به نام پروردگار دانایی

آموزش مقدمات MATLAB

سید کاوه احمدی

نمای کلی ارائه

- محیط کاری MATLAB
- متغیرها
- آرایه ها
- Loops
- Flow Control
- M-Files
- Plotting

محیط کاری MATLAB

- Command Window: محل وارد کردن دستورات برای پردازش توسط MATLAB
- Command History Window: دستوراتی که قبلا در پنجره Command اجرا شده‌اند را نگهداری می‌کند.
- Current Directory: GUI مورد نیاز برای تغییر دادن پوشه‌ی جاری و فایل‌های MATLAB
- Workspace: متغرهای تعریف شده در طول برنامه را نشان می‌دهد.

محیط کاری MATLAB

- اجرای دستورات
  - who و whos اطلاعات مربوط به workspace را در command window نمایش می‌دهد.
  - clear کل workspace را خالی می‌کند.
  - clear name: متغیر با نام name را از workspace حذف می‌کند.
  - Clc منجر به پاک شدن command window می‌شود.
  - help command اطلاعاتی در مورد دستور مورد نظر به ما ارائه می‌دهد
  - lookfor keyword: دستورانی که شامل عبارت مشخص شده است را نمایش می‌دهد

متغیرها

- نام متغیرها Case sensitive است.
- متغیرها می‌توانند حداکثر ۳۱ کاراکتر طول داشته باشند.
- نام متغیرها می‌تواند شامل حروف + اعداد + "_" باشد
  - ابتدای نام متغیرها حتما باید حرف باشد.
- How_about_this_variable_name
- کلمات رزرو شده:
  - for, end, if, function, return, case, continue, persistent, global, try, catch

متغیرها

- متغیرها به شکل loosly type هستند.
- انتساب یک مقدار به عبارت:
  - X = 1
  - X = 1+2+3
  - Y = X+1;
  - Z = X+Y;
- قرار دادن «;» در انتهای دستورات منجر به عدم نمایش نتیجه پردازش در Command Window می‌شود.

متغرهای ویژه

- `Inf`: به جای مقدار بی نهایت
- `NaN`: `Not a Number` – جایگزین مقدار غیر عددی
- `nargin`: تعداد آرگومان های ورودی تابع
- `nargout`: تعداد آرگومان های خروجی تابع
- `ndims`: تعداد ابعاد آرایه را بازمی‌گرداند
- `pi`: عدد π
- `realmin`: کوچکترین عدد مثبت قابل استفاده
- `realmax`: بزرگترین عدد مثبت قابل استفاده

آرایه‌های یک بعدی

- تعریف آرایه: قرار دادن مقادیر بین دو براکت
  - `Array = [1 2 3 4];`
- آرایه‌های سطری
  - عناصر آرایه باید با `space` یا `,` از هم جدا شوند
  - `Array = [1 2 3 4]`
    - `Array =`

```
     1     2     3     4
```

- آرایه‌های ستونی
  - عناصر با `;` از هم جدا می‌شوند
  - `Array = [1; 2; 3; 4]`
    - `Array =`

```
     1
     2
     3
     4
```

آرایه‌های یک بعدی

```
R = [12, 11, 9];
S = [1, 4];
T = [R, S]
```


تعریف بردار و ماتریس در Matlab

- بردار: آرایه‌ی یک بعدی
  - بردار سطری
  - بردار ستونی
- ماتریس: آرایه چند بعدی

```
Array=[1 2 3; 4 5 6; 7 8 9; 10 11 12]
```

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

اندیس گذاری آرایه ها

- ذخیره‌سازی آرایه‌ها در Matlab به شکل ستونی است.
  - `Array=[1 2 3; 4 5 6; 7 8 9; 10 11 12]`
    - `Array =`

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

- `Array(1) = ?`
  - `1`
- `Array(3) =`
  - `7`
- `Array(?) = 2`
  - `5`

دسترسی به عناصر آرایه‌ها

- اندیس آرایه‌ها از ۱ شروع می‌شود.
- برای دسترسی به یک عنصر، اندیس آن را بین ( ) قرار می‌دهیم:
  - در آرایه یک بعدی:
    - `Array(1);`
  - در آرایه دو بعدی
    - `Array(2, 3)`

دستیابی بلوکی

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

- عناصر آرایه براساس محل ذخیره‌سازی
  - `Array(1:3)`
    ```
    1 4 7
    ```
- تمامی عناصر یک سطر
  - `Array(2, :)`
    ```
    4 5 6
    ```
- تمامی عناصر یک ستون
  - `Array(:, 3)`
    ```
    3
    6
    9
    12
    ```
- `Array(:, :) = ?`

دستیابی بلوکی

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

- انتخاب چند سطر
  - `Array(2:4, :)`
    ```
    4   5   6
    7   8   9
    10 11 12
    ```
- انتخاب چند ستون
  - `Array(:, 2:3)`
- ترکیب
  - `Array(2:4, 2:3)`
    ```
    5   6
    8   9
    11 12
    ```

دستیابی بلوکی

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

- انتخاب چند سطر
  - `Array(2:4, 2:end)`

```
   5   6
   8   9
   11  12
```


سایر نکات

- تغییر مقدار عناصر آرایه:
  - `Array(2, 3) = 8`
  - `Array(2:4, 2:3) = [1 1; 1 1; 1 1]`
- حذف یک سطر یا یک ستون از آرایه
  - `Array(2, :) = []` → حذف سطر دوم

سایر نکات

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

- `Array([1 1 1 1], :)`

```
  1   2   3
  1   2   3
  1   2   3
  1   2   3
```

- `Array([1,2,1], :)`

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


ایجاد آرایه‌های منظم

- `Array = 1:6`
  - `1 2 3 4 5 6`
- `Array = 1:2:12`
  - `1 3 5 7 9 11`
- `linspace(start, end, items number)`
  - `linspace(1, 2, 5)`
    - `1  1.2  1.5  1.75  2`

توابع ایجاد آرایه

- `zeros(n)`
  `ones(n)`
  - `zeros(3)`

```
     0   0   0
     0   0   0
     0   0   0
```

`ones(3)`

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

- `zeros(n, m)`
  `ones(n, m)`

توابع ایجاد آرایه

- `eye(n)`
  - `eye(3)`

```
     1   0   0
     0   1   0
     0   0   1
```

- `diag(1:2:10)`

```
   1   0   0   0   0
   0   3   0   0   0
   0   0   5   0   0
   0   0   0   7   0
   0   0   0   0   9
```

ترانهاده کردن یک آرایه

- Array =
  - 1 2 3
- Array' =
  - 1
    2
    3

- Array =
  - 1 2 3
    4 5 6
- Array' =
  - 1 4
    2 5
    3 6

عملیات روی آرایه‌ها

```
X = [2, 3, 6]
```

- **X + 2 =**
  4 5 8 (تمامی عناصر با ۲ جمع می‌شوند)
- **4 - X =**
  2 1 -2
- **X * 3 =**
  6 9 18
- **X / 3 =**
  0.6667 1.0000 2.0000
- **3 / X =**
  - Error!
- **X ^ 2 =**
  - Error!
  - power(X, 2) is Correct!

عملیات روی آرایه‌ها

```
X = [1 2 3; 1 2 3]
Y = [4 5 6; 4 5 6]
Z = [1 2; 1 2; 1 2]
```

- **X + Y =**

```
    5   7   9
    5   7   9
```

  - ماتریس‌ها باید ابعاد یکسانی داشته باشند

- **X + Z =**
  - Error!

عملیات روی آرایه‌ها

```
X = [1 2 3; 1 2 3]
Y = [4 5 6; 4 5 6]
Z = [1 2; 1 2; 1 2]
```

- **X * Z =**

```
    6   12
    6   12
```

  - عملیات ضرب ماتریسی انجام می‌شود.
    - یک ماتریس n*m را می‌توان در یک ماتریس m*k ضرب کرد. حاصل یک ماتریس n*k خواهد بود.

- **X * Y =**
  - Error!

عملیات آرایه‌ای نظیر به نظیر

- در برخی مواقع هدف ما از ضرب دو ماتریس، ضرب ماتریسی آنها نیست و می‌خواهیم عناصر نظیر به نظیر در هم ضرب شوند.
  - همانند جمع دو ماتریس
  - دو ماتریس باید ابعاد یکسان داشته باشند

```
X = [1 2 3; 1 2 3]
Y = [4 5 6; 4 5 6]
```

- **X .* Y =**

```
     4    10    18
     4    10    18
```


عملیات آرایه‌ای نظیر به نظیر

```
X = [2, 3, 6]
```

- **X .^ 2 =** <--> power(X, 2)

```
     4     9    36
```

- **2 .^ X =**

```
     4     8    64
```

- **3 ./ X =**

```
    1.5000    1.0000    0.5000
```

```
X = [1 2 3; 1 2 3]
Y = [4 5 6; 4 5 6]
```

- **X .^ Y =**

```
     1    32   729
     1    32   729
```

عملیات آرایه‌ای

```
a = [1 2 3]
b = [7 8 9]
a * b'

a = [1 2 3; 4 5 6]
b = [7 8 9; 10 11 12]
a .* b
```


توابع سودمند!

- طول بزرگترین بعد:
  - `length(Array)`
- طول آرایه یک بعدی یا اندازه‌ی سطرهای آرایه دو بعدی:
  - `size(Array)`
- اگر طول همه ابعاد آرایه را بخواهیم:
  - `[n, m] = size(Array)`
  - اگر فقط تعداد ستون‌ها مورد نیاز باشد:
  - `[~, m] = size(Array)`
  - `m = size(Array, 2)`

توابع سودمند!

```
X = [4 6 7 6 5 2 4 7 1]
```

- پیدا کردن بزرگترین عنصر آرایه یک بعدی:
  - `max(X) =`
    `7`
  - اگر بزرگترین و محل قرارگیری بزرگترین عنصر را بخواهیم:
  - `[value, index] = max(X)`
    `value = 7`
    `index = 3`
- به همین ترتیب برای `min`

توابع سودمند!

```
X = [3 5 2; 9 7 3; 7 8 9]
```

- پیدا کردن بزرگترین عنصر آرایه دو بعدی:
  - در آرایه دو بعدی به طور پیش‌فرض بزرگترین عنصر در هر ستون بازگردانده می‌شود
  - `[value, index] = max(X)`
    `value = 9 8 9`
    `index = 2 3 3`
  - اگر بزرگترین عناصر در سطرها را بخواهیم:
  - `[value, index] = max(X')`
  - `[value, index] = max(X, [], 2)`
- به همین ترتیب برای `min`

توابع سودمند!

- جمع عناصر آرایه
  - `sum(X)`
    - در آرایه دو بعدی عناصر ستون‌ها با هم جمع می‌شوند:

```
X = [3 5 2; 9 7 3; 7 8 9]
sum(X) =
     19   20   14
```

    - اگر جمع همه عناصر آرایه دو بعدی را بخواهیم:
  - `sum(sum(X))`
  - `sum(X(:))`

توابع سودمند!

- `class(a)`
- `str2num(a)`
- `det(a)`
- `sqrt(a)`
- `abs(X)`

گرفتن ورودی از کاربر

- دستور **input**:

```
n = input('Please Enter a Number: ');
disp(n ^ 2);
```


M-Fileها در Matlab

- برنامه های بزرگ
- حجم زیاد دستورات
- دستورهای تکراری
- ایجاد یک M-file
- قرار دادن آن در پوشه ی جاری
- فراخوانی آن : نوشتن نام M-File به عنوان دستور در Command window.
- استفاده از % به منظور اضافه کردن Comment به برنامه

ساختارهای تصمیم

```
a = input('a: ');
if(a <= 0)
    error('"a" Could not be less than 0');
elseif(a > 30)
    error('"a" Could not be more than 30');
else
    ezplot('sin(x)',[0,a]);
end
```


تکرار

```
y=0;
for i = 1:10
    y = y+1;
end
```

y = 10

```
y=0;
for i = 1:2:10
      y = y+1;
end
```

y = 5

```
x = [2 5 6 8];
for i = x
      disp(i);
end
```

2
5
6
8

مثال

- در آرایه‌ی زیر تعداد عناصر بزرگتر از ۳ را بیابید:
- `X = [ 2 4 7 5 1 2 3 4 5];`

- روش اول

```
cnt = 0;
for(i = 1:length(X))
    if(X(i) > 3)
        cnt = cnt + 1;
    end
end
disp(cnt);
```

- روش دوم

```
sum(X>3)
```


مثال

- محل قرارگیری تمام عناصر با بیشترین مقدار را در آرایه زیر بدست آورید و در یک آرایه ذخیره کنید:
  - `X = [1 3 4; 5 6 3; 6 2 5; 6 6 4];`

```
index = [ , ];
m = max(X(:));
for i = 1:size(X, 1)
    for j = 1:size(X, 2)
        if(X(i, j) == m)
            index = cat(1, index, [i, j]);
        end
    end
end
```

مثال

- روش دوم:

```
m = max(X(:));
[row, col] = find(X==m);
```


حلقه شرطی

```
y = 0;
x = 10;
while(x~= 0)
    x = x-1;
    y = y+1;
end
```

کنترل جریان

```
switch expression
    case test expression1
        command
    case test expression2
        command
    …
    otherwise
        command
end
```


تعریف توابع در Matlab

```
function [output variables] = name(input variables);
    commands
end
```

- نکته: نام تابع باید با نام `m-file` یکسان باشد.

تعریف توابع در Matlab

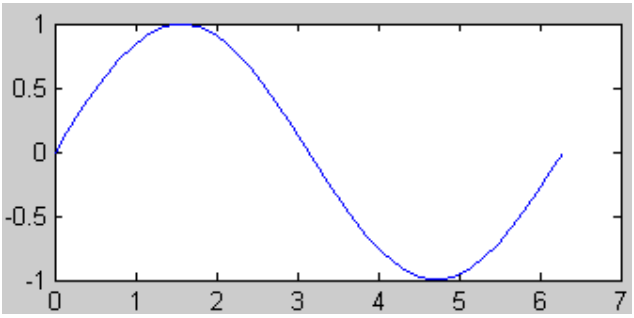
- مثال:

```
function [y] = test(x);
    y=0;
    while(x~=0)
        x=x-1;
        y=y+1;
    end
end
```

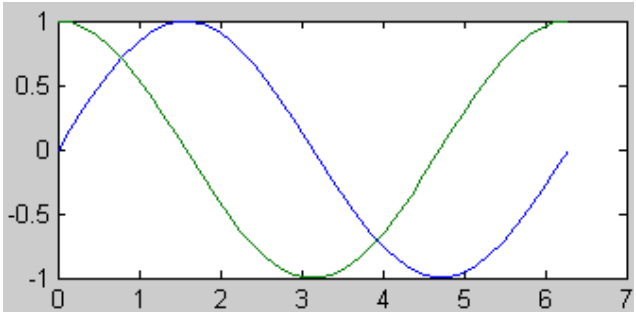
- اجرا در خط فرمان یا M-File‌ها:
  - `Y = test(10);`

رسم نمودار

```
X= linspace(0,2*pi,40);
Y= sin(X);
plot(X, Y);
```




```
Z= cos(X);
plot(X,Y,X,Z);
```




```
plot(X, Y, 'r')
plot(X, Y, '* b')
```

رسم نمودار

- دستوراتی جهت تکمیل نمودارها
  - `xlable('statement');`
  - `ylable('statement');`
  - `title('statement');`
  - `text(variables , statements);`
  - `legend('statement' , 'statement',…);`
  - `hold on`
  - `hold off`
  - `figure`
  - `close`
  - `grid`

رسم نمودار

- `subplot(a, b, c);`
- مثال

```
X = linspace(0,2*pi,40);
Y = sin(X);
Z = cos(X);
W = 2*sin(X).*cos(X);

subplot(2, 2, 1);
plot(X, Y);

subplot(2, 2, 2);
plot(X, Z);

subplot(2, 2, 3);
plot(X, W);
```

نمایش مقادیر متغیر ها در داخل یک عبارت

```
clc
a = input('insert a :');
b = input('insert b :');
c = a + b;
fprintf('Add %g and %g is : %g \n', a, b, c)
```