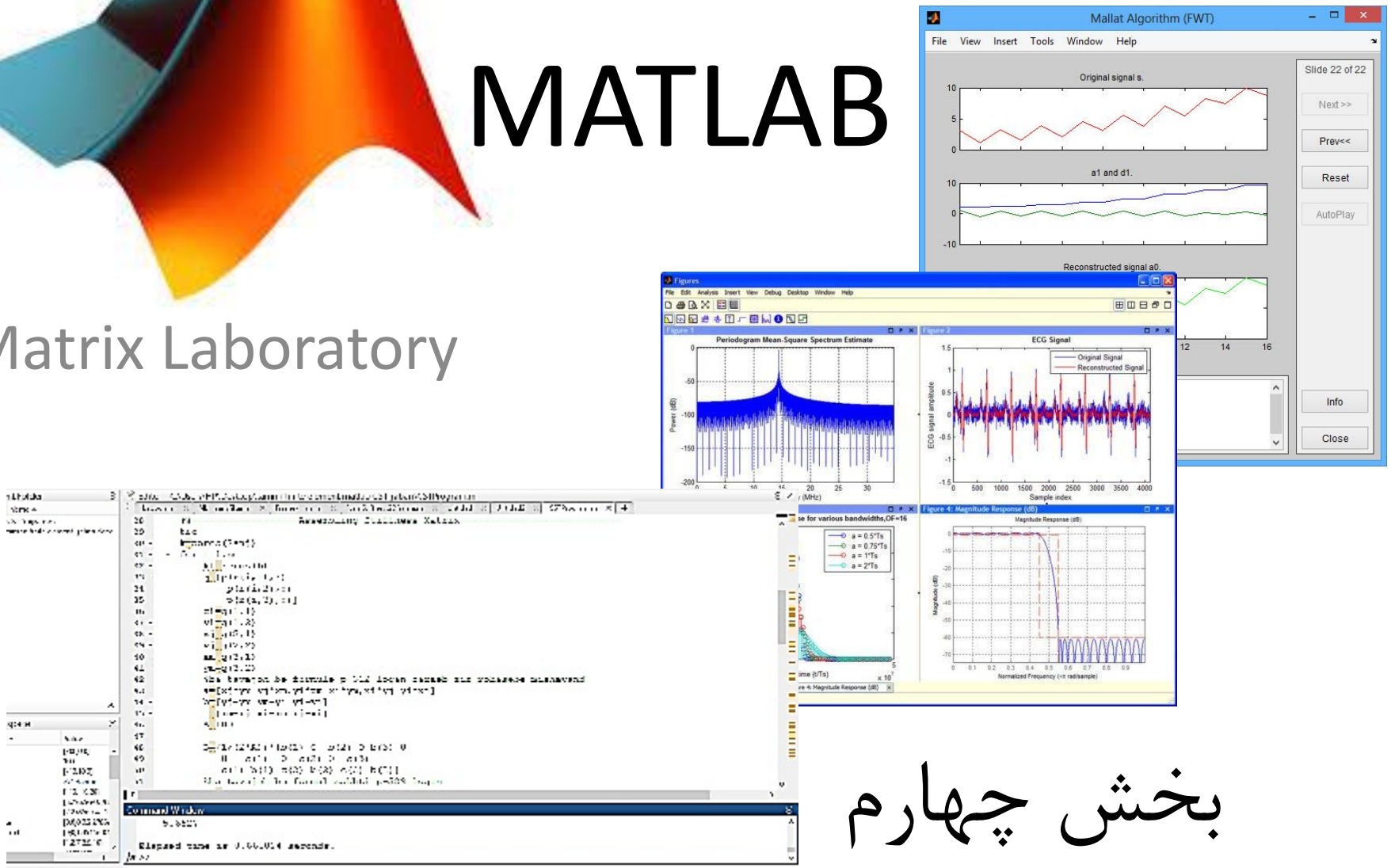
آموزش زبان برنامه نویسی متلب

MATLAB

Matrix Laboratory



بخش چهارم

تابع (Function)

ساختار کلی نوشتن تابع:

function [متغیرهای خروجی]= **نام تابع** (متغیرهای ورودی)

دستورالعمل جهت برآورد هدف تابع

end

نام تابع می تواند شامل حروف، اعداد و زیرخط تیره (آندرلاین _) باشد. همچنین نام تابع نمی تواند با عدد شروع شود.

متغیرهای خروجی یک تابع میتواند بیش از یکی باشند، یعنی قابلیت تعریف چند خروجی وجود دارد. اگر خروجی یک متغیر باشد نیازی به استفاده از [ ] نیست.

```
function [ output_args ] = untitled( input_args )
%UNTITLED Summary of this function goes here
%   Detailed explanation goes here



end
```

تابع (Function)

مثال: تابعی بنویسید که میانگین چند عدد را حساب کند.

```
function avg = average(x)
%Mean calculatione
% in tabe miyangin adad ra heseb mikonad
n=length(x);
avg=sum(x(:))/n;
end
```

برای مشاهده help تابع که خودمان نوشته ایم در پنجره دستور تایپ کنید:

help average

برای مشاهده جواب در پنجره دستور تایپ کنید:

average([1 2 3]) **A=average([1 2 3])** **[avg]=average([1 2 3])**

تابع (Function)

مثال: تابعی بنویسید که مساحت و محیط یک دایره را حساب کند.

```
function [p,a] = Circle_P_A(D)
%This function calculates circle perimeter and area
%Mohaseneye mohit va masaahat
p=D*pi;
a=pi*D^2/4;
end
```

برای مشاهده جواب در پنجره دستور تایپ کنید:

[p a]=Circle_P_A(2)

نیازی نیست که حتماً همان متغیر های خروجی را فراخوانی کنیم:

[mohit masahat]=Circle_P_A(2)

مثال: تابعی بنویسید که ماتریس زیر را تولید کند.

$$a_{ij} = \min(i, j)$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

```
function A=MatrixCreator(m,n)

A=zeros(m,n);
   for i=1:m
    for j=1:n
        A(i,j)=min(i,j);
    end
   end
end
```

خروجی

```
>> A=MatrixCreator(4,4)

A =

     1     1     1     1
     1     2     2     2
     1     2     3     3
     1     2     3     4
```

مثال: تابعی بنویسید که ماتریس زیر را تولید کند.

$$a_{ij} = \max(i, j)$$

$$A = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 \\ 1 & 2 & 3 & 3 \\ 1 & 2 & 3 & 4 \end{bmatrix}$$

```
function A=MatrixCreator(m,n)
if nargin<1
    error('you must enter at least one argument!');
end
if nargin<2
    n=m;
end
A=zeros(m,n);
for i=1:m
    for j=1:n
        A(i,j)=min(i,j);
    end
end
end
```

خروجی ۱

```
>> A=MatrixCreator(4,4)

A =

     1     1     1     1
     1     2     2     2
     1     2     3     3
     1     2     3     4
```

خروجی ۲

```
>> CreateMyMatrix()
Error using CreateMyMatrix (line 4)
You must enter at least one argument.
```

مثال: تولید یک تابع که تمامی اعداد وارد شده ا در هم ضرب کند:

```
function p=Product(varargin)

    p=varargin{1};

    n=numel(varargin);

    for i=2:n
        p=p.*varargin{i};
    end

end
```

هر تعدادی که عدد وارد شود همگی را در یک آرایه سلولی ذخیره می کند.

تعداد المان های وارد شده را می شمارد

خروجی

```
>> Product(1,2,3)

ans =

     6
```

مثال: تولید یک تابع که تمامی اعداد وارد شده ا در هم ضرب کند:

```
function p=Product(varargin)

%       p=varargin{1};
%
%       n=numel(varargin);
%
%       for i=2:n
%           p=p.*varargin{i};
%       end

    x=cell2mat(varargin);
    p=prod(x);

end
```

مادامی که ورودی ها اعداد اسکالر باشند، این برنامه بهتر کار میکند. اما اگر ورودی ها ماتریسی باشند، این برنامه جواب نخواهد داد و می بایست از کد قبلی استفاده کرد.

بررسی و تشخیص اعداد اول
Prime Numbers

چنانچه بخواهیم اول بودن عدد N را بررسی کنیم کافی است بخش پذیری عدد N بر اعداد ۲ تا رادیکال N بررسی شوند. اگر N بر هیچکدام از اعداد این بازه بخش پذیر نباشد، اول است.

البته در مورد دو عدد ۱ و ۲ بایستی رویکرد مجزایی در نظر گرفته شود

بررسی و تشخیص اعداد اول

```
clc;
clear;
close all;
n=input('Enter a positive integer: ');

IsPrime=true;

for i=2:floor(sqrt(n))
    if mod(n,i)==0
        IsPrime=false;
        break;
    end
end

if IsPrime
    disp([num2str(n) ' is a Prime Number.']);
else
    disp([num2str(n) ' is NOT a Prime Number.']);
end
```

بررسی و تشخیص اعداد اول به کمک تابع:

```
function b=IsPrime(n)

    b=true;

    for i=2:floor(sqrt(n))
        if mod(n,i)==0
            b=false;
            break;
        end
    end

end
```

یادآوری ساختار کلی تابع:

```
function [out1, out2, ...] = funcname(in1, in2, ...)
...
In1…in2….➔….out1…..out2
...
end
```

```
clc;
clear;
close all;

n=input('Enter a positive integer: ');

if IsPrime(n)
    disp([num2str(n) ' is a Prime Number.']);

else
    disp([num2str(n) ' is NOT a Prime Number.']);

end
```

بدست آوردن لیستی از مقسوم علیه های یک عدد:

```
function DL=GetDividers(n)

    DL=[1 n];
    for i=2:floor(sqrt(n))
        if mod(n,i)==0
            DL=[DL i n/i];
        end
    end

    DL=unique(DL);

    %DL=sort(DL);

end
```

حذف تکراری بودن ارقام درایه های ماتریس

مرتب کردن اعداد درایه های یک بردار از کوچک به بزرگ

نوار عمودی سمت راست m فایل زرد رنگ است. پس برنامه مشکل دارد.
برای Ok کردن این ایراد از دستور %#OK در انتهای خط دستور استفاده می کنیم.

تمرین: برنامه ای برای تجزیه یک عدد به عوامل اول بنویسید.

$$20 = 2 \times 2 \times 5$$

$$36 = 2 \times 2 \times 3 \times 3$$

$$99 = 3 \times 3 \times 11$$

$$30 \div 2 = 15 \rightarrow$$

$$\rightarrow 15 \div 3 = 5$$

$$\rightarrow 5 \div 5 = 1$$

$$3 = 2 \times 3 \times 5$$