

FORTRAN

高維陣列

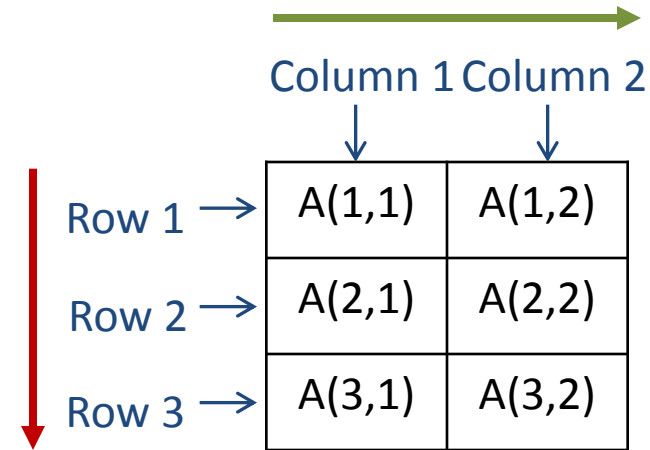
(multi-dimensional arrays)

多維陣列

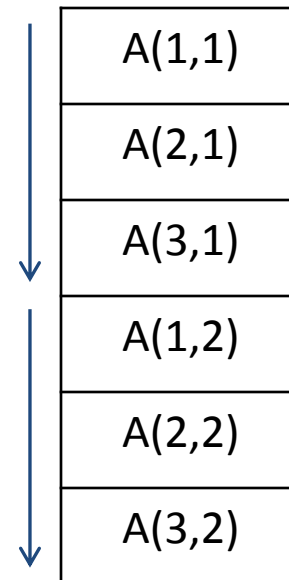
- 前面介紹的都是一維陣列，但FORTRAN可以接受多維陣列，宣告時以DIMENSION設定各維度有幾個元素，例如下面的二維陣列：

REAL, DIMENSION (3, 2) :: A

1st dimension 2nd dimension



- 記憶體中儲存二維陣列元素的順序：
先儲存第一直行，
再接著儲存第二直行
(Column major order)



多維陣列

- 宣告多維陣列時，一樣可以設定subscript的範圍：

REAL, DIMENSION (3,2) :: A

INTEGER, DIMENSION (-1:1,0:2) :: B

CHARACTER(len=10), DIMENSION (-1:1,1:3:2) :: name

| | |
|--------|--------|
| A(1,1) | A(1,2) |
| A(2,1) | A(2,2) |
| A(3,1) | A(3,2) |

每個元素儲存
一個實數

| | | |
|---------|---------|---------|
| B(-1,0) | B(-1,1) | B(-1,2) |
| B(0,0) | B(0,1) | B(0,2) |
| B(1,0) | B(1,1) | B(1,2) |

每個元素儲存
一個整數

| | |
|------------|------------|
| name(-1,1) | name(-1,3) |
| name(0,1) | name(0,3) |
| name(1,1) | name(1,3) |

每個元素儲存一個
含有10字元的字串

利用巢狀迴圈將數值填入多維陣列一範例

```
PROGRAM 2DArray
INTEGER, DIMENSION (3,5) :: a=0
INTEGER :: I, J
DO J=1,5
  DO I=1,3
    a(I,J) = I+J
  END DO
END DO
WRITE (*,*) a(:,3)
END PROGRAM 2DArray
```

1. 程式中宣告一個陣列。請問該陣列屬於什麼類型（整數、實數...etc）？

Integer

2. 該陣列有幾個「維度」（dimension）？

2-D

3. 該陣列總共有幾個元素？

3x5=15

4. 請問程式運算結束後，a(2,5)的數值是多少？

a(2,5)=2+5=7

| | | | | |
|----------------|----------------|----------------|----------------|----------------|
| a(1,1)= 1+1 | a(1,2)= 1+2 | a(1,3)= 1+3 | a(1,4)= 1+4 | a(1,5)= 1+5 |
| a(2,1)= 2+1 | a(2,2)= 2+2 | a(2,3)= 2+3 | a(2,4)= 2+4 | a(2,5)= 2+5 |
| a(3,1)= 3+1 | a(3,2)= 3+2 | a(3,3)= 3+3 | a(3,4)= 3+4 | a(3,5)= 3+5 |

二維陣列與巢狀迴圈一小提醒

- 注意Do 迴圈counter與陣列subscript是否一致
 - 外層、內層迴圈的counter，相對應於陣列的第幾個dimension？
 - Counter變動的範圍有無超過陣列宣告的大小？
- 因為記憶體儲存是column major order，外層的counter若對應陣列的第二個dimension，就不用跳躍式讀寫記憶體的資料，會加快程式執行速度

(慢)

```
DO I=1,3
  DO J=1,5
    a(I,J) = REAL(I+J)
  END DO
END DO
```

(快)

```
DO J=1,5
  DO I=1,3
    a(I,J) = REAL(I+J)
  END DO
END DO
```

(j=1) (j=2) ...

↓ ↓

| | |
|--------|--------|
| a(1,1) | a(1,2) |
| a(2,1) | a(2,2) |
| a(3,1) | a(3,2) |

多維陣列—取出部分元素

- 語法和一維陣列相同。舉例：

| | | | | |
|----------------|----------------|----------------|----------------|----------------|
| a(1,1)= 1+1 | a(1,2)= 1+2 | a(1,3)= 1+3 | a(1,4)= 1+4 | a(1,5)= 1+5 |
| a(2,1)= 2+1 | a(2,2)= 2+2 | a(2,3)= 2+3 | a(2,4)= 2+4 | a(2,5)= 2+5 |
| a(3,1)= 3+1 | a(3,2)= 3+2 | a(3,3)= 3+3 | a(3,4)= 3+4 | a(3,5)= 3+5 |

WRITE (*, *) a (1, :)

→

| | | | | |
|-----|-----|-----|-----|-----|
| 1+1 | 1+2 | 1+3 | 1+4 | 1+5 |
|-----|-----|-----|-----|-----|

WRITE (*, *) a (:, 3)

→

| |
|-----|
| 1+3 |
| 2+3 |
| 3+3 |

WRITE (*, *) a (2:3, 1:5:2)

→

| | | |
|-----|-----|-----|
| 2+1 | 2+3 | 2+5 |
| 3+1 | 3+3 | 3+5 |

利用巢狀迴圈將數值填入多維陣列一範例

```
PROGRAM 2DArray
INTEGER, DIMENSION(3,5) :: a=0
INTEGER :: I, J
DO J=1,5
  DO I=1,3
    a(I,J) = I+J
  END DO
END DO
WRITE(*,*) a(:,3)
END PROGRAM 2DArray
```

5. 請問若執行此程式，螢幕上輸出的結果為？

a(:,3)
= a(1,3), a(2,3), a(3,3)
= 4, 5, 6

| | | | | |
|----------------|----------------|----------------|----------------|----------------|
| a(1,1)= 1+1 | a(1,2)= 1+2 | a(1,3)= 1+3 | a(1,4)= 1+4 | a(1,5)= 1+5 |
| a(2,1)= 2+1 | a(2,2)= 2+2 | a(2,3)= 2+3 | a(2,4)= 2+4 | a(2,5)= 2+5 |
| a(3,1)= 3+1 | a(3,2)= 3+2 | a(3,3)= 3+3 | a(3,4)= 3+4 | a(3,5)= 3+5 |

從檔案讀取數值存到多維陣列一範例

文字檔
(array1.txt)

```
1.1, 1.2, 1.3, 1.4, 1.5  
2.1, 2.2, 2.3, 2.4, 2.5  
3.1, 3.2, 3.3, 3.4, 3.5
```

```
PROGRAM 2DArray_read  
REAL, DIMENSION(3,5) :: b=0.  
INTEGER :: I, J  
OPEN(10,file='array1.txt')  
DO I=1,3  
    READ(10,*) (b(I,J), J=1,5)  
END DO  
WRITE(*,*) b(2,4:5)  
END PROGRAM 2DArray_read
```

利用一層**顯式迴圈**（**explicit loop**）跟一層**隱式迴圈**（**implicit loop**），**一列一列**將資料讀入陣列中

| | | | | |
|----------------|----------------|----------------|----------------|----------------|
| b(1,1) =1.1 | b(1,2) =1.2 | b(1,3) =1.3 | b(1,4) =1.4 | b(1,5) =1.5 |
| b(2,1) =2.1 | b(2,2) =2.2 | b(2,3) =2.3 | b(2,4) =2.4 | b(2,5) =2.5 |
| b(3,1) =3.1 | b(3,2) =3.2 | b(3,3) =3.3 | b(3,4) =3.4 | b(3,5) =3.5 |

從檔案讀取數值存到多維陣列—練習

文字檔
(array2.txt)

```
A1    B1    A2    B2    A3    B3
1.1,  1.2,  1.3,  1.4,  1.5,  1.6
2.1,  2.2,  2.3,  2.4,  2.5,  2.6
```

如何將資料分別讀入
A與B兩個2x3的陣列？

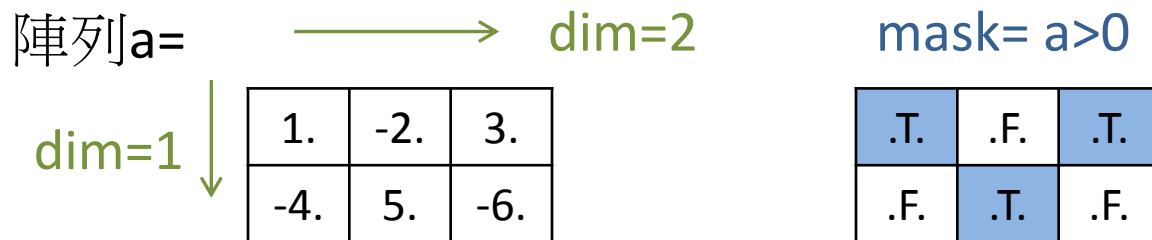
```
PROGRAM 2DArray_read2
REAL, DIMENSION(2,3) :: A=0., B=0.
INTEGER :: I, J
OPEN(10,file='array2.txt')
READ(10,*) ! pass header in file
DO I=1,2
    READ(10,*) (A(I,J),B(I,J), J=1,3)
END DO
WRITE(*,*) A(2,3), B(1,2)
END PROGRAM 2DArray_read2
```

| | | |
|----------------|----------------|----------------|
| A(1,1) =1.1 | A(1,2) =1.3 | A(1,3) =1.5 |
| A(2,1) =2.1 | A(2,2) =2.3 | A(2,3) =2.5 |

| | | |
|----------------|----------------|----------------|
| B(1,1) =1.2 | B(1,2) =1.4 | B(1,3) =1.6 |
| B(2,1) =2.2 | B(2,2) =2.4 | B(2,3) =2.6 |

陣列相關的內建函數

- 之前介紹過幾個用在完整陣列的內建函數，一樣可以使用在多維陣列上，並且可以選擇只對特定的維度做運算
 - 語法舉例：`SUM(array, [dim], [mask])`
 - `array` = 要運算的陣列；
 - `dim` = 要進行運算的維度（如果不指定，就對整個陣列做運算）
 - `mask` = 邏輯判斷陣列（如`array>0.0`）篩選符合條件的元素



陣列相關的內建函數—加總、計數

- `SUM(array, [dim], [mask])`

- 將陣列元素加總

`b = SUM(a)`

`b = -3.`

`c = SUM(a, a>0.)`

`c = 1.+3.+5. = 9.`

`d = SUM(a, 2, a>0.)`

`d = [(1.+3.), 5.] = [4., 5.]`

a=

| | | |
|-----|-----|-----|
| 1. | -2. | 3. |
| -4. | 5. | -6. |

mask= a>0

| | | |
|-----|-----|-----|
| 1. | -2. | 3. |
| -4. | 5. | -6. |

- `COUNT(mask, [dim])`

- 計算mask條件成立的元素總共有幾個（結果為整數）

`e = COUNT(a>0.)`

`e = 3`

`f = COUNT(a>0., 2)`

`f = [2, 1]`

陣列相關的內建函數—最大最小值

- `MAXVAL(array, [dim], [mask])`, `MINVAL(array, [dim], [mask])`

– 找出陣列元素中的最大值、最小值

`b = MAXVAL(a)`

`b = 5.`

`c = MINVAL(a, a>0.)`

`c = 1.`

`d = MINVAL(a, 1)`

`d = [-4, -2, -6.]`

a=

| | | |
|-----|-----|-----|
| 1. | -2. | 3. |
| -4. | 5. | -6. |

mask= a>0

| | | |
|-----|-----|-----|
| 1. | -2. | 3. |
| -4. | 5. | -6. |

- `MAXLOC(array, [dim], [mask])`, `MINLOC(array, [dim], [mask])`

– 找出陣列元素中的最大值、最小值所在的subscript位置（運算結果是整數）

`e = MAXLOC(a)`

`e = [2, 2]`

`f = MINLOC(a, a>0.)`

`f = [1, 1]`

`g = MINLOC(a, 1)`

`g = [2, 1, 2]`