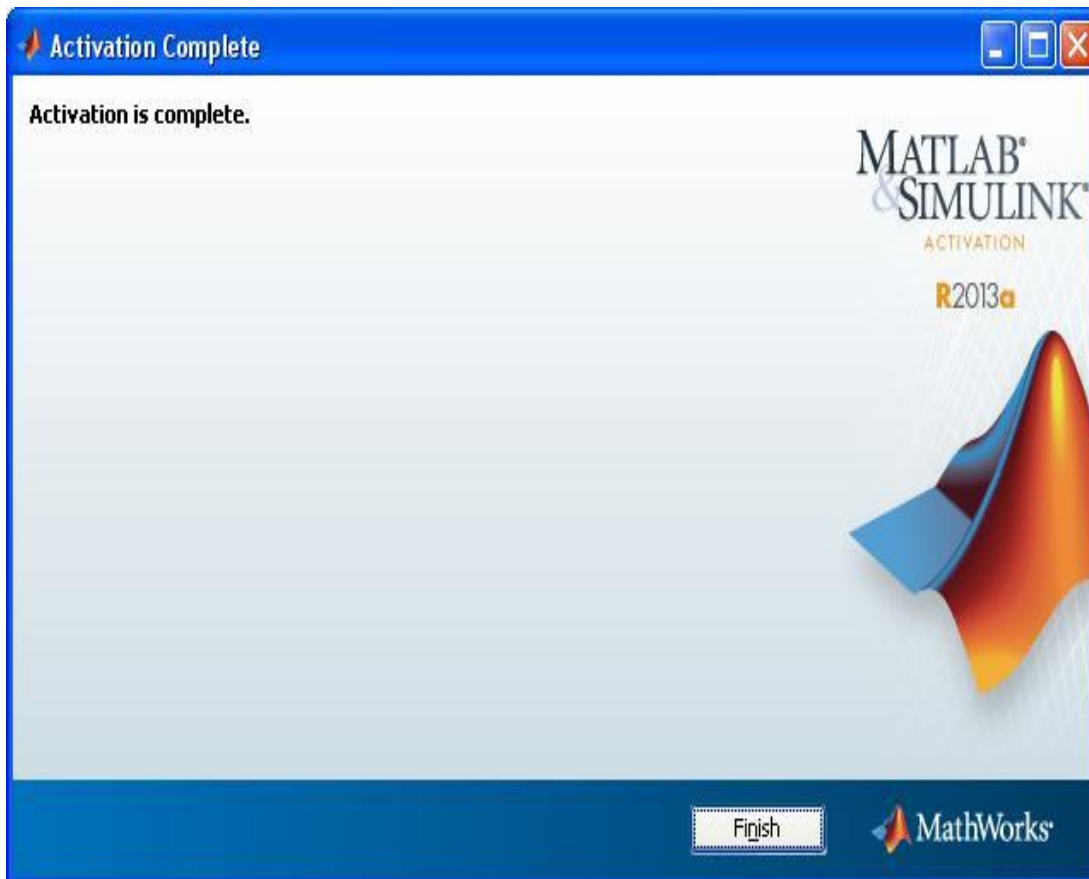
مراحل نصب نرم افزار:

در قسمت Enter the full path to your license file, including the file name باید آدرس فایل License.lic را که در فولدر سریال است را می دهیم:



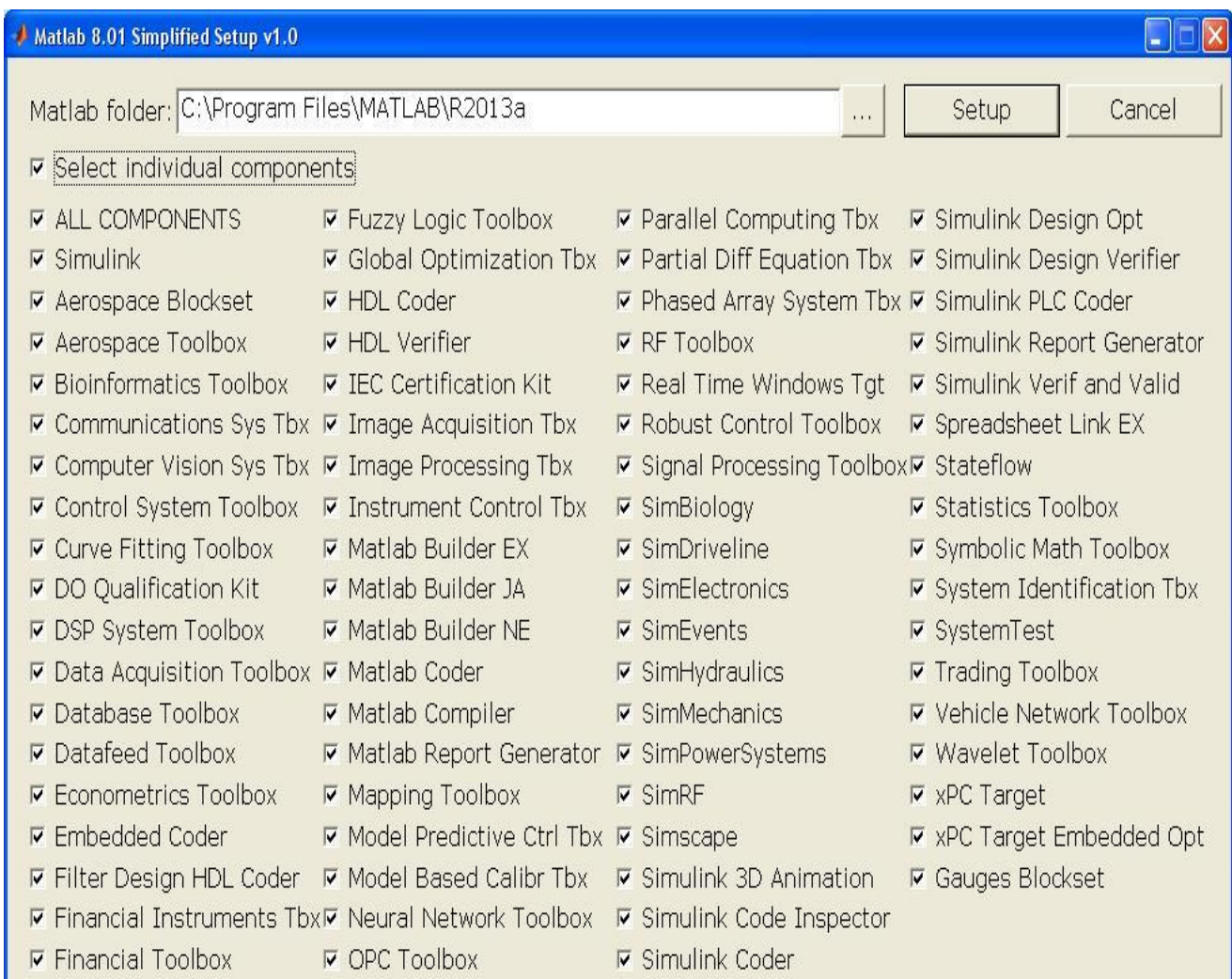
مراحل نصب نرم افزار:

نصب برنامه به اتمام رسید و می توانید آنرا اجرا کنید.



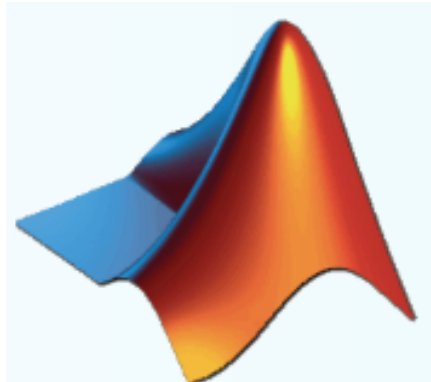
مراحل نصب نرم افزار :

در صورتی که کامپوننتی را در قسمت custom نصب نکرده باشید می توانید فایل SetupSimple.exe را اجرا کنید که پنجره زیر ظاهر می شود و می توانید کتابخانه هایی را که نیاز دارید تیک زده و آنها را نصب کنید.

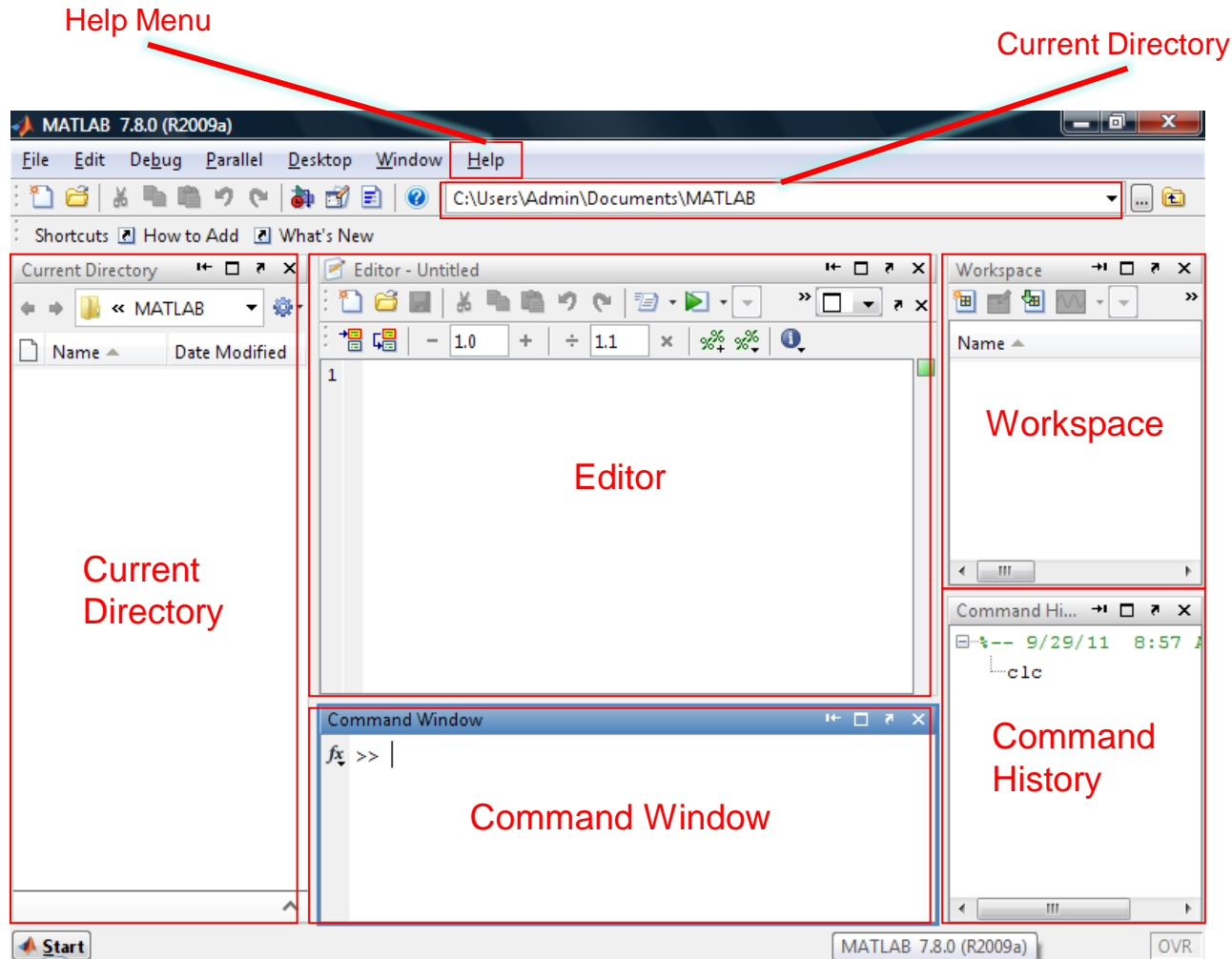


❖ اجرا کردن MATLAB

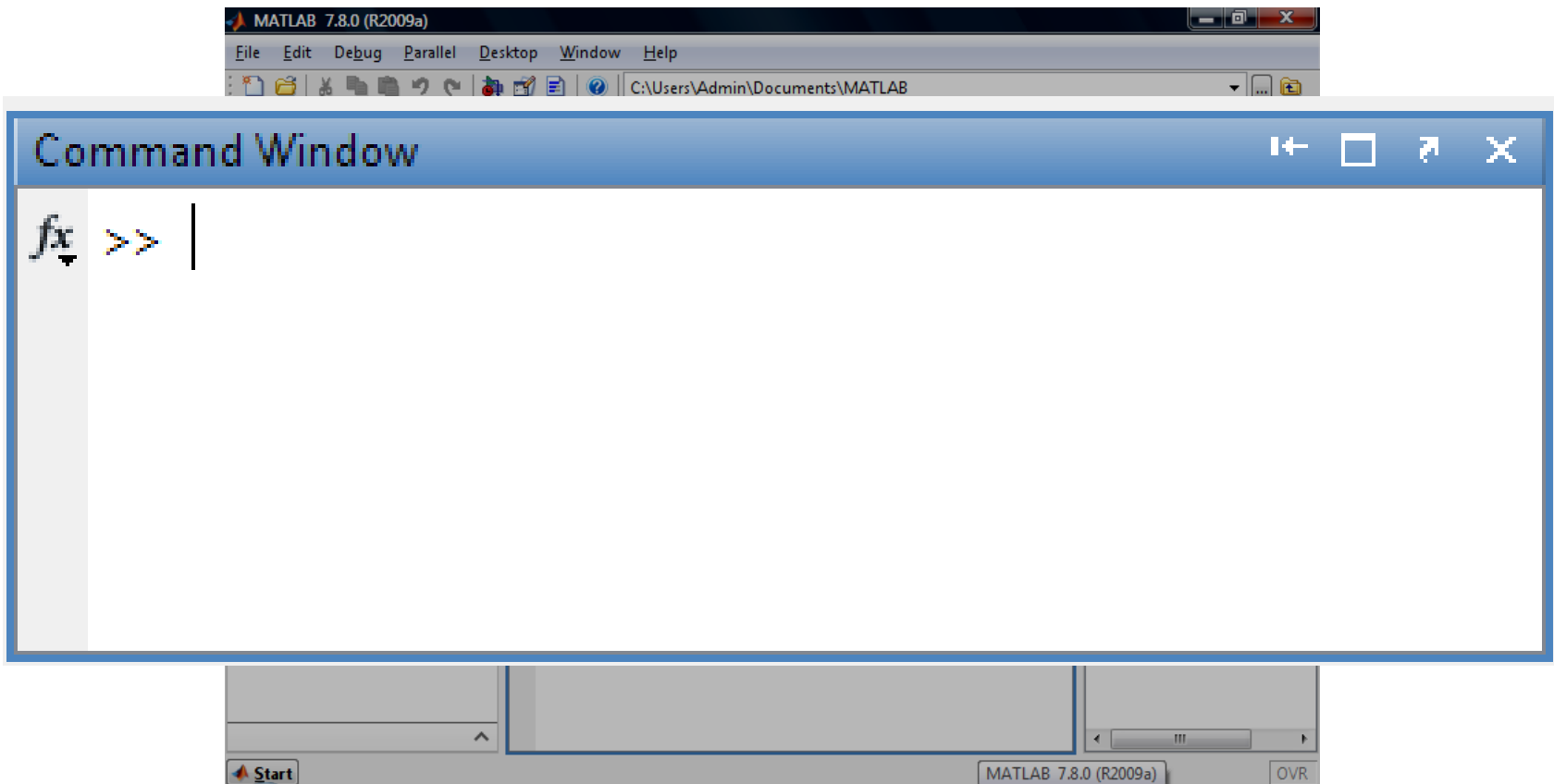
▪ بر روی آیکن زیر در دسکتاپ کامپیوتر خود کلیک کنید



محیط MATLAB

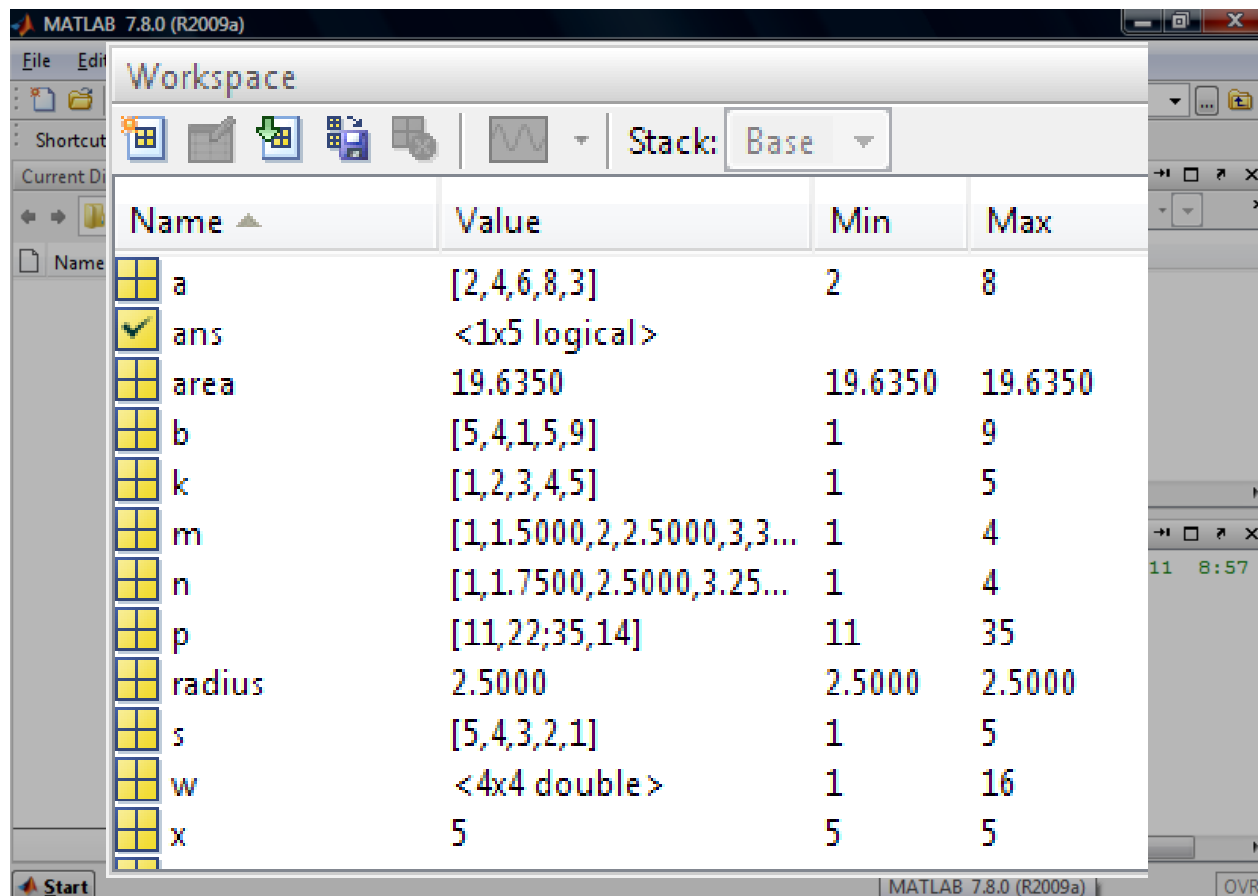


- ❖ پنجره فرمان Command Window
- برای اجرای فوری فرمان‌ها به کار می‌رود



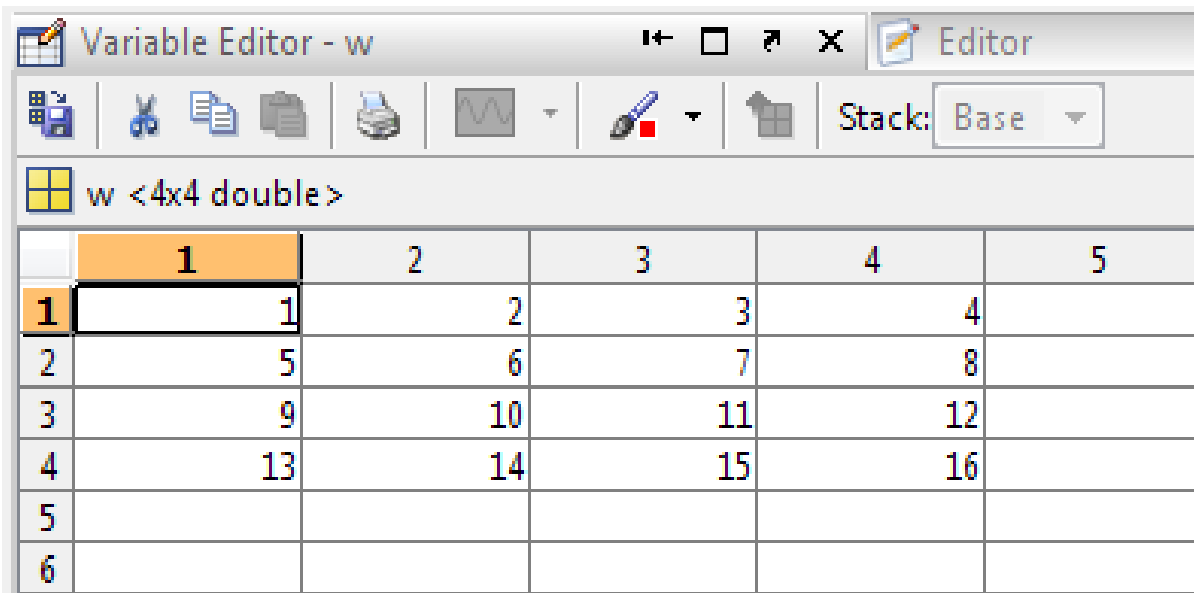
❖ پنجره Workspace

- متغیرهای موجود را نمایش میدهد



❖ پنجره Workspace

- اگر بر روی یک متغیر در پنجره workspace دو بار کلیک کنید، محتوای آن متغیر در پنجره variable editor به نمایش در می آید

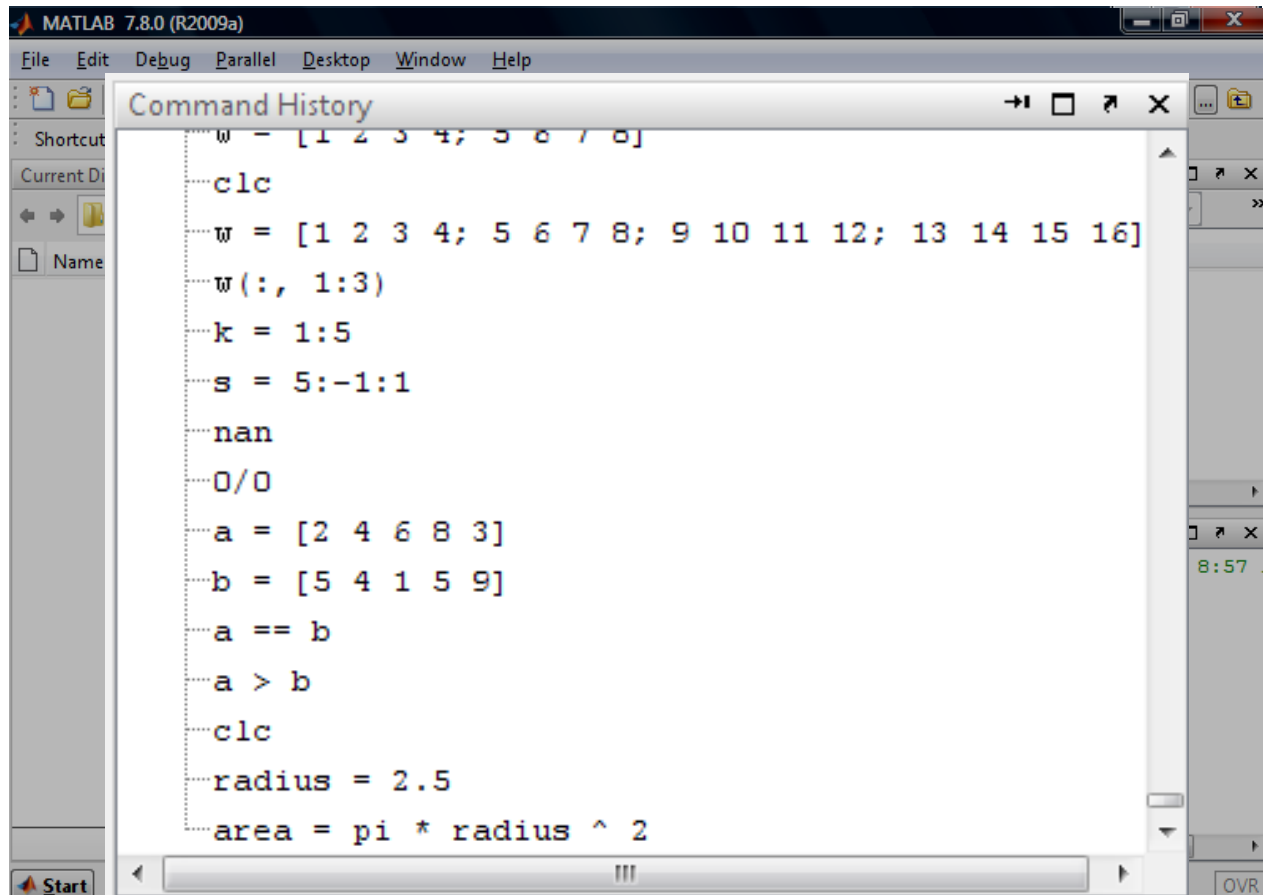


The screenshot shows the MATLAB Variable Editor window titled "Variable Editor - w". The window has a toolbar with icons for various actions like copy, paste, and zoom. Below the toolbar, the variable "w" is identified as a "4x4 double" matrix. The matrix is displayed in a table format with columns numbered 1 to 5 and rows numbered 1 to 6. The first four rows contain the data of the 4x4 matrix, and the last two rows are empty.

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

❖ پنجره تاریخچه دستورات Command History

■ لیست دستورات قبلی اجرا شده را نگهداری میکند



The screenshot shows the MATLAB 7.8.0 (R2009a) interface. The 'Command History' window is open, displaying a list of commands that have been executed in the current session. The commands are as follows:

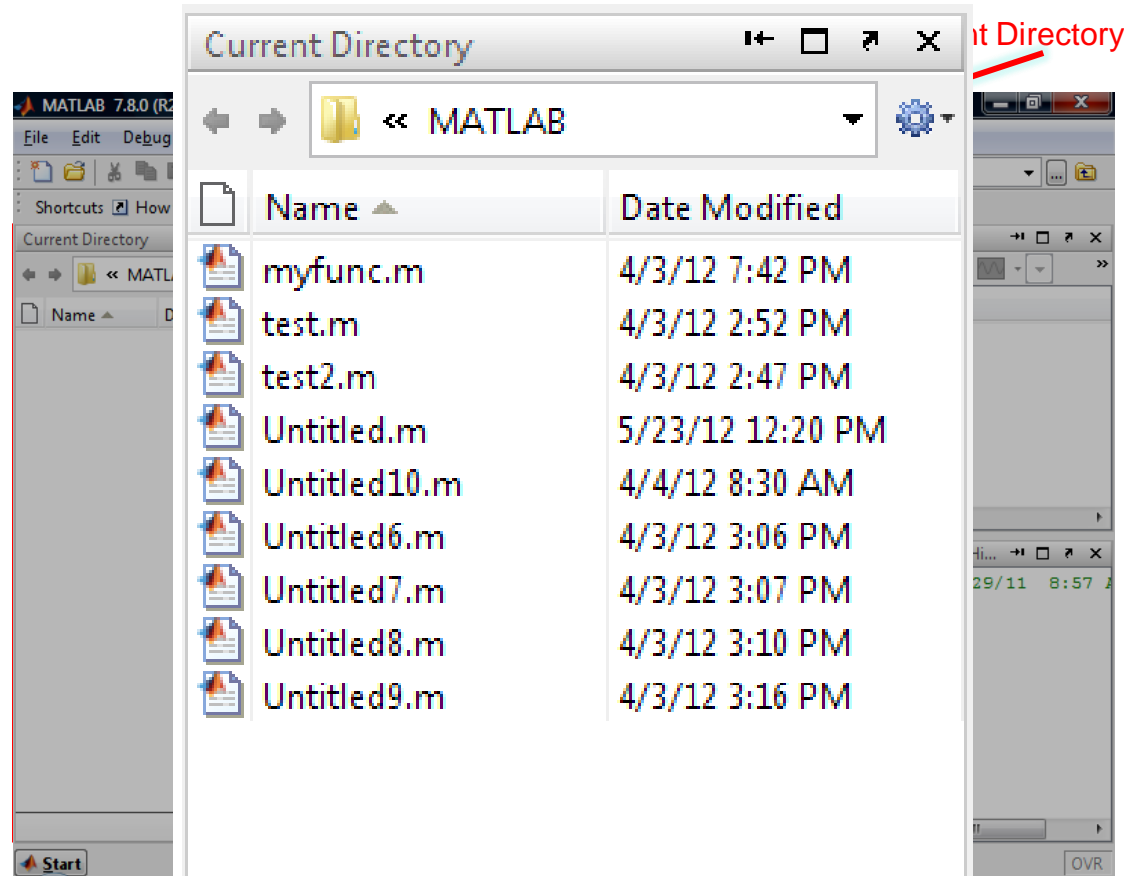
```

w = [1 2 3 4; 5 6 7 8]
clc
w = [1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16]
w(:, 1:3)
k = 1:5
s = 5:-1:1
nan
0/0
a = [2 4 6 8 3]
b = [5 4 1 5 9]
a == b
a > b
clc
radius = 2.5
area = pi * radius ^ 2
    
```

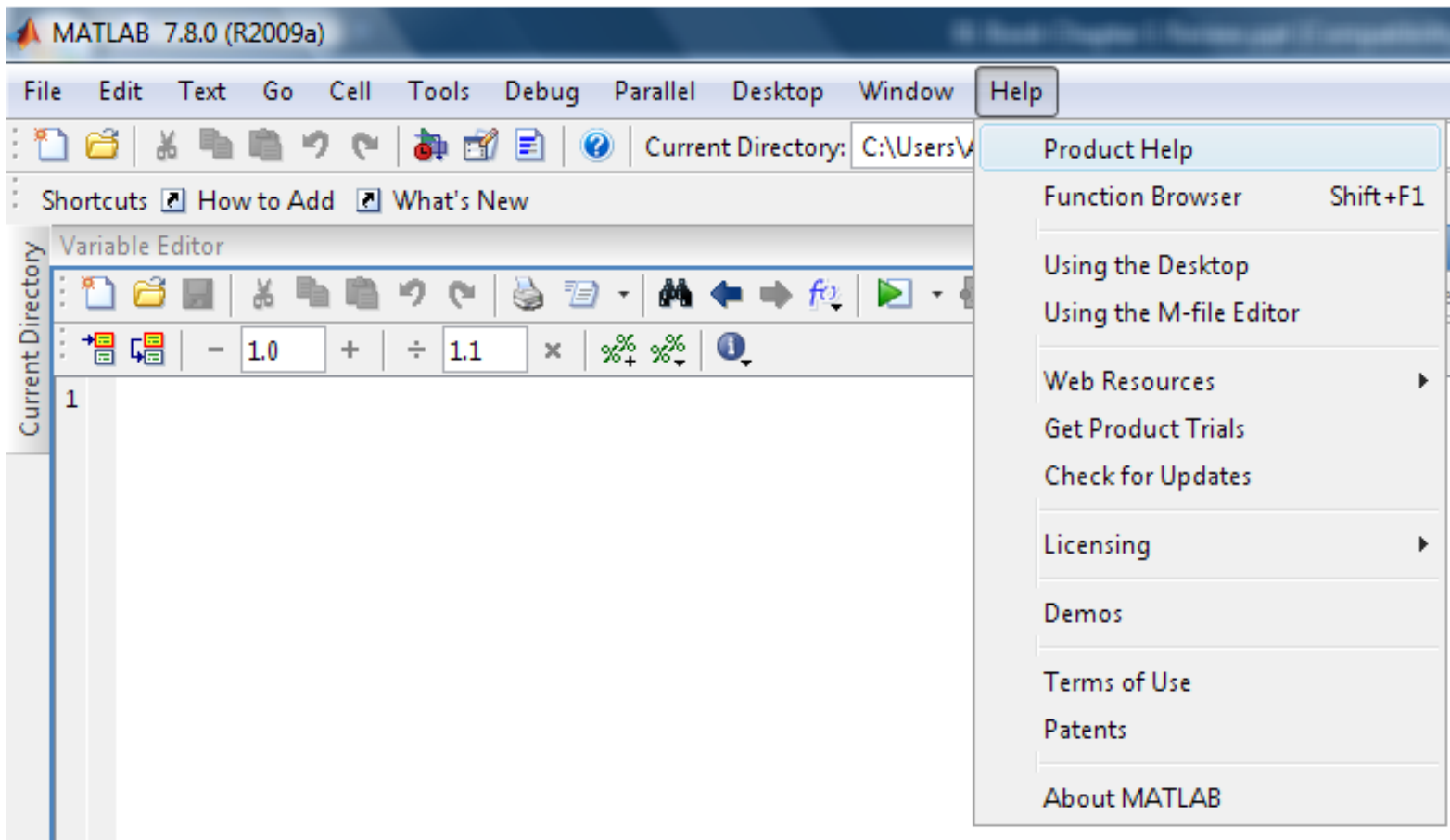
The window includes standard window controls (minimize, maximize, close) and navigation buttons (back, forward, search, etc.). The MATLAB Start button is visible at the bottom left of the interface.

❖ پنجره مسیر جاری Current Directory

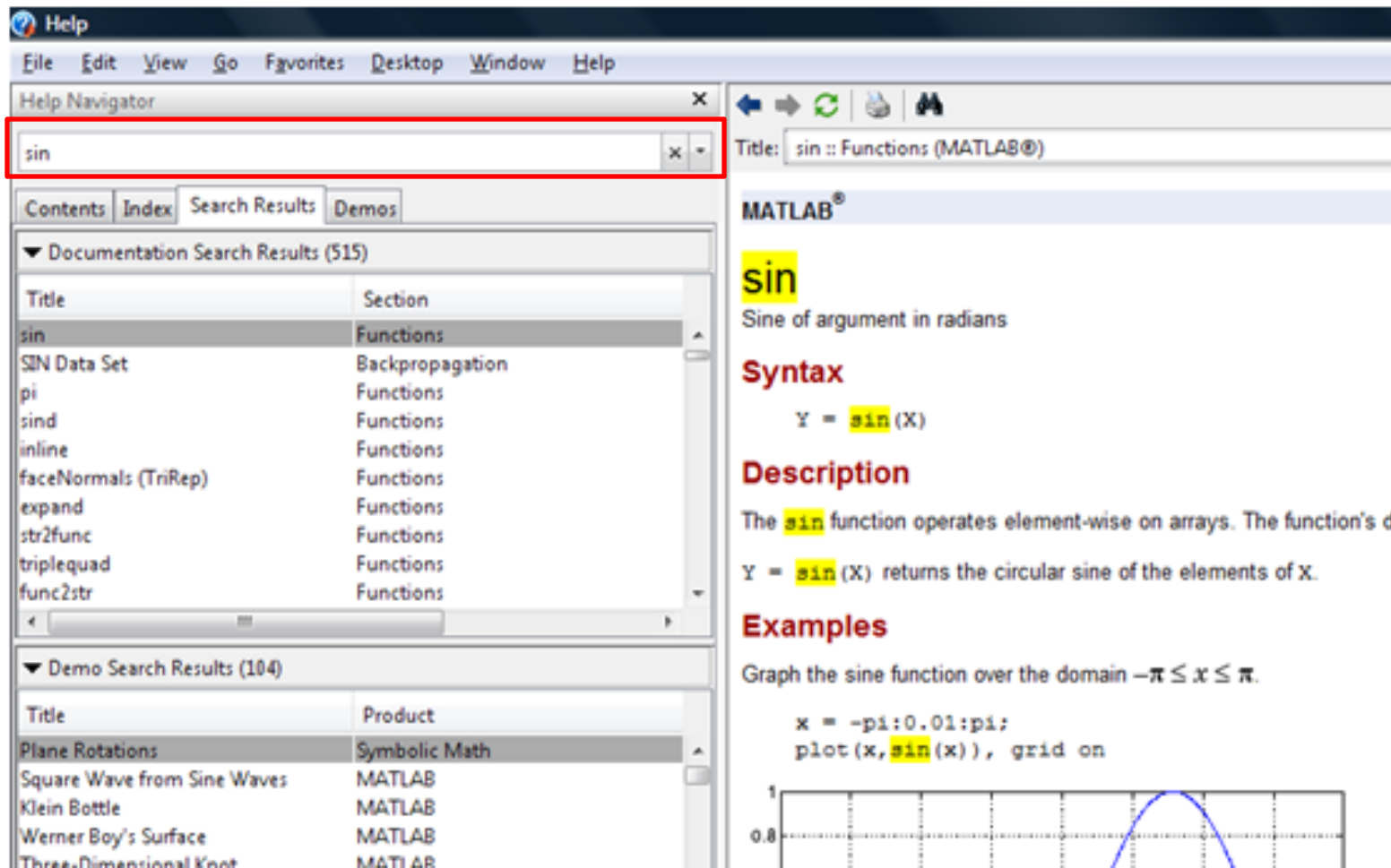
■ مسیر جاری MATLAB را تنظیم و نمایش میدهد



❖ پنجره کمک



❖ پنجره کمک



The screenshot shows the MATLAB Help interface. On the left, the 'Help Navigator' window has a search bar containing 'sin', which is highlighted with a red rectangle. Below the search bar, the 'Search Results' tab is active, showing a list of documentation search results. The first result, 'sin', is selected and highlighted. The main window displays the documentation for the 'sin' function. It includes the title 'sin', a description 'Sine of argument in radians', the syntax 'Y = sin(X)', a description of the function's operation, and examples of its use, including a plot of the sine function over the domain $-\pi \leq x \leq \pi$.

Help Navigator

sin

Contents Index Search Results Demos

▼ Documentation Search Results (515)

Title	Section
sin	Functions
SIN Data Set	Backpropagation
pi	Functions
sind	Functions
inline	Functions
faceNormals (TriRep)	Functions
expand	Functions
str2func	Functions
triplequad	Functions
func2str	Functions

▼ Demo Search Results (104)

Title	Product
Plane Rotations	Symbolic Math
Square Wave from Sine Waves	MATLAB
Klein Bottle	MATLAB
Werner Boy's Surface	MATLAB
Three-Dimensional Knot	MATLAB

MATLAB®

sin

Sine of argument in radians

Syntax

`Y = sin(X)`

Description

The **sin** function operates element-wise on arrays. The function's default data type is `double`.

`Y = sin(X)` returns the circular sine of the elements of `X`.

Examples

Graph the sine function over the domain $-\pi \leq x \leq \pi$.

```
x = -pi:0.01:pi;
plot(x, sin(x)), grid on
```

The plot shows the sine function over the domain $-\pi \leq x \leq \pi$. The x-axis ranges from -3.14 to 3.14, and the y-axis ranges from -1 to 1. The plot is a blue line representing the sine wave.

❖ پنجره فرمان Command Window

■ برای اجرای فوری فرمان‌ها به کار می‌رود

```
>> 5 + 10
```

```
ans =
```

```
15
```

❖ متغیرها

- متغیر محلی در حافظه کامپیوتر است که برای ذخیره و بازیابی موقت مقادیر به کار میرود
- هر متغیر یک نام دارد
- هر متغیر یک مقدار دارد
- مقدار متغیر میتواند در طول زمان تغییر داده شود
- مقدار متغیر در حافظه کامپیوتر ذخیره میشود

❖ متغیرهای رایج در MATLAB

■ اسکالر

- یک مقدار عددی را نگهداری میکند

■ آرایه

- گروهی از مقادیر را نگهداری میکند که به هر یک از مقادیر میتوان با کمک اندیس دسترسی پیدا کرد

■ بردار

- آرایه یک بعدی
- هر یک از عناصر، توسط یک اندیس مورد دسترسی قرار میگیرد

■ ماتریس

- آرایه دو بعدی
- هر یک از عناصر، توسط دو اندیس مورد دسترسی قرار میگیرد

❖ تعریف متغیر

- برای تعریف یک متغیر جدید، یک مقدار اولیه را به یک نام جدید اختصاص دهید

مقدار = نام متغیر

- تعریف متغیر اسکالر

```
>> x = 5
```

```
x =
```

```
5
```

❖ تعریف متغیر

■ تعریف بردار

- مجموعه مقادیر بردار باید بین براکت باز و بسته `[]` قرار داده شوند
- مقادیر به کمک فاصله یا کاما `,` باید از هم جدا شوند

```
>> y = [10 12 35 14 11]
```

```
y =
```

```
10    12    35    14    11
```

❖ تعریف متغیر

■ تعریف بردار

- برای ایجاد بردارهای طولانی میتوان از عملگر کولن : استفاده کرد
مقدار پایان : گام : مقدار شروع
- اگر مقدار گام ذکر نشود، بطور پیش فرض مقدار گام یک در نظر

```
>> m = 1:0.5:4
```

```
m =
```

```
1.0000    1.5000    2.0000    2.5000    3.0000    3.5000    4.0000
```

```
>> k = 1:5
```

```
k =
```

```
1    2    3    4    5
```

```
>> s = 5:-1:1
```

```
s =
```

```
5    4    3    2    1
```

❖ تعریف متغیر

■ تعریف بردار

- برای ایجاد بردارهای طولانی میتوان از دستور linspace استفاده کرد

(تعداد , مقدار پایان , مقدار شروع) linspace

```
>> n = linspace(1,4,5)
```

```
n =
```

```
1.0000    1.7500    2.5000    3.2500    4.0000
```


❖ تعریف متغیر

■ تعریف ماتریس

- برای شروع یک سطر جدید از سمبل سمی کالن ; استفاده میشود

```
>> z = [10 12; 35 14; 11 22]
```

```
z =
```

```
10    12
35    14
11    22
```

❖ نامگذاری متغیرها

- زبان MATLAB به اندازه حروف حساس است یعنی نامهای radius، Radius، rAdius سه نام متفاوت هستند
- نام متغیر باید با یک حرف شروع شود
- نام متغیر میتواند شامل حروف، اعداد، و زیر خط باشد
- نام متغیر باید معنی دار باشد

my_5_variable یک نام صحیح

5my_variable یک نام غلط

❖ دسترسی به محتوای متغیرها

- برای دسترسی به محتوای یک متغیر اسکالر کافی است در یک دستور یا عبارت محاسباتی از نام آن متغیر استفاده کنید

```
>> x
```

```
x =
```

```
5
```

```
>> 2*x
```

```
ans =
```

```
10
```

❖ دسترسی به محتوای متغیرها

- برای دسترسی به یک عنصر خاص از یک بردار، باید اندیس آن عنصر را بین پرانتز باز و بسته () پس از نام متغیر ذکر کرد

```
>> y
```

```
y =
```

```
10    12    35    14    11
```

```
>> y(3)
```

```
ans =
```

```
35
```

❖ دسترسی به محتوای متغیرها

- برای دسترسی به یک عنصر خاص از یک ماتریس، باید اندیس سطر و ستون آن عنصر را بین پرانتز باز و بسته () پس از نام متغیر ذکر کرد

```
>> z
```

```
z =
```

```
10    12  
35    14  
11    22
```

```
>> z(3,1)
```

```
ans =
```

```
11
```

❖ دسترسی به زیر آرایه ها

- برای دسترسی به تمام عناصر یک سطر از علامت کولن :
بجای شماره سطر استفاده کنید.
- برای دسترسی به تمام عناصر یک ستون از علامت کولن :
بجای شماره ستون استفاده کنید.

```
>> z
```

```
z =
```

```
10    12
35    14
11    22
```

```
>> z(:,1)
```

```
ans =
```

```
10
35
11
```

```
>> z(2,:)
```

```
ans =
```

```
35    14
```

❖ دسترسی به زیر آرایه ها

- برای دسترسی به چند سطر یا ستون خاص، از یک بردار بعنوان اندیس سطر یا ستون استفاده کنید. این بردار باید حاوی شماره سطرها یا ستونهاى مورد نظر باشد.

```
>> z
```

```
z =
```

```
10    12
35    14
11    22
```

```
>> p = z([3 2], :)
```

```
p =
```

```
11    22
35    14
```

❖ دسترسی به زیر آرایه ها

- برای دسترسی به کلیه عناصر آرایه، بعنوان یک بردار ستونی از دستور زیر استفاده کنید.

```
>> z
```

```
z =
```

```
10    12
35    14
11    22
```

```
>> z(:)
```

```
ans =
```

```
10
35
11
12
14
22
```


❖ دسترسی به زیر آرایه ها

- برای دسترسی به چند سطر یا ستون خاص، از یک بردار بعنوان اندیس سطر یا ستون استفاده کنید. این بردار باید حاوی شماره سطرها یا ستونهاى مورد نظر باشد.

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

w =

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

```
>> w(:, 1:3)
```

ans =

1	2	3
5	6	7
9	10	11
13	14	15

❖ عملگرهای ریاضی

■ عملگرهای ریاضی برای کار با متغیرهای اسکالر

ARITHMETIC OPERATIONS WITH SCALARS

<u>Operation</u>	<u>Symbol</u>	<u>Example</u>
Addition	+	5+3
Subtraction	-	5-3
Multiplication	*	5*3
Right Division	/	5/3
Left Division	\	5\3=3/5
Exponentiation	^	5^3



```
>> 5 + 10
```

```
ans =
```

```
15
```

```
>> 10 ^ 2
```

```
ans =
```

```
100
```

```
>> x = 5
```

```
x =
```

```
5
```

```
>> 5*x^2 + 3*x + 4
```

```
ans =
```

```
144
```

❖ عملگرهای ریاضی

- عملگرهای ریاضی برای کار با ماتریس ها
- بر اساس قوانین جبر خطی

عملگر	عملیات	توضیحات
$a + b$	جمع	هر عنصر ماتریس a را با عنصر متناظرش از ماتریس b جمع میکند
$a - b$	تفریق	هر عنصر ماتریس b را از عنصر متناظرش در ماتریس a کم میکند
$a * b$	ضرب	حاصلضرب ماتریس a در ماتریس b را محاسبه میکند
a / b	تقسیم راست	$a * \text{inv}(b)$
$a \setminus b$	تقسیم چپ	$\text{inv}(a) * b$

❖ عملگرهای ریاضی

■ عملگرهای ریاضی برای کار با ماتریس ها

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

```
a =
```

```
1    5
2    4
```

```
>> b = [-2 1; 3 2]
```

```
b =
```

```
-2    1
3     2
```

```
>> a + b
```

```
ans =
```

```
-1    6
5     6
```

```
>> a - b
```

```
ans =
```

```
3     4
-1    2
```

```
>> a * b
```

```
ans =
```

```
13    11
8     10
```

```
>> a / b
```

```
ans =
```

```
1.8571    1.5714
1.1429    1.4286
```

❖ عملگرهای آرایه ای

- عملگرهای آرایه ای عنصر به عنصر انجام میشوند
- بر روی عناصر متناظر از دو آرایه هم اندازه انجام میگیرند

عملگر	عملیات	توضیحات
$a + b$	جمع	هر عنصر از آرایه a را با عنصر متناظرش از آرایه b جمع میکند
$a - b$	تفریق	هر عنصر از آرایه b را از عنصر متناظرش از آرایه a تفریق میکند
$a .* b$	ضرب	هر عنصر از آرایه a را در عنصر متناظرش از آرایه b ضرب میکند a یا b میتوانند اسکالر نیز باشند
$a ./ b$	تقسیم راست	هر عنصر از آرایه a را بر عنصر متناظرش از آرایه b تقسیم میکند a یا b میتوانند اسکالر نیز باشند
$a .\ b$	تقسیم چپ	هر عنصر از آرایه b را بر عنصر متناظرش از آرایه a تقسیم میکند a یا b میتوانند اسکالر نیز باشند
$a .^ b$	توان	هر عنصر از آرایه a را به توان عنصر متناظرش از آرایه b میرساند a یا b میتوانند اسکالر نیز باشند



```
>> a = [1 5 3 2]
```

```
a =
```

```
1    5    3    2
```

```
>> b = [2 1 0 4]
```

```
b =
```

```
2    1    0    4
```

```
>> a + b
```

```
ans =
```

```
3    6    3    6
```

```
>> a - b
```

```
ans =
```

```
-1    4    3   -2
```

```
>> a .* b
```

```
ans =
```

```
2    5    0    8
```

```
>> a ./ b
```

```
ans =
```

```
0.5000    5.0000         Inf    0.5000
```

```
>> a .\ b
```

```
ans =
```

```
2.0000    0.2000         0    2.0000
```

❖ مقادیر منطقی

- در منطق دو مقداری یک گزاره صحیح یا غلط است
- نوع داده منطقی logical در MATLAB برای پشتیبانی از منطق دو مقداری در نظر گرفته شده است
- مقادیر منطقی
 - صحیح با اعداد غیر صفر نشان داده میشود
 - غلط با عدد صفر نمایش داده میشود

❖ عملگرهای رابطه‌ای

- این عملگرها دو عملوند را دریافت کرده و صحت روابط مختلف را بین آنها بررسی میکنند
- حاصل عملگرهای رابطه‌ای، مقادیر منطقی صحیح یا غلط است.

عملگر	شرح
==	تساوی
~=	عدم تساوی
>	بزرگتر از
>=	بزرگتر یا مساوی
<	کوچکتر از
<=	کوچکتر یا مساوی

❖ عملگرهای رابطه ای

```
>> a = [2 4 6 8 3]
```

```
a =
```

```
2     4     6     8     3
```

```
>> b = [5 4 1 5 9]
```

```
b =
```

```
5     4     1     5     9
```

```
>> a == b
```

```
ans =
```

```
0     1     0     0     0
```

```
>> a > b
```

```
ans =
```

```
0     0     1     1     0
```

❖ عملگرهای منطقی

■ عملگرهای منطقی بر روی مقادیر منطقی (صحیح/غلط) عمل میکنند

عملگر	شرح
&	AND منطقی
	OR منطقی
~	NOT منطقی

❖ عملگرهای منطقی

■ جدول صحت عملگرهای منطقی

Inputs		and	or	not
A	B	A & B	A B	$\sim A$
false	false	false	false	true
false	true	false	true	true
true	false	false	true	false
true	true	true	true	false

❖ عملگرهای منطقی

```
>> a = 5
```

```
>> b = 2
```

```
>> c = 3
```

```
>> a>b
```

```
ans =
```

```
1
```

```
>> (a>b) && (a>c)
```

```
ans =
```

```
1
```

```
>> (a>b) || (a<c)
```

```
ans =
```

```
1
```

```
>> a==c
```

```
ans =
```

```
0
```

```
>> ~(a==c)
```

```
ans =
```

```
1
```

❖ عملگر ترانهاده

■ عملگر $'$ ترانهاده یک ماتریس را محاسبه میکند.

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

a =

1	2
3	4
5	6

```
>> a'
```

ans =

1	3	5
2	4	6

❖ جلوگیری از چاپ حاصل

- برای جلوگیری از چاپ حاصل یک عبارت، کافی است در انتهای دستور از سمبل سمی کالن ; استفاده کنید

```
>> b = a*2
```

```
b =
```

```

     2     4
     6     8
    10    12
```

```
>> b = a*2;
```

```
>>
```

❖ اولویت عملگرها

1. (,)
2. ^, -
3. ~
4. * /
5. + -
6. > , >=, <, <=, ==, ~=
7. &
8. |

❖ اولویت عملگرها

```
>> a = 5
```

```
>> b = 2
```

```
>> c = 3
```

```
>> a - b >= c
```

```
ans =
```

```
1
```

```
>> (a - b) >= c
```

```
ans =
```

```
1
```

```
>> a - (b >= c)
```

```
ans =
```

```
5
```

❖ یک برنامه ساده

```
>> radius = 2.5
```

```
radius =
```

```
2.5000
```

```
>> area = pi * radius ^ 2
```

```
area =
```

```
19.6350
```

❖ متغیرهای از پیش تعریف شده

نام متغیر	محتوا
pi	مقدار عدد π
i	مقدار عدد $\sqrt{-1}$
j	مقدار عدد $\sqrt{-1}$
inf	بینهایت
eps	اپسیلون
nan	Not a Number حاصل تقسیم صفر بر صفر
ans	متغیر پیش فرض برای پاسخ محاسبات

❖ متغیرهای از پیش تعریف شده

```
>> 2/0
```

```
ans =
```

```
Inf
```

```
>> 0/0
```

```
ans =
```

```
NaN
```

```
>> i
```

```
ans =
```

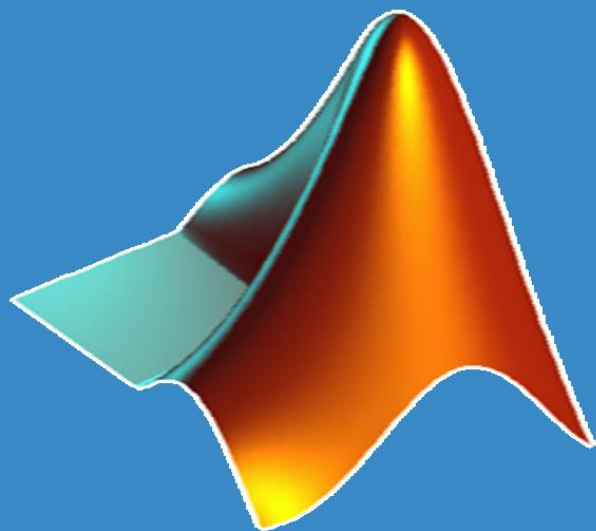
```
0 + 1.0000i
```

```
>> 2+5i
```

```
ans =
```

```
2.0000 + 5.0000i
```

دستورات ابتدایی و ماتریس های خاص



clear

کلید متغیرهای موجود در حافظه را پاک میکند

clear var_name

یک متغیر خاص را از حافظه پاک میکند

clear radius

```
var_name = input('پیام')
```

عددی را از ورودی دریافت کرده و باز میگرداند.

```
var_name = input('s', 'پیام')
```

یک رشته را از ورودی دریافت کرده و باز میگرداند.

```
radius = input('enter radius: ')
```

```
disp( var_name )
```

مقدار متغیر را در خروجی چاپ میکند

```
disp( radius )
```

```
clc
```

پنجره command window را پاک میکند

format type

نحوه نمایش مقادیر عددی را در پنجره command window تعیین

دستور	شرح	مثال
format short	4 رقم اعشار	3.1416
format long	14 رقم اعشار	3.141592653589793
format short e	4 رقم اعشار و توان	3.1416e+000
format short g	4 رقم اعشار و با/بدون توان	3.1416
format long e	15 رقم اعشار با توان	3.141592653589793e+000
format long g	15 رقم اعشار با/بدون توان	3.14159265358979
format bank	دو رقم اعشار	3.14
format rat	اعداد گویا	355/113

format type

```
>> x = pi
```

```
x =
```

```
3.1416
```

```
>> format bank
```

```
>> x
```

```
x =
```

```
3.14
```

```
>> format short
```

```
>> x
```

```
x =
```

```
3.1416
```

```
>> format long
```

```
>> x
```

```
x =
```

```
3.141592653589793
```

```
>> format rat
```

```
>> x
```

```
x =
```

```
355/113
```

zeros(n)

یک ماتریس $n \times n$ با عناصر صفر ایجاد میکند

zeros(m,n)

یک ماتریس $m \times n$ با عناصر صفر ایجاد میکند

ones(n)

یک ماتریس $n \times n$ با عناصر یک ایجاد میکند

ones(m,n)

یک ماتریس $m \times n$ با عناصر یک ایجاد میکند

eye(n)

یک ماتریس واحد $n \times n$ ایجاد میکند

eye(m,n)

یک ماتریس واحد $m \times n$ ایجاد میکند

```
>> a = ones(3)
```

```
a =
```

```
1     1     1
1     1     1
1     1     1
```

```
>> a = ones(3,4)
```

```
a =
```

```
1     1     1     1
1     1     1     1
1     1     1     1
```

```
>> a = ones(3,4)*5
```

```
a =
```

```
5     5     5     5
5     5     5     5
5     5     5     5
```

```
>> a = eye(3)
```

```
a =
```

```

1     0     0
0     1     0
0     0     1

```

```
>> a = eye(3,4)
```

```
a =
```

```

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

```

`rand(n)``rand(m,n)`

یک ماتریس حاوی اعداد تصادفی بین صفر و یک، با توزیع یکنواخت ایجاد میکند.

```
>> 10*rand(3,4)
```

```
ans =
```

9.5717	1.4189	7.9221	0.3571
4.8538	4.2176	9.5949	8.4913
8.0028	9.1574	6.5574	9.3399

`randn(n)``randn(m,n)`

یک ماتریس حاوی اعداد تصادفی منفی و مثبت حول صفر، با توزیع نرمال ایجاد میکند.

```
>> 10*randn(3,4)
```

```
ans =
```

```
4.8889 -3.0344 8.8840 -8.0950
```

```
10.3469 2.9387 -11.4707 -29.4428
```

```
7.2689 -7.8728 -10.6887 14.3838
```

magic(n)

یک ماتریس $n \times n$ حاوی اعداد یک تا n^2 ایجاد میکند که مجموع هر سطر و ستون یکسان است

```
>> A = magic(3)
```

```
A =
```

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

pascal(n)

یک ماتریس پاسکال $n \times n$ ایجاد میکند.

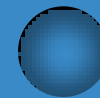
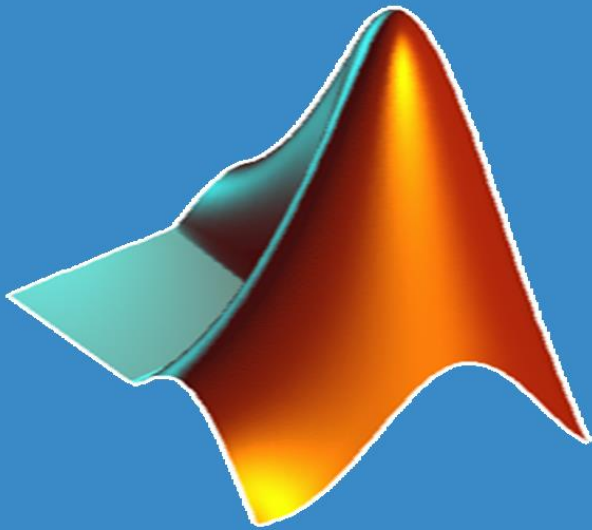
```
>> A = pascal(4)
```

A =

1	1	1	1
1	2	3	4
1	3	6	10
1	4	10	20

$$6+4 = 10$$

توابع آرایه‌ای



```
numel( var_name )
```

تعداد عناصر را باز میگرداند

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

```
a =
```

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

```
>> numel(a)
```

```
ans =
```

```
6
```

```
length( var_name )
```

طول بزرگترین بعد را باز میگرداند

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

```
a =
```

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

```
>> length(a)
```

```
ans =
```

```
3
```

```
d = size( var_name )
```

اندازه را باز میگرداند

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

```
>> d = size(a)
```

```
a =
```

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

```
d =
```

```
3     2
```

```
[row col] = size( var_name )
```

اندازه ماتریس را باز میگرداند

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

```
a =
```

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

```
>> [row col] = size(a)
```

```
row =
```

```
3
```

```
col =
```

```
2
```


min(var_name)

اگر یک بردار سطری یا ستونی به این تابع داده شود کوچکترین عنصر آنرا باز میگرداند

اگر یک ماتریس به این تابع داده شود کوچکترین مقدار هر ستون را باز میگرداند

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

```
a =
```

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

```
>> min(a)
```

```
ans =
```

```
1    2
```

```
>> min(a(:))
```

```
ans =
```

```
1
```

max(var_name)

اگر یک بردار سطری یا ستونی به این تابع داده شود بزرگترین عنصر آنرا باز میگرداند
اگر یک ماتریس به این تابع داده شود بزرگترین مقدار هر ستون را باز میگرداند

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

```
a =
```

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

```
>> max(a)
```

```
ans =
```

```
5     6
```

```
>> max(a(:))
```

```
ans =
```

```
6
```

sort(var_name)

اگر یک بردار سطری یا ستونی به این تابع داده شود آنرا مرتب کرده باز میگرداند
اگر یک ماتریس به این تابع داده شود، هر ستون را مرتب کرده و باز میگرداند

```
>> z = [10 12; 35 14; 11 22]
```

```
z =
```

```
10    12
35    14
11    22
```

```
>> sort(z)
```

```
ans =
```

```
10    12
11    14
35    22
```

```
>> sort(z(:))
```

```
ans =
```

```
10
11
12
14
22
35
```

sum(var_name)

اگر یک بردار سطری یا ستونی به این تابع داده شود حاصل جمع عناصر آنرا باز میگرداند
اگر یک ماتریس به این تابع داده شود حاصل جمع هر ستون را باز میگرداند

```
>> z = [10 12; 35 14; 11 22]
```

```
z =
```

```
10    12
35    14
11    22
```

```
>> sum(z)
```

```
ans =
```

```
56    48
```

```
>> sum(z(:))
```

```
ans =
```

```
104
```

mean(var_name)

اگر یک بردار سطری یا ستونی به این تابع داده شود میانگین عناصر آنرا باز میگرداند
اگر یک ماتریس به این تابع داده شود میانگین عناصر هر ستون را باز میگرداند

```
>> z = [10 12; 35 14; 11 22]
```

```
z =
```

```
10    12
35    14
11    22
```

```
>> mean(z)
```

```
ans =
```

```
18.6667    16.0000
```

```
>> mean(z(:))
```

```
ans =
```

```
17.3333
```

```
ind = find( var_name )
```

اندیس عناصر غیر صفر را باز میگرداند

```
>> b = [4 3 2 5 7]
```

```
b =
```

```
4      3      2      5      7
```

```
>> ind = find(b==5)
```

```
ind =
```

```
4
```

```
[ row col ] = find( var_name )
```

شماره سطر و ستون عناصر غیر صفر را باز میگرداند

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

```
a =
```

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

```
>> [row col] = find(a==4)
```

```
row =
```

```
2
```

```
col =
```

```
2
```

isempty(var_name)

در صورت خالی بودن متغیر، مقدار یک و در غیر اینصورت مقدار صفر را باز میگرداند

```
>> isempty(z)
```

```
ans =
```

```
0
```

```
>> isempty([])
```

```
ans =
```

```
1
```


isnan(var_name)

آرایه ای هم اندازه آرایه ورودی باز میگرداند که برای هر عنصر، اگر مقدار عنصر مذکور nan باشد حاوی یک و در غیر اینصورت حاوی صفر خواهد بود

```
>> w = [5 7 0]
```

```
w =
```

```
5      7      0
```

```
>> isnan(p)
```

```
ans =
```

```
0      0      1
```

```
>> p = w ./ 0
```

```
p =
```

```
Inf     Inf    NaN
```

isinf(var_name)

آرایه ای هم اندازه آرایه ورودی باز میگرداند که برای هر عنصر، اگر مقدار عنصر مذکور بینهایت باشد حاوی یک و در غیر اینصورت حاوی صفر خواهد بود

```
>> w = [5 7 0]
```

```
w =
```

```
5      7      0
```

```
>> isinf(p)
```

```
ans =
```

```
1      1      0
```

```
>> p = w ./ 0
```

```
p =
```

```
Inf    Inf    NaN
```

diag(var_name)

عناصر قطر اصلی را باز میگرداند

```
>> m = magic(3)
```

```
m =
```

8	1	6
3	5	7
4	9	2

```
>> diag(m)
```

```
ans =
```

8
5
2

trace(var_name)

مجموع عناصر قطر اصلی را باز میگرداند

```
>> m = magic(3)
```

```
m =
```

8	1	6
3	5	7
4	9	2

```
>> trace(m)
```

```
ans =
```

```
15
```

rank(var_name)

تعداد ستونهای مستقل خطی را باز میگرداند

```
>> m = magic(3)
```

```
m =
```

8	1	6
3	5	7
4	9	2

```
>> rank(m)
```

```
ans =
```

3

det(var_name)

دترمینان را محاسبه کرده و باز میگرداند

```
>> m = magic(3)
```

```
m =
```

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

```
>> det(m)
```

```
ans =
```

```
-360
```

Questions ?

