

# 10 條件迴圈



## 10-1. 重複結構

我們在聽音樂時，如果聽到喜歡的歌曲，可以設定讓這首歌持續播放，而不用每次動手重新播放音樂。就好比上體育課時，老師叫我們跑操場，而且一跑就是十圈，重複跑十次操場。



### Question 1: 什麼是迴圈？

迴圈就是讓一段程式碼，重複執行很多次的結構。

**Question 2: 迴圈有幾種？** Python 有兩種常用的迴圈，分別是條件迴圈（`while`）與計次迴圈（`for`）。

#### (1) `while` 條件迴圈：

在 `while` 迴圈中，依據測試條件是否成立，決定要不要繼續或跳出迴圈。

`while` 迴圈通常用於不固定次數的迴圈，只要條件符合就繼續做。

例如：猜數字遊戲

兩人（A 與 B）玩猜數字遊戲，一人(A)寫下心中所想的一個數，由另一人(B)去猜。然後提示B所猜數字的回答要「猜大一點」或「猜小一點」，直到B猜對為止。

#### (2) `for` 計次迴圈

在 `for` 迴圈中，依據計數變數值的變化，從初值到終值反覆執行區塊程式碼，當變數等於終值時離開迴圈。

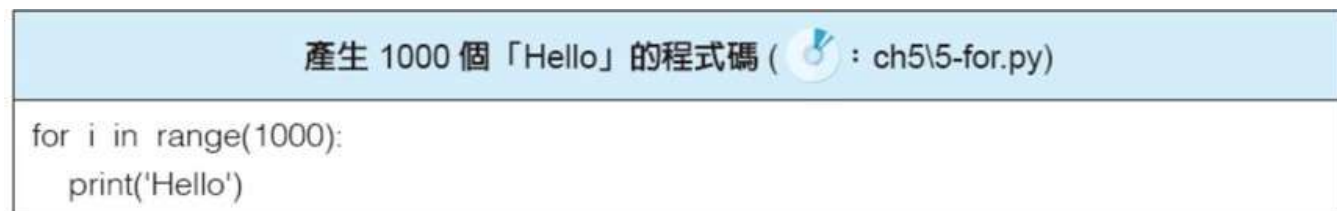
`for` 迴圈結構通常用於已知重複次數的程式。

例如：假設要撰寫程式產生1000個「Hello」。

方法一：寫1000個「`print('Hello')`」。



方法二：使用迴圈結構，簡化程式碼，達成相同功能。



## 10-2. while 條件迴圈

### 基本觀念

1. 條件迴圈(conditional loop): 程式碼要執行的確切次數未知。

只要布林條件式的判斷結果為True，電腦就會繼續執行迴圈，也就是重複執行這個程式區塊中的程式碼。通常，不知道要重複多少次時，就使用條件迴圈。比如玩猜數字遊戲，只要使用者沒有猜對就繼續玩，直到猜對為止。

2. 條件迴圈用while敘述式來建立。

範例1：從if 條件式到 while迴圈

```

1  ## if 條件式到 while迴圈
2  ## if 條件式
3  n = 5
4  if n < 10:
5      print("n小於10!")
6
7  ## while迴圈 (一): 將if改成while，電腦列不停止印
8  n = 5
9  while n < 10:
10     print("n小於10!")  # ctrl+c 停掉
11
12  ## while迴圈 (二): 要加上「變更布林條件的值」
13  n = 5  # 初始值
14  while n < 10:  # 當n小於10印出
15     print("n小於10!")
16     n = 100  # 變更條件
17     print("我要出迴圈!!")
18  print("我出來了")
19
20  ## while迴圈 (三): 靈活地「變更布林條件的值」
```

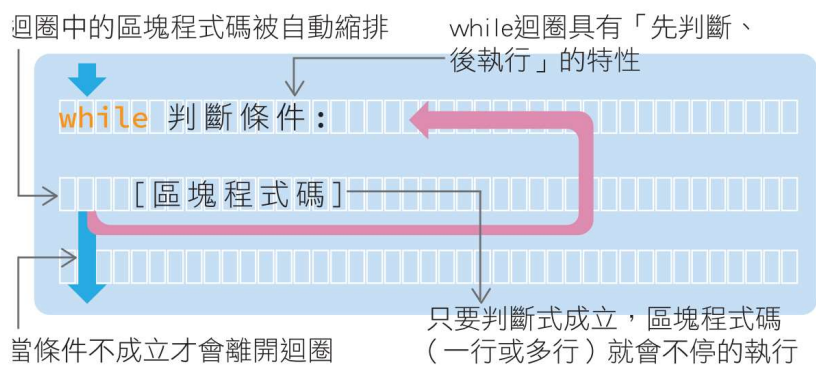
```

21  n = 5          # 初始值start
22  while n < 10:  # 判斷條件：當x小於10印出 stop
23      print("n小於10!")
24      n = n + 1  # 變更條件step
25  print("我出來了")

```

## 基礎語法

1. while 指令後面的測試條件，若為真就執行該區塊程式碼，直到測試條件為假時跳出。



### while 迴圈

while是指「當」判斷條件成立或為真值時，才會執行底下區塊的內容，執行完區塊後，再回while判斷是否條件繼續成立，若不成立，則離開迴圈，若成立就一直反覆執行區塊內容。

圖 4-3.5 while 迴圈語法說明

## 2. 虛擬程式碼: 所有條件迴圈的結構形式(3個s)

設定初始條件            # 初始條件 (Start)

條件為True時：        # 判斷條件 (Stop)

    程式碼每次在迴圈內執行

    變更條件            # 變更條件 (Step)

## 3. 流程圖

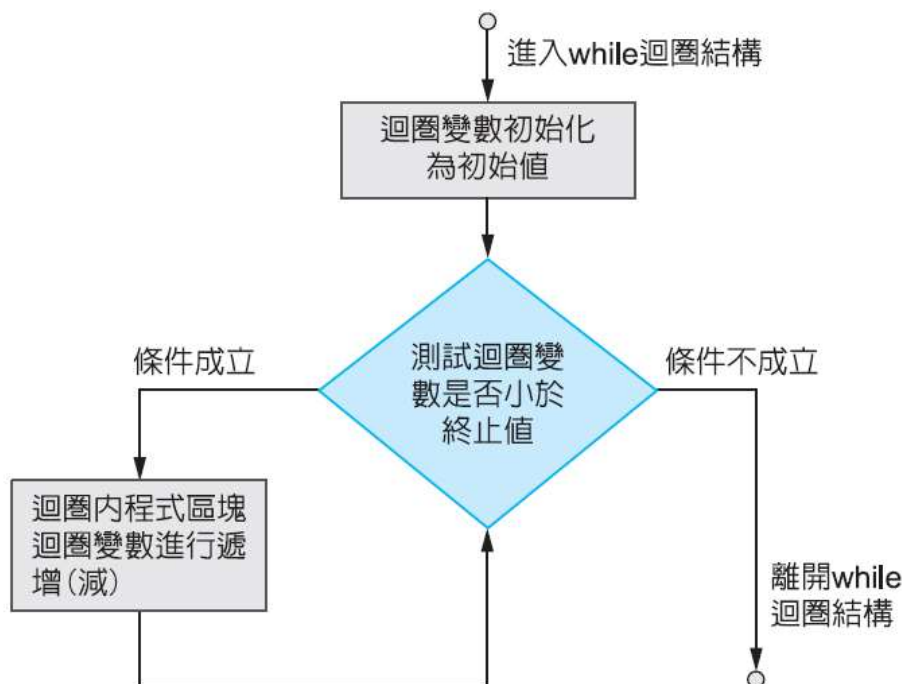


圖 A-14 流程圖

## 範例2：列印10次hello

```

1  ## 一般型迴圈
2  ## 方法 (一) :
3  n = 0 # 初始條件 Start (Python習慣從0開始算)
4  while n < 10: # 判斷條件 Stop (n = 9 · 印第10次 ; n = 10 · 跳出迴圈。 )
5      print("hello")
6      n = n + 1 # 更新條件 Step

```

hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello  
hello

```

1  ## 方法 (二) : (順序不同 · 更合邏輯)
2  n = 0 # 初始條件
3  while n < 10: # 判斷條件
4      n = n + 1 # 更新條件
5      print("這是第", n, "次的hello")

```

這是第 1 次的hello  
這是第 2 次的hello  
這是第 3 次的hello  
這是第 4 次的hello  
這是第 5 次的hello  
這是第 6 次的hello  
這是第 7 次的hello  
這是第 8 次的hello  
這是第 9 次的hello  
這是第 10 次的hello

```

1  ## 方法 (三) : (更完整)
2  n = 0 # 初始條件
3  while n < 10: # 判斷條件
4      n = n + 1 # 更新條件
5      left = 10 - n
6      print("這是第", n, "次的hello, 還有", left, "次機會")

```

這是第 1 次的hello, 還有 9 次機會  
這是第 2 次的hello, 還有 8 次機會  
這是第 3 次的hello, 還有 7 次機會  
這是第 4 次的hello, 還有 6 次機會  
這是第 5 次的hello, 還有 5 次機會  
這是第 6 次的hello, 還有 4 次機會  
這是第 7 次的hello, 還有 3 次機會  
這是第 8 次的hello, 還有 2 次機會  
這是第 9 次的hello, 還有 1 次機會  
這是第 10 次的hello, 還有 0 次機會

## 【隨堂練習1】：從1到5逐一列印

```
n= 1
n= 2
n= 3
n= 4
n= 5
Finished
```

範例3：求算1到10之和， $1+2+3+\dots+8+9+10=?$ 

```
1  ## 記憶體迴圈
2  n = 0          # 初始條件
3  sum = 0
4  ## 1+2+3+...+8+9+10=?
5  while n < 10:  # 判斷條件
6      n = n + 1
7      sum = sum + n # 更新條件
8  print("答案是:", sum) # 最後結果
```

答案是：55

```
1  ## 類似寫法，但初始條件不同、(使得程式的順序也不同)
2  n = 1          # 初始條件(Start)
3  sum = 0
4  while n <= 10: # 判斷條件 (一直加到最後一項的數字)(Stop)
5      sum = sum + n # 更新條件 (Step)
6      n = n + 1
7  print("The answer is", sum)
```

The answer is 55

【隨堂練習2】：求算1到9奇數之和， $1+3+5+7+9=?$ 

~ 請注意Start, Stop, and Step的設定

$1+3+5+7+9=25$

範例4：求算1到n之和大於50， $1+2+3+\dots+n > 50$ 的n是多少？

```
1  ## 記憶體迴圈
2  n = 0          # 初始條件
3  sum = 0
4  while sum <= 50: # 判斷條件 (只要總和小於等於50就繼續加)
5      n = n + 1
6      sum = sum + n # 更新條件
7  print("最後加的n是", n) # 最後結果
```

最後加的n是 10

## Summary: while迴圈三大功能 - 列印、計數、加總

## 10-3 進階迴圈

### 特殊的指令

在特殊需求下，迴圈會使用**break**與**continue**，以及**else** 指令。

- 1. 要跳出迴圈，可以使用**break**指令讓迴圈終止不再執行

```
1  ## 針對對while迴圈執行
2  n = 0
3  while (n < 7):
4      n = n + 1
5      if n == 4:
6          break
7      print(n)
```

```
1
2
3
```

- 要跳過某一迴圈不執行，但還是繼續做後面的迴圈，則使用**continue**。

範例5：請寫一個程式，輸入總樓層，列出大樓所有樓層名稱。

華人有時蠻避諱「4樓」的，會把「4樓」稱為「5樓」。

```
1  ##針對對while迴圈執行continue
2  n = 0
3  while (n < 7):
4      n += 1
5      if n == 4:
6          continue
7      print(n)
```

```
1
2
3
5
6
7
```

- 迴圈正常結束時，執行**else** 程式區塊，若迴圈經由**break** 中斷，就不會執行**else**程式區塊。

### 10.0.1 無窮迴圈

當迴圈使用「**while True:**」指令時，測試條件會永遠成立，形成無窮迴圈。

- 要跳出迴圈，請使用**break**中斷。

```
while True :
    條件數值運算式
    if 條件 :
```

要跳出的指令

## 範例6: 猜大猜小遊戲

大家是否玩過一個遊戲，兩人(A與B)一起玩，A心中想一個數字，B猜A心中所想的數字。B每猜一次，A就回答「猜大一點」、「猜小一點」與「猜中了」，當B猜到A所想的數字遊戲就結束。我們可以將此遊戲寫成程式，並假設所猜的數字介於1到100之間。

### 解題想法

使用while 迴圈結構，不斷允許使用者輸入數字進行猜測，測試猜測值與目標值是否相等。若相等則終止迴圈，否則根據猜測值與目標值的大小關係，顯示「猜大一點」、「猜小一點」與「猜中了」等提示。

```
1  ## 直接變數賦值比較簡單
2  target = 88
3  while True:
4      guess = int(input("Enter a number: "))
5      if guess == target:
6          print("Well done!猜中了!")
7          break # 跳出迴圈
8      else:
9          print("Try again")
10         if guess > target:
11             print("Guess smaller, 猜小一點")
12         else:
13             print("Guess larger, 猜大一點")
```

### 加分練習題：積少成多 - 52週存錢法

小芳的叔叔是個月光族，往往會覺得“零錢不是錢”，小芳告訴叔叔一個52周存錢法，這是一種階梯式存錢法。在一年52周內，每週遞增，比如：如果第1周存10元，則第2周存30元、第3周50元.....以此遞進，每週堅持，到年底也是個可觀的數位哦。

試編寫52周存錢的程式，為體現程式的靈活性，第1周起初存入的金額和每週遞增的金額，均可以通過鍵盤輸入。

### 加分練習題：剪刀石頭布

下課期間，有的同學三三兩兩在玩“石頭剪刀布”的小遊戲，大家玩的好激烈、好開心呀。一到週末，小芳同學總是想找人一起玩這個遊戲，可是鄰居家小朋友生病了，怎麼辦呢？小芳突發奇想，能否和電腦一起玩這個遊戲，一決雌雄呢？電腦裡沒有現成的這個遊戲，那就用Python語言創作一個吧。

要求：電腦先出完後，人接著輸入（只能是“石頭剪刀布”中的一種），電腦自動判斷，人是否贏了，如果人贏了將結束運行，退出程式；如果人輸了或平局，程式將繼續運行。

這個遊戲最有趣的是，永遠也不知道電腦會出石頭、剪刀還是布。遊戲是不可預測的，隨機性的，正是這種隨機性才讓遊戲很有趣。在程式設計中就需要random模組來生成隨機性，建立石頭、剪刀和布的列表，然後使用choice()函數從清單中選擇一種結果。同時列出3種贏的可能性列表，即石頭贏剪刀、剪刀贏布、布贏石頭。如果人和電腦出的內容在贏的3種清單中，則顯示“你贏了”。用迴圈語句和條件陳述式來控制整個遊戲程式的運行與結束，判斷遊戲的最後輸贏。

第一步：導入隨機和時間模組；第二步：創建列表test，用於存放石頭、剪刀、布3種手勢；第三步：創建清單win，用於存放贏的3種情況；第四步：用while True讓程式迴圈運行，直到人贏為止；第五步：電腦隨機出一種手勢，接著輸入人出的手勢，用條件陳述式，判斷電腦和人的手勢誰贏了（通過和win列

表比較)；第六步：如果人贏了電腦，程式中斷，遊戲結束；如果人和電腦平局，則提示“平局”，遊戲繼續；如果人輸給了電腦，則提示“你輸了，遊戲繼續”。