

程式與科學計算

– Basic Elements of FORTRAN

- What does a FORTRAN program look like (basic structure)
- How to use a FORTRAN program (compile, execute)
- Variable types
- Simple calculation, intrinsic functions
- Simple input (from keyboard)/output (to screen)

預習問題—準備工作

- 請登入study主機，在你的work目錄下新增子目錄
fortran
- 複製/home/teachers/weitingc/lecture_ex/first.f95到你的work/fortran/下
- 將first.f95更改名稱為product.f95

Basic Structure of a FORTRAN PROGRAM

product.f95

```
PROGRAM product
IMPLICIT NONE
! This is our first example of a FORTRAN program.
! It asks the user to input two numbers, then the product of
! the two numbers is calculated, and output on the screen.
! (weitingc, 2012/09/04)

! Declare the variables used in this program
REAL :: i, j, k ! i, j are input numbers, k is result

! Start read in i and j
WRITE(*,*) 'Please input two numbers:'
READ(*,*) i, j

! Calculate the product (k) of i and j, and output on screen
k = i*j
WRITE(*,*) 'The product of these two numbers is:', k

END PROGRAM product
```

Declaration
Section
宣告區

Execution
Section
執行區

Termination
Section
終止區

每一支Fortran主程式都必須具備 宣告區、執行區、終止區 三個段落

Basic Structure of a FORTRAN PROGRAM

- 宣告區：

PROGRAM代表程式的第一行，宣告程式的名稱，之後宣告變數種類（real, integer...）

Declaration
Section
宣告區

```
PROGRAM product ← 程式的名稱是 product
IMPLICIT NONE
! This is our first example of a FORTRAN program.
! It asks the user to input two numbers, then the product of
! the two numbers is calculated, and output on the screen.
! (weitingc, 2012/09/04)
! (edited 2013/09/16)
! Declare the variables used in this program
REAL :: i, j, k ! i, j are input numbers, k is result
          ↑
          宣告三個實數變數 i, j, k
```

product.f95

Basic Structure of a FORTRAN PROGRAM

- **執行區：**（必須在所有變數宣告完成後）
執行程式主要的功能，例如：
設定變數數值、代數運算（加減乘除）、
迴圈（重複執行某些段落）、邏輯判斷（選擇執行某些段落）、輸入（read）、輸出（write）等

Execution
Section
執行區

product.f95

```
! Start read in i and j  
WRITE(*,*) 'Please input two numbers:'  
READ(*,*) i, j  
!  
! Calculate the product (k) of i and j, and output on screen  
k = i*j  
WRITE(*,*) 'The product of these two numbers is:', k
```

輸出（螢幕顯示）文字與變數的數值

Basic Structure of a FORTRAN PROGRAM

- 終止區：
END標示程式的最後一行，通常會加上與程式名稱相符的標籤



Basic Structure of a FORTRAN PROGRAM

product.f95

Comments
註解
(!之後的
文字)

```
PROGRAM product
IMPLICIT NONE
! This is our first example of a FORTRAN program.
! It asks the user to input two numbers, then the product of
! the two numbers is calculated, and output on the screen.
! (weitingc, 2012/09/04)
! (edited 2013/09/16)
! Declare the variables used in this program
REAL :: i, j, k ! i, j are input numbers, k is result

! Start read in i and j
WRITE(*,*) 'Please input two numbers:'
READ(*,*) i, j

! Calculate the product (k) of i and j, and output on screen
k = i*j
WRITE(*,*) 'The product of these two numbers is:', k

END PROGRAM product
```

Comments 註解的重要

- 每一行 **!** 符號之後的文字，`compiler`解讀為註解，不會當作程式編碼處理
- 寫註解的目的：
 - 在寫程式的時候先掌握基本架構（程式的草稿）
 - 讓寫程式的人可以對程式內容進行口語化的說明、提醒，幫助自己或他人日後使用、修改時方便參考、回憶
- 可以考慮寫成註解的事項
 - 在程式開頭，解釋整個程式的用處、基本架構、版本、更新日期
 - 在變數宣告區，列出各變數代表的意義，與相對應的單位
 - 在執行區，解釋每一段程式要執行的功能、計算相對應的原始方程式、迴圈或邏輯判斷的文字解釋等等
 - 其他對自己或程式讀者重要的提醒

How to run a FORTRAN program

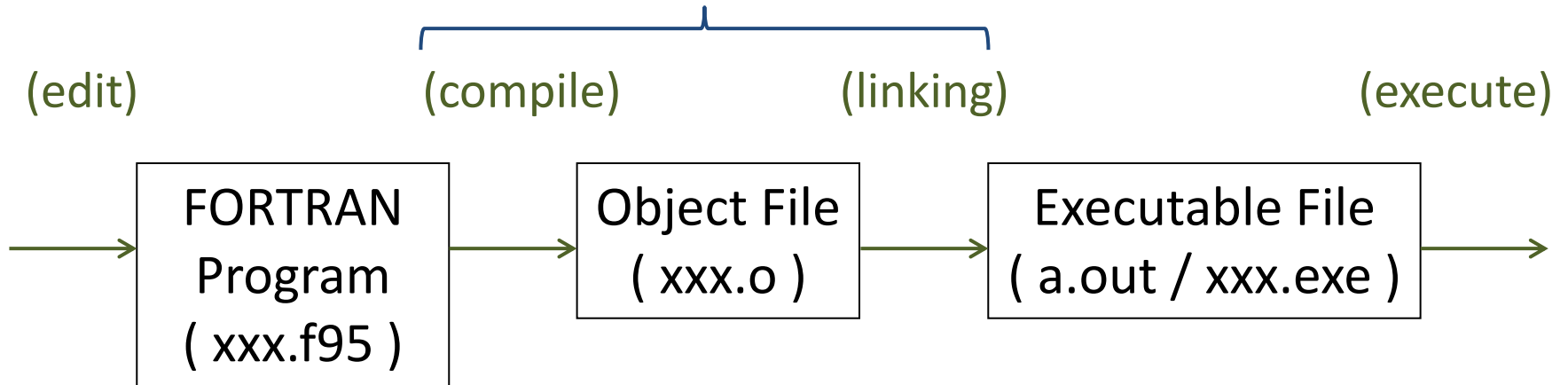
- 寫好的FORTRAN程式（通常取名為 ***.f95**）本質上仍然是個文字檔，可以用nano等文字編輯軟體開啟，進行修改。
- 文字檔沒有辦法直接「執行」，必須經過「編譯」（**compile**）的步驟，將程式碼轉換成「執行檔」，使用執行檔才能指揮電腦進行運算

（可以把.f95程式碼想成把指令寫在word檔裡，執行檔想成是手機上的APP，編譯的過程是把word檔變成APP的icon）

- 執行檔通常取名為***.exe**，但編譯器預設產生的執行檔檔名是**a.out**，為了避免混淆、檔案被覆蓋，建議編譯時要自行指定執行檔的檔名。

How to compile and run a FORTRAN program?

FORTRAN Compiler (e.g., f95)



編譯product.f95程式，產生的執行檔是預設名稱 **a.out**
執行a.out這個執行檔

Example 1: `>f95 product.f95`
`>./a.out`

Example 2: `>f95 product.f95 -o product.exe`
`>./product.exe` 用 **-o** (小寫) 自訂執行檔名稱

Example 3: `>f95 product.f95`
`>mv a.out product.exe` 用 **mv** 更改執行檔名稱
`>./product.exe` (建議使用)

預習問題—第一部分

請編譯（compile）product.f95程式，用ls看會出現什麼新檔案

```
> f95 product.f95
```

請執行編譯好的執行檔（不要省略”.”）

```
> ./a.out
```

```
weitingc@study1:~/work
[weitingc@study1 work]$ ls
product.f95
[weitingc@study1 work]$ f95 product.f95
[weitingc@study1 work]$ ls
a.out product.f95
[weitingc@study1 work]$ ./a.out
Please input two numbers:
3.1
2.2
The product of these two numbers is: 6.820000
[weitingc@study1 work]$
```

預習問題一第二部分

請編譯product.f95，在指令中用-o讓產生的執行檔為product.exe

```
> f95 product.f95 -o product.exe
```

請執行編譯好的執行檔，應該要與前面執行a.out相同：

```
> ./product.exe
```

每年都發生的悲劇...（尤其在期中考）

- 悲劇一：編譯指令錯誤，將程式碼(`xxx.f95`)覆蓋（無法挽救，只能重寫程式！）

```
>f95 product.f95 -o product.f95
```

`product.f95`變成執行檔，無法再做任何編輯，也看不到程式碼！



- 悲劇二：修改程式(`xxx.f95`)後忘記編譯，反覆執行舊版的執行檔(`a.out`或`xxx.exe`)，鬼打牆debug...



宣告變數

product.f95

加上這行
就會關閉
預設變數
類型的功
能

→

```
PROGRAM product
```

```
IMPLICIT NONE
```

```
! This is our first example of a FORTRAN program.
```

```
! It asks the user to input two numbers, then the product of  
! the two numbers is calculated, and output on the screen.
```

```
! (weitingc, 2012/09/04)
```

```
! (edited 2013/09/16)
```

```
! Declare the variables used in this program
```

→

```
REAL :: i, j, k ! i, j are input numbers, k is result
```

```
! Start read in i and j
```

```
WRITE(*,*) 'Please input two numbers:'
```

```
READ(*,*) i, j
```

```
! Calculate the product (k) of i and j, and output on screen
```

```
k = i*j
```

```
WRITE(*,*) 'The product of these two numbers is:', k
```

```
END PROGRAM product
```

宣告三個
實數(real)
變數:i, j,
k

Fortran Constant and Variable types

- 可宣告的類型（該變數/常數可被賦予「數值」類別）
 - 整數 Integer（沒有小數點）
 - 實數 Real（必有小數點）
 - 字串 Character（文字，用單引號' '或雙引號" "夾起）
 - 其他：邏輯 Logical, 複數 Complex, 自訂種類 Derived data type
- Variable（變數） vs. Constant（常數）
 - 常數與變數均可宣告為整數、實數、字串
 - 變數：在程式執行過程中，其值可能會因運算改變
 - 常數：在程式執行過程中，其值不可改變
- 取名規則：變數名字最多**31**個字元（只能是字母、數字、_的組合，首字元必須是字母）

Air_RH_700hPa	O	700hPa_Air_RH	X
Air-RH_700hPa	X	Air_RH@700hPa	X

Declaration of Variables and Constants

- 宣告變數類型

INTEGER :: a

REAL :: b, c, d

CHARACTER(len=3) :: name1

CHARACTER(5) :: name2

宣告字串變數時，需設定變數要儲存的長度有幾個字元（len=?），與變數名稱長度無關

- 宣告常數類型

INTEGER, PARAMETER :: k=3

REAL, PARAMETER :: PI=3.1416, Q=1.2E+6

（常數在宣告時就要「賦值」，其值之後不可更動）

Default Typing and IMPLICIT NONE

- 預設變數類型：如果一個變數沒有被宣告，**FORTRAN**會依照變數名稱的第一個字母，預設此變數的類型，規則如下
 - 開頭是 i, j, k, L, M, N（大小寫都包括）：預設為整數 Integer
 - 其他開頭：預設為實數 REAL
 - 預設類型不包括字串、邏輯、複數變數
- 在宣告變數前加上 **IMPLICIT NONE**，可關閉預設變數類型的功能：**compiler**如果檢查到未被宣告的變數，會出現錯誤訊息提醒。

預習問題—第三部分

用nano或其他編輯器瀏覽product.f95。

(1) 請問變數i是什麼類型？你是怎麼判斷的？

```
PROGRAM product
→ IMPLICIT NONE
! This is our first example of a FORTRAN program.
! It asks the user to input two numbers, then the product of
! the two numbers is calculated, and output on the screen.
! (weitingc, 2012/09/04)
! (edited 2013/09/16)
! Declare the variables used in this program
→ REAL :: i, j, k ! i, j are input numbers, k is result
...
```

變數 i 被宣告成實數 (real)

(如果移除程式中IMPLICIT NONE與REAL :: i, j, k 這兩行，會發生什麼事？)

設定變數數值（賦值）、數學運算

product.f95

```
PROGRAM product
IMPLICIT NONE
! This is our first example of a FORTRAN program.
! It asks the user to input two numbers, then the product of
! the two numbers is calculated, and output on the screen.
! (weitingc, 2012/09/04)
! (edited 2013/09/16)
! Declare the variables used in this program
REAL :: i, j, k ! i, j are input numbers, k is result

! Start read in i and j
WRITE(*,*) 'Please input two numbers:'
READ(*,*) i, j

! Calculate the product (k) of i and j, and output on screen
k = i*j
WRITE(*,*) 'The product of these two numbers is:', k

END PROGRAM product
```

從鍵盤接收
數值，為變
數 i, j 賦值 →

對變數 k 賦
值，其值是
變數 i 與 j 的
乘積

Assign a value to a variable (賦值)

(在fortran程式中，用等號(=)來賦予變數數值或字串內容，例如 `C=-100.` 這行程式，口語化的意思就是「令實數變數C為-100.」)

`INTEGER :: a=10` 整數變數，不含小數點

`REAL :: b, c, d`

`CHARACTER(len=3) :: name1`

`CHARACTER(5) :: name2`

`b = +1.234` → 實數變數，含一個小數點，+號可省略

`C = -100.` → 整實數，小數點後的0可省略

`d = 1.23E+4` → 實數變數可用科學記號表示 (1.23×10^4)

`name1 = 'PTT'` → 字串變數由一對單引號或雙引號夾起

`name2 = "' @ $! '"` → 字串變數內容可帶有空白、符號

對變數與常數賦值

- 對變數賦值（可在宣告區或執行區）

INTEGER :: a = 10

REAL :: b, c, d

...

b = 1.234

- 對常數賦值（只能在宣告區）

INTEGER, PARAMETER :: k=3

REAL, PARAMETER :: PI=3.1416, Q=1.2E+6

（常數在宣告之後，其值不可更動）

賦值：常犯的錯誤或潛在問題

INTEGER :: a

REAL :: b, c, d

CHARACTER(len=3) :: name1

CHARACTER(5) :: name2

△ **a = 10.23**

→ 整數變數，小數點後自動無條件捨去（程式仍可執行，但可能造成計算錯誤！）

× **b = +1.234.**

→ 實數變數只可含一個小數點

△ **c = 1/3**

→ 實數數值用整數運算取得，會有精確度問題

× **d = 1.23E+4.**

→ 實數變數若用科學記號表示，指數需是整數

× **name1 = 'PTT'**

→ 字串變數需由一對單引號或雙引號夾起

× **name2 = "' @ \$! ' "**

→ 字串變數需由一對單引號或雙引號夾起

常數、變數類型宣告 – 秘訣

- 永遠使用 **IMPLICIT NONE**，避免使用預設變數類型（i.e. 所有的變數都要宣告），以免造成錯誤混淆
- 取名字盡量符合變數代表的意義，方便瞭解（例如：`year`, `Tair`, `distance`, `name`...）
- 自己決定變數名稱大小寫的使用習慣（例如常數一律首字母大寫，整數一律全大寫...等），確實遵守
- 在宣告區，利用註解說明各變數、常數代表的意義、單位、功用，例如：

```
REAL :: Tair
```

```
! Tair = air temperature from input file (K)
```

Simple Arithmetic Calculation (代數運算)

INTEGER :: a

REAL :: b, c, d

a = 10 → 等號右邊可以是數字

b = a+1.234 → 等號右邊也可以是代數運算式

c = (a-4.)*b

d = (a**b)/c

Write(*,*) a, b, c, d

基本運算符號

+	加
-	減
*	乘
/	除
**	次方

輸出結果：10 11.23400 67.40400 2.5428137E+09

Simple Arithmetic Calculation – 運算規則

- 等號左邊只能有一個變數

$$A = 3+B \quad \text{O}$$

$$A-B = 3 \quad \text{X}$$

- 等號右邊可以是數字或代數式，但運算符號（+，-，*，/，**）不可連用，要適當用括號分開

$$A = 3*(-B) \quad \text{O}$$

$$A = 3*-B \quad \text{X}$$

$$x = u*(y+z) \quad \text{O}$$

$$x = u(y+z) \quad \text{X}$$

- 負數取次方時，次方部份必須為整數

$$q = (-3.0)**2 \quad \text{O}$$

$$q = (-3.0)**(2.0) \quad \text{X}$$

（理由：電腦會用 $e^{x \ln y}$ 來計算 y^x ，而負數的自然對數無法定義）

（提醒：等號（=）是用來賦予變數數值，例如 $A=3+B$ 這行程式，口語化的意思就是「令變數A為3加上變數B」）

Simple Arithmetic Calculation – 運算規則

- 運算符號的優先順序
 - () 裡的優先
 - **先算（由右到左），再來是*, /, +, -（由左到右）
 - 為避免錯誤、混淆，適當使用 ()，或是 & 換行

$$\begin{aligned}x &= 3.*(2. + 5.)*4. + 10./2.**3 \\ &= 3.*7.*4. + 10./2.**3 \\ &= 3.*7.*4. + 10./8. \\ &= 84. + 1.25 \\ &= 85.25\end{aligned}$$

字串變數的長度

```
CHARACTER(5) :: name='R='
```

```
REAL :: x=8.314
```

- 如果字串變數被賦值的內容比宣告時的長度短，不足的部分會自動補上空格，

```
WRITE(*,*) name,x
```

輸出結果 R=□□□8.314

- 可以利用內建函數TRIM() 去除字串尾端的空格：

```
WRITE(*,*) TRIM(name),x
```

輸出結果 R=8.314

字串的连接

- 用雙斜線 **//** 可以連接字串
- 可作用於字串變數，也可以用於字串內容
- 連接後的結果可直接輸出，也可以用來對新的字串賦值

```
CHARACTER (5) :: name1='abc' // 'de'
```

```
CHARACTER (5) :: name2='ghijk'
```

```
WRITE (*,*) '123' // '45'
```

```
WRITE (*,*) name1 // name2
```

```
WRITE (*,*) name1 // 'gh'
```

輸出結果

```
12345  
abcdeghijk  
abcdegh
```

Intrinsic Functions (內建函數)

- Fortran內建常用的數學函數（大小寫均可）
 - 平方根：SQRT(x)（三次方根要用 $x^{**}(1./3.)$ 求得）
 - 絕對值：ABS(x)
 - 指數函數：EXP(x)
 - 對數函數：自然對數 LOG(x), 十為底的對數 LOG10(x)
 - 三角函數：SIN(x), COS(x), TAN(x)（x的單位是radian）
 - 反三角函數（ $\sin^{-1}(x)$ 等）：ASIN(x), ACOS(x), ATAN(x)
 - 餘數：MOD(x,y)
 - 較大值、較小值：MAX(x,y), MIN(x,y)
- Fortran沒有內建圓周率常數pi，要自己使用三角函數計算得出，如 $pi = 4.*ATAN(1.)$ 。

Intrinsic Functions

`a = SQRT(10)`

→ 函數括號內可以是數字

`G = LOG10(b)`

→ 函數括號內可以是變數

`Temp = SIN(x*y)`

→ 函數括號內可以是運算式

`K = EXP(X)**MAX(x,y)`

→ 函數之間也可以運算

`Log_a = LOG(ABS(x))`

→ 函數括號內可以是另一個函數
(注意括號的數目與位置)

XXX `COS(X)=10.XXX`

→ 函數不可以出現在等號左邊！！

從鍵盤輸入變數數值、由螢幕輸出變數數值

product.f95

```
PROGRAM product
IMPLICIT NONE
! This is our first example of a FORTRAN program.
! It asks the user to input two numbers, then the product of
! the two numbers is calculated, and output on the screen.
! (weitingc, 2012/09/04)

! Declare the variables used in this program
REAL :: i, j, k ! i, j are input numbers, k is result

! Start read in i and j
WRITE(*,*) 'Please input two numbers:'
READ(*,*) i, j

! Calculate the product (k) of i and j, and output on screen
k = i*j
WRITE(*,*) 'The product of these two numbers is:', k

END PROGRAM product
```

螢幕輸出文字

→ WRITE(*,*) 'Please input two numbers:'

從鍵盤接

→ READ(*,*) i, j

收變數i, j

的數值

! Calculate the product (k) of i and j, and output on screen

k = i*j

螢幕輸出

→ WRITE(*,*) 'The product of these two numbers is:', k

文字與變

數k的數值

Simple Input/Output (List-directed I/O)

- READ(*,*) 讀入鍵盤輸入的數值或字串內容
 - 依序讀入變數，讀取格式由compiler按照變數類型設定（之後會教如何在程式裡設定想要讀取的格式）

READ (*, *) a, b

“Unit”

設定讀取/輸出的檔案

*代表使用預設

（自鍵盤讀取，從螢幕輸出）

“format”

設定讀取/輸出的格式

*代表使用預設

（依照讀寫的變數類型）

Simple Input/Output (List-directed I/O)

- **READ(*,*)** 讀入鍵盤輸入的數值或字串內容
 - 輸入時可以一個變數一行（按一次enter），或數個變數一行（中間用空格或逗號隔開）。輸入的字串如果含有空格要用“ ”夾起。
 - 輸入的值要符合變數類型（若變數宣告為整數，輸入實數或字串會產生執行錯誤；若變數宣告為實數，輸入整數可以被接受，會自動補上小數點）

– 範例一

```
INTEGER :: N  
REAL :: x  
CHARACTER(len=12) :: Text  
READ(* ,*) x, N, Text  
... .
```

鍵盤輸入

1.0, 2, 'Hello, you '
<enter>

or

鍵盤輸入

1.0 <enter>
2, 'Hello, you ' <enter>

(結果會完全相同)

↓
程式讀取結果

x	1.000000
N	2
Text	'Hello, you '

Simple Input/Output (List-directed I/O)

- **READ(*,*)** 讀入鍵盤輸入的數值或字串內容
 - 每一個**READ**，程式會換新的一行 (<enter>) 讀取。
 - 如果**READ**要讀取好幾個變數，輸入的數值會一個一個依序被讀入（無論有沒有換行），直到每個要讀取的變數都被讀入為止
 - 如果輸入的數值太多，超過**READ**需要讀入的變數數目，後面多餘的數值會被略過

– 範例二

```
INTEGER :: N, O, P, S  
READ (*,*) N, O  
READ (*,*) P, S  
... .
```

鍵盤輸入

```
1 <enter>  
2, 3, 4<enter>  
5, 6, 7, 8<enter>
```

程式讀取結果

N	1
O	2
P	5
S	6

Simple Input/Output (List-directed I/O)

- `WRITE(*,*)` 將數值或字串顯示在螢幕上
 - 每一個`WRITE`指令輸出一行，要輸出多個變數時，程式中用逗號隔開（執行時，逗號不會顯示在畫面上）
 - 可輸出變數、算式、函式。輸出格式由`compiler`預設（之後會教如何在程式裡設定想要輸出的格式）
 - 範例

```
INTEGER :: N
REAL :: x
CHARACTER(len=12) :: Text
x=1.
N=2
Text='Bye!'
WRITE(*,*) x, N, x/N, Text
... .
```



螢幕顯示結果 1.000000 2 0.5000000 Bye!

預習問題一第三部分

用nano或其他編輯器瀏覽product.f95檔案。

(2) 請問READ(*,*) i, j這行，會讓程式執行時發生什麼事情？

product.f95

```
PROGRAM product  
IMPLICIT NONE
```

```
...
```

```
! Start read in i and j  
WRITE(*,*) 'Please input two numbers:'
```

→ READ(*,*) i, j 接收鍵盤輸入的兩個數值，為實數變數 i 與 j 賦值

```
! Calculate the product (k) of i and j, and output on screen  
k = i*j
```

→ WRITE(*,*) 'The product of these two numbers is:', k

```
END PROGRAM product
```

(如果將第二個WRITE改成WRITE(*,*) i, 'x', j, '=', k，會發生什麼事?)

簡單輸出指令

- 若是單純螢幕輸出，`write(*,*)` 和 `print*`，（逗號不能省）的結果是相同的。

The Free Format of F90

- 幾個FORTRAN90程式的規則
 - 用 **!** 標記之後的文字為註解
 - 每行最多不可超過**132**個字元
 - 一行的最後若出現 **&** 符號，代表這一行與下一行要連在一起，成為完整的一行程式

F90 Format -- example

一行最多 132 字元 (超過的直接省略)

```
PROGRAM square
```

```
IMPLICIT NONE
```

```
!This program calculates the square of integers
```

```
Integer :: a=2
```

```
WRITE (*,*) '1^2, 2^2,...6^2='
```

```
! Starting
```

```
WRITE (*,*) 1**a,
```

```
&
```

```
2**a,
```

```
&
```

```
3**a,
```

```
&
```

```
4**a,
```

```
&
```

```
5**a,
```

```
&
```

```
6**a
```

```
END PROGRAM square
```

整行都是註解



此行後半為註解

此行與下一行連結

(適當的分行可以讓程式整齊清爽)

Mixed-Mode Arithmetic 混型計算

- 當運算式裡同時有實數和整數，要注意最後的結果是否為你預期的。
 - 整數與整數運算 → 結果為整數，**小數點下無條件捨去**
 $3/4 \rightarrow 0$
 - 實數與實數運算 → 結果為實數 $3./4. \rightarrow 0.75$
 - 整數與實數運算 → 結果為**實數** $3./4 \rightarrow 0.75$
- 問題通常發生在算式裡同時有 $+ - * /$ 的時候，按照運算規則， $* /$ 會先算，再算 $+ -$
 - $1 + 3/4$ (整數 + 整數) $\rightarrow 1 + 0 \rightarrow 1$
 - $1. + 3/4$ (實數 + 整數) $\rightarrow 1. + 0 \rightarrow 1.$
 - $1. + 3./4$ (實數 + 實數) $\rightarrow 1. + 0.75 \rightarrow 1.75$

Mixed-Mode Arithmetic 混型計算

- 為了避免預期外的混型運算結果，可以利用 `INT(X)`, `NINT(X)`, `REAL(I)` 轉換變數類型
 - `INT(X)` : 實數轉換成整數，小數點下無條件捨去
 - `NINT(X)` : 實數轉換成整數，小數點下四捨五入
 - `REAL(I)` : 整數轉換成實數

$$1 + \text{INT}(3./4.) \quad \rightarrow \quad 1 + 0 \quad \rightarrow \quad 1$$

$$1 + \text{NINT}(3./4.) \quad \rightarrow \quad 1 + 1 \quad \rightarrow \quad 2$$

$$1. + \text{REAL}(3)/\text{REAL}(4) \quad \rightarrow \quad 1. + 0.75 \quad \rightarrow \quad 1.75$$

$$1. + \text{REAL}(3/4) \quad \rightarrow \quad 1. + 0. \quad \rightarrow \quad 1.$$

本日小結

- FORTRAN程式基本結構
 - 宣告區、執行區、終止區
- 編譯、執行程式
- 變數類型、宣告方式、數值設定
 - 整數、實數、字串
- 基本代數運算、內建函數與規則
 - $a+b$, $a-b$, $a*b$, a/b , $a**b$
 - $\text{sqrt}(x)$, $\text{abs}(x)$, $\text{cos}(y)$, $\text{exp}(y)$, $\text{log}(z)$, $\text{max}(z)$...
- 基本變數數值讀寫（鍵盤輸入、螢幕輸出）
 - $\text{read}(*,*)$, $\text{write}(*,*)$

本日小結

- 要設定一個變數/常數的值，有幾種方法：

- 變數，可在宣告時賦值

```
INTEGER :: time = 0
```

- 變數，可在程式執行區利用 = 賦值

```
A = 5
```

```
x = a*exp(b)
```

- 變數，可由使用者輸入賦值（在執行區用read）

```
READ(*,*) a, b
```

- 常數：只能在宣告時賦值，在執行區更動常數數值，編譯程式會出現做錯誤訊息

```
REAL, PARAMETER :: PI=3.1416, Q=1.2E+6
```

秘訣：養成宣告並設定每個變數初始數值的習慣（歸零、給定數值、讀入數值...），避免compiler給的「驚喜」...