

Ch.05 陣列的建立與運算

各維度的函數型式和引數個數

函數型式	維度
Function_Name(n)	一維，一個輸入引數
Function_Name(n,m)	二維，二個輸入引數
Function_Name(n,m,o)	三維，三個輸入引數
Function_Name(n,m,o,p)	四維，四個輸入引數

```

x1=fltarr(3)
x2=findgen(3,2)
x3=findgen(4,3,2)
help,x1,x2,x3
print,x1,x2,x3
y3=replicate(2.0,4,3,2)
help,y3 & print,y3
z3=randomu(seed,4,3,2)
help,z3 & print,z3
a3=Lindgen(4,3,2)
help,a3 & print,a3
end

```

```

% Compiled module: $MAIN$.
X1      FLOAT   = Array[3]
X2      FLOAT   = Array[3, 2]
X3      FLOAT   = Array[4, 3, 2]
  0.000000  0.000000  0.000000
  0.000000  1.000000  2.000000
  3.000000  4.000000  5.000000
  0.000000  1.000000  2.000000  3.000000
  4.000000  5.000000  6.000000  7.000000
  8.000000  9.000000 10.0000  11.0000

 12.0000  13.0000  14.0000  15.0000
 16.0000  17.0000  18.0000  19.0000
 20.0000  21.0000  22.0000  23.0000
Y3      FLOAT   = Array[4, 3, 2]
  2.00000  2.00000  2.00000  2.00000
  2.00000  2.00000  2.00000  2.00000
  2.00000  2.00000  2.00000  2.00000

  2.00000  2.00000  2.00000  2.00000
  2.00000  2.00000  2.00000  2.00000
  2.00000  2.00000  2.00000  2.00000
Z3      FLOAT   = Array[4, 3, 2]
  0.912302  0.971843  0.736085  0.0513237
  0.337118  0.371067  0.380022  0.845369
  0.227484  0.971529  0.842384  0.327832

  0.555780  0.778354  0.863558  0.102703
  0.786978  0.996309  0.512195  0.264418
  0.316899  0.822380  0.618322  0.192623
A3      LONG    = Array[4, 3, 2]
   0      1      2      3
   4      5      6      7
   8      9     10     11

  12     13     14     15
  16     17     18     19
  20     21     22     23
IDL>

```

轉換陣列資料型態的函數(A為陣列)

函數	功能
BYTE(A)	轉換A中所有元素為短整數
FIX(A)	轉換A中所有元素為整數
UINT(A)	轉換A中所有元素為無號整數
LONG(A)	轉換A中所有元素為長整數
ULONG(A)	轉換A中所有元素為無號長整數
LONG64(A)	轉換A中所有元素為64位元長整數
ULONG64(A)	轉換A中所有元素為64位元無號長整數
FLOAT(A)	轉換A中所有元素為浮點數
DOUBLE(A)	轉換A中所有元素為雙精度浮點數
COMPLEX(A)	轉換A中所有元素為複數的實部
DCOMPLEX(A)	轉換A中所有元素為雙精度複數的實部
STRING(A)	轉換A中所有元素為字元

```

a=findgen(4,3,2)
help,a & print,a
a=a/3.
print,a
b=fix(a)
help,b & print,b
c=round(a)
help,c & print,c
d=string(a)
help,d & print,d
e=['abc','a12','12a']
help,e & print,e
f=fix(e)
help,f & print,f
g=float(d)
help,g & print,g
end

```

```

% Compiled module: $MAIN$.
A      FLOAT  = Array[4, 3, 2]
  0.000000  1.00000  2.00000  3.00000
  4.00000  5.00000  6.00000  7.00000
  8.00000  9.00000 10.0000  11.0000

 12.0000 13.0000 14.0000 15.0000
 16.0000 17.0000 18.0000 19.0000
 20.0000 21.0000 22.0000 23.0000
 0.000000 0.333333 0.666667 1.000000
 1.333333 1.66667 2.00000 2.33333
 2.66667 3.00000 3.33333 3.66667

 4.00000 4.33333 4.66667 5.00000
 5.33333 5.66667 6.00000 6.33333
 6.66667 7.00000 7.33333 7.66667
B      INT    = Array[4, 3, 2]
  0  0  0  1
  1  1  2  2
  2  3  3  3

  4  4  4  5
  5  5  6  6
  6  7  7  7
C      LONG   = Array[4, 3, 2]
  0  0  1  1
  1  2  2  2
  3  3  3  4

  4  4  5  5
  5  6  6  6
  7  7  7  8
D      STRING = Array[4, 3, 2]
  0.000000 0.333333 0.666667 1.000000
 1.333333 1.66667 2.00000 2.33333
 2.66667 3.00000 3.33333 3.66667

 4.00000 4.33333 4.66667 5.00000
 5.33333 5.66667 6.00000 6.33333
 6.66667 7.00000 7.33333 7.66667
E      STRING = Array[3]
abc a12 12a
% Type conversion error: Unable to convert given STRING to Integer.
% Detected at: $MAIN$      13 H:\300GB138_F\Courses\1061_2017Sept\1
061_IDL\IDL_ch05_02.pro
% Type conversion error: Unable to convert given STRING to Integer.
% Detected at: $MAIN$      13 H:\300GB138_F\Courses\1061_2017Sept\1
061_IDL\IDL_ch05_02.pro
F      INT    = Array[3]
  0  0 12
G      FLOAT  = Array[4, 3, 2]
  0.000000 0.333333 0.666667 1.000000
 1.333333 1.66667 2.00000 2.33333
 2.66667 3.00000 3.33333 3.66667

```

陣列與函數所使用的區隔符號

符號	說明
() 小括號	函數用小括號隔參數
[] 中括號	變數用中括號區隔下標

IDL的優先將名稱當作函數來處理，最好的方式是變數的名稱避免使用到函數相同的名稱。

```
complex= bindgen(5,4)
print,complex
print,complex[3:4,2:3]
print,'complex[4,3]=' ,complex[4,3]
print,'complex(4,3)=' ,complex(4,3)
end
```

```
% Compiled module: $MAIN$.
```

```
0 1 2 3 4
```

```
5 6 7 8 9
```

```
10 11 12 13 14
```

```
15 16 17 18 19
```

```
13 14
```

```
18 19
```

```
complex[4,3]= 19
```

```
complex(4,3)=( 4.00000, 3.00000)
```

```
IDL>
```

下標操作符號和處理下標的數學函數(A為陣列)

符號	說明
0	代表下標的開始
*	代表全部下標
:	宣告下標範圍或下標增加量, $n1 : n2$, 起始下標 $n1$, 結束下標 $n2$ 。 $n1:n2:n3$ 起始下標 $n1$, 結束下標 $n2$, 下標增加量 $n3$ 。
,	區隔陣列的維度
<code>Result = Min(A, Subscript)</code>	計算最小值, Result 記錄著最小值, Subscript 記錄著最小值的下標位置。
<code>Result2= ARRAY_INDICES (A, Subscript)</code>	將一維的下標變成多維的下標 Result2

```

x=indgen(4,3,2)
help,x & print,x
print,'x[0:1,1:*,*]=' & print,x[0:1,1:*,*]
print,'x[:,0:1,*]=' & print,x[:,0:1,*]
Max_x = max(x,S_max)
print,'Max_x=',Max_x,' S_max=',S_max
Max_pos = array_indices(x,S_max)
print,'Max_pos=',Max_pos
end

```

```

% Compiled module: $MAIN$.
X      INT      = Array[4, 3, 2]
  0     1     2     3
  4     5     6     7
  8     9    10    11

 12    13    14    15
 16    17    18    19
 20    21    22    23
x[0:1,1:*,*]=
  4     5
  8     9

 16    17
 20    21
x[:,0:1,*]=
  0     1     2     3
  4     5     6     7

 12    13    14    15
 16    17    18    19
Max_x=   23 S_max=   23
% Compiled module: ARRAY_INDICES.
Max_pos=    3     2     1
IDL>

```


陣列轉換的函數

函數	功能
[A, B]	將陣列A和B橫向併排，亦即擴充行(column)
[[A],[B]]	將陣列A和B縱向併排，亦即擴充列(row)
[[[A]],[[B]]]	將陣列A和B縱向併排，亦即擴充層(page)
REVERSE(A, k)	倒轉陣列A中元素的順序，k是倒轉的維度
SHIFT(A, c, d, e)	平移陣列A中元素的順序，c,d和e代表三個維度的各個平移量
REFORM(A, c, d, e)	重新排列陣列A中至維度為c x d x e的陣列但元素總數目不變
TRANSPOSE(A)	轉置陣列A

```
x1=indgen(4,3,2) & x2=x1*2
help,x1,x2 & print,x1,x2
y1=[x1,x2]
help,y1 & print,y1
y2=[[x1],[x2]]
help,y2 & print,y2
y3=[[x1],[x2]]
help,y3 & print,y3
y5_r1=reverse(x1,1)
help,y5_r1 & print,y5_r1
y5_r2=reverse(x1,2)
help,y5_r2 & print,y5_r2
y5_r3=reverse(x1,3)
help,y5_r3 & print,y5_r3
```

```
y6_s200=shift(x1,2,0,0)
help,y6_s200 & print,y6_s200
y6_s210=shift(x1,2,1,0)
help,y6_s210 & print,y6_s210
y6_s001 = shift(x1,0,0,1)
help,y6_s001 & print,y6_s001
y7_rf24_1_1=reform(x1,24,1,1)
help,y7_rf24_1_1 &
print,y7_rf24_1_1
y7_rf12_2_1=reform(x1,12,2,1)
help,y7_rf12_2_1 &
print,y7_rf12_2_1
y7_rf3_2_4=reform(x1,3,2,4)
help,y7_rf3_2_4 &
print,y7_rf3_2_4
y8 = transpose(x1)
help,y8 & print,y8
end
```

陣列的數學運算(陣列間的維度要相同)

指令	說明
$A - B$	將陣列A中的元素減去陣列B中相同位置的元素
$A + B$	將陣列A中的元素加上陣列B中相同位置的元素
$A * B$	將陣列A中的元素乘以陣列B中相同位置的元素
$A + b$	將陣列A中的每一個元素加純量b
A / b	將陣列A中的每一個元素除以純量b
$A ^ b$	將陣列A中的每一個元素取純量b次方
$A \text{ MOD } b$	將陣列A中的每一個元素除以純量b後的餘數
$\text{SIN}(A)$ 、 $\text{COS}(A)$ 、 $\text{TAN}(A)$	將陣列A中的每一個元素聯三角函數
$\text{EXP}(A)$	將陣列A中的每一個元素取自然指數
$\text{ALOG}(A)$ 、 $\text{ALOG10}(A)$	將陣列A中的每一個元素取自然對數值、基底為10的對數值
$\text{ABS}(A)$	將陣列A中的每一個元素取絕對值
$\text{SQRT}(A)$	將陣列A中的每一個元素開根號

```

x1=randomn(seed,4,3,2)
help,x1 & print,x1
x2=indgen(4,3,2)
help,x2 & print,x2
y1=x1+x2
help,y1 & print,y1
y2= x2 mod 3
help,y2 & print,y2
y3 = x2*y2
help,y3 & print,y3
y4=x2*2
help,y4 & print,y4
y5=y3/x2
help,y5 & print,y5
y6=x2^2
help,y6 & print,y6
end

```

```

% Compiled module: $MAIN$.
X1      FLOAT  = Array[4, 3, 2]
  1.94739 -1.57274  1.32265  0.683352
 -0.0531724 -0.529277 -1.34891 -0.508688
  0.00288020 -0.134394 -0.696246  0.520491

-0.120369  0.917640  0.0324694  0.945601
-0.590168 -0.162266  0.241629 -1.95307
  0.437506 -0.927470 -2.02146  0.0902465
X2      INT    = Array[4, 3, 2]
  0  1  2  3
  4  5  6  7
  8  9 10 11

12 13 14 15
16 17 18 19
20 21 22 23
Y1      FLOAT  = Array[4, 3, 2]
  1.94739 -0.572735  3.32265  3.68335
  3.94683  4.47072  4.65109  6.49131
  8.00288  8.86561  9.30375 11.5205

11.8796 13.9176 14.0325 15.9456
15.4098 16.8377 18.2416 17.0469
20.4375 20.0725 19.9785 23.0902
Y2      INT    = Array[4, 3, 2]
  0  1  2  0
  1  2  0  1
  2  0  1  2

  0  1  2  0
  1  2  0  1
  2  0  1  2
Y3      INT    = Array[4, 3, 2]
  0  1  4  0
  4 10  0  7
16  0 10 22

  0 13 28  0
16 34  0 19
40  0 22 46
Y4      INT    = Array[4, 3, 2]
  0  2  4  6
  8 10 12 14
16 18 20 22

24 26 28 30
32 34 36 38
40 42 44 46

```

```

Y5      INT    = Array[4, 3, 2]
  0  1  2  0
  1  2  0  1
  2  0  1  2

  0  1  2  0
  1  2  0  1
  2  0  1  2
Y6      INT    = Array[4, 3, 2]
  0  1  4  9
16 25 36 49
64 81 100 121

144 169 196 225
256 289 324 361
400 441 484 529
% Program caused arithmetic error: Integer
divide by 0
IDL>

```

```
x1=[[[0,!pi/6],[!pi/4,!pi/3]],[[!pi/2,!pi],[!pi*7./6.,!pi*5./4.]]]
help,x1 & print,x1
y1=sin(x1) & y2=cos(x1)
help,y1,y2 & print,y1,y2
y3=asin(y1)*!radeg
help,y3 & print,y3
y6=abs(y3)
help,y6 & print,y6
x2=[[[0.1,1],[2,3]],[[10,20],[100,200]]]
help,x2 & print,x2
y4=alog(x2)
help,y4 & print,y4
y5=alog10(x2)
help,y5 & print,y5
y7=sqrt(x2)
help,y7 & print,y7
end
```

查詢陣列相關資訊的函數

函數	功能
N_ELEMENTS(A)	求出陣列A中元素的個數
FINITE(A)	判斷陣列A中各個元素是否為有限

```

x1=findgen(3,2,3)
help,x1 & print,x1
x1[1,1,2]=!values.f_infinity
x1[1,0,1]=!values.F_NAN
help,x1 & print,x1
y1=finite(x1)
help,y1 & print,y1
y2=n_elements(x1)
help,y2 & print,y2
end

```

```

% Compiled module: $MAIN$.
X1      FLOAT  = Array[3, 2, 3]
0.000000  1.00000  2.00000
3.00000  4.00000  5.00000

6.00000  7.00000  8.00000
9.00000  10.0000  11.0000

12.0000  13.0000  14.0000
15.0000  16.0000  17.0000
X1      FLOAT  = Array[3, 2, 3]
0.000000  1.00000  2.00000
3.00000  4.00000  5.00000

6.00000    NaN  8.00000
9.00000  10.0000  11.0000

12.0000  13.0000  14.0000
15.0000    Inf  17.0000
Y1      BYTE   = Array[3, 2, 3]
1 1 1
1 1 1

1 0 1
1 1 1

1 1 1
1 0 1
Y2      LONG   =      18
18
IDL>

```

查詢陣列中相關資訊的函數 (Array為陣列)

函數	功能
MAX(Array)	取出最大值
MIN(Array)	取出最小值
MEDIAN(Array)	取出中間值
TOTAL(Array)	求出總合
PRODUCT(Array)	求出乘積

```

x1=indgen(3,2,2)
help,x1 & print,x1
print,'Max_x=',max(x1)
Min_x=min(x1,S_min)
print,Min_x & print,'S_min=',S_min
Min_pos=array_indices(x1,S_min)
print,'Min_pos=',Min_pos
print,'Med_x=',median(x1)
y1=total(x1)
help,y1 & print,y1
y2=product(x1+1)
help,y2 & print,y2
y3=factorial(12)
help,y3 & print,y3
end

```

```

% Compiled module: $MAIN$.
X1          INT      = Array[3, 2, 2]
    0      1      2
    3      4      5

    6      7      8
    9     10     11
Max_x=      11
    0
S_min=        0
Min_pos=        0      0      0
Med_x=    6.00000
Y1          FLOAT    =    66.0000
    66.0000
Y2          DOUBLE   =  4.7900160e+008
    4.7900160e+008
% Compiled module: FACTORIAL.
Y3          DOUBLE   =  4.7900160e+008
    4.7900160e+008
IDL>

```


查詢陣列相關資訊的函數(Array為陣列)

關鍵字	說明
SIZE(Array)	回傳陣列的全部資訊。假如第1個元素是維度n，接續的 n 個元素代表各個維度的大小，然後是資料型態碼，最後一個元素是陣列元素的總數目。
SIZE(Array,/DIMENSIONS)	只回傳陣列的各個維度的大小
SIZE(Array,/N_DIMENSIONS)	只回傳陣列的維度
SIZE(Array,/N_ELEMENTS)	只回傳陣列元素的總數目
SIZE(Array,/TNAME)	只回傳陣列的型態名稱
SIZE(Array,/TYPE)	只回傳陣列的型態碼

```

x1=indgen(4,2,3)
help,x1 & print,x1
y1=size(x1)
print,'y1=',y1
y2=size(x1,/N_dimensions)
print,'y2=',y2
y3=size(x1,/Dimensions)
print,'y3=',y3
y4=size(x1,/N_elements)
print,'y4=',y4
y5=size(x1,/TYPE)
print,'y5=',y5
y6=size(x1,/TNAME)
print,'y6=',y6
end

```

```
% Compiled module: $MAIN$.
```

```
X1          INT      = Array[4, 2, 3]
```

```
  0    1    2    3
```

```
  4    5    6    7
```

```
  8    9   10   11
```

```
 12   13   14   15
```

```
 16   17   18   19
```

```
 20   21   22   23
```

```
y1=      3      4      2      3      2      24
```

```
y2=      3
```

```
y3=      4      2      3
```

```
y4=     24
```

```
y5=      2
```

```
y6=INT
```

```
IDL>
```