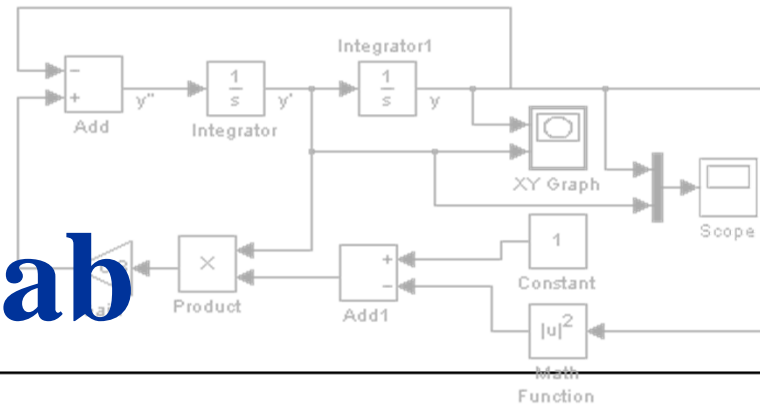


認識 Matlab

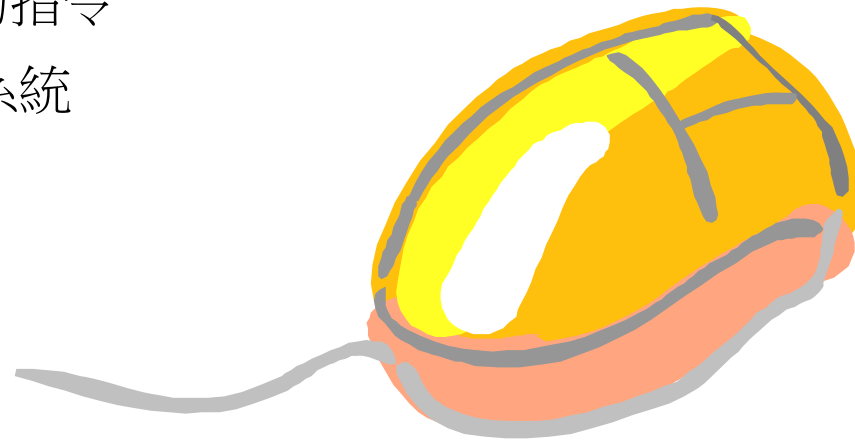


認識Matlab的歷史

熟悉Matlab的工作環境

練習輸入與執行Matlab的指令

學習如何使用線上求助系統





Matlab 簡介

Matlab是由美國Mathworks公司所推出，名稱為Matrix Laboratory 縮寫而成。

Matlab的主要功能：

數值運算(numerical computation)

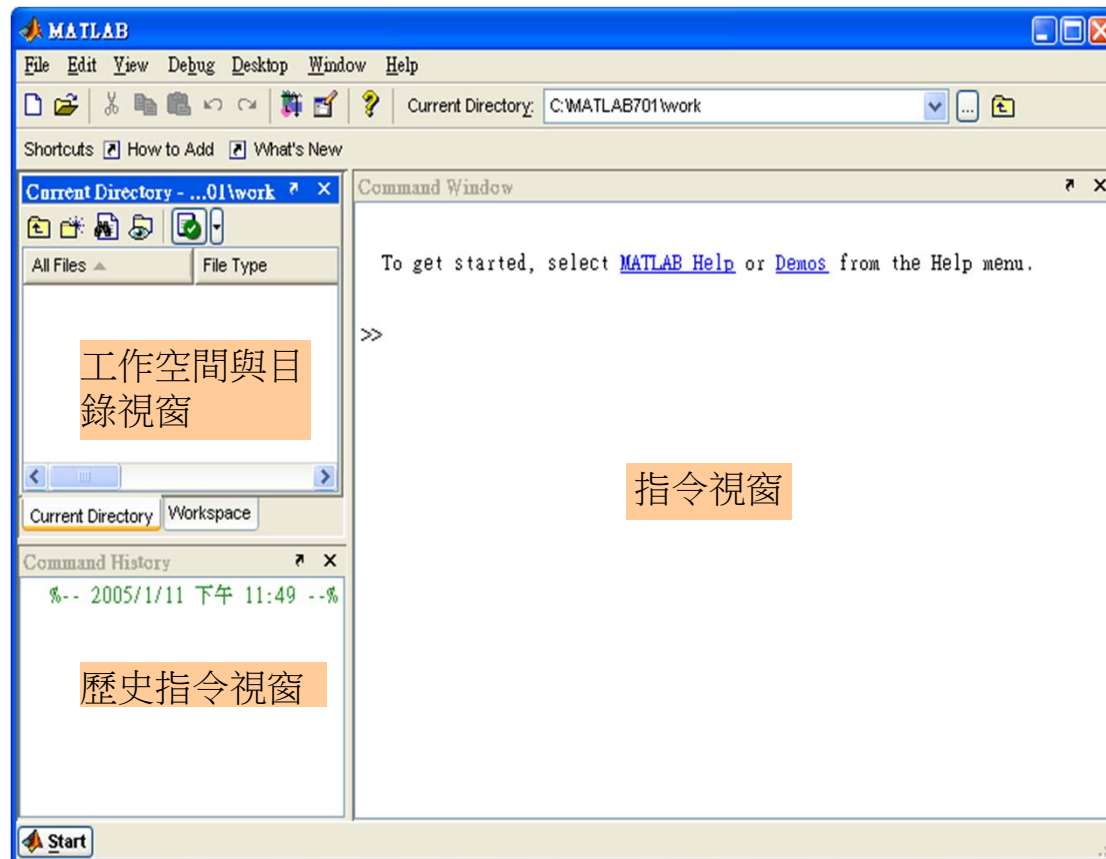
符號運算(symbolic manipulation)

Simulink (系統的動態模擬)

各種工具箱(toolbox)

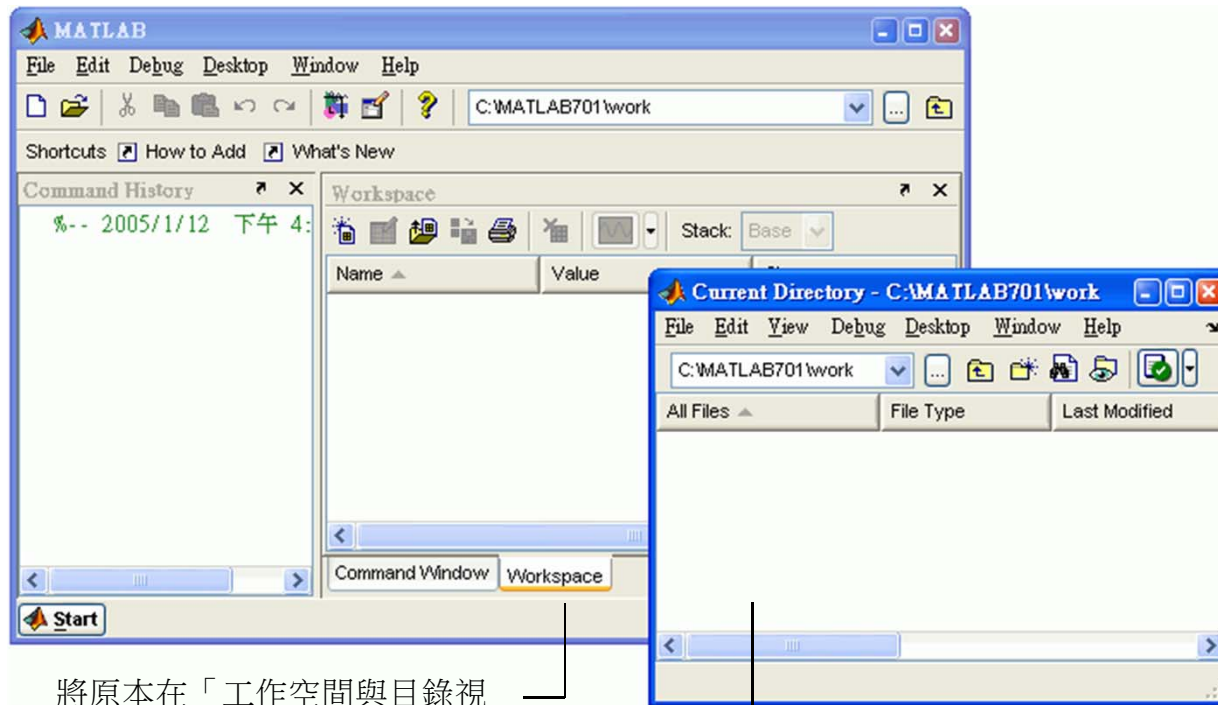
工作環境介紹

- 啟動Matlab之後，會開啟一個新的工作環境：



調整工作環境

- 按住視窗標題列，可移動視窗到另一個位置：



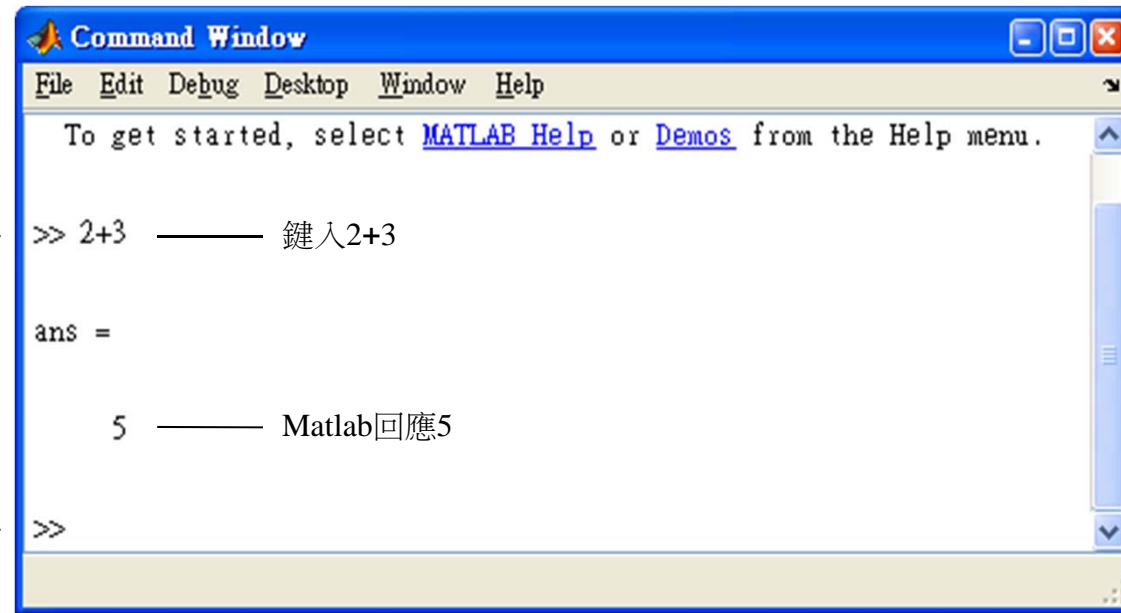
將原本在「工作空間與目錄視窗」裡的「Workspace」標籤拖在「指令視窗」裡，形成另一個標籤

將原本在「工作空間與目錄視窗」裡的「Current Directory」標籤獨立成一個視窗

基本範例

輸入與執行

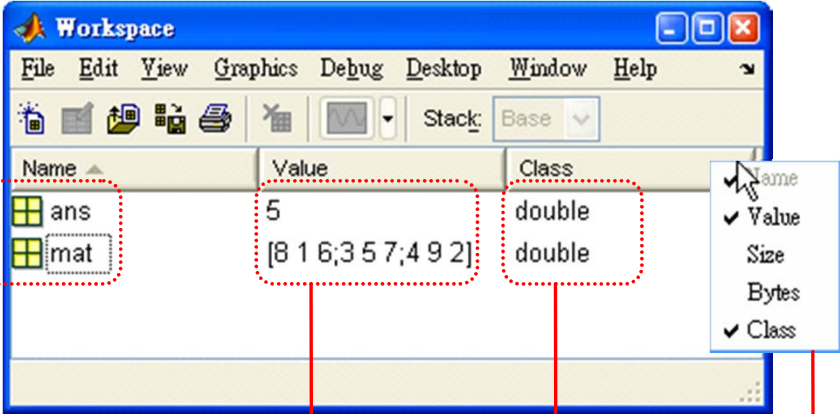
提示符號，由
系統自動產生



```
Command Window
File Edit Debug Desktop Window Help
To get started, select MATLAB Help or Demos from the Help menu.
>> 2+3      鍵入2+3
ans =
     5      Matlab回應5
>>
```

查看變數的內容

- 如果想知道目前已定義有哪些變數，可到 Workspace視窗裡查看：



The screenshot shows the MATLAB Workspace window with the following table of variables:

Name	Value	Class
ans	5	double
mat	[8 1 6;3 5 7;4 9 2]	double


A context menu is open over the 'ans' variable, showing the following options:

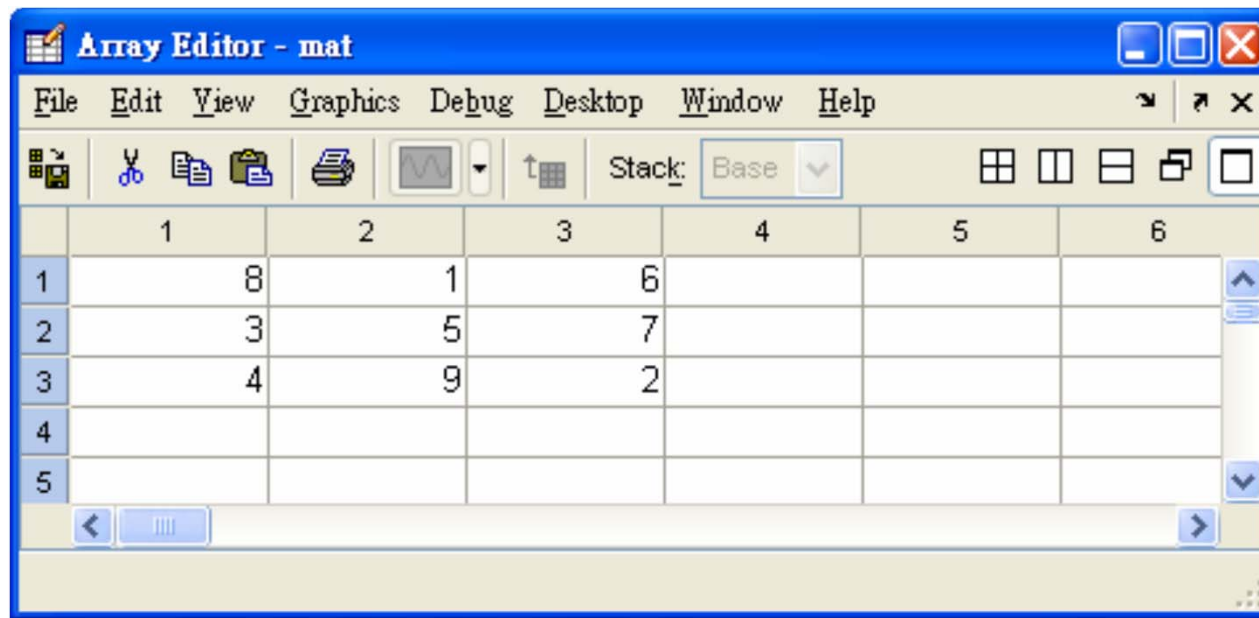
- Name
- Value
- Size
- Bytes
- Class

Annotations in the image:

- 變數名稱 (Variable Name) points to the 'ans' and 'mat' entries in the Name column.
- 變數值 (Variable Value) points to the value '5' for 'ans' and the matrix '[8 1 6;3 5 7;4 9 2]' for 'mat'.
- 資料型態 (Data Type) points to the 'double' class for both variables.
- 按下滑鼠右鍵，於出現的選單裡 可選擇視窗裡要呈現的項目 (Right-click to show a menu where you can select the items to be displayed in the window) points to the context menu.

利用Array Editor編修陣列

- 想查看或修改變數mat的內容，可在  mat 圖示上方連按兩下滑鼠(此為陣列變數)：



其它的介面操作

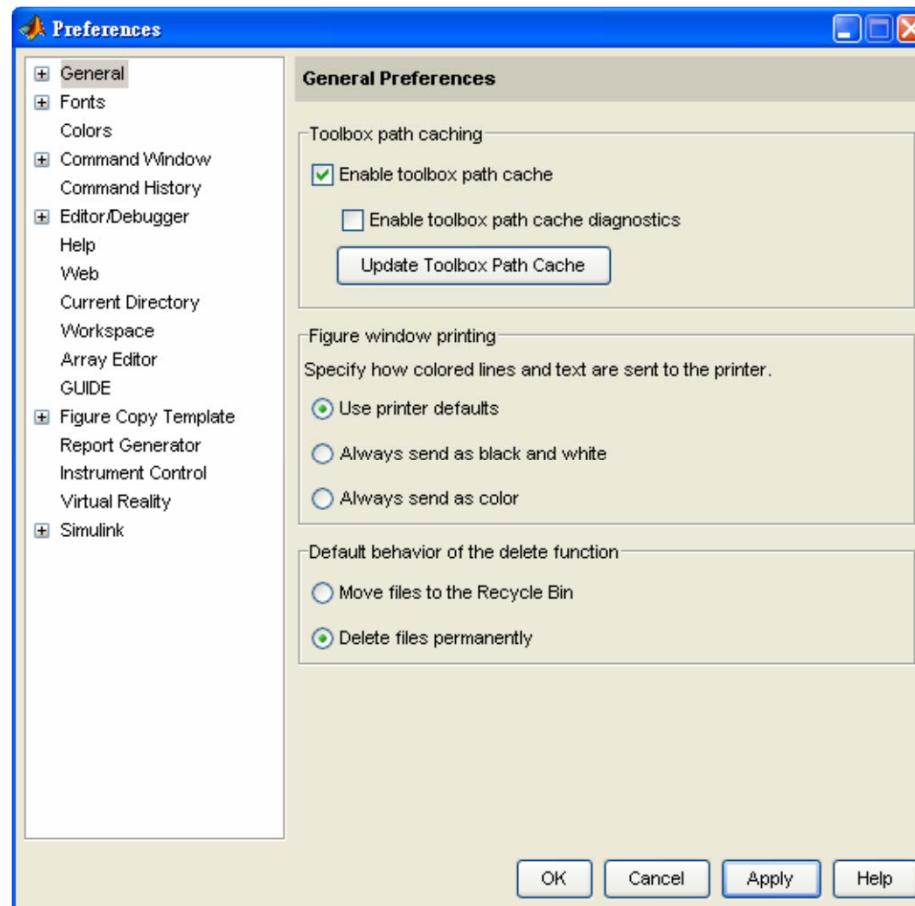
清除視窗裡的内容

Edit	
U <u>ndo</u>	Ctrl+Z
R <u>edo</u>	Ctrl+Y
Cu <u>t</u>	Ctrl+X
C <u>o</u> py	Ctrl+C
P <u>a</u> ste	Ctrl+V
Duplicate	Ctrl+D
Paste Special...	
Select <u>A</u> ll	Ctrl+A
D <u>e</u> lete	Delete
R <u>e</u> name	
Ed <u>i</u> t V <u>a</u> lue	
F <u>i</u> nd...	
Find Files...	
Clear Command <u>W</u> indow	
Clear Command <u>H</u> istory	
Clear <u>W</u> orkspace	

} 可用來清除特定視窗的内容

設定Matlab的工作環境

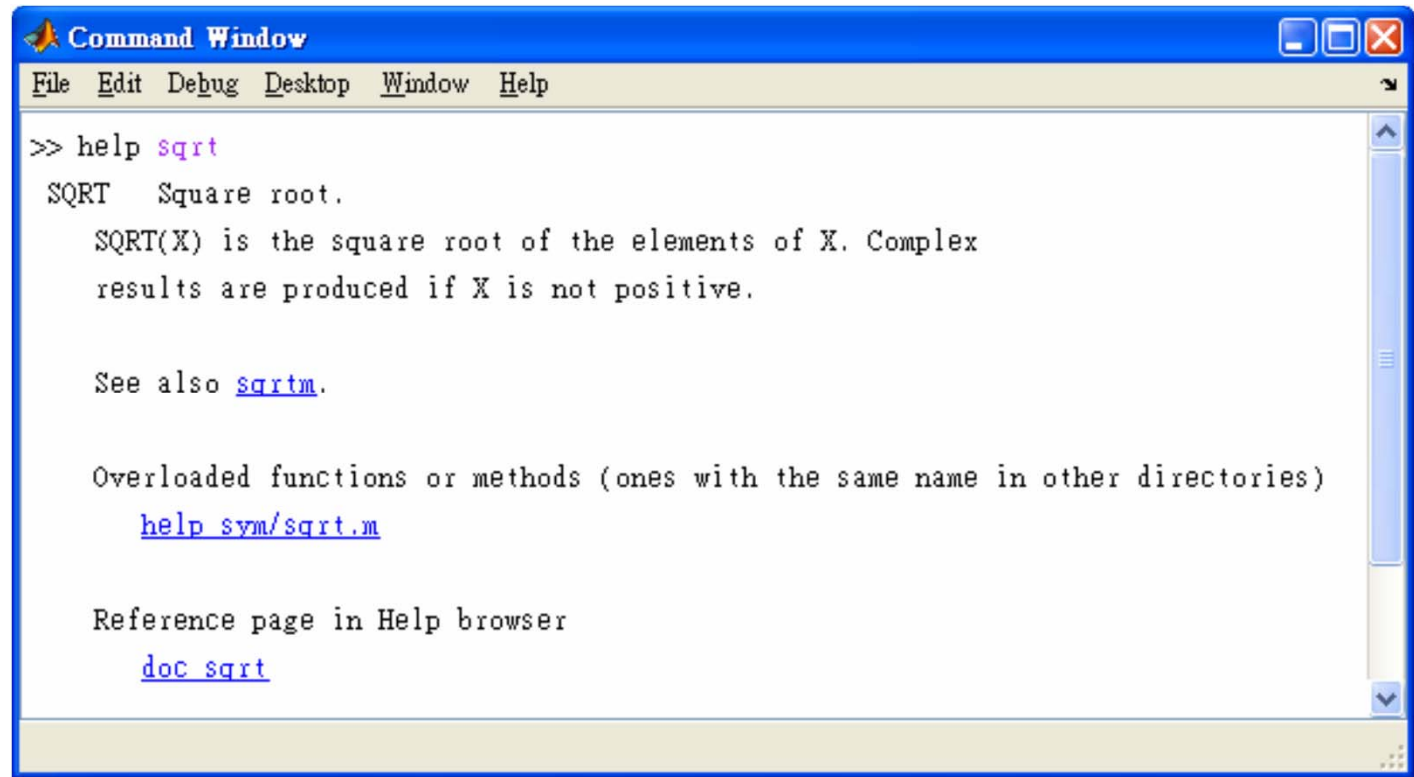
- 「Preferences」對話方塊：



使用線上求助系統

利用help指令來查詢

```
>> help sqrt
```



```
Command Window
File Edit Debug Desktop Window Help
>> help sqrt
SQRT Square root.
SQRT(X) is the square root of the elements of X. Complex
results are produced if X is not positive.

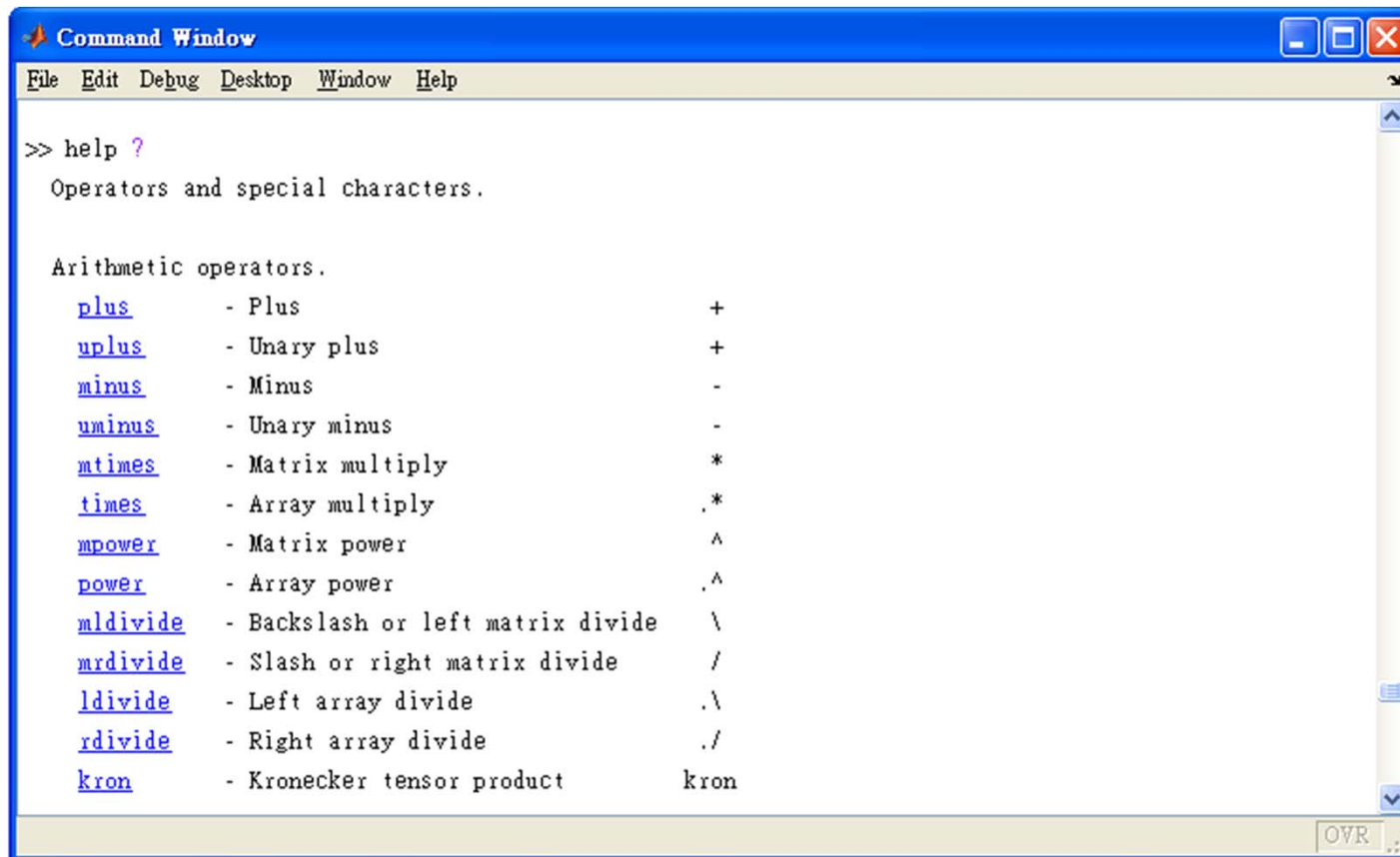
See also sqrtm.

Overloaded functions or methods (ones with the same name in other directories)
help sym/sqrt.m

Reference page in Help browser
doc sqrt
```

列出所有指令

```
>> help ?
```

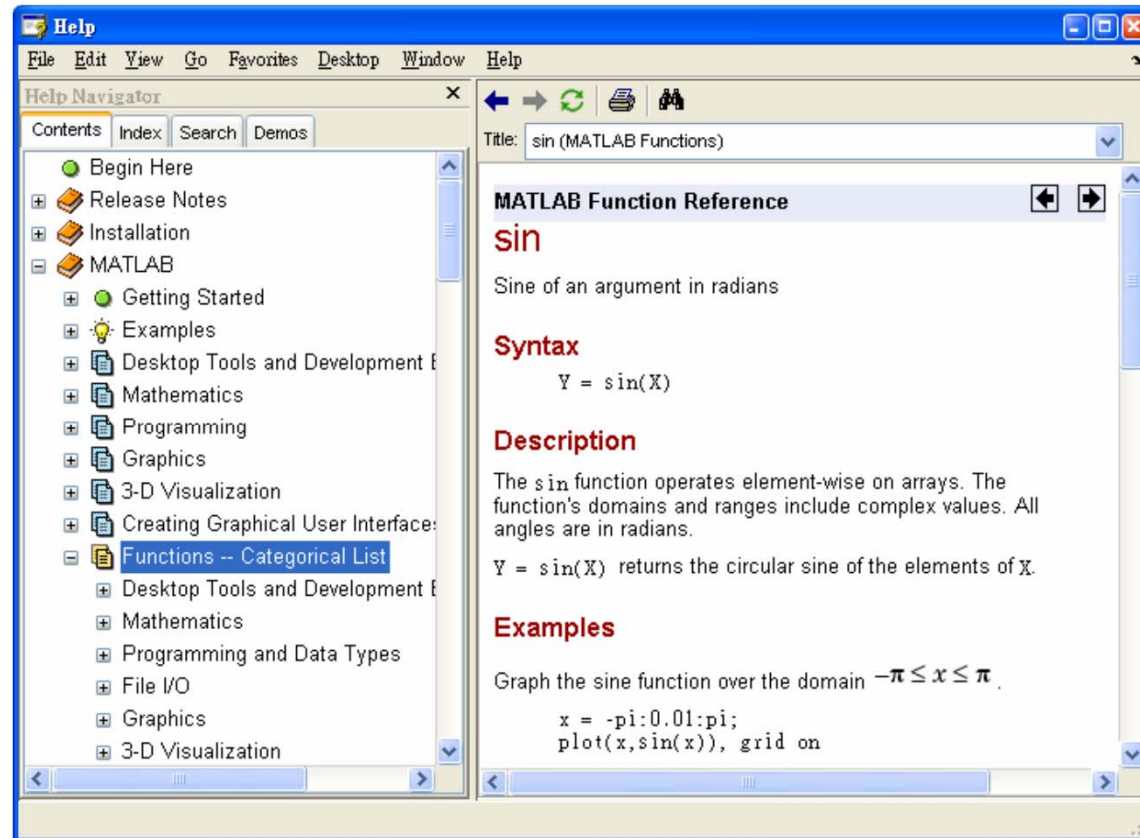


```
Command Window
File Edit Debug Desktop Window Help
>> help ?
Operators and special characters.

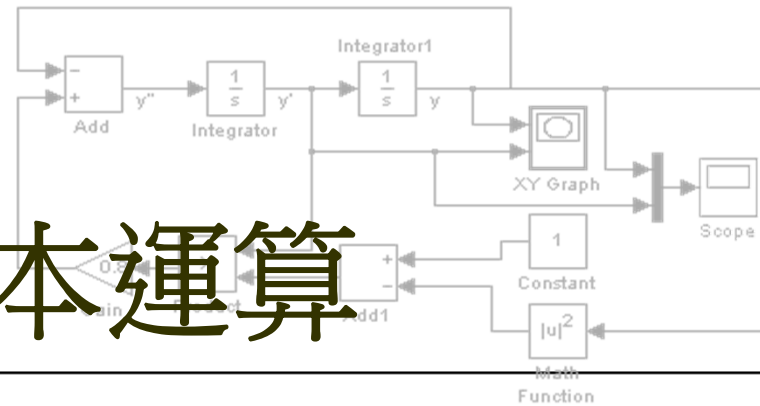
Arithmetic operators.
  plus      - Plus                +
  uplus     - Unary plus          +
  minus     - Minus               -
  uminus    - Unary minus        -
  mtimes    - Matrix multiply     *
  times     - Array multiply      .*
  mpower    - Matrix power        ^
  power     - Array power         .^
  mldivide  - Backslash or left matrix divide \
  mrdivide  - Slash or right matrix divide /
  ldivide   - Left array divide   .\
  rdivide   - Right array divide  ./
  kron      - Kronecker tensor product kron
```

利用doc指令來查詢

>> doc sin



Matlab基本運算

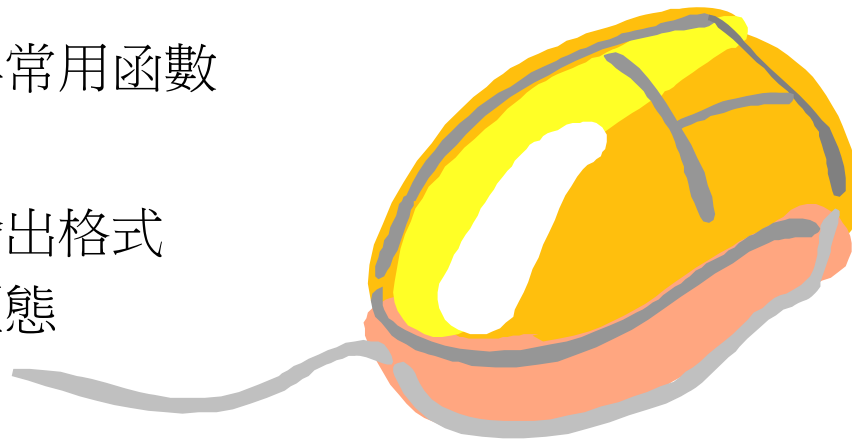


學習Matlab的基本語法與常用函數

學習陣列的輸入方式

學習如何控制Matlab的輸出格式

認識Matlab的基本資料型態





基本運算

- 基本的數學運算
- 變數的設定與清除
- 永久常數



基本的數學運算

- 下表列出了Matlab基本運算裡所使用的符號：

表 2.1.1 Matlab 的基本運算符號

運算符號	代表意義	範例
+	加法	5+3
-	減法、負號	6-4、-6
*	乘法	12*5
/	除法	6/23
^	次方	2^3



變數的設定與清除

- 變數名稱是由英文字母、數字或底線所組成
- 開頭的第一個字元必須是英文字母
- 名稱長度不能超過32個字元
- 會區分變數的大小寫
- 變數不必宣告便可直接使用


- 
-
- 查詢工作區裡有哪些變數已被定義過，可使用 `who` 或 `whos` 指令
 - 如要清除已定義過的變數，可用 `clear` 指令

表 2.1.2 查詢工作區裡所使用的變數

指令	說明
<code>who</code>	查詢於目前的工作區內，有哪些變數正在使用
<code>whos</code>	同 <code>who</code> ，但會列出每一個變數詳細的資訊
<code>whos var</code>	查詢變數 <code>var</code> 的詳細資訊
<code>clear</code>	清除工作區內的所有變數
<code>clear var</code>	清除工作區內的變數 <code>var</code>



永久常數

- Matlab 永久常數（permanent constant）如下表所列：

表 2.1.3 Matlab 所使用的永久常數

永久常數	說明
pi	圓周率
inf 或 Inf	無限大
i、j	虛數 (imaginary numbers)
NaN 或 nan	不存在的數 (not a number)
realmax	系統所能表示之最大數值，其值為 1.797693134862316e+308
realmin	系統所能表示之最小數值，其值為 2.225073858507201e-308



常用的數學函數

- 三角函數
- 指數運算相關的函數
- 複數運算相關的函數
- 捨位與取餘數函數
- 其它常用的函數



三角函數

- Matlab所提供的三角函數有兩種版本：
 - 以角度（degree）為單位
 - 以弧度（radian）為單位

表 2.2.1 三角函數與反三角函數

	數學函數	說明
	<code>sin</code> 、 <code>cos</code> 、 <code>tan</code> 、 <code>cot</code> 、 <code>sec</code> 、 <code>csc</code>	三角函數（角度單位為弧度）
	<code>asin</code> 、 <code>acos</code> 、 <code>atan</code> 、 <code>acot</code> 、 <code>asec</code> 、 <code>acsc</code>	反三角函數（角度單位為弧度）
	<code>sind</code> 、 <code>cosd</code> 、 <code>tand</code> 、 <code>cotd</code> 、 <code>secd</code> 、 <code>cscd</code>	三角函數（角度單位為度）
	<code>asind</code> 、 <code>acosd</code> 、 <code>atand</code> 、 <code>acotd</code> 、 <code>asecd</code> 、 <code>acscd</code>	反三角函數（角度單位為度）


- 
-
- Matlab所提供了下面的雙曲線與反雙曲線函數：

表 2.2.2 雙曲線函數與反雙曲線函數

數學函數	說明
<code>sinh</code> 、 <code>cosh</code> 、 <code>tanh</code> 、 <code>coth</code> 、 <code>sech</code> 、 <code>csch</code>	雙曲線函數
<code>asinh</code> 、 <code>acosh</code> 、 <code>atanh</code> 、 <code>acoth</code> 、 <code>asech</code> 、 <code>acsch</code>	反雙曲線函數



指數運算相關的函數

- Matlab提供了計算指數、對數、開根號以及開 n 次方等運算的函數：

表 2.2.3 指數與對數函數

數學函數	說明
<code>exp(x)</code>	自然指數函數，計算 e^x
<code>log(x)</code>	計算 x 的自然對數（以 e 為底）
<code>log2(x)</code>	計算 x 的對數（以 2 為底）
<code>log10(x)</code>	計算 x 的對數（以 10 為底）
<code>sqrt(x)</code>	開根號函數，計算 \sqrt{x}
<code>nthroot(x,n)</code>	開 n 次方函數，計算 $\sqrt[n]{x}$



複數運算相關的函數

- Matlab是以小寫的 i 或 j 來表示 $\sqrt{-1}$ 。如：
 $z = 2+3i$ 或 $z = 2+3j$ 或 $z = \text{complex}(2,3)$

表 2.2.4 與複數運算相關的函數

數學函數	說明
<code>abs(z)</code>	計算 z 的絕對值
<code>angle(z)</code>	計算複數 z 的幅角 (argument)
<code>complex(a,b)</code>	建立複數，並指定實部為 a ，虛部為 b
<code>conj(z)</code>	求出複數 z 的共軛複數 (conjugate complex)
<code>imag(z)</code>	取出複數 z 的虛部 (imaginary part)
<code>real(z)</code>	取出複數 z 的實部 (real part)



捨位與取餘數函數

- 下表列出了可用來做捨位處理的函數：

表 2.2.5 捨位與取餘數函數

數學函數	說明
<code>fix(x)</code>	捨棄數值 x 的小數部份
<code>floor(x)</code>	取出小於或等於 x 的最大整數
<code>ceil(x)</code>	取出大於或等於 x 的最小整數
<code>round(x)</code>	取出最靠近 x 的整數
<code>rem(x,y)</code>	取出 x/y 的餘數 (remainder)



其它常用的函數

- Matlab也供了一些常用的數學函數，列表如下：

表 2.2.6 其它常用的數學函數

數學函數	說明
<code>abs(x)</code>	計算 x 的絕對值 (absolute value)
<code>factor(x)</code>	求出整數 x 的所有質因數 (prime factors)
<code>factorial(x)</code>	計算 x 的階乘 (factorial)
<code>gcd(a,b)</code>	計算 a 與 b 的最大公因數 (greatest common divisor)
<code>lcm(a,b)</code>	計算 a 與 b 的最小公倍數 (least common multiplier)
<code>primes(x)</code>	找出小於等於 x 的所有質數 (prime)
<code>isprime(x)</code>	查詢整數 x 是否為質數，若是，則回應 1，否則回應 0



陣列觀念

- 陣列（array）依其維度可概分為一維、二維與多維
- 若陣列只有一維，稱之為向量（vector）
- 陣列若是二維，則稱之為矩陣（matrix）
- 向量可再細分為列向量（row vector）與行向量（column vector）



$[2 \ 6 \ 8 \ 3]$ ——— 這是列向量。它是一維陣列，但也
可看成是維度為 1×4 （1 列 4 行）

$\begin{bmatrix} 1 \\ 7 \\ 4 \end{bmatrix}$ ——— 這是行向量。它雖是一維陣列，但也
可看成是維度為 3×1 （3 列 1 行）的

$\begin{bmatrix} 3 & 9 & 0 & 1 \\ 2 & 4 & 4 & 2 \\ 7 & 7 & 9 & 2 \end{bmatrix}$ ——— 這是 3×4 （3 列 4 行）的矩陣



一維陣列


- Matlab的向量是以一維陣列來表示
- 列向量裡，元素與元素之間可以用空白鍵，或者是用逗號來隔開
- 行向量是以分號來隔開元素



○ Matlab所提供建立列向量與行向量的函數：

表 2.3.1 建立向量的指令與函數

指令與函數	說明
$a:b$	從 a 到 b ，間距為 1，建立一個列向量
$a:step:b$	從 a 到 b ，間距為 $step$ ，建立一個列向量
<code>linspace(a,b)</code>	從 a 到 b ，建立一個具有 100 個元素的列向量
<code>linspace(a,b,n)</code>	從 a 到 b ，建立一個具有 n 個元素的列向量
<code>length(v)</code>	查詢向量 v 的元素個數
v'	將向量 v 轉置，也就是列向量變行向量，行向量變列向量



○ Matlab所提供的基本的向量處理函數：

表 2.3.2 基本的向量處理函數

函 數	說 明
<code>sum(v)</code>	計算向量 v 的總和
<code>prod(v)</code>	計算向量 v 的乘積
<code>max(v)</code>	取出向量 v 的最大值
<code>min(v)</code>	取出向量 v 的最小值
<code>sort(v)</code>	將向量 v 裡的元素由小到大排列
<code>sort(v, 'descend')</code>	將向量 v 裡的元素由大到小排列
<code>cumsum(v)</code>	計算向量 v 的累加 (cumulative sum)
<code>cumprod(v)</code>	計算向量 v 的累乘 (cumulative product)



二維陣列

- 我們稱二維陣列為矩陣（matrix）
- 一個 $m \times n$ 的矩陣代表矩陣具有 m 個橫列， n 個直行
- 要建立矩陣，同一列的元素用空白隔開，列與列之間用分號隔開

```
>> m1=[1 3 4; 3 5 7]
```

```
>> m2=[2,3,1,4; 4,8,5,0; 3,3,1,2]
```

- 
-
- 查詢陣列的維度，以及陣列元素的個數的函數：

表 2.3.3 用來查詢陣列相關資訊的函數

函 數	說 明
<code>size(<i>m</i>)</code>	查詢陣列 <i>m</i> 的維度 (dimensions)
<code>length(<i>m</i>)</code>	查詢陣列 <i>m</i> 的行數
<code>ndims(<i>m</i>)</code>	查詢陣列 <i>m</i> 的維數
<code>numel(<i>m</i>)</code>	查詢陣列 <i>m</i> 元素的總數



控制Matlab的顯示方式

- 顯示或不顯示運算結果
- 指令跨行的控制
- 資料輸出格式的控制



顯示或不顯示運算結果

- 在同一行裡撰寫數個Matlab的敘述，可用逗號隔開
- 如果不想讓運算結果在螢幕上，可加上分號

表 2.4.1 控制顯示或不顯示運算結果

敘述型式	說明
敘述 1, 敘述 2, 敘述 3;	執行敘述 1~3，但敘述 3 的結果不顯示
敘述 1; 敘述 2; 敘述 3;	執行敘述 1~3，且每一個結果均不顯示



指令跨行的控制

- 如果敘述較長，則可利用跨行字元「...」（連續三個點）

```
>> 12.04+sin(1.4)-12*cos(3.1405)+...  
    6+tan(0.13)
```

```
ans =  
    31.1562
```



資料輸出格式的控制

- Matlab在顯示數值時：
 - 位數少於或等於9個，便會全數輸出它
 - 位數大於9個，則會以指數的型式來表示它
- 如果是帶有小數的數值：
 - 若數值大於等於1000，或者是小於等於0.001，會以指數來表示
 - Matlab預設以4個位數的小數來顯示帶有小數的數值

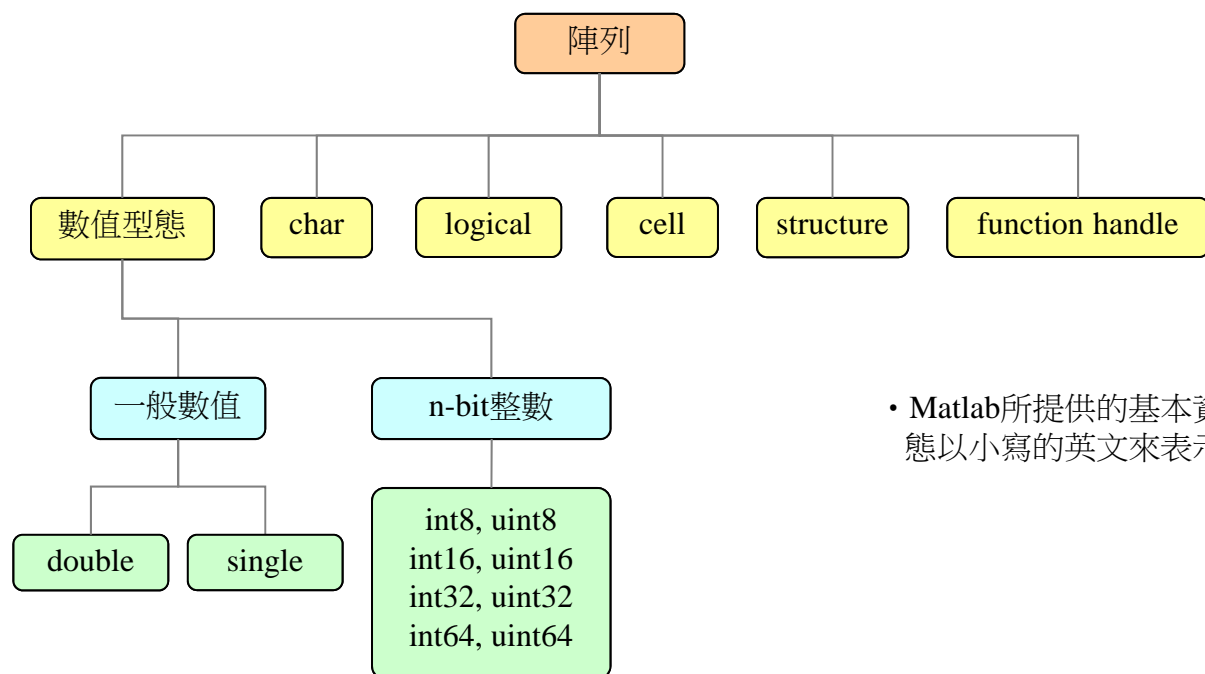
- 
-
- 利用format指令，可以更改預設的數值顯示方式：

表 2.4.2 控制 Matlab 的輸出格式

格式指令	說明
format	Matlab 的預設格式
format short	精簡格式，其格式同 format
format short g	若數值帶有小數，則以總共 5 個數字來顯示數值部分
format short e	若數值帶有小數，以指數的型式來顯示
format long	完整格式，以 16 個位數來顯示數字
format long g	完整格式，以 15 個位數來顯示數字。
format long e	完整格式，以指數型式來顯示完整格式
format compact	簡潔格式，即在指令輸入與結果輸出之間不留任何空行
format loose	寬鬆格式，即在指令輸入與結果輸出之間空一行

Matlab所提供的資料型態

- Matlab常用的資料型態，以及彼此之間的關係圖：



- Matlab所提供的基本資料型態以小寫的英文來表示



數值資料型態

- 一般數值型態
可分為single（單精度）與double（倍精度）兩種

表 2.5.1 單精度與倍精度型態

資料型態	說明	位元組	最大值	最小值
single	單精度	4	3.4028×10^{38}	1.1755×10^{-38}
double	倍精度	8	1.7977×10^{308}	2.2251×10^{-308}



○ **n-bit整數型態**

可分為有號（signed）與無號（unsigned）兩種
依大小可分為8、16、32與64個位元（bits）的整數

表 2.5.2 n-bit 整數型態

資料型態	說明	位元組	最大值	最小值
int8	8-bit 整數	1	127	-128
uint8	8-bit 無號整數	1	255	0
int16	16-bit 整數	2	32767	-32768
uint16	16-bit 無號整數	2	65535	0
int32	32-bit 整數	4	2147483647	-2147483648
uint32	32-bit 無號整數	4	4294967295	0
int64	64-bit 整數	8	9223372036854775807	-9223372036854775808
uint64	64-bit 無號整數	8	18446744073709551615	0



- 查詢數值資料型態的範圍

要查詢每一種數值資料型態所能表示的範圍，可用如下的函數：

表 2.5.3 查詢數值資料型態所能表示的範圍的函數

函 數	說 明
<code>realmax('data type')</code>	查詢所指定之一般數值資料型態的最大值
<code>realmin('data type')</code>	同 <code>realmax</code> ，不過是查詢最小值
<code>intmax('data type')</code>	查詢所指定之整數資料型態的最大值
<code>intmin('data type')</code>	同 <code>intmax</code> ，不過是查詢最小值



字元資料型態

- 在Matlab裡，字元是以成對的單引號括起來
- 每一個字元佔了兩個bytes

```
>> ch='A'
```

```
ch =  
A
```

```
>> whos('ch')
```

Name	Size	Bytes	Class
ch	1x1	2	char array

```
Grand total is 1 element using 2 bytes
```



邏輯資料型態

- Matlab是以1代表運算結果為true，而以0代表運算結果為false
- 邏輯資料型態只佔了一個位元組的記憶空間

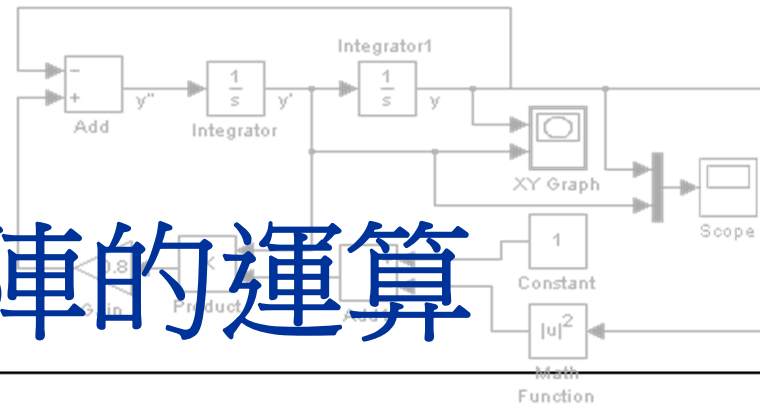
```
>> t1=isprime(13)
```

```
t1 =  
    1
```

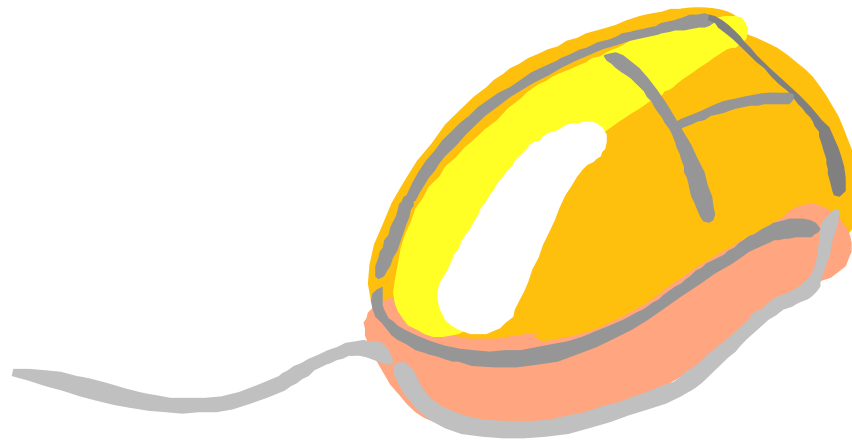
```
>> whos t1
```

Name	Size	Bytes	Class
t1	1x1	1	logical array

向量與矩陣的運算



陣列裡元素的結構
多維陣列的建立
編修矩陣的內容
基本的矩陣數學運算





陣列元素的處理

向量元素的操作

- Matlab的陣列索引值是從1開始
- C語言的陣列索引值是從0開始
- 下面是向量元素操作的範例：

```
>> v1=[6 7 8 9];
```

```
>> 2*v1+1
```

```
ans =
```

```
13    15    17    19
```

```
>> v1(2)
```

```
ans =
```

```
7
```



向量運算

- 純量的四則運算在向量運算中分別代表相加、相減、內積和投影。
- 例如 $A=[2\ 2]$, $B=[2\ 1]$, 則
 - $A+B=[4\ 3]$
 - $A-B=[0\ 1]$
 - $\text{dot}(A,B)=6 \rightarrow A$ 與 B 的內積(不可使用 $A*B$)
 - $A.*B=[4\ 2] \rightarrow$ 向量對應元素相乘
 - $\text{norm}(A)=2.8284 \rightarrow A$ 的長度
即, $\text{sqrt}(\text{dot}(A,A))$
 - $A/B=1.2 \rightarrow A$ 在 B 的投影量
即, $\text{dot}(A,B)/\text{norm}(B)$
 - $A./B=[1\ 2] \rightarrow$ 向量對應元素相除



矩陣元素的操作

- 矩陣必須有列與行兩個索引值才能取得陣列裡的特定元素：

```
>> M=[1 2 3 4;5 6 7 8;9 10 11 12]
```

```
M =
```

```
     1     2     3     4
     5     6     7     8
     9    10    11    12
```

```
>> M(2,3)
```

```
ans =
```

```
7
```

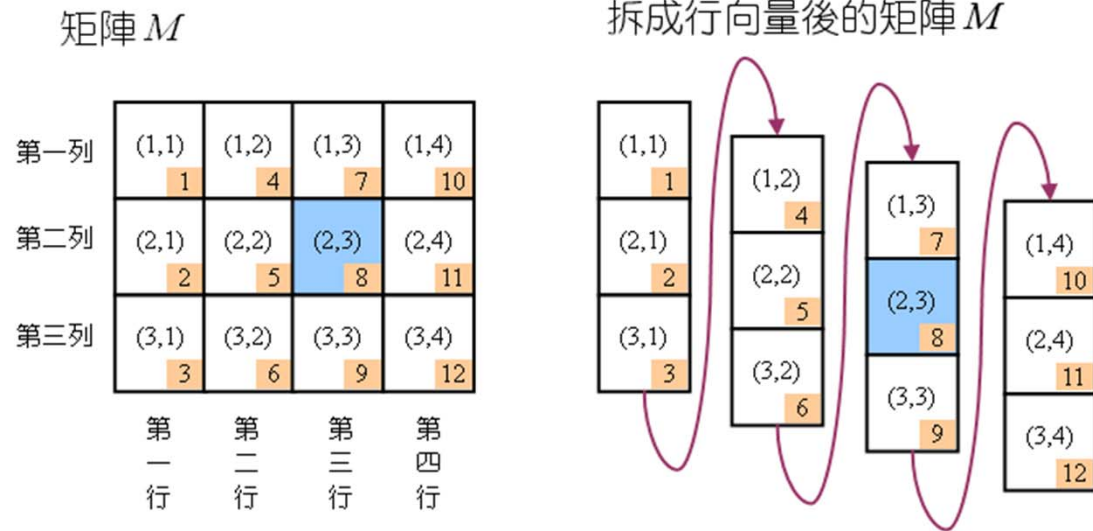
```
>> M(3,[1 2 3])
```


```
ans =
```

```
9    10    11
```

矩陣的索引值之結構

- Matlab的矩陣是利用「以行為主」的結構來儲存，如下圖所示





對於一個 $m \times n$ 的矩陣 M 而言，二維索引值 (i, j) 和一維索引值 k 之間的關係可用下面的公式來表示：

$$M(k) = M(i + (j - 1) \times m)$$

例如，若陣列 M 的維度為 3×4 ，則第二列第三行的元素 $M(2,3)$ ，換成以一維的索引值來表示時，可換算得

$$M(k) = M(2 + (3 - 1) \times 3) = M(8)$$



多維陣列

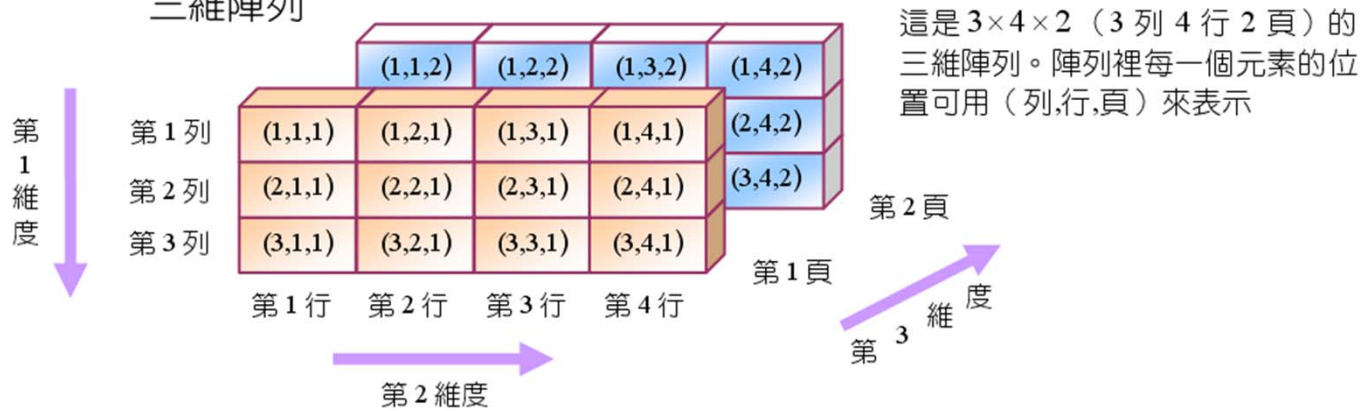
- 只要陣列的維度多於二維，我們就稱之為多維陣列
- 對於三維陣來說，需要列、行與頁（page）三個維度來描述




二維陣列



三維陣列



- 
-
- 要建立一個三維陣列，可針對每一頁分別建立二維陣列：

```
>> A(:,:,1)=[1 2 3 4;5 6 7 8;9 10 11 12]
```

```
A =  
     1     2     3     4  
     5     6     7     8  
     9    10    11    12
```

```
>> A(:,:,2)=[7 4 2 1;6 1 5 2;3 1 4 5]
```

```
A(:,:,1) =  
     1     2     3     4  
     5     6     7     8  
     9    10    11    12
```

```
A(:,:,2) =  
     7     4     2     1  
     6     1     5     2  
     3     1     4     5
```

常用的陣列建立函數

- Matlab常用的陣列建立函數如下表所列：

表 3.3.1 常用的矩陣建立函數

函數	說明
<code>zeros(n)</code>	建立一個 $n \times n$ 的全零矩陣
<code>zeros(m,n,...,p)</code>	建立一個 $m \times n \times \dots \times p$ 的全零矩陣
<code>ones(n)</code>	建立一個 $n \times n$ 的全 1 矩陣
<code>ones(m,n,...,p)</code>	建立一個 $m \times n \times \dots \times p$ 的全 1 矩陣
<code>eye(n)</code>	建立一個 $n \times n$ 的單位矩陣 (
<code>eye(m,n)</code>	建立一個 $m \times n$ ，且對角線為 1，其它元素為 0 的矩陣
<code>diag(v)</code>	以向量 v 為對角元素，建立一個矩陣
<code>magic(n)</code>	建立一個 $n \times n$ 的魔術方陣 (magic square)



○ 亂數陣列

Matlab也提供了產生均勻分佈，或者是常態分佈的亂數陣列函數：

表 3.3.2 以亂數來建立陣列之函數

函 數	說 明
rand	建立一個 0~1 之間均勻分佈的亂數
rand(<i>n</i>)	建立一個 0~1 之間，維度為 $n \times n$ 之均勻分佈的亂數矩陣
rand(<i>m</i> , <i>n</i> , ..., <i>p</i>)	建立一個 0~1 之間，維度為 $m \times n \times \dots \times p$ 之均勻分佈的亂數矩陣
randn	建立一個平均值為 0，標準差為 1 的常態分佈亂數
randn(<i>n</i>)	同 rand，但是建立常態分佈的亂數矩陣
randn(<i>m</i> , <i>n</i> , ..., <i>p</i>)	同 rand，但是建立常態分佈的亂數矩陣



陣列元素的其它操作

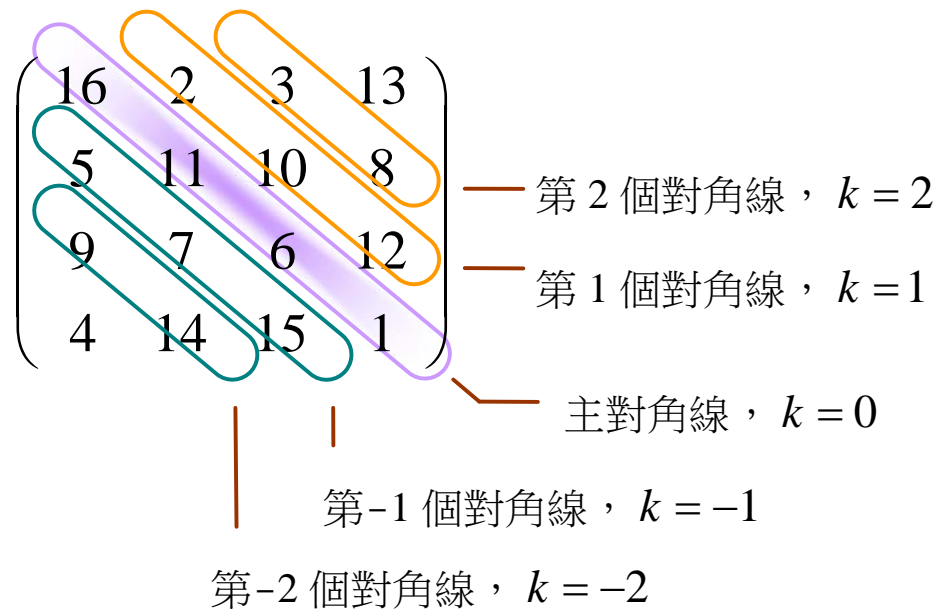
陣列元素的提取

- 要提取陣列的對角線元素，或是上三角形或下三角形矩陣，可利用如下表的函數：

表 3.4.1 陣列元素的提取函數

函 數	說 明
<code>diag(A)</code>	取出陣列 A 的主對角線 (main diagonal) 元素
<code>diag(A, k)</code>	取出陣列 A 的第 k 個對角線元素
<code>triu(A)</code>	取出陣列 A 之主對角線以上之元素，其它元素則設為 0
<code>triu(A, k)</code>	取出陣列 A 之第 k 個對角線以上之元素，其它元素則設為 0
<code>tril(A)</code>	取出陣列 A 之主對角線以下之元素，其它元素則設為 0
<code>tril(A, k)</code>	取出陣列 A 之第 k 個對角線以下之元素，其它元素則設為 0

- 主對角線與第 k 個對角線的示意圖：





陣列元素的重排

- 陣列轉換函數，可將陣列拆解成另一種形式：

表 3.4.2 陣列轉換函數

函 數	說 明
<code>fliplr(A)</code>	將陣列 A 的元素左右翻轉 (flip left/right)
<code>flipud(A)</code>	將陣列 A 的元素上下翻轉 (flip up/down)
<code>flipdim(A, n)</code>	將陣列 A 的元素依第 n 個維度翻轉
<code>reshape(A, m, n, ..., p)</code>	將陣列 A 的元素依由上到下，由左到右的次序重新排列成一個 $m \times n \times \dots \times p$ 的矩陣
<code>repmat(A, m, n, ..., p)</code>	以陣列 A 為單位，將陣列 A 以類似排列磁磚的方式排成 $m \times n \times \dots \times p$ 個陣列 A
<code>rot90(A)</code>	將陣列 A 逆時針旋轉 90°
<code>rot90(A, k)</code>	將陣列 A 逆時針旋轉 $k \times 90^\circ$ ， k 為整數



陣列的合併

- 合併陣列的指令如下表所示：

表 3.4.3 陣列轉換函數

函 數	說 明
$[A, B]$	將陣列 A, B 橫向併排，組合成一個新的陣列
$[A; B]$	將陣列 A, B 垂直併排，組合成一個新的陣列
$\text{cat}(dim, A, B, \dots)$	依 dim 所指定的方向合併陣列 A, B, \dots



矩陣的數學運算

基本的矩陣運算函數

- 下表列出了矩陣的基本運算函數：

矩陣的運算	說明
$A+B$	矩陣 A 加上矩陣 B
$A-B$	矩陣 A 減去矩陣 B
$A*B$	矩陣 A 乘上矩陣 B
A^n	矩陣 A 的 n 次方，即矩陣 A 連乘 n 次， A 必須為方陣
A'	計算矩陣 A 的共軛轉置 (conjugate transpose)
$\text{inv}(A)$	計算矩陣 A 的反矩陣 (inverse)
$\text{det}(A)$	計算矩陣 A 的行列式 (determinate)
$\text{expm}(A)$	計算矩陣 A 的指數 (matrix exponential)
$\text{logm}(A)$	計算矩陣 A 的對數 (matrix logarithm)
$\text{sqrtn}(A)$	計算矩陣 A 的開平方根




矩陣的左除與右除

- 左除「\」與右除「/」運算子，可分別用 $AX=B$ 與 $XA=B$ 來計算：

表 3.5.2 矩陣的數學運算

指令	說明
$A \setminus B$	A 左除 B ，此運算相當於把 A 的反矩陣乘以 B ，也就是 $A^{-1}B$
B / A	B 右除 A ，此運算相當於把 B 乘上 A 的反矩陣，也就是 BA^{-1}



例如，設


$$AX = B, \quad A = \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}, \quad B = \begin{pmatrix} 10 \\ 4 \end{pmatrix}$$

若要求解向量 X ，則

$$X = A^{-1}B = \begin{pmatrix} 1 & -2 \\ -0.5 & 1.5 \end{pmatrix} \begin{pmatrix} 10 \\ 4 \end{pmatrix} = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

上式可以寫上 $X = \text{inv}(A) * B$ ，或用 A 來左除 B ，即

>> A\B



右除是用在另外的一種情況，例如，設

$$XA = B \quad A = \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix} \quad B = (10 \quad 14)$$

則

$$X = BA^{-1} = (10 \quad 14) \begin{pmatrix} 1 & -2 \\ -0.5 & 1.5 \end{pmatrix} = (3 \quad 1)$$

上式可以寫上 $X=B*\text{inv}(A)$ ，或用 B 右除 A ，即

>> B/A



陣列元素對元素的運算

在 Matlab 裡，矩陣 A 乘上 B 可以寫成 $A*B$ 。例如，設

$$A = \begin{pmatrix} 2 & 4 \\ 3 & 1 \end{pmatrix}, B = \begin{pmatrix} 3 & 2 \\ 4 & 6 \end{pmatrix};$$

則

$$A*B = \begin{pmatrix} 2 & 4 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 3 & 2 \\ 4 & 6 \end{pmatrix} = \begin{pmatrix} 2 \times 3 + 4 \times 4 & 2 \times 2 + 4 \times 6 \\ 3 \times 3 + 1 \times 4 & 3 \times 2 + 1 \times 6 \end{pmatrix} = \begin{pmatrix} 22 & 28 \\ 13 & 12 \end{pmatrix}$$

如果是希望陣列 A 內的元素乘上陣列 B 內相對應的位置，也就是計算

$$A.*B = \begin{pmatrix} 2 & 4 \\ 3 & 1 \end{pmatrix} \begin{pmatrix} 3 & 2 \\ 4 & 6 \end{pmatrix} = \begin{pmatrix} 2 \times 3 & 4 \times 2 \\ 3 \times 4 & 1 \times 6 \end{pmatrix} = \begin{pmatrix} 6 & 8 \\ 12 & 6 \end{pmatrix}$$

則可以利用 Matlab 所提供的 "元素對元素"的計算指令


- 
-
- 元素對元素的計算符號是在矩陣的計算符號之前加上一個「.」：

表 3.5.3 陣列的數學運算

指令	說明
$A.*B$	將矩陣 A 內的元素乘上矩陣 B 內相同位置的元素
$A.^n$	計算矩陣 A 的 n 次方，即矩陣 A 連乘 n 次， A 必須為方陣
$A.'$	計算矩陣 A 的轉置 (transpose) 矩陣。
$A./B$	將 A 裡面的每一個元素除以 B 裡面每一個相對應的元素
$A.\backslash B$	將 B 裡面的每一個元素除以 A 裡面每一個相對應的元素



```
>> A=[2 4;3 1]
```

```
A =
```

```
    2    4  
    3    1
```

```
>> B=[3 2;4 6]
```

```
B =
```

```
    3    2  
    4    6
```

```
>> A*B
```

```
ans =
```

```
    22    28  
    13    12
```

```
>> A.*B
```

```
ans =
```

```
     6     8  
    12     6
```

常用的向量或矩陣函數

- 例如 $A=[1 \ -1 \ 1]$, $B=[-1 \ 1 \ 1]$, 則

運算函數	結果	說明
>> size(A)	ans = 1 3	陣列的大小(維度及元素數)
>> size(A')	ans = 3 1	轉置陣列的大小
>> length(A)	ans = 3	向量元素數目
>> prod(1:4)	ans = 24	元素的連乘積(即 $4!=1*2*3*4$)
>> prod(A)	ans = -1	向量元素的乘積(即 $1*(-1)*1$)
>> dot(A,B)	ans = -1	向量內積
>> A*B'	ans = -1	利用陣列乘法求向量內積
>> sum(A.*B)	ans = -1	利用陣列元素相乘求內積
>> sqrt(dot(A,A))	ans = 1.7321	利用向量內積求長度
>> norm(A)	ans = 1.7321	利用正規函數求向量長度
>> A/B	ans = -0.3333	向量投影
>> acos(A/B)*180/pi	ans = 109.4712	利用向量投影求夾角

常用的向量或矩陣函數

>> cross(A,B)	ans = -2 -2 0	向量外積
>> dot(A,cross(A,B))	ans = 0	驗證向量內外積值為0
>> diff(A)	ans = -2 2	向量相鄰元素差值
>> sort(A)	ans = -1 1 1	向量或陣列元素排序
>> max(A)	ans = 1	向量或陣列元素最大值
>> min(A)	ans = -1	向量或陣列元素最小值
>> mean(B)	ans = 0.3333	向量或陣列元素均值
>> std(B)	ans = 1.1547	向量或陣列元素標準差
>> median(B)	ans = 1	向量或陣列元素中位數
>> cov(A)	ans = 1.3333	向量或陣列元素斜方差
>> ismember(A,1)	ans = 1 0 1	檢察輸入元素1使否屬於A
>> intersect(A,B)	ans = -1 1	取AB交集
>> union(A,B)	ans = -1 1	取AB聯集
>> setdiff(A,B)	ans = Empty matrix:1-by-0	取AB差集
>> unique(A)	ans = -1 1	傳回A集合中唯一的元素
>> setxor(A,B)	ans = Empty matrix:1-by-0	取AB之XOR
>> eig(A)	[V,D] = eig(A)	計算A之特徵值及特徵向量