Matlab – 自訂函數 function

- 自訂函數（Function）基本語法
- 選擇性傳入或回傳變數
- 次函數（subfunction）
- 函數的相關設定：變數傳遞、目錄
- 匿名函數

Matlab的自訂函數（Function）

- Matlab內建函數與自訂函數都是一種.m檔案
  - 可以用type來看函數的程式碼，例如 type contour.m
- 自訂函數可以接受主程式傳進來的變數，並且將計算結果回傳
- 函數取名的限制（變數名的限制相同）
  - 以英文字母作為開頭，最多 31 個（MATLAB 5.x）或**63**個（MATLAB 6.x 和 7.x）字元
- 函數名稱最好和.m檔案名稱相同：
  - 如果函數名稱和.m檔案名稱不同，要使用.m檔案名稱呼叫函數，而函數名稱將被忽略

Matlab自訂函數的語法

```
function out_var=F_NAME(in_var)
%(This line shows up in "lookfor")
%(Comments explaining the contents and usage of
% this function, will  show up in "help")
%

out_var=...   % Execution section of the function
```

- 第一行用來定義函數，以**function**（全小寫）開頭
- **out_var**：函數計算回傳的結果變數，
- **F_NAME**：函數的名字，最好和.m檔案的名字相同
- **in_var**：由主程式傳入的變數清單
- **%**：註解，解釋自訂函數的功能和用法。第一行會在**lookfor**查詢時顯示，整個連續的註解會在**help**查詢時顯示

自訂函數：範例一

```
function average =avevec(vector)
% avevec computes the average of all elements
% in a vector
%
% Usage of this function: output = avevec(input)
% "output" is the average of all elements in
% the input vector "input".

% compute average
average = sum(vector)/length(vector);
```

使用結果:

```
>> lookfor avevec
 avevec computes the average value of all elements in a vector

>> result = avevec ([1 5 3])
result =
    3
```

自訂函數：範例二 傳入、回傳多個變數

```
function [ave1, ave2] =ave2vec(vec1, vec2)
% ave2vec computes average values of two vectors
%
% Usage of this function:
% [out1 out2]=ave2vec(in1, in2)
% "out1 out2" are the average values of the input
% vector2 "in1 in2".
%

ave1 = sum(vec1)/length(vec1); % average of vec1
ave2 = sum(vec2)/length(vec2); % average of vec2
```

使用結果:

```
>> [a, b] = ave2vec ([1 2 3], [4 5 6 7 8])
a =
    2
b =
    6
```

自訂函數：選擇性傳入、回傳某些變數

- 之前教過的一些Matlab函數（指令），有些輸入的變數是可以選擇性 (optional) 提供的
（例如：plot(x,y,'-k')，第三個設定線條顏色格式的變數就算省略，指令仍然可以順利執行）
自訂函數也可以這樣設定

- 基本步驟：
  - 在函數一開始先使用
內建變數**nargin** ，取得主程式傳進函數的變數個數，或
內建變數**nargout**，取得主程式要求函數回傳的變數個數
  - 自訂函數內利用邏輯判斷（if），由輸入、輸出變數的數目，分別決定不同的運算方式。

自訂函數：範例三 選擇性傳入、回傳變數

```
function [mag, angle] =polar_value(x, y)
% polar_value converts (x,y) to polar coordinate
%  angle - output angel in degree (optional)
%  mag - output magnitude
%  x - input x value
%  y - input y value (optional)

% if 2nd input is missing, set it to 0
if nargin == 1
  y=0;
end

mag=sqrt(x.^2+y.^2)   % calculate magnitude

%if 2nd output is requested
if nargout == 2
  angle=atan2(y,x) * 180/pi;  % calculate angle
end
```

選擇性傳入、回傳某些變數：測試結果

```
>> [m, a] = polar_value
??? Error using ==> polar_value
Not enough input arguments

>>[m, a] = polar_value(1,-1,1)
??? Error using ==> polar_value
Too many input arguments

>>[m, a] = polar_value(1)
m =
    1
a =
    0

>>m = polar_value(1,-1)
m =
   1.4142
```

← 輸入變數太少，自動傳回錯誤訊息

← 輸入變數太多，自動傳回錯誤訊息

← 可接受只傳入一個變數

← 可接受傳入兩個變數，只回傳一個變數

次函數（Subfunctions）

- 一個 M 檔案可以包含一個以上的函數
  - 一個主函數（Primary Function），寫在m檔案的開頭
  - 其他則為次函數（Subfunctions）
  - 次函數只能被同一個m檔案中的函數（主函數或次函數）呼叫，但無法被不同檔案呼叫
- 主函數與次函數的位置
  - 主函數必需出現在最上方
  - 主函數後為任意數目的次函數
  - 次函數的次序並無任何限制

主程式與自訂函數的變數傳遞

- 自訂函數與主程式的變數工作空間（work space）是獨立的
- 被傳進自訂函數中的變數，就算數值在自訂函數中被修改，也不會影響同個變數在主程式中的值（傳值不傳址）

- 這一點與Fortran的變數傳遞設定（傳址不傳值）不同

自訂函數的目錄

- 如果希望某個自訂函數在任何目錄內均可執行，須將存放此函數m檔案的目錄，加入MATLAB的搜尋路徑中：
  - 可將功能相關的自訂函數，存放在同一個目錄內
  - 使用 addpath 指令將此目錄加入搜尋路徑
- 如果希望某些函數不會被其他函數呼叫，要存放在私有化資料夾（Private Directory）中：
  - 建立名稱為 private 的私有化資料夾
  - 在private資料夾內的函數，只能被上層資料夾的程式所呼叫，不能被其他目錄的函數呼叫

函數搜尋次序

- 從 M 檔案呼叫一個函數時，MATLAB 會依下面的順序搜尋函數是否存在：
  - 檢查此函數是否為次函數（在同一個.m檔案中）
  - 再檢查此函數是否在私有化（private）目錄中
  - 最後檢查此函數是否在系統的搜尋路徑中

- MATLAB 找到第一個檔名相符的函數，會立即取用

匿名函數 Anonymous Function

- **從Matlab 2011b之後的版本**，提供「匿名函數」的功能
- 如果函數的只有一行數學運算式，不需要另寫m檔，在指令視窗或主程式的m檔，即可設定匿名函數，語法如下：

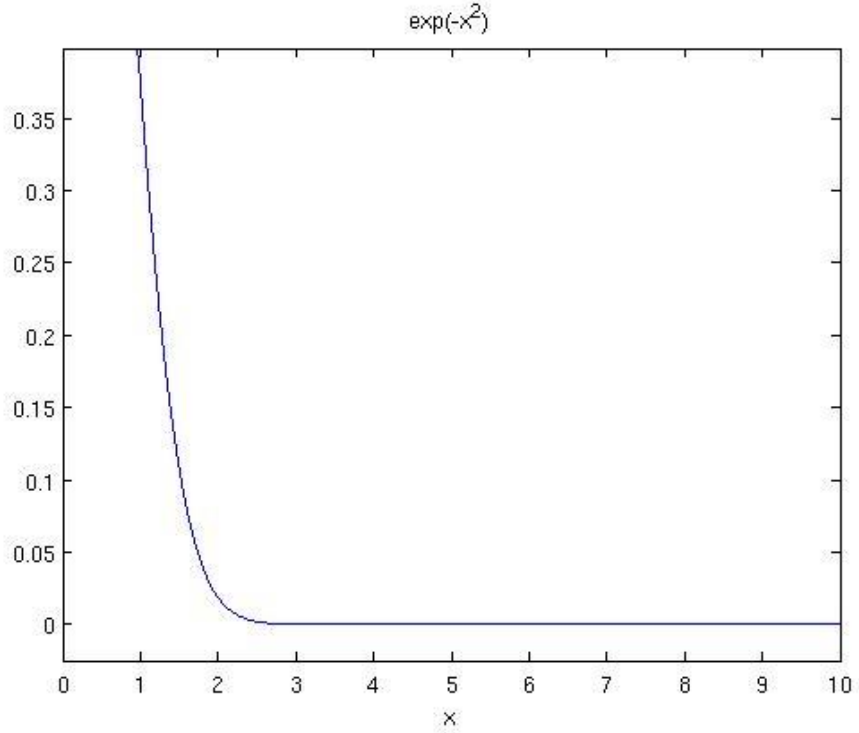
Fun = @(x) x.^2+sin(x)+1

  - **Fun**：匿名函數名稱（呼叫函數時使用）
  - **@(x)**：傳入函數的變數清單（x可以是純量或向量）
  - **x.^2+sin(x)+1**：函數的數學運算式（建議使用向量element-to-element的運算符號，才能處理x為向量的情形）

- 呼叫匿名函數的方法與呼叫一般自訂函數相同，以上面設定的匿名函數為例
  - **Y=Fun(10)** – 把x=10代入函數，回傳結果Y為一純量
  - **Y=Fun([1,10,100])** – 把x向量代入函數，回傳結果Y為向量，長度與x相同（分別把x=1, 10 ,100代入函數的結果）

匿名函數：範例

```
>> f=@(x) exp(-x.^2);
>> ezplot(f, [0,10])
```




```
>> myfun = @(x,y) (x.^2 + y.^2 + x.*y);
>> a = 1; b = 10;
>> z = myfun(a,b)
```

z = 111