

# فصل ۴

## استفاده از فایل‌های اسکریپت و مدیریت داده‌ها

- این فصل به این مطالب می پردازد:
- در یک فایل اسکریپت چگونه داده وارد میشود
  - MATLAB چگونه داده ها را ذخیره میکند
  - روشهای نمایش و ذخیره داده ها
  - چگونگی تبادل داده ها بین MATLAB و برنامه های دیگر

فضای کاری (*workspace*) ساخته شده  
از متغیرهایی است که در یک نوبت اجرای  
MATLAB تعریف و ذخیره میشوند. شامل  
متغیرهایی که

- در پنجره فرمان تعریف شده اند
  - در فایل‌های اسکریپت تعریف شده اند
- فایل‌های اسکریپت به تمام متغیرهایی که در  
پنجره فرمان تعریف شده اند دسترسی دارند

# دستور whos مانند دستور who است (فصل ۱) ولی با اطلاعات بیشتر

```
>> 'Variables in memory'
```

Typing a string.

```
ans =  
Variables in memory
```

The string is assigned to ans.

```
>> a = 7;  
>> E = 3;  
>> d = [5, a+E, 4, E^2]
```

Creating the variables a, E, d, and g.

```
d =  
      5      10      4      9  
>> g = [a, a^2, 13; a*E, 1, a^E]
```

```
g =  
      7      49      13  
     21      1     343  
>> who
```

Your variables are:

```
E      a      ans      d      g
```

The who command displays the variables currently in the workspace.

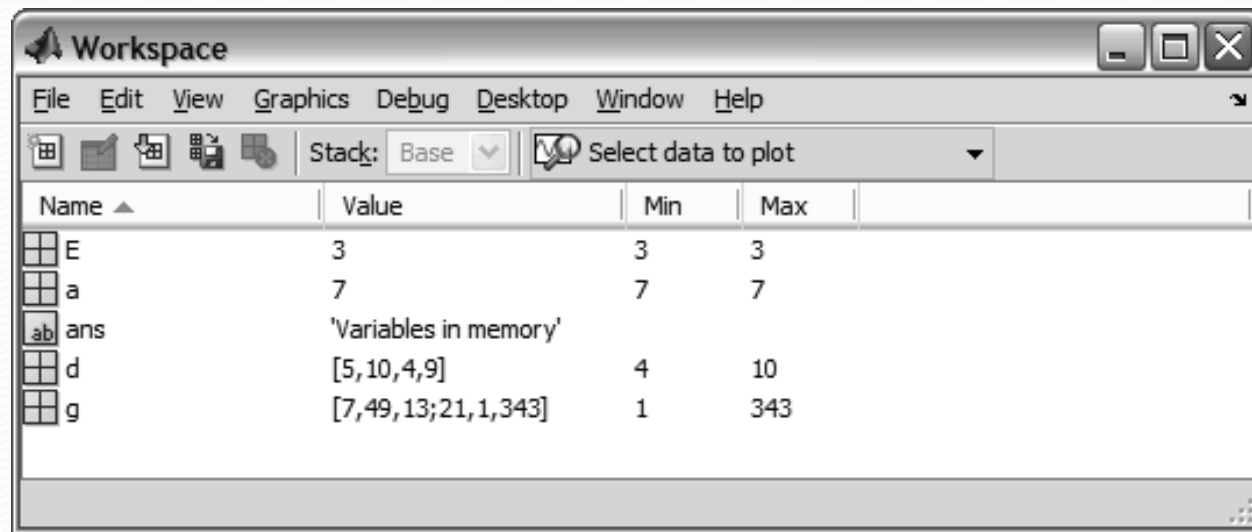
```
>> whos
```

Name	Size	Bytes	Class	Attributes
E	1x1	8	double	
a	1x1	8	double	
ans	1x19	38	char	
d	1x4	32	double	
g	2x3	48	double	

The whos command displays the variables currently in the workspace and information about their size and other information.

# متغیرهای فضای کاری در پنجره فضای کاری نیز نمایش داده می شوند

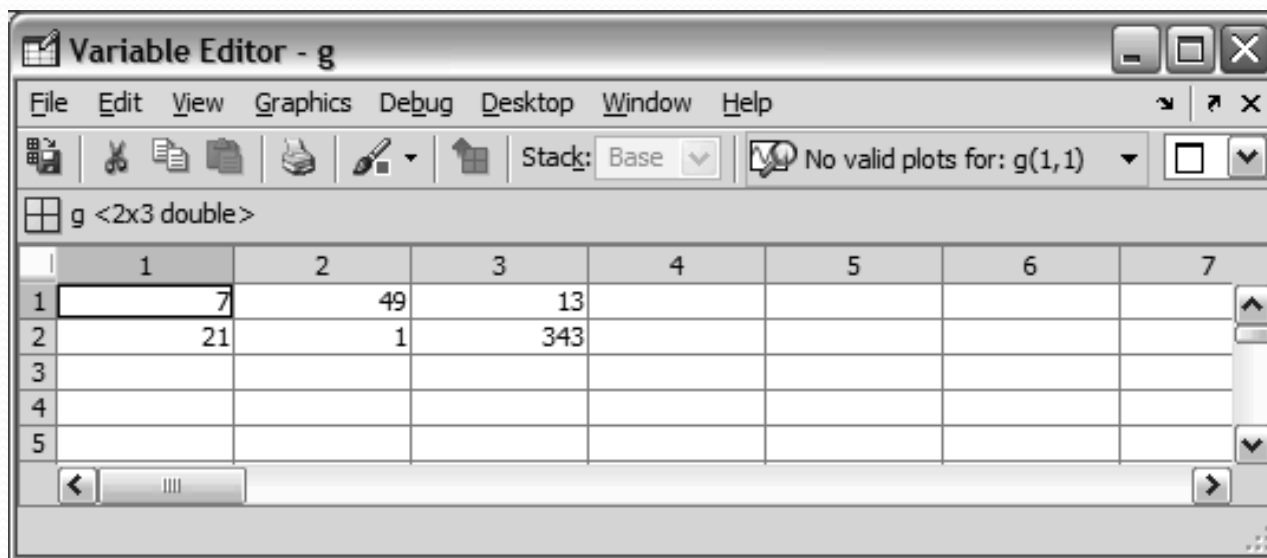
- برای باز کردن پنجره فضای کاری، روی آیکن Layout و سپس Workspace کلیک کنید



## برای ویرایش (تغییر) یک متغیر در پنجره فضای کاری

1. روی متغیر دو بار کلیک کنید تا پنجره ویرایش متغیر باز شود

2. در آن پنجره می توانید عددها را تغییر دهید



## در پنجره ویرایش متغیر

- برای تغییر یک کاراکتر، نشانگر را سمت راست آن قرار دهید و BACKSPACE را بزنید یا سمت چپ آن قرار دهید و DELETE را بزنید
- برای پاک کردن یک عدد، با کشیدن ماوس یا دو بار کلیک کردن آن را انتخاب کنید، سپس DELETE یا BACKSPACE را بزنید

- برای حذف یک متغیر از پنجره فضای کاری
- متغیر را با کشیدن ماوس یا دو بار کلیک کردن انتخاب کنید، سپس  
DELETE یا BACKSPACE را بزنید  
یا
  - راست کلیک کنید و Delete را انتخاب کنید
  - یک متغیر را از پنجره فرمان نیز با این دستور  
میتوان حذف کرد  

```
>> clear variable_name
```

  
مثلاً  

```
>> clear g
```



زمانی که MATLAB یک فایل اسکریپت را اجرا می کند، باید به هر متغیری که در فایل استفاده شده مقدار یا تخصیص داده شده باشد، یعنی متغیرها باید در فضای کاری باشند

به سه روش میتوان به یک متغیر مقدار اختصاص داد


## ۱. تخصیص مقدار در فایل اسکریپت

- جمله مقداردهی قسمتی از متن است
- برای تخصیص مقادیر مختلف، باید فایل را ویرایش، ذخیره و دوباره اجرا کرد

نکته – زمانی که مقدار متغیر (یک عدد) قسمتی از متن است، آن مقدار *hard-coded* نامیده میشود

The following is an example of such a case. The script file (saved as Chapter4Example2) calculates the average points scored in three games.

```
% This script file calculates the average points scored in three games.
% The assignment of the values of the points is part of the script file.
game1=75;
game2=93;
game3=68;
ave_points=(game1+game2+game3)/3
```

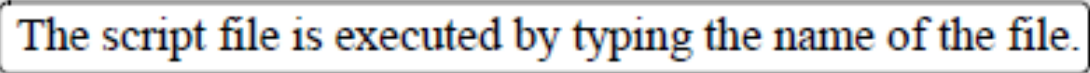


The variables are assigned values within the script file.

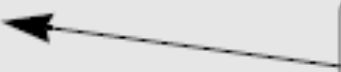
The display in the Command Window when the script file is executed is:

```
>> Chapter4Example2

ave_points =
    78.6667
>>
```



The script file is executed by typing the name of the file.

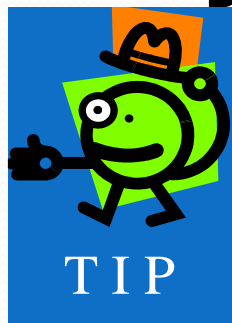


The variable ave\_points with its value is displayed in the Command Window.

## ۲. تخصیص مقدار در پنجره فرمان

- متغیر را در پنجره فرمان تعریف و مقدار دهی کنید
- از قبل میدانیم که فایل‌های اسکریپت متغیر را می‌شناسند
- برای تخصیص مقادیر مختلف، مقداردهی جدید را در پنجره فرمان انجام دهید و فایل را دوباره اجرا کنید
- نیازی به ذخیره مجدد فایل نیست

به جای تایپ مجدد کل دستور، از کلید جهت بالا برای دوباره آوردن دستور و ویرایش آن استفاده کنید



For the previous example in which the script file has a program that calculates the average of points scored in three games, the script file (saved as Chapter4Example3) is:

```
% This script file calculates the average points scored in three games.  
% The assignment of the values of the points to the variables  
% game1, game2, and game3 is done in the Command Window.  
  
ave_points=(game1+game2+game3)/3
```

The Command Window for running this file is:

```
>> game1 = 67;  
>> game2 = 90;  
>> game3 = 81;
```

← The variables are assigned values in the Command Window.

```
>> pChater4Example3
```

← The script file is executed.

```
ave_points =  
    79.3333
```

← The output from the script file is displayed in the Command Window.

```
>> game1 = 87;  
>> game2 = 70;  
>> game3 = 50;
```

← New values are assigned to the variables.

```
>> Chapter4Example3
```

← The script file is executed again.

```
ave_points =  
    69  
>>
```

← The output from the script file is displayed in the Command Window.

### ۳. تخصیص مقدار با درخواست از کاربر

فایل اسکریپت از کاربر می خواهد تا مقداری را وارد کند (*prompt*)، سپس آن مقدار را به متغیر مربوطه اختصاص می دهد

از دستور `input` برای گرفتن مقدار از کاربر استفاده کنید

```
variable_name=input( 'prompt' )
```

prompt متنی است که دستور input در پنجره فرمان نمایش می دهد

- باید متن را بین دو آپوستروف قرار دهید



```
variable_name=input( 'prompt' )
```

زمانی که دستور input اجرا می شود

1. پیغام مربوطه (prompt) در پنجره فرمان نمایش داده می شود

2. نشانگر در سمت راست پیغام قرار می گیرد

3. کاربر مقدار را تایپ کرده و ENTER را می زند

4. مقدار وارد شده توسط کاربر به متغیر مربوطه

تخصیص داده شده و نمایش داده می شود، مگر

اینکه در انتهای دستور input نقطه ویرگول باشد

```
% This script file calculates the average of points scored in
three games.

% The points from each game are assigned to the variables by
% using the input command.

game1=input('Enter the points scored in the first game ');
game2=input('Enter the points scored in the second game ');
game3=input('Enter the points scored in the third game ');
ave_points=(game1+game2+game3)/3
```

### خروجی (در پنجره فرمان)

```
>> Chapter4Example4
Enter the points scored in the first game    67
Enter the points scored in the second game   91
Enter the points scored in the third game    70

ave_points =
    76
>>
```

The computer displays the message. Then the value of the score is typed by the user and the Enter key is pressed.



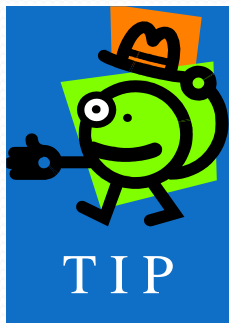
مناسب است که یک فاصله، یا دو نقطه و فاصله، در انتهای پیغام قرار گیرد تا ورودی کاربر از پیغام جدا شود.  
مثال:

```
age = input('Age in 2012');
```

```
age = input('Age in 2012 ');
```

```
age = input('Age in 2012: ');
```

خروجی دستور پس از تایپ عدد ۳۰ توسط کاربر



Age in 201230

نامناسب

Age in 2012 30

مناسب

Age in 2012: 30

بهتر

برای تخصیص یک رشته به یک متغیر نیز می توان  
از کاربر درخواست کرد  
روش ۱

از دستور input به همان صورت که گفته شد  
استفاده کنید، اما کاربر باید آپوستروف ابتدا و انتها  
را وارد کند


```
>> name = input( 'Your name: ' )
```

```
Your name: 'Joe'
```

```
name =
```

```
Joe
```

کاربر باید آپوستروف ها را  
وارد کند



## روش ۲

's' را به عنوان ورودی دوم input بدهید. کاربر  
نباید آپوستروف ها را تایپ کند

```
variable_name=input('prompt', 's')
```

```
>> name=input('Your name: ', 's')
```

Your name: Joe کاربر نباید آپوستروف ها را تایپ کند

```
name =
```

```
Joe
```

وقتی نقطه ویرگول آخر دستور را حذف کنید، نتیجه نمایش داده خواهد شد. کنترلی بر روی ظاهر نمایش نتیجه، مثل تعداد خطوط یا دقت اعداد ندارید. می توانید از دستور `disp` برای داشتن بعضی کنترل ها روی ظاهر نتیجه و از دستور `fprintf` برای داشتن کنترل کامل استفاده کنید

دستور disp مقدار متغیرها یا یک متن را روی صفحه نمایش نشان می دهد

- هر بار در یک خط جدید نمایش می دهد
- اسم متغیر را نشان نمی دهد

یا `disp(variable_name)`

`disp('text string')`

```
>> abc = [5 9 1; 7 2 4] A 2×3 array is assigned to variable abc.
>> disp(abc) The disp command is used to display the abc array.
    5     9     1
    7     2     4 The array is displayed without its name.

>> disp('The problem has no solution.')
The problem has no solution.
>>
```

The disp command is used to display a message.



برای نمایش جدول های دارای عنوان می توانید  
از `disp` استفاده کنید

- کار با آن به علت نداشتن کنترل روی عرض  
ستون ها سخت است – باید عنوانها را با  
اضافه کردن فاصله تنظیم کنید
- بهتر است از `fprintf` استفاده کنید

# fprintf

- مخفف file print formatted
- متن دارای قالب (*formatted*) متنی است که توسط انسان قابل خواندن و درک باشد
- متن بدون قالب از دید انسان درهم و برهم است، ولی کامپیوتر می تواند آن را بخواند
- می تواند روی صفحه نمایش یا در فایل بنویسد
- می تواند اعداد و متن را در خروجی داشته باشد
- کنترل کاملی بر نحوه نمایش خروجی دارد
- استفاده از آن پیچیده است

استفاده از دستور fprintf برای نمایش متن

برای نمایش متن مورد نظر

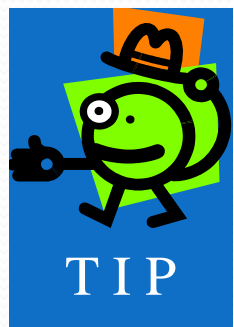
```
fprintf( 'Text to display' )
```

مثال

```
>> fprintf( 'Howdy neighbor' )
```

```
Howdy neighbor>> ← فروجه  
دستور
```

اشکال – اعلان پنجره فرمان (>>) در انتهای  
متن، و نه در ابتدای خط بعد نمایش داده  
میشود



برای اینکه نوشته های بعدی MATLAB

پس از استفاده از دستور fprintf

در ابتدای خط جدید نمایش داده شوند، دو

کاراکتر “\n” را در انتهای متن بنویسید

```
>> fprintf( 'Howdy neighbor\n' )
```

```
Howdy neighbor
```

```
>>
```

میتوانید از `\n` در وسط متن هم استفاده کنید  
تا بقیه آن در خط بعدی نمایش داده شود

```
>> fprintf('A man\nA  
plan\nPanama\n')
```

```
A man
```

```
A plan
```

```
A canal
```

```
Panama
```

```
>>
```

\n یک کاراکتر گریز است، ترکیب خاصی از دو کاراکتر که به fprintf میگوید به جای نمایش آنها، کار خاصی انجام دهد

\n باعث نمایش ادامه متن در ابتدای خط - \n بعد میشود

\t جدول بندی افقی - \t

چند مورد دیگر هم هست

```
fprintf( format, n1, n2, n3 )
```

ورودی

تعیین کننده حالت نمایش عدد

```
>> fprintf( 'Joe weighs %6.2f kilos', n1 )
```

رشته قالب بندی

```
>> fprintf( 'Joe weighs %6.2f kilos', n1 )
```



رشته قالب بندی

- میتواند شامل متن و/یا تعیین کننده حالت نمایش عدد باشد
- باید بین دو آپوستروف قرار گیرد، نه گیومه



```
>> fprintf( 'Joe is %d weighs %f kilos', age, weight )
```



### ورودی ها

- تعداد ورودیها و تعیین کننده های حالت نمایش اعداد باید برابر باشد
- سمت چپی ترین تعیین کننده حالت نمایش عدد، تعیین کننده حالت نمایش اولین ورودی، بعدی دومین ورودی، و به همین ترتیب است

## تعیین کننده حالت نمایش عدد

```
>> fprintf( 'Joe weighs %f kilos', n1 )
```

## تعیین کننده های حالت نمایش عدد رایج

- %f ممیز ثابت (ممیز همیشه بین یکان و دهه اعشار است، مثل ۳.۱۴ و ۵۶.۸)
- %e نماد علمی، مثل 2.99e+008
- %d اعداد صحیح (بدون ممیز)
- %s رشته ای از کاراکترها

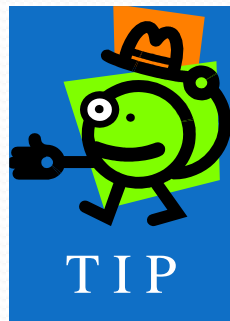
تعیین کننده حالت نمایش عدد

```
>> fprintf( 'Joe weighs %6.2f kilos', n1 )
```

برای کنترل نحوه نمایش در حالت ممیز ثابت یا  
نماد علمی، از `%w.pf` یا `%w.pe` استفاده کنید

- `w` = عرض: حداقل تعداد کاراکترها برای نمایش
- `p` = دقت: تعداد ارقام در سمت راست ممیز

اگر `w` را حذف کنید، MATLAB اعداد  
را با طول و دقت درست و مناسب  
نمایش خواهد داد



```
>> e = exp( 1 );
>> fprintf( 'e is about %4.1f\n', e )
e is about 2.7
>> fprintf( 'e is about %10.8f\n', e )
e is about 2.71828183
>> fprintf( 'e is about %10.8e', e )
e is about 2.71828183e+000
>> fprintf( 'e is about %10.2e', e )
e is about 2.72e+000
>> fprintf( 'e is about %f\n', e )
e is about 2.718282
```

از کاراکترهای گریز برای نمایش کاراکترهای  
استفاده شده در تعیین کننده حالت نمایش اعداد  
استفاده کنید

- برای نمایش علامت درصد، از %% استفاده کنید
- برای نمایش آپوستروف، از ' ' استفاده کنید
- برای نمایش ممیز بر عکس، از \. استفاده کنید

## این رشته ها را ایجاد کنید

- Mom's apple 3.14
- Mom's apple 3.1415926
- Mom's apple 3.1e+000

```
>> fprintf( 'Mom''s apple %.2f\n', pi )
```

```
Mom's apple 3.14
```

```
>> fprintf( 'Mom''s apple %.7f\n', pi )
```

```
Mom's apple 3.1415927
```

```
>> fprintf( 'Mom''s apple %.1e\n', pi )
```

```
Mom's apple 3.1e+000
```

# رشته های قالب بندی معمولاً طولانی هستند. میتوان یک رشته را اینگونه در چند خط نوشت

1. یک کروشه باز ([) در ابتدای اولین آپوستروف بگذارید
2. در انتهای هر خط یک آپوستروف بگذارید
3. پس از آن آپوستروف سه نقطه بگذارید
4. ENTER را بزنید تا نشانگر به خط بعد برود
5. بقیه متن را داخل آپوستروف بنویسید
6. یک کروشه بسته ] بگذارید
7. بقیه دستور fprintf را بنویسید

## مثال

```
>> weight = 178.3;  
>> age = 17;  
>> fprintf( ['Tim weighs %.1f lbs'...  
' and is %d years old'], weight, age )
```

Tim weighs 178.3 lbs and is 17 years old



`fprintf` برداری است، یعنی اگر ورودی یک بردار یا ماتریس باشد، دستور برای نمایش تمام درایه ها تکرار می شود

- این کار در ماتریس ها ستون به ستون انجام میشود

```
x=1:5;
```

Create a vector x.

```
y=sqrt(x);
```

Create a vector y.

```
T=[x; y]
```

Create  $2 \times 5$  matrix T, first row is x, second row is y.

```
fprintf('If the number is: %i, its square root is: %f\n',T)
```

The `fprintf` command displays two numbers from T in every line.

When this script file is executed, the display in the Command Window is:

```
T =
    1.0000    2.0000    3.0000    4.0000    5.0000
    1.0000    1.4142    1.7321    2.0000    2.2361
If the number is: 1, its square root is: 1.000000
If the number is: 2, its square root is: 1.414214
If the number is: 3, its square root is: 1.732051
If the number is: 4, its square root is: 2.000000
If the number is: 5, its square root is: 2.236068
```

The 2 × 5 matrix T.

The fprintf command repeats five times, using the numbers from the matrix T column after column.

استفاده از دستور fprintf برای نمایش خروجی در یک فایل:

برای نوشتن در فایل سه گام لازم است

گام اول: - فایل را باز کنید

```
fid=fopen('file_name','permission')
```

مشخص کننده فایل، فایلی که باید خروجی - fid

در آن نوشته شود را برای دستور مشخص می کند

مشخص میکند فایل چگونه - permission

استفاده خواهد شد، مثلاً برای خواندن، نوشتن یا هر دو

## برخی permission های رایج

- `r` - فایل را برای خواندن باز میکند
  - `w` - فایل را برای نوشتن باز میکند. اگر فایل از قبل موجود باشد محتوای آن پاک خواهد شد. اگر موجود نباشد ایجاد خواهد شد
  - `a` - مشابه `w` با این تفاوت که در صورت وجود فایل داده ها به انتهای آن اضافه خواهد شد
  - اگر هیچکدام مشخص نشده باشند، `fopen` از `r` استفاده میکند
- Help دستور `fopen` را برای بقیه موارد ببینید

## گام دوم:

با استفاده از `fprintf` در فایل بنویسید.  
مانند قبل از آن استفاده کنید، فقط قبل از  
رشته قالب بندی `fid` را بنویسید، یعنی

```
fprintf(fid, 'format  
string', variables)
```

دستور از طریق `fid` میفهمد که باید به جای  
نمایش روی مانیتور، در یک فایل بنویسد

## گاه سوم:

زمانی که نوشتن در فایل تمام شد، آن را با این دستور ببندید

```
fclose(fid)
```

- زمانی که فایل را ببندید، دیگر نمیتوانید از آن `fid` استفاده کنید، تا وقتی که با دستور `fopen` یک `fid` جدید بگیرید

هر فایل را پس از استفاده ببندید.

فایل‌های باز زیاد برای MATLAB

مشکل ایجاد میکنند



## نکات دیگر

- اگر نام فایلی که به `fopen` می دهید مسیر نداشته باشد، MATLAB آن را در پوشه جاری ایجاد میکند
- میتوانید چند فایل باز همزمان داشته باشید و از `fprintf` برای نوشتن در تمام آنها با `fid`های مختلف استفاده کنید
- فایل‌های ایجاد شده توسط `fprintf` را میتوان توسط هر ویرایشگری خواند، شامل ویرایشگر MATLAB یا غیر از آن

از دستور save برای ذخیره فضای کاری یا داده ها استفاده کنید

از دستور load برای بازیابی فضای کاری یا داده های ذخیره شده استفاده کنید

از هر دو میتوان برای تبادل اطلاعات با دیگر برنامه ها استفاده کرد



از دستور save برای ذخیره همه یا قسمتی از  
متغیرهای فضای کاری استفاده کنید  
دو حالت دارد

```
save file_name  
save( 'file_name' )
```

هر دو همه متغیرهای فضای کاری را ذخیره  
میکند، شامل نام، نوع، اندازه و مقدار آن ها

برای ذخیره متغیرهای مشخص، آنها را پس از نام فایل بنویسید. مثلاً، برای ذخیره دو متغیر به نام های `var1` و `var2`:

```
save file_name var1 var2  
save( 'file_name', 'var1', 'var2' )
```

همه حالتها متغیرها را در فایلی به نام  
"file\_name.mat" ذخیره میکنند

- فایل "mat" نامیده می شود
- این فایل بدون قالب است (binary)
- فقط MATLAB میتواند آن را بخواند
- نمیشود آن را در ویرایشگرهای متن یا ویرایشگر  
MATLAB خواند

برای ذخیره به صورت قالب دار (ASCII)

```
save file_name -ascii
```

مهم - فقط مقدار متغیرها را ذخیره میکند، نه اطلاعات دیگر،  
حتی نام آنها

- مشابه قبل، میتواند تنها متغیرهای خاصی را ذخیره کند
- معمولاً برای ذخیره مقدار تنها یک متغیر به کار میرود

برای بازیابی داده های یک فایل mat در فضای کاری

```
load file_name
```

```
load( 'file_name' )
```

برای بازیابی متغیرهای مشخص از یک فایل mat،  
مثلاً var1 و var2

```
load file_name var1 var2
```

```
load( 'file_name' , 'var1' , 'var2' )
```

- اگر متغیر از قبل در فضای کاری موجود باشد،  
مقدار آن با مقدار فایل جایگزین خواهد شد



# برای بازیابی داده های یک فایل متنی در فضای کاری

```
load file_name
```

```
variable = load( 'file_name' )
```

- در حالت اول، متغیری به نام file\_name ایجاد کرده و همه داده ها را در آن ذخیره میکند
- اگر همه سطرهای فایل تعداد ستونهای یکسان نداشته باشند خطا می دهد
- حتی اگر داده ها مربوط به چند متغیر باشند، load همه آنها را در یک متغیر می ریزد
- در این حالت مفید نیست



- MATLAB معمولاً برای پردازش داده های ایجاد شده توسط برنامه های دیگر به کار می رود
- بعضی وقتها لازم است داده های MATLAB به برنامه های دیگر منتقل شوند
- در این فصل فقط داده های عددی مطرح میشوند
- MATLAB دستوراتی برای ذخیره و بازیابی داده ها از تعدادی از برنامه های دیگر دارد
- همچنین میشود به MATLAB گفت داده ها چه قالبی دارند

- برای معرفی تبادل داده با برنامه های دیگر، Microsoft Excel به عنوان نمونه بیان میشود
- معمولاً برای ذخیره داده ها استفاده میشود
- با بسیاری از برنامه ها که داده تولید میکنند کار میکند
- معمولاً توسط افرادی استفاده میشود که داده های فنی دارند ولی MATLAB برای آنها به درد نمیخورد



## تبادل داده با Excel:

### وارد کردن (خواندن) داده ها از Excel با

```
variable_name=xlsread('filename')
```

- همه داده ها را در یک متغیر ذخیره میکند
- اگر فایل Excel چند شیت داشته باشد، اولی را میخواند
- برای خواندن بقیه شیت ها، نام آنها را به دستور بدهید
- میتوان قسمت خاصی از شیت را با مشخص کردن آن در دستور خواند

## صدور (نوشتن) داده ها به Excel با

```
xlswrite('filename', variable_name)
```

- میتوان در دستور نام شیت و ناحیه خاصی از آن را مشخص کرد

منوی جادویی وارد کردن داده ها در MATLAB  
راهی نیمه خودکار برای خواندن داده ها از هر فایل  
است

- منوی جادویی قالبی که مدس میزند را نشان میدهد
- سپس کاربر میتواند قالب را تنظیم کند

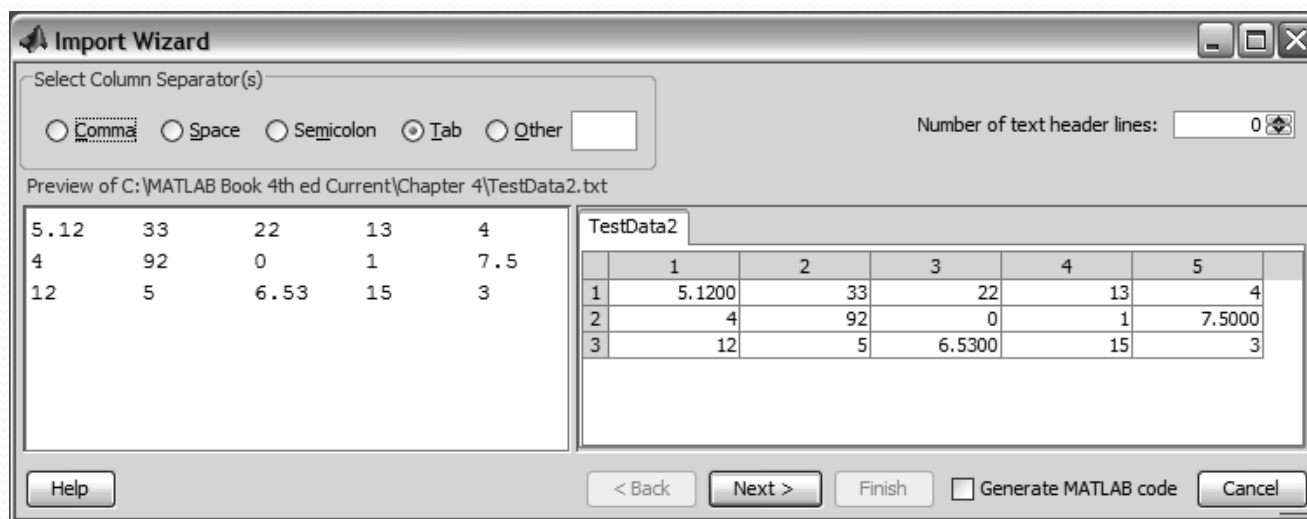
دو راه برای باز کردن منوی جادویی وارد کردن

1. در پنجره اصلی MATLAB آیکن Import Data را  
کلیک کنید

2. با دستور `uiimport`

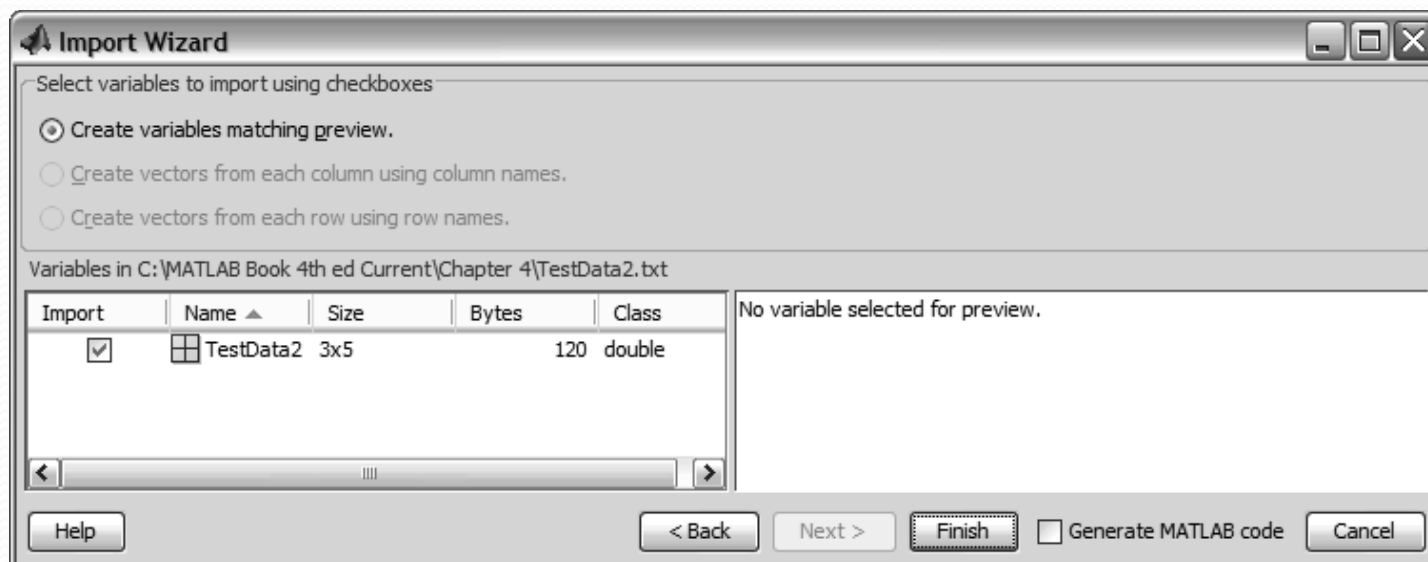
# نمای اول منوی جادویی

- منوی جادویی جعبه گفتگوی انتخاب فایل را نشان میدهد
- کاربر فایل را انتخاب میکند
- منوی جادویی قسمتی از داده ها را به صورتی که در فایل هست و به صورتی که آن ها را پیش بینی میکند نشان میدهد
- کاربر میتواند جدا کننده ستونها یا تعداد خطوط عناوین را تغییر دهد (به گونه ای که منوی جادویی آن ها را بخواند)



## نمای دوم منوی جادویی

- نام و اندازه متغیری که ایجاد خواهد کرد را نشان میدهد
- زمانی که کاربر Finish را بزند، منوی جادویی آن متغیر را در فضای کاری ایجاد خواهد کرد
- نام متغیر نام فایل است



# شماره تمرین های منتخب

۲۴ •

۲۶ •

۲۸ •

۲۹ •

۶ •

۱۳ •

۱۴ •

۱۵ •

۱۶ •

۱۷ •

۱۹ •

۲۱ •

۲۳ •