while迴圈

- while迴圈基本概念
- 迴圈控制：break與continue
- while與for的差異
- 無窮迴圈

摘要：重複結構While

- Who(對象)：重複結構
- Why(目的緣由)：為什麼需要重複結構
  - ➢ 模擬現實生活中重複執行相同動作的情境
- ➢ When(應用時機)/Where(應用場域)：重複結構的應用情境
  - 根據條件的判斷結果來決定是否要重複執行某段程式碼區段
- What(內容定義)：重複結構的類型-While
- How(操作機制)：
  1. 設定條件式：根據條件式判斷結果決定是否要重複執行程式碼區段
  2. 定義重複動作的內容

摘要

本章要介紹的是的while迴圈，
while與for迴圈一樣都是為了要執行某段重複的動作，
for 迴圈會限定執行次數，
while迴圈則不會限定迴圈的執行次數，
也就是當while後方的條件式成立時，
將會不斷執行迴圈內的程式碼區段，直到條件式不成立為止。
While亦會搭配break、continue來跳脫迴圈。

重複結構

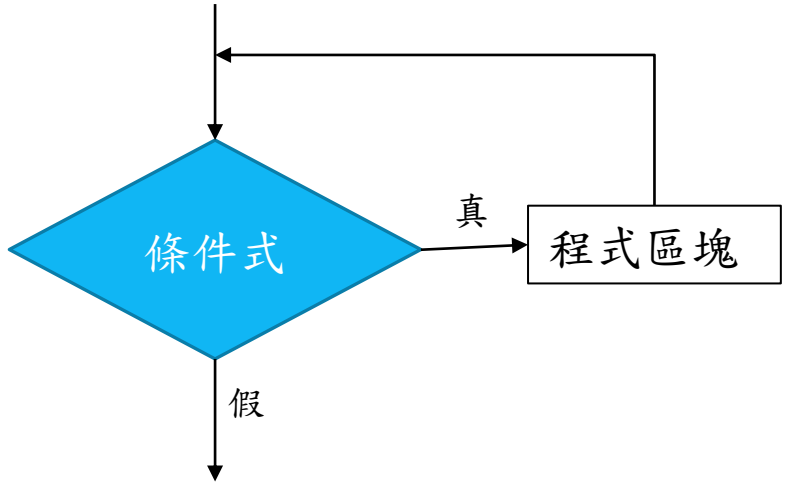
- 重複結構是電腦針對某個程式碼區塊，重複執行的流程結構，又稱為電腦的迴圈。
- Python程式語言常用的重複結構有for 迴圈或是while 迴圈。
- 我們在前面單元介紹過for迴圈的用法，也用for迴圈完成了許多程式。

接下來，我們要介紹第二種迴圈的用法：while迴圈。

While迴圈

- while迴圈主要根據迴圈變數的條件式的結果，決定迴圈是否要重複執行某段程式碼區塊。
  - 當條件式的結果為真(true)，反覆執行迴圈內的程式碼。
  - 若條件式的結果為假(false)，則不再執行迴圈內的程式碼。
- 當條件式的結果由True變成False時，會跳出迴圈停止重複。
- 條件式中迴圈變數的內容值，需在重複程式區段進行改變。
- 語法：

```
while(迴圈變數的條件式):
    迴圈程式碼
```



範例：while

- 目標：印出3次由使用者輸入的字串
- 程式流程：
  - 使用input得到字串
  - 設定迴圈變數repeatT初始值
  - while迴圈重複執行迴圈程式碼
  - 程式區塊：print()
  - 迴圈變數repeatT的變化

```
repeatS = input("input string：")
repeatT = 3
while repeatT>0:
    print(repeatS)
    repeatT = repeatT - 1
```

```
input string：test
test
test
test
```

改寫：for in range ➔while loop

- 目標：輸入2個整數(起始值、結束值)，計算從起始值到結束值間，所有數值的加總。
- for in range：

```
start = int(input("起始值："))
stop = int(input("結束值："))
total = 0
for i in range(start,(stop+1)):
    total = total + i
print ("總和為：", total)
```

```
起始值：1
結束值：10
總和為： 55
```

- while loop：

```
start = int(input("起始值："))
stop = int(input("結束值："))
total=0
value=start
while (value<=stop):
    total = total + value
    value=value+1
print ("總和為：",total)
```

while搭配continue與break的說明

while迴圈與for迴圈都可以藉由continue和break來改變迴圈的執行流程!

搭配if選擇判斷結構，決定是否要執行continue/break命令

- continue：跳過後面的程式碼，並返回迴圈的開頭以執行下一次迴圈。
- break ：強制跳離並結束迴圈。

While 迴圈變數的條件式：# 這個條件式控制迴圈是否執行

```
    # 迴圈程式碼
    # 迴圈變數的變化
    …
    if 關係運算式:
        continue
    …
    if 關係運算式:
        break
     …
```

while 與 for的差異(1/2)

- for用range()設定次數，或是list列表處理資料組，

都需要自訂的迴圈變數來記錄目前處理的資料或是位置。

- while迴圈是根據迴圈變數的條件式是否成立來決定是否執行迴圈程式碼，並且在迴圈程式碼中需要針對迴圈變數進行修正，不然會產生無窮迴圈的問題。

while 與 for的差異(2/2)

- for使用時機是在明確知道要做這個事情「幾次」的時候。

透過可以計數的資料型態來設定迴圈重複執行次數等，

例如range設定範圍、list列表。

- while 是一種未知次數的重複循環流程，

適合處理不固定次數的計算，

也就是重複程式碼執行的次數是無法事先確定。

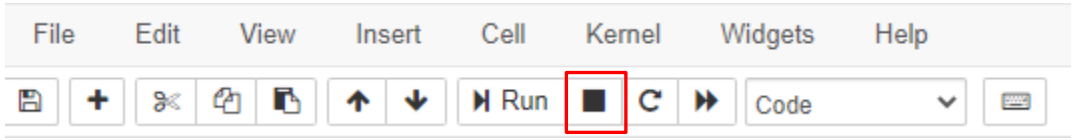
無窮迴圈

- 若條件式沒有控制好，產生永遠無法結束的迴圈，我們稱為無窮迴圈。
- 舉例：

```
while (True):
    print ("When will be end?")
```

```
When will be end?
When will be end?
When will be end?
When will be end?
When will be end?
When will be end?
When will be end?
When will be end?
.
.
.
```

- 發生無窮迴圈時，請先停止程式執行再回頭修正程式。



OJ習題Problem：C6_P1科技溫暖

1. 進入無限迴圈，持續接收使用者輸入
2. 讀取使用者輸入的訊息
3. 根據使用者輸入的訊息做出不同的回應
   - 如果訊息是“Hello”，印出 "Hello"
   - 如果訊息是“Hi”，印出 "Hi"
   - 如果訊息是“我叫阿軒”，印出特定的回應”阿軒你好，我是你的好朋友。”
   - 如果訊息是“你還在嗎?”，印出特定的回應"恩恩，我在聽。"
   - 如果訊息是“bye”，印出“阿軒再見。”，並且結束迴圈，程式結束執行
   - 如果訊息都不是上述特定指令，則不做任何回應，繼續下一次迴圈

OJ習題Problem：C6_P2迴圈終止

請求使用者輸入一個數字，若輸入的數字為零則終止迴圈，並輸出所有輸入的數字的總和。

1. 初始化總和變數 sum_result
2. 進入無限迴圈，持續接收使用者輸入的數字
3. 讀取使用者輸入的數字 number
4. 如果輸入為 0，結束迴圈，程式結束執行
5. 將使用者輸入的數字加到總和變數中
6. 印出最終的總和

OJ習題Problem：C6_P3周末野餐購物清單

讓使用者可以逐一輸入他們計劃購買的食材，當使用者輸入"完成"表示購物清單已經完成。最後，程式應該輸出整個野餐的購物清單。

1. 創建一個空的購物清單列表 shopping_list
2. 進入無限迴圈，持續接收使用者輸入的物品
3. 讀取使用者輸入的物品 item
4. 如果輸入為“完成”，結束迴圈，程式結束執行
5. 將使用者輸入的物品加入購物清單列表
6. 印出整個購物清單，使用空格連接列表中的每個物品

OJ習題Problem：C6_P4喝可樂

1. 讀取凱元購買的可樂瓶數量作為空瓶數量bottles
2. 讀取規定換新瓶所需的空瓶數量exchange_rule
3. 初始化凱元最終可以喝到的可樂瓶數量total_bottles為購買的可樂瓶數量
4. 當手上的空瓶數量(bottles)足以換新瓶時，持續進行換瓶操作
5. 計算換新瓶後手上剩餘的空瓶數量
6. 將換取的新瓶加入凱元最終可以喝到的可樂瓶數量中
7. 換新瓶後的空瓶數量加上新瓶的數量，繼續檢查是否可以再換新瓶
8. 輸出凱元最終可以喝到的可樂瓶數量

OJ習題

Problem

C6_P1：科技溫暖
C6_P2：迴圈終止
C6_P3：期末野餐購物清單
C6_P4：喝可樂

Exercise

C6_E1：猜數字遊戲
C6_E2：迴圈計算平均數
C6_E3：找最大質數

章節總結活動

Q 本章關鍵字
本章重點概念簡述

A 請將你的章節總結寫在Memo上，Memo的title請設定為 總結