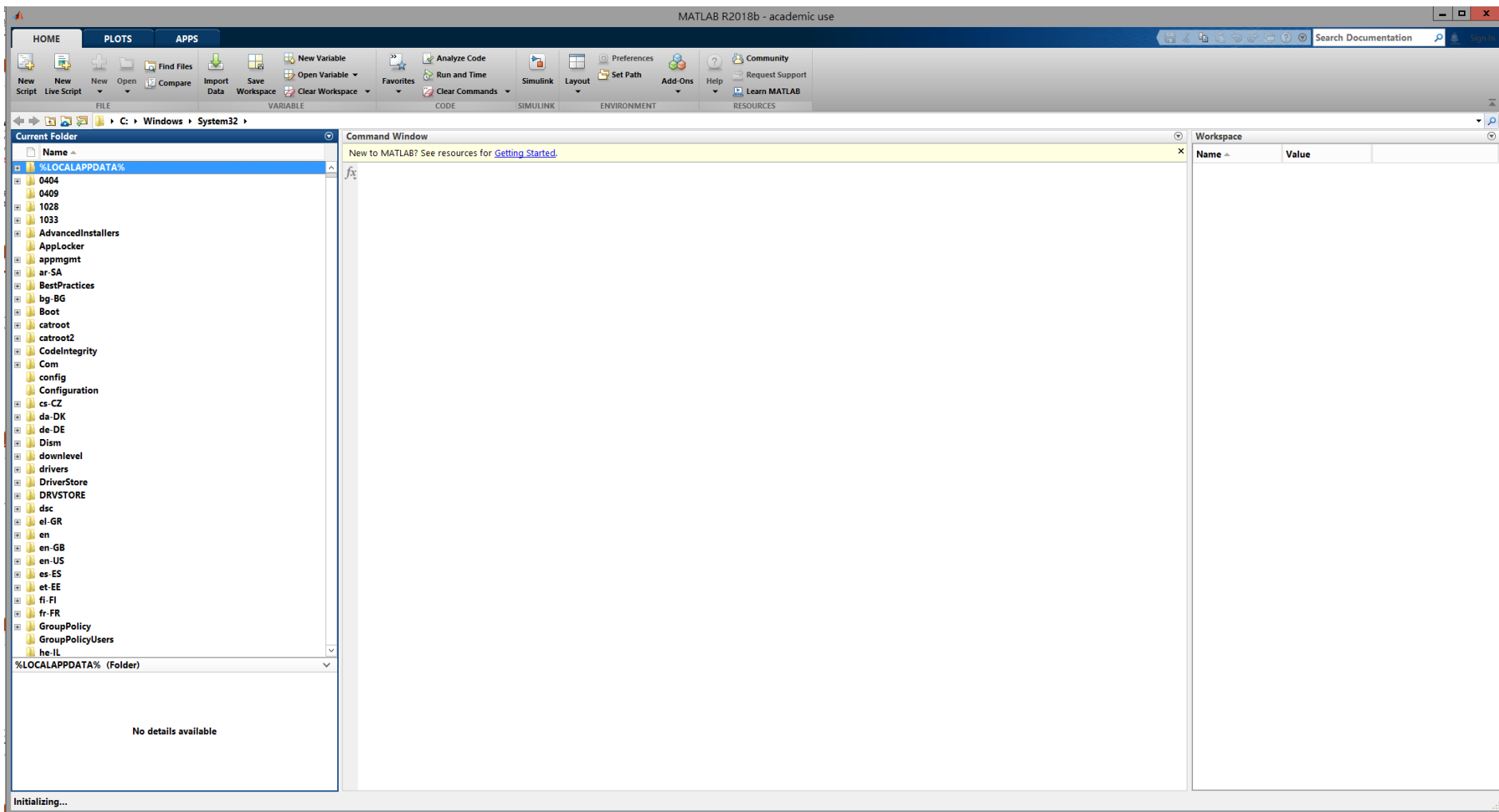


雲端電腦教室

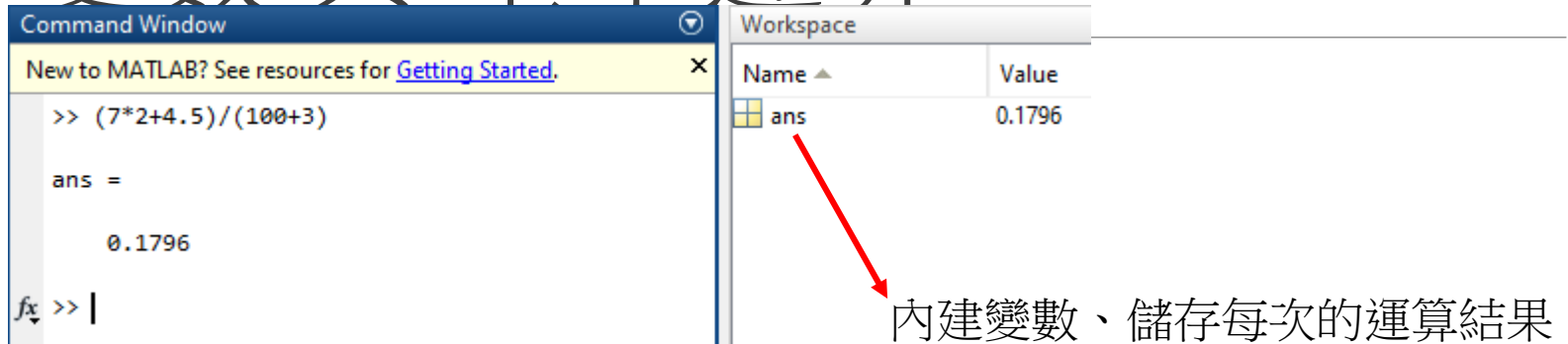
[HTTP://IC.CGU.EDU.TW/FILES/14-1018-37731,R40-1.PHP?LANG=ZH-TW](http://ic.cgu.edu.tw/files/14-1018-37731,R40-1.php?lang=zh-tw)



Matlab#1

INTRODUCTION

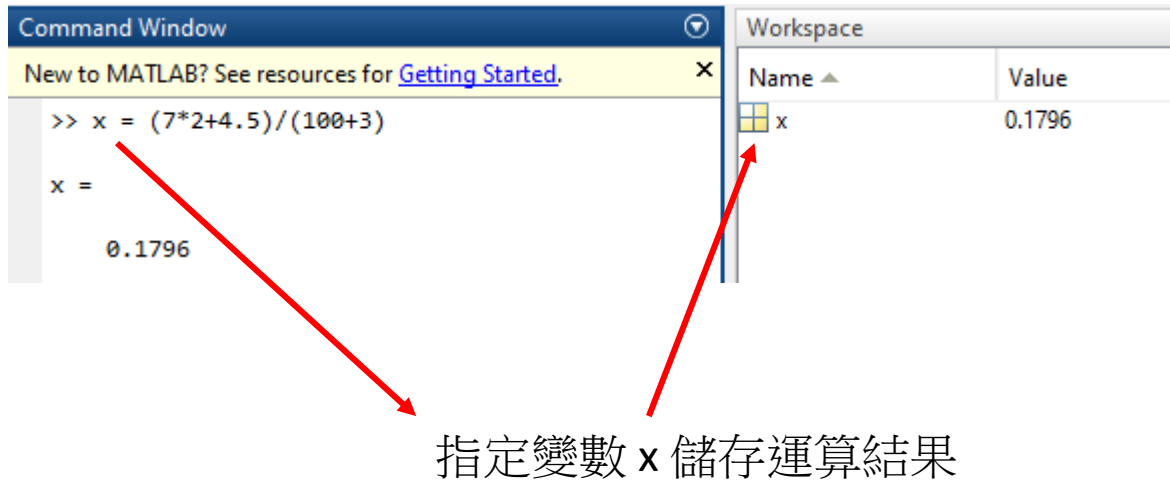
變數與基本運算



The screenshot shows the MATLAB Command Window and Workspace. In the Command Window, the command `>> (7*2+4.5)/(100+3)` has been executed, resulting in `ans = 0.1796`. The Workspace window on the right displays a table with two columns: 'Name' and 'Value'. The table contains one entry: 'ans' with a value of 0.1796. A red arrow points from the 'ans' variable in the Workspace to the text '內建變數、儲存每次的運算結果'.

Name	Value
ans	0.1796

內建變數、儲存每次的運算結果



The screenshot shows the MATLAB Command Window and Workspace. In the Command Window, the command `>> x = (7*2+4.5)/(100+3)` has been executed, resulting in `x = 0.1796`. The Workspace window on the right displays a table with two columns: 'Name' and 'Value'. The table contains one entry: 'x' with a value of 0.1796. Two red arrows point from the text '指定變數 x 儲存運算結果' to the 'x' variable in the Command Window and the 'x' variable in the Workspace.

Name	Value
x	0.1796

指定變數 x 儲存運算結果

變數命名規則

第一個字必需是英文，後面可以接數字或是底線

- eg. **a1234**, **a_variable**, **a001**

最多 31 個字母

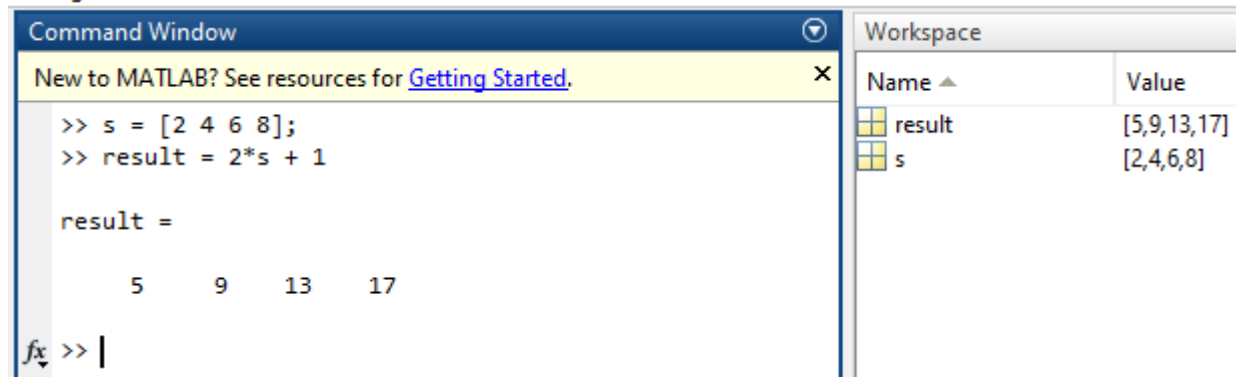
不需宣告變數

數值變數預設均以double 資料型態(8 bytes)儲存

註解

```
fx >> x = (7*2+4.5)/(100+3) % This is comment |
```

向量 (Vectors) & 矩陣 (Matrix)



The image shows a screenshot of the MATLAB interface. The Command Window on the left contains the following code and output:

```
>> s = [2 4 6 8];  
>> result = 2*s + 1  
  
result =  
  
     5     9    13    17  
fx >> |
```

The Workspace window on the right shows the following variables:

Name	Value
result	[5,9,13,17]
s	[2,4,6,8]

矩陣處理

Workspace	
Name ▲	Value
result	[5,9,13,17]
s	[2,4,6,8]

```
>> s(2) = 10    % 將向量 s 的第2個元素更改為 10  
s =  
    2    10     6     8  
fx >> |
```

```
>> s(8)=99      % 在向量 s 加入第8個元素，其值為 99, 注意會補 0  
s =  
    2    10     6     8     0     0     0    99  
fx >> |
```

```
>> s(2) = []    % 刪除第2個元素  
s =  
    2     6     8     0     0     0    99
```


建立大小為 $m \times n$ 的矩陣

在row結尾加上分號(;)

```
>> A = [1 2 3 4; 5 6 7 8; 9 7 6 4];
```

% 建立 3×4 的矩陣 A

```
>> A
```

% 顯示矩陣 A 的內容

```
A =
```

```
1   2   3   4
```

```
5   6   7   8
```

```
9   7   6   4
```

矩陣的內部儲存方式

MATLAB的二維矩陣的內部儲存方式是column-major (C/C++語言則是row-major)

- MATLAB的二維矩陣可以看成是行向量的集合
- 例如：

```
>> A = [1 2 3; 5 6 7; 9 7 6];
```

% 建立 3x3 的矩陣 A

```
>> A
```

% 顯示矩陣 A 的內容

A =

1 2 3

5 6 7

9 7 6



內部儲存方式：
1 5 9 2 6 7 3 7 6

mxn矩陣的各種處理 (1)

```
>> A(2,3) = 5    % 將矩陣 A 第二列、第三行的元素值，改變為 5
```

```
A =
```

```
    1    2    3    4
    5    6    5    8
    9   10   11   12
```

```
>> B = A(2,1:3)  % 取出矩陣 A 的第二橫列、第一至第三直行，並儲存成矩陣 B
```

```
B =
```

```
    5    6    5
```

mxn矩陣的各種處理 (2)

```
>> A = [A B']      % 將矩陣 B 轉置後、再以行向量併入矩陣 A
```

```
A =
```

1	2	3	4	5
5	6	5	8	6
9	10	11	12	5

```
>> A(:, 2) = []     % 刪除矩陣 A 第二行（：代表所有橫列，[]代表空矩陣）
```

```
A =
```

1	3	4	5
5	5	8	6
9	11	12	5

mxn矩陣的各種處理 (3)

>> A = [A; 4 3 2 1] % 在原矩陣 A 中，加入第四列

A =

1	3	4	5
5	5	8	6
9	11	12	5
4	3	2	1

>> A([1 4], :) = [] % 刪除第一、四列（：代表所有直行，[]是空矩陣）

A =

5	5	8	6
9	11	12	5

mxn矩陣的各種處理 (4)

```
>> B = A(:, [4 2 3])           % 從矩陣 A 抽出第4, 2, 3行
```

```
B =
```

```
    6    5    8
```

```
    5   11   12
```

M 檔案編輯器 (I)

M 檔案是文字檔

- 可以用各種文字編輯器修改
- 儲存時，需以文字模式儲存

MATLAB 在 Windows 及 Mac 平台上，提供了內建的「M 檔案編輯器」（M-File Editor）

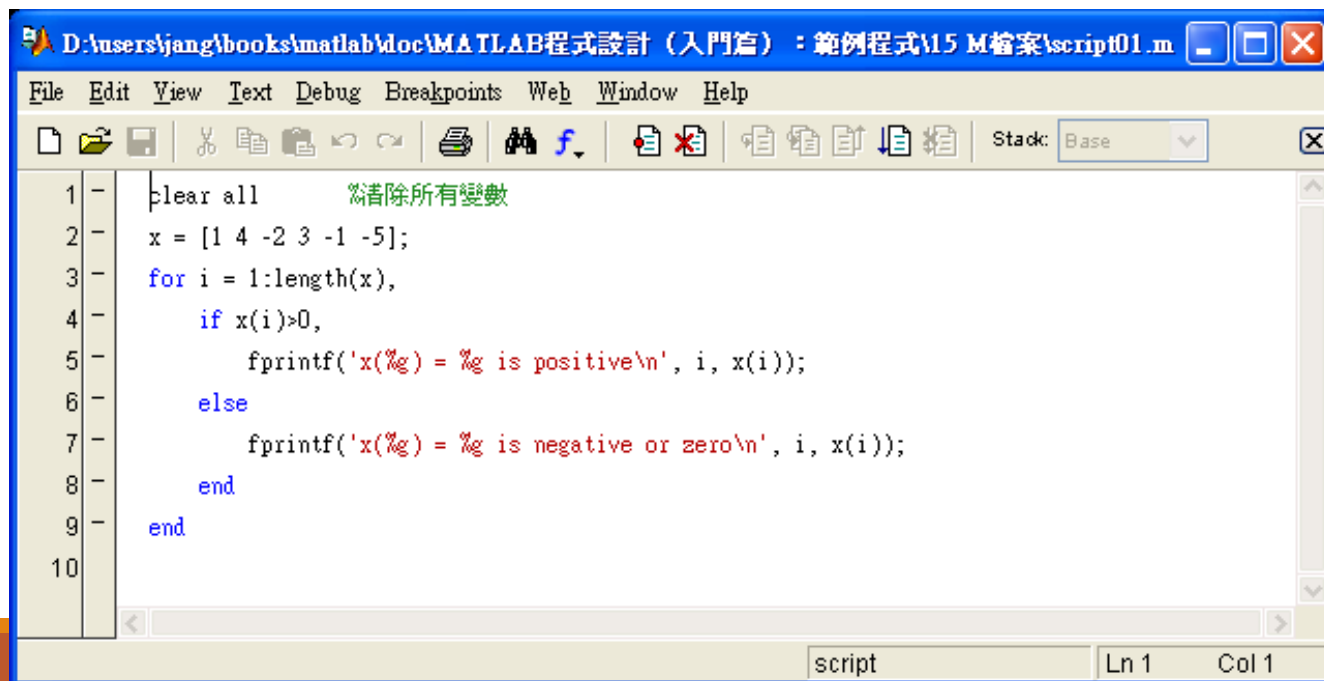
- 點選指令視窗的 `file/open` 下拉式選單，開啟 M 檔案編輯器
- 或在指令視窗直接鍵入「`edit filename.m`」或「`open filename.m`」

M 檔案編輯器 (II)

開啟 Script01.m，可輸入

>> edit script01.m

即可開啟 M 檔案編輯器：



The image shows a screenshot of the MATLAB M-file editor window. The title bar indicates the file path: D:\users\jiang\books\matlab\doc\MATLAB程式設計 (入門篇) : 範例程式\15 M檔案\script01.m. The menu bar includes File, Edit, View, Text, Debug, Breakpoints, Web, Window, and Help. The toolbar contains various icons for file operations and editing. The main editing area displays the following MATLAB code:

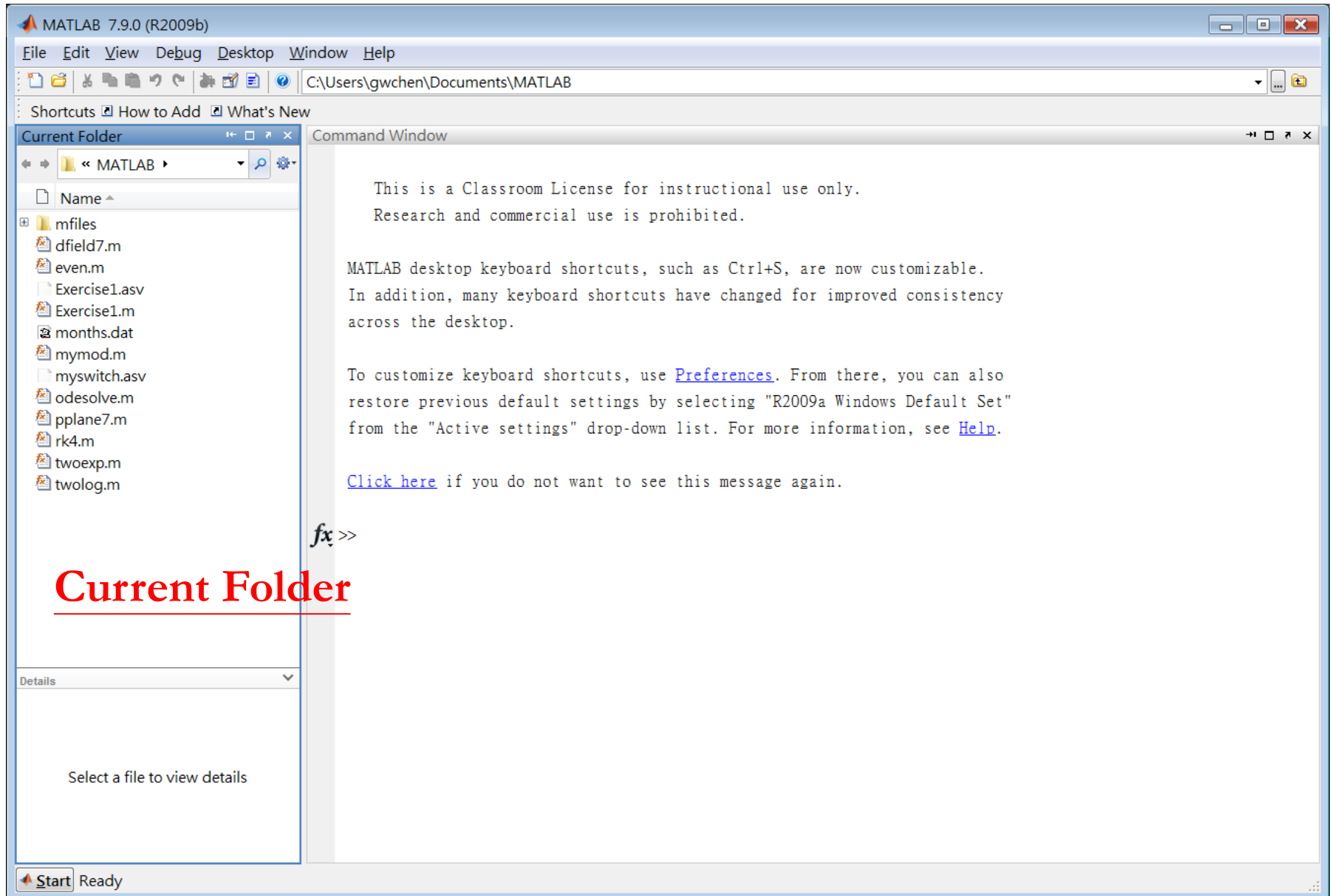
```
1 - clear all      %清除所有變數
2 - x = [1 4 -2 3 -1 -5];
3 - for i = 1:length(x),
4 -     if x(i)>0,
5 -         fprintf('x(%g) = %g is positive\n', i, x(i));
6 -     else
7 -         fprintf('x(%g) = %g is negative or zero\n', i, x(i));
8 -     end
9 - end
10
```

The status bar at the bottom shows 'script' and 'Ln 1 Col 1'.

函數

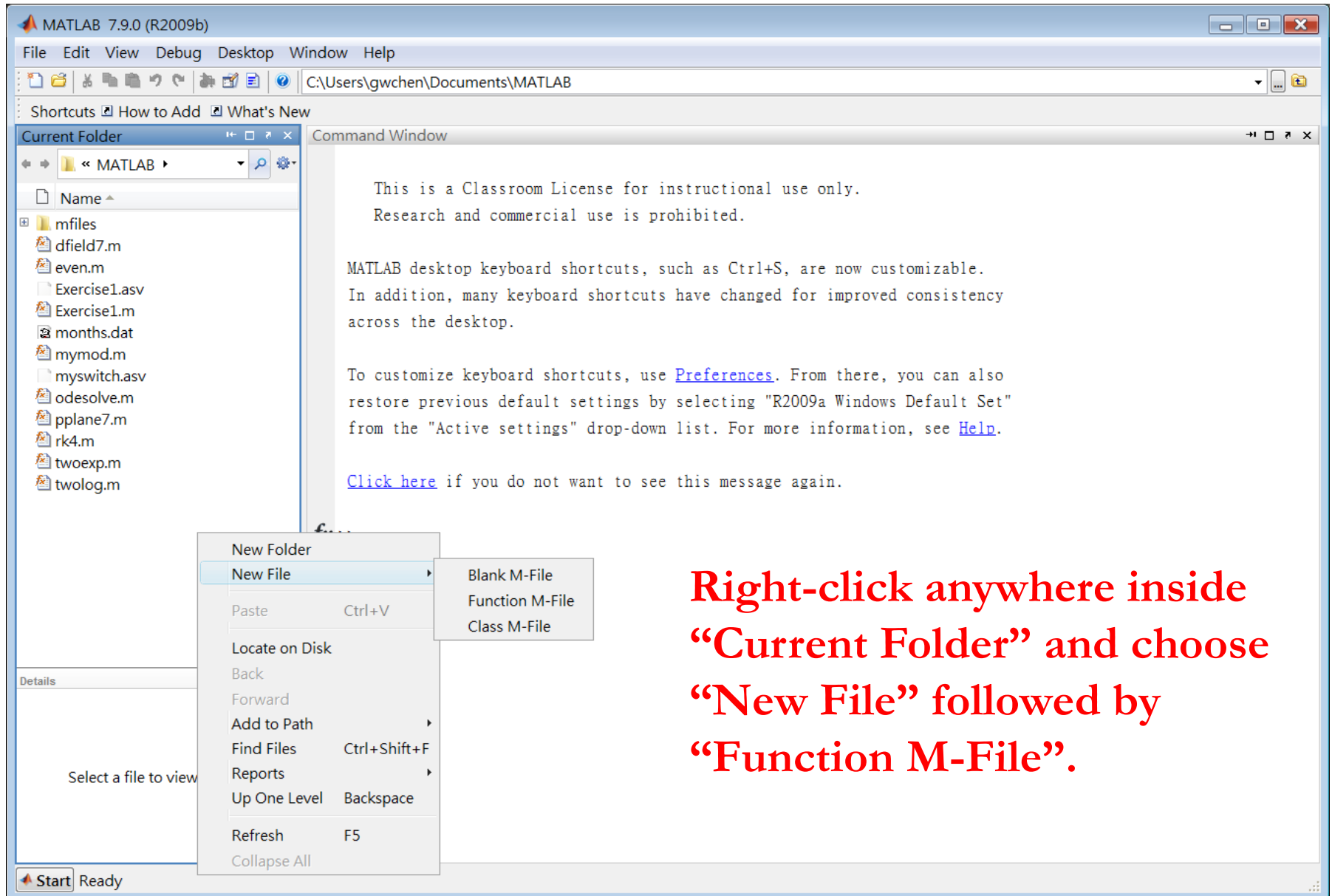
函數

- 也是 **M** 檔案的一種
- 可接受輸入變數，並將結果送至輸出變數
- 運算過程產生的變數都存放在函數本身的工作空間
 - 不會和 **MATLAB** 基本工作空間的變數相互覆蓋
- 函數適用於大型程式碼
 - 使程式碼模組化（**Modularized**）並易於維護與改進



Current Folder

fx >>



The screenshot displays the MATLAB environment with three main panels:

- Current Folder:** Shows a list of files including 'Name', 'CSIE.m', and 'gasprices.xls'. An orange arrow points from the text to the 'Untitled.m' file (not explicitly labeled but implied by the context).
- Command Window:** Displays the prompt 'fx >>'.
- Workspace:** Shows a table with columns 'Name' and 'Value'.

CSIE.m (Function)

CISE Summary of this function goes here

CSIE(input_args)

Workspace

Name	Value
------	-------

A new file “Untitled.m” is created. You may rename it to, for example, CSIE.m. Double-click it will open the template for editing.

MATLAB R2016a - academic use

HOME PLOTS APPS EDITOR PUBLISH VIEW

New Open Save Find Files Compare Print Go To Find Insert Comment Indent Breakpoints Run Run and Advance Run Section Advance Run and Time

FILE NAVIGATE EDIT BREAKPOINTS RUN

Current Folder: \clientc\$\Users\Ching\OneDrive\課程演講\10701 工程數學\MATLAB 投影片\10701_mm_matlab_01

CSIE.m CSIE2.m

```
1 function [ SUM ] = CSIE( initial, N )
2 %CSIE Summary of this function goes here
3 % Detailed explanation goes here
4
5 SUM = 0;
6 for i = initial : (N + initial - 1 )
7     SUM = SUM + i;
8
9 end
```

Workspace: ans

CSIE2.m (Function)

CSIE Summary of this function goes here

CSIE(initial, N)

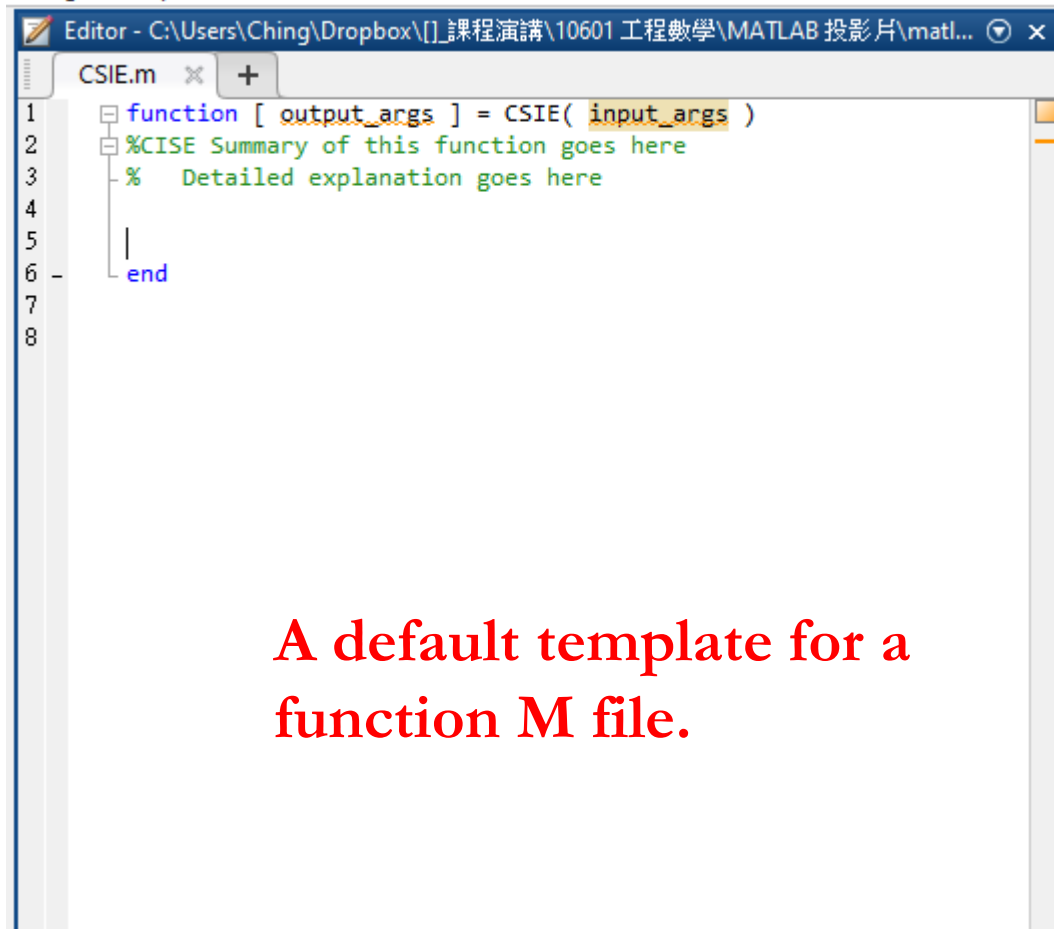
Command Window

New to MATLAB? See resources [Getting Started](#)

```
>> CSIE(10,20)
ans =
    390
fx >>
```

路徑都要對，可以選擇本機電腦的路徑 (要選 Add path)

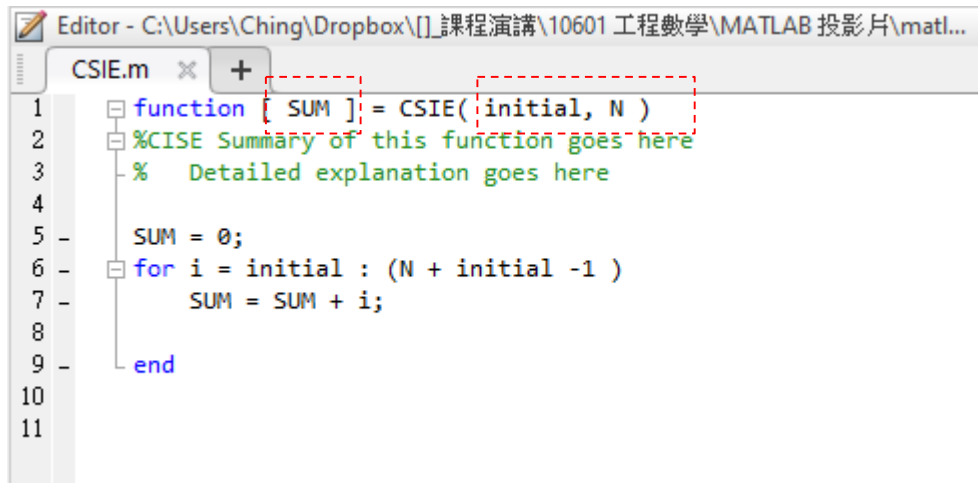
執行的時候直接輸入 function 的名子+參數 CSIE (10, 20)



The image shows a MATLAB Editor window with the title bar "Editor - C:\Users\Ching\Dropbox\[]_課程演講\10601 工程數學\MATLAB 投影片\matl...". The window contains a single tab for the file "CSIE.m". The code in the editor is as follows:

```
1 function [ output_args ] = CSIE( input_args )
2 %CISE Summary of this function goes here
3 % Detailed explanation goes here
4
5 |
6 end
7
8
```

**A default template for a
function M file.**



```
Editor - C:\Users\Ching\Dropbox\[ ]_課程演講\10601 工程數學\MATLAB 投影片\matl...
CSIE.m  x  +
1  function [ SUM ] = CSIE( initial, N )
2  %CISE Summary of this function goes here
3  % Detailed explanation goes here
4
5  SUM = 0;
6  for i = initial : (N + initial -1 )
7      SUM = SUM + i;
8
9  end
10
11
```

```

1  function [ SUM ] = CSIE( initial, N )
2  %CISE Summary of this function goes here
3  % Detailed explanation goes here
4
5  SUM = 0;
6  for i = initial : (N + initial -1 )
7      SUM = SUM + i;
8
9  end
10
11

```

Command Window

```

>> help CSIE
CISE Summary of this function goes here
Detailed explanation goes here

```

```
>> CSIE (1, 10)
```

1+2+3+4+5+6+7+8+9+10

```
ans =
```

```
55
```

```
>> CSIE (6, 1)
```

6

```
ans =
```

```
6
```

```
>> CSIE (6, 2)
```

6+7

```
ans =
```

```
13
```

```
>> CSIE (6, 5)
```

6+7+8+9+10

```
ans =
```

```
40
```

```
>> CSIE (1,5)
```

1+2+3+4+5

```
ans =
```

```
15
```


Exercise:

請改寫前述程式，把相加改寫成相乘
上傳到 e-learning matlab 隨堂練習1105
下課前上傳完畢

- 1) .m 檔案
- 2) 執行結果 PowerPoint 截圖
測試 (0,100), (3,5), (40,50) 的執行結果

Numerical Methods

Polynomials and Interpolation

Polynomials

- roots - Find polynomial roots.
- poly - Convert roots to polynomial.

Interpolation

- interp1 - 1-D interpolation (table lookup).

roots

ROOTS Find polynomial roots

- It computes the roots of the polynomial whose coefficients are the elements of the vector C.
- If C has N+1 components, the polynomial is $C(1)*X^N + \dots + C(N)*X + C(N+1)$.

Example: Solve for the roots of the polynomial

$$x^3 + 2x^2 - x - 2 = 0$$

```
>> C=[1 2 -1 -2];  
>> roots(C)  
ans =  
    1.0000  
   -2.0000  
   -1.0000  
>>
```

Consider what we learned before – find the roots for $f(x)=x^2-3x+1$

```
>> C=[1 -3 1];  
>> roots(C)  
ans =  
    2.6180  
    0.3820  
>>
```

Consider what we learned before – find the roots for $f(x)=2x^4-2x^3-x^2+1$

```
>> C=[2 -2 -1 0 1];  
>> roots(C)  
ans =  
    1.0000 + 0.0000i  
    1.0000 - 0.0000i  
   -0.5000 + 0.5000i  
   -0.5000 - 0.5000i  
>>
```

1, 1, -5+0.5i, -5-0.5i

Matlab Homework 1

請利用 Fixed-point iteration 與 Newton's method.

請寫一個 fixedpoint.m 與 newton.m 二個函數

找出 $f(x)=x-2\sin x$ 之近似解(至小數第四位)

各別存成2個 .m 檔案

函數有一個引數(parameter) 代表第一個指定的 x_0

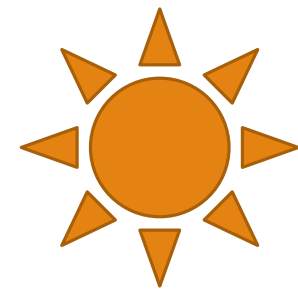
程式須有能力判斷是否收斂或發散，如果發散則提示使用者重新指定 x_0

作業評分標準 (11/19 23:59交)

可以執行、無錯誤訊息、有訊息輸出: 20%

測試條件: 80%

不接受任何理由補交，請提早上傳到e-learning
作業區 "matlab作業1"



fix, floor, ceil, round.

fix

朝零方向取整，如 $\text{fix}(-1.3)=-1$; $\text{fix}(1.3)=1$;

floor

朝負無限大方向取整，如 $\text{floor}(-1.3)=-2$; $\text{floor}(1.3)=1$;

ceil

朝正無限大方向取整，如 $\text{ceil}(-1.3)=-1$; $\text{ceil}(1.3)=2$;

round

四舍五入到最近的整数，

- $\text{round}(-1.3)=-1$;
- $\text{round}(-1.52)=-2$;
- $\text{round}(1.3)=1$;
- $\text{round}(1.52)=2$ ◦

參考資料

程式流程控制

迴圈

條件指令

課程slides資料修改自

台大資工系 張智星 教授

MATLAB 程式設計入門篇 教材

<http://mirlab.org/jang/books/matlabProgramming4beginner/>

Formats of For Loops

Format 1:

```
for 變數 = 向量  
    運算式  
end
```

在上述語法中，變數的值會被依次設定為向量的**每一個元素值**，來執行介於 **for** 和 **end** 之間的運算式。

Format 2:

```
for 變數 = 矩陣  
    運算式  
end
```

在上述語法中，變數的值會被依次設定為矩陣的**每一個直行**，來執行介於 **for** 和 **end** 之間的運算式。

程式流程控制之範例一

```
x = zeros(1,6);
```

```
for i = 1:6
```

```
    x(i) = 1/i;
```

```
end
```

% 變數 x 是一個 1x6 大小的零矩陣 **x[0 0 0 0 0 0]**

```
x
```

% 顯示 x

```
x =
```

```
1.0000 0.5000 0.3333 0.2500 0.2000 0.1667
```

在上例中，矩陣 x 最初是一個 1x6 大小的零矩陣，在 for 迴圈中，變數 i 的值依次是 1 到 6，因此矩陣 x 的第 i 個元素的值依次被設為 1/i。

我們接著可用分數形式來顯示此數列：

```
>> format rat
```

% 使用**最簡分數**形式來顯示數值

```
>> disp(x)
```

```
1      1/2    1/3    1/4    1/5    1/6
```

程式流程控制之範例二

for 迴圈可以是多層或巢狀式 (Nested) 的，在下例中即產生一個 6x6 的 Hilbert 矩陣 h，其中為第 i 列、第 j 行的元素為：

$$h_{i,j} = \frac{1}{i+j-1}$$

```
h = zeros(6);           % 變數 x 是一個 6x6 大小的零矩陣
for i = 1:6
    for j = 1:6
        h(i,j) = 1/(i+j-1);
    end
end
format rat              % 使用分數形式來顯示所有數值
```

h % 顯示 h

h =

1	1/2	1/3	1/4	1/5	1/6
1/2	1/3	1/4	1/5	1/6	1/7
1/3	1/4	1/5	1/6	1/7	1/8
1/4	1/5	1/6	1/7	1/8	1/9
1/5	1/6	1/7	1/8	1/9	1/10
1/6	1/7	1/8	1/9	1/10	1/11

程式流程控制之範例三

在下例中，for 迴圈列出先前產生的 Hilbert 矩陣的每一直行的平方和：

```
format short                                % 回到預設形式來顯示所有數值
for i = h
    disp(norm(i)^2);                        % 印出每一行的平方和
end
```

1.4914

0.5118

0.2774

0.1787

0.1262

0.0944

h =						
1.0000	←	1	1/2	1/3	1/4	1/5
0.2500	←	1/2	1/3	1/4	1/5	1/6
0.1111	←	1/3	1/4	1/5	1/6	1/7
0.0625	←	1/4	1/5	1/6	1/7	1/8
0.0400	←	1/5	1/6	1/7	1/8	1/9
0.0278	←	1/6	1/7	1/8	1/9	1/10

在上例中，由於 **h** 是一個矩陣，因此每一次 **i** 的值就是矩陣 **h** 的一直行的內容。

disp()

```
>>disp('本期樂透彩中獎號碼為：')
```

```
ans=本期樂透彩中獎號碼為：
```

```
>>disp(rand(5,4)*10^3)
```

```
ans=
```

```
582.7917    225.9499    209.0694    567.8287
```

```
423.4963    579.8069    379.8184    794.2107
```

```
515.5118    760.3650    783.3286     59.1826
```

```
333.9515    529.8231    680.8458    602.8691
```

```
432.9066    640.5265    461.0951     50.2688
```


程式流程控制之範例四

若要跳出 **for** 迴圈，可用 **break** 指令。例如，若要找出最小的 **n** 值，滿足 $n! > 1e100$ ，可輸入如下：

範例16-4 : break01.m

```
for i = 1:1000
```

```
    if prod(1:i) > 1e100
```

```
        fprintf('%g! = %e > 1e100\n', i, prod(1:i));
```

```
        break; % 跳出 for 迴圈
```

```
    end
```

```
end
```

70! = 1.197857e+100 > 1e100

prod(x:y)

$$\text{prod}(1:3) = 1 \times 2 \times 3 = 6$$

$$\text{prod}(2:5) = 2 \times 3 \times 4 \times 5 = 120$$

$$\text{prod}(1:10) = 1 \times 2 \times 3 \times \dots \times 9 \times 10 = 3628800$$

程式流程控制之範例五

在一個迴圈內若要直接跳至到此迴圈下一回合的執行，可使用 `continue` 指令。

```
x = [1 -2 3 -4 5];  
Total = 0;  
for i = 1:length(x)  
    if x(i)<0, continue; end  
    Total=Total+x(i);  
end  
posTotal  
posTotal =  
    9
```

`length(x)` 在這邊是 5 → 顯示陣列的元素個數

相當於 C 裡面的：
`if (x[i]<0)`
`{`
 `continue;`
`}`
% 若 `x(i)` 小於零，跳到此迴圈的下一回合

% 顯示 Total 的值

Format of While Loops

While loop 的格式如下：

```
while 條件式  
    運算式；  
end
```

在上述語法中，只要條件式為真，運算式就會一再被執行

程式流程控制之範例六

先前範例一 (for) 的例子，亦可用 while 迴圈改寫如下：

```
x = zeros(1,6);
```

```
i = 1;
```

```
while i<=6
```

```
    x(i) = 1/i;
```

```
    i = i+1;
```

```
end
```

相當於 C 裡面的：

```
while(i<=6)
```

```
{
```

```
    x[i] = 1/i;
```

```
    i++;
```

```
}
```

```
x
```

% 顯示 x

```
x =
```

```
1.0000  0.5000  0.3333  0.2500  0.2000  0.1667
```

程式流程控制之範例七

若要用 **while** 指令找出最小的 n 值，使得 $n! > 1e100$ ，可輸入如下：

```
n = 1;
```

```
while prod(1:n) < 1e100  
    n = n+1  
end
```

相當於 C 裡面的：

```
while( prod(1:n) < 1e100 )  
{  
    n++;  
}
```

```
fprintf('%g! = %e > 1e100\n', n, prod(1:n));
```

```
70! = 1.197857e+100 > 1e100
```

與前述的 **for** 迴圈相同，在任何時刻若要跳出 **while** 迴圈，亦可使用 **break** 指令；若要跳到下一回合的 **while** 迴圈，也可以使用 **continue** 指令。

無論是 **for** 或 **while** 迴圈，均會降低 **MATLAB** 的執行速度，因此盡量使用向量化的運算（**Vectorized Operations**）而盡量少用迴圈。

break 指令若用在多重迴圈中，每次只跳出包含 **break** 指令的最內部迴圈。

條件指令

MATLAB 支援二種條件指令（ Branching Command ）

- if-else 條件指令
- switch-case-otherwise 條件指令（ MATLAB 在第五版之後開始支援 ）

if-else 條件指令

最常用的條件指令是 **if-else**，其使用語法為

if 條件式

 運算式一;

else

 運算式二;

end

相當於 C 裡面的:

```
if( 條件式 )  
{  
    運算式一;  
}else{  
    運算式二;  
}
```

在上述語法中，當條件式成立時，**MATLAB** 將執行運算式一，否則，就執行運算式二。若不需使用運算式二，則可直接省略 **else** 和運算式二。

程式流程控制之範例八

在數值運算的過程中，若變數值為 NaN（即 Not A Number）時，我們要立刻印出警告訊息，可輸入如下例：

```
x = 0/0;
```

```
if isnan(x)
    disp('Warning: NaN detected!');
end
```

Warning: Divide by zero.

...

Warning: NaN detected!

在上例中，第一個警告訊息是 MATLAB 自動產生的，第二個警告訊息則是我們的程式碼產生的，其中 `isnan(x)` 可用於判斷 `x` 是否為 NaN，若是，則傳回 1（真），否則即傳回 0（偽）。

程式流程控制之範例九

在下例中，我們可根據向量 **y** 的元素值為奇數或偶數，來顯示不同的訊息：

```
y = [0 3 4 1 6];
```

```
for i = 1:length(y)
    if rem(y(i), 2)==0
        fprintf('y(%g) = %g is even.\n', i, y(i));
    else
        fprintf('y(%g) = %g is odd.\n', i, y(i));
    end
end
```

y(1) = 0 is even.

y(2) = 3 is odd.

y(3) = 4 is even.

y(4) = 1 is odd.

y(5) = 6 is even.

上述的 **if-else** 為雙向條件，亦即程式只會執行「運算式一」或「運算式二」，不會有第三種可能。

輸入 help rem 可查詢指令

```
>> help rem
rem    Remainder after division.
      rem(x,y) is x - n.*y where n = fix(x./y) if y ~= 0. If y is not an
      integer and the quotient x./y is within roundoff error of an integer
      then n is that integer. The inputs x and y must be real arrays of the
      same size, or real scalars.

      By convention:
          rem(x,0) is NaN.
          rem(x,x), for x~=0, is 0.
          rem(x,y), for x~=y and y~=0, has the same sign as x.

      Note: MOD(x,y), for x~=y and y~=0, has the same sign as y.
      rem(x,y) and MOD(x,y) are equal if x and y have the same sign, but
      differ by y if x and y have different signs.

      See also mod.

      Reference page for rem
      Other functions named rem
```

 >> |

程式流程控制之範例十

MATLAB 亦可執行多向條件，若要進行更多向的條件，只需一再重覆 `elseif` 即可。例如，欲判斷 `y` 向量之元素是屬於 $3n$ 、 $3n+1$ 、或 $3n+2$ ，可輸入如下：

```
y = [3 4 5 9 2];  
for i = 1:length(y)  
    if rem(y(i),3)==0  
        fprintf('y(%g)=%g is 3n.\n', i, y(i));  
    elseif rem(y(i), 3)==1  
        fprintf('y(%g)=%g is 3n+1.\n', i, y(i));  
    else  
        fprintf('y(%g)=%g is 3n+2.\n', i, y(i));  
    end  
end  
  
y(1)=3 is 3n.  
y(2)=4 is 3n+1.  
y(3)=5 is 3n+2.  
y(4)=9 is 3n.  
y(5)=2 is 3n+2.
```

switch-case-otherwise 條件指令

MATLAB 在第五版開始支援 switch-case-otherwise 的多向條件指令，其使用語法如下：

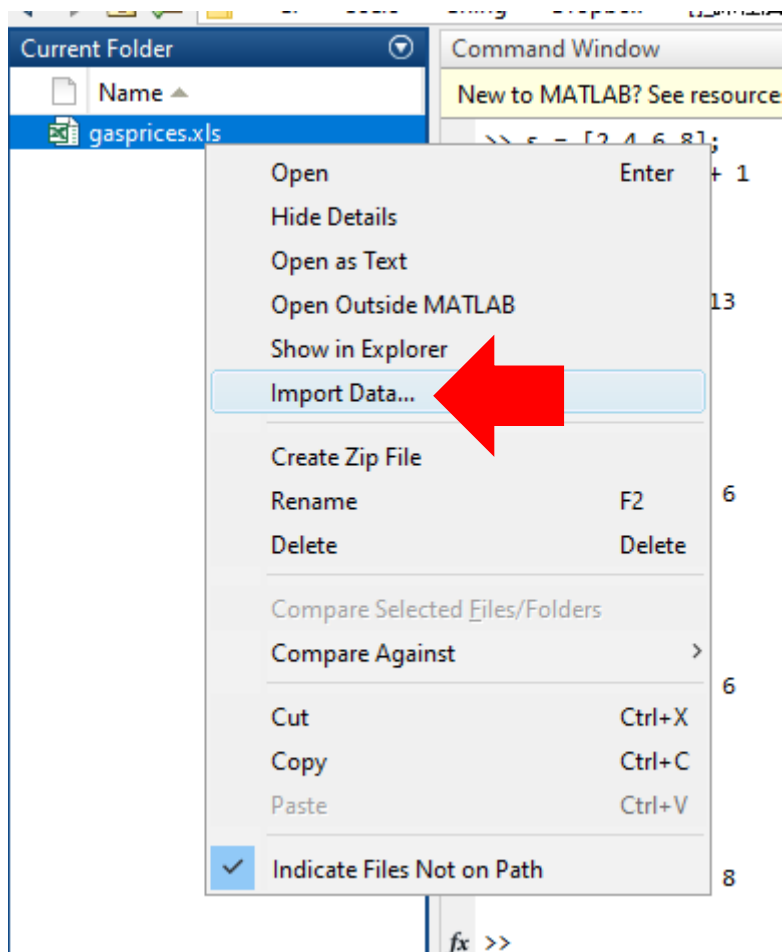
```
switch expression
    case value(1)
        statement(1)
    case value(2)
        statement(2)
    ...
    case value(n-1)
        statement(n-1)
    otherwise
        statement(n)
end
```

在上述語法中，`expression` 為一數值或字串，當其值和 `value(k)` 相等時，MATLAB 即執行 `statement(k)` 並跳出 `switch` 指令。若 `expression` 不等於 `value(k)`， $k=1, 2, \dots, n-1$ ，則 MATLAB 會執行 `statement(n)` 並跳出 `switch` 指令。

導入資料、畫圖

gasprices.xls

[illegible]



Numeric Matrix

IMPORT VIEW

Range: A6:K24

Variable Names Row: 5

Column vectors

Numeric Matrix

Cell Array

Table

Replace

unimportable cells with NaN

Import Selection

gasprices.xls

	A	B	C	D	E	F	G	H	I	J	K
						gasprices					
1	Average An...										
2	Source: US ...										
3	http://ww...										
4											
5	Year	Australia	Canada	France	Germany	Italy	Japan	Mexico	South Korea	UK	USA
6	1990		1.8700	3.6300	2.6500	4.5900	3.1600		2.0500	2.8200	1.1600
7	1991	1.9600	1.9200	3.4500	2.9000	4.5000	3.4600	1.3000	2.4900	3.0100	1.1400
8	1992	1.8900	1.7300	3.5600	3.2700	4.5300	3.5800	1.5000	2.6500	3.0600	1.1300
9	1993	1.7300	1.5700	3.4100	3.0700	3.6800	4.1600	1.5600	2.8800	2.8400	1.1100
10	1994	1.8400	1.4500	3.5900	3.5200	3.7000	4.3600	1.4800	2.8700	2.9900	1.1100
11	1995	1.9500	1.5300	4.2600	3.9600	4	4.4300	1.1100	2.9400	3.2100	1.1500
12	1996	2.1200	1.6100	4.4100	3.9400	4.3900	3.6400	1.2500	3.1800	3.3400	1.2300
13	1997	2.0500	1.6200	4	3.5300	4.0700	3.2600	1.4700	3.3400	3.8300	1.2300
14	1998	1.6300	1.3800	3.8700	3.3400	3.8400	2.8200	1.4900	3.0400	4.0600	1.0600
15	1999	1.7200	1.5200	3.8500	3.4200	3.8700	3.2700	1.7900	3.8000	4.2900	1.1700
16	2000	1.9400	1.8600	3.8000	3.4500	3.7700	3.6500	2.0100	4.1800	4.5800	1.5100
17	2001	1.7100	1.7200	3.5100	3.4000	3.5700	3.2700	2.2000	3.7600	4.1300	1.4600
18	2002	1.7600	1.6900	3.6200	3.6700	3.7400	3.1500	2.2400	3.8400	4.1600	1.3600
19	2003	2.1900	1.9900	4.3500	4.5900	4.5300	3.4700	2.0400	4.1100	4.7000	1.5900
20	2004	2.7200	2.3700	4.9900	5.2400	5.2900	3.9300	2.0300	4.5100	5.5600	1.8800
21	2005	3.2300	2.8900	5.4600	5.6600	5.7400	4.2800	2.2200	5.2800	5.9700	2.3000
22	2006	3.5400	3.2600	5.8800	6.0300	6.1000	4.4700	2.3100	5.9200	6.3600	2.5900
23	2007	3.8500	3.5900	6.6000	6.8800	6.7300	4.4900	2.4000	6.2100	7.1300	2.8000
24	2008	4.4500	4.0800	7.5100	7.7500	7.6300	5.7400	2.4500	5.8300	7.4200	3.2700

Sheet1

Variables - gasprices

gasprices

19x11 double

	1	2	3	4	5	6
1	1990	NaN	1.8700	3.6300	2.6500	4.5900
2	1991	1.9600	1.9200	3.4500	2.9000	4.5000
3	1992	1.8900	1.7300	3.5600	3.2700	4.5300
4	1993	1.7300	1.5700	3.4100	3.0700	3.6800
5	1994	1.8400	1.4500	3.5900	3.5200	3.7000
6	1995	1.9500	1.5300	4.2600	3.9600	4
7	1996	2.1200	1.6100	4.4100	3.9400	4.3900
8	1997	2.0500	1.6200	4	3.5300	4.0700
9	1998	1.6300	1.3800	3.8700	3.3400	3.8400
10	1999	1.7200	1.5200	3.8500	3.4200	3.8700
11	2000	1.9400	1.8600	3.8000	3.4500	3.7700
12	2001	1.7100	1.7200	3.5100	3.4000	3.5700
13	2002	1.7600	1.6900	3.6200	3.6700	3.7400
14	2003	2.1900	1.9900	4.3500	4.5900	4.5300
15	2004	2.7200	2.3700	4.9900	5.2400	5.2900
16	2005	3.2300	2.8900	5.4600	5.6600	5.7400
17	2006	3.5400	3.2600	5.8800	6.0300	6.1000

Workspace

Name	Value
gasprices	19x11 dou



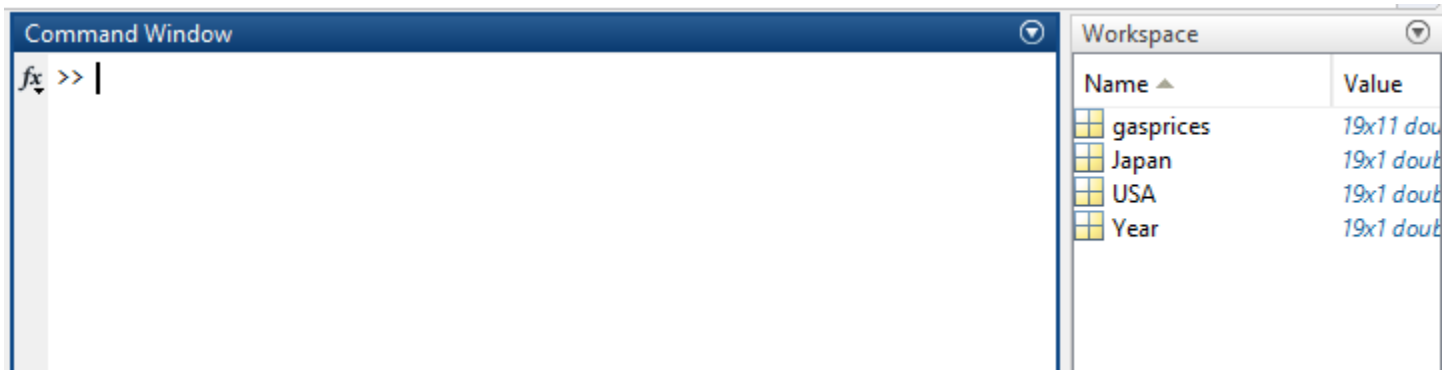
Double click

Command Window

New to MATLAB? See resources for [Getting Started](#).

fx >>

Please import Japan, USA, Year



The image shows a screenshot of the MATLAB Command Window and Workspace. The Command Window on the left has a blue header and contains the prompt `fx >> |`. The Workspace on the right has a light gray header and displays a table of variables in the current workspace.

Name ▲	Value
gasprices	19x11 dou
Japan	19x1 doub
USA	19x1 doub
Year	19x1 doub

HOME PLOTS APPS

Year Japan

plot Plot as mu... bar area pie histogram scatter semilog

SELECTION PLOTS: Year, Japan OPTIONS

Reuse Figure New Figure

C:\Users\Ching\Dropbox\[]課程演 10601 工程數學 MATLAB 投影片 matlab_01

Current Folder

Name ▲

gasprices.xls

Command Window

fx >>

Workspace

Name ▲	Value
gasprices	19x11 doub
Japan	19x1 doub
USA	19x1 doub
Year	19x1 doub

