小朋友玩大蟒蛇 2.1

目錄 2.1

第3章 指令 (Statements)

問：複習一下，我們到目前為止學習過的指令有記得多少？

答：試試大家的記憶體有多大：

1. 列印指令
2. 指派指令
3. f-字串指令
4. 註解指令
5. 空白行指令
6. 假設指令

以上指令都有學過嘴！忘了回頭複習一下吧。

其中除了註解指令及空白行指令外，全都是 Python 的執行指令，接下來我們要學習迴路指令。

這裡我們將介紹更多指令，叫做迴路或迴圈指令。

對於重複的任務我們可以用 Python 迴路指令來設定在某種條件下，由一個程式區塊來執行重複的任務。

Python 的迴路指令，只有兩種：

1. for 迴路指令
2. while 迴路指令

3.1 介紹 for 迴路指令

語法

```
for <條件敘述> 若為真：

                <程式區塊>
```

其中：

1. <條件敘述> 最常用的型態是 <物件變數> in <序列物件>
2. 序列物件的特性就是有可迭圈（iterable）的迭代性 目前我們學習過的字串物件，就是序列物件的一個例子 下一章我們還會學習更多的序列物件，包含有： 列表、元組、集合、字典等序列物件
3. 當序列物件從頭到尾的迴圈中， Python 逐個元素載入物件變數，執行此程式區塊
4. 在程式區塊中，Python 又設計兩個控制指令，如下：
   A. 破圈指令（break statement）：讓我們終止執行 for 迴路指令破圈而出，接著執行 for 迴路指令下面的指令
   B. 跳圈指令（continue statement）：讓我們終止執行此程式區塊並執行下一輪迴路的<條件敘述>

注意：

保留字: for 那一行最後的字元必須是「:」 保留字: in <序列物件> 就是要求 Python 把此序列中元素建成一個迴圈 (iterable)

看一下 for 迴路指令的程式，例子如下：

In [4]:
```
message = '哈嘎, 尤勇, 歡迎!'

print('for 迴路開始...')

for char in message:
    print(f'{char}', end='-')
print()
print('for 迴路結束')
```

```
for 迴路開始...
哈-嘎-,-  -尤-勇-,-  -歡-迎-!-
for 迴路結束
```

In [19]:
```
# 破圈指令 (break statement) 例子
message = '哈嘎, 尤勇, 歡迎!'

print('for 迴圈開始...')

for char in message:

    if char == ",":
        print(f'\n====破到 "{char}" 破圈而出!')

        # 破圈指令 break
        break

    print(f'{char}', end='-')

print('for 迴圈終止\n')
```

```
for 迴圈開始...
哈-嘎-,-  -尤-勇-
====破到 "," 破圈而出!
for 迴圈終止
```

In [20]:
```
# 跳圈指令 (continue statement)例子
print('for 迴圈開始...')

for char in message:

    if char == ",":
        print(f'\n====破到 "{char}" 跳下一迴圈!')

        # 跳圈指令 continue
        continue

    print(f'{char}', end='-')
print()
print('for 迴圈終止')
```

```
for 迴圈開始...
哈-嘎-,-  -尤-勇-
====破到 "," 跳下一迴圈!
 -歡-迎-!-
for 迴圈終止
```

問：迴路指令中，有內建計數器的功能嗎？

答：是的。

一般較古老計數器程式是這樣寫：

```
counter = 0        # 計數器設初值
counter_end = 5    # 計數器設終值

for counter < counter_end:
        <程式區塊>
        counter += 1   # 計數器加一
```

Python 簡化成用一個函式 range(範圍) 來當計數器。

以下程式說明：

In [23]:
```
# range() 函式例子
print('for 計數器開始...')

for num in range(3):
    print(f'{num}', end='-')
print('\nfor 計數器終止\n')
```

```
for 計數器開始...
0-1-2-
for 計數器終止
```

In [24]:
```
# 兩層迴路的例子
print('兩層迴圈開始...')
print('====第一層迴圈開始...')

for num in range(3):
    print(f'========第二層迴圈開始...')
    for char in 'abc':
        print(f'\t{num}-{char}', end='-')
    print(f'\n========第二層迴圈終止')
print('====第一層迴圈終止')
print('兩層迴圈終止')
```

```
兩層迴圈開始...
====第一層迴圈開始...
========第二層迴圈開始...
        0-a-    0-b-    0-c-
========第二層迴圈終止
========第二層迴圈開始...
        1-a-    1-b-    1-c-
========第二層迴圈終止
========第二層迴圈開始...
        2-a-    2-b-    2-c-
========第二層迴圈終止
====第一層迴圈終止
兩層迴圈終止
```

3.2 while 迴圈指令

語法

```
while <條件敘述>為真：
                <程式區塊>
```

其中：

1. 條件敘述若為真，就執行迴圈內程式區塊。
2. while 迴圈指令也可在程式區塊中， 呼叫破圈指令（break）和跳圈指令（continue）。

注意：

保留字 while 那一行最後的字元必須是「:」。

看一下 while 迴圈指令的程式例子如下：

In [26]:
```
counter = 0   # counter 設初值

print('while 迴圈開始...')
while counter < 10:
    print(f'\t counter = {counter}')

    counter += 1   # counter 加一

print('while 迴圈終止\n')
```

```
while 迴圈開始...
         counter = 0
         counter = 1
         counter = 2
         counter = 3
         counter = 4
         counter = 5
         counter = 6
         counter = 7
         counter = 8
         counter = 9
while 迴圈終止
```

In [27]:
```
counter = 0   # counter 設初值

print('while 迴圈開始...')
while counter < 10:
    print(f'\t counter = {counter}')

    if counter == 5:
        print(f'\t while 迴圈中破到 {counter} 破圈而出...')
        break

    counter += 1   # counter 加一

print('while 迴圈終止\n')
```

```
while 迴圈開始...
         counter = 0
         counter = 1
         counter = 2
         counter = 3
         counter = 4
         counter = 5
         while 迴圈中破到 5 破圈而出...
while 迴圈終止
```