

Chapter 9

FUNCTION

函數 (Function)

函數是一組可重複使用的程式碼，用來實現單一特定的功能

而函數和迴圈的概念很像，目的都是

重複執行部份的程式碼

但函數只有需要時才會被呼叫

函數 (Function)

回想一下之前學的 `print(“Hello Python!!!”)`, `input(“ 請輸入一個整數”)`
這二個都是 Python 內建的函數

Python 有內建許多好用的函數，例如之前使用的 `range()`, `index()`, `count()`...

但若想要設計自己想要的功能，就必須要自訂方法 !!!

函數 (Function)

自訂函數

使用 `def` 指令，而完整的函數是由函數名稱、參數以及函數指令組成

如果函數有傳回值，則需要使用 `return` 敘述傳回計算結果

```
def <函數名稱> ( 參數 ) :  
    <statement>  
    return [ 傳回值 ]
```

函數 (Function)

自訂函數

```
def < 函數名稱 > ( 參數 ) :  
    <statement>  
    return [ 傳回值 ]
```

函數內容以：開始，記得要**縮排縮排縮排**

return [傳回值] 結束函數，回傳值為選擇性的

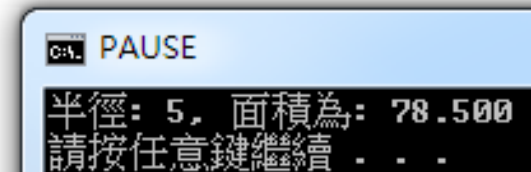
若不回傳任何資料，相當於回傳 None

函數 (Function)

```
def <函數名稱> ( 參數 ) :  
    <statement>  
    return [ 傳回值 ]
```

Ex 1

```
1 def area(r):  
2     x = r * r * 3.14  
3     return x  
4  
5 r = 5  
6 ans = area(r)  
7  
8 print("半徑: %d, 面積為: %.3f" % (r , ans) )  
9
```



PAUSE
半徑: 5, 面積為: 78.500
請按任意鍵繼續 . . .

說明	
函數名稱	area
收到的參數	r
目的	計算以 r 為半徑的圓面積
回傳	x

函數 (Function)

```
def <函數名稱> ( 參數 ) :  
    <statement>  
    return [ 傳回值 ]
```

Ex 1

```
def area(r):  
    x = r * r * 3.14  
    return x  
  
r = 5  
ans = area(r)  
print("1. 半徑: %d, 面積為: %.3f" % (r , ans) )  
print("")  
r = 10  
ans = area(r)  
print("2. 半徑: %d, 面積為: %.3f" % (r , ans) )  
print("")  
r = 23  
ans = area(r)  
print("3. 半徑: %d, 面積為: %.3f" % (r , ans) )
```

PAUSE

```
1. 半徑: 5, 面積為: 78.500  
2. 半徑: 10, 面積為: 314.000  
3. 半徑: 23, 面積為: 1661.060  
請按任意鍵繼續 . . .
```

函數 (Function)

```
def <函數名稱> ( 參數 ) :  
    <statement>  
    return [ 傳回值 ]
```

Ex 2

Function sum(n)

n 是輸入的值

此程式為 $1+2+\dots+n$

abc 是輸入的參數

tmp 接收 sum(abc)
傳回來的計算結果

```
def sum(n):  
    icy=0  
    for i in range(n+1):  
        icy = icy + i  
    return icy  
  
abc = eval(input("請輸入一個數字: "))  
tmp = sum(abc)  
print("1+2+3+...+%d=" % (abc) ,tmp)
```

```
請輸入一個數字: 10  
1+2+3+...+10= 55  
請按任意鍵繼續 . . .
```


函數 (Function)

```
def <函數名稱> ( 參數 ) :  
    <statement>  
    return [ 傳回值 ]
```

Ex 3

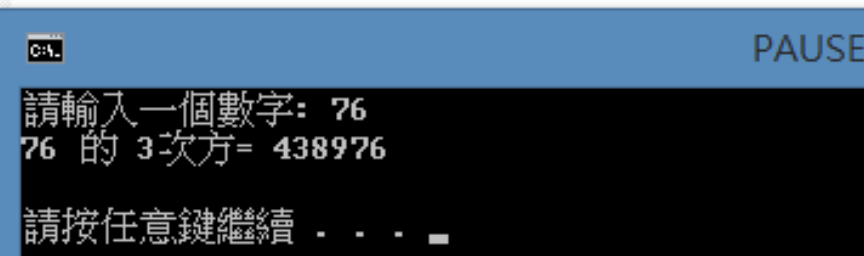
這隻程式會
回傳 x 的 3 次方

abc 是輸入的值

tmp 接收
power(abc)
傳回來的計算結果

```
def power ( x = 3 ) :  
    return x ** 3
```

```
abc = eval(input("請輸入一個數字: "))  
tmp = power(abc)  
print("%d 的 3次方=" % (abc) , tmp)
```



```
C:\> PAUSE  
請輸入一個數字: 76  
76 的 3次方= 438976  
請按任意鍵繼續 . . .
```

函數 (Function)

```
def <函數名稱> ( 參數 ) :  
    <statement>  
    return [ 傳回值 ]
```

Ex 4

n 是輸入的值

此程式會印出
n*n 的乘法表

abc 是輸入的值
multiTable(abc)
會印出 N*N 乘法表

```
def multiTable(n):  
    k=0  
    for i in range (1, n+1):  
        for j in range (1, n+1):  
            k=i*j  
            print("%d*%d=%d\t" % (i,j,k), end=" ")  
        else:  
            print("")  
    return  
  
abc = eval(input("請輸入一個數字: "))  
multiTable(abc)
```

C:\ PAUSE

請輸入一個數字: 8

1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40
6*1=6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36	6*7=42	6*8=48
7*1=7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42	7*7=49	7*8=56
8*1=8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48	8*7=56	8*8=64

請按任意鍵繼續 . . .

函數 (Function)

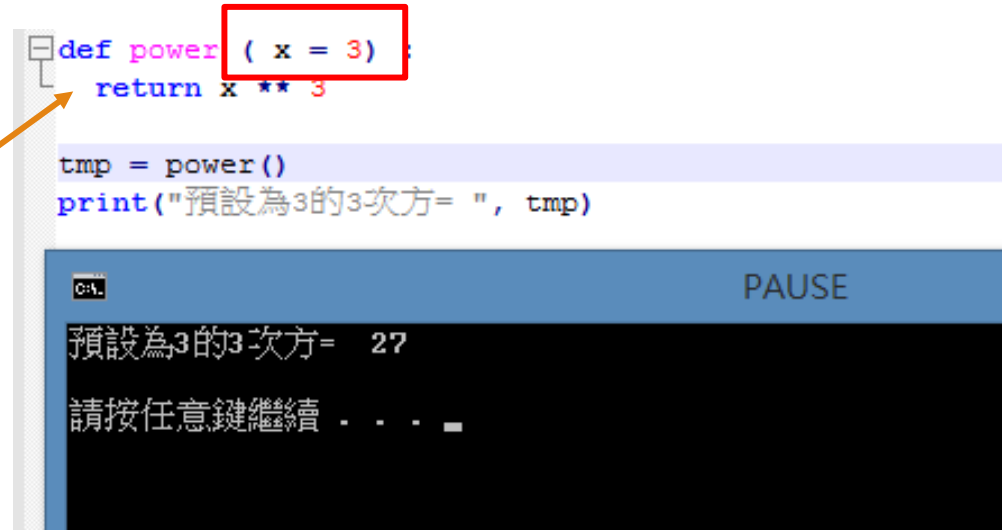
```
def <函數名稱> ( 參數 ) :  
    <statement>  
    return [ 傳回值 ]
```

Ex 5

這隻程式會
回傳 x 的 3 次方

沒有傳值進入函數

所以回傳的結果
為預設的 3 的 3 次方



```
def power ( x = 3 ) :  
    return x ** 3  
  
tmp = power()  
print("預設為3的3次方= ", tmp)
```

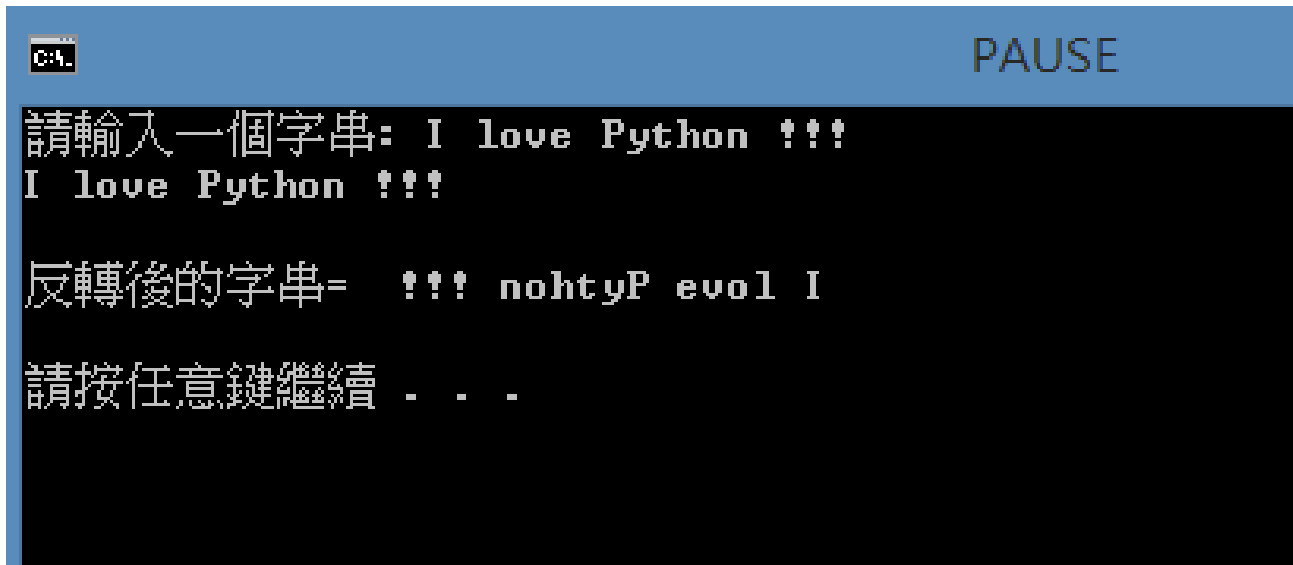
PAUSE

預設為3的3次方= 27

請按任意鍵繼續 . . .

隨堂練習

1. 請撰寫一個函式，輸入一個字串
2. 輸出是將這字串反轉



A screenshot of a Windows command prompt window. The title bar is blue and contains the text "C:\>" on the left and "PAUSE" on the right. The command prompt area has a black background with white text. The text displayed is as follows:

```
請輸入一個字串: I love Python !!!  
I love Python !!!  
  
反轉後的字串=   !!! nohtyP evol I  
  
請按任意鍵繼續 . . .
```

函數 (Function)

資料回傳可以超過一個數值（會組合成 tuple 資料）

```
def multi_para(r):
```

```
    x = r * r * 3.14
```

```
    y = r * r
```

```
    z = r * r * r
```

```
    return x, y, z
```

```
r = 5
```

```
ans = multi_para(r)
```

```
print(type(ans))
```

```
print(ans[0])
```

```
print(ans[1])
```

```
print(ans[2])
```

PAUSE

```
<class 'tuple'>
```

```
78.5
```

```
25
```

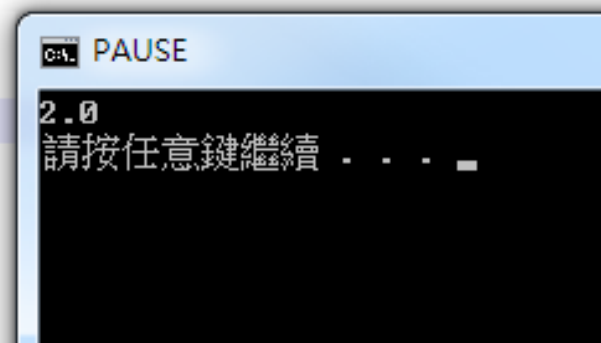
```
125
```

```
請按任意鍵繼續 . . .
```

函數 (Function)

關鍵字引數（可以不需要照順序排傳入參數）

```
def test(a,b):  
    c = a / b  
    return c  
  
ans = test(b=3, a=6)  
print(ans)
```



函數 (Function)

* 運算子 (可搭配 tuple 來接收多餘的參數)

```
def test(a,*b):  
    print("學生: ", a)  
    c = sum(b)  
    print("總分: ", c)  
    c = c/len(b)  
    print("平均成績: ", c)  
  
ans = test("Curry", 90, 90, 90, 100)
```

A screenshot of a terminal window with a blue title bar that says "PAUSE". The terminal has a black background with white text. It displays the output of the Python code: "學生: Curry", "總分: 370", "平均成績: 92.5", and "請按任意鍵繼續 . . .".

```
PAUSE  
學生: Curry  
總分: 370  
平均成績: 92.5  
請按任意鍵繼續 . . .
```

函數 (Function)

lambda 函數

沒有函數名稱，只會以一行敘述來表達其函數內容

```
1  
2 icy = lambda i, j : i ** j  
3 print(icy(5,3))  
4  
5  
6
```

PAUSE

125

請按任意鍵繼續 . . .

Any Questions !?