

第四章 陣列的基本操作與運算

本章學習目標

- 認識陣列裡元素的結構
- 學習多維陣列的建立
- 學習編修矩陣的內容
- 學習基本的矩陣數學運算

向量元素的操作 (1/2)

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

```
>> 2*v1+1
ans =
    13    15    17    19
```

```
>> v1(2)
ans =
     7
```

```
>> v1([2,4])
ans =
    1.9
```

```
>> v1([2 4])
ans =
     7     9
```

```
>> v1(:)
ans =
     6
     7
     8
     9
```

```
>> v1([2])
ans =
     7
```

```
>> v1(3)=0
v1 =
     6     7     0     9
```

```
>> v1(5)
??? Index exceeds matrix dimensions.
```

向量元素的操作 (2/2)

```
>> v1(7)=12
```

```
v1 =  
    6    7    0    9    0    0   12
```

```
>> v1(3:5)=1
```

```
v1 =  
    6    7    1    1    1    0   12
```

```
>> v1(end)
```

```
ans =  
    12
```

```
>> v1(end-1)
```

```
ans =  
     0
```

```
>> v1(1:2:end)
```

```
ans =  
     6     1     1   12
```

```
>> v1(end:-1:1)
```

```
ans =  
    12     0     1     1     1     7     6
```

```
>> v1(6)=[]
```

```
v1 =  
     6     7     1     1     1   12
```

```
>> v1(5:end)=[]
```

```
v1 =  
     6     7     1     1
```


矩陣元素的操作 (1/2)

```
>> 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
```

```
>> M(3,1:3)
```

```
ans =
     9    10    11
```

```
>> M(3,:)
```

```
ans =
     9    10    11    12
```

```
>> M(end,3)
```

```
ans =
    11
```

```
>> M(1:2,4)
```

```
ans =
     4
     8
```

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

```
ans =
     2     3
     6     7
    10    11
```

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

```
ans =
    10    11
```

矩陣元素的操作 (2/2)

```
>> M(2,3)=99
```

M =

1	2	3	4
5	6	99	8
9	10	11	12

```
>> M(end,:)=[]
```

M =

1	2	3	4
5	6	99	8

```
>> M(:, [2,4])=[]
```

M =

1	3
5	99

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

Subscripted assignment dimension mismatch.

```
>> M=[M,[4;7]]
```

M =

1	3	4
5	99	7

```
>> M=[[8,9,10];M]
```

M =

8	9	10
1	3	4
5	99	7

```
>> M(3:-1:1,:)
```

ans =

5	99	7
1	3	4
8	9	10

```
>> M(2,:)=0
```

M =

8	9	10
0	0	0
5	99	7

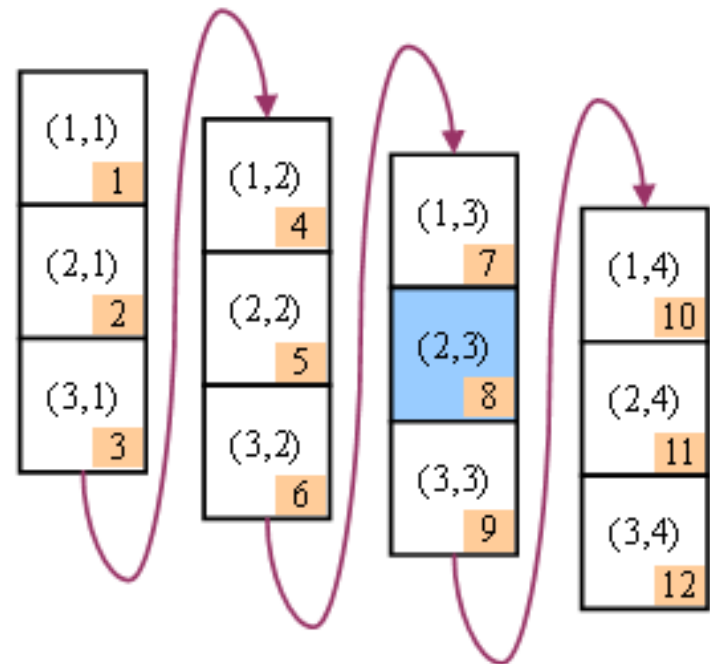
矩陣的索引值之結構

- Matlab的矩陣是利用「以行為主」的結構來儲存
- 每一個陣列可以看成是由數個行向量串接而成
- 存取矩陣的元素時，可以使用一維或二維索引值來存取

矩陣 M

第 1 列	(1,1) 1	(1,2) 4	(1,3) 7	(1,4) 10
第 2 列	(2,1) 2	(2,2) 5	(2,3) 8	(2,4) 11
第 3 列	(3,1) 3	(3,2) 6	(3,3) 9	(3,4) 12
	第 1 行	第 2 行	第 3 行	第 4 行

拆成行向量後的矩陣 M



矩陣的索引|值的範例

```
>> M=[3 6 4 3;7 4 9 8;0 4 3 1]
```

```
M =
```

```
     3     6     4     3
     7     4     9     8
     0     4     3     1
```

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

```
ans =
```

```
     9
```

```
>> M(8)
```

```
ans =
```

```
     9
```

```
>> M(4:7)
```

```
ans =
```

```
     6     4     4     4
```

```
>> M([5;7;9])
```

```
ans =
```

```
     4
```

```
     4
```

```
     3
```

```
> M([4 6;8 9])
```

```
ans =
```

```
     6     4
     9     3
```

```
>> M(:)
```

```
ans =
```

```
     3
```

```
     7
```

```
     0
```

```
     6
```

```
     4
```

```
     4
```

```
     4
```

```
     9
```

```
     3
```

```
     3
```

```
     8
```

```
     1
```

矩陣元陣的提取

矩陣 M

第一列	3 1	6 4	4 7	3 10
第二列	7 2	4 5	9 8	8 11
第三列	0 3	4 6	3 9	1 12
	第一行	第二行	第三行	第四行

只有一個引數（一個二維陣列）

$M([4, 6; 8, 9])$

ans =

6 4

9 3

取出第 4, 6, 8, 9 個元素，
然後將它們排成一個二
維陣列。

有兩個引數（兩個一維陣列）

$M([1, 3], [1, 3, 4])$

ans =

3 4 3

0 3 1

取出第 1, 3 列，第 1, 3, 4 行
的元素

矩陣元陣提取的範例

```
>> M([1,3],[1,3,4])
```

```
ans =
```

```
3     4     3
0     3     1
```

```
>> M(2:3,1:2)
```

```
ans =
```

```
7     4
0     4
```

矩陣 M

第一列	3 1	6 4	4 7	3 10
第二列	7 2	4 5	9 8	8 11
第三列	0 3	4 6	3 9	1 12
	第一行	第二行	第三行	第四行

```
>> M(4:6)=0
```

```
M =
```

```
3     0     4     3
7     0     9     8
0     0     3     1
```

```
>> M(2:3,2:4)=1
```

```
M =
```

```
3     0     4     3
7     1     1     1
0     1     1     1
```

```
>> M([4,8,12])=[]
```

```
M =
```

```
3     7     0     1     1     4     1     3     1
```

一維與二維索引值的轉換

表 4.1.1 一維索引值與二維索引值的轉換

函 數	說 明
<code>ind=sub2ind(size,row,col)</code>	將大小為 <i>size</i> 之矩陣的二維索引值 (<i>row,col</i>)轉換成一維索引值 <i>ind</i>
<code>[row,col]=ind2sub(size,ind)</code>	將大小為 <i>size</i> 之矩陣的一維索引值 <i>ind</i> 轉換成二維索引值 (<i>row,col</i>)

```
>> ind=sub2ind([3,4],2,3)
ind =
     8
```

```
>> [row,col]=ind2sub([3,4],8)
row =
     2
col =
     3
```

3 1	6 4	4 7	3 10
7 2	4 5	9 8	8 11
0 3	4 6	3 9	1 12

```
>> [row,col]=ind2sub([3,4],[8,4,12])
row =
     2     1     3
col =
     3     2     4
```

```
>> ind=sub2ind([3,4],row,col)
ind =
     8     4    12
```

A=[2 4 6;5 7 3;2 5 0]

>> max(A)

ans =

5 7 6

>>mean(A)

ans =

3.0000 5.3333 3.0000

$$A = \begin{bmatrix} 2 & 4 & 6 \\ 5 & 7 & 3 \\ 2 & 5 & 0 \end{bmatrix}$$

>> max(A(:))

ans =

7

>> mean(A(:))

ans =

3.7778

陣列中有Nan值的最大值與平均值的運算

nanmean(x) Mean ignoring NaNs

```
B=[nan 1 3;2 4 1;1 0 nan]
```

```
B =
```

```
NaN    1    3
```

```
    2    4    1
```

```
    1    0 NaN
```

```
>> max(B)
```

```
ans =
```

```
    2    4    3
```

```
>> max(B(:))
```

```
ans =
```

```
    4
```

```
>> mean(B)
```

```
ans =
```

```
NaN    1.6667    NaN
```

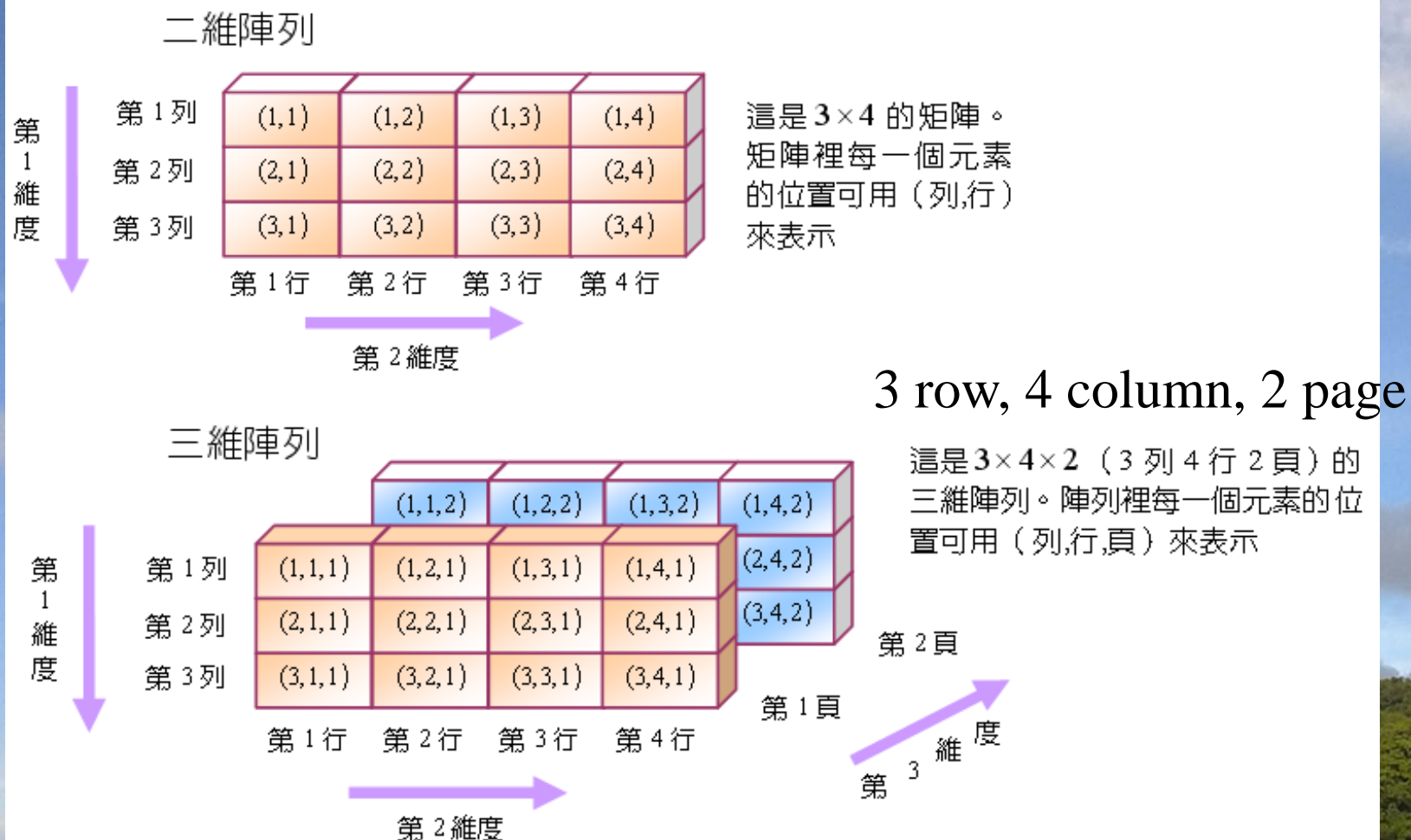
```
>> mean(B(:))
```

```
ans =
```

```
NaN
```

多維陣列

- 陣列的維度多於二維，稱為多維陣列
- 三維陣來必須以列、行與頁三個維度來描述



建立三維陣列的範例

```
>> 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

```
>> size(A)
```

```
ans =
```

```
3      4      2
```

```
>> ndims(A)
```

```
ans =
```

```
3
```

```
>> numel(A)
```

```
ans =
```

```
24
```

```
>> A(11:14)
```

```
ans =
```

```
8      12      7      6
```

```
>> A(1,3,2)
```

```
ans =
```

```
2
```


三維陣列的最大值與平均值

```
>> max(C)
```

```
ans(:,:,1) =
```

```
    9    10    11    12
```

```
ans(:,:,2) =
```

```
    7     4     5     5
```

```
>> max(C(:))
```

```
ans =
```

```
    12
```

```
>> mean(C)
```

```
ans(:,:,1) =
```

```
    5     6     7     8
```

```
ans(:,:,2) =
```

```
  5.3333  2.0000  3.6667  2.6667
```

```
>> mean(C(:))
```

```
ans =
```

```
  4.9583
```

常用的陣列建立函數

表 4.3.1 常用的陣列建立函數

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（對角線元素為 1，其它元素為 0）
<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 ）

陣列建立函數使用範例

```
>> zeros(3)
```

```
ans =  
      0      0      0  
      0      0      0  
      0      0      0
```

```
>> zeros(2,3)
```

```
ans =  
      0      0      0  
      0      0      0
```

```
>> ones(3,2)
```

```
ans =  
      1      1  
      1      1  
      1      1
```

```
>> eye(4)
```

```
ans =  
      1      0      0      0  
      0      1      0      0  
      0      0      1      0  
      0      0      0      1
```

```
>> eye(3,4)
```

```
ans =  
      1      0      0      0  
      0      1      0      0  
      0      0      1      0
```

```
>> diag([1 2 3])
```

```
ans =  
      1      0      0  
      0      2      0  
      0      0      3
```

```
>> magic(3)
```

```
ans =  
      8      1      6  
      3      5      7  
      4      9      2
```

```
>> magic(4)
```

```
ans =  
     16      2      3     13  
      5     11     10      8  
      9      7      6     12  
      4     14     15      1
```


建立亂數陣列

表 4.3.2 以亂數來建立陣列之函數

函 數	說 明
<code>randi(imax,n)</code>	建立 $n \times n$ 個 1 到 $imax$ 之間均勻分佈的整數亂數
<code>randi(imax,[m,n,...,p])</code>	建立 $m \times n \times \cdots \times p$ 個 1 到 $imax$ 之間均勻分佈的整數亂數
<code>randi([imin,imax],[m,n,...,p])</code>	同上，但整數亂數的範圍為 $imin$ 到 $imax$
<code>rand()</code>	建立一個 0 到 1 之間均勻分佈的亂數
<code>rand(n)</code>	建立 $n \times n$ 個 0 到 1 之間均勻分佈的亂數
<code>rand(m,n,...,p)</code>	建立 $m \times n \times \cdots \times p$ 個 0 到 1 之間均勻分佈的亂數
<code>randn()</code>	建立一個平均值為 0，標準差為 1 的常態分佈亂數
<code>randn(n)</code>	建立 $n \times n$ 個 0 到 1 之間常態分佈的亂數
<code>randn(m,n,...,p)</code>	建立 $m \times n \times \cdots \times p$ 個 0 到 1 之間常態分佈的亂數
<code>rng(seed)</code>	設定亂數種子為 $seed$

亂數陣列的使用範例

```
>> randi(9,[2,5])
```

```
ans =
```

2	4	2	3	2
5	8	4	7	8

```
>> randi([0,1],[1,6])
```

```
ans =
```

0	1	1	1	0	0
---	---	---	---	---	---

```
>> rand()
```

```
ans =
```

0.9501

```
>> rand(3)
```

```
ans =
```

0.2311	0.8913	0.0185
0.6068	0.7621	0.8214
0.4860	0.4565	0.4447

```
>> rng(999)
```

```
>> R=randi(9,[1,8])
```

```
R =
```

8	5	2	6	1	3	4	5
---	---	---	---	---	---	---	---

```
>> rng(999)
```

```
>> randi(9,[1,2])
```

```
ans =
```

1.5

```
>> randi(9,[1,1])
```

```
ans =
```

2

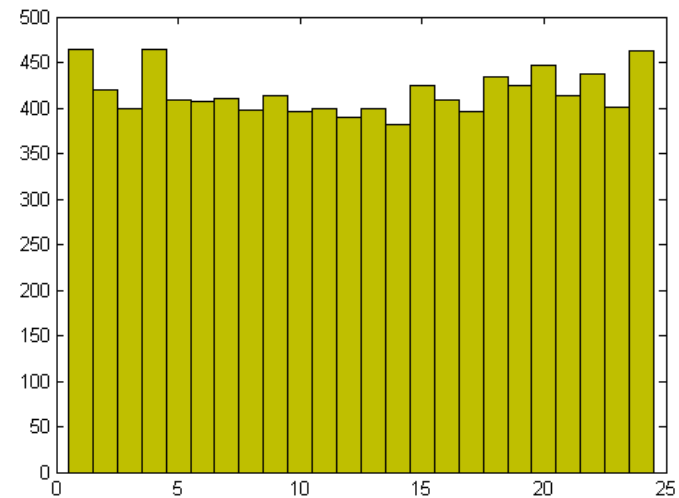
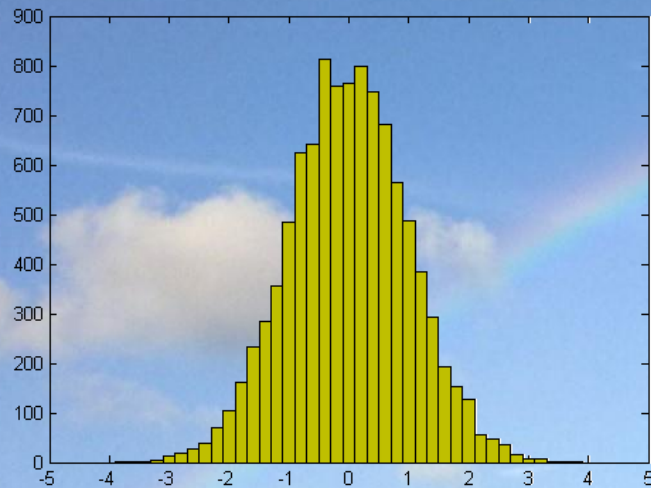
```
>> randi(9,[1,3])
```

```
ans =
```

6	1	3
---	---	---

測試亂數分佈的情況

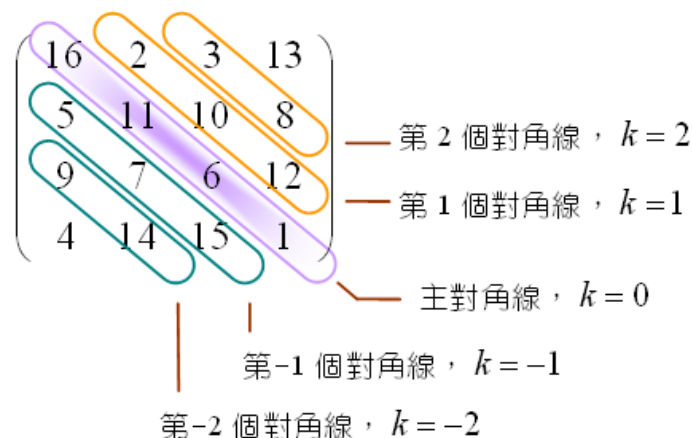
```
>> A=randn(1,10000) >> B=randi([1,24],[1,10000]);  
>> hist(A,-4:0.2:4) >> hist(B,1:24)
```



陣列元素的提取

表 4.4.1 陣列元素的提取函數

函 數	說 明
$\text{diag}(A)$	取出陣列 A 的主對角線 (main diagonal) 元素
$\text{diag}(A, k)$	取出陣列 A 的第 k 個對角線元素
$\text{triu}(A)$	取出陣列 A 之主對角線以上之元素，其它元素則設為 0 (即上三角矩陣， upper triangular matrix)
$\text{triu}(A, k)$	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
$\text{tril}(A)$	取出陣列 A 之主對角線以下之元素，其它元素則設為 0 (即下三角矩陣， lower triangular matrix)
$\text{tril}(A, k)$	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



元素的提取的範例

```
>> A=magic(4)
```

```
A =
```

16	2	3	13
5	11	10	8
9	7	6	12
4	14	15	1

```
>> diag(A)
```

```
ans =
```

16
11
6
1

```
>> diag(A,-1) '
```

```
ans =
```

5	7	15
---	---	----

```
>> diag(A,2)
```

```
ans =
```

3
8

```
>> triu(A,1)
```

```
ans =
```

0	2	3	13
0	0	10	8
0	0	0	12
0	0	0	0

```
>> tril(A,-2)
```

```
ans =
```

0	0	0	0
0	0	0	0
9	0	0	0
4	14	0	0

陣列元素的重排

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的元素依由上而下，由左而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

重排陣列元素的範例

```
>> A=magic(3)
```

```
A =
```

8	1	6
3	5	7
4	9	2

```
>> fliplr(A)
```

```
ans =
```

6	1	8
7	5	3
2	9	4

```
>> flipdim(A,2)
```

```
ans =
```

6	1	8
7	5	3
2	9	4

```
>> reshape(A,1,9)
```

```
ans =
```

8	3	4	1	5	9	6	7	2
---	---	---	---	---	---	---	---	---

```
>> reshape(A,2,4)
```

Error using reshape

To RESHAPE the number of elements must not change.

```
>> repmat(A,2,2)
```

```
ans =
```

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

```
>> repmat(9,[3,4,2])
```

```
ans(:,:,1) =
```

9	9	9	9
9	9	9	9
9	9	9	9

```
ans(:,:,2) =
```

9	9	9	9
9	9	9	9
9	9	9	9

- $Y = \text{circshift}(A, K, \text{dim})$ circularly shifts the values in array A by K positions along dimension dim . Inputs K and dim must be scalars.

$a = [1 \ 2 \ 3; 11 \ 12 \ 13; 21 \ 22 \ 23];$

$bh1 = \text{circshift}(a, 1, 1)$

$bh2 = \text{circshift}(a, 2, 1)$

$bv1 = \text{circshift}(a, 1, 2)$

$bv2 = \text{circshift}(a, 2, 2)$

$$a = \begin{bmatrix} 1 & 2 & 3 \\ 11 & 12 & 13 \\ 21 & 22 & 23 \end{bmatrix}$$

$$bh1 = \begin{bmatrix} 21 & 22 & 23 \\ 1 & 2 & 3 \\ 11 & 12 & 13 \end{bmatrix}$$

$$bh2 = \begin{bmatrix} 11 & 12 & 13 \\ 21 & 22 & 23 \\ 1 & 2 & 3 \end{bmatrix}$$

$$bv1 = \begin{bmatrix} 3 & 1 & 2 \\ 13 & 11 & 12 \\ 23 & 21 & 22 \end{bmatrix} \quad bv2 = \begin{bmatrix} 2 & 3 & 1 \\ 12 & 13 & 11 \\ 22 & 23 & 21 \end{bmatrix}$$

陣列的合併

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
cat(dim, A, B, ...)	依dim所排定的方向合併排(concatenate) 陣列A, B, ...

```
>> A=[1 2 3;3 4 5]
```

```
A =
```

```
     1     2     3
     3     4     5
```

```
>> B=2*A
```

```
B =
```

```
     2     4     6
     6     8    10
```

```
>> [A,B]
```

```
ans =
```

```
     1     2     3     2     4     6
     3     4     5     6     8    10
```

```
>> [A;B]
```

```
ans =
```

```
     1     2     3
     3     4     5
     2     4     6
     6     8    10
```


cat() 的使用範例

```
>> cat(1,A,B)
```

```
ans =
```

1	2	3
3	4	5
2	4	6
6	8	10

```
>> A=[1 2 3;3 4 5]
```

```
A =
```

1	2	3
3	4	5

```
>> B=2*A
```

```
B =
```

2	4	6
6	8	10

```
>> cat(2,A,B)
```

```
ans =
```

1	2	3	2	4	6
3	4	5	6	8	10

```
>> cat(2,A,A,B)
```

```
ans =
```

1	2	3	1	2	3	2	4	6
3	4	5	3	4	5	6	8	10

```
>> cat(3,A,B)
```

```
ans(:, :, 1) =
```

1	2	3
3	4	5

```
ans(:, :, 2) =
```

2	4	6
6	8	10

```
>> cat(4,A,B)
```

```
ans(:, :, 1, 1) =
```

1	2	3
3	4	5

```
ans(:, :, 1, 2) =
```

2	4	6
6	8	10

基本的矩陣運算

矩陣的數學運算

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)。如困矩陣A的所有元素都實數，則A'相當於是A的轉置矩陣
$\text{inv}(A)$	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(matric logarithm)
$\text{sqrtm}(A)$	計算矩陣A的平方根

矩陣運算的範例 (1/2)

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

```
A =
```

```
     2     4
     3     1
```

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

```
B =
```

```
     3     2
     4     6
```

```
>> A+3
```

```
ans =
```

```
     5     7
     6     4
```

```
>> A+B
```

```
ans =
```

```
     5     6
     7     7
```

```
>> [2,3]*A
```

```
ans =
```

```
1. 11
```

```
>> A*[2,3]
```

```
Error using *
```

```
Inner matrix dimensions must agree.
```

```
>> A*[1;5]
```

```
ans =
```

```
    22
     8
```

```
>> A*B
```

```
ans =
```

```
    22    28
    13    12
```

```
>> A'
```

```
ans =
```

```
     2     3
     4     1
```


矩陣運算的範例 (2/2)

```
>> inv(A)
```

```
ans =  
    -0.1000    0.4000  
    0.3000   -0.2000
```

```
>> det(A)
```

```
ans =  
   -10
```

```
>> M=expm(A)
```

```
M =  
    84.8655    84.7302  
    63.5476    63.6830
```

```
>> logm(M)
```

```
ans =  
    2.0000    4.0000  
    3.0000    1.0000
```

```
>> Z=sqrtm(A)
```

```
ans =  
    1.2778+0.6061i    1.2778-0.8081i  
    0.9583-0.6061i    0.9583+0.8081i
```

```
>> Z^2
```

```
ans =  
    2.0000+0.0000i    4.0000+0.0000i  
    3.0000-0.0000i    1.0000+0.0000i
```

```
>> real(Z^2)
```

```
ans =  
    2.0000    4.0000  
    3.0000    1.0000
```

```
>> A^0.5
```

```
ans =  
    1.2778+0.6061i    1.2778-0.8081i  
    0.9583-0.6061i    0.9583+0.8081i
```

```
>> A^-1
```

```
ans =  
   -0.1000    0.4000  
    0.3000   -0.2000
```

$$A = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

$$A^{-1} = \frac{\begin{bmatrix} d & -b \\ -c & a \end{bmatrix}}{\det(A)}$$

矩陣的左除(\)與右除(/)

- 左除(\)計算 $AX=B$
- 右除(/)計算 $XA=B$

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 $A^{-1}B$ $A \setminus B$ 的結果相當於 $AX=B$ 的解。亦即，若 $AX=B$ ，則 $X=A^{-1}B=A \setminus B$
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} B/A 的結果相當於 $XA=B$ 的解。亦即，若 $XA=B$ ，則 $X=BA^{-1}=B/A$

- 矩陣的左除應用

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

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

- 矩陣的右除應用

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

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

求解 $AX=B$

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

```
A =
```

```
    3    4
```

```
    1    2
```

```
>> B=[10;4]
```

```
B =
```

```
    10
```

```
     4
```

```
>> inv(A)*B
```

```
ans =
```

```
     2
```

```
     1
```

```
>> A\B
```

```
ans =
```

```
    2.0000
```

```
    1.0000
```

求解 $XA=B$

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

```
A =
```

```
3    4
```

```
1    2
```

```
>> B=[10 14]
```

```
B =
```

```
10   14
```

```
>> B*inv(A)
```

```
ans =
```

```
3    1
```

```
>> B/A
```

```
ans =
```

```
3    1
```


- 解二元一次聯立方程式

$$\begin{cases} x + y = 6 \\ x - y = 2 \end{cases}$$

$$A = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}, B = \begin{bmatrix} 6 \\ 2 \end{bmatrix}$$

$$\text{Delt}(A) = \begin{vmatrix} 1 & 1 \\ 1 & -1 \end{vmatrix} = -2$$

$$\text{Inv}(A) = \begin{bmatrix} \frac{-1}{-2} & -\frac{1}{-2} \\ -\frac{1}{-2} & \frac{1}{-2} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 \\ 0.5 & -0.5 \end{bmatrix}$$

$$\text{inv}(A)A X = \text{inv}(A)B, \quad X = \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

練習

- 解二元一次聯立方程式

$$\begin{cases} 8x - \frac{1}{8}y = 190 \\ -\frac{1}{8}x + 8y = 125 \end{cases}$$

$$x=24, y=16$$

陣列元素對元素的運算

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內，個別元素的n次方
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.\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
```

```
>> A^3  
ans =  
    68    76  
    57    49
```

```
>> A.^3  
ans =  
     8    64  
    27     1
```

隨堂測驗

矩陣A, B, C, X, F分別為

$$A = \begin{pmatrix} 3 & 2 & 4 \\ 5 & 7 & 3 \\ 1 & 6 & 0 \end{pmatrix}$$

$$B = \begin{pmatrix} 0 & 2 & -2 \\ 7 & 4 & 3 \\ 8 & -4 & -5 \end{pmatrix}$$

$$X = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix}$$

$$F = \begin{pmatrix} -6 \\ 2 \\ 1 \end{pmatrix}$$

$C(:,:,1)=A$; $C(:,:,2)=B$,

1. $[r,c,p]=\text{ind2sub}(\text{size}(C),10)$, 則r, c, p分別為何?
2. $\text{ind0} = \text{sub2ind}(\text{size}(C), 2,2,2)$, 則ind0值為何?
3. $E = \text{reshape}(A,1,9)$, 則 $\text{sum}(E)$ 值為何?
4. $A \cdot X=F$, 求 (x_1,x_2,x_3) 分別為何?