

Chapter 6

[LOOP 迴圈]

縮排

縮排在寫程式時是一件重要的事

因為 Python 是直譯的程式，所以一定要有縮排的習慣

如何縮排 很簡單

1. 按下鍵盤的 Tab 鍵
2. 按下鍵盤的 空白鍵

TAB

|
->|

迴圈

什麼是迴圈？

當我們要重複執行相同的動作，就需要使用迴圈跟電腦說它該作的事

ex. 當你去 7-11 買 Java 咖哩飯的時候，店員會

1. 結帳並找錢
2. 放入微波爐加熱
3. 準備提袋，並放入餐具
4. 把咖哩飯從微波爐拿出來
5. 打包所有的東西給客人

迴圈

什麼是迴圈？

如果沒有重複的動作寫成迴圈，那會讓程式變的非常的亂

ex, 有人一次買了 10 碗 Java 咖哩飯…

1. 結帳並找錢
2. 放入微波爐加熱
3. 準備提袋，並放入餐具
4. 把咖哩飯從微波爐拿出來
5. 打包所有的東西給客人

…1-5 的動作就要 copy and paste 一樣的程式碼 10 次…

迴圈

(前置作業) 比較運算子 :

運算子

==

<(=)

>(=)

!=

is

in

function

二個運算元是否相等

二個運算元是否小於 (等於)

二個運算元是否大於 (等於)

二個運算元是否不相等

可判斷兩個元素是否為相同物件

判斷一個元素是否為一集合的元素

迴圈

(前置作業) in 運算子 :

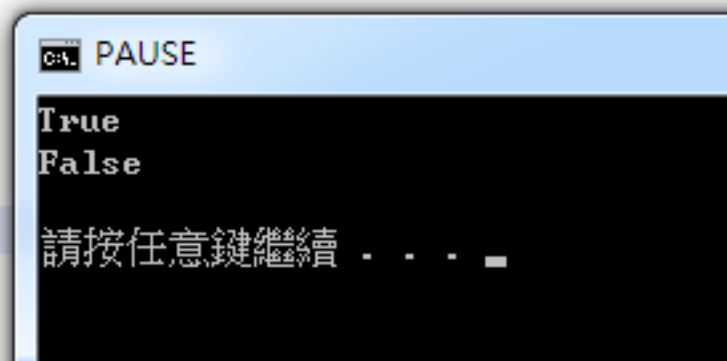
運算子
in

function

判斷一個元素是否為一集合的元素

```
x = [1, 2, 3, 4, 5]
y = 3
z = 6

print ( y in x )
print ( z in x )
```



迴圈

(前置作業) range() 函數 :

在介紹迴圈之前 , 先介紹一個很好用的 function, range()

function	meaning
range([start], stop, [step])	建立整數串列

start: 起始值 , 預設是 0

step: 計數器 , 預設是 1 , 可以自行設定疊代數字

stop: 結束值 , 就是 range() 結束的條件

迴圈

range() 函數：

通常 range() 可以轉型成其他容器的資料型態，如 list、tuple 等

```
a = range(10)

print(a, "\n")
# for a in range(10):
#     print(a)

print(list(range(15)))
print(list(range(1, 10, 2)))
print("\n")
print(tuple(range(15)))
print(tuple(range(10, 1, -2)))
```

PAUSE

range(0, 10)

[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14]

[1, 3, 5, 7, 9]

<0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14>

<10, 8, 6, 4, 2>

請按任意鍵繼續 . . .

迴圈

for... in ... :

for 迴圈是程式語言中最常用也最直觀的迴圈

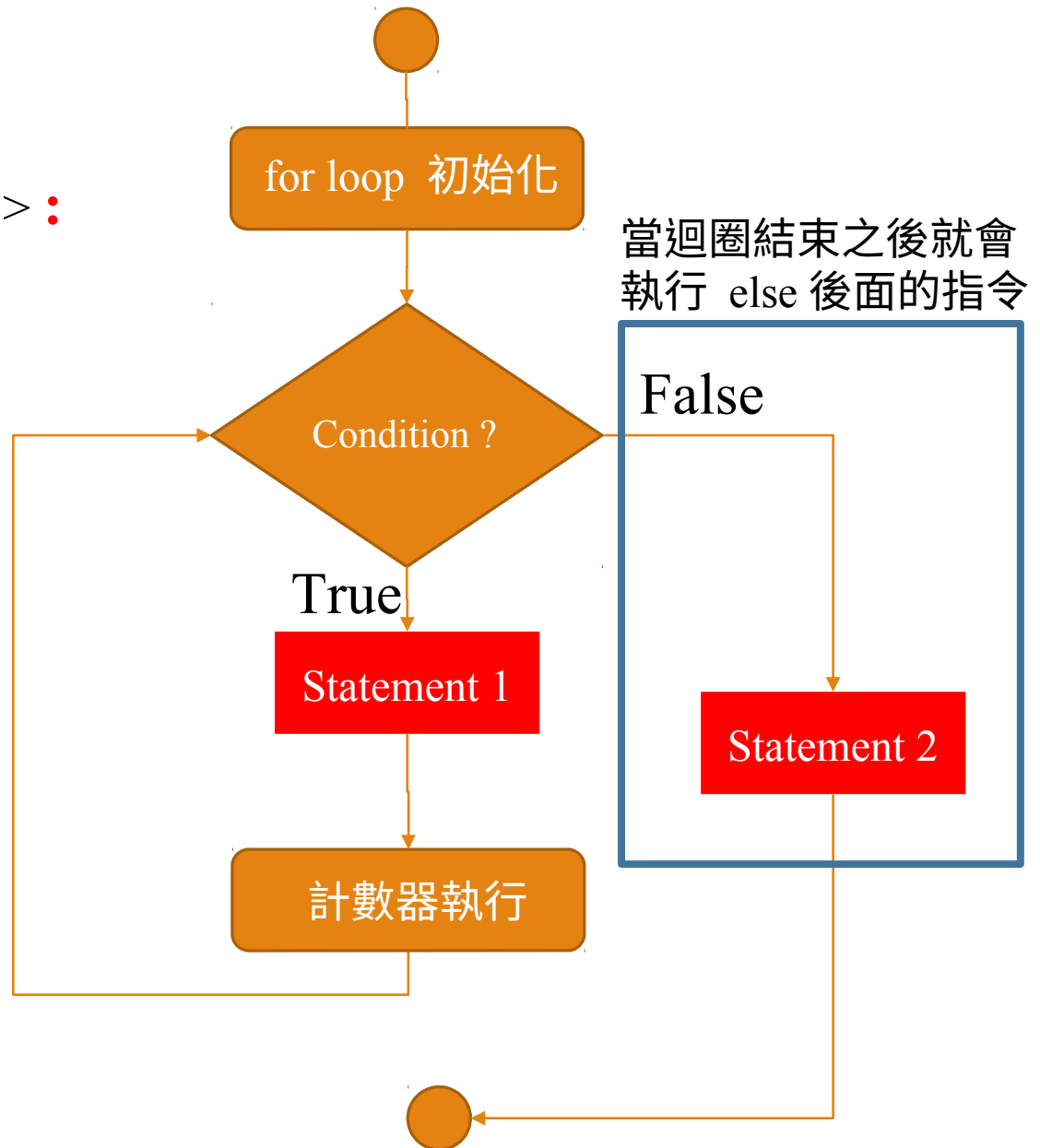
使用 for loop 可能會需要控制變數起點、終點以及疊代計數器的值

就先從最簡單的 for loop 開始吧 ~

```
for <變數> in <物件 (資料) 集合> :  
    <statement 1>  
else :  
    <statement 2>
```

迴圈

```
for <變數> in <物件 (資料) 集合> :  
    <statement 1>  
else :  
    <statement 2>
```

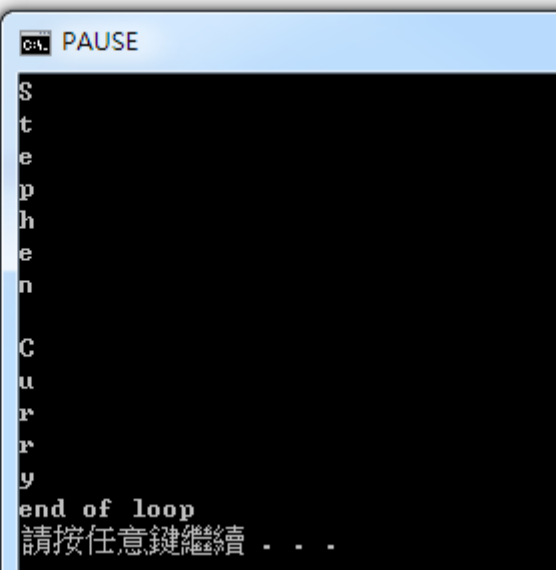


迴圈

for... in ... :

[Note. For 指令結束後的 “:” 要記得加]

```
1
2 SC = "Stephen Curry"
3 for kunkun in SC:
4     print(kunkun)
5 else:
6     print("end of loop")
```



PAUSE

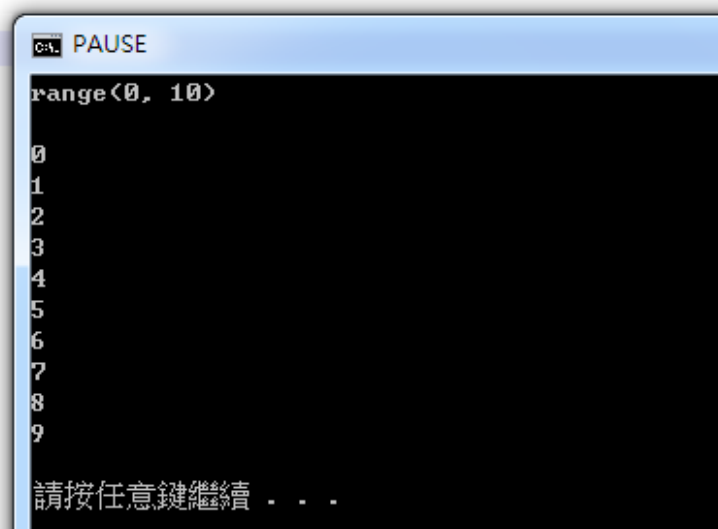
S
t
e
p
h
e
n
C
u
r
r
y
end of loop
請按任意鍵繼續 - - -

迴圈

for... in range... :

function	meaning
range([start], stop, [step])	建立整數串列

```
a = range(10)
print (a, "\n")
for a in range (10) :
    print(a)
```



The screenshot shows a terminal window titled "PAUSE" with the command `range(0, 10)` entered. The output displays the numbers 0 through 9, each on a new line. At the bottom of the window, there is a prompt in Chinese: "請按任意鍵繼續 . . .".

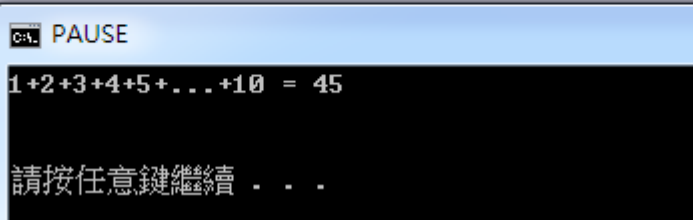
迴圈

for... in range... :

[Note: 注意 Range 的範圍]

for i in range (10) , 事實上只有跑 0 – 9

```
icy = 0
for i in range (10) :
    icy += i
else:
    print( "1+2+3+4+5+...+10 = %d " %(icy) )
    print("")
```



迴圈

for... in range... :

怎麼解決 (range 的值 +1)

```
icy = 0
icy2 = 0
for i in range (11) :
    icy += i
else:
    print( "1+2+3+4+5+...+10 = %d " %(icy) )
    print("")

for i in range (1,11,1) :
    icy2 += i
else:
    print( "1+2+3+4+5+...+10 = %d " %(icy2) )
    print("")
```

PAUSE

1+2+3+4+5+...+10 = 55

1+2+3+4+5+...+10 = 55

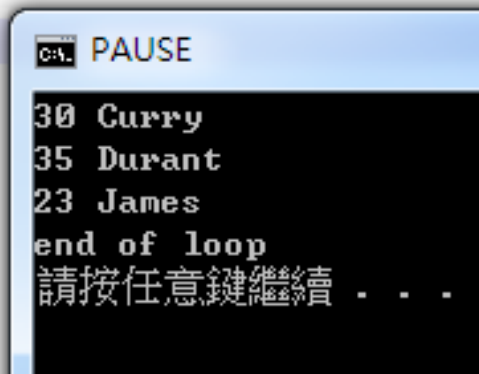
請按任意鍵繼續 . . .

迴圈

for... in (字典資料)... :

[Note. 不同資料型態中間 要記得用 “,” 隔開]

```
player = {30:"Curry", 35: "Durant", 23: "James"}  
for kunkun in player:  
    print(kunkun , player[kunkun])  
else:  
    print("end of loop")
```



```
PAUSE  
30 Curry  
35 Durant  
23 James  
end of loop  
請按任意鍵繼續 . . .
```

迴圈

for... in ... :

不縮排會發生什麼事

```
player = {30:"Curry", 35: "Durant", 23: "James"}  
for kunkun in player:  
print(kunkun , player[kunkun])
```

PAUSE

File "F:\Course\1062\python\teach_code\ch6\4.py", line 4
print(kunkun , player[kunkun])
^

IndentationError: expected an indented block

請按任意鍵繼續 . . .

迴圈

break:

如果需要跳出迴圈怎麼做？ 使用 break 指令
在迴圈裡加入 if 和 break 的組合來跳出迴圈

```
for <express> in <物件集合> :  
    if <condition> :  
        <statement 1 >  
        break  
    else :  
        <statement 2 >  
else :  
    <statement 3 >
```

迴圈

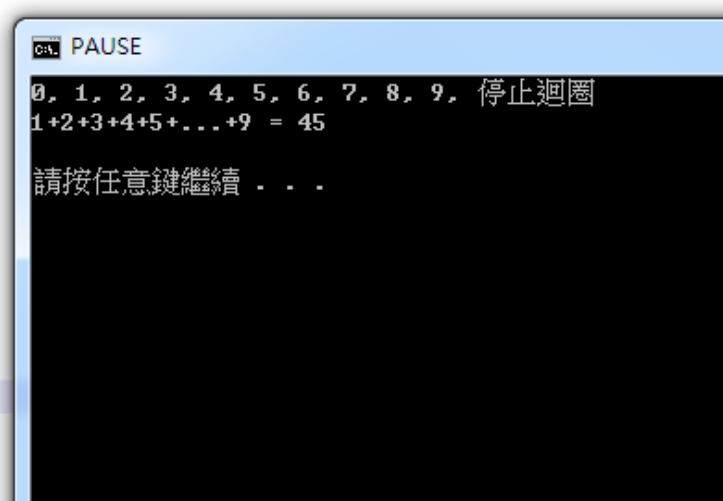
break:

如果 $i > 9$, 使用第一個 **break** 來跳出迴圈, 也不會執行第 2 個 **else**

第 1 個 **else**: 印出 i , 再修改 **end** 參數使得印完 i 後不會直接跳行 (預設是 **end=** “\n”)

```
icy = 0
for i in range (11) :
    if(i > 9):
        print("停止迴圈")
        break
    else:
        print(i, end=" ")
        icy += i
else:
    print("\n迴圈結束")

print( "1+2+3+4+5+...+9 = %d " %(icy) )
```



```
PAUSE
0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 停止迴圈
1+2+3+4+5+...+9 = 45
請按任意鍵繼續 . . .
```

迴圈

continue: (少用)

如果需要跳過目前的迴圈，直接執行下一次迴圈怎麼做？

在迴圈裡加入判斷式，再使用 continue 來跳到下一次迴圈

```
for <express> in <物件集合> :  
    if <condition> :  
        <statement 1 >  
        continue  
    else :  
        <statement 2 >  
else :  
    <statement 3 >
```

迴圈

continue:

迴圈裡加入判斷式 ($i > 10$, and $i \% 3 == 0$), 如果 i 是 3 的倍數
則使用 `continue` 來跳到下一次的迴圈

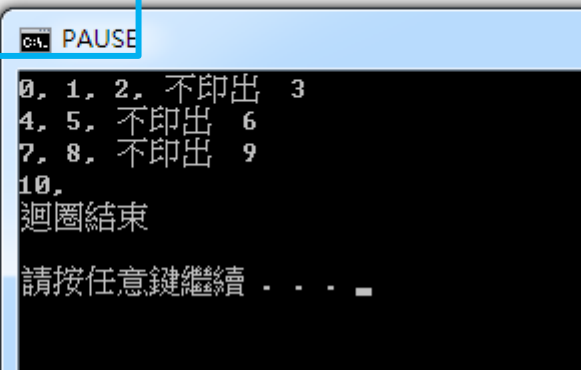
```
icy = 0

for i in range (11) :

    if( (i>0 ) and (i % 3 == 0) ):
        print("不印出 ", i)
        continue

    else:
        print(i,end=" ", " ")

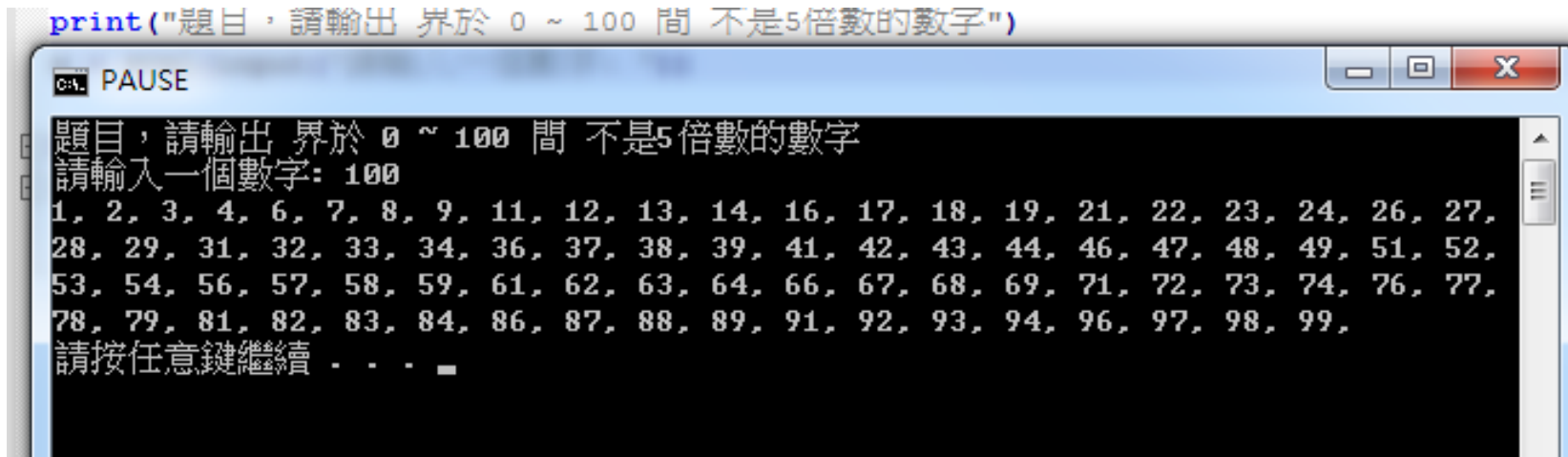
else:
    print("")
    print("迴圈結束")
```



隨堂練習

請輸出 0 ~ 100 間不是 5 倍數的數字

```
print("題目，請輸出 界於 0 ~ 100 間 不是5倍數的數字")
```



```
題目，請輸出 界於 0 ~ 100 間 不是5倍數的數字  
請輸入一個數字: 100  
1, 2, 3, 4, 6, 7, 8, 9, 11, 12, 13, 14, 16, 17, 18, 19, 21, 22, 23, 24, 26, 27,  
28, 29, 31, 32, 33, 34, 36, 37, 38, 39, 41, 42, 43, 44, 46, 47, 48, 49, 51, 52,  
53, 54, 56, 57, 58, 59, 61, 62, 63, 64, 66, 67, 68, 69, 71, 72, 73, 74, 76, 77,  
78, 79, 81, 82, 83, 84, 86, 87, 88, 89, 91, 92, 93, 94, 96, 97, 98, 99,  
請按任意鍵繼續 . . . ■
```

迴圈

while :

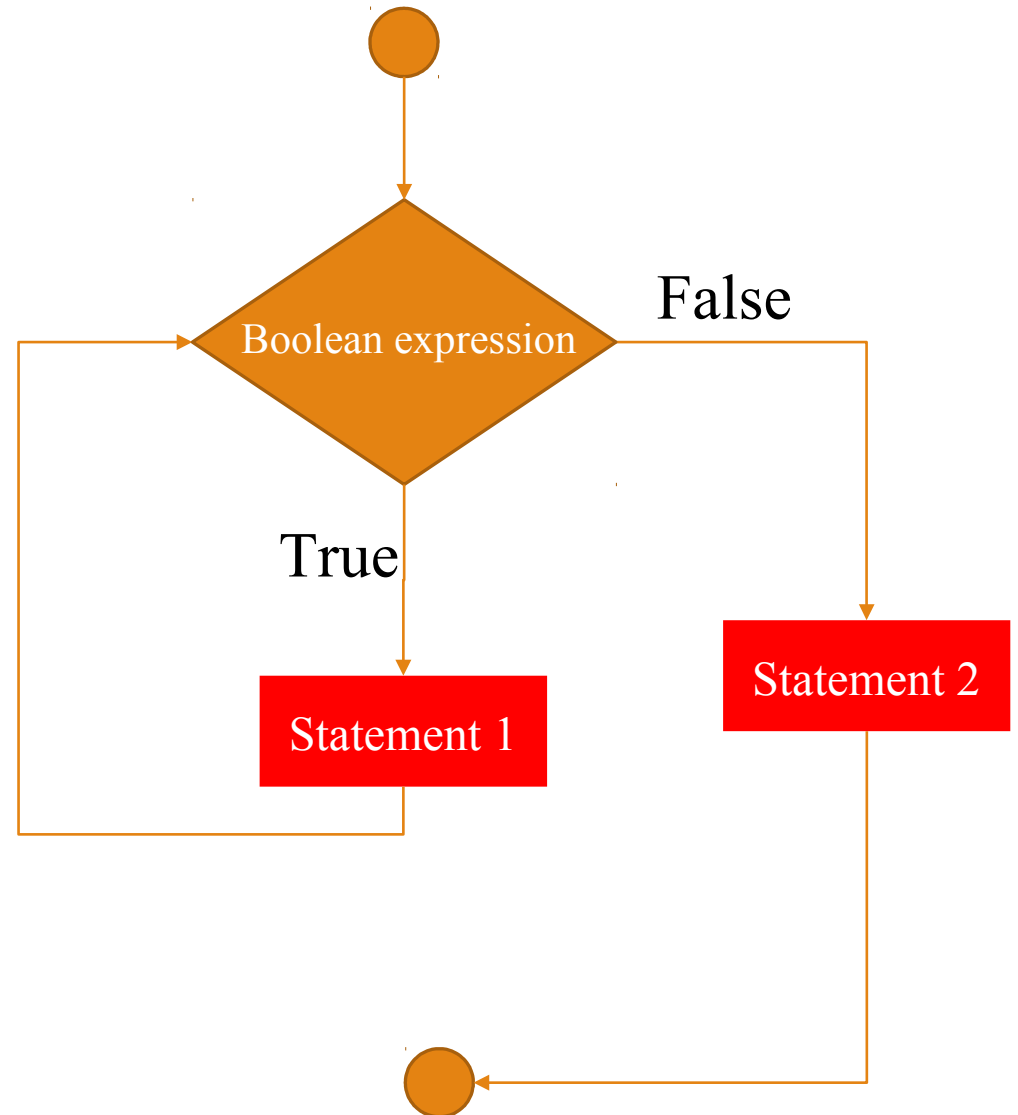
while 迴圈是程式語言中另一種迴圈，但是較 for loop 困難一些
使用需要一個判斷式決定迴圈是否執行

while 迴圈適合用在你不知道迴圈終點是什麼時候

```
while <Boolean expression> :  
    <statement 1>  
else :  
    <statement 2>
```

迴圈

```
while <Boolean expression> :  
    <statement 1>  
else :  
    <statement 2>
```



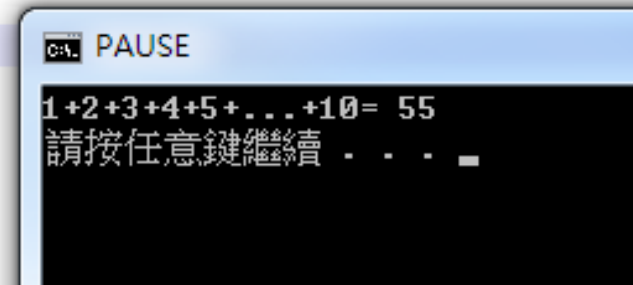
迴圈

while :

note: 如果要用計數器來設計 while 迴圈，記得在迴圈內讓計數器增加（減少）值

ex. 計算 1+2+3+4+...+10

```
1 i = 0
2 icy = 0
3
4 while ( i <= 10):
5     icy += i
6     i += 1
7
8 else:
9     print("1+2+3+4+5+...+10=", icy)
10
```

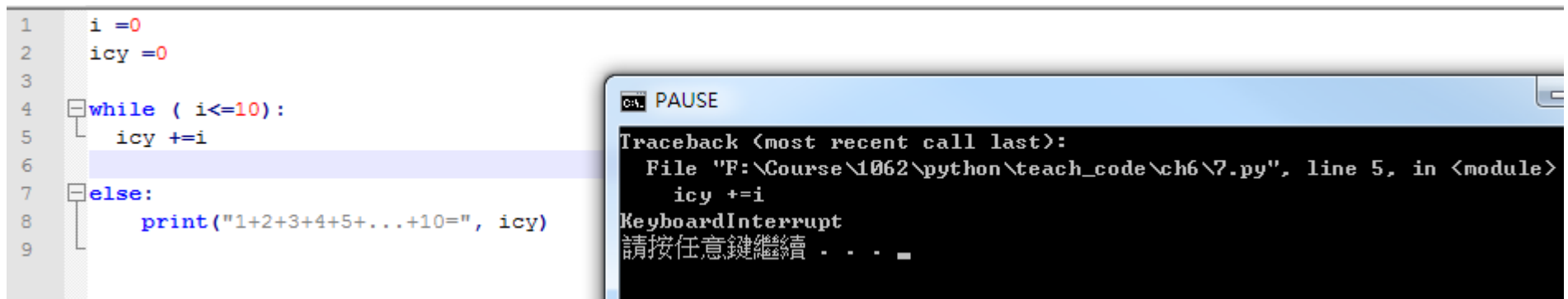


迴圈

while :

如果沒讓 i 增加會發生什麼事

程式不會停，而且需要按下 `ctrl + c` 才能讓程式停下來



```
1 i =0
2 icy =0
3
4 while ( i<=10):
5     icy +=i
6
7 else:
8     print("1+2+3+4+5+...+10=", icy)
9
```

```
PAUSE
Traceback (most recent call last):
  File "F:\Course\1062\python\teach_code\ch6\7.py", line 5, in <module>
    icy +=i
KeyboardInterrupt
請按任意鍵繼續 . . .
```

迴圈

巢狀迴圈

有些問題不太可能只使用一個迴圈就可以寫出程式

所以迴圈跟 if 判斷式一樣是可以一次執行很多層

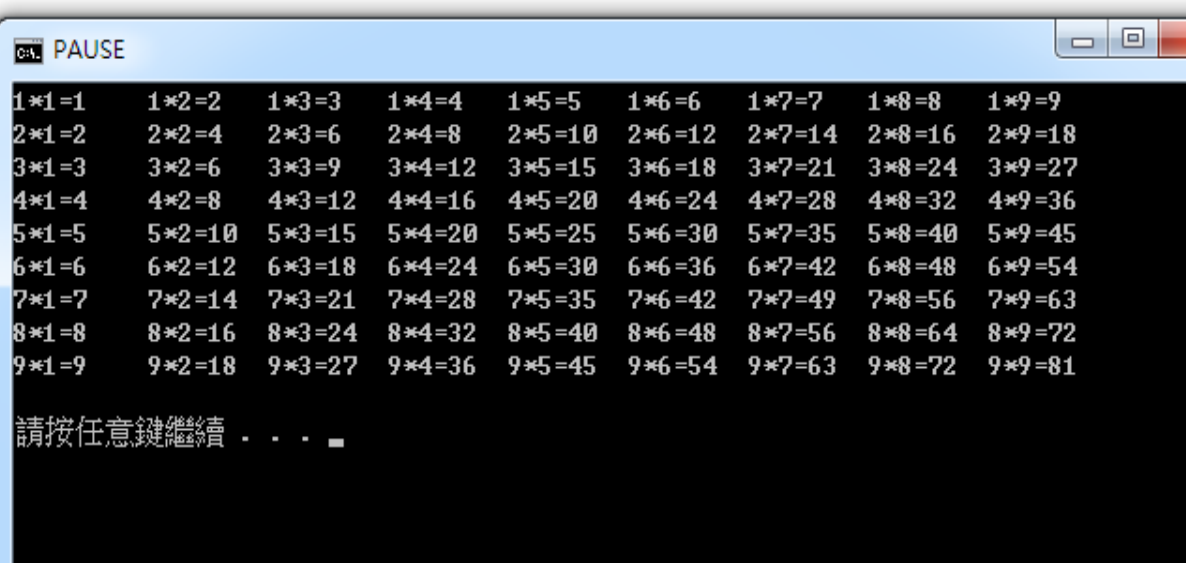
要記得的是內部迴圈做完才會做外部迴圈

迴圈

99 乘法表，使用巢狀迴圈：

1. 2 個 for loop (1 to 9)
2. $k=i*j$
3. 印出 $i*j=k$
4. 修改 end
5. 印完一行後
使用內迴圈 else 換行

```
k=0;
for i in range (1, 10):
    for j in range (1, 10):
        k = i*j
        print( "%d*%d=%d\t" %(i,j,k), end= " " )
    else:
        print("")
```



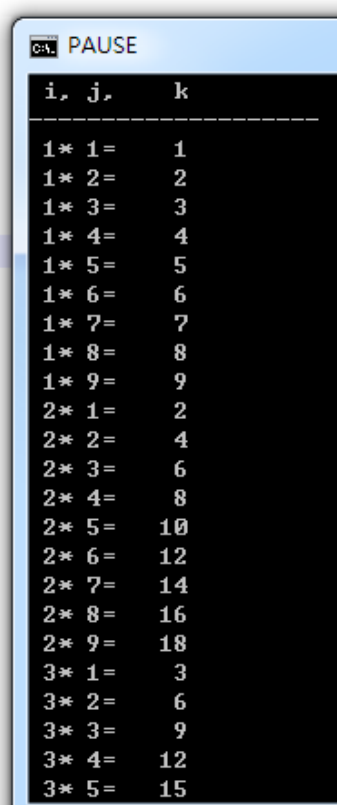
1*1=1	1*2=2	1*3=3	1*4=4	1*5=5	1*6=6	1*7=7	1*8=8	1*9=9
2*1=2	2*2=4	2*3=6	2*4=8	2*5=10	2*6=12	2*7=14	2*8=16	2*9=18
3*1=3	3*2=6	3*3=9	3*4=12	3*5=15	3*6=18	3*7=21	3*8=24	3*9=27
4*1=4	4*2=8	4*3=12	4*4=16	4*5=20	4*6=24	4*7=28	4*8=32	4*9=36
5*1=5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30	5*7=35	5*8=40	5*9=45
6*1=6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

請按任意鍵繼續 . . .

迴圈

99 乘法表，使用巢狀迴圈 和格式化輸出：

```
k=0
print(" i, j,      k")
print("-"*20)
for i in range(1, 10):
    for j in range(1,10):
        k=i*j
        print('{0:2d}*{1:2d}={2:>5d}'.format(i,j,k))
```



PAUSE

i, j,	k
1* 1=	1
1* 2=	2
1* 3=	3
1* 4=	4
1* 5=	5
1* 6=	6
1* 7=	7
1* 8=	8
1* 9=	9
2* 1=	2
2* 2=	4
2* 3=	6
2* 4=	8
2* 5=	10
2* 6=	12
2* 7=	14
2* 8=	16
2* 9=	18
3* 1=	3
3* 2=	6
3* 3=	9
3* 4=	12
3* 5=	15

隨堂練習

請輸入一個數字 (n)，並找到最靠近 n 的二個質數
其中，二數字當中一個要比 n 大，一個要比 n 小

Ex `a<-100`, output : 97, 101

`a<-93`, output : 89, 97

Any Questions !?

下課囉 ~