

FORTRAN

自訂函數 (function)

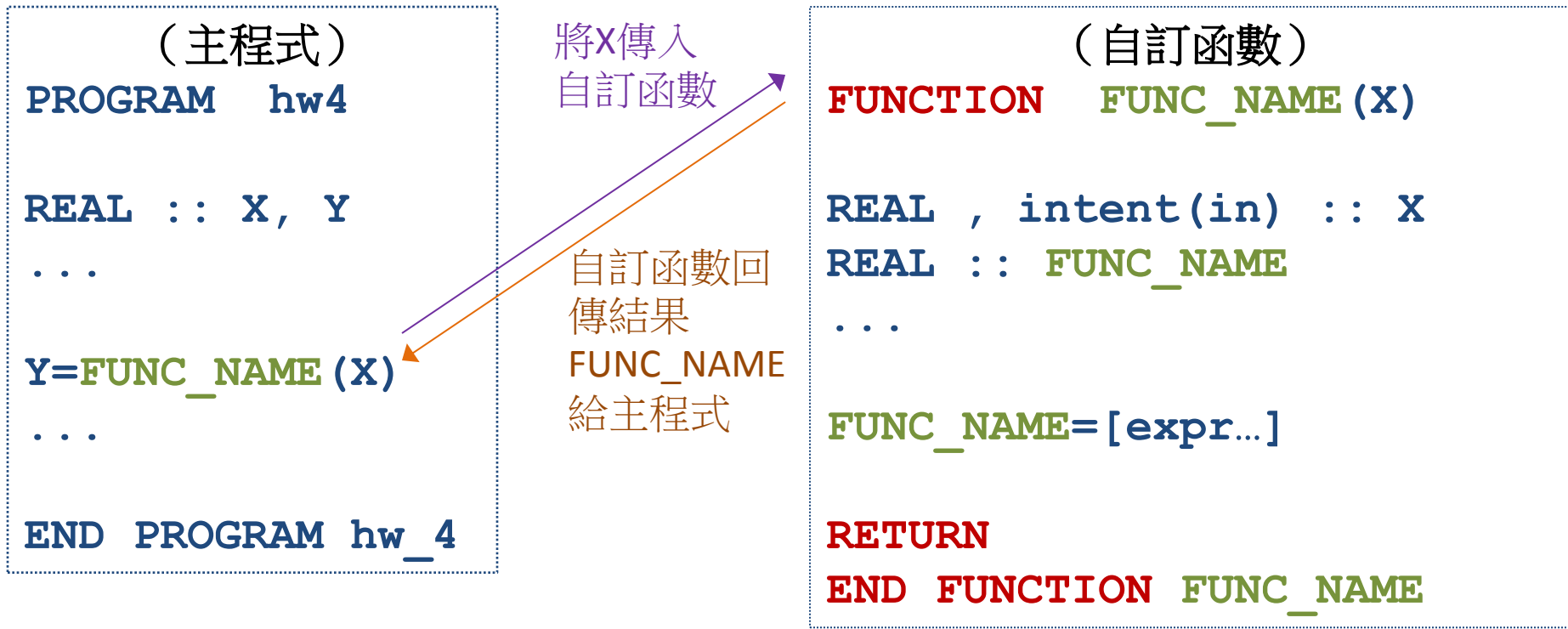
回憶：之前介紹過FORTRAN的內建函數

- Fortran內建常用的數學函數
 - 平方根：SQRT(x)
 - 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)
 - 反三角函數（ $\sin^{-1}(x)$ 等）：ASIN(x), ACOS(x), ATAN(x)
 - ...

想要使用的數學函數FORTRAN沒有內建，怎麼辦？
(例如 $\cot(x)=1/\tan(x)$ ， $\cosh(x)=(\exp(x)+\exp(-x))/2\dots$)
難道每次用到都要自己重新寫程式計算嗎？

自訂函數 FUNCTION

- 一些常用但FORTRAN沒有內建的函數，可以另外寫成自訂函數，這樣在不同的主程式中，都能方便使用，不需要複製相同的程式碼來計算這個函數



自訂函數的範例(1)—cot(x)

舉例: $COT(x)=1/TAN(x)$

```
FUNCTION  COT (x)
REAL  ::  COT
REAL  ,  intent(in)  ::  x

COT = 1./TAN(x)

RETURN
END FUNCTION COT
```

- 這個自訂函數的名稱在第一行，叫COT，從主程式接收自變數x
- 在宣告區宣告一個實數變數COT，與自訂函數名稱完全相同。
- 變數x在宣告時設定為INTENT(in)，表示此變數只供傳入，在自訂函數中不能改變其數值。
- COT變數的值，就是將變數x的值代入想要計算的函數，得到應變數，即 $\cot(x) = 1/\tan(x)$ 。
- 這個函數計算完畢後，就會將變數COT的值以實數傳回主程式

自訂函數的範例(1)—使用cot自訂函數

(主程式，使用自訂函數COT)

```
PROGRAM test_cot
IMPLICIT NONE
REAL :: COT
REAL :: x

WRITE(*,*) 'Enter x:'
READ(*,*) x

WRITE(*,*) COT(x)

END PROGRAM test_cot
```

將x的值傳入
自訂函數COT

算出COT的值
回傳給主程式

(剛才寫好的自訂函數COT)

```
FUNCTION COT(x)
REAL :: COT
REAL, intent(in) :: x

COT = 1./TAN(x)

RETURN
END FUNCTION COT
```

- 在主程式也要宣告一個實數變數COT，與自訂函數名稱完全相同，類型也要與自訂函數內的宣告一致。
- 傳入自訂函數的變數x，在主程式與自訂函數宣告的類型要一致。
- 使用自訂函數的語法，與使用內建函數的語法類似，函數名稱後加上括號，填入自變數

自訂函數本身的語法

```
FUNCTION  FUNC_NAME (arg_list)
(Declaration section)
IMPLICIT NONE
TYPE :: FUNC_NAME
TYPE, INTENT(in) :: arg_list
TYPE :: ...
(Execution section)
...
FUNC_NAME = [expr...]

RETURN
END FUNCTION FUNC_NAME
```

- 函數內部使用到的其他變數，也要進行宣告。可以使用**IMPLICIT NONE**設定
- 執行區進行函數相關的運算，算出的應變數結果，儲存在**FUNC_NAME**變數中，才會正確傳回主程式

- 以**FUNCTION**開頭，以**RETURN**與**END FUNCTION**結尾
- **FUNC_NAME**：函數的名字（自訂），也是函數運算最後的結果，其數值是由傳入的變數運算得來，之後傳回主程式。不可與主程式名稱相同。
- **arg_list**：由主程式傳入的變數（自變數）清單，可以傳入多個，用逗號隔開
- 宣告區要記得**宣告函數本身的類型**
- 傳入的變數在宣告時要加上**INTENT(in)**的設定，表示這些變數只供傳入，在自訂函數中不能改變其數值

自訂函數的範例(2)－預習問題

```
PROGRAM test_quadf
IMPLICIT NONE

REAL :: quadf
REAL :: a, b, c, x

WRITE(*,*) 'For a*x^2+b*x+c'
WRITE(*,*) 'Enter a, b, c, x:'
READ(*,*) a, b, c, x

WRITE(*,100) quadf(x, a, b, c)
100 format(F12.4)

END PROGRAM test_quadf
```

$$quadf = ax^2 + bx + c$$

```
FUNCTION quadf(x,a,b,c)
IMPLICIT NONE
REAL :: quadf
REAL, INTENT(in) :: x,a,b,c
quadf = a*(x**2) + b*x + c
RETURN
END FUNCTION quadf
```

這裡有一個自訂函數，以及一個使用了這個自訂函數的主程式。請問：

1. 自訂函數的名稱是？
2. 主程式傳入自訂函數的變數是？
3. 自訂函數回傳給主程式的結果，屬於什麼類型？（整數、實數或字串？單一變數或陣列？）

使用自訂函數，如何編譯程式碼？(1)

- 方法一：在同一個f95檔案中，把自訂函數的程式碼完整地貼在主程式碼的下方（END PROGRAM之後），再編譯這個f95檔案，即可產生執行檔。（例如將自訂函數貼在main.f95中）

```
>f95 -o main.exe main.f95
```

- 用此方法，主程式與要使用的自訂函數存在一起，方便隨時參照、修改。
- 缺點是，如果另一個主程式要用到相同的自訂函數，就要重新複製貼上到主程式下方（新舊版本容易搞混）

使用自訂函數，如何編譯程式碼？(2)

- 方法二：把自訂函數存成個別的.f95檔案（如func1.f95, func2.f95），與主程式的f95檔案同時編譯，指令如下：

```
>f95 -o main.exe main.f95 func1.f95 func2.f95
```

- 如果在其他主程式也會使用到這些自訂函數，但有可能對自訂函數進行修改，建議用方法二，就不用經常剪貼複製自訂函數的程式碼。
- 缺點是編譯的指令很長（可能需要寫shell script方便編譯、執行）。
- （shell script本課程暫時不會教到，有興趣的同學可以google自學）

使用自訂函數，如何編譯程式碼？(3)

- 方法三：把自訂函數存成個別的.f95檔案，並且先個別編譯（用f95 -c指令）產生.o檔（object file）。之後編輯主程式時，把.o檔連結進去，指令如下：

```
>f95 -c func1.f95
```

```
>f95 -c func2.f95
```

```
>f95 -o main.exe main.f95 func1.o func2.o
```

- 自訂函數如果在其他主程式也會使用，並且不常修改，建議方法三，可減少編譯等待的時間。

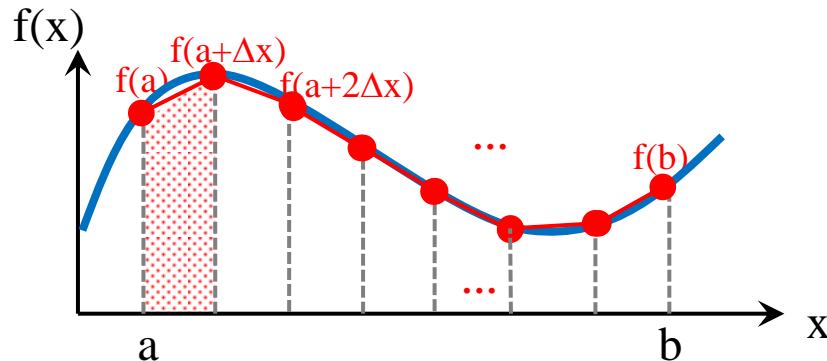
自訂函數FUNCTION：重點提示

- 記得在自訂函數與主程式中宣告自訂函數的種類
- 主程式傳入的變數，在自訂函數中用intent(in)宣告，且在自訂函數中不能更動
- 自訂函數會回傳給主程式的只有單一變數（它自己）
- 仔細檢查傳入自訂函數的變數清單，包括變數個數、種類、順序，在主程式與自訂函數都要一致

自訂函數的範例(3)—將int_x2程式改寫為自訂函數

- 程式範例檔：`/home/teachers/weitingc/lecture_ex/int_x2.f95`
- 程式說明：請參考FORTRAN_ex5_INTX2的講解影片
- 使用者由鍵盤輸入積分區間 $[a, b]$ ，等分成 N 個小區間，利用「梯形法」，求取 x^2 這個函數在 $[a, b]$ 區間的定積分數值解，螢幕輸出積分結果 (INTX2)
- 如何將這支主程式改寫為自訂函數？

$$\int_a^b f(x)dx \approx \frac{(b-a)}{2N} [f(a) + 2f(a+\Delta x) + 2f(a+2\Delta x) + \dots + f(b)]$$



PROGRAM INT_X2

IMPLICIT NONE

!*****

! Program to approximate the integral of x^2 over the
! Interval $[a,b]$ using the trapezoidal method.

! Variables used are:

! a, b : the endpoints of the interval of integration

! N : the number of subintervals used

! I : counter

! Dx : the length of the subintervals

! x : a point of subdivision

! F : x^2 at each point of subdivision

! INT : the approximating integral

!

! Input: a, b

! Output: INT (Approximation to integral of x^2 on $[a,b]$)

!*****

INTEGER, PARAMETER :: N=200

INTEGER :: I

REAL :: a=0.0, b=0.0, Dx=0.0, x=0.0, INT=0.0

REAL, DIMENSION(1:N+1) :: F=0.0

```

WRITE(*,*) 'ENTER THE INTERVAL ENDPOINTS [a,b]:'
READ(*,*) a, b

! Calculate subinterval length and initialize X
Dx = (b-a) / REAL(N)
! Now calculate the Y=F(X) at each subdivision point
DO I = 1, N+1
    x = a + (REAL(I)-1)*Dx
    F(I) = x**2
ENDDO
! Now calculate the integral of f(X) over [a,b]
! using trapezoidal method
DO I= 2, N
    INT = INT + 2*f(I)
ENDDO
INT = (Dx/2) * (f(1) + INT + f(N+1))

! Output result on screen
WRITE(*,*) 'integral of x^2 over [a,b]=' , INT

END PROGRAM INT_X2

```

```
PROGRAM INT_X2 FUNCTION INT_X2(a,b)
IMPLICIT NONE
...
REAL :: INT_X2
REAL, INTENT(in) :: a, b
INTEGER, PARAMETER :: N=200
INTEGER :: I
REAL :: a=0.0, b=0.0, Dx=0.0, x=0.0, INT=0.0
REAL, DIMENSION(1:N+1) :: F=0.0
...
```

```
END PROGRAM INT_X2 RETURN
END FUNCTION INT_X2
```

```

WRITE(*,*) 'ENTER THE INTERVAL ENDPOINTS [a,b]:'
READ(*,*) a, b
! Calculate subinterval length and initialize X
Dx = (b-a) / REAL(N)
! Now calculate the Y=F(X) at each subdivision point
DO I = 1, N+1
    x = a + (REAL(I)-1)*Dx
    F(I) = x**2
ENDDO
! Now calculate the integral of f(X) over [a,b]
! using trapezoidal method
DO I= 2, N
    INT = INT + 2*f(I)
ENDDO
INTINT_X2 = (Dx/2) * (f(1) + INT + f(N+1))

! Output result on screen
WRITE(*,*) 'integral of x^2 over [a,b]=', INT

END PROGRAM INT_X2RETURN
END FUNCTION INT_X2

```


使用自訂函數INT_X2的主程式範例

```
PROGRAM test_INT_X2
IMPLICIT NONE
!*****
! Program to approximate the integral of x^2 over the
! 5 intervals [a,b] from an input file using the
! trapezoidal method.
! Variables used are:
!     MAX   : number of intervals to integrate = 5
!     a, b  : the endpoints of the interval of integration
!     I     : counter
!     out   : arrays of the approximating integrals
! Input:   a, b (read from input file)
! Output:  out (integral of x^2 on [a,b], to output file)
!*****
INTEGER :: I
INTEGER, PARAMETER :: MAX=5
REAL :: a=0.0, b=0.0, out=0.0
REAL :: INT_X2
```

這支主程式會從interval.txt檔案依序讀入5組積分區間（a, b的值），利用自訂函數INT_X2，計算出X²在這些區間的定積分數值解，再依序寫入INTx2.txt檔案中

```
OPEN (8, FILE='interval.txt', FORM='formatted', STATUS='old')
OPEN (9, FILE='INTx2.txt', FORM='formatted', STATUS='unknown')

DO I=1, MAX
    READ (8, *) a, b
    out=INT_X2(a,b)
    WRITE (9, *) 'Integral of x^2 over (' ,a, ', ', b, ')=' ,out
ENDDO

END PROGRAM test_INT_X2
```

這支主程式會從interval.txt檔案依序讀入5組積分區間（a, b的值），利用自訂函數INT_X2，計算出 $F(X)=X^2$ 在這些區間的定積分數值解，再依序寫入INTx2.txt檔案中