03迴圈 (loop statement)

迴圈主要目的用於程式中一些敘述需要重複執行。迴圈敘述的三個主要條件為：初值設定、條件運算式、控制運算式

3.1 while 敘述

while的迴圈敘述如下：

初值設定

while條件運算式：

主體敘述

(**控制條件**)

與if條件分支的結構相似，迴圈的主體敘述要向內縮排，慣例縮排4格，只要是在條件運算式後縮排的敘述都被視為迴圈區塊範圍內，當回到同一排列位置時，表示已經脫離迴圈執行的區塊範圍。

範例：由1顯示到5

i = 1

while i < 6:  #表示i 小於 6時迴圈會執行

  print(i)

  i += 1

輸出結果

1

2

3

4

5

範例：由1 加到 100的總和：

設定一個用來存放加總數的變數 total 、 迴圈初始值設定 i

total = 0

i = 1           #設定迴圈運算初始值

while i <= 100: #迴圈條件運算式

    total += i  #迴圈主體敘述

    i += 1      #迴圈控制式 (在迴圈主體敘述內)

print ('total =',total)

輸出結果

total = 5050

注意條件運算式後面要加 ： ，初始值設定 i =1 ；條件運算式用來判斷 i <=100是否為真；控制運算式是 i += 1 ，在這裡用來控制迴圈運轉次數，迴圈主體敘述total +=i和控制式都向內縮排，屬於while迴圈的內容，若是將i+=1放到迴圈之外，無法進行i的計算，就會成為**無窮迴圈**

total = 0

i = 1           #設定迴圈運算初始值

while i <= 100: #迴圈條件運算式

    total += i  #迴圈主體敘述

i += 1      #迴圈控制式 (執行不到這個控制式，while迴圈跳不出來)

print ('total =',total)

輸出結果

無窮迴圈，無法跳出結果

上面三個要素：初始值、條件運算式、控制式三個項目任何一個改變，都可能造成迴圈執行結果變化，例如將i = 2 ，結果就是由2加到100。若是將條件運算式改為 i < 100 就是 1加到 99，若是將條件式改為 i += 2，就是i到100的奇數和。

total = 0

i = 2           #設定迴圈運算初始值

while i <= 100: #迴圈條件運算式

    total += i  #迴圈主體敘述

    i += 1      #迴圈控制式 (在迴圈主體敘述內)

print ('total =',total)

輸出結果

total = 5049

total = 0

i = 1           #設定迴圈運算初始值

while i < 100: #迴圈條件運算式

    total += i  #迴圈主體敘述

    i += 1      #迴圈控制式 (在迴圈主體敘述內)

print ('total =',total)

輸出結果

total = 4950

total = 0

i = 1           #設定迴圈運算初始值

while i <= 100: #迴圈條件運算式

    total += i  #迴圈主體敘述

    i += 2      #迴圈控制式 (在迴圈主體敘述內)

print ('total =',total)

輸出結果

total = 2500

3.2 for...in range 迴圈

這是python提供的第二種迴圈，其語法如下：

for i in range (**start**, **end,** step):

語法中的 i(變數) 表示從start 開始 到 end -1 結束，中間過程每次加 step的值。

Start內定值為0，若初始值為0，則可省略start；step內定值為1，若每次迴圈加1，可省略step。

要將for...in range對應while的話，上述語法相當於：

i = start

while i <= end-1

... #主體敘述

i += step

for x in range(6):

  print(x)

輸出結果

0

1

2

3

4

5

for x in range(2, 6):

  print(x)

輸出結果

2

3

4

5

for x in range(2, 30, 3):

  print(x)

輸出結果

2

5

8

11

14

17

20

23

26

29

若將上述1加到100的程式重寫一次，結果如下：

total = 0

for i in range(1,101,1):  #迴圈初始值為1 結束值為101-1 每次加1(可省略)

    total += i            #迴圈主體敘述

print ('total =',total)

輸出結果

total = 5050

比照上面敘述，改為由2加到100，結果如下：

total = 0

for i in range(2,101,1):  #迴圈初始值為2 結束值為101-1 每次加1(可省略)

    total += i            #迴圈主體敘述

print ('total =',total)

輸出結果

total = 5049

若改為1加到99，結果如下：

total = 0

for i in range(1,100,1):  #迴圈初始值為1 結束值為100-1 每次加1(可省略)

    total += i            #迴圈主體敘述

print ('total =',total)

輸出結果

total = 5049

若改為1到100奇數和，結果如下：

total = 0

for i in range(1,101,2):  #迴圈初始值為1 結束值為101-1 每次加1(可省略)

    total += i            #迴圈主體敘述

print ('total =',total)

輸出結果

total = 2500

3.3 巢狀迴圈(nested loop)

巢狀迴圈又稱多重迴圈，顧名思義為迴圈內又一迴圈，表示外迴圈內有內迴圈，參考以下範例：

# 巢狀迴圈(nested loop)

print('=====')

for x in range(1,6):

    print('x = %d'%x)

    for y in range(1,6):

        print ('    y = %d'%y)

    print('=====')

輸出結果：

=====

x = 1

y = 1

y = 2

y = 3

y = 4

y = 5

=====

x = 2

y = 1

y = 2

y = 3

y = 4

y = 5

=====

x = 3

y = 1

y = 2

y = 3

y = 4

y = 5

=====

x = 4

y = 1

y = 2

y = 3

y = 4

y = 5

=====

x = 5

y = 1

y = 2

y = 3

y = 4

y = 5

=====

程式中外圈變數 x =1的時候，內圈變數y會執行1到5。從輸出結果可以很清楚看見，x = 2時，y會從頭再執行一次1 到5；如此反覆執行，直到x > 5停止。

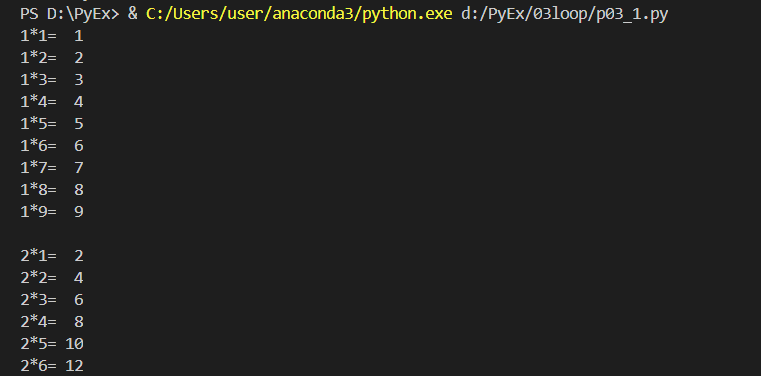
有上述概念後，可以寫一支印出99乘法表的程式。

for x in range(1,10):ˊ

    for y in range(1,10):

        print(f'{x}\*{y} = {x\*y:2}')

輸出結果：(片段)



這個程式要點在於這一行：

print ('%d\*%d= %2d'%(x,y,x\*y))

透過格式控制調整輸出結果，但這輸出結果缺點是每次都會跳行，佔據版面。若想不讓它跳行，只要在這敘述括弧內，加上 , end =' ' 即可，如下面範例所示：

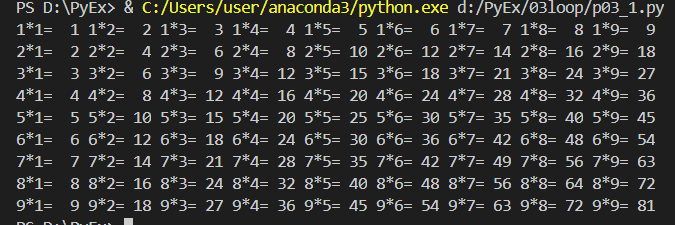
for x in range(1,10):

    for y in range(1,10):

        print ('%d\*%d= %2d'%(x,y,x\*y),end = ' ')

    print()

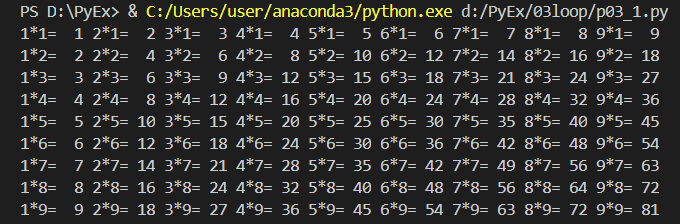
輸出結果：



這樣排版看起來好多了，但實際上九九乘法表排列順序並不是這樣，所以換個列印排版方式：

 print ('%d\*%d= %2d'%(y,x,x\*y),end = ' ')

輸出結果：



練習題

1. 使用迴圈敘述撰寫程式，讓使用者輸入兩個正整數 a、b (a<b) ，利用迴圈計算從a開始連加到b的總和。例如：輸入a=1、b=100 (1+2+3+…+100)，則輸出結果為5050
   1. 輸入說明： 請輸入兩個正整數(a、b且a<b) 輸入範例

請輸入第一個正整數66

請輸入第二個正整數666

* 1. 輸出說明： 計算從 a 開始到 b的總和 輸出範例

219966

1. 使用迴圈敘述撰寫程式，讓使用者輸入兩個正整數 a、b (a<b) ，利用迴圈計算從a開始連加到b的**偶數**總和。例如：輸入a=1、b=100 (2+4+6+…+100)，則輸出結果為2550
   1. 輸入說明： 請輸入兩個正整數(a、b且a<b) 輸入範例

請輸入第一個正整數6

請輸入第二個正整數88

* 1. 輸出說明： 計算從 a 開始到 b的偶數總和 輸出範例

1974

1. 使用迴圈敘述撰寫程式，讓使用者輸入一個正整數 (<30)然後以三角形方式依序輸出此數階乘結果
   1. 輸入說明： 請輸入一個正整數(<30) 輸入範例

12

* 1. 輸出說明： 以三角形方式依序輸出此數階乘結果 輸出範例

1

2 4

3 6 9

4 8 12 16

5 10 15 20 25

6 12 18 24 30 36

7 14 21 28 35 42 49

8 16 24 32 40 48 56 64

9 18 27 36 45 54 63 72 81

10 20 30 40 50 60 70 80 90 100

11 22 33 44 55 66 77 88 99 110 121

12 24 36 48 60 72 84 96 108 120 132 144

1. 使用迴圈敘述撰寫程式，讓使用者輸入一個正整數a，利用迴圈計算從1到a之間(含)所有5的倍數和。
   1. 輸入說明： 請輸入一個正整數 輸入範例

500

* 1. 輸出說明： 所有5的倍數和 輸出範例

25250

1. 請使用迴圈敘述撰寫程式，讓使用者輸入一個正整數，將此數反轉順序輸出 (利用 % 與 // 處理)
   1. 輸入說明： 請輸入一個正整數 輸入範例

31283

* 1. 輸出說明： 將數值反順序輸出 輸出範例

38213

1. 請使用迴圈敘述撰寫程式，讓使用者輸入一個正整數n，利用迴圈計算n!的值。
   1. 輸入說明： 請輸入一個正整數 輸入範例

15

* 1. 輸出說明： 計算n！值 輸出範例

1307674368000

1. (1) 請使用迴圈敘述撰寫程式，讓使用者輸入一個正整數n (n < 10)，顯示 n\*n乘法表。

(2)每項運算式格式化排列整齊，每個運算子與運算元輸出的欄寬為2，而每項乘積輸出的欄寬為4，皆靠左對齊不跳行。

* 1. 輸入說明： 請輸入一個正整數n (n<10)

3

* 1. 輸出說明： 輸出格式化n \* n乘法表

1\* 1 = 1 2\* 1 = 2 3\* 1 = 3

1\* 2 = 2 2\* 2 = 4 3\* 2 = 6

1\* 3 = 3 2\* 3 = 6 3\* 3 = 9

1. 請使用迴圈敘述撰寫程式，讓使用者輸入一個正整數，代表後面要測試的次數。 每次測試都是輸入一個正整數，將所有位數加起來輸出結果
   1. 輸入說明： 請輸入一個正整數，代表要測試數字的”次數”。

每次輸入一個兩位數以上資料。

* 1. 輸出說明： 看所有位數加總和
  2. 若輸入大於或等於10，顯示”輸入錯誤！數字不能大於或等於10”

輸出結果：

請輸入測試次數 2

請輸入正整數 32145

32145 數字拆開後相加為 15

請輸入正整數 654789

654789 數字拆開後相加為 39

1. 使用者輸入一筆存款金額、年利率及經過的月份，並顯示每個月存款總額。輸出到小數點第二位數。

例：存款10000 年利率5.75%

滿一個月，存款為：10000+10000\*5.75/1200=10047.92

滿二個月，存款為：10047.92+10047.92\*5.75/1200=10096.06

滿三個月，存款為：10096.06+10096.06\*5.75/1200=10144.44 餘下類推

輸出結果：

請輸入本金50000

請輸入利率3

請輸入月份5

Month Amout

1 50125.00

2 50250.31

3 50375.94

4 50501.88

5 50628.13