Function in Mat_lab

دوال ملفات M

M – File Function

الدوال function :- الدوال هي مجموعة من الاوامر او الجمل البرمجية المكتوبة في ملفات M – File والتي تستخدم لتؤدي وظيفة معينة ولها اسم مميز يعبر عن وظيفتها لتؤدي امر او مجموعة من الاوامر الجاهزة .

انواع الدوال : function types

1. يحتوي برنامج MATLAB على مئات من الدوال الداخلية الجاهزة مبنية في بنية البرنامج **MATLAB Built in Function** بدلا من كتابة او برمجة هذه الدوال في كل مرة مثل sum , prod , mean , inv , det , size , length , round , fix , rem , angle , abs وغيرها من الدوال التي تم برمجتها مسبقا في برنامج MATLAB .
2. دوال يتم انشاءها من قبل المستخدم **User – Defined Function** تكتب بصيغة برنامج MATLAB ويسمى هذا النوع من البرمجة بدوال ملفات M (M – File Function) .
   - تضاف هذه الدوال بعد انشاءها الى مكتبة الدوال الداخلية الجاهزة (MATLAB Built in Function) ، ويتم تسميتها من قبل المستخدم بتصميمه لها حسب وظيفتها ، ويتم كتابة البرامج بصيغة function بالشكل التالي :-

```
function [ list of output variables ] = function_name (list of input variables)
function (o/p)                        = function_name(i/p)
```

OR

```
function  function_name (list of input variables)
function function_name(i/p)
```

شروط تسمية وانشاء دوال ملفات M (M – File Function) :-

1- يسمى اسم الملف بنفس اسم الدالة التي تم كتابتها في البرنامج .
2- السطر الاول من البرنامج يجب ان يبدأ بالامر function .
3- يتبع اسم الدالة شروط تسمية المتغيرات في برنامج MATLAB .
4- لايجوز استخدام اسم الدالة من اسماء مشابهة لاسماء المتغيرات المستخدمة في البرنامج .

الفرق بين برمجة الدوال (function) والبرمجة النصية (script) :-

1- برنامج الدوال يبدأ بالامر function بينما برنامج الـ script يبدأ بالبرنامج مباشرة .
2- يتطلب كتابة برنامج الدوال function تعريف المخرجات والمدخلات مع اسم الدالة بينما في البرمجة النصية الـ script لايحتاج الى ذلك .
3- تستطيع في برنامج الدوال function عند تنفيذه تغيير قيم المدخلات بينما في البرمجة النصية الـ script لايمكن ذلك .

4- المتغيرات التي تحسب داخل الـ function لا تحفظ في الـ workspace بينما في الـ script تحفظ في الـ workspace ، وتسمى متغيرات محلية (local variable) .

معرفة عدد المدخلات والمخرجات للدوال(nargout ,nargin) :-

يستخدم الامر **nargin** لمعرفة عدد المعاملات (المتغيرات) المستخدمة داخل الدالة بالشكل التالي

a = nargin (' function_name ')

ex :-

a = nargin ('sum')

a =

3

ويستخدم الامر **nargout** لمعرفة عدد المعاملات (المتغيرات) المستخدمة كمخرجات لهذه الدالة وبالشكل التالي :-

a = nargout (' function_name ')

ex :-

a = nargout (' sum ')

a =

1

Example 1: Write m-file function in Matlab to find Maximum number among three input a,b,c ?

Solution:

```
function max=my_max(a,b,c)
max=a;
if(b>max)
    max=b;
elseif(c>max)
    max=c;
end
end
```

```
Run:in command window call the function
>> max=my_max(50,11,0)
    Max=50
```

Example2: Write m-file function in Matlab to test input number if its Odd or Even?

```
function n=Test_no(a)
if mod(a,2)==0
    n='Even no';
else
n='odd number';
end
end
```

Run:in command window call the function :

```
>>n=Test_no(9)
  N=odd
>>n=Test_no(0)
  N=Even
```

Example3: write m-file function to find factorial of any input number?

```
function f=fact(n)
f=1;
for i=1:n
    f=f*i;
end
end
```

Run: call in command window call the function :

```
>> f=fact(4)
   f=24

>> f=fact(5)
    f=125
```

Example4: Write m-file function in Matlab to test number if positive or negative?

Solution:

```
 function PN=Po_Ne(x)
if (x>=0)
    PN='POsitive';
else
    PN='Negative';
end
end
```

Run: call the function in command window

```
>> PN=Po_Ne(0)
   PN=positive
>> PN=Po_Ne(-1)
    PN=Nagtive
```

Exaple5: Write m-file function in Matlab to find production table of any number n?

Solution:

```
function production(n)
for i=1:10

disp([num2str(n),'*',num2str(i),'=',num2str(n*i)]);
end
```

Run: in command window call the function

```
>> production(3)
>>3*1=3
>>3*2=6
>>3*3=9
.
.
.
.
3*10=30
```

Example5: write m-file function to find solution for equation second degree by using Quardtic equation?

$$x = \begin{cases} \dfrac{-b \pm \sqrt{b^2 - 4ac}}{2a} & b^2 - 4ac > 0 \\ \dfrac{-b}{2a} & b^2 - 4ac = 0 \end{cases}$$

واذا كان قيمة تحت الجذر اقل من صفر اطبع العبارة (the root is complex)

Solution:

```
function quadratic_equation ( a , b , c )

delta = b^2 – 4 *a * c
if delta > 0
x1 = ( – b + sqrt ( delta )) / ( 2 * a )
x2 = ( – b – sqrt ( delta )) / ( 2 * a ) elseif
delta < 0
disp ( ' the root is complex ')
else
```

```
x1_2=( -b/(2*a))
end
```

```
Run: call the function in command window
```

```
>> quadratic_equation ( 4 , 6 , 2 )
```

```
delta = 4
x1 = -0.5000
```

Function that deals with array

عند التعامل مع المصفوفات في كتابة الداله

سيكون البرنامج عبارة عن جزئين رئيسيين

1. الجزء الخاص بالداله حيث يتم كتابه عمل الداله وخزن البرنامج بنفس اسم الداله
2. الجزء الثاني وخاص بالجزء الرئيسي من خلاله يتم ادخال الداله وأيضا استدعاد الداله وطباعه النتائج وحفظ البرنامج باي اسم يريده المستخدم
3. التفيذ يكون في جزء command window حيث تم كتابه اسم البرنامج الرئيسي فقط.
4. اسم الداله يكون قريب من وظيفة الداله.

Example: write m-file function to read 5*5 Matrix

And then find multiplication of number greater than 2?

Part of function	Part of main program

`function t=mul(x);`
`t=1;`
`for i=1:5`
`  for j=1:5`
`  if x(i,j)>2`
`  t=t*x(i,j);`
`  end`
`  end`
`end`
 	
Save the function with name:mul.m
	`x=input('matrix=');`
	`for i=1:5`
	`  for j=1:5`
	`  t=mul(x);`
	`  end`
	`end`
	`disp(['the multplcation of elemnts that than two=',num2str(t)]);`
	save the script with name :mm1

Run: in command window call the main program >>mm1 matrix=[3 3 3;1 1 1;0 0 0] the multplcation of elemnts that greater the two=27	

Example: write m-file function to read 4*3 Matrix to count and sum the even elements in this Matrix?

Solution

Part of function	Part of main program

function
 	
function
[c,s]=co_su(x)
s=0;
c=0;
for i=1:4
for j=1:3
if mod
(x(i,j),2)==0
c=c+1;
 	
s=s+x(i,j);
end
end
end
 	
Save the function with name:co_sum
	clear;
	clc;
	x=input('matrix=');
	for i=1:4
	for j=1:3
	[c,s]=co_su(x);
	
	end
	end
	disp(['the number of Even no. =',num2str(c)]);
	disp(['the sum of Even no. =',num2str(s)]);
	
	save the script with name :cc1

```
Run: in command window call the main program
>>cc1
matrix=[3 3 2 ; 3 4 6 ;1 1 2 ;5 7 9]
the number of Even no. =4
the sum of Even no. =14
```

Example: : write m-file function to read any 3*3 Matrix and calculate positive and negative number in matrix?

Part of function

```
function [p,n]=po_ng(x)
p=0;
n=0;
for i=1:3
    for j=1:3
        if x(i,j)>=0
            p=p+1;
        elseif x(i,j)<0
            n=n+1;
        end
    end
end
```

Save the function with name:po_ng

Part of main program

```
x=input('matrix=');
for i=1:3
    for j=1:3
        [p,n]=po_ng(x);
    end
end
disp(['the even
no=',num2str(p)]);
disp(['the odd
no=',num2str(n)]);
```

save the function with name:mm3

Run :call the main program by write in command win:

```
>>mm3

matrix=[1 -2 0;0 0 0;-1 1 1]
```

```
the positive no no=7

the negative =2
```

Example: write m-file function to read any 3*3 Matrix and calculate Even and Odd number in matrix?

Part of function

```
function[E,O]=count_EO(x)
E=0;
O=0;
for i=1:3
    for j=1:3
        if
mod(x(i,j),2)==0
            E=E+1;
        else
            O=O+1;
        end
    end
end
```

Part of main program

```
clear;
clc
x=input('matrix=');
for i=1:3
    for j=1:3
  [E,O]=count_EO(x);
    end
end
disp(['the no of Even
no=',num2str(E)]);
disp(['the no of Odd
no=',num2str(O)]);
```

Save the function with name: count_EO	save the function with name:mm3

Run :call the main program by write in command win:

```
>>mm3

matrix=matrix=[0 12 17; 11 10 30; 40 50 90]

the no of Even no=7

the no of Odd no=2
```