

حضرت علی علیه السلام:

«أَشْبَعُ النَّاسِ مَنْ غَلَبَ الْجَهْلَ بِالْعِلْمِ»

شجاع‌ترین مردم کسی است که به وسیله دانش بر نادانی خود غلبه کند.



# آموزش نرم افزار MATLAB

علی محمد لطیف و محمد رضا دهقانی محمود آبادی

دانشگاه شهردر

دانشکده برق و کامپیوتر



😊 An easy way to learn MATLAB is to sit down at a computer and follow along with the examples given in this tutorial.

😊 MATrix LABoratory =MATLAB



😊 برای نگهداری برنامه‌هایتان یک فلد بسازید.

😊 از کنار دکمه‌ی درایو جاری از **Browse** استفاده کنید و بعد از دکمه‌ی **Make New Folder** استفاده کنید.

😊 برای اجرای برنامه‌هایتان درایو جاری را درایو خودتان قرار دهید.

😊 می‌توانید از منوی فایل از گزینه **Set Path** درایو برنامه‌هایتان را به مسیر **Matlab** اضافه کنید.





😊 از منوی فایل گزینه‌ی **preferences** می‌توانید تنظیمات صفحه کار خود را تنظیم کنید.

😊 با توجه به این که فونت برای آموزش ریز می‌باشد با انتخاب فونت آن را تغییر دهید.

😊 زبان برنامه‌نویسی **Matlab** مانند **Java** به صورت **interpreted** است. برای اجرا خط به خط اجرا می‌شود.



😊 این زبان **autocomplete** است. یعنی با تایپ قسمتی از دستور و استفاده از کلید **tab** می توان دستور مورد نظر را انتخاب و کامل نمود.

😊 توابع را در این زبان به دو طریق **Script** و **Function** می توان نوشت.

`edit test1.m`



😊 از اولین خط **Function** می‌توان برای تعریف **help** تابع استفاده کرد.

😊 متغیرهای تعریف شده در **Script** داخل پنجره **Matlab** قابل دسترسی هستند ولی متغیرهای **Function** داخل پنجره **Matlab** قابل دسترسی یا تغییر نیستند.





MATLAB 7.7.0 (R2008b)

File Edit Debug Parallel Desktop Window Help

Current Directory: C:\Acads\CSE455

Shortcuts How to Add What's New

Current Directory

Name	Date Modified
CSE 455.pptx	1/3/10 2:04 PM
~\$CSE 455.pptx	1/3/10 2:00 PM

CSE 455.pptx (PPTX File)

No details available

9

Alimohammad Latif & MohammadReza  
DehghaniMahmoudAbadi, Yazd University

Command Window

fx >> |

Workspace

Name	Value	Min
Workspace		

Command History

```
%-- 1/3/10 2:06 PM --%
cd ..
ls
cd Acads
ls
cd CSE455
clear
clear 3/2/2019
clrscr
```



😊 در این نرم افزار متغیرها به صورت بردار می باشند.

😊 برای اجرای هر دستور کافی است پس از تایپ آن کلید **Enter** زده شود.

😊 برای تعریف بردارهای سطری متغیر با علامت کروشه باز و بسته و نوشتن اعداد داخل آن تعریف می شود.

$a = [1 \ 2 \ 3 \ 4 \ 5]$



😊 برای جدا کردن عضوهای یک بردار سطری می توان از کاما نیز استفاده کرد.

$a = [1, 2, 3, 4, 5]$

😊 برای جدا کردن عضوهای یک بردار سطری می توان از کاما و فاصله نیز استفاده کرد.

$a = [1\ 2, 3\ 4, 5]$



😊 برای تعریف بردار ستونی بین عضوها از علامت  
سمی کلون استفاده می شود.

$$a = [1; 2; 3; 4; 5]$$

😊 برای تعریف ماتریس عضوهای یک سطر با کاما یا فاصله  
جدا می شوند و سطرهای ماتریس با سمی کلون از هم جدا  
می شوند.

$$a = [1 \ 2 \ 3; 4 \ 5 \ 6; 7 \ 8 \ 9]$$



😊 این نرم افزار **case-sensitive** است یعنی متغیر  
با حرف کوچک و متغیر با حرف بزرگ با هم فرق دارند.

`a = [1 2 3]`

`A`

??? Undefined function or variable 'A'.



😊 اگر در انتهای یک دستور سمی کلون نگذاریم نتیجه‌ی محاسبات دیده می‌شود؛ ولی اگر سمی کلون بگذاریم عملیات انجام می‌شود ولی نتیجه‌ی محاسبات دیده نمی‌شود.

```
>> a = 2+3-5
```

```
a =
```

```
0
```

```
>> a = 2+3-5;
```

```
>>
```





😊 برای تعریف یک ماتریس با زدن **Enter** می توان  
سطرها را از هم جدا کرد.

$a = [1 \ 2 \ 3$

$4 \ 5 \ 6$

$7 \ 8 \ 9]$

😊 دقت کنید ایندکس بردار و ماتریس از یک شروع  
می شود.



😊 اگر یک سطر طولانی باشد می‌توان با گذاشتن سه نقطه ادامه دستور را در سطر بعد نوشت.

```
g = [1 2 3 ...  
4 5; 6 7 8 9 10; 11 ...  
1 2 13 14 15]
```



😊 بعد از تعریف یک متغیر می توان این متغیر را گسترش داد. عناصر تعریف نشده با مقدار صفر پر می شود.

$$g(4,1) = 16$$

😊 برای مقداردهی به یک عنصر ماتریس به صورت زیر عمل می شود.

$$g(4, 2) = 15$$

😊 تذکره: از کروش استفاده نکنید.



😊 برای تعریف یک محدوده از اعداد به فرم زیر عمل می‌شود.

$$e = 1:5$$

$$p = 1:2:10$$

$$q = 10:0$$

$$e = 100:-10:50$$



😊 دقت کنید با قرار دادن مقدار جدید در یک متغیر مقدار قبلی پاک خواهد شد.

😊 دقت کنید در یک خط می‌توان چندین فرمان را با هم اجرا کرد. فقط کافی است فرمان‌ها را با کاما یا با سمیکلن از هم جدا کرد.

😊 دقت کنید هرگاه نتیجه‌ی محاسبات در متغیری قرار داده نشود، نتیجه در متغیر **ans** قرار می‌گیرد.



😊 برای دسترسی به یک عضو ماتریس با مشخص کردن سطر و ستون می‌توان به آن عضو دسترسی پیدا کرد.

$$t = g(2, 3)$$

😊 تذکره: در برنامه‌های ابتدایی که دانشجویان می‌نویسند از گروه استفاده می‌کنند و باعث ایجاد خطا می‌شود.





😊 برای دسترسی یک عنصر ماتریس از ایندکس گذاری یک بعدی نیز می توان استفاده کرد. ایندکس به صورت ستونی شمرده می شود.

$$t = g(7)$$

😊 اگر ایندکس خارج از رنج مربوطه باشد با دادن پیام، اشتباه مربوطه اعلام می شود.

$$t = g(4,1)$$



😊 از علامت : برای انتخاب تمام سطر یا تمام ستون استفاده می شود.

$$c = a(:, 3)$$

$$b = a(1:3, 3)$$

$$r = a(2, :)$$

$$t = a(1:2, 1:3)$$



😊 برای پاک کردن صفحه‌ی نمایش از دستور **clc** استفاده می‌شود.

😊 برای حذف کردن یک متغیر از دستور **clear** به همراه نام متغیر یا متغیرها استفاده می‌شود.

😊 برای حذف همه‌ی متغیرها از دستور **clear all** استفاده می‌شود.



😊 برای مشاهده‌ی کلیه‌ی متغیرهای تعریف شده از دستور **who** استفاده می‌شود.

😊 برای مشاهده‌ی اطلاعاتی در مورد یک متغیر به صورت زیر عمل می‌شود.

اسم متغیر **whos**

**whos a, c**



😊 از کلیدهای جهت‌نمای بالا و پایین می‌توان دستورات بعدی و قبلی را آورد.

😊 اگر بر روی یک دستور در قسمت History کلیک کنید آن دستور مجدداً اجرا می‌شود.

😊 برای توقف اجرای یک برنامه می‌توان از کلید ترکیبی CTRL+C یا در بعضی کامپیوترها از CTRL+BREAK استفاده کرد.



😊 برای استفاده انتهای یک متغیر از **end** استفاده می‌شود.

**g(3, 2: end)**

😊 برای مشخص کردن سطر یا ستون خاصی از شماره آن به فرم زیر استفاده می‌شود.

**h = g(:, [1 3 2 4 5])**





😊 همان طور که ملاحظه می شود ترتیب انتخاب را نیز به دلفواه می توان تعیین نمود.

😊 در این نرم افزار می توان ابعاد یک ماتریس را گسترش داد.

$a = [1 \ 2 \ 3; 4 \ 5 \ 6]$

$b = [a; 7 \ 8 \ 9]$



```
m = []
```

```
for i = 1:10
```

```
m = [m; i]
```

```
end
```

😊 در این مثال ماتریس **m** ابتدا خالی است و سپس در حلقه‌ی **for** این ماتریس بزرگ‌تر می‌شود.



😊 برای تبدیل یک ماتریس به یک بردار با استفاده از :  
می توان این عمل را به صورت زیر انجام داد.

$$b = a(:)$$

😊 در این حالت عناصر ماتریس به صورت ستونی و از  
ستون اول به بعد پشت سر هم قرار می گیرند. به عبارتی  
در یک ماتریس با  $r$  ردیف و  $c$  ستون عنصر  $a(i, j)$  در  
 $a(i+r(j-1))$  قرار می گیرد.



😊 در این نرم افزار چندین متغیر پر استفاده به صورت کلمه کلیدی تعریف شده است. نام این متغیر به همین صورت باید استفاده شود.

`const = [pi, j, inf, NaN]`

**NaN : Not a Number**

😊 از `i` و `j` می توان به عنوان یک متغیر نیز استفاده کرد.

`i = 5`



😊 اگر برای  $i$  و  $j$  مقداری تعریف نشود عدد موهومی می باشد.

clear i

😊 لازم به ذکر است متغیر موهومی نیز جداگانه قابل تعریف است.

$ii = \text{sqrt}(-1)$



😊 برای دیدن مقدار یک متغیر کافی است اسم متغیر را  
نوشت

i

😊 برای تعریف یک متغیر به صورت زیر عمل می شود.

num = 25

z = 2+3i

z = [1+2i 7-3i 3+4i; 6-2i 9i 4+7i ]





😊 برای متغیرهای عددی ۶۴ بیت و برای متغیرهای کاراکتری ۱۶ بیت استفاده می‌شود.

`s = 'Hello World'`

😊 برای نام متغیر اولین کاراکتر باید حرف باشد.

😊 سعی کنید از کلمات رزرو شده استفاده نکنید.

😊 تابع **iskeyword** برای چک کردن کلمات رزرو شده استفاده می‌شود.



😊 از علامت ‘ برای ترانهاده کردن و مزدوج کردن استفاده می‌شود و از علامت ’. برای ترانهاده استفاده می‌شود.

z.’

😊 این عمل با دستور **transpose** نیز قابل اجرا است.

😊 برای این که نرم‌افزار با یک خط به صورت توضیحات رفتار کند کافی است جلو آن خط از علامت % استفاده شود.



😊 عملیات اصلی که در این نرم افزار مثل سایر زبان‌های برنامه‌نویسی انجام می‌شود.

$$c = (-2 + 2^5) / (3 * 2)$$

😊 اولویت‌های اپراتورها مانند سایر نرم افزارها می‌باشد.



😊 این نرم افزار برای ضرب دو عمگر دارد. یکی ضرب معمولی \* و دیگری ضرب نظیر به نظیر و با علامت \*.

😊 در ضرب معمولی باید دقت نمود با توجه به سطر و ستون ماتریس ضرب امکان پذیر باشد.

😊 در ضرب نظیر به نظیر باید دقت نمود ابعاد دو ماتریس با هم مساوی باشد.



$$a = [1 \ 2; 3 \ 4];$$

$$b = [5 \ 6 \ 7; 8 \ 9 \ 10];$$

$$c = a * a$$

$$c = a .* a$$

$$c = a * b$$

😊 اگر ضرب امکان پذیر نباشد پیام خطا خواهد داد.

$$c = b * a$$



😊 جمع و تفریق ماتریس‌ها مطابق معمول انجام می‌شود.

$$a = [1 \ 2 \ 3];$$

$$b = [4 \ 5 \ 6];$$

$$c = a + b;$$

$$d = a - b;$$

$$e = 2 * a + 3 * b;$$



😊 این نرم افزار یک عملگری به صورت  $\backslash$  نیز دارد.

$$d = 4 \backslash 2$$

😊 این دستور در مورد ماتریس‌ها به صورت دومی در معکوس اولی استفاده می‌شود.





😊 لازم به ذکر است با تقسیم می توان دستگاه چند معادله چند مجهول را حل نمود.

$$A * X = B$$

$$X = A \setminus B$$

$$X * A = B$$

$$X = B / A$$

😊 تقسیم از راست به صورت زیر تعریف می شود.

$$B / A = (A' \setminus B')'$$



😊 یک ویژگی مهم این نرم افزار، برداری بودن این نرم افزار است.

```
t = 0:10;
```

```
x = cos(2*t);
```

```
y = t.*cos(t);
```



😊 دقت کنید این دستورات را در یک زبان برنامه نویسی دیگر باید به صورت زیر نوشت.

```
for k = 1:10  
x(k) = cos(k);  
y(k) = t*cos(k);  
end
```



😊 برای دریافت ورودی از دستور `input` به فرم زیر استفاده می شود.

```
t = input('Input the value of T: ')
```

😊 در این حالت می توان اسکالر، بردار و یا کاراکتر دریافت کرد.



😊 برای خروج از این نرم افزار عبارت **exit** را در خط فرمان تایپ کنید.

😊 قبل از خروج می توان متغیرهای استفاده شده را در یک فایل ذخیره کرد و دفعه بعد این متغیرها را بار نمود.

اسم متغیرها **save Myfile**

اسم فایل **load**





رسول اکرم (ص) مشاهده کرد که در مسجد، دو مجلس تشکیل شده است، یکی مجلس علم که در آن از معارف اسلامی بحث می‌شود و دیگری مجلس دعا که در آن خدا را می‌خوانند. رسول اکرم فرمود: این هر دو مجلس خوب و مورد علاقه ی من است. آن گروه دعا می‌کنند و این گروه درس می‌خوانند و درس می‌گویند، ولی گروه علمی برتر و بالاتر از گروه دعا هستند و من از طرف پروردگار برای تعلیم مردم، مبعوث شده ام. سپس رسول اکرم (ص) به گروه معلمین و مصلّین پیوست و با آنان در مجلس علم نشست.

منبع: بهار، جلد ۱، ص ۶۴





# آموزش نرم افزار MATLAB

علی محمد لطیف  
دانشگاه شهردر  
دانشکده برق و کامپیوتر



😊 نرم افزار **Matlab** تعدادی تابع ریاضی آماده دارد که برای معرفی آنها از دستور زیر استفاده کنید.

**help elfun**

😊 با تایپ دستور فوق کلیدی تابع های ریاضی اولیه با مختصر توضیحی ارائه می شود.



تابع	توضیح
sin	سینوس مثلثاتی
sind	سینوس (آرگومان بر حسب درجه)
sinh	سینوس هیپربولیک
asin	آرک سینوس
asind	آرک سینوسی (درجه)
asinh	آرک سینوس هیپربولیک

😊 توابع مثلثاتی مثل سینوس، کسینوس، ... که برای محاسبه در حالت رادیان استفاده می‌شوند.



😊 تابع hypot

😊 توابع نمایی

😊 توابع لگاریتم

😊 توابع اعداد مقلط

😊 توابع گرد کردن



توضیح	تابع
جزر مجموع مربعات	hypot
تابع نمایی	exp
تابع نمایی منهای یک	expm1
لگاریتم نپیرین	log
لگاریتم عدد به اضافه یک	log1p
لگاریتم مبنای ده	log10
لگاریتم مبنای دو	log2
دو به توان یک عدد	pow2



توضیح	تابع
جزر یک عدد	sqrt
ریشه یک عدد	nthroot
نزدیک ترین توانی از دو	nextpow2
قدر مطلق	abs
زاویه	angle
سافت یک عدد مختلط	complex
قسمت موهومی	image
قسمت حقیقی	real
آیا عدد حقیقی است؟	isreal



توضیحات	تابع
عدد را به سمت صفر گرد می‌کند.	fix
عدد را به سمت منهای بینهایت گرد می‌کند.	floor
عدد را به سمت مثبت بینهایت گرد می‌کند.	ceil
عدد را به سمت نزدیک‌ترین عدد صحیح گرد می‌کند.	round
مد عدد اول به عدد دوم را برمی‌گرداند.	mod
باقی مانده تقسیم دو عدد را برمی‌گرداند.	rem
علامت عدد را برمی‌گرداند.	sign





😊 این زبان تعدادی تابع اولیه مربوط به ماتریس‌ها دارد که لیست این توابع با دستور زیر قابل مشاهده است.

help elmat

😊 این تابع‌ها در مورد تولید ماتریس‌های خاص و پر کاربرد می‌باشد.



توضیحات	تابع
یک ماتریس پر از صفر تولید می‌کند.	zeros
یک ماتریس پر از یک تولید می‌کند.	ones
یک ماتریس واحد تولید می‌کند.	eye
یک ماتریس را تکرار می‌کند	repmat
ماتریس تصادفی یکنواخت استاندارد	rand
ماتریس تصادفی نرمال استاندارد	randn



توضیح	تابع
تولید اعداد با خواص یکسان	linspace
تولید یک شبکه	meshgrid
ابعاد یک متغیر	size
طول یک متغیر	length
تعداد بعد	ndims
تعداد عناصر	numel
برای نمایش	display
آیا متغیر خالی است؟	isempty
آیا متغیر مساوی است؟	isequal



تابع	توضیح
cat	پسباندن
reshape	تغییر شکل
diag	قطری
tril	پایین مثلثی
trilu	بالا مثلثی
fliplr	چرفش چپ و راست
flipud	چرفش بالا و پایین
flipdim	چرفش حول یک بعد
rot90	چرفش ۹۰ درجه



تابع	توضیح
find	برای یافتن عناصر غیر صفر
end	پایان
circshift	چرخش دایره‌ای
eps	اپسیلون
realmax	بزرگترین عدد حقیقی
realmin	بزرگترین عدد حقیقی
hadamard	تولید ماتریس هادامارد
magic	تولید ماتریس جادویی



چند مثال: 😊

$\rho = (1 + \sqrt{5})/2$

$a = \text{abs}(3 + 4i)$

$a = 5 * \text{ones}(3, 3)$

$z = \text{zeros}(3, 4)$

$i = \text{ones}(3)$

$n = \text{round}(10 * \text{rand}(1, 10))$



```
r = randn(3)
```

```
x = -5:0.1:5
```

```
y = randn(1000,1)
```

😊 برای رسم هیستوگرام می‌توان به صورت زیر عمل کرد.

```
hist(y,x)
```

😊 هیستوگرام نشان دهنده تابع توزیع است.



$$b = [1 \ 2; 3 \ 4]$$

$$c = [b \ b, \ b+4 \ b-1]$$

😊 برای حذف کردن عناصر ماتریس می‌توان به خرم زیر عمل نمود.

$$c(:,2) = []$$





```
c = [b, b; b+4 b-1]
```

```
c(1:3:4, :) = []
```

```
c(:, 1:3:4) = []
```

```
c = [b b; b+4 b-1]
```

```
c(1:2:16) = []
```

```
a = rand(3)
```

```
b = [a, zeros(3,2); zeros(2,3), eye(2)]
```



```
n = (0:10)'
```

```
pows = [n, n.^2, 2.^n]
```

```
x = (1:0.1:2)'
```

```
logs = [x, log10(x)]
```



😊 نرم افزار **Matlab** برای رسم نمودارها امکانات خوبی دارد.

😊 دستورات رسم منحنی به صورت زیر می باشند.

`plot`

`xlabel`

`ylabel`



title

grid

axis

stem

subplot

hold on

hold off



```
t = -pi:0.1:pi;  
y = cos(t);  
plot(t, y)  
z = sin(t);  
plot(t, y, t, z)  
plot(t, y, '--')  
plot(t, y, '+.')
```



`plot(t, y, 's')`

`plot(t, y, '-g')`

😊 چند منحنی را با هم می توان رسم کرد.

😊 نوع رسم و رنگ منحنی را می توان تنظیم نمود.

😊 با استفاده از دستور زیر، مشخصات کامل این دستور را بررسی کرد.

`help plot`



😊 بعد از رسم منحنی می‌توان برای محور افقی و عمودی منحنی برچسب زد و برای منحنی یک عنوان انتخاب کرد.

```
xlabel({'first line';'second line'})
```

```
ylabel('George"s  
Popularity','fontsize',12,'fontweight','b')
```



😊 در برچسب‌گذاری می‌توان از فرمول‌نویسی **tex** نیز استفاده کرد.

`xlabel("\it{\omega}_N = e^{\{(-2\pi i)/N\}}")`

😊 بعد از رسم منحنی می‌توان منحنی را شبکه‌بندی کرد.

`grid`





😊 بعد از رسم منحنی با دستور **axis** می توان رنج تغییرات منحنی را تنظیم نمود.

```
axis([-2, 2, 1, 10])
```

😊 از فرمان **stem** برای رسم نمودارهای گسسته استفاده می شود.

```
y=1:10  
stem(y)
```



😊 از فرمان subplot برای رسم هم‌زمان چند منحنی می‌توان استفاده کرد.

```
t = 1:10;
```

```
z = cos(t)
```

```
y = t.^2;
```

```
subplot(211)
```

```
plot(t, z)
```



subplot(212)  
plot(t, y)

😊 می‌توان صفحه را به چهار قسمت تبدیل کرد و در هر قسمت منحنی را رسم کرد.

subplot(221), ....., subplot(222), ....  
subplot(223), ....., subplot(224)



111

211

221

222

121

122

212

223

224

221

222

121

222

212

224



😊 بعد از رسم یک منحنی اگر بفواهیم منحنی بعدی روی همین محورها رسم شود کافی است بعد از رسم اولین منحنی از فرمان **hold on** استفاده کرد.

😊 هرگاه بفواهیم از این ویژگی دیگر استفاده نکنیم باید از فرمان **hold off** استفاده کرد.



😊 تابع زیر را تایپ کنید توابع مهم داده لیست می شوند. در ادامه این توابع بررسی خواهند شد.

help datafun



توضیح	تابع
ماکزیمم	max
مینیمم	min
متوسط	mean
میان	median
انحراف معیار	std
واریانس	var
مرتب کردن	sort



تابع	توضیح
sum	جمع عناصر
prod	ضرب عناصر
hist	هیستوگرام
cumsum	توزیع تجمعی
cumprod	توزیع تجمعی ضربی
diff	مشتق
corrcoef	ضریب همبستگی
cov	کوواریانس





چند مثال: 😊

$b = [5 \ 1 \ 2; 3 \ 9 \ 4; 7 \ 6 \ 8]$

$\text{index} = \text{find}(b == 6)$

$[r, c] = \text{find}(b == 6)$

$m = \max(b)$

$m = \max(\max(b))$

$[v, r] = \max(b)$



`min(b)`

`s = size(b)`

`d = b(2,:)`

`s = size(d)`

`l = length(d)`

`max(size(d))`



$n = \text{ndims}(b)$

$\text{length}(\text{size}(b))$

$p = \text{numel}(b)$

در MATLAB می‌توان بردارهای سه بعدی نیز  
تعریف کرد. 😊

$a = [5 \ 7 \ 8; 0 \ 1 \ 9; 4 \ 3 \ 6]$

$a(:, :, 2) = [1 \ 0 \ 4; 3 \ 5 \ 6; 9 \ 8 \ 7]$



$a(:, :, 3) = 5$

$r = \text{rand}(4, 3, 2)$

$r(4, 1, 2)$

$r([1\ 3\ 4], 2, 1)$

$r(3, :, 2)$

$s = \text{size}(r)$



😊 برای چسباندن دو ماتریس به هم و تشکیل ماتریس جدید از دستور **cat** استفاده می‌شود.

```
b = cat(3, [2,8; 0 5], [1 3; 7 9])
```

```
a = [1 2; 3 4]
```

```
b = [5 6; 7 8]
```

```
c = cat(1, a, b)
```

```
[a; b]
```



$c = \text{cat}(2, a, b)$   
 $[a, b]$

از  $\text{cat}(3, \dots)$  برای چسباندن ماتریس در بعد سوم استفاده می‌شود. با این کار می‌توان تصاویر رنگی ساخت.



😊 در این قسمت چندین مثال جهت یادآوری مطالب گذشته ارائه می شود.

$$a = 1 + 2i$$

$$b = \text{Real}(a)$$

$$c = \text{imag}(a)$$



`abs(a)`

`angle(a)`

`conj(a)`

`complex(2,3)`

`pow2(5)`

`nextpow2(13)`





`log10(10)`

`x = rand(1,10)`

`x(4:-1:2)`

`x = (0:0.1:1)*pi`

`linspace(0, pi, 11)`

`logspace(0, 2, 11)`

`linspace(0,10,11)`



😊 دستور **linespace** با سه پارامتر استفاده می شود. بین پارامتر اول و دوم به اندازه پارامتر سوم نقطه ایجاد می کند.

😊 دستور **logspace** با سه پارامتر استفاده می شود. بین لگاریتم پارامتر اول و لگاریتم پارامتر دوم به اندازه پارامتر سوم نقطه ایجاد می کند.

😊 پیش فرض تعداد نقاط **linespace** عدد ۱۰۰ و **logspace** عدد ۵۰ می باشد.



`linspace(0,50)`

`logspace(0,50)`

😊 جهت داشتن توضیحات از علامت % استفاده می شود.

`%a = 50`

😊 برای نمایش عدد به صورت نماد علمی به فرم زیر  
عمل می کند.

`x = 2e5`



```
b = [5:-1:1 3 8]
```

```
c = [b, 0]
```

```
d = [a(1:2:5) 1 0]
```

```
c, d
```

```
who
```

```
clear b c
```



$2^4$

$a = [1 \ 2 \ 3]$

$a.^2$

$a.^3$

$a = 2:3:8$

$\text{size}(a)$

$b = [a' \ a' \ a']$



size(b)

c = b(1:2:3, 1:2:3)

size(c)

d = a+b(2, :)

size(d)

w = [zeros(1, 3) ones(3,1)' 3:5']

size(w)



$$b([1, 3], 2) = b([3, 1], 2)$$

Size(b)

$$e = 1:-1:5$$



😊 برای تغییر فرمت نمایش اعداد از دستور `format` استفاده می‌شود.

`format short`

😊 اعداد را با `%` رقم اعشار نشان می‌دهد (به صورت پیش‌فرض).

`format long`

😊 اعداد را با `%14` رقم اعشار نشان می‌دهد.





format bank

😊 اعداد را به صورت ۲، رقم اعشار نشان می‌دهد.

format rat

😊 اعداد را به صورت کسری نشان می‌دهد.

pi

format bank

pi



pi

format long

pi

😊 دستور **format** نمایش تعداد اعشار را به حالت پیش فرض آن تبدیل می کند.



😊 نرم افزار Matlab دستوراتی درباره عملیات رابطه‌ای دارد. لیست این توابع در دستور زیر است که در ادامه این توابع بررسی می‌شوند.

help relop



توضیحات	تابع
بزرگ تر	$>$
کوچک تر	$<$
بزرگ تر یا مساوی	$>=$
کوچک تر یا مساوی	$<=$
مساوی	$=$
مخالف	$\sim$
اپراتور عطف	$\&$



توضیحات	تایع
عطف سریع	&&
اپراتور فصلی	
فصل سریع	
نقیض	~
یای انحصاری	xor



چند مثال 😊

```
tf = [30 40 50 60 70]>40
```

```
a = [2.5 6.7 9.2 inf 4.8 NaN];
```

```
b = isfinite(a)
```

```
c = islogical(a)
```

```
d = islogical(b)
```



```
x = -3:3
```

```
tf = logical(x)
```

```
x = randperm(12)
```

```
x = reshape(x,3,4)
```

```
tf = (x<=5)
```

```
whos tf, x
```



```
c = [true false]
```

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

```
b = rem(a,2) ~= 1
```

```
a(b)
```

```
clear
```

```
a = 0;
```

```
c = a&b
```





`c = a && b`

😊 در **Matlab** یک سری ساختارهایی مشابه سایر زبان‌های برنامه‌نویسی وجود دارد. لیست کامل این توابع در تابع زیر می‌باشد.

`help lang`



توضیحات	تابع
تابع شرطی	if
تابع شرطی	else
تابع شرطی	elseif
پایان شرط و حلقه	end
حلقه	for
حلقه	while
خروج از حلقه	break



```
if expression
    statements
end
```

```
if expression
    statements
else
    statements
end
```



if expression

statements

elseif expression

statements

else

statements

end



```
for variable = value1:value2
    statement
    ...
    statement
end

n=100;
for r = 1:n
    for c = 1:n
         $a(r,c) = 1/(r+c-1);$ 
    end
end
```



```
x = [];  
for i = 1:10  
x = [x, i^2];  
end
```



```
while expression  
    statements
```

```
end
```

```
switch switch_expr
```

```
    case case_expr
```

```
        statement, ..., statement
```

```
    case {case_expr1, case_expr2, case_expr3,...}
```

```
        statement, ..., statement
```

```
        otherwise
```

```
        statement, ..., statement
```

```
end
```



```
method = 'Bilinear';  
    switch lower(method)  
    case {'linear','bilinear'}  
        disp('Method is linear')  
    case 'cubic'  
        disp('Method is cubic')  
    case 'nearest'  
        disp('Method is nearest')  
    otherwise  
        disp('Unknown method.')  
    end
```





```
for n=1:10
```

```
x(n)=sin(n*pi/10)
```

```
end
```

```
n=1:10;
```

```
y=sin(n*pi/10);
```



نوع مناسبه  eps

```
num = 0;  
eps = 1;  
while (1+eps) > 1  
eps = eps/2;  
num = num+1;  
end  
eps = eps*2
```



$a=1; b=2; c=1;$

$\text{if}(b^2-4*a*c)<0$

$\text{disp('This equation has two complex root.')}$

$\text{elseif}(b^2-4*a*c)==0$

$\text{disp('This equation has two identical real roots.')}$

$\text{else}$

$\text{disp('This equation has two distinct real roots.')}$

$\text{end}$



```
eps = 1;  
for num = 1:1000  
    eps = eps/2;  
    if(1+eps) <= 1  
        eps = eps*2  
        break  
    end  
end
```



```
eps = 1;  
for num = 1:1000  
    eps = eps/2;  
    if(1+eps)>1  
        continue  
    end  
    eps = eps*2;  
    break  
end
```



دانشگاه یزد

$x = 2.7$

units = 'm'

switch units

case {'inch','in'}

$y = 2.54 * x$

case {'feet','ft'}

$y = 2.54 * x / 12$

case {'millimeter','mm'}

$y = x$

case {'meter','cm'}

$y = x / 100$

otherwise

disp('unkonwn Units!')

$y = \text{NaN}$

end



```
a = ones(4,2);  
b = 5*eye(3);  
try  
c = a*b;  
catch  
errmsg = lasterr;  
disp(errmsg)  
end
```



😊 تابع **lower** کاراکتر ورودی را به کاراکتر کوچک تبدیل می‌کند.

**lower('A')**

😊 تابع **upper** کاراکتر کوچک را به بزرگ تبدیل می‌کند.

**upper('a')**





😊 برای دیدن لیست کلمات کلیدی از فرمان زیر استفاده می‌شود.

`iskeyword`

😊 برای چک نمودن این که آیا یک دستور از کلمات کلیدی است یا خیر از فرمان زیر استفاده می‌شود.

`tf=iskeyword('while')`



😊 برای اندازه‌گیری مدت زمان اجرای یک برنامه از دو دستور **tic** و **toc** استفاده می‌شود.

**tic**

برنامه مورد نظر

**toc**



مثال: 😊

```
tic  
for i = 1:10^6  
sin(i);  
end  
toc
```



tic

$i = 1:10^6;$

sin(i);

toc

😊 ملاحظه می شود برنامه ی دوم زمان کمتری را مصرف نموده است. بنابراین حتی الامکان از حلقه **for** استفاده نکنید.



😊 برای استفاده از **help** این نرم افزار می توان به صورت زیر عمل کرد.

نام دستور **help**

**help plot**

😊 در این حالت توضیحات مفیدی در مورد این دستور داده می شود. اگر حجم اطلاعات زیاد باشد و شما بخواهید این اطلاعات را به صورت صفحه ای ببینید به صورت زیر عمل



more on

help plot

more off 😊 برای غیر فعال کردن این دستور از استفاده کنید.

😊 از دستور **help help** اطلاعاتی در مورد نحوه استفاده از **help** داده می شود.



😊 با استفاده از دستور **doc help** می‌توان از امکانات **help** استفاده‌های دیگری نمود.

**doc help**

**help doc**

😊 دستور اول از **help** به صورت **html** یک توضیحاتی می‌دهد و دستور دوم **help doc** را در پنجره کار نشان می‌دهد.



😊 از دستور **lookfor** می توان هر اطلاعات مفیدی را  
در مورد یک عمل خاص جستجو کرد.

## lookfor string

😊 در این دستور کلیه توابعی که با **string** سر و کار دارند  
را لیست می کند.





😊 انواع تصویر

grayscale images (gray level)

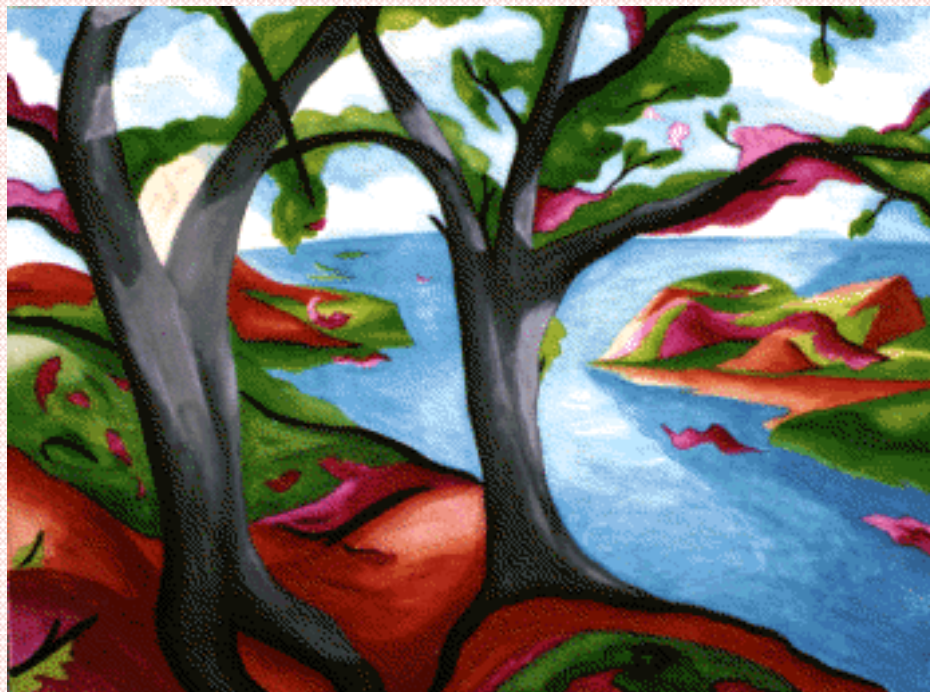
RGB images

index color images

BW images











😊 از دستور **imtool** میتوان با توجه به حرکت موس  
مختصات، روشنایی، رنگ، نوع تصویر را مشاهده کرد.

```
imtool('board.tif')
```

😊 یک تصویر را با ابزار حرکت موس نمایش می‌دهد

```
[X,map] = imread('trees.tif');
```

```
imtool(X,map)
```

😊 یک تصویر ایندکس دار را نمایش میدهد

```
I = imread('cameraman.tif');
```

```
imtool(I)
```

😊 می‌توان ابتدا تصویر را توسط **imread** خواند و سپس نمایش داد.

```
h = imtool(I,[0 80]);
```

```
close(h)
```

😊 پیکسل‌های یک تصویر را که در یک رنج مشخص قرار دارند نمایش می‌دهد و سپس با استفاده از دستور **close** آن پنجره را می‌بندد.





😊 با استفاده از دستور **impixel** می‌توان با دادن مختصات یک تصویر، رنگ قرمز، سبز، آبی تصویر را استخراج کرد.

```
RGB = imread('peppers.png');  
c = [12 146 410];  
r = [104 156 129];  
pixels = impixel(RGB,c,r)
```

😊 در صورتی که آرگومان این تابع یک تصویر سطوح خاکستری باشد قسمت‌های قرمز، سبز، آبی مساوی خواهد



😊 با استفاده از دستور **imfinfo** می‌توان اطلاعاتی در مورد یک تصویر به دست آورد.

**imfinfo('cameraman.tif')**

😊 این اطلاعات در مورد نوع تصویر، طول تصویر، عرض تصویر، عمق بیت، ساینز و ... را مشخص می‌کند. این اطلاعات از هدر فایل تصویر خوانده می‌شود.





😊 انواع داده‌ها

Int8 یک عدد ۸ بیتی بین منفی ۱۲۸ تا ۱۲۷

Unit8 یک عدد ۸ بیتی بین ۰ تا ۲۵۵

Int16 یک عدد ۱۶ بیتی (مثبت و منفی)

UInt16 یک عدد ۱۶ بیتی مثبت

Int32 یک عدد ۳۲ بیتی (مثبت و منفی)

UInt32 یک عدد ۳۲ بیتی مثبت



# double یک عدد حقیقی