

# Fortran

## Chapter 3 輸出入及宣告

### 3-1 WRITE, PRINT

Example:

```
program main
write(*,*) "hello, world!"
stop
end program main
```

程式架構：

```
程式開始→      program main      ←自訂的名稱
                  {
主程式碼→      .....
                  .....
                  .....
程式結束→      stop                ←這一行可省略
主程式碼結束→  end
                  end program
                  end program main
```

```
      ↓ 輸出的位置使用內定值(也就是螢幕)
write (*,*) "Hello, World"
      ↑ 不特別設定輸出的格式
```

完整寫法：

```
write (unit = *, FMT = *) "Hello, World"
```

unit = \* 與 unit = 6 相同，皆輸出至螢幕

上一行程式亦可改成

```
print *, "Hello, World"
```

### 3-2 宣告(Declaration)

在程式當中，程式設計師要問電腦的作業系統，要求電腦的記憶體中，預留一個存放程式進行所需要資料的空間。

Example :

```
Program ex0405
implicit none
integer :: A
real :: B
character(len=1) :: C
logical :: D
A = 1
B = 1.0
C = 'c'
D = .TRUE.
write(*,*) "A=",A, "B=",B, "C=",C, "D=",D
stop
end program ex0405
```

bit：最小儲存單位      一個 bit 只能儲存一個 0 或是 1

1 byte = 8 bit：位元組      通常電腦資料是以位元組來做為最小的儲存單位

#### (a) integer 整數

##### (i) integer(kind=2)

需 16 bits (2 bytes) 儲存，整數介於±32767 之間  
(i.e.  $2^{15} - 1$ 及  $-2^{15} + 1$  之間)

##### (ii) integer(kind=4)

需 32 bits (4 bytes) 儲存，整數介於±2147483647 之間  
(i.e.  $2^{31} - 1$ 及  $-2^{31} + 1$  之間)

#### (b) real 浮點數

##### (i) real(kind=4)單精確度

佔 32 bits 的長度，有效位數為 6~8 位  
max. value =  $3.4 \times 10^{38}$  min. value =  $1.18 \times 10^{-38}$

##### (ii) real(kind=8)單精確度

佔 64 bits 的長度，有效位數為 15~61 位  
max. value =  $1.79 \times 10^{308}$  min. value =  $2.23 \times 10^{-308}$

(c) complex a+bi 表示之；a，b 為浮點數

(d) character

(e) logical "True" and "False"

(1) 整數型態：

`integer :: A` 宣告一個叫做 A 的整數變數

Example：

Program ex0406

`implicit none`

`integer :: A`

`A = 2+2*4-3`

`write(*,*) "2+2*4-3=",A`

`stop`

`end program ex0406`

以  $2+2*4-3$  計算結果來設定變數 A 的數值

Note:

$A = 3/2 \rightarrow A = 1$

$A = 1/2 \rightarrow A = 0$

} 小數點的部分無條件忽略

`integer :: A,B,C` → A,B,C 皆為整數變數

`integer (kind = 2) :: a` → 使用兩個位元組來記錄一個整數

`integer (kind = 4) :: b` → 使用四個位元組來記錄一個整數

省略(kind)敘述時，會以(kind = 4)為 integer 型態的內定值

## (2) real (浮點數)

real :: a                      a 為單精確度浮點數

real (kind = 4) :: a           a 為單精確度浮點數

real (kind = 8) :: a           a 為雙精確度浮點數

電腦在儲存浮點數時，都會先把它轉成以指數來表示的科學符號型式。

e.g.

12345        →       $0.123450 \times 10^5$

12345678    →       $0.123457 \times 10^8$

小數點以後的部分可以儲存 6 位小數

Example:

```
Program ex0409
```

```
implicit none
```

```
real :: a, b
```

```
a = 100000.0
```

```
b = 0.0001
```

```
write (*,*) a, "+", b, "=", a + b
```

```
stop
```

```
end program ex0409
```

### (3) complex

complex :: a

complex (kind = 4) :: a

complex (kind = 8) :: a

$A = (x,y)$ ，其中  $x$  為實部， $y$  為虛部

e.g.  $A = (3.2, 2.5)$  表示  $A = 3.2 + 2.5i$

### (4) character

character (len = 1) :: a

character (1) :: a

character\*1 :: a

character (len = 80) :: a

character (80) :: a

character\*80 :: a

}

a 為一個字元變數

}

a 為一個字元變數，字串長度為 80  
個字元

`a = "hello, world "` or `a = 'hello world '`

### (5) logical

logical :: A

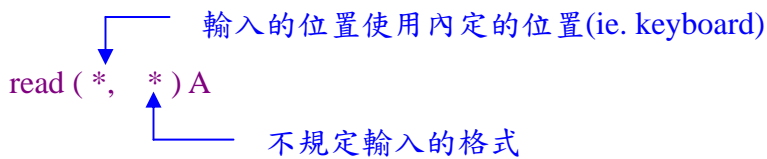
`A = .TRUE.` 真值

`A = .FALSE.` 假值

### 3-3 Read 輸入敘述

Example

```
Program ex0418
integer :: A
real :: B
complex :: C
character (len = 40) :: D
logical :: E
write (*,*) "Please input a (integer) number:"
read (*,*) A
write (*,*) "please input a (real) number:"
read (*,*) B
write (*,*) A, B
stop
end program ex0418
```



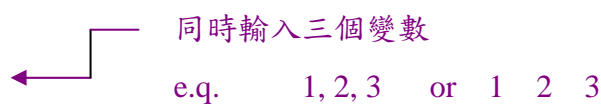
嚴謹使用法：

```
read (unit = 5, FMT = *) A
```

unit = \* 與 unit = 5 相同，皆指由 keyboard 輸入

Example

```
program ex0419
integer :: A,B,C
read (*,*) A, B, C
write (*,*) A, B, C
stop
end program ex0419
```



## 格式化輸出 (Format):

### Example

```
program ex0420
integer :: A
A = 100
write(*, 100) A
format( I4 )
stop
end program ex0420
```

100 }  $\equiv$  write(\*,'(I4)') A

↑ 格式控制敘述

### 重要格式控制敘述：

**Aw** : 以 w 個字元寬來輸出字串

**Dw.d** : 以 w 個字元的寬度來輸出浮點數，小數部分佔 d 個字元寬

**Ew.d** : 以 w 個字元的寬度來輸出指數型態的浮點數，小數部分佔 d 個字元寬

**nX** : 把輸出的位置向下跳過 n 個位置

e.g.

```
write(*,'(1X,A10)') 'hello'
```

→ □□□□□hello

```
write(*,'(E(15.7)') 123.45
```

→ □□0.1234500E+03

$w \geq d+7$

7 位

15 位

**Fw.d** : 以 w 個字元的寬度來輸出浮點數，小數部分佔 d 個字元寬

```
write(*,'(1X,F9.3)') 123.45
```

→ □□123.450

**Iw** : 以 w 個字元的寬度來輸出整數

```
write(*,'(1X,I5)') 100
```

→ □□100

**Lw** : 以 w 個字元的寬度來輸出 T 或 F 的真假值

```
write(*,'(1X,L5)') .TRUE.
```

→ □□□□T

### Notes:

(1) write(\*,'(1X,I3)') 10000

→ \*\*\*

10000 需 5 個欄位，但只設定 3 個欄位不足，會輸出\*





### 3-4 宣告的其它補述

#### (1) implicit

第一個字母為 I,J,K,L,M,N 的變數內定成整數型態，其它的變數則被當成浮點數來使用  
易發生人為錯誤 →

##### implicit integer (A-F, I, K)

把 A 到 F 開頭及 I, K 開頭的變數都當成整數

##### implicit real (F-K)

把 F 到 K 開頭的變數都當成浮點數

##### implicit none

把”內定型態”的功能關閉

#### (2) parameter

Example

```
program ex0428
implicit none
real, parameter :: pi = 3.14159
write(*, '(1X,A10,F5.2)') 'sin(pi/6)=', sin(pi/6.0)
stop
end program ex0428
```

real, parameter :: pi = 3.14159

↑ 不能省略

pi 為浮點常數，其值為 3.14159

pi 不是變數，所以不佔記憶體(直接放在暫存器)，程式執行中不能改變其值

#### (3) 設定變數的初值

Example

```
Program ex0430
implicit none
integer :: a = 1
real :: b = 1.0
complex :: c = (1.0, 1.0)
character(len=1) :: d = 'A'
logical :: e = .TRUE.
write (*, '(I3,TR1,F4.2,TR1,(F4.2,TR1,F4.2),A2,L3)') a,b,c,d,e
stop
end program ex0430
```

(4) 宣告在程式中的結構

```
program main
```

```
implicit none ← implicit 要放在宣告的最前端
```

```
integer :: A } ← 開始變數宣告  
real :: B
```

```
..... ← 主程式敘述區
```

```
stop ← 程式結束
```

```
end program main
```

Example:

```
Program ex0431
```

```
implicit none
```

```
integer :: A = 2, B = 1                   把整數型態轉換成浮點數型態
```

```
real :: C
```

```
C = B / A                   → C = real(B) / real(A)
```

```
write(*,*) C               → 0.00000
```

```
stop
```

```
end program ex0431
```

```
A + int(B) / C ← A, C 為 integer, B 為 real 時
```

```
↑  
把 real 轉換成 integer
```