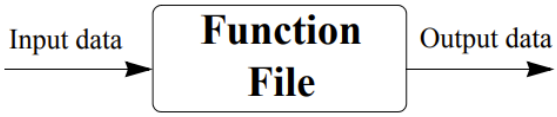
الفصل الخامس

User – Defined Functions and Function Files

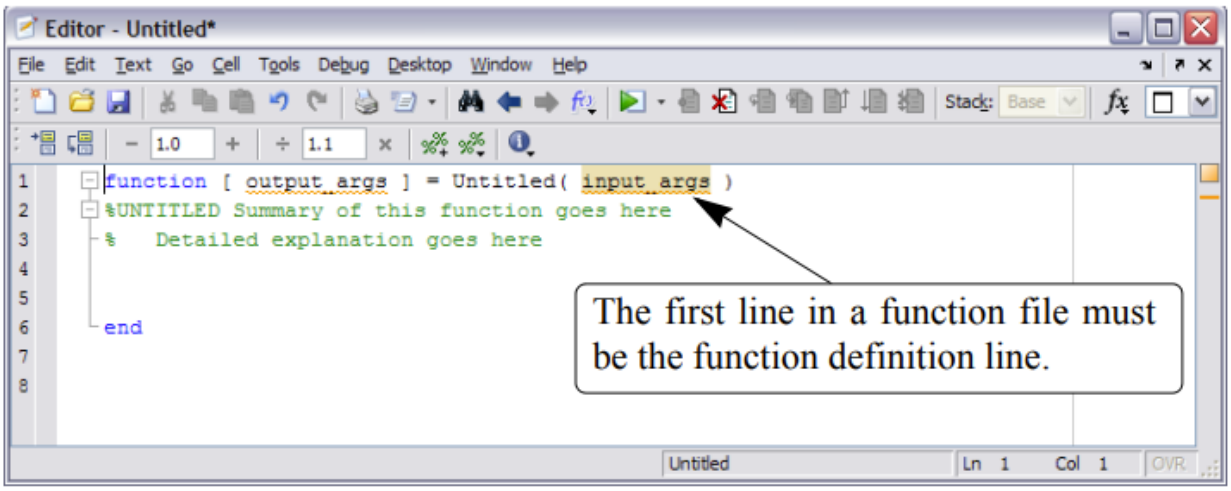
تتم برمجة العديد من الدوال داخل MATLAB كدوال مدمجة، ويمكن استخدامها في التعبيرات الرياضية ببساطة عن طريق كتابة اسمها مع متغير ويتم إنشاء دالة من قبل المستخدم وخزنها في ملف يعرف بملف الدالة



عندما يكون تعبير الدالة بسيطاً ويحتاج إلى حسابه مرة واحدة فقط، فيمكن كتابته كجزء من البرنامج. ومع ذلك، عندما تحتاج الدالة إلى التقييم عدة مرات لقيم مختلفة من المتغيرات، فمن الملائم إنشاء دالة "محددة من قبل المستخدم". بمجرد إنشاء (حفظ) دالة محددة من قبل المستخدم، يمكن استخدامها تماماً مثل الدوال المضمنة.

1- إنشاء ملف الدالة

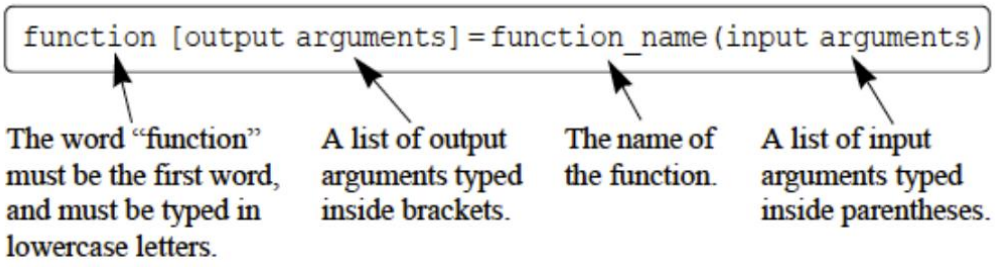
يتم إنشاء ملف الدالة وتحريرها من خلال نافذة المحرر script كما في ملف النصي. يتم فتح هذه النافذة من خلال قائمة New ثم اختيار Function كما في الشكل أدناه:



كما يمكن استخدام الملف النصي script لكتابة الدالة

2- تعريف الدالة Function definition line

يتم تعريف الدالة في السطر الأول من الملف النصي ويكون بالشكل التالي:



يجب أن تكون الكلمة "function" المكتوبة بأحرف صغيرة هي الكلمة الأولى في سطر تعريف الدالة تظهر الكلمة باللون الأزرق على الشاشة. يتم كتابة اسم الدالة بعد علامة المساواة. يمكن أن يتكون الاسم من أحرف وأرقام وعلامة الخط المنخفض (لا يمكن أن يتضمن الاسم مسافة). قواعد الاسم هي نفس قواعد تسمية المتغيرات ويجب تجنب أسماء الدوال المضمنة مسبقاً بواسطة MATLAB.

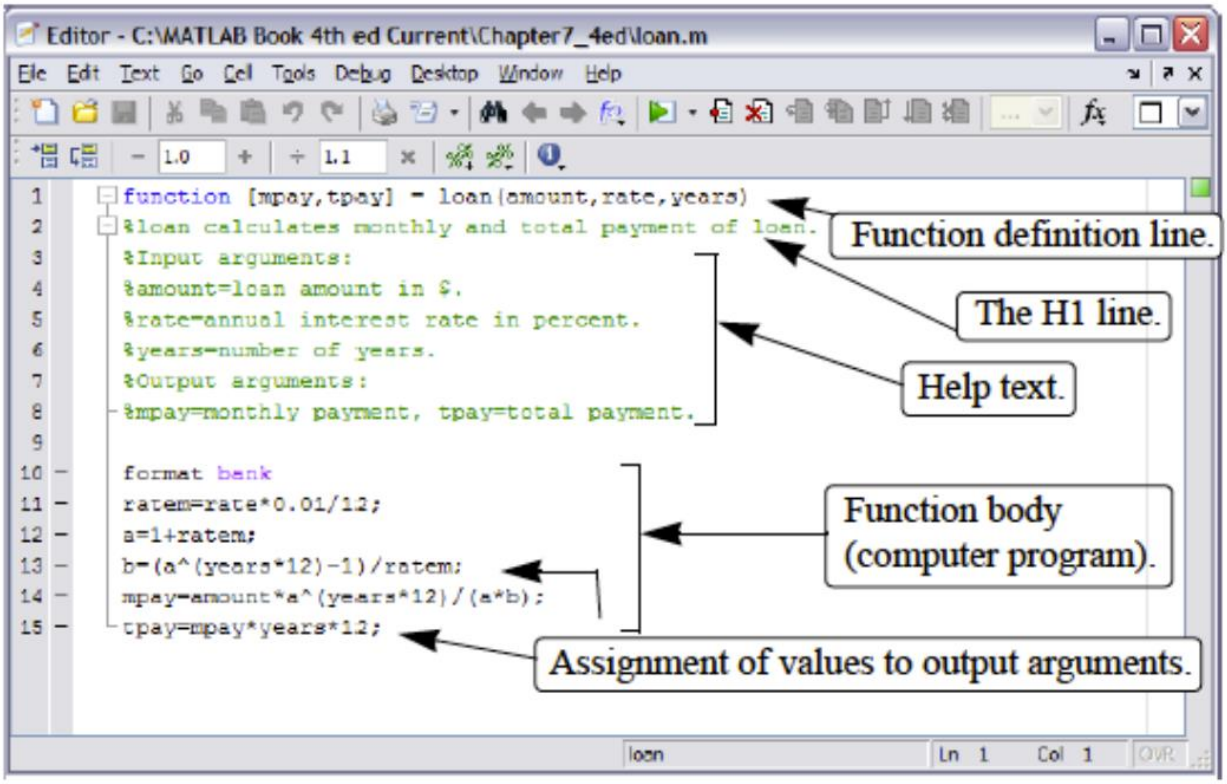
3- متغيرات الأدخال والأخراج Input and Output Arguments

يتم أستخدام مدخلات البرنامج لتحويل البيانات المستخدمة في الدالة لايجاد مخرجات الدالة. الامثلة أدناه لتعريف الدالة لتركيبات مختلفة من متغيرات الادخال والاخراج:

Function definition line	Comments
function [mpay,tpay] = loan(amount,rate,years)	Three input arguments, two output arguments.
function [A] = RectArea(a,b)	Two input arguments, one output argument.
function A = RectArea(a,b)	Same as above; one output argument can be typed without the brackets.
function [V, S] = SphereVolArea(r)	One input variable, two output variables.
function trajectory(v,h,g)	Three input arguments, no output arguments.

4- بنية الدالة Function Body

تحتوي بنية الدالة على البرنامج (الكود) الذي يقوم بالفعل بإجراء العمليات الحسابية. يمكن للكود استخدام جميع ميزات برمجة MATLAB. يتضمن ذلك الحسابات، والواجبات، وأي دوال مدمجة أو محددة من قبل المستخدم، والتعليقات، والأسطر الفارغة، والمدخلات والمخرجات التفاعلية. ويجب إنهاء هذه الخطوات بكلمة end.



* يتم خزن ملف الدالة من خلال اختيار Save as من الشريط او من File حسب سنة الأصدار ويتم خزن الملف قبل استخدام الدالة. يتم حفظ ملف الدالة بنفس الاسم الذي تم اختياره في السطر الاول من تعريف الدالة، لذلك يتم تنفيذ او استدعاء الدالة بنفس الأسم الذي تم خزن ملف الدالة فيها. كما نلاحظ في الأمثلة أدناه:

Function definition line	File name
function [mpay,tpay] = loan(amount,rate,years)	loan.m
function [A] = RectArea(a,b)	RectArea.m
function [V, S] = SphereVolArea(r)	SphereVolArea.m
function trajectory(v,h,g)	trajectory.m

* يتم استخدام الدالة المعرفة User-defined function من قبل المستخدم بنفس الطريقة التي تستخدم فيها الدوال الجاهزة ويمكن استدعاء هذه الدالة في نافذة الامر. هناك طريقتين لتنفيذ الدالة المعرفة والمبينة بالامثلة الاتية :

```
>> [month total]=loan(25000,7.5,4)
```

First argument is loan amount, second is interest rate, and third is number of years.

```
month =
        600.72
total =
      28834.47
```

```
>> a=70000;   b=6.5;
>> [x y]=loan(a,b,30)
```

Define variables a and b.

Use a, b, and the number 30 for input arguments and x (monthly pay) and y (total pay) for output arguments.

```
x =
        440.06
y =
     158423.02
```

مثال:

اكتب ملف الدالة للدالة $f(x) = \frac{x^4\sqrt{3x+5}}{(x^2+1)^2}$ ، أستخدم الدالة لحساب

1) f(x) عندما x=6
2) f(x) عندما x=1,3,5,7, 9,11

الحل:

ملف الدالة لدالة f(x) كالتالي:

```
function y=chp7one(x)
y=(x.^4.*sqrt(3*x+5))./(x.^2+1).^2;
```

Function definition line.

Assignment to output argument.

1) يمكن حساب الدالة عن طريق كتابة chp7one(6) في نافذة الأوامر، أو عن طريق تعيين قيمة الدالة لمتغير جديد:

```
>> chp7one(6)
ans =
    4.5401
>> F=chp7one(6)
F =
    4.5401
```

2) لحساب الدالة لعدة قيم x، يتم إنشاء متجه بقيم x.

```
>> x=1:2:11
x =
     1     3     5     7     9    11
```

```
>> chp7one(x)
ans =
    0.7071    3.0307    4.1347    4.8971    5.5197    6.0638
```

وبطريقة أخرى

```
>> H=chp7one([1:2:11])
H =
    0.7071    3.0307    4.1347    4.8971    5.5197    6.0638
```

تمرين:

اكتب ملف دالة لتحويل درجات الحرارة من الفهرنهايت F الى السيليزية C باستخدام العلاقة الاتية:

$$C = \frac{5}{9}\ (F - 32)$$

احسب الدالة عند 40 F

5- مقارنة بين ملفات البرامج النصية وملفات الدوال

أوجه التشابه والاختلاف بين ملف البرنامج النصي وملف الدالة و يتم تلخيصه كالتالي:

- يتم خزن وحفظ كلا الملفين بالامتداد (.m) ولهذا السبب تسمى الملفات (m-file)
- السطر الأول القابل للتنفيذ في ملف الدالة هو يجب أن يكون تعريف الدالة

•المتغيرات في ملف الدالة تعريفها يكون موقعياً بينما يتم التعرف على المتغيرات الموجودة في ملف البرنامج النصي في نافذة الأوامر.

•يمكن لملفات البرنامج النصي استخدام المتغيرات التي تم تعريفها في مساحة العمل.

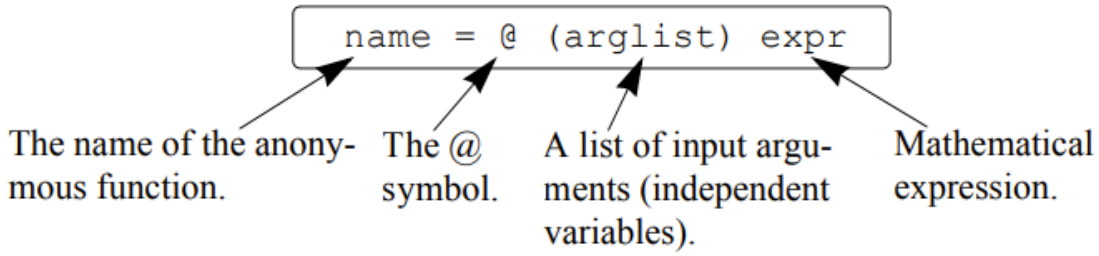
•تحتوي ملفات البرنامج النصي على سلسلة من أوامر (الجمل).

•يمكن لملف الدالة قبول البيانات من خلال متغيرات الإدخال ويمكنها إرجاع البيانات من خلال متغيرات الإخراج.

•عند حفظ ملف دالة، يجب أن يكون اسم الملف هو نفسه اسم الدالة.

6- الدالة Anonymous Functions

هي دالة بسيطة (من سطر واحد) محددة من قبل المستخدم ويتم تعريفها دون إنشاء ملف دالة منفصل (M-File يمكن إنشاء Anonymous Functions في نافذة الأوامر، أو داخل ملف نصي، أو ضمن دالة عادية محددة من قبل المستخدم. يتم إنشاء Anonymous Functions عن طريق كتابة الأمر التالي:



مثال بسيط **cube = @ (x) x^3** وهي دالة لحساب حجم المكعب بمتغيرمستقل واحد مثال آخر يتضمن متغيرين وهو **circle = @ (x, y) 16*x^2+9*y^2**

* يمكن أن يتضمن التعبير الرياضي أي دالة مضمنة أو محددة من قبل المستخدم

* يجب كتابة التعبير الرياضي وفقًا لأبعاد arglist (عنصرًا بعنصر أو حسابات الجبر الخطي).

* يمكن أن يتضمن التعبير متغيرات تم تعريفها بالفعل عند تعريف Anonymous Functions. على سبيل المثال **parabola = @ (x) a*x^2+b*x+ c** ، إذا تم تعريف ثلاثة متغيرات a وb وc (تم تعيين قيم عددية لها)، فيمكن استخدامها في التعبير عن Anonymous Functions.

لاحظ المثال أدناه لدالة $f(x) = \frac{e^{x^2}}{\sqrt{x^2+5}}$

```
>> FA = @ (x) exp(x^2)/sqrt(x^2+5)
FA =
    @(x)exp(x^2)/sqrt(x^2+5)
```

يمكن بعد ذلك استخدام الدالة لقيم مختلفة لـ x، كما هو موضح

```
>> FA(2)
ans =
   18.1994
>> z = FA(3)
z =
  2.1656e+003
```

اذا كانت x عبارة عن متجه فيكون التنفيذ كالتالي:

```
>> FA = @ (x) exp(x.^2)./sqrt(x.^2+5)
FA =
    @(x)exp(x.^2)./sqrt(x.^2+5)
>> FA([1 0.5 2])                    Using a vector as input argument.
ans =
    1.1097    0.5604   18.1994
```

مثال على Anonymous Functions مع عدة متغيرات مستقلة:

```
>> HA = @ (x,y) 2*x^2 - 4*x*y + y^2
HA =
    @(x,y)2*x^2-4*x*y+y^2
```

يمكن استخدام Anonymous Functions لقيم مختلفة لـ x و y. على سبيل المثال، كتابة HA(2,3):

```
>> HA(2,3)
ans =
    -7
```

7- الدالة Inline Functions

يمكن استخدام تعريف هذه الدالة في نافذة الامر، الملف نصي وكذلك ضمن ملف الدالة. هذه الدالة تفيد المستخدم باستخدامها في البرنامج بدون الحاجة الى إنشاء ملف دالة. يتم إنشاء Inline Functions عن طريق كتابة الأمر التالي:

```
name = inline('math expression typed as a string')
```

كمثال بسيط لحساب حجم المكعب **cube = inline ('x^3')**

* يمكن أن يحتوي التعبير الرياضي على متغير مستقل واحد أو أكثر.

* يمكن استخدام أي حرف باستثناء i و j كمتغيرات مستقلة في التعبير.

* يمكن أن يتضمن التعبير الرياضي أي دالة مدمجة أو محددة من قبل المستخدم.

* يجب كتابة التعبير وفقاً لبعد argument (عنصر بعنصر أو حسابات الجبر الخطي).

* لا يمكن أن يتضمن التعبير متغيرات معينة مسبقًا.

* بمجرد تعريف الدالة، يمكن استخدامها عن طريق كتابة اسمها وقيمة argument بين قوسين.

* يمكن استخدام الدالة المضمنة كمتغير في دوال أخرى

لاحظ المثال أدناه لدالة $f(x) = \frac{e^{x^2}}{\sqrt{x^2+5}}$

```
>> FA=inline('exp(x.^2)./sqrt(x.^2+5)')
FA =
     Inline function:
     FA(x) = exp(x.^2)./sqrt(x.^2+5)
>> FA(2)
ans =
   18.1994
>> FA([1 0.5 2])
ans =
    1.1097    0.5604   18.1994
```

Expression written with element-by-element operations.

Using a scalar as the argument.

Using a vector as the argument.

مثال من عدة متغيرات مستقلة

```
>> HA=inline('2*x^2-4*x*y+y^2')
HA =
     Inline function:
     HA(x,y) = 2*x^2-4*x*y+y^2
```

يمكن استخدام Inline Functions لقيم مختلفة لـ x و y. على سبيل المثال، كتابة HA(2,3):

```
>> HA(2,3)
ans =
    -7
```

8- دالة FUNCTION FUNCTIONS

هناك العديد من الحالات التي تعمل فيها الدالة (Function A) في دالة أخرى (Function B). هذا يعني أنه عند تنفيذ الدالة A، يجب تزويدها بالدالة B. تسمى الدالة التي تقبل دالة أخرى في MATLAB دالة الدالة Function functions.

أ- استخدام الدالة الماسكة لتمريرها كدالة الدالة

تُستخدم الدالة الماسكة لتمرير (استيراد) دوال المحددة من قبل المستخدم، والدوال المضمنة، والدوال المجهولة إلى دوال دالة(Function functions) يمكنها قبولها.

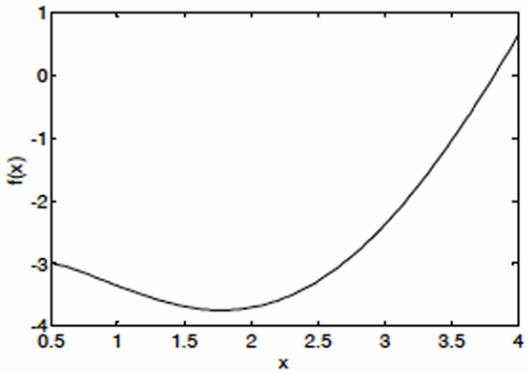
- يتم إنشاء الدالة الماسكة وذلك بوضع @ امام الدالة الجاهزة او الدالة المعرفة من قبل المستخدم. مثال على ذلك @cos هي الدالة الماسكة لدالة الجاهزة cos وكذلك @Ftoc هي الدالة الماسكة لدالة الجاهزة Ftoc.
- يمكن توظيف الدالة الماسكة الى إسم متغيير. مثال على ذلك cosHandle=@cos يمكن استخدام هذه الدالة لتميرها كدالة ماسكة.
- وتعتبر الدالة المجهولة دالة ماسكة

لاحظ المثال التالي بطريقتين، الطريقة الأولى هي انشاء دالة معرفة من قبل المستخدم

```
function y=Fdemo(x)
y=exp(-0.17*x).*x.^3-2*x.^2+0.8*x-3;
```

```
>> ydemo=funplot(@Fdemo,0.5,4)
ydemo =
    0.5000   -2.9852
    2.2500   -3.5548
    4.0000    0.6235
```

Enter a handle of the user-defined function Fdemo.



في حالة استخدامها كدالة مجهولة

```
>> FdemoAnony=@(x) exp(-0.17*x).*x.^3-2*x.^2+0.8*x-3
FdemoAnony =
    @(x) exp(-0.17*x).*x.^3-2*x.^2+0.8*x-3

>> ydemo=funplot(FdemoAnony,0.5,4)
ydemo =
    0.5000   -2.9852
    2.2500   -3.5548
    4.0000    0.6235
```

Create an anonymous function for $f(x)$.

Enter the name of the anonymous function (FdemoAnony).

ب- استخدام اسم الدالة لتمريرها كدالة إلى دالة الدالة

الطريقة الثانية لتمرير دالة إلى دالة هي عن طريق كتابة اسم الدالة التي يتم استيرادها كمتغير حرفي في دالة الدالة. يمكن استخدام الطريقة التي تم استخدامها قبل تقديم الدالة الماسكة لاستيراد الدوال المحددة من قبل المستخدم مثال على ذلك الدالة feval

عندما يتم استيراد دالة محددة من قبل المستخدم باستخدام اسمها، يجب حساب قيمة الدالة المستوردة داخل دالة الدالة باستخدام الأمر feval. يختلف هذا عن الحالة التي يتم فيها استخدام الدالة الماسكة، مما يعني أن هناك اختلافاً في طريقة كتابة التعليمات البرمجية في الدالة التي تعتمد على كيفية تمرير الدالة المستوردة. تكون صيغة العامة كالتالي:

```
variable = feval('function name', argument value)
```

يمكن تعيين القيمة التي يتم تحديدها بواسطة feval إلى متغير، أو إذا تمت كتابة الأمر بدون إسناد الى متغير، يعرض MATLAB (ans=) وقيمة الدالة.

- تتم كتابة اسم الدالة كمتغير حرفي. من الممكن توظيف feval الى اسم متغير وكما يمكن عدم توظيف الامر.
- يمكن أن تكون دالة الدالة جاهزة أو محددة من قبل المستخدم.
- إذا كان هناك أكثر من متغير إدخال، يمكن اضافة المتغيرات ويتم الفصل بينهما باستخدام الفاصلة
- إذا كان هناك أكثر من متغير إخراج، فإن المتغيرات يتم اضافتها داخل اقواس مربعة وتفصل بينهما فارزة

مثال

```
>> feval('sqrt',64)
ans =
     8
>> x=feval('sin',pi/6)
```

```
x =
    0.5000
```

مثال

```
>> [M,T]=feval('loan',50000,3.9,10)
M =
        502.22
T =
      60266.47
```

A $50,000 loan, 3.9% interest, 10 years.

Monthly payment.

Total payment.

9- الدوال الفرعية Subfunction

يمكن أن يحتوي ملف الدالة على أكثر من دالة محددة من قبل المستخدم. يتم كتابة الدوال واحد تلو الأخر. تبدأ كل دالة بخط تعريف الدالة. الدالة الأولى تسمى الدالة الأساسية وبقية الدوال تسمى الدوال الفرعية. يمكن كتابة الدوال الفرعية بأي ترتيب. يجب أن يتوافق الملف المحفوظ مع اسم الوظيفة الأساسية. يمكن لكل دالة في الملف استدعاء أي من الدوال الأخرى في الملف. يمكن لدوال الخارجية أو البرامج (ملفات البرامج النصية) استدعاء الدوال الأساسية فقط. كل دالة في الملف لها مساحة عمل خاصة بها، مما يعني أن المتغيرات محلية في كل منها. بمعنى آخر، لا يمكن للدالة الأساسية والدوال الفرعية الوصول إلى متغيرات بعضها البعض (ما لم يتم الإعلان عن المتغيرات على أنها عالمية).

مثال بسيط حول المتوسط الحسابي والإنحراف المعياري لقائمة الدرجات 95 80 65 89 79 60 91 75 80 81 50

```
function [me SD] = stat(v)
n=length(v);
me=AVG(v,n);
SD=StandDiv(v,me,n);

function av=AVG(x,num)
av=sum(x)/num;

function Sdiv=StandDiv(x,xAve,num)
xdif=x-xAve;
xdif2=xdif.^2;
Sdiv= sqrt(sum(xdif2)/(num-1));
```

The primary function.

Subfunction.

Subfunction.

في نافذة الأمر

```
>> Grades=[80 75 91 60 79 89 65 80 95 50 81];
>> [AveGrade StanDeviation] = stat(Grades)
AveGrade =
   76.8182
StanDeviation =
   13.6661
```

10- الدوال المتداخلة NESTED FUNCTIONS

الدالة المتداخلة هي دالة محددة من قبل المستخدم ومكتوبة داخل دالة أخرى محددة من قبل المستخدم ايضاً. يبدأ جزء الكود الذي يتوافق مع الدالة المتداخلة بسطر تعريف الدالة وينتهي ببيان النهاية end.

ادناه الصيغة العاملة لدالة متداخلة واحدة:

```
function y=A(a1,a2)
.......
    function z=B(b1,b2)
    .......
    end
.......
end
```

تنسيق الدالة المعرفة من قبل المستخدم (A تسمى الدالة الأساسية) التي تحتوي على دالة متداخلة واحدة B:

ادناه الصيغة العاملة لدالتين متداخلتين:

```
function y=A(a1,a2)
.......
    function z=B(b1,b2)
    .......
    end
.......
    function w=C(c1,c2)
    .......
    end
.......
end
```

الدوال الثلاثة اعلاه كل دالة تكون ضمن مساحة عمل الدالة الأخرى ويمكن ان تستدعي كل دالة من قبل الأخرى.

ادناه الصيغة العاملة لمستويين من الدوال المتداخلة:

```
function y=A(a1,a2)              (Primary function A.)
.......
    function z=B(b1,b2)          (B is nested function in A.)
    .......
        function w=C(c1,c2)      (C is nested function in B.)
        .......
        end
    end
    function u=D(d1,d2)          (D is nested function in A.)
    .......
        function h=E(e1,e2)      (E is nested function in D.)
        .......
        end
    end
.......
end
```

* الدالة المتداخلة يمكن استدعائها من قبل الدالة في المستوى العلوي (اي الدالة A تستدعي B او D ولكن لا يمكن استدعاء الدالتين C و E)

* يمكن استدعاء دالة متداخلة من دالة متداخلة على نفس المستوى داخل الوظيفة الأساسية. (في المثال السابق، يمكن للدالة B استدعاء D، ويمكن لـ D استدعاء B).

* يمكن استدعاء دالة متداخلة من دالة متداخلة في أي مستوى أقل.

* يتم التعرف على المتغير المعرف في الدالة الأساسية ويمكن إعادة تعريفه بواسطة الدالة متداخلة على أي مستوى داخل الدالة الأساسية.

مثال/

أكتب ملف الدالة لدالة $y\ (x) = -0.2x^4 + e^{-0.5x}\ x^3 + 7x^2$ استخدم الدالة لإيجاد قيمة y(x)

a) استخدم الدالة لحساب y(-2.5) و y(3)
b) استخدم الدالة لرسم الدالة y(x) للفترة $-3 \leq x \leq 4$

In script File

ggg.m | HW7_2.m

```
clear
clc
clear, clc
disp('Part (a)')
x=[-2.5 3];
y=HW7_2(x);
disp('The test values for y(x) are:')
disp(y)
%
%part b
x=-3:0.1:4;
plot(x,HW7_2(x));
title('y(x)=-0.2*x.^4+exp(-0.5*x).*x.^3+7*x.^2')
xlabel('x-->')
ylabel('y-->')
```

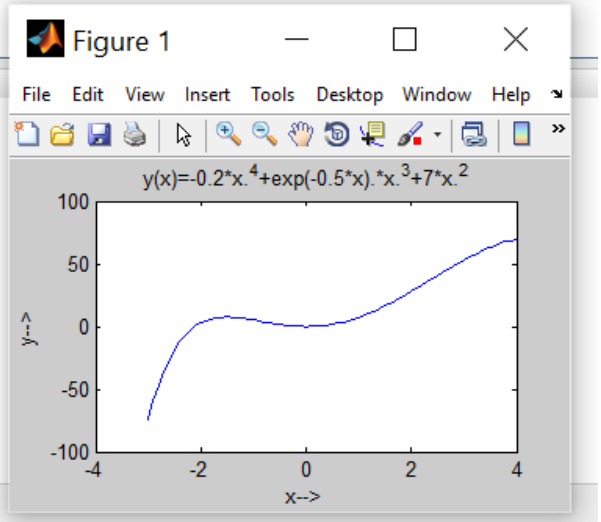
In Function file:

ggg.m | HW7_2.m

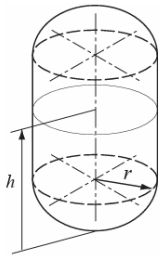
```
function y=HW7_2(x)
 y=-0.2*x.^4+exp(-0.5*x).*x.^3+7*x.^2;



end
```

In command window

```
Command Window
  Part (a)
  The test values for y(x) are:
    -18.5991    52.8245

fx >>
```



مثال/يحتوي خزان الوقود العمودي الأسطواني على أغطية نهائية نصف كروية كما هو موضح. نصف قطر الاسطوانة والأغطية r = 15 إنج، وارتفاع الجزء الأوسط الأسطواني 40 إنج. استخدم تعريف الدالة التي تحسب حجم الوقود (بالجالون) كدالة للأرتفاع h، واستخدم لرسم البياني كدالة للارتفاع ل $0 \le h \le 70$ إنج.



استخدم الدالة التالية V = Volfuel(h)

User-defined function:

VolFuel.m | CH_7_7.m

```
function V=VolFuel(h)
  r=15; H=40;
  if h <= r
     V=pi*h^2*(3*r-h)/3;
  elseif h > r & h <= (r+H)
     V=2*pi*r^3/3+pi*r^2*(h-r);
  elseif h > (H+r) & h <= (H+2*r)
     V=4*pi*r^3/3+pi*r^2*H-pi*(H+2*r-h)^2*(r-H+h)/3;
  end
```

Script file:

Editor - C:\Users\Marwah\Documents\MATLAB\CH_7

VolFuel.m | CH_7_7.m

```
clear,clc
 h=0:0.25:70;
 n=length(h);
 for i=1:n
 V(i)=VolFuel(h(i));
 end
 plot(h,V)
 xlabel('h (in.)')
 ylabel('V (in.^3)')
```