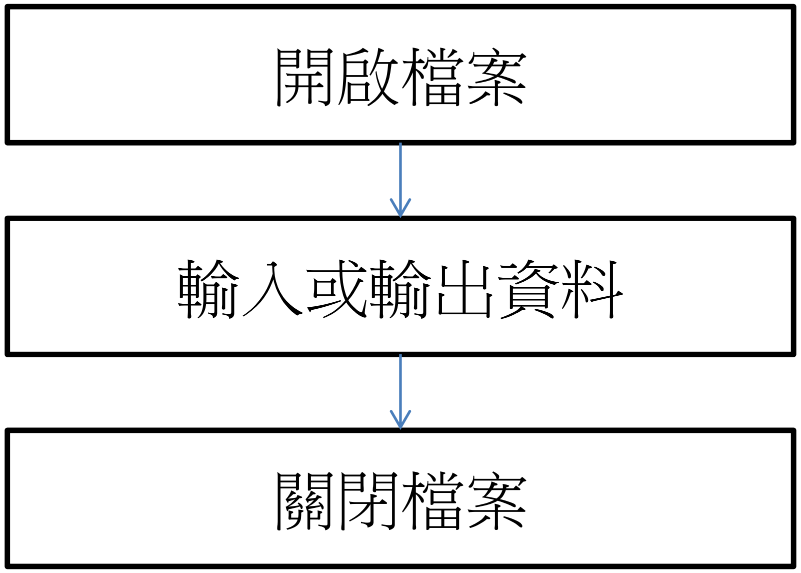
Ch. 16 一般資料的存取

IDL常用的資料格式

資料格式	說明
ASCII	易懂且易讀 資料為格式化(formatted) 適用不同作業平台
Binary	機器碼，讀取速度快 資料為非格式化(unformatted) 各平台有自己的機器碼，不能跨平台使用
SAVE	IDL特有的格式，單一指令即可讀取 是BINARY格式, 可以跨電腦平台使用，但只能用IDL軟體讀取

IDL存取檔案的程式

種類	說明
基本程式(OPEN、READ、WRITE、CLOSE)	IDL內建的基本存取指令，具靈活性
進階程式(READ_ASCII、READ_BINARY)	進階程式是由基本程式寫成，具方便性



IDL開啟檔案的程序

程序	功能
OPENR, Unit, Filename [,/GET_LUN]	開啟現有的檔案，只允許讀取，關鍵字/GET_LUN讓系統自動指定識別碼至引數Unit
OPENW, Unit, Filename [,/GET_LUN]	開啟新的檔案，舊內容會被覆蓋
OPENU, Unit, Filename [,/GET_LUN]	更新現有的檔案

IDL關閉檔案的程序

程序	功能
CLOSE, Unit	關閉己打開的檔案，Unit為檔案識別碼

```
OPENR,1,'ascii.txt'
CLOSE,1
b1=FILE_SEARCH('ascii.txt')
print,'b1=',b1
b2=FILE_SEARCH('ascii_data.txt')
print,'b2=',b2
;OPENR,2,'ascii_data.txt'
;CLOSE,2
b3=FILE_TEST('ascii_out.txt')
print,'b3=',b3
OPENW,3,'ascii_out.txt'
CLOSE,3
b4=FILE_TEST('ascii_out.txt')
print,'b4=',b4
OPENR,unit,'ascii.txt',/GET_LUN
print,'unit=',unit
CLOSE,unit
end
```

EOF函數的語法

程序	功能
Result = EOF(Unit)	判斷檔案是否到底，Unit為檔案識別碼 Result = 0表示到底部，否則為0

```
OPENR,1,'ascii.txt'
result_1=EOF(1)
print,'result_1=',result_1
CLOSE,1
b1=FILE_SEARCH('ascii.txt')
print,'b1=',b1

b3=FILE_TEST('ascii_out.txt')
print,'b3=',b3
OPENW,3,'ascii_out.txt'
result_3=EOF(3)
print,'result_3=',result_3
CLOSE,3
END
```

```
result_1=       0
b1= ascii.txt
b2=
b3=        1
result_3=       1
```

IDL讀取資料的程序

程序	功能
READ, Var1,…,VarN	從工作視窗或指令列讀取變數內容 Var1,…,VarN為一串列的變數名稱
READF, Unit, Var1,…,VarN	讀取格式化(formatted)的檔案，Unit為檔案識別碼
READU, Unit, Var1,…,VarN	讀取非格式化(unformatted)的檔案
READS, Input, Var1,…,VarN	從變數Input中讀取字元

```
a=1
READ,a
print,'1..a=',a
help,a
READ,b1,b2,PROMPT='Enter 2 Number>'
print,'2..b1,b2=',b1,b2
help,b1,b2
END
```

```
IDL> .go
% Compiled module: $MAIN$.
: 2
1..a=       2
A               INT       =        2
Enter 2 Number>4.7,7.3
2..b1,b2=      4.70000      7.30000
B1              FLOAT     =       4.70000
B2              FLOAT     =       7.30000
IDL>
```

```
ascii_data_1.dat
1 3 5 7
2.2 4.4 6.6 9.9
```

```
OPENR,1,'ascii_data_1.dat'
result_1=EOF(1)
print,'result_1=',result_1
a1=10 & a2=11 & a3=12 & a4=13
READF,1,a1,a2,a3,a4
print,a1,a2,a3,a4,format='("a1=",I3," a2=",I3," a3=",I3," a4=",I3)'
result_2=EOF(1)
print,'result_2=',result_2
b1=1.0 & b2=2.0 & b3=3.0 & b4=4.0
READF,1,b1,b2,b3,b4
print,b1,b2,b3,b4,format='("b1=",F4.2," b2=",F4.2," b3=",F4.2," b4=",F4.2)'
result_3=EOF(1)
print,format='(a10,"=",i3)','result_3',result_3
close,1
END
```

```
IDL> .go
% Compiled module: $MAIN$.
result_1=       0
a1=  1 a2=  3 a3=  5 a4=  7
result_2=       0
b1=2.20 b2=4.40 b3=6.60 b4=9.90
 result_3=  1
IDL>
```

```
b1=FILE_TEST('ascii_data_1.dat')
print,'b1=',b1
OPENR,1,'ascii_data_1.dat'
result_1=EOF(1)
print,'result_1=',result_1
READF,1,a1
result_2=EOF(1)
print,'result_2=',result_2
READF,1,a2
print,'a1=',a1,' a2=',a2
result_3=EOF(1)
print,'result_3=',result_3
close,1
end
```

```
b1=          1
result_1=      0
result_2=      0
a1=      1.00000 a2=      2.20000
result_3=      1
IDL>
```

```
subdir=['examples','data']
file_1=FILEPATH('worldelv.dat', SUBDIRECTOR=subdir)
OPENR,2,file_1
READU,2,image_1
CLOSE,2
help,image_1
file_2=FILEPATH('worldelv.dat', SUBDIRECTOR=subdir)
image_2=BYTARR(360,360)
print,'max(image_2)=',max(image_2)
OPENR,3,file_2
READU,3,image_2
CLOSE,3
help,image_2
!p.background=255
!p.color=0
device,decomposed=0
print,'2 max(image_2)=',max(image_2)
LOADCT,0
TV,image_2
LOADCT,13
TV,image_2
END
```

```
d='Oceanogrphy'
c=''
READS,d,c
print,'d=',d
print,'c=',c
end
```

```
IDL> .go
% Compiled module: $MAIN$.
d=Oceanogrphy
c=Oceanogrphy
IDL>
```

IDL寫入資料的程序

程序	功能
PRINT [, Expr1,...,ExprN]	寫入格式化的資料至螢幕
PRINTF, Unit [,Expr1,...,ExprN]	寫入格式化的資料至檔案識別碼為Unit的檔案
WRITE, Unit [,Expr1,...,ExprN]	寫入非格式化的資料至檔案識別碼為Unit的檔案

Expr1,...,ExprN 為一串列的變數名稱

IDL讀取與寫入程序共用的關鍵字

關鍵字	說明
FORMAT=value	定義欄位的格式

IDL欄位格式碼的寫法

寫法	說明
[n] FC [+][-][width]	n為重複的數目，FC為格式碼 width為寬度

IDL的常用格式碼

格式碼	說明
A	定義字元格式
F、E、G	定義浮點數格式
I	定義整數格式
H、quoted String	定義字元格式
X	定義空白格式

IDL supports the following format codes

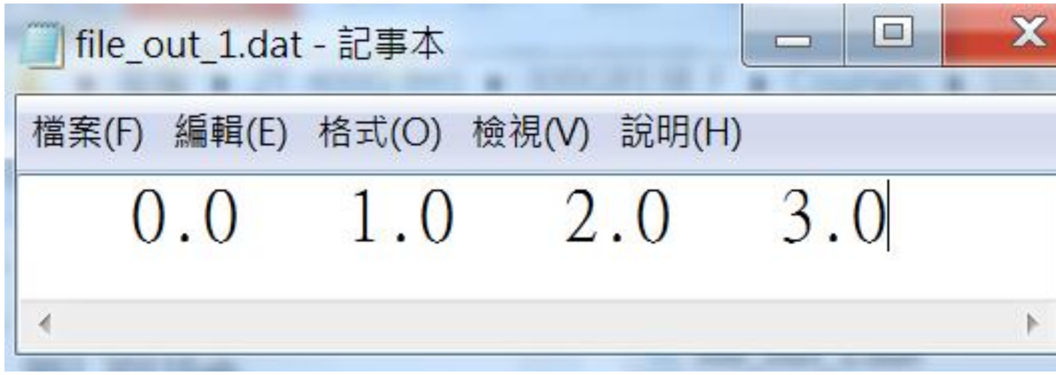
Format Code	Description
A	Transfers character and string values.
:	Terminates format processing if no more items remain in the argument list. No effect if data still remains on the list.
$	On output, suppresses the newline. Ignored on input.
F, D, E, and G	Transfer floating-point values.
B, I, O, and Z	Transfer binary, integer, octal, or hexadecimal values.
Q	On input, returns the number of characters that remain to be transferred. During output, skips the corresponding output list element.
"string" and H	On output, the string contents are written out. Ignored on input.
T*n*	Tab to the *n*-th absolute position in the current record.
TL*n*	Tab left *n* characters.
TR*n*	Tab right *n* characters.
*n*X	Skips *n* character positions.
C()	Transfers calendar data.
C printf-Style	Use a C printf-style format string within a FORTRAN-style format.

```
PRINT,FORMAT='(6(F4.1,","))',INDGEN(7)  ;格式全部使用
print,FORMAT='(6(I1,:,","))',INDGEN(7)  ;結束時後面的格式不使用
txt='PC' & w1=2.3
print,FORMAT='("The ",A0," weights ",F5.1," kg")',txt,w1
print,FORMAT='(3(F4.1,","),$)',INDGEN(7)  ;結束後不換行
end
```

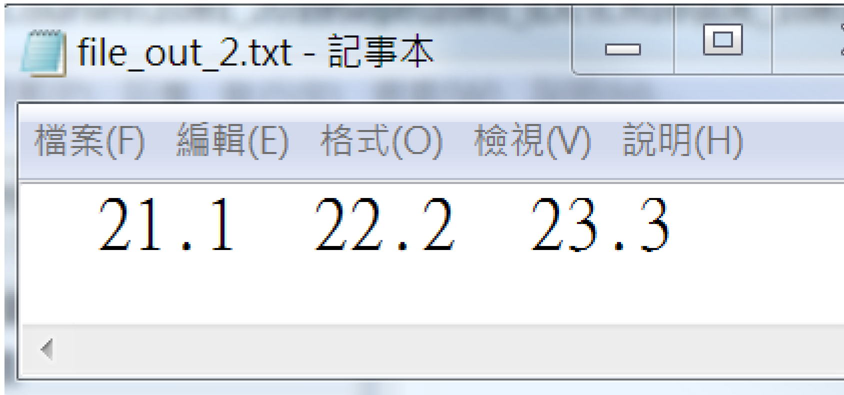
```
0.0, 1.0, 2.0, 3.0, 4.0, 5.0,
 6.0,
0,1,2,3,4,5,
6
The PC weights   2.3 kg
 0.0, 1.0, 2.0,
 3.0, 4.0, 5.0,
 6.0,IDL>
```

```
a=indgen(4)
PRINT,a,FORMAT='(4I3)'
PRINT,a-2,FORMAT='(4F6.1)'
PRINT,a-2,FORMAT='(4F-6.1)'

OPENW,2,'file_out_1.dat'
PRINTF,2,a,FORMAT='(4F6.1)'
CLOSE,2

READ,b1,b2,b3
OPENW,3,'file_out_2.txt'
PRINTF,3,b1,b2,b3,FORMAT='(4F6.1)
CLOSE,3
END
```

```
DL> .go
% Compiled module: $MAIN$.
  0  1  2  3
  -2.0  -1.0   0.0   1.0
-2.0  -1.0  0.0   1.0
IDL>
```



```
   0.0   1.0   2.0   3.0
```



```
  21.1  22.2  23.3
```

```
a=100.0
PRINT,a,FORMAT='(E8.1)'
PRINT,a,FORMAT='(F8.1)'
PRINT,a,FORMAT='(E8.2)'
PRINT,a,FORMAT='(G8.2)'
print,'a2..=',a,FORMAT='(A5,E9.2)'
print,'a2..=',a,FORMAT='(A0,E9.2)'
PRINT,FORMAT='("1234567890")
PRINT,FORMAT='(2Hok,1X,"yes")'
END
```

```
IDL> .go
% Compiled module: $MAIN$.
1.0E+002
   100.0
********
1.0E+002
a2..=1.00E+002
a2..=1.00E+002
1234567890
ok yes
IDL>
```

PLOT, x, y, XTICKFORMAT = 'LABEL_DATE'

Result = LABEL_DATE([, DATE_FORMAT=string/string array]

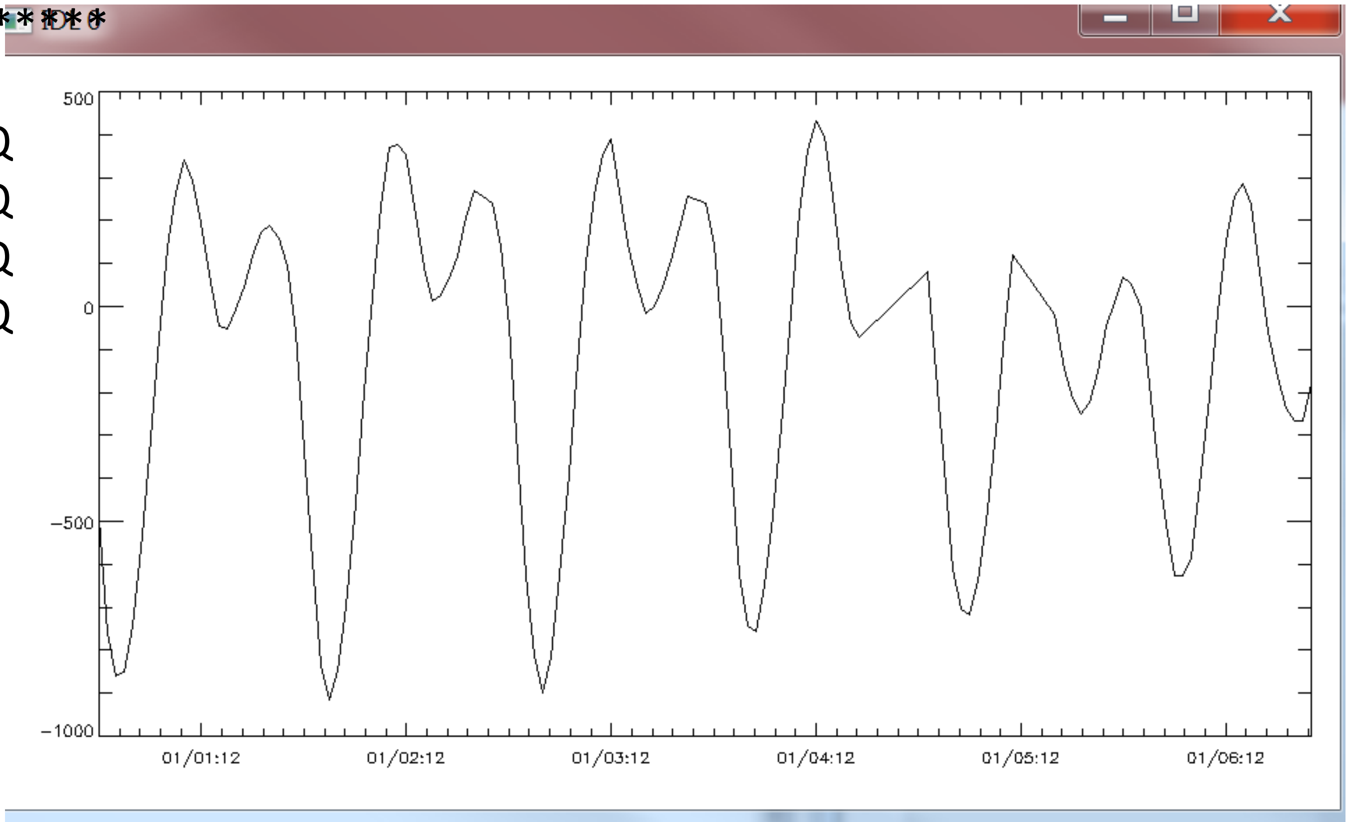
[, AM_PM=2-element vector of strings]
[, DAYS_OF_WEEK=7-element vector of strings]
[, MONTHS=12-element vector of strings] [, OFFSET=value] [, /ROUND_UP])

DATE_FORMAT

Code	Description
%M	Month name
%N	Month number (two digits)
%D	Day of month (two digits)
%Y	Year (four digits, or five digits for negative years)
%Z	Last two digits of the year
%W	Day of the week
%A	AM or PM (%H is then 12-hour instead of 24-hour)
%H	Hours (two digits)
%I	Minutes (two digits)
%S	Second (two digits), followed optionally by %n, where n is an integer (0-9) representing the number of digits after the decimal point for seconds. The default is no decimal places.
%%	Represents the % character

基隆逐時潮位觀測資料
位置：25°09'18"N 121°45'08"E； 基隆港西33號碼頭
儀器型式：Aquatrak 4100 series 超音波式
潮高基準相對臺灣高程基準(TWVD2001):+0.000cm
資料來源：氣象局
yyyymmddhh：西元年月日時
height：潮高
潮高單位：公厘。
時間不連續，表示缺觀測資料。
時間：民國103年1月－民國103年12月

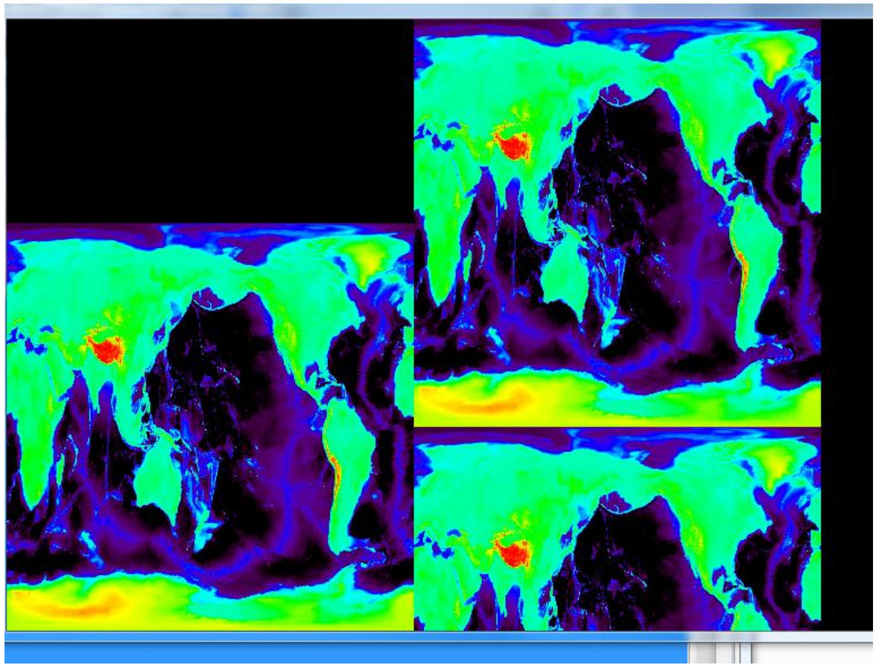
```
***************************
*st    yyyymmddhh height
1516   2014010100  -494 Q
1516   2014010101  -761 Q
1516   2014010102  -861 Q
1516   2014010103  -852 Q
```



```
;實際資料列數建立變數陣列
aa = 0L
bb = " "
cc = 0L
dd = " "
station  = lonarr(nrows)
datehour = strarr(nrows)
height   = lonarr(nrows)
qc       = strarr(nrows)
;跳開一次非資料列
openr,1,'kl_tide_1.txt'
dummy = " "
for i = 1,12 do begin
readf,1,dummy
endfor
;讀實際資料，並存入變數陣列
count = 0
while (NOT EOF(1)) do begin
          readf,1,format = "(I4,3X,A10,2X,I4,1X,A1)",aa,bb,cc,dd
          station(count) = aa
  datehour(count)= bb
  height(count)  = cc
  qc(count)  = dd
          count = count + 1
endwhile
close,1
```

```
;實際時間分出
year  =  fix(strmid(datehour,0,4))
month =  fix(strmid(datehour,4,2))
day   =  fix(strmid(datehour,6,2))
hour  =  fix(strmid(datehour,8,2))

;畫簡圖
time=julday(month,day,year,hour,0,0)
void=label_date(date_format='%N/%D:%H')
plot,time,height,xstyle=1,xtickformat='label_date'
end
```

```
subdir=['examples','data']
file_1=FILEPATH('worldelv.dat', SUBDIRECTOR=subdir)
print,file_1
OPENR,2,file_1
image_1=BYTARR(360,360)
READU,2,image_1
CLOSE,2
help,image_1
TV,image_1
LOADCT,13
TV,image_1,1
OPENW,3,'elv_world.dat'
WRITEU,3,image_1
CLOSE,3
OPENR,4,'elv_world.dat'
image_2=BYTARR(360,360)
READU,4,image_2
CLOSE,4
TV,image_2,3
end
```



讀取與寫入IDL SAVE資料格式

指令	功能
SAVE [,Var1,…,VarN]	以IDL SAVE格式儲存資料 Var1,…,VarN
RESTORE	讀取已儲存的SAVE檔
RESOLVE_ALL	把與主程式相關的所有程式 (包括IDL內建函數和程序)呼叫進入系統
SAVE,/ROUTINES	進行應用程式的封包作業

```
a=1 & b=2.1
SAVE   ;儲存工作區的所有變數或特定變數至預設檔案idlsave.dat
end
```

```
RESTORE  ;將儲存在idlsave.dat中的資料重新載入，仍保留原變數和資料
print,'a,b=',a,b,FORMAT='(A0,2F6.2)'
END
```

```
% Compiled module: $MAIN$.
a,b=  1.00  2.10
IDL>
```

```
a=1 & b=2.1
SAVE,FILENAME='idl_save_file_1.sav' ;將所有變數及資料存入
idl_save_file_1.sav檔案中
print,FORMAT='(2(A5,F5.2))','a=',a,'b=',b
;RESTORE,'idl_save_file_1.sav'  ;在命令行使用,
END
```

```
RESTORE,'idl_save_file_1.sav'  ;將
idl_save_file_1.sav中的所有變數及內容重新載入
print,FORMAT='(2(A5,F5.2))','a=',a,'b=',b
delvar,a,b
end
```

```
% Compiled module: $MAIN$.
  a= 1.00   b= 2.10
% Active main program terminated by call to DELVAR.
IDL>
```

```
pro IDL_ch16_main_2
c=7.2
print,'c=',c
sub1,c,c1
print,'c1=',c1
sub2,c,c2
print,'c2=',c2
END
```

```
pro sub1,c,d
PRINT,'Calling sub1.pro'
d=c-5.0
RETURN
END
```

```
PRO sub2,c,e
 PRINT,'Calling sub2.pro'
 e=2*c
 RETURN
END
```

結果

```
IDL> idl_ch16_main_2
% Compiled module: IDL_CH16_MAIN_2.
c=      7.20000
% Compiled module: SUB1.
Calling sub1.pro
c1=      2.20000
% Compiled module: SUB2.
Calling sub2.pro
c2=      14.4000
IDL>
```

```
;把主程式相關的所有程式(包括IDL內建函數和程序)呼叫進入系統
IDL> resolve_all
;把已經在系統的所有程式打包至file_sav_8.sav
 IDL> save,/routines,filename='file_sav_8.sav‘
;恢復file_sav_8.sav檔案中的所有程式
IDL> restore,'file_sav_8.sav‘
; 直接執行主程式idl_ch16_main_2，因為file_sav_8.sav內的副程式已經編譯過
IDL> idl_ch16_main_2
c=      7.20000
Calling sub1.pro
c1=      2.20000
Calling sub2.pro
c2=      14.4000
IDL>
```

READ_ASCII函數的語法

語法	說明
Result = READ_ASCII([Filename])	讀取檔案Filename的內容至變數Result

```
ascii_data_1.dat 資料
1 3 5 7
2.2 4.4 6.6 9.9
```

```
a1=READ_ASCII('ascii_data_1.dat')
help,a1,/STRUCTURE
end
```

```
IDL> .go
% Compiled module: $MAIN$.
** Structure <23525f0>, 1 tags, length=32, data length=32, refs=1:
   FIELD1          FLOAT     Array[4, 2]
IDL>
```

```
name1='file_5.dat'
data1=READ_ASCII(name1,COUNT=num,DATA_START=2, $
 HEADER=head_1) ;定義前2行是表頭資料，儲存至變數head_1，第3行開始至最後一行是資料
print,'header=',head_1
print,'header(0)=',head_1(0)
print,'data=',data1
print,'data1=',data1.FIELD1(0:2)
data2=READ_ASCII(name1,RECORD_START=3,NUM_RECORDS=2)
;定義資料讀取從第4開始，讀取2行
help,data2
print,'data2=',data2
print,data2.FIELD1(1)
print,data2.FIELD1(1:3)
print,data2.FIELD1(0:1,1)
end
```

```
file_5.dat
Header Line 1
Header Line 2
1.1 1.2 1.3
2.1 2.2 2.3
3.1 3.2 3.3
```

```
IDL> .go
% Compiled module: $MAIN$.
header= Header Line 1 Header Line 2
header(0)=Header Line 1
data={      1.10000      1.20000      1.30000
      2.10000      2.20000      2.30000
      3.10000      3.20000      3.30000
}
data1=      1.10000      1.20000      1.30000
** Structure <2f943e10>, 1 tags, length=24, data length=24, refs=1:
   FIELD1          FLOAT     Array[3, 2]
data2={      2.10000      2.20000      2.30000
      3.10000      3.20000      3.30000
}
      2.20000
      2.20000      2.30000      3.10000
      3.10000      3.20000
IDL>
```

READ_ASCII函數的關鍵字

關鍵字	說明
COUNT=variable	讀取資料的行數
DATA_START=lines_of_skip	跳掉行數
HEADER=variable	取出表頭
NUM_RECORDS=value	預設資料讀取的總行數
RECORD_START=index	設定資料讀取的開始行數
COMMENT_SYMBOL=string	定義註解的符號
DELIMITER=string	各個欄位的區隔符號
MISSING_VALUE=value	無資料時的指定值
TEMPLATE=value	資料排列版型，由 ASCII_TEMPLATE程序決定

```
x=findgen(10)
y=x^2+3*x+1
openw,3,'IDL_ascii_xy.dat'
printf,3,'x-value','y-value',format='(2A8)'
n=size(x)
for i=0,n(1)-1 do begin
   printf,3,x(i),y(i),format='(2F8.2)'
endfor
close,3
plot,x,y
end
```

```
name1='IDL_ascii_xy.dat'
data1=READ_ASCII(name1,COUNT=num,DATA_START=1, $
  HEADER=head_1)  ;定義前2行是表頭資料，儲存至變數head_1，第3行開始至最後一行是資料
help,head_1
print,head_1
head_1_n=strsplit(head_1,/EXTRACT)
head_1_x = head_1_n(0)
head_1_y = head_1_n(1)
print,head_1_y
help,data1
print,data1
n=size(data1.FIELD1)
xa=data1.FIELD1(0,*)
ya=data1.FIELD1(1,*)
help,xa
erase
plot,xa,ya,xtitle=head_1_x,ytitle=head_1_y
end
```

```
name1='IDL_ascii_xy.dat'
data2=READ_ASCII(name1,RECORD_START=1,NU
M_RECORDS=5)
help,data2
xb=data2.FIELD1(0,*)
yb=data2.FIELD1(1,*)
plot,xb,yb
end
```

READ_BINARY函數的語法

語法	說明
Result = READ_BINARY([Filename])	輸入檔名Filename，輸出變數是Result

READ_BINARY函數的關鍵字

關鍵字	說明
DATA_START=value	開始讀取資料的位元組，亦即跳掉的位元組數目
DATA_TYPE=type_codes	資料的型態碼
DATA_DIMS=array	資料的維度
TEMPLATE=template	資料排列版型，由BINARY_TEMPLATE函數決定
ENDIAN=string	設立資料位元組在記憶體中的排列方式

資料型態碼(type code)

型態碼	資料型態	型態碼	資料型態
0	UNDEFINED	8	STRUCT
1	BYTE	9	DCOMPLEX
2	INT	10	POINTER
3	LONG	11	OBJREF
4	FLOAT	12	UINT
5	DOUBLE	13	ULONG
6	COMPLEX	14	LONG64
7	STRING	15	ULONG64