

به نام خدا

آموزش متلب

مدرس:

دکتر فریده قریشی، زهرا مهدور

راه‌های ارتباطی:

Email : zahra.mahdevar@yahoo.com

Telegram : Z_Mhdr

جلسه اول: آشنایی با محیط نرم‌افزار متلب و مفاهیم اولیه

مقدمه‌ای بر متلب

متلب که مخفف عبارت آزمایشگاه ماتریسی است، یک نرم‌افزار قدرتمند و محیط برنامه‌نویسی سطح بالا است که به طور ویژه برای انجام محاسبات عددی، تحلیل داده، طراحی الگوریتم‌ها، شبیه‌سازی و رسم نمودارهای علمی و مهندسی طراحی شده است. این نرم‌افزار به دلیل قابلیت‌های منحصر به فردش در رشته‌های مهندسی، ریاضیات، فیزیک، اقتصاد و علوم داده مورد استفاده گسترده قرار می‌گیرد.

Mat + Lab --> Matrix + Laboratory

بخش اول: لینک‌های مورد نیاز

لینک لیست جعبه ابزارهای متلب:

https://www.mathworks.com/products.html?s_tid=nav_products

لینک دانلود نرم‌افزار متلب:

<https://soft98.ir/software/engineering/1291-%D8%AF%D8%A7%D9%86%D9%84%D9%88%D8%AF-%D9%85%D8%AA%D9%84%D8%A8-%DA%86%D9%87%D8%A7%D8%B1-1e.html>

لینک دانلود مستندات متلب:

<https://downloadlynet.ir/2020/26/9256/03/matlab-2/16/?#/9256-matlab-r-212511045909.html>

لینک راهنمای نصب مستندات متلب:

<https://vdelta.ir/post/4>

بخش دوم: مزایا و معایب

مزایا:

--> زبان سطح بالا و ساده

--> مستقل از سیستم عامل

--> کتابخانه‌ها و جعبه‌ابزارهای گسترده

--> قابلیت ترسیم نمودارهای پیشرفته

--> سازگاری با زبان‌های دیگر

معایب:

--> نرم‌افزار غیررایگان و گران‌قیمت

--> سرعت کمتر نسبت به زبان‌های سطح پایین

--> پشتیبانی محدود از کار با وب و موبایل

--> حجم بالای نرم‌افزار و نیاز به سخت‌افزار قوی

نکته: متلب هم یک نرم‌افزار است و هم یک زبان برنامه‌نویسی. در واقع می‌توان آن را ترکیبی از این دو دانست

بخش سوم: معرفی بخش‌های مهم محیط متلب

Command Window

پنجره فرمان یکی از بخش‌های اصلی محیط متلب است که به کاربر امکان می‌دهد تا دستورات را به‌صورت تعاملی و فوری اجرا کند. این پنجره مانند یک ماشین حساب پیشرفته عمل می‌کند و برای انجام محاسبات سریع، تست توابع، مشاهده خروجی‌ها و اجرای کدهای کوتاه و آزمایشی بسیار مناسب است



ویژگی‌ها

--> اجرای سریع دستورات

--> مناسب برای محاسبات فوری و آزمایش‌ها

--> نمایش فوری خروجی

--> تعامل مستقیم با محیط متلب

معایب

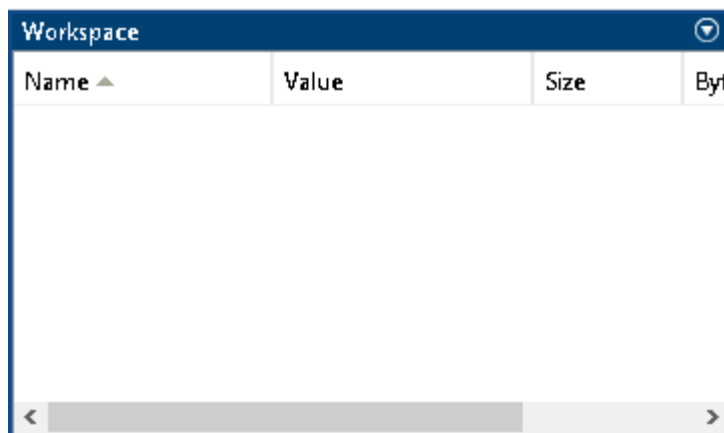
--> عدم ذخیره‌سازی دائمی دستورات

--> نامناسب برای پروژه‌های بزرگ یا پیچیده

--> اشکال در مدیریت خطاهای پیچیده

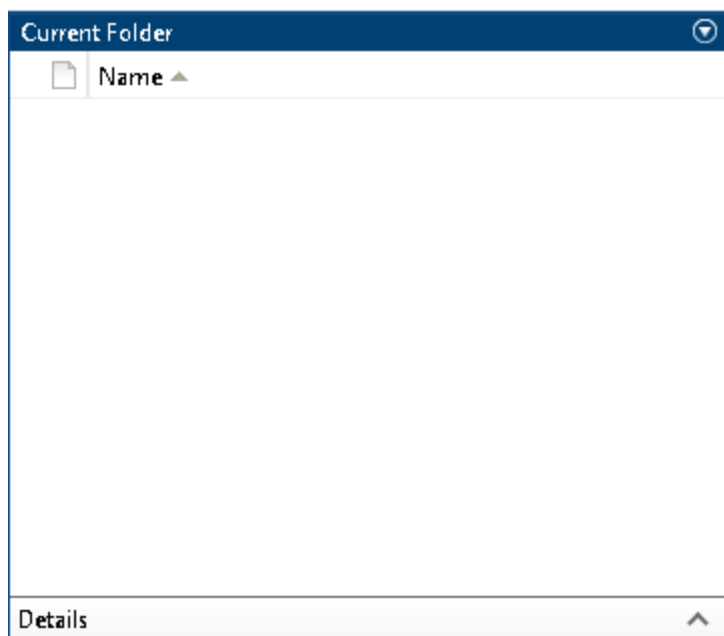
Workspace

پنجره‌ی متغیرها یکی از بخش‌های کلیدی محیط متلب است که تمام متغیرهای فعال و اطلاعات مربوط به آن‌ها (مانند اندازه، نوع، مقدار و ...) را نمایش می‌دهد. این پنجره به کاربران امکان می‌دهد تا در حین اجرای برنامه، متغیرها را مشاهده، بررسی و حتی تغییر دهند.



Current Folder

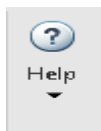
پنجره پوشه جاری در متلب محل نمایش فایل‌ها و پوشه‌های موجود در مسیر جاری کاری است. این پنجره به کاربران امکان می‌دهد به راحتی فایل‌های مورد نیاز یا داده‌ها را مشاهده، اجرا، ویرایش یا حذف کند. همچنین می‌توان از طریق این پنجره مسیر کاری را تغییر داد و فایل‌های پروژه را بهتر مدیریت کرد. این بخش برای سازمان‌دهی و دسترسی سریع به فایل‌های مرتبط با پروژه، بسیار کاربردی و مهم است.



Help

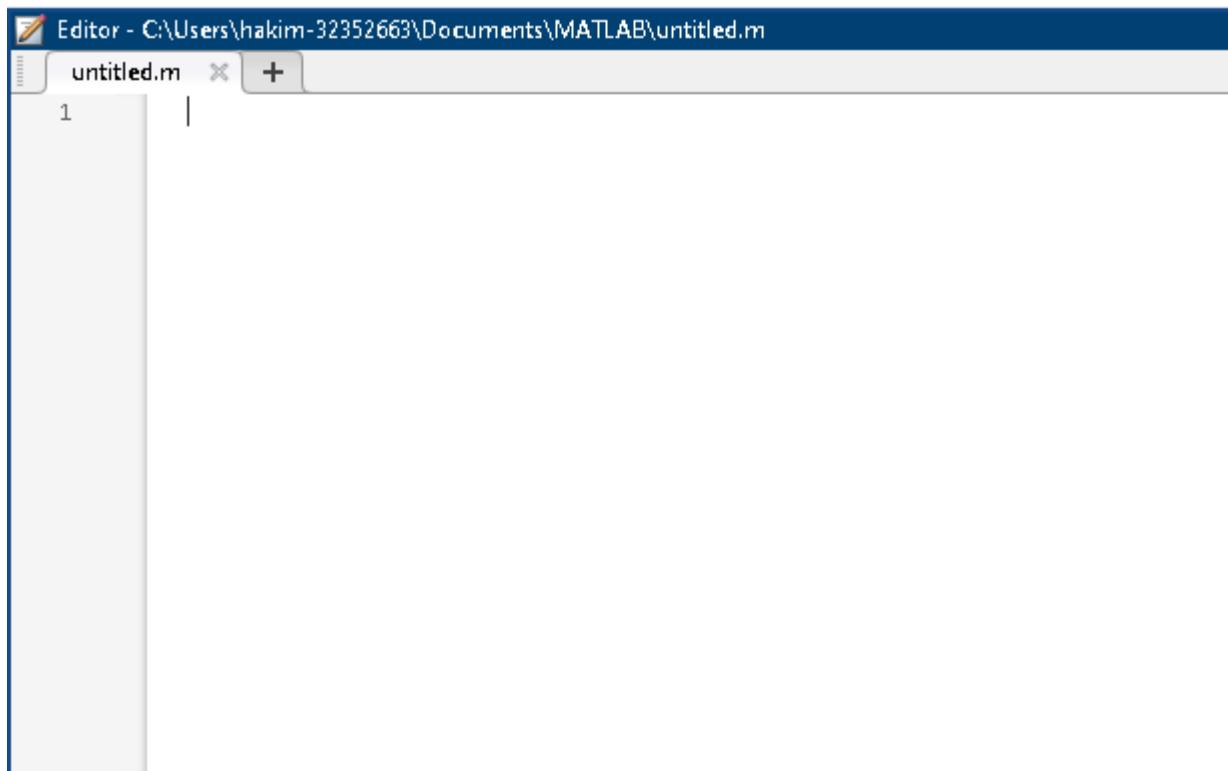
راهنمای متلب یکی از ابزارهای کاربردی این نرم‌افزار است که به کاربران امکان می‌دهد مستندات، راهنماها و اطلاعات جامعی در مورد دستورات، توابع و قابلیت‌های مختلف متلب را مشاهده و بررسی کنند. از طریق این پنجره، کاربران می‌توانند به توضیحات دقیق، نمونه کدها و نکات کلیدی مربوط به هر دستور دسترسی داشته باشند و در صورت نیاز، اطلاعات بیشتری درباره نحوه عملکرد و کاربرد توابع مختلف به‌دست آورند. شایان ذکر است که

از نسخه ۲۰۲۰ به بعد، این راهنما به صورت آنلاین ارائه می‌شود و دیگر همراه با نصب نرم‌افزار در حالت آفلاین در دسترس نیست. برای دسترسی به مستندات راهنما، نیاز به دانلود و نصب جداگانه آن است



Editor

صفحه ویرایشگر در متلب محیطی قدرتمند برای نوشتن، ویرایش و اجرای کدهای اسکریپت و توابع است. این پنجره با امکاناتی مانند رنگ‌آمیزی خودکار دستورها، تکمیل هوشمند کد، نمایش لحظه‌ای خطاها و قابلیت اجرای مرحله‌به‌مرحله کد تجربه‌ی برنامه‌نویسی را ساده‌تر و حرفه‌ای‌تر می‌کند. همچنین ابزارهای اشکال‌زدایی تعبیه‌شده در آن به کاربران کمک می‌کند تا خطاهای احتمالی را به راحتی شناسایی و اصلاح کنند

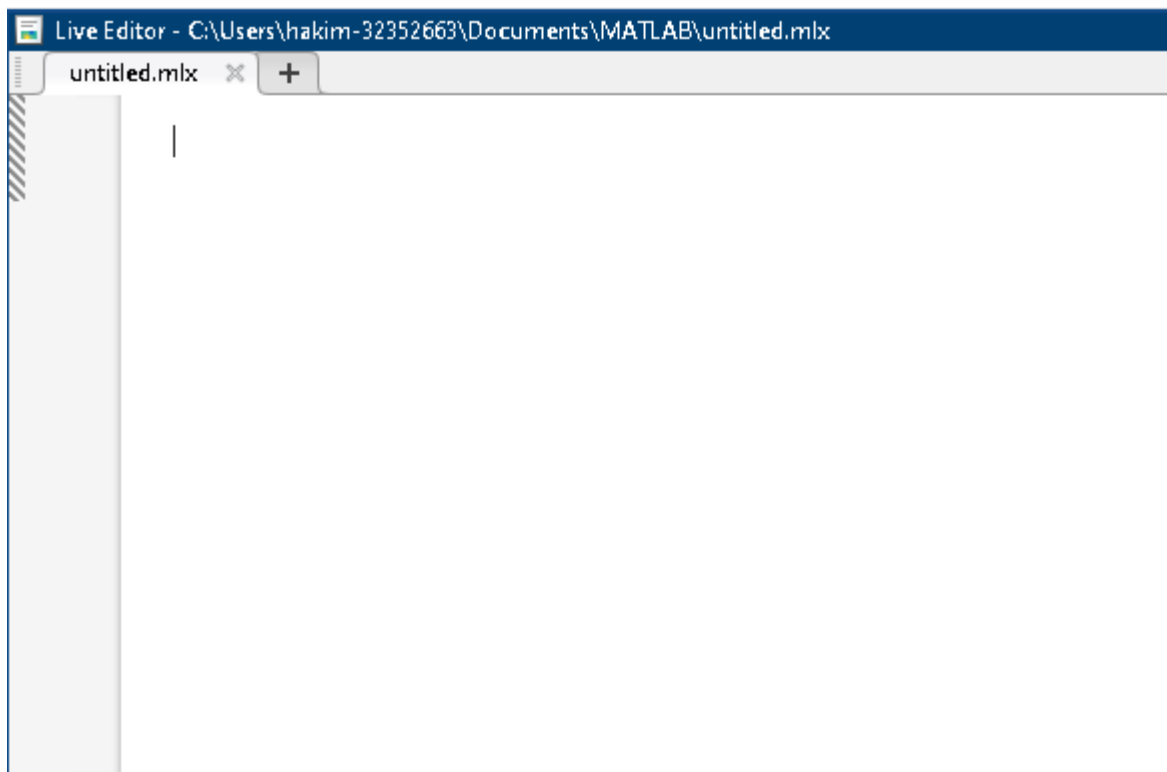


نکته: برای اجرا برنامه در ویرایشگر از منوی بالای ویرایشگر روی دکمه زیر کلیک کنید یا در پنجره فرمان نام اسکریپت را بدون پسوند وارد کنید



Live Editor

صفحه ویرایشگر تعاملی در متلب یک محیط پیشرفته و تعاملی برای نوشتن، اجرای کد و مستندسازی هم‌زمان است. این صفحه به کاربران اجازه می‌دهد که کدهای خود را همراه با توضیحات متنی، فرمول‌های ریاضی با قالب لتکس و نمودارهای تعاملی در یک محیط یکپارچه بنویسند. از جمله ویژگی‌های شاخص آن می‌توان به نمایش نتایج در کنار کد، قالب‌بندی حرفه‌ای اسناد، امکان درج تصاویر، جدول‌ها و عناوین و قابلیت ذخیره فایل‌ها با فرمت‌های مختلف برای ارائه یا اشتراک‌گذاری اشاره کرد



نکته: برای اجرای برنامه در ویرایشگر تعاملی، از منوی بالای ویرایشگر روی یکی از دکمه‌های زیر برای اجرای کل فایل یا بخش مورد نظر استفاده کنید



Command History

پنجره تاریخچه‌ی دستورات در متلب، تاریخچه‌ای از تمام دستورات اجرا شده در محیط پنجره فرمان را نمایش می‌دهد. این پنجره به کاربران اجازه می‌دهد تا دستورات قبلی را مرور کرده، آن‌ها را دوباره اجرا کنند یا ویرایش کرده و به‌سادگی در کدهای جدید استفاده نمایند. این ویژگی به‌ویژه هنگام انجام آزمایش‌های مکرر یا تست توابع بسیار مفید است، زیرا نیاز به تایپ مجدد دستورات کاهش می‌یابد.



نوار ابزار

نوار ابزار در محیط متلب یکی از بخش‌های کلیدی رابط کاربری گرافیکی است که امکان دسترسی سریع به دستورات پرکاربرد، تنظیمات و قابلیت‌های متلب را فراهم می‌سازد. شناخت نوار ابزار برای کار روان‌تر و سریع‌تر با این نرم‌افزار ضروری است

HOME	PLOTS	APPS	EDITOR	PUBLISH	VIEW
------	-------	------	--------	---------	------

HOME	PLOTS	APPS	LIVE EDITOR	INSERT	VIEW
------	-------	------	-------------	--------	------

HOME

New Script \ Live Script: ایجاد یک فایل جدید

Open: باز کردن فایل‌ها

Clear Workspace: پاک کردن همه متغیرها

Simulink: محیط گرافیکی برای مدل‌سازی، شبیه‌سازی و تحلیل سیستم‌های دینامیکی و کنترلی

Preferences: تغییر تنظیمات متلب

Layout: تنظیم چیدمان پنجره‌ها

...

PLOTS

plot: رسم نمودار

bar: رسم نمودار میله‌ای

scatter: رسم نمودار پراکندگی

histogram: رسم هیستوگرام

surf / contour: رسم نمودارهای سه بعدی

...

APPS

در این بخش برنامه‌های کاربردی متلب نمایش داده می‌شود که برای تحلیل داده‌ها، پردازش تصویر، یادگیری ماشین و ... استفاده می‌شوند

Optimization: حل مسائل بهینه‌سازی

Signal Analyzer: تحلیل سیگنال

...

EDITOR

Run: اجرای اسکریپت

Find: جستجو متن در کد

Indent: مرتب‌سازی خودکار کد

Comment / Uncomment: اضافه یا حذف کردن کامنت

...

INSERT

Text / Equation: نوشتن توضیحات و فرمول‌ها

Image / Hyperlink: درج تصویر یا لینک

Code / Section Break: افزودن کد یا تقسیم‌بندی

...

PUBLISH: تبدیل کردن کدها به فایل‌های گزارش

Publish: تبدیل اسکریپت به پی‌دی‌اف، ورد و غیره

VIEW: مدیریت نحوه نمایش بخش‌های مختلف مطلب

بخش چهارم: آشنایی با دستورات پایه‌ای مطلب، متغیرها، توابع پایه‌ای و قالب‌بندی خروجی

برخی دستورات پرکاربرد

clear	% حذف همه متغیرها از فضای کاری یا پنجره متغیرها
clc	% پاک کردن پنجره فرمان
close all	% بستن تمام پنجره‌های نمودار
who	% نمایش لیست متغیرها
whos	% نمایش لیست متغیرها با جزئیات (نوع، اندازه، حافظه)
which	% نمایش مسیر فایل یا تابع مورد استفاده توسط مطلب
class	% نمایش نوع داده‌ی یک متغیر

```
which('exp')
```

```
built-in (C:\Program Files\MATLAB\R2024a\toolbox\matlab\elfun\@double\exp) % double  
method
```

```
class(pi)
```

```
ans = 'double'
```

چند دستور برای استفاده از راهنمای مطلب

doc	% باز کردن راهنمای کامل و تصویری تابع یا دستور موردنظر در پنجره
help	% نمایش توضیحات کوتاه متنی درباره‌ی یک دستور در پنجره فرمان
helpwin	% مشاهده دستور فوق اما در پنجره‌ای جداگانه و قابل جستجو
lookfor	% جستجوی توابع مرتبط با یک واژه کلیدی در توضیحات قسمت راهنما

برخی متغیرهای پیش‌فرض در مطلب

ans	% نتیجه آخرین دستور بدون نام‌گذاری متغیر
pi	% عدد پی (نسبت محیط دایره به قطر آن)، تقریباً 3.1416
i, j	% واحد موهومی، برای محاسبات مختلط
inf	% بی‌نهایت
NaN	% عدد تعریف نشده

eps % کوچکترین عدد مثبت قابل تشخیص توسط متلب، نشان‌دهنده دقت محاسبات
realmin, realmax % کوچکترین و بزرگترین عدد اعشاری قابل نمایش در متلب

```
realmin  
realmax
```

```
ans = 2.2251e-308
```

```
ans = 1.7977e+308
```

```
inf / inf
```

```
ans = NaN
```

```
2 * realmax
```

```
ans = Inf
```

```
i
```

```
ans = 0.0000 + 1.0000i
```

تعریف متغیرها

قوانین نام‌گذاری متغیرها

--> باید با حروف شروع شود

--> حساس به حروف بزرگ و کوچک

--> نباید از نام توابع داخلی متلب استفاده شود

--> متلب حداکثر 63 کاراکتر را برای نام متغیرها پشتیبانی می‌کند

--> حروف مجاز در نام متغیرها: حروف انگلیسی، اعداد، علامت زیرخط

نکته: می‌توانید از تابع زیر برای بررسی معتبر بودن نام متغیر استفاده کنید

```
isvarname('نام متغیر')
```

```
isvarname('a_1')
```

```
ans = logical
```

```
1
```

```
isvarname('2_b')
```

```
ans = logical
```



```
isvarname('a*')
```

```
ans = logical
```

نکته: در محیط متلب، نیازی به تعریف نوع داده به صورت صریح وجود ندارد؛ کفایت مقداری به یک متغیر اختصاص داده شود و متلب به طور خودکار نوع داده‌ی مناسب را تشخیص می‌دهد

```
a = 2;
b = 5.2;
c = a+b      % استفاده نکنید، نتیجه در پنجره فرمان چاپ می‌شود
class(c)
```

```
c = 7.2000
```

```
ans = 'double'
```

عملگرهای ریاضی پایه‌ای

```
a = 2;
b = 4;
c = a + b      % جمع
d = c - b      % تفریق
e = d * a      % ضرب
f = e / c      % تقسیم از راست
g = a \ e      % تقسیم از چپ
h = a ^ f      % توان
```

برخی از توابع ریاضی پایه‌ای

```
x = 45;
sqrt(x)        % ریشه دوم
abs(x)         % قدر مطلق

exp(x)         % تابع نمایی
log(x)         % (پایه e) لگاریتم طبیعی
log10(x)       % لگاریتم پایه 10
log(x) / log(b); % Logarithm to base b

sin(x)         % بر حسب رادیان
cos(x)
tan(x)
cot(x)

sind(x);       % بر حسب درجه
cosd(x);
tand(x);
cotd(x);
```

```
deg2rad(x);      % تبدیل درجه به رادیان
asin(x);         % arcsin (Inverse sine)
acos(x);         % arccos
atan(x);         % arctan
acot(x);         % arccot
```

نمایش خروجی

در مطلب، برای نمایش مقدار متغیرها یا پیامها در پنجره‌ی فرمان، می‌توان از دستورهای مختلفی استفاده کرد. دو دستور رایج و پرکاربرد برای این کار عبارت‌اند از

1. disp

این دستور برای نمایش ساده‌ی مقدار یک متغیر یا متن در پنجره‌ی فرمان استفاده می‌شود. این روش برای مواقعی مناسب است که فقط می‌خواهید مقدار را بدون هیچ قالب‌بندی خاصی نشان دهید

```
x = 30;
disp(x)      % چاپ مقدار متغیر
```

30

2. fprintf

این دستور برای نمایش متن با قالب‌بندی دلخواه استفاده می‌شود که برای چاپ دقیق‌تر، مانند مشخص کردن تعداد اعشار، تنظیم فاصله، افزودن متن همراه با مقدار متغیر و... کاربرد دارد

```
x = 20;
fprintf('x = %.2f\n', x) % عدد اعشاری با دو رقم اعشار چاپ شود
```

x = 20.00

چاپ هم‌زمان متن و مقدار متغیر

```
name = 'zahra';
score = 18.95;
fprintf('Student %s got %.2f out of 20\n', name, score)
```

Student zahra got 18.95 out of 20

پرکاربردترین کدهای قالب‌بندی

```
Code = {'%d'; '%f'; '%.2f'; '%e'; '%s'; '\n'; '\t'; '%'};
Explanation = {'عدد صحیح'; 'عدد اعشاری'; 'عدد اعشاری با دو رقم اعشار'; 'رشته متنی'; 'نمایش علمی'; 'رشته متنی'; 'چاپ علامت درصد'; '(Tab) رفتن به خط جدید'; 'فاصله تب'};

T = table(Code, Explanation, 'VariableNames', {'کد قالب', 'توضیح'});
disp(T)
```

کد قالب

توضیح

{ '%d' }	{ 'عدد صحیح' }
{ '%f' }	{ 'عدد اعشاری' }
{ '%.2f' }	{ 'عدد اعشاری با دو رقم اعشار' }
{ '%e' }	{ 'نمایش علمی' }
{ '%s' }	{ 'رشته متنی' }
{ '\n' }	{ 'رفتن به خط جدید' }
{ '\t' }	{ 'فاصله تب (Tab)' }
{ '%%' }	{ 'چاپ علامت درصد' }

کنترل دقت نمایش

```
format short % نمایش اعداد با 4 رقم اعشار (پیشفرض متلب)
format long % نمایش اعداد با 15 رقم اعشار
format bank % فقط دو رقم اعشار
format short e % نمایش اعداد به صورت نمایی با 4 رقم اعشار
format long e % نمایش اعداد به صورت نمایی با 15 رقم اعشار
```

نکته: این دستور فقط نحوه‌ی نمایش را تغییر می‌دهد و بر مقدار واقعی عدد تأثیری ندارد

```
x = pi;

format short
disp(x)
```

3.1416

```
format long
disp(x)
```

3.141592653589793

```
format bank
disp(x)
```

3.14

```
format long e
disp(x)
```

3.141592653589793e+00

ادامه دستور در چند خط

```
x1 = 1 + 1/3 + ...
      + 4/3
```

practice 1. Write a code that computes the following expression for $x=2$ and prints the result using `fprintf`

$$f(x) = \sqrt{x^2 + \log(x) + 2}$$

بخش چهارم: بردارها و ماتریس‌ها

```
v1 = [1 2 3 4];      % بردار سطری  
v2 = [1; 2; 3; 4];   % بردار ستونی
```

در متلب، عملگر : به معنی تولید یک بازه‌ی عددی با گام مشخص است

```
v3 = 1:5  
v4 = 0:2:10    % مشخص کردن گام حرکت 2
```

v3 = 1×5

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

v4 = 1×6

0	2	4	6	8	10
---	---	---	---	---	----

تعریف یک ماتریس 3 در 3

```
A = [1 2 3; 4 5 6; 7 8 9];  
A = [1, 2, 3; 4, 5, 6; 7, 8, 9]; % می‌توان به این صورت هم ماتریس را تعریف کرد
```

دسترسی به عناصر ماتریس

برای ایجاد ماتریس از آکولاد استفاده می‌کنیم و برای فراخوانی ماتریس و دسترسی به عناصر آن از پرانتز استفاده می‌کنیم

```
A = [1 2 3 4; 5 6 7 8; 9 10 11 12; 13 14 15 16];  
A(2,4)           % عنصر سطر دوم، ستون چهارم  
A(2,2:3)         % سطر دوم، ستون‌های دوم تا سوم  
A(3,2:4)%==A(3,2:end) % سطر سوم، ستون‌های دوم تا آخر  
A(:,1)          % تمام سطرها، ستون اول  
A(1,:)          % سطر اول، تمام ستون‌ها  
A(:, :)%==A  
A(4)            % عنصر چهارم در حالت شمارش ستونی  
A(2:6)          % عناصر دوم تا ششم در حالت شمارش ستونی
```

جایگزینی یک مقدار

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

A = 4×4

1	2	3	4
5	6	7	8
9	10	11	12

13 14 15 16

$$A(3,2) = 100$$

A = 4×4

1	2	3	4
5	6	7	8
9	100	11	12
13	14	15	16

$$A(3,5) = 20$$

A = 4×5

1	2	3	4	0
5	6	7	8	0
9	100	11	12	20
13	14	15	16	0

عملگرهای ماتریسی

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

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

```
C1 = A * B;      % ضرب ماتریسی: ستون‌های ماتریس اول = سطرهای ماتریس دوم
```

```
C2 = A .* B;      % ضرب عنصر به عنصر: هر دو ماتریس باید هم اندازه باشند
```

```
C3 = A ./ B;      % تقسیم عنصر به عنصر
```

```
C4 = A .^ B;      % توان عنصر به عنصر
```

توابع پرکاربرد مربوط به بردار و ماتریس

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

```
transpose(A)%= A'      % ترانپوز ماتریس
```

```
size(A)      % ابعاد (سطر و ستون) ماتریس
```

```
sum(A)      % مجموع عناصر ماتریس
```

```
mean(A)      % میانگین عناصر ماتریس
```

```
inv(A)      % وارون ماتریس (در صورتی که ماتریس مربعی و وارون‌پذیر باشد)
```

```
max(A)      % بزرگترین مقدار ماتریس
```

```
min(A)      % کوچکترین مقدار ماتریس
```

```
length(A)      % بیشترین مقدار بین تعداد سطرها و ستونها
```

```
numel(A)      % تعداد کل عناصر موجود در ماتریس
```

```
ndims(A)      % ابعاد ماتریس
```

```
repmat(A, 2, 1)      % برای تکرار یک ماتریس یا آرایه
```

isempty(A)	% بررسی می‌کند که آیا ماتریس خالی است یا نه
isvector(A)	% بررسی می‌کند که آیا ماتریس یک بردار است یا خیر

ماتریس‌های خاص

zeros(3,4)	% ماتریس صفر 3 در 4
ones(2,2)	% ماتریس تمام یک
eye(4)	% ماتریس همانی 4 در 4
rand(2,3)	% ماتریس تصادفی با اعداد در بازه 0 و 1
randn(3,3)	% ماتریس تصادفی با اعداد تصادفی نرمال
diag(2)	% ماتریس قطری
triu(A)	% برای استخراج بخش بالایی (بالای قطر) یک ماتریس
tril(A)	% برای استخراج بخش پایینی (پایین قطر) یک ماتریس
magic(3)	% ماتریس جادویی 3 در 3

```
A = [1 2 3; 4 5 6; 7 8 9];
U = triu(A)
```

U = 3×3

1	2	3
0	5	6
0	0	9

```
magic(4)
```

ans = 4×4

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

Practice 2. Create a **6×6 matrix** that satisfies the following conditions:

1. The elements of the **main diagonal** should be 5.
2. The elements **above and below the main diagonal** (i.e., ± 1 from the main diagonal) should be -1.
3. All **other elements** should be 0.

Then, calculate the **inverse** of this matrix (if it is invertible) and check that the **product of the matrix and its inverse** results in the **identity matrix**.

Practice 3. Create a **7×7 matrix** that satisfies the following conditions:

1. The **main diagonal** must contain the number 3.
2. The **anti-diagonal** (from top-right to bottom-left) must contain the number 4.
3. All elements in **even-numbered rows** (i.e., rows 2, 4, and 6) must be 1.
4. The **third column** must contain values from 7 to 1 in descending order.
5. All **other elements** should be 0.

Finally, calculate the **sum of all elements** in the matrix.