

# Python 教學

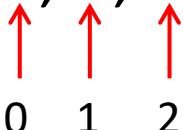
- 變數 Variable
- 數列 list[]
- 迴圈 loop, while
- 判斷 if
- 自訂函數 def()

# 變數 Variable

- 變數型態 `type()`
  - 整數(int) `a=5`
  - 浮點數(float) `b=3.`
  - 字串(str) `'a=5'`
  - 複數(complex) `2+3j`
- 註解 # 與文字 `'a=5'`
  - 註解為綠色文字不被程式作用，而文字則相反

```
>>> x=1
>>> type(x)
<type 'int'>
>>>
>>> x=1.0
>>> type(x)
<type 'float'>
>>>
>>> x=1.0+1.0j
>>> type(x)
<type 'complex'>
```

# 數列

- 數列 tuple (其內容與性質固定)
  - $x=(1, 2, 3)$
- 數列 list (其內容與性質可變更)
  - $x=[1, 2, 3]$
- $x=[1, 2, 3]$   


0   1   2
- $x[0]=1, x[2]=3$

# list 注意事項

```
>>> a=[1,2,3]
>>> b=a
>>> print a,'\n',b
[1, 2, 3]
[1, 2, 3]
```



```
>>> b[1]=4
>>> print a,'\n',b
[1, 4, 3]
[1, 4, 3]
```



```
>>> b[1]=4
>>> print a,'\n',b
[1, 2, 3]
[1, 4, 3]
```



```
>>> a=[1,2,3]
>>> b=a[:]
>>> print a,'\n',b
[1, 2, 3]
[1, 2, 3]
```

# 迴圈 loop, while

- 迴圈 loop 為重複執行相同程式

>>> for 變數 in range(次數):

... 迴圈內執行的程式

```
>>> for i in range(5):  
...     print (i)  
...  
0  
1  
2  
3  
4  
>>>
```

```
>>> s=0  
>>> for i in range(11):  
...     s=s+i  
...     print(i,s)  
...  
(0, 0)  
(1, 1)  
(2, 3)  
(3, 6)  
(4, 10)  
(5, 15)  
(6, 21)  
(7, 28)  
(8, 36)  
(9, 45)  
(10, 55)  
>>>
```

- 迴圈 while 為條件迴圈，滿足條件者重複執行相同程式，直到不滿足條件後離開迴圈

>>> while(進入迴圈的條件):

... 迴圈內執行的程式

```
>>> n=0
>>> while(n<10):
...     print n
...     n+=1
...
0
1
2
3
4
5
6
7
8
9
>>> .
```

# 判斷 if

>>> if (條件):

... 符合條件者的執行程式

... elif(其他條件):

... 符合條件者的執行程式

... else:

... 其餘條件以外者的執行程式

```
>>> for i in range(10):
...     if(i%3==0):
...         print 0
...     elif(i%3==1):
...         print 1
...     else:
...         print 2
...
0
1
2
0
1
2
0
1
2
0
>>>
```



# 符號應用

若  $a=1$ , 當  $b=a$ , 則  $b=1$

$b$ 複製 $a$ 數值

若  $a=1$   $b=1$ , 當  $b==a$ , 則  $\text{true}$

判斷相等

若  $a=1$   $b=1$ , 當  $b!=a$ , 則  $\text{false}$

判斷不相等

若  $a=[1,2]$  當  $b=a$ , 則  $b=[1,2]$

$b$ 複製 $a$ 數值包含性質

若  $a=[1,2]$  當  $b=a[:]$ , 則  $b=[1,2]$

$b$ 只複製 $a$ 數值

若  $a=(1,2)$   $b=[1,2]$ , 當  $b==a$ , 則  $\text{false}$

# 自訂函數 def()

>>>def 函數名稱(變數1, 變數2...):

... 自訂函數內容

... return 回傳數值1, 數值2...

```
>>> def fun(x):  
...     y=x**2  
...     return y  
...  
>>> for i in range(11):  
...     print i,fun(i)  
...  
0 0  
1 1  
2 4  
3 9  
4 16  
5 25  
6 36  
7 49  
8 64  
9 81  
10 100  
>>>
```

# 巢狀式結構

```
for i in range(4):  
    for n in range(5):  
        if i+n==3:  
            print(i, n, 'YES')  
        else:  
            print(i, n)
```

```
>>>  
>>>  
>>> for i in range(4):  
...     for n in range(5):  
...         if i+n==3:  
...             print(i,n,'YES')  
...         else:  
...             print(i,n)  
...  
(0, 0)  
(0, 1)  
(0, 2)  
(0, 3, 'YES')  
(0, 4)  
(1, 0)  
(1, 1, 'YES')  
(1, 2, 'YES')  
(1, 3)  
(1, 4)  
(2, 0)  
(2, 1, 'YES')  
(2, 2)  
(2, 3)  
(2, 4)  
(3, 0, 'YES')  
(3, 1)  
(3, 2)  
(3, 3)  
(3, 4)  
>>>
```