

# FORTRAN

## 陣列 (Arrays)

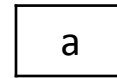
在同一個變數名稱下，儲存多個數值

- 請複製下面幾個檔案到你的work目錄下：  
/home/teachers/weitingc/lecture\_ex/ave1.f95  
/home/teachers/weitingc/lecture\_ex/ave2.f95  
/home/teachers/weitingc/lecture\_ex/ave3.f95

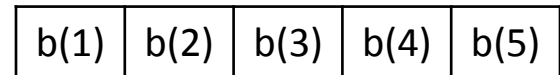
# 變數陣列 Array – 基本介紹

- 之前教過的變數宣告方式，一個變數只能儲存一個數值（如 `REAL :: Time=4`）
- 變數陣列（**Array**）：在同一個變數名稱下，儲存多個數值（類型必須相同）。宣告時，用**DIMENSION**設定陣列的大小。比較下面兩個宣告：

`REAL :: a`



`REAL, DIMENSION (5) :: b`



（**a**是單一實數變數，只能儲存一個數值；**b**是實數陣列，可同時儲存5個不同的實數數值）

# 陣列的subscript

- **REAL, DIMENSION (5) :: b**

b(1)	b(2)	b(3)	b(4)	b(5)
------	------	------	------	------

- 儲存在同個陣列下的個別數值，稱為「元素（element）」。  
（如：**b**陣列有五個元素，每個都可以儲存一個實數數值）
- 元素在該陣列中的位置，稱為「subscript」（如：**b(1)**代表陣列中第一個元素，**b(5)**是第五個元素）
- 程式中，如果直接寫 **b**，指的是整個陣列，如果是 **b(1)**，則特別指**b**陣列的第一個元素。
- Subscript一定是整數 **XXX b (1.5) XXX**

# 一次就對陣列中所有元素賦值的方法

- 方法一：在宣告時直接填入

```
REAL, DIMENSION(5) :: a = (/1., 2., 3., 4., 5./)
```

```
REAL, DIMENSION(10) :: b = 0.
```

- 方法二：在執行區用“=”賦值

```
REAL, DIMENSION(5) :: a
```

```
REAL, DIMENSION(10) :: b
```

```
a = (/1., 2., 3., 4., 5./)
```

```
b= 0.
```

# 對陣列中個別元素賦值的方法

- 方法一：用=，並用subscript選定元素，填入數值

```
REAL, DIMENSION(5) :: a
```

```
a(1) = 1.
```

```
a(2) = 2.
```

```
...
```

- 方法二：用=，並利用Do迴圈的counter做subscript

```
REAL, DIMENSION(10) :: b
```

```
INTEGER :: i
```

```
DO i=1,10
```

```
    b(i) = ...
```

```
END DO
```

# 用READ對陣列賦值

```
REAL, DIMENSION(5) :: a
```

```
INTEGER :: i
```

```
OPEN (10, FILE='input.txt', FORM='FORMATTED',  
& STATUS='OLD')
```

```
DO i=1,5
```

```
    READ (10,*) a(i)
```

每一次迴圈，會從input.txt讀一行檔案

```
END DO
```

- （用write輸出陣列也是使用類似的語法）

# 範例檔 ave1.f95

- 接收使用者從鍵盤輸入的十個數字，對十個變數賦值，計算其平均，將十個數字與平均值結果輸出在螢幕上

```
PROGRAM AVE1  
IMPLICIT NONE
```

ave1.f95

```
REAL :: x1, x2, x3, x4, x5, x6, x7, x8, x9, x10
```

```
REAL :: average=0., total=0.
```

```
INTEGER :: i
```

```
WRITE(*,*) 'enter 10 numbers'
```

```
READ(*,*) x1, x2, x3, x4, x5, x6, x7, x8, x9, x10
```

```
total= x1+x2+x3+x4+x5+x6+x7+x8+x9+x10
```

```
average=total/10
```

```
WRITE(*,*) 'the average of', x1, x2, x3, x4, x5, x6, x7, x8, x9, x10
```

```
WRITE(*,*) '=', average
```

```
END PROGRAM AVE1
```

# 範例檔 ave2.f95

- 使用一個長度有十個元素的陣列x，接收使用者從鍵盤輸入的十個數字，對陣列各元素賦值，加總平均，將十個數字與平均值結果輸出

```
PROGRAM AVE2
IMPLICIT NONE

REAL, DIMENSION(10) :: x=0.
REAL :: average=0., total=0.
INTEGER :: i

WRITE(*,*) 'enter 10 numbers'
DO i=1,10
  READ(*,*) x(i)
  total=total+x(i)
ENDDO
average=total/10

WRITE(*,*) 'the average of', x
WRITE(*,*) '=', average

END PROGRAM AVE2
```

ave2.f95



# 陣列—規則與秘訣 (1)

- 超出宣告所設定的subscript範圍，會發生什麼事？
  - 程式會在執行時，去存取記憶體的另一個位置

```
REAL, DIMENSION (5) :: a=(/1,2,3,4,5/)
```

```
REAL, DIMENSION (5) :: b=(/10,20,30,40,50/)
```

```
...
```

```
WRITE (*,*) a(6) → 10.0
```

(記憶體儲存 a 陣列資料之後，會緊接著儲存 b 陣列的資料，所以程式若企圖讀寫 a(6)，就會存取到 b(1) 的數值)

# 範例檔 ave3.f95

- 與ave2.f95類似，但只要修改整數常數iMAX的值，就可調整程式要計算幾個數字的平均（用iMAX設定陣列大小與迴圈次數）

```
PROGRAM AVE3
IMPLICIT NONE
INTEGER, PARAMETER :: iMAX=5
REAL, DIMENSION (iMAX) :: x=0.
REAL :: average=0., total=0.
INTEGER :: i
```

ave3.f95

→

用iMAX的數值控制要處理幾個數字

```
WRITE(*,*) 'enter' , iMAX, ' numbers '
DO i=1, iMAX
  READ(*,*) x(i)
  total=total+x(i)
ENDDO
average=total/iMAX

WRITE(*,*) 'the average of', x
WRITE(*,*) '=', average

END PROGRAM AVE3
```

## 陣列—規則與秘訣 (2)

- 為了方便日後修改程式，建議在宣告陣列變數的之前，先宣告一個常數來設定陣列的大小，舉例說明：

```
INTEGER, PARAMETER :: MAX_SIZE = 5
```

```
REAL, DIMENSION(MAX_SIZE) :: a1
```

```
REAL, DIMENSION(2*MAX_SIZE) :: a2
```

- 這樣以後只要更改常數MAX\_SIZE的數值，就可改變陣列大小。
- 後續如果需要用到迴圈對陣列元素賦值，也可用常數MAX\_SIZE設定迴圈執行次數
- 如果程式中有多個陣列變數的大小相同（或相關），或是陣列的大小會隨程式的功能所需而更動，就特別適合使用常數宣告陣列大小的方法