

دوال ملفات M

Function M – File

MATLAB Programs:

1- Script M – Files

A script is simply a sequence of commands written in a file.

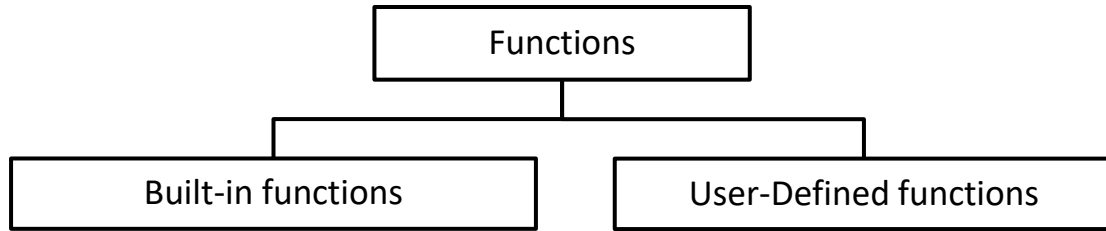
البرامج السابقة التي كتبت هي جميعها مجموعة سكريبتات (تعريف الـ script: وهو النص المكتوب بلغة البرمجة) وهو مجموعة الاوامر المتسلسلة ولها اسم محدد تحفظ في البرنامج .

2- Function M – File

A function M-file is a type of script file (i.e. a file with an extension of .m) which you can ‘call’ interactively, or from other scripts, in specific ways.

2- 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



انواع الدوال

1. MATLAB Built in Function

يحتوي برنامج MATLAB على مئات من الدوال الداخلية الجاهزة مبنية في بنية البرنامج بدلا من كتابة او برمجة هذه الدوال في كل مرة مثل sin ,cos ,tan , size , det , inv , mean , prod , sum , length , round , fix , rem , angle , abs وغيرها من الدوال التي تم برمجتها مسبقا في برنامج MATLAB .

2. User – Defined Function

دوال يتم انشاءها من قبل المستخدم تكتب بصيغة برنامج MATLAB ويسمى هذا النوع من البرمجة بدوال ملفات (Function M – File) (. M)

✓ تضاف هذه الدوال بعد انشاءها الى مكتبة الدوال الداخلية الجاهزة (MATLAB Built in Function) ، ويتم تسميتها من قبل المستخدم بتصميمه لها حسب وظيفتها .

كيفية كتابة الدالة (Structure of Function M-Files)

، ويتم كتابة البرامج بصيغة function بالشكل التالي :-
1- سطر تعريف الدالة function definition line

يجب ان يبدأ اول سطر من برنامج الدالة بكلمة function ويعرف بعدها مخرجات البرنامج واسم الدالة ومدخلاتها ، وبالشكل التالي .

```
function [ list of output variables ] = function_name (list of input variables)
```

OR

```
function function_name (list of input variables)
```

1- كلمة function

2- مخرجات البرنامج [list of output variables] :- عدد المتغيرات المطلوبة كمخرجات من عمل ووظيفة الدالة .

3- اسم الدالة function_name :- وهو اسم الملف في نفس الوقت يكتب كاسم للدالة كشرط رئيسي من طريقة كتابة الدالة (اسم الدالة ≡ اسم البرنامج) .

4- مدخلات البرنامج (list of input variables) :- يجب تعريف مجموعة المتغيرات التي يحتاجها برنامج الدالة .

2- سطر الملاحظات (% several comment lines :-

ويتبع سطر تعريف الدالة بعض الاسطر للملاحظات ويبدأ بعلامة (%) ويسمى help text ويتم عرض الملاحظات هذه عند كتابة الامر help قبل اسم الدالة في نافذة الاوامر

```
>>help function _name
```

3- اسطر اوامر برنامج الدالة function body :-

الاسطر المتبقية هي لكتابة اوامر برنامج ماتلاب MATLAB statements التي يعمل على اداء وظيفة الدالة وحساب مخرجات الدالة .

الشكل النهائي للدالة هو كما في الترتيب التالي :-

```
function [outarg1, outarg2 ...] = name (inarg1, inarg2 ...)
```

```
% comments to be displayed with help
```

```
...
```

```
outarg1 = ... ;
```

```
outarg2 = ... ;
```

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

- 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 لا يحتاج 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 .

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

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

```
a = nargin ( ' function_name ' )
```

ex :-

```
a = nargin ('sin')
```

```
a =
```

```
1
```

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

:-

```
a = nargout ( ' function_name ' )
```

ex :-

```
a = nargout ( ' sum ' )
```

```
a =
```

```
1
```

بعض الامثلة:-

مثال 1:- اكتب برنامج يحل معادلة من الدرجة الثانية بطريقة الدستور (Quadratic equation)

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

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

Sol.

`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`

نتائج تنفيذ البرنامج

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

`delta =`

`4`

`x1 =`

`-0.5000`

`x2 =`

`-1`

تمرين للطالب ، نفذ البرنامج لقيم اخرى لكل من a, b, c .

مثال 2 :- اكتب برنامج بلغة MATLAB لايجاد القيمة العظمى من بين ثلاث قيم ندخلها للبرنامج

Sol.

```
function max1( a , b , c )
if a > b
max = a ;
    if c > max
max = c ;
    end
else
max = b ;
    if c > max
max = c ;
    end
end
max
```

لتنفيذ البرنامج اكتب اسم البرنامج مع ادخال القيم مع اسم البرنامج

```
>> max1 ( 10 , 2 , 18 )
```

```
max =
```

```
18
```

تمرين للطالب ، ادخل قيم اخرى مع هذه الدالة .

مثال 3 :- اكتب برنامج بلغة MATLAB لادخال رقم واختبار هل هو زوجي ام فردي .

Sol.

```
function n= test_number( a )
if mod ( a , 2 ) == 0
```

```
n = 'even no.';
else
n = 'odd no.';
end
```

لتنفيذ البرنامج ، اكتب اسم البرنامج مع اعطاء قيمة بين القوسين :-

```
>>test_number ( 4 )
```

even no.

تمرين للطالب ، ادخل قيم اخرى .

مثال 4 :- اكتب برنامج بلغة MATLAB لاجاد مضروب العدد .

Sol.

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

حلقة الدوران الاولى	حلقة الدوران الثانية	حلقة الدوران الثالثة
i=1 (i<=n)	i=2 (i<=n)	i=3,.....,n
v = 1	v = 1	v = 2
i = 1	i = 2	i = 3
v = 1*1	v = 1*2	v = 2*3
1	2	6

لتنفيذ البرنامج ، اكتب اسم البرنامج مع اعطاء قيمة للعدد المطلوب ايجاد قيمة المضروب له :-

```
>>fact( 3 )
```

v =

6

```
>> fact ( 10 )
```

v =

3628800

تمرين للطالب ، ادخل قيم اخرى .

مثال 5 :- اكتب برنامج بلغة MATLAB لظهار جدول الضرب لاي رقم .

Sol.

```
function production ( num )
if length ( num ) ~= 1 || ~ isnumeric ( num )
disp ( ' error : please enter one number ' )
else
for i = 1 : 10
disp ( [ num2str( num ) , ' x ' , num2str( i ) , ' = ' , num2str( num * i ) ] )
end
end
```

ملاحظة :- 1- يفحص البرنامج اذا كان الرقم المدخل هو ليس عنصر واحد او قيم ليست عددية ، فيطبع العبارة التصحيحية لتبين عدم ادخال صحيح .

2- يعمل الامر disp على طبع مصفوفة مكونة من 5 عناصر يجب تحويل كل عنصر من عناصرها الى نصوص (سلسلة حرفية string) , لايمكن استخدام مصفوفة مكونة من ارقام و حروف ، يجب ان تكون المصفوفة اما حروف لوحدها او ارقام لوحدها :-

```
[ num2str ( num ) , ' x ' , num2str ( i ) , ' = ' , num2str ( num * i ) ]
```

ex :-

```
1 x 1 = 1
```

لتنفيذ البرنامج اكتب اسم الدالة في الـ command window وادخل الرقم بين قوسين صغيرين

```
>> production([12 5])
```

```
error : please enter one number
```

```
>>production('g')
```

```
error : please enter one number
```

يجب ادخال رقم واحد وليس مصفوفة او سلسلة حرفية

```
>> production(12)
```

```
12 x 1 = 12
```

```
12 x 2 = 24
```

$$12 \times 3 = 36$$

$$12 \times 4 = 48$$

$$12 \times 5 = 60$$

$$12 \times 6 = 72$$

$$12 \times 7 = 84$$

$$12 \times 8 = 96$$

$$12 \times 9 = 108$$

$$12 \times 10 = 120$$

تمرين للطالب ، ادخل عدة قيم اخرى .