



به نام خدا



آموزش مقدماتی متلب - جلسه دوم

مدرس: دکتر احسان فتحی

مدیر و موسس آموزشگاه فنی و حرفه ای فتحی

Telegram & Instagram: @FathiTrainingGroup

Website: FathiTrainingGroup.com

Email: ehsanfathi_eh@yahoo.com

Tel: 09386249330 - 05191012910

بخش اول: آشنایی با اپراتورها

□ تعریف اعداد مختلف در متلب

1, -1, 0.25, 1+i, 1-j

□ انواع اپراتورها در متلب

+, -, *, /, \, ^, ' ,

4+8, 4-8, 4*8, 4/8, 4\8 or 8/4, 4^2, 2e3

□ محدودیت برای تعداد کاراکترهای یک متغیر ۶۳ کاراکتر می باشد

a=namelengthmax

✓ نرم افزار متلب به حروف کوچک و بزرگ حساس است.

□ قدر مطلق

abs(-6)=6



بخش اول: آشنایی با اپراتورها

بزرگترین و کوچکترین عدد حقیقی در متلب ☐

`realmax , realmin`

✓ مقادیر بزرگتر و کوچکتر از دو عدد فوق به ترتیب به عنوان مثبت بی نهایت و منفی بی نهایت در نظر گرفته می شود.

متغیرهای تعریف شده در متلب ☐

`i , j , eps , pi , inf , NaN`

می توان به متغیرهای بالا مقدار داد: ☐

`eps=2`

`Clear eps`

بخش دوم: آشنایی با انواع داده ها در متلب

□ انواع داده ها در متلب:

- ۱- داده های عددی شامل اعداد مختلط، صحیح، اعشاری و اصم می باشد.
- ۲- داده های منطقی که برای درستی یا نادرسی یک شرط به کار می روند.
- ۳- داده های رشته ای که برای نمایش یک متن به کار می روند.

□ به کمک دستور های زیر می توان اعداد صحیح و طبیعی در بازه های مختلف تعریف کرد:

`int8 , int16, int32, int64`

`uint8, uint16, uint32, uint64`

`x=int32(4)`

✓ برای بررسی نوع متغیر می توان از دستور `whos` استفاده کرد:

`whos x`

✓ به کمک دستور `isinteger` می توان صحیح بودن یک متغیر را تشخیص داد.

بخش دوم: آشنایی با انواع داده ها در متلب

□ در متلب اعداد اعشاری با دو دقت single و Double ذخیره می شوند. این اعداد به ترتیب ۳۲ بیت و ۶۴ بیت حافظه اشغال می کنند. لازم به ذکر است به صورت پیش فرض یک عدد با دقت Double در متلب ذخیره می شود.

```
x=25.125
```

```
whos x
```

```
y=single(x)
```

```
whos(y)
```

✓ به کمک دستور isfloat می توان اعشاری بودن یک متغیر را تشخیص داد.

```
isfloat(x)
```



بخش دوم: آشنایی با انواع داده ها در متلب

□ روش های مختلف نمایش اعداد در متلب:

`format short, format long, format short e, format long e, format rat , format hex`

□ به کمک دستور زیر می توان فرمت جاری برای نمایش اعداد را تشخیص داد:

`get(0,'format')`

□ علاوه بر روش ها فوق با استفاده از گزینه Preferences می توان فرمت جاری را برای نمایش اعداد تغییر داد.

بخش دوم: آشنایی با انواع داده ها در متلب

□ توابع مختلط دارای دو بخش حقیقی و موهومی می باشد.

```
x=2+3i
```

```
x=2+3j
```

```
x=complex(2,3)
```

```
whos x
```

✓ به کمک دستور `real` و `imag` می توان بخش حقیقی و موهومی یک تابع را مشخص کرد.

```
a=real(x)
```

```
b=imag(x)
```

✓ به کمک دستور `isreal` می توان حقیقی یا موهومی بودن یک عدد را تشخیص داد.

```
isreal(a)
```



بخش دوم: آشنایی با انواع داده ها در متلب

□ برای ایجاد داده منطقی به صورت زیر عمل می کنیم:

```
x=[true false true]
```

```
whos x
```

```
class(x)
```

```
islogical(x)
```

□ برای ایجاد متغیر رشته ای یا کاراکتر متن نوشته شده را داخل کتیشن قرار می دهیم:

```
x='ehsan fathi'
```

```
whos x
```

```
class(x)
```

```
ischar(x)
```

✓ هر کاراکتر دو بایت فضا اشغال می کند.

✓ برای تبدیل یک متغیر کاراکتری به آرایه ای از اعداد از تابع double استفاده می شود:

```
y=double(x)
```



بخش دوم: آشنایی با انواع داده ها در متلب

□ برای مقایسه متغیرهای عددی به صورت زیر عمل می کنیم. حاصل به صورت یک عدد منطقی نشان داده می شود:

$5 < 13$

$16 \leq 14$

$a = [1 \ 4 \ 5; 5 \ 6 \ 7; 9 \ 5 \ 10]$

$a < 5$

$14 \sim 15$

$14 == 14$

□ عملگرهای منطقی

$\& = \text{And}$

$| = \text{Or}$

$x = 5$

$-1 < x \ \& \ x < 15$



بخش دوم: آشنایی با انواع داده ها در متلب

□ برای مقایسه دو متغیر رشته ای از یکی از دستورهای strcmp، strcmpi و strncmpi استفاده می شود. اگر دو متغیر با یک دیگر برابر باشند حاصل با عدد یک نشان داده می شود.

```
x1='ehsan'
```

```
x2='Ehsan'
```

```
strcmp(x1,x2)
```

```
strcmpi(x1,x2)
```

```
strncmpi(x1,x2,2)
```

□ برای مقایسه نظیر به نظیر کاراکترها به صورت زیر عمل می شود:

```
a='fall'
```

```
b='fool'
```

```
a==b
```

بخش دوم: آشنایی با انواع داده ها در متلب

□ به کمک دستورهای `isspace` و `isletter` می توان تشخیص داد کاراکترهای داخل یک متغیر رشته ای حرف هستند یا فضای خالی می باشند.

```
x='ehsan 123'
```

```
isletter(x)
```

```
isspace(x)
```

□ به کمک دستور `strrep` می توان یک کاراکتر را با کاراکتر دیگر جایگزین کرد:

```
x='date: 2018/08/22'
```

```
strrep(x,'22','19')
```

□ برای پیدا کردن محل یک کاراکتر از دستور `findstr` استفاده می شود:

```
findstr('a',x)
```

□ برای تبدیل اعداد یک آرایه به کاراکتر از تابع `char` استفاده می شود:

```
x=[77 65 84 76 65 66]
```

```
char(x)
```



بخش دوم: آشنایی با انواع داده ها در متلب

□ برای تبدیل یک عدد صحیح یا اعشاری به متغیر رشته ای از دستورهایی زیر استفاده می شود:

`int2str(4)`

`num2str(2.25)`

□ برای تبدیل یک متغیر رشته ای که با تعداد عدد تعریف شده است به عدد از دستور زیر استفاده می شود:

`str2num('426')`

بخش دوم: آشنایی با انواع داده ها در متلب

□ برای نمایش تاریخ و زمان از دستورهایی زیر استفاده می شود:

`date - now - clock`

✓ دستور `date` تاریخ را به صورت یک رشته (سال، ماه و روز) نشان می دهد. دستور `Now` تاریخ را در غالب یک عدد نمایش می دهد و دستور `clock` تاریخ را در غالب یک بردار نمایش می دهد.

□ تابع `datenum` رشته تاریخ را به صورت یک عدد نمایش می دهد:

`a1=datenum('20-nov-2018')`

□ تابع `datestr` عدد تاریخ را به صورت یک رشته نمایش می دهد:

`a2=datestr(a1+10)`

✓ با دستور زیر می توان ترتیب نمایش تاریخ را تغییر داد:

`datestr(a2,2)`

تابع `datevec` عدد یا رشته تاریخ را به صورت یک بردار نمایش می دهد:

`datevec(a2)`



بخش سوم: عملیات جبری بر روی ماتریس ها

□ برای محاسبه اندازه یک بردار از دستور norm استفاده می شود:

`norm(a)`

□ برای محاسبه بردار یکه به صورت زیر عمل می شود:

`e=a/norm(a)`

□ برای محاسبه زاویه بین دو بردار به صورت زیر عمل می شود:

`a=[1 2 3]`

`b=[4 5 6]`

`acos(dot(a,b)/(norm(a)*norm(b)))`

□ برای محاسبه تصویر یک بردار روی برداری دیگر به صورت زیر عمل می شود:

`(dot(a,b)/(norm(b)^2))*b`



بخش سوم: عملیات جبری بر روی ماتریس ها

□ عملیات جبری بر روی ماتریس ها

```
a=[1 5 3;4 2 6;0 8 9]
```

```
b=a'
```

```
c=a+b
```

```
c=a-b
```

```
c=a*b
```

```
c=a.*b
```

```
c=a/b
```

```
a*a , a^2
```

```
a.*a , a.^2
```

```
sqrt(a) , a.^0.5
```



بخش چهارم: آشنایی با ماتریس های اولیه

□ ماتریس با درایه های صفر

`zeros(2,4)`

□ ماتریس با درایه های یک

`ones(2,3) , 5*ones(2)`

□ ماتریس با درایه های تصادفی بین ۰ و ۱

`rand(1,10)`

□ ماتریس همانی

`eye(3) , eye(2,1)`



بخش پنجم: سایر دستوره‌ای مرتبط با ماتریس

□ میانگین ماتریس

mean(a)
mean(a,1)
mean(a,2)
mean(a(:))

□ اندازه ماتریس (محاسبه تعداد سطرها و ستون ها در یک ماتریس)

size(a)
[x,y]=size(a)
size(a,1)
size(a,2)



بخش پنجم: سایر دستورهای مرتبط با ماتریس

□ نمایش بعد بزرگتر یک ماتریس

`length(a)`

□ نمایش تعداد درایه های ماتریس

`numel(a)`

□ پیدا کردن درایه یک عدد در ماتریس

`find(a==20)`

`find(a>=20)`

`find(a<20)`

`[a,b]=find(a==20)`



بخش پنجم: سایر دستورهای مرتبط با ماتریس

□ مرتب کردن آرایه های یک ماتریس به صورت افزایشی و کاهشی

```
c=[25 4 65 2 1 80]
```

```
sort(c)
```

```
sort(c,'ascend')
```

```
sort(c,'descend')
```

```
[r,n]=sort(c)
```

□ ایجاد یک ماتریس به صورت تصاعدی

```
linspace(0,10,2) , 0:10:10
```

```
linspace(0,10,3) , 0:5:10
```



بخش پنجم: سایر دستورهای مرتبط با ماتریس

□ حذف درایه های تکراری از ماتریس و مرتب کردن آن به صورت افزایشی

```
a=[5 20 1 5 3 2 1 3 5 20 4 9 9 4 10]
```

```
Unique(a)=1 2 3 4 5 9 10 20
```

```
unique(a,'stable')=5 20 1 3 2 4 9 10
```

□ تکرار یک ماتریس در چند سطر و ستون

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

```
repmat([1 2 3],3,1)
```

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

```
repmat([1 2 3],3,2)
```

بخش ششم: تمرین های فصل

❖ تمرین: ماتریس زیر را به صورت جمع یک ماتریس متقارن و پاد متقارن بنویسید
سپس خواسته های زیر را بدست آورید:

$$a = \begin{pmatrix} 0 & 4 & 2 \\ 3 & 1 & 5 \\ -2 & 4 & 3 \end{pmatrix}$$

- ۱- محاسبه ترانهاده a و قرار دادن آن در متغیری به نام b
- ۲- انجام عملیات جبری جمع، تفریق، ضرب و تقسیم بین a و b
- ۳- محاسبه معکوس a و قرار دادن آن در متغیری به نام c
- ۴- بدست آوردن ماتریس همانی با استفاده از عملیات جبری بر روی a و c
- ۵- محاسبه میانگین تمامی درایه های a
- ۶- مرتب کردن آرایه های ماتریس به صورت افزایشی

امام حسن عسکری علیه السلام:

تواضع و فروتنی نعمتی است که کسی بر آن حسودی نمی کند.