函式

- 自定函式
- 傳送參數給函式
- 從函式回傳值
- 全域變數與區域變數
- 範例集錦
- 預設參數值

7-1 自定函式

- 函式有兩種
  - 一為系統本身所提供的函式，如print()與input()函式，您可以直接直接呼叫使用
  - 二為使用者自訂（user-defined）的函式

- 假設要輸出以下的結果：


```
**********
Learning Python now!!!
**********
```

#p7-3.py

```
def printStar():
    for i in range(20):
        print('*', end=' ')
    print(' ')

def main():
    printStar()
    print('Learning Python now!!! ')
    printStar()

main()
```

當程式需要印出20個星星時，只要呼叫此函式即可，所以，撰寫一次印出星星的函式就可以了。

- 定義函式的語法如下：
  - def functionName(parameter-list):
  - 　　statement(s)
- 以def關鍵字為起啟點，接下來是使用者自訂的函式名稱，如上一範例程式的printStar()。
- 最後要加上冒號（:）。

- 我們也定義了一函式名稱main()，此函式呼叫printStar()、print()，最後再呼叫printStar() 等三個函式。
- 程式的最後呼叫main()加以啟動。

if __name__ == '__main__':

- Python 程式裡的 __name__ 可以用來分辨程式是直接執行還是被 import 的
- 意思就是說**讓你寫的腳本模塊既可以導入到別的模塊中用，另外該模塊自己也可執行**。
- 所以用 __name__ 就可以分辨我的程式是被 import 當成模組還是被直接執行的。這樣附帶的好處就是如果我寫的程式平常可以被 import 來使用，但有時它自己也可以直接執行。

在九九乘法表的上、下加上星星

****

1*1= 1	2*1= 2	3*1= 3	4*1= 4	5*1= 5	6*1= 6	7*1= 7	8*1= 8	9*1= 9
1*2= 2	2*2= 4	3*2= 6	4*2= 8	5*2=10	6*2=12	7*2=14	8*2=16	9*2=18
1*3= 3	2*3= 6	3*3= 9	4*3=12	5*3=15	6*3=18	7*3=21	8*3=24	9*3=27
1*4= 4	2*4= 8	3*4=12	4*4=16	5*4=20	6*4=24	7*4=28	8*4=32	9*4=36
1*5= 5	2*5=10	3*5=15	4*5=20	5*5=25	6*5=30	7*5=35	8*5=40	9*5=45
1*6= 6	2*6=12	3*6=18	4*6=24	5*6=30	6*6=36	7*6=42	8*6=48	9*6=54
1*7= 7	2*7=14	3*7=21	4*7=28	5*7=35	6*7=42	7*7=49	8*7=56	9*7=63
1*8= 8	2*8=16	3*8=24	4*8=32	5*8=40	6*8=48	7*8=56	8*8=64	9*8=72
1*9= 9	2*9=18	3*9=27	4*9=36	5*9=45	6*9=54	7*9=63	8*9=72	9*9=81

****

#p7-6.py

```
def printStar():
    for i in range(72):
        print('*', end = ' ')
    print(' ')


def multiply():
    for i in range(1, 10):
        for j in range(1, 10):
            print('%d*%d=%2d '%(j, i, i*j), end = ' ')
        print(' ')
```

```
def main():
    printStar()
    multiply()
    printStar()


main()
```

7-2　傳送參數給函式

- 先印出20個星星，最後再列印30個星星

#p7-7.py

```
def printStar(n):
    for i in range(1, n+1):
        print('*', end=' ')
    print(' ')

def main():
    printStar(20)
    print('Learning Python now!!! ')
    printStar(30)

main()
```

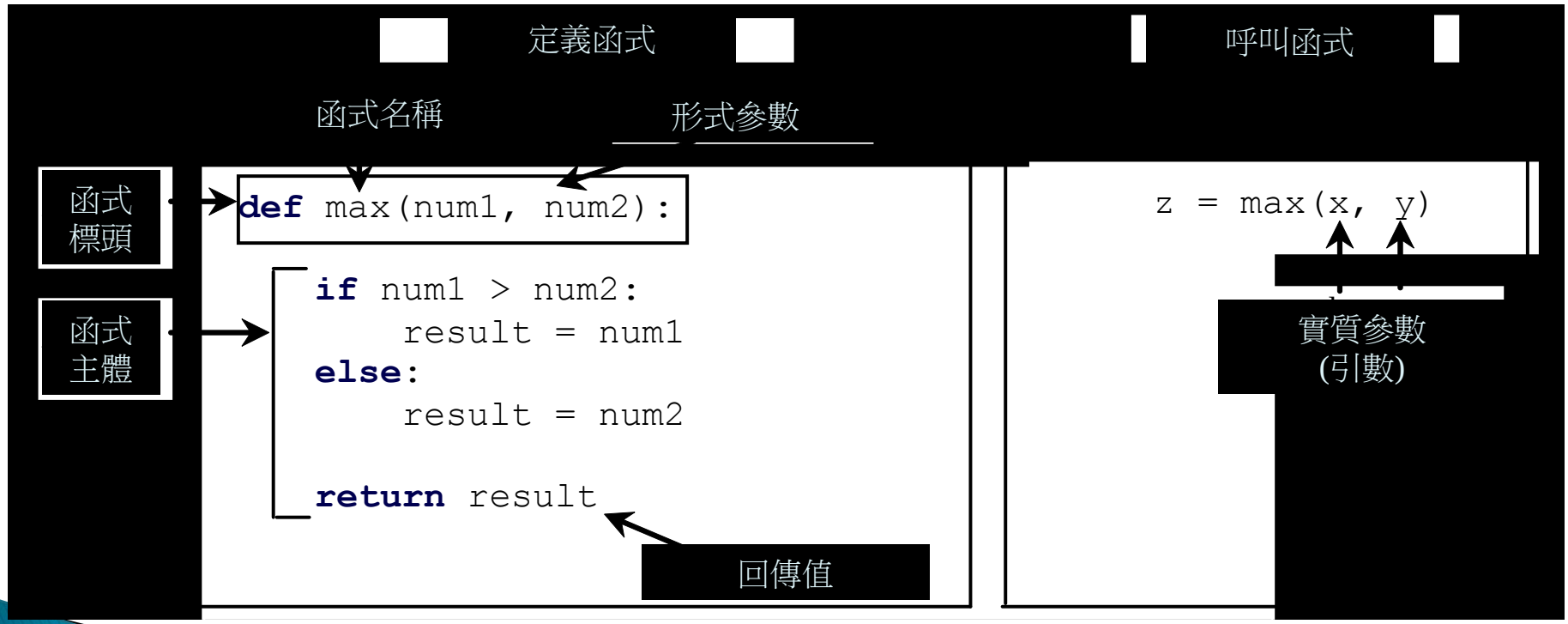
輸出結果

```
**********
Learning Python now!!!")
**********
```

- 程式中printStar(n)函式中的n，接收由main()函式第一次呼叫printStar(20)傳送來的20，第二次呼叫printStar(30)傳送來的30。
- 20或30我們稱之為實際參數（actual parameter），而n稱之為形式參數（formal parameter）。
- 這表示n是由20或30所給的。

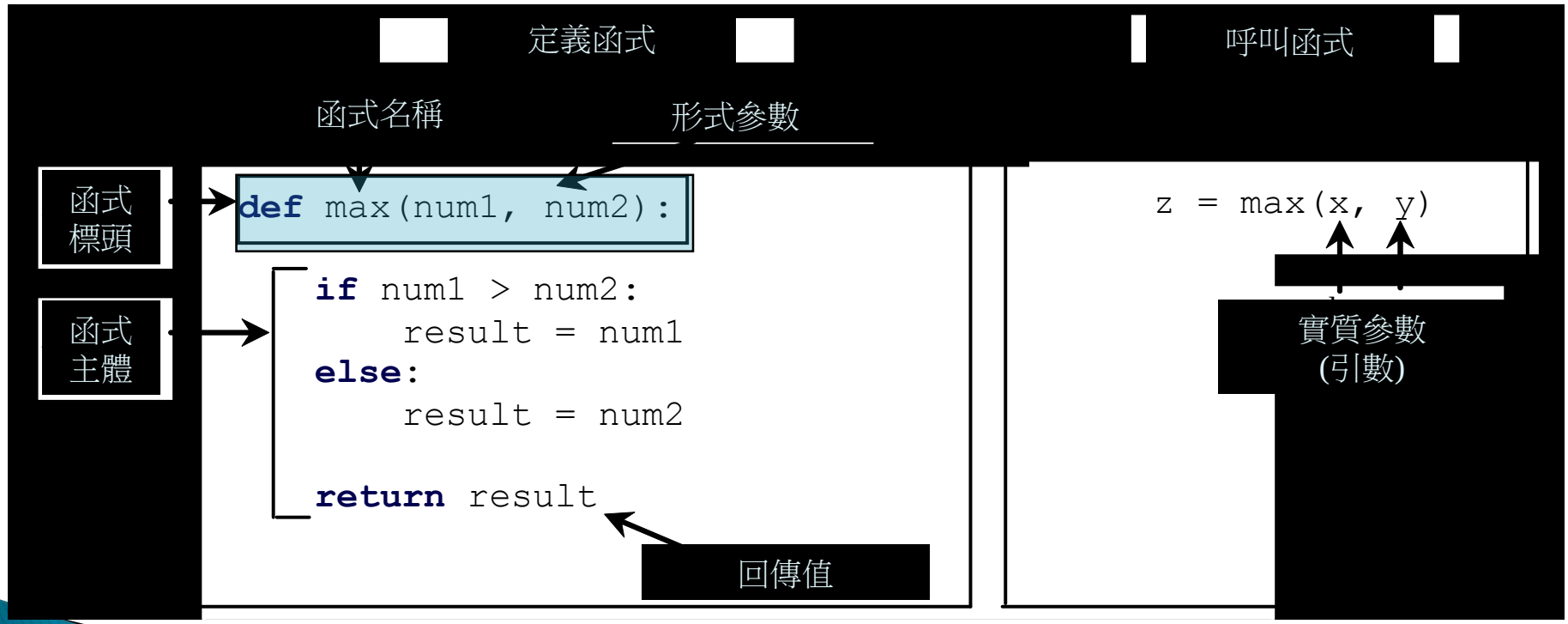
定義函式

函式定義由函式名稱、參數，以及主體內容所組成。



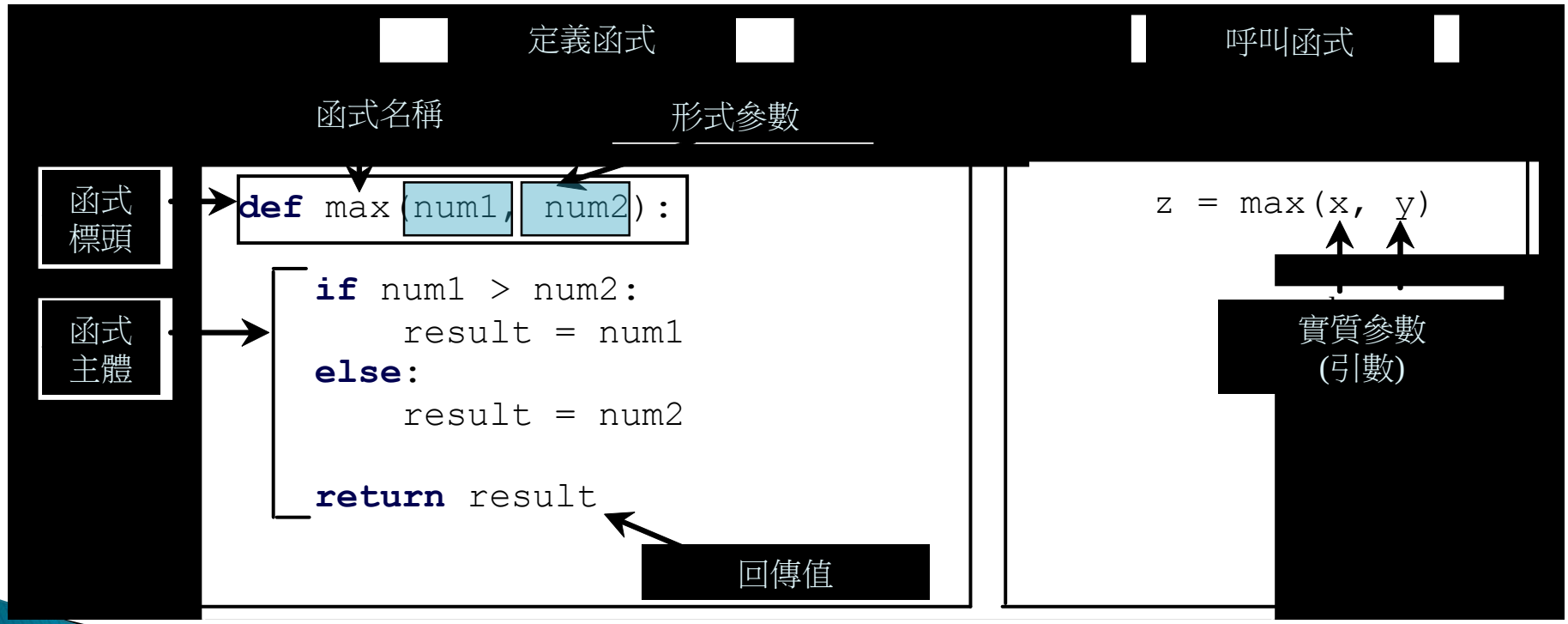
函式標頭

函式包含標頭和主體。標頭(header)起源於def關鍵字，後接函式名稱和參數，最後以冒號結尾。



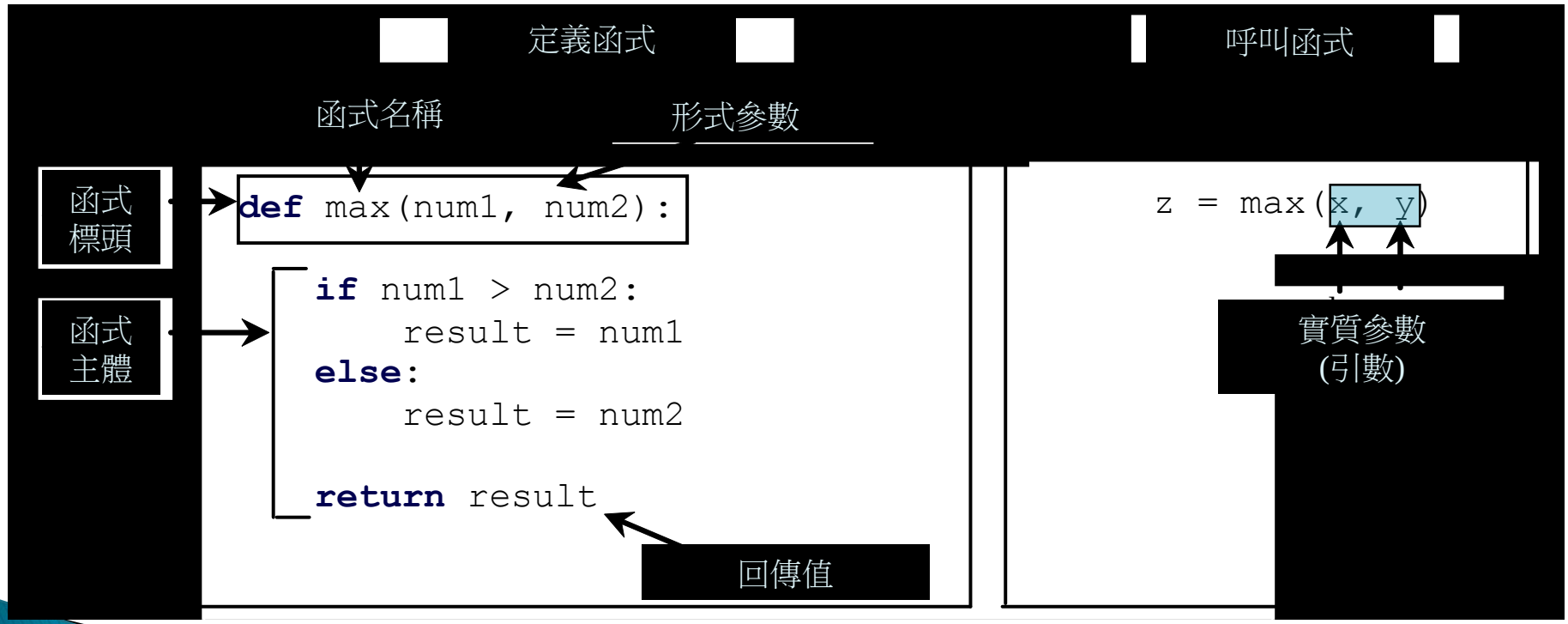
形式參數

定義於函式標頭的變數則稱為形式參數。



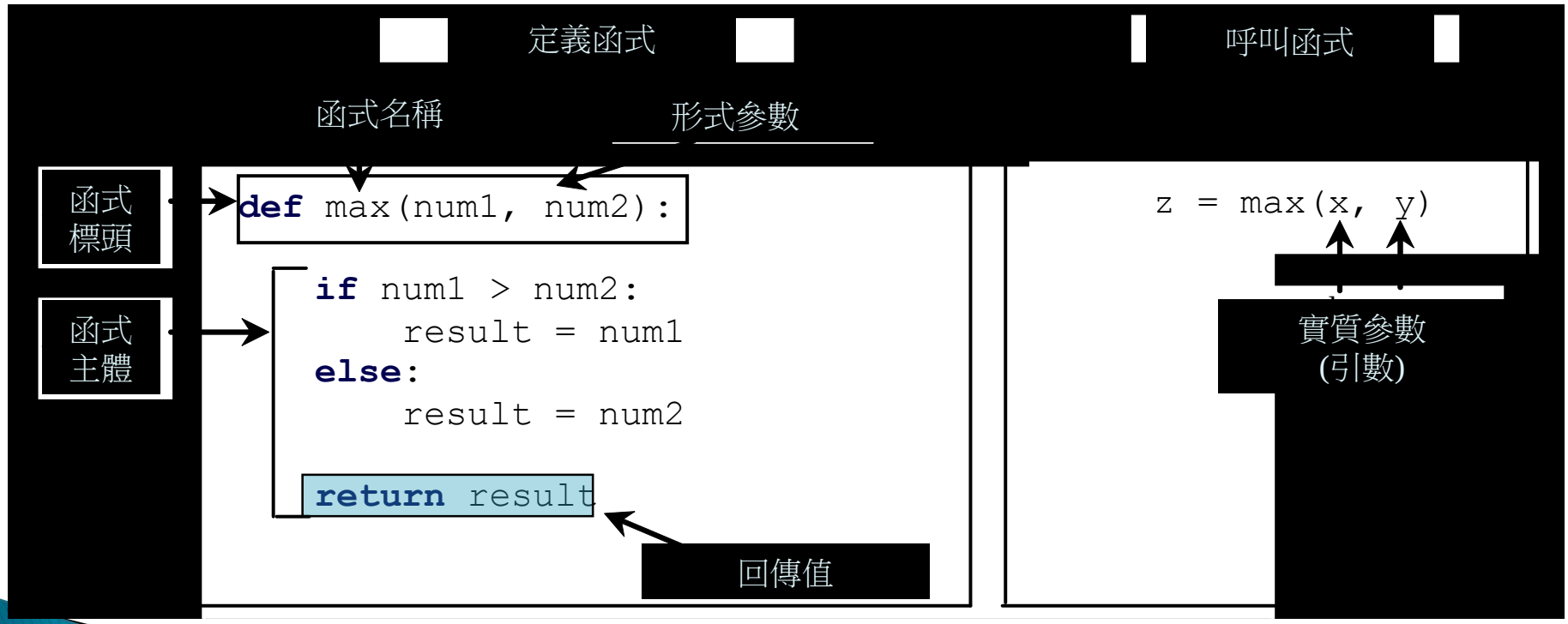
實質參數(引數)

當函式被呼叫時，你必須給予的參數則被稱為實質參數(引數)。



回傳值

函式會使用關鍵字return 回傳一個結果數值，此數值被稱為回傳值。



#p7-8.py

```
def printStar(n):
    for i in range(1, n+1):
        print('*', end=' ')
    print('')

def main():
    for k in range(1, 11):
        printStar(k)

main()
```

使用變數當做實際參數。

輸出結果

```
*
**
***
****
*****
******
*******
********
*********
**********
```

程式解析

- 程式列印十列，第一列有一個星星，第二列有兩個星星，第三列有三個星星，依此類推，在第十列有十個星星。在main()函式中呼叫printStar(k)，其中的k是實際參數。

比較
P7-7.py 與
P7-8.py

#p7-9.py

```
def printNumber(n):
    for i in range(1, n+1):
        print(i, end=' ')
    print("")

def main():
    for k in range(1, 10):
        printNumber(k)

main()
```

```
1
12
123
1234
12345
123456
1234567
12345678
123456789
```

比較
P7-9.py 與
P7-8.py

7-3 從函式回傳值

- 撰寫一程式，提示使用者輸入起始值和終止值，然後從起始值開始，每次累加1，直到終止值。

#p7-11.py

```
def sum(f, end):
    total = 0
    for k in range(f, end+1):
        total += k
    return total


def main():
    f  = eval(input('Enter from number: '))
    end = eval(input('Enter end number: '))


    tot = sum(f, end)
    print('%d + %d + ... + %d = %d'%(f, f+1, end, tot))


main()
```

輸出結果 1

```
Enter from number: 1
Enter end number: 100
1 + 2 + ... + 100 = 5050
```

輸出結果 2

```
Enter from number: 2
Enter end number: 100
2 + 3 + ... + 100 = 5049
```

輸出結果 3

```
Enter from number: 1
Enter end number: 99
1 + 2 + ... + 99 = 4950
```

- 程式中的sum函式，接收了兩個參數，分別是f與end。
- 之後利用for 迴圈計算從f到end的總和。
- 注意，在range中要將將end+1。最後，利用return將total的值回傳。
- 當main函式呼叫sum函式時，其total的回傳值指定給tot。所以tot就是從f到end的加總。

7-4 全域變數與區域變數

- 定義在函式外面的變數稱為全域變數（global variable），而定義函式內部的稱之為區域變數（local variable）。
- 函式會先使用區域變數，若沒有區域變數才會使用全域變數。

#p7-14.py

```
a = 100

def main():
    a = 200
    print('a = ', a)

main()
print('a = ', a)
```

p7-14.py
p7-14-1.py

輸出結果

```
a = 200
a = 100
```

- 由於main() 函式有定義區域變數a，所以在其裡面的print()函式會使用區域變數的a，而在main()函式外面的print()函式會使用全域變數的a。

#p7-15.py

```
a = 100

def main():
    global a
    a = 200
    print('a = ', a)

main()
print('a = ', a)
```

在main() 函式中使用全域變數

輸出結果

```
a =  200
a =  200
```

7-5　範例集錦

7-5-1　求兩數的GCD

- 兩個整數的最大公因數（Great Common divisor, GCD），表示可以整除這兩整數的最大的整數。

#p7-16.py

```
def gcdFunction(n1, n2):
    gc = 1
    k = 2
    while k <= n1 and k <= n2:
        if n1 % k == 0 and n2 % k == 0:
            gcd = k
        k += 1
    return gcd
```

```
def main():
    number1 = eval(input('Please input an integer: '))
    number2 = eval(input('Please input an integer: '))
    answer = gcdFunction(number1, number2)
    print('The GCD for', number1, 'and', number2, 'is', answer)


main()
```

輸出結果

```
Please input an integer: 24
Please input an integer: 32
The GCD for 24 and 32 is 8
```

7-5-2　判斷是否為質數

- 一整數若只被1和本身整除，則稱此數為質數（prime number）。

#p7-17.py

```
def primeFunction(n):
    divisor = 2
    flag = 1
    while divisor < n:
        if n % divisor == 0:
            flag = 0
            break
        divisor += 1
    return flag
```

比較p6-35.py

```
def main():
    number = eval(input('Please input an integer: '))
    result = primeFunction(number)
    if result == 0:
        print(number, 'is not prime number')
    else:
        print(number, 'is prime number')
```

```
main()
```

輸出結果 1

```
Please input an integer: 17
17 is prime number
```

輸出結果 2

```
Please input an integer: 22
22 is prime number
```

7-5-3　計算BMI

- 依據體重和身高計算BMI。其中身高是以公尺為單位，而體重是以公斤為單位。

#p7-18.py

```
def bmiFunction(w, h):
    hInCentimeter = (h/100)


    bmi = w / (hInCentimeter * hInCentimeter)
    print('Your BMI is', format(bmi, '.2f'))

    if bmi < 18.5:
        print('Underweight')
    elif bmi < 25:
        print('Normal')
    elif bmi < 30:
        print('Overweight')
```

比較p5-21.py

```
def main():
    weight = eval(input('Please input your weight(kilogram): '))
    height = eval(input('Please input your height(centimeter): '))
    bmiFunction(weight, height)


main()
```

輸出結果

```
Please input your weight(kilogram): 67
Please input your height(centimeter): 175
Your BMI is 21.88
Normal
```

7-5-4　計算GPA

```
#p7-20.py
def gpaFunction(score):
    if score >= 80:
        print('Grade A')
    elif score >= 70:
        print('Grade B')
    elif score >= 60:
        print('Grade C')
    elif score >= 50:
        print('Grade D')
    else:
```

比較p5-18.py

```
def main():
    score = eval(input('Please input your score: '))
    gpaFunction(score)


main()
```

輸出結果 1

```
Please input your score: 86
Grade A
```

輸出結果 2

```
Please input your score: 67
Grade C
```

7-6-3 兩數對調

```
#p7-24.py
def main():
    a = eval(input("Please input a: "))
    b = eval(input("Please input b: "))
    print("a = ", a, "b = ", b)


    # swap two numbers
    temp = a
    a = b
    b = temp


    print("a = %d, b = %d" %(a, b))
```

輸出結果

```
Please input a: 100
Please input b: 200
a =  100 b =  200
a =  200 b =  100
```

題目

- 將兩數的對調撰寫為一函式，然後加以回傳給呼叫者。

#p7-25.py

```
def swapNumbers(x, y):
    x, y = y, x
    return x, y


def main():
    a = eval(input("Please input a: "))
    b = eval(input("Please input b: "))
    print("a = ", a, "b = ", b)


    a, b = swapNumbers(a, b)
    print("a = ", a, "b = ", b)
```

輸出結果

```
Please input a: 100
Please input b: 200
a =  100 b =  200
a =  200 b =  100
```

7-7 預設參數值

- 在函式參數方面，Python提供預設參數值（default argument value）的方法，亦即當呼叫函式時，若沒有給予參數值，此時會取預設參數值來執行。
- 若呼叫函式時，有給予參數值，則預設參數值會被忽略。

#p7-26.py

```
def sum(begin, end=100):
    sum = 0
    for k in range(begin, end+1):
        sum += k
    return sum
```

```
def main():
    total = sum(1)
    print('The number from', 1, 'to', 100)
    print('Summation is', total)

    total = sum(1, 10)
    print("\nThe number from", 1, "to", 10)
    print("Summation is", total)

main()
```

輸出結果

```
The number from 1 to 100
Summation is 5050

The number from 1 to 10
Summation is 55
```

#p7-27.py

```
def sum(begin=1, end=100):
    sum = 0
    for k in range(begin, end+1):
        sum += k
    return sum


def main():
    total = sum()
    print('The number from', 1, 'to', 100)
    print("Summation is", total)
```

可以將函式的參數皆設為有預設值，如此一來，當呼叫sum函式時，給予一個或不給予參數值皆是正確的。

```
total = sum(2)
    print('\nThe number from', 2, 'to', 100)
    print('Summation is', total)

    total = sum(2, 10)
    print('\nThe number from', 2, 'to', 10)
    print('Summation is', total)

main()
```

輸出結果

```
The number from 1 to 100
Summation is 5050

The number from 2 to 100
Summation is 5049
```

```
The number from 2 to 10
Summation is 54
```

題目

- 將上述九九乘法表印出的星星數以預設參數值加以表示。其預設值為72。

#p7-29.py

```
def printStar(starNumbers = 72):
    for i in range(starNumbers):
        print('*', end = '')
    print("")


def multiply():
    for i in range(1, 10):
        for j in range(1, 10):
            print('%d*%d=%2d '%(j, i, i*j), end = ' ')
        print('')
```

```
def main():
    printStar()
    multiply()
    printStar(60)

main()
```

輸出結果

****

1*1= 1 2*1= 2 3*1= 3 4*1= 4 5*1= 5 6*1= 6 7*1= 7 8*1= 8 9*1= 9

1*2= 2 2*2= 4 3*2= 6 4*2= 8 5*2=10 6*2=12 7*2=14 8*2=16 9*2=18

1*3= 3 2*3= 6 3*3= 9 4*3=12 5*3=15 6*3=18 7*3=21 8*3=24 9*3=27

1*4= 4 2*4= 8 3*4=12 4*4=16 5*4=20 6*4=24 7*4=28 8*4=32 9*4=36

1*5= 5 2*5=10 3*5=15 4*5=20 5*5=25 6*5=30 7*5=35 8*5=40 9*5=45

1*6= 6 2*6=12 3*6=18 4*6=24 5*6=30 6*6=36 7*6=42 8*6=48 9*6=54

1*7= 7 2*7=14 3*7=21 4*7=28 5*7=35 6*7=42 7*7=49 8*7=56 9*7=63

1*8= 8 2*8=16 3*8=24 4*8=32 5*8=40 6*8=48 7*8=56 8*8=64 9*8=72

1*9= 9 2*9=18 3*9=27 4*9=36 5*9=45 6*9=54 7*9=63 8*9=72 9*9=81

****

Ex6-1題目：顯示反轉的整數

- (顯示反轉的整數)請撰寫一個帶有以下標頭的函式,反向顯示一個整數:

def reverse(number)

比方說,reverse(3456)會顯示 6543。

請撰寫一測試程序，提示user輸入一個整數並顯示其反轉。

ex6-4.py