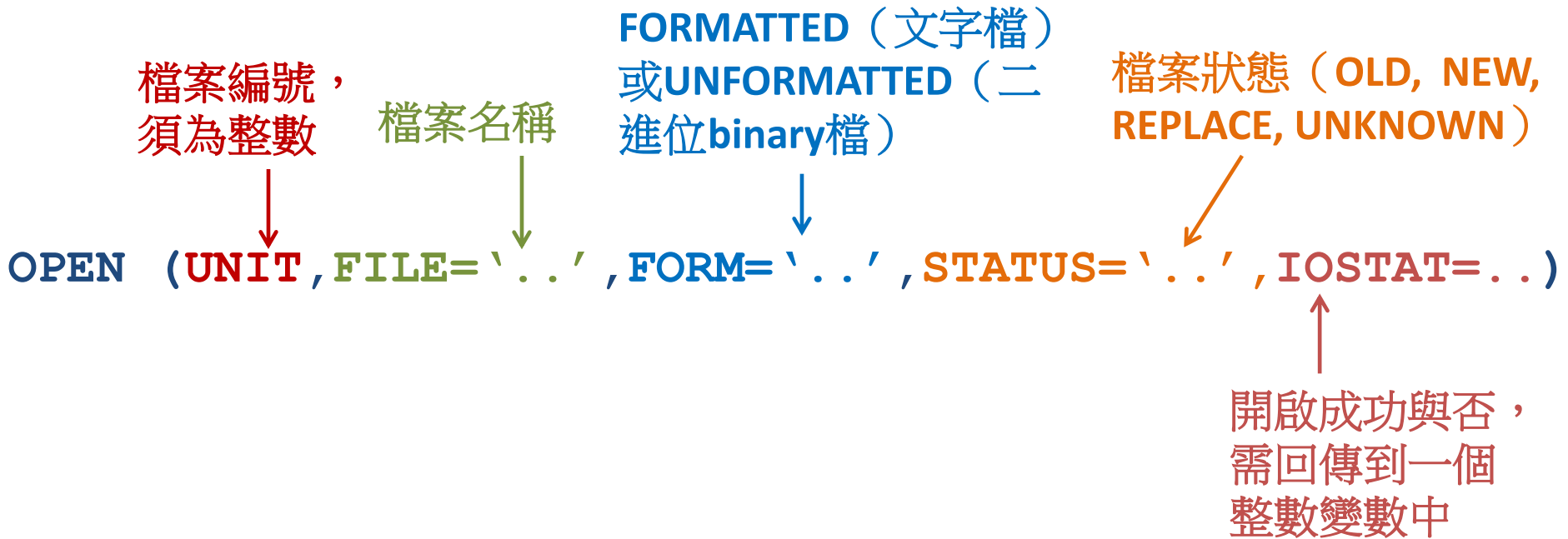


# FORTRAN

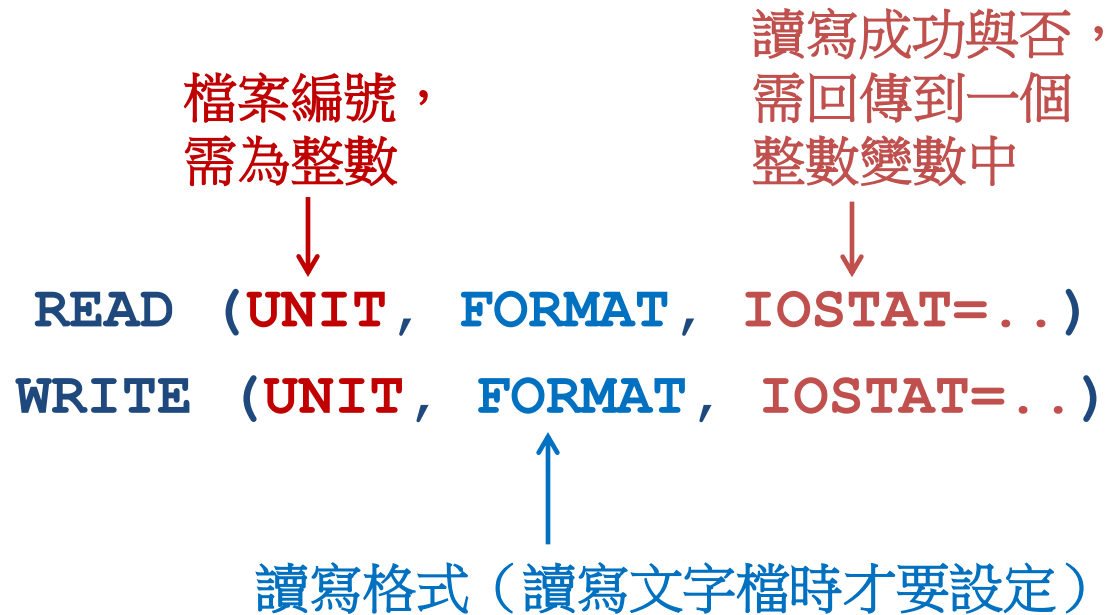
格式化讀寫 (formatted I/O)

# 整理：開啟檔案的各種設定



這段影片是針對FORM='FORMATTED'，也就是文字檔的讀寫設定做介紹，其餘部份請參考其他教學影片

# 整理：讀寫檔案的設定



這段影片介紹如何控制文字檔的讀寫格式，其餘部份請參考其他教學影片

# 回憶：用預設格式輸出檔案

- 語法：**WRITE(UNIT,\*)**

↑  
要輸出的檔案代碼，  
\*=使用預設（鍵盤）

↑  
使用預設格式輸出檔案  
（由FORTRAN自動判斷）

```
PROGRAM write_format
```

```
IMPLICIT NONE
```

```
INTEGER :: A=123
```

```
REAL :: B=1.23456
```

```
CHARACTER (LEN=8) :: C='FORTRAN!'
```

```
OPEN(10,FILE='format_output.txt',FORM='FORMATTED',&  
STATUS='UNKNOWN')
```

```
WRITE(10,*) A, B, C
```

```
END PROGRAM write_format
```

程式範例檔：

/home/teachers/weitingc/lecture\_ex/write\_format.f95

輸出檔案結果：（\_表示空格）

        123        1.23456        FORTRAN!

如何改變輸出數字的精確度，或設定數字、文字之間空格數？

→ 格式化輸出

# 格式化輸出 -- FORMAT

- 如果想要完善地控制程式輸出檔案的格式，就要給定格式化輸出的設定
- 語法：**WRITE (unit, nnn) ...**  
**nnn FORMAT (...)**
  - **nnn**為自訂的數字標籤，需是整數，標記在FORMAT格式碼那一行的開頭，並且在WRITE指令中填入，就可設定輸出的格式碼位置。
  - FORMAT (...) 內填入格式碼：與要輸出的變數種類必須一致，用來設定變數寫出的格式。格式碼之間用 **,** 隔開。

```
...  
WRITE (10,100) A, B, C  
100 FORMAT (2x,I3,2x,F5.2,2x,A8)  
...
```

輸出檔案結果：   123   1.23   FORTRAN!

# 實數變數的FORMAT格式碼 (1)

- 浮點數格式：**Fw.d** (**w**=總字元數，包括小數點、負號、空格；**d**=小數點後的位數)
- 舉例：

```
...  
WRITE(10,100) A, B, C  
100 FORMAT(2x,I3,2x,F5.2,2x,A8)  
  
WRITE(10,101) A, B, C  
101 FORMAT(2x,I3,2x,F5.3,2x,A8)  
...
```

輸出檔案結果：

```
___ 123 ___ 1.23 ___ FORTRAN!  
___ 123 ___ 1.235 ___ FORTRAN!
```

## 實數變數的FORMAT格式碼 (2)

- 科學記號格式：**Ew.d****ek** (**w**=總字元數，包括小數點、正負號、E、空格；**d**=小數點後位數，**k**=指數位數)
- 舉例：

```
...  
WRITE(10,100) A, B, C  
100 FORMAT(2x,I3,2x,F5.2,2x,A8)  
  
WRITE(10,102) A, B, C  
102 FORMAT(2x,I3,2x,E11.3e3,2x,A8)  
...
```

輸出檔案結果：

```
___ 123 ___ 1.23 ___ FORTRAN!  
___ 123 ___ 0.123E+001 ___ FORTRAN!
```

# 整數變數的FORMAT格式碼

- 整數變數：**Iw.d** (**w**=總字元數，包括負號，**d**=至少要輸出的位數，若不足用空格補滿)

- 舉例：

```
WRITE(10,100) A, B, C
```

```
100 FORMAT(2x,I3,2x,F5.2,2x,A8)
```

```
WRITE(10,103) A, B, C
```

```
103 FORMAT(2x,I4,2x,F5.2,2x,A8)
```

```
WRITE(10,104) A, B, C
```

```
104 FORMAT(2x,I2,2x,F5.2,2x,A8)...
```

輸出檔案結果：

```
  123   1.23  FORTRAN!  
  123   1.23  FORTRAN!  
  **    1.23  FORTRAN!
```

！！整數、實數變數：若給定的總字元數不足，輸出結果會變成\*號！！



# 字串變數與空格的FORMAT格式碼

- 字串變數：**Aw** (**w**=總字元數，若輸出的字串變數長度不足，前方會用空格補滿)
- 空格：**wX** (**w**=連續空格總數)
- 舉例：

```
WRITE(10,100) A, B, C
100 FORMAT(2X,I3,2X,F5.2,2X,A8)

WRITE(10,105) A, B, C
105 FORMAT(1X,I3,1X,F5.2,1X,A4)

WRITE(10,106) A, B, C
106 FORMAT(2X,I3,2X,F5.2,A10)
```

輸出檔案結果：

```
 123  1.23  FORTRAN!
 123  1.23  FORT
 123  1.23  FORTRAN!
```

# 設定輸出格式 - 加入字串

- 可以在格式碼中直接設定輸出字串。字串要用單引號夾起，用,與其他格式碼分隔。

- 舉例：

```
WRITE(10,100) A, B, C
```

```
100 FORMAT(2X,I3,2X,F5.2,2X,A8)
```

```
WRITE(10,107) A, B, C
```

```
107 FORMAT(1X,'A=',I3,1X,'B=',F5.2,1X,'C=',A8)
```

輸出檔案結果：

```
 123    1.23  FORTRAN!
```

```
A=123 B= 1.23 C=FORTRAN!
```

# 設定輸出格式 – 連續輸出相同格式

- 若需要連續輸出數個格式、類型都相同的變數，可以用格式碼前的數字設定重複輸出的次數（如 **3F6.1**，代表用 F6.1 格式連續輸出三個變數）
- 如何格式化輸出下面的資料檔？

B00209000 89.5 90.2 75.0 20

**FORMAT (A9, 3F6.1, I4.2)** 或

**FORMAT (A9, 3(2X, F4.1), 2X, I2)**

# 格式化讀取

- 使用類似的語法：**READ (unit, nnn) . . . .**  
**nnn FORMAT ( . . . . )**

舉例--讀入檔案的內容：    123    1.23    FORTRAN!

```
PROGRAM read_format
```

```
IMPLICIT NONE
```

```
INTEGER :: A=0
```

```
REAL :: B=0.0
```

```
CHARACTER (LEN=8) :: C
```

```
OPEN (20, FILE='format_input.txt', FORM='FORMATTED', STATUS='OLD')
```

```
READ (20, 100) A, B, C
```

```
100 FORMAT (2X, I3, 2X, F5.2, 2X, A8)
```

```
WRITE (*, *) A, B, C
```

```
END PROGRAM read_format
```

程式範例檔：

/home/teachers/weitingc/lecture\_ex/read\_format.f95

執行程式，螢幕顯示：          123          1.230000          FORTRAN!

# 設定讀取格式 – 利用空格跳過部份字元

- 可以利用空格(X) 跳過檔案中的部份字元不讀入

◦ 舉例：

讀入檔案的內容：

123 1.23 FORTRAN!

```
PROGRAM read_format
IMPLICIT NONE
INTEGER :: A=0
REAL :: B=0.0
CHARACTER(LEN=8) :: C
OPEN(20,FILE='format_input.txt',FORM='FORMATTED',STATUS='OLD')
READ(20,101) A, C
101 FORMAT(2X,I3,9X,A8)
WRITE(*,*) A, B, C
END PROGRAM read_format
```

執行程式，螢幕顯示：

123 0.000000 FORTRAN!

只讀取了A, C兩個變數，用9x跳過檔案中1.23這個數字，故B=0.0

# 檔案讀寫與格式化輸入輸出一秘訣

- 記得先開啟要讀寫的檔案，檢查檔案名稱、路徑、狀態設定
- 不同檔案要搭配各自的unit編號，不可共用
- 檢查READ, WRITE的unit編號是否正確（小心不要把資料輸出到input檔案中！！）
- FORMAT格式碼必須和要讀寫的變數類型相符，注意空格數目、變數總字元數
- 格式化輸出的開頭第一個字元會被解讀成列印控制碼，盡量保留為空格(1x)，避免自動換行、換頁輸出

# 格式化讀寫：範例（IO\_format.f95）

- 範例檔

`/home/teachers/weitingc/lecture_ex/IO_format.f95`

- 輸入資料檔

`/home/teachers/weitingc/lecture_ex/IO_input.txt`

- 程式說明：讀入資料檔中的表格，輸出到另一個檔案，將表格轉置，且數值精確度改為小數點下兩位

讀入檔案的內容 (IO\_input.txt)：

50.0	47.0	70.0	89.0	30.0	46.0	=	55.33
37.0	67.0	85.0	65.0	68.0	98.0	=	70.00
25.0	45.0	65.0	48.0	10.0	36.0	=	38.17
89.0	56.0	82.5	45.0	30.0	65.0	=	61.25
68.0	78.0	95.0	76.0	98.0	65.0	=	80.00
====	====	====	====	====	====		
53.8	58.6	79.5	64.6	47.2	62.0		

```

PROGRAM IO_format
IMPLICIT NONE
INTEGER,PARAMETER :: nrow=5, ncol=6
REAL,DIMENSION(1:nrow,1:ncol) :: Exam = 0.0
REAL,DIMENSION(1:nrow) :: P_ave = 0.0
REAL,DIMENSION(1:ncol) :: S_ave = 0.0
INTEGER :: r,c
...

```

(宣告區)

Exam(1:nrow,1:ncol)

讀入檔案的內容 (IO\_input.txt) :

P\_ave(1:nrow)

50.0	47.0	70.0	89.0	30.0	46.0	=	55.33
37.0	67.0	85.0	65.0	68.0	98.0	=	70.00
25.0	45.0	65.0	48.0	10.0	36.0	=	38.17
89.0	56.0	82.5	45.0	30.0	65.0	=	61.25
68.0	78.0	95.0	76.0	98.0	65.0	=	80.00

====

S\_ave(1:ncol)

53.8	58.6	79.5	64.6	47.2	62.0
------	------	------	------	------	------



```

...
! open input file and read table
OPEN(10,FILE='IO_input.txt',STATUS='old')
DO r=1,nrow
    READ(10,100) (Exam(r,c),c=1,ncol), P_ave(r)
100 FORMAT(6(1x,f4.1),3x,f5.2)
END DO
READ(10,*) ! skip a line
READ(10,110) S_ave(1:ncol)
110 FORMAT(6(1x,f4.1))
...

```

(讀入檔案)

讀入檔案的內容 (IO\_input.txt) :

Do r=1, nrow	6(1x,f 4.1)						3x	f5.2
READ(10,100)...	50.0	47.0	70.0	89.0	30.0	46.0	= 55.33	
100 FORMAT...	37.0	67.0	85.0	65.0	68.0	98.0	= 70.00	
ENDDO	25.0	45.0	65.0	48.0	10.0	36.0	= 38.17	
	89.0	56.0	82.5	45.0	30.0	65.0	= 61.25	
	68.0	78.0	95.0	76.0	98.0	65.0	= 80.00	
READ(10,*)	====	====	====	====	====	====		
READ(10,11)...	53.8	58.6	79.5	64.6	47.2	62.0		
110 FORMAT...								

```

! output the transposed table, with 1-digit accuracy
OPEN(20,FILE='IO_output.txt',STATUS='unknown')
DO c=1,ncol
  WRITE(20,200) (Exam(r,c),r=1,nrow),S_ave(c)
200 FORMAT(5(1x,f5.2),' = ',f5.2)
ENDDO
WRITE(20,201) ' ===== ===== ===== ===== ===== '
201 FORMAT(A30)
WRITE(20,210) (P_ave(r),r=1,nrow)
210 FORMAT(5(1x,f5.2))
END PROGRAM IO_format

```

(輸出檔案)

讀入檔案的內容 (IO\_input.txt) :

50.0	47.0	70.0	89.0	30.0	46.0	=	55.33
37.0	67.0	85.0	65.0	68.0	98.0	=	70.00
25.0	45.0	65.0	48.0	10.0	36.0	=	38.17
89.0	56.0	82.5	45.0	30.0	65.0	=	61.25
68.0	78.0	95.0	76.0	98.0	65.0	=	80.00
=====	=====	=====	=====	=====	=====		
53.8	58.6	79.5	64.6	47.2	62.0		

→

輸出檔案的內容 (IO\_output.txt) :

50.00	37.00	25.00	89.00	68.00	=	53.80
47.00	67.00	45.00	56.00	78.00	=	58.60
70.00	85.00	65.00	82.50	95.00	=	79.50
89.00	65.00	48.00	45.00	76.00	=	64.60
30.00	68.00	10.00	30.00	98.00	=	47.20
46.00	98.00	36.00	65.00	65.00	=	62.00
=====	=====	=====	=====	=====		
55.33	70.00	38.17	61.25	80.00		

表格被轉置，且數值精確度都改為小數點下兩位

## 設定讀取格式－練習(2)

- 檔案中的空格，可以被當作變數的一部份讀入

讀入檔案的內容：123 1.23 FORTRAN!

```
PROGRAM read_format
IMPLICIT NONE
INTEGER :: A=0
REAL :: B=0.0
CHARACTER(LEN=8) :: C
OPEN(20,FILE='format_input.txt',FORM='FORMATTED',STATUS='OLD')
READ(20,103) A, B, C
103 FORMAT(I5,F7.2,2X,A8)
WRITE(*,*) A, B, C
END PROGRAM read_format
```

執行程式，螢幕顯示：123 1.230000 FORTRAN!

與之前使用 100 FORMAT(2x,I3,2x,F5.2,2x,A8)讀取結果相同