

# 一步一腳印 精通Python 程式設計

指令條件控制與迴圈

課程講師：陳建銘

2018/05/16

# 課程大綱

---

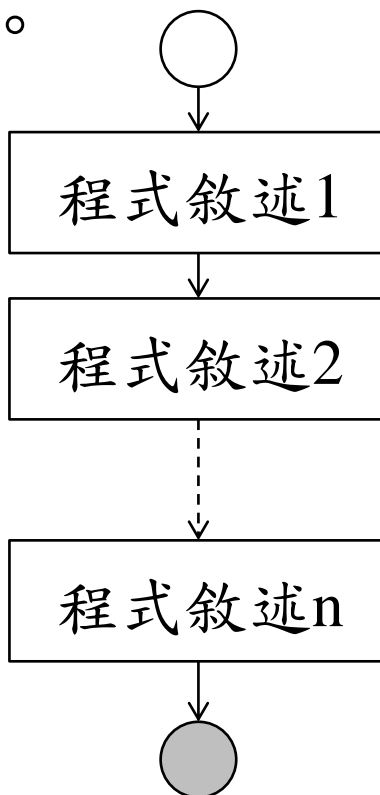
- 條件控制
  - 流程
  - 單一條件控制
  - 二選一條件控制
  - 多重條件控制
- 迴圈
  - while
  - for



# 條件控制

## 流程－

程式執行的方向稱為流程，流程可透過條件控制或迴圈來改變方向。



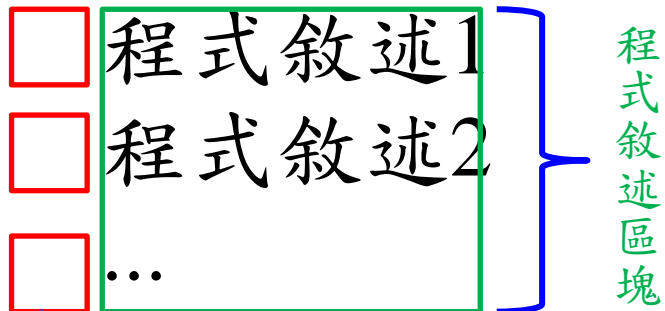
# 條件控制 Cont.

## 單一條件控制－

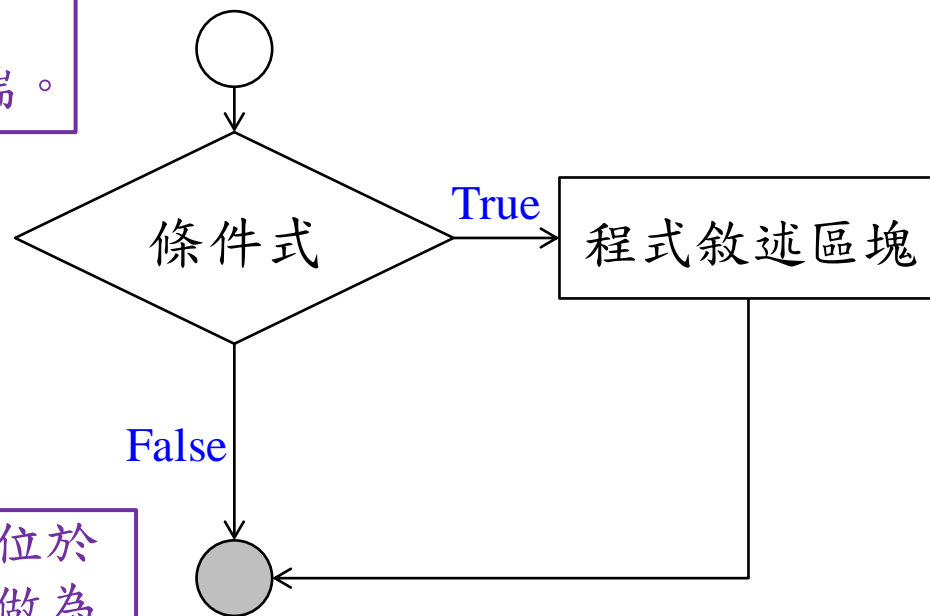
當某單一條件滿足時，則執行被定義的動作。

if 條件式：

區塊的開頭必須  
以冒號做為起始開端。



當條件滿足時，欲執行的所有動作皆須位於這個if區塊內，並且以四個空格為開頭做為此區塊的行內縮排。



條件式是一種運算式，其運算結果為布林值。

- 
- 條件控制
    - 流程
    - 單一條件控制
    - 實例說明
    - 二選一條件控制
    - 實例說明
    - 多重條件控制
    - 實例說明
  - 迴圈
    - while
    - 實例說明
    - for
    - 實例說明

# 條件控制 Cont.

## 範例1: 成績是否及格?

```
if __name__ == '__main__':
```

```
    score = 75
```

```
    description = '分及格'
```

```
    if score < 60:
```

```
        description = '分不及格'
```

```
    print(str(score) + description)
```

score = 75

輸出: 75分及格

score = 58

輸出: 58分不及格

# 條件控制 Cont.

## 範例2: 奇數或偶數?

```
if __name__ == '__main__':
```

```
    number = 5
```

```
    description = '是偶數'
```

```
    if number%2 == 1:
```

```
        description = '是奇數'
```

```
    print(str(number) + description)
```

number = 5

輸出: 5是奇數

number = 8

輸出: 8是偶數

# 條件控制 Cont.

## 範例3: 是否包含子字串?

```
if __name__ == '__main__':
```

```
    subString = 'gram'
```

```
    stringInfo = 'Python Programming'
```

```
    description = '包含於'
```

```
    if subString not in stringInfo:
```

```
        description = '不包含於'
```

```
    print(subString + description + stringInfo)
```

subString = 'gram'

輸出: gram 包含於 Python Programming

subString = 'ton'

輸出: ton 不包含於 Python Programming



# 條件控制 Cont.

## 範例4: 是否為長單字?

使用者可透過input函式顯示之提示訊息做文字輸入。

```
if __name__ == '__main__':
```

```
    word = input('請輸入一個英文單字:')
```

```
    description = '不是長單字'
```

```
    if len(word) >= 10:
```

```
        word = word + '長度' + str(len(word)) + ', '
```

```
        description = '是長單字'
```

```
    print(word + description)
```

word = 'businessman'

輸出: businessman長度11，是長單字

word = 'superman'

輸出: superman不是長單字

# 條件控制 Cont.

## 練習1: 判斷真偽

```
if __name__ == '__main__':  
    mathScore = 55  
    chineseScore = 66  
    status = True  
  
    if ((mathScore+11)>chineseScore) or \  
        ((chineseScore-5)<=mathScore):  
        status = False  
  
    print(status)
```

# 條件控制 Cont.

## 練習2: 判斷真偽

```
if __name__ == '__main__':  
    floatNumber = 22.5566  
    status = False  
  
    if (int(floatNumber)<floatNumber) and \  
        (len(str(floatNumber))>=7):  
        status = True  
  
    print(status)
```

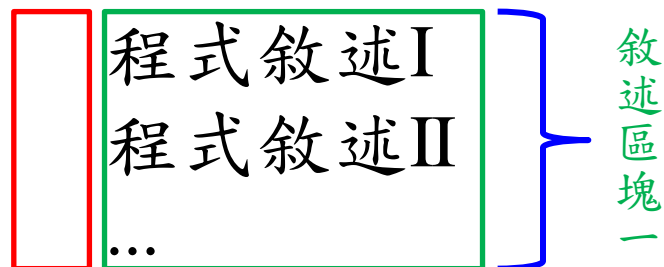
- 
- 條件控制
    - 流程
    - 單一條件控制
    - 實例說明
    - 二選一條件控制
    - 實例說明
    - 多重條件控制
    - 實例說明
  - 迴圈
    - while
    - 實例說明
    - for
    - 實例說明

# 條件控制 Cont.

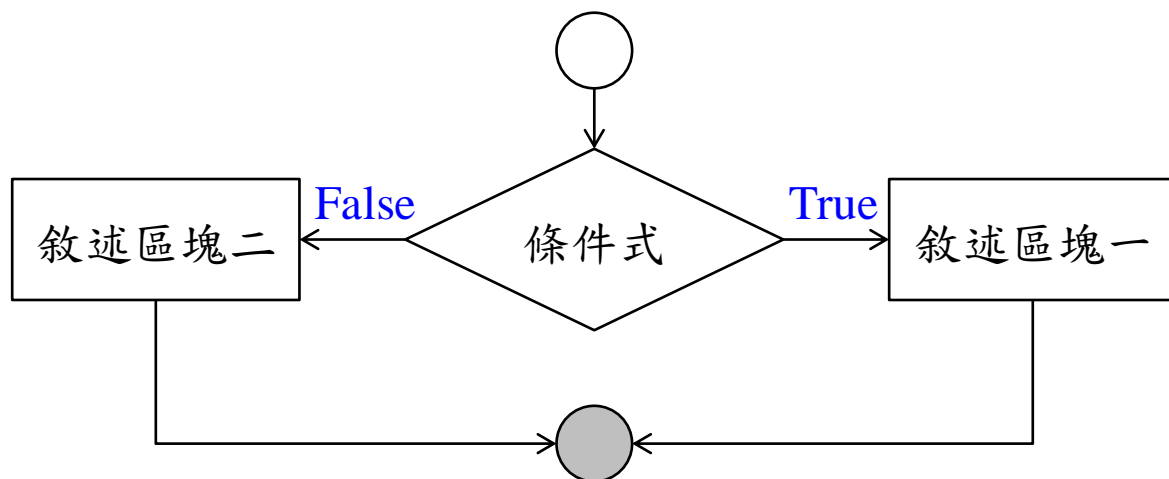
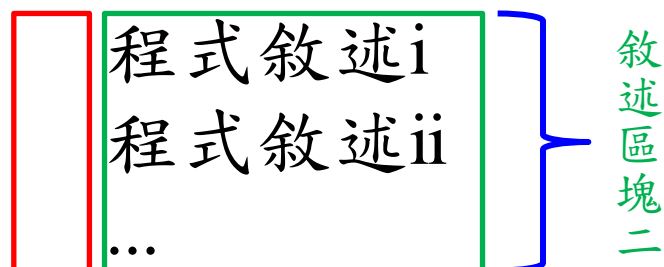
## 二選一條件控制－

當某條件滿足時，則執行此條件所定義的動作。  
否則執行另一條件所定義的動作。

if 條件式:



else:



# 條件控制 Cont.

## 範例1: 成績是否及格?

```
if __name__ == '__main__':
```

```
    score = 75
```

```
    description = "
```

```
    if score >= 70:
```

```
        description = '分及格'
```

```
    else:
```

```
        description = '分不及格'
```

```
    print(str(score) + description)
```

score = 75

輸出: 75分及格

score = 66

輸出: 66分不及格

# 條件控制 Cont.

## 範例2: 奇數或偶數?

```
if __name__ == '__main__':
```

```
    number = 5
```

```
    description = "
```

```
    if number%2 == 0:
```

```
        description = '是偶數'
```

```
    else:
```

```
        description = '是奇數'
```

```
    print(str(number) + description)
```

number = 5

輸出: 5是奇數

number = 8

輸出: 8是偶數

# 條件控制 Cont.

## 範例3: 是否為長單字?

```
if __name__ == '__main__':  
    word = input('請輸入一個英文單字:')  
    description = "
```

```
if len(word) >= 10:  
    word = word + '長度' + str(len(word)) + ', '  
    description = '是長單字'  
else:  
    description = '不是長單字'
```

```
print(word + description)
```



# 條件控制 Cont.

## 範例4: 成績狀態判定

```
if __name__ == '__main__':  
    score = int(input('請輸入成績: '))  
    status = "
```

```
if score >= 60:
```

```
    status = '成績及格'
```

```
else:
```

```
    if score >= 50:
```

```
        status = '請參加補考'
```

```
    else:
```

```
        status = '請重修'
```

```
print('狀態判定: ' + status)
```

score = 58

輸出: 狀態判定: 請參加補考

score = 75

輸出: 狀態判定: 成績及格

score = 41

輸出: 狀態判定: 請重修

- 
- 條件控制
    - 流程
    - 單一條件控制
    - 實例說明
    - 二選一條件控制
    - 實例說明
    - 多重條件控制
    - 實例說明
  - 迴圈
    - while
    - 實例說明
    - for
    - 實例說明

# 條件控制 Cont.

## 多重條件控制－

一筆一筆執行條件判斷，若找到符合的條件，則執行此條件所定義的動作。

if 條件式1:

□ ...

elif 條件式2:

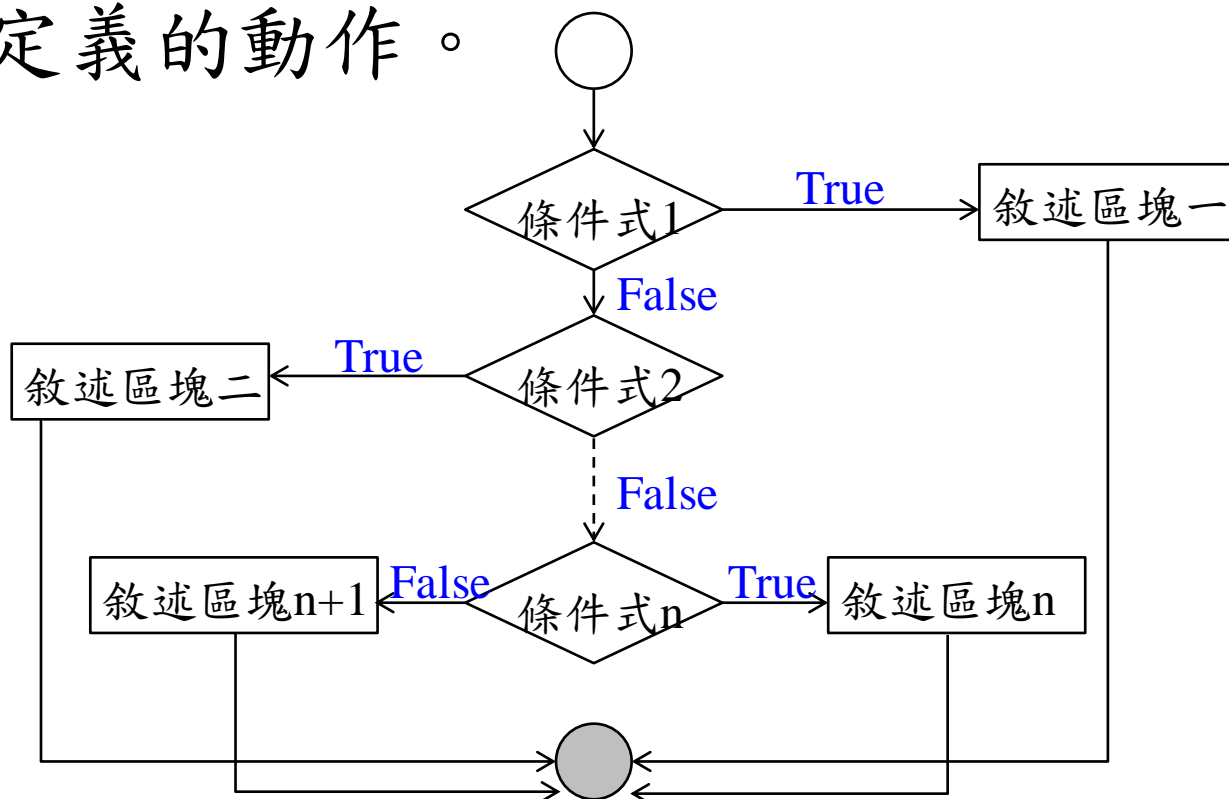
□ ...

elif ...:

□ ...

else:

□ ...



# 條件控制 Cont.

## 範例1: 分數等級判定

score = 87

輸出: 87分判定為甲

score = 62

輸出: 62分判定為丙

score = 94

輸出: 94分判定為優

```
if __name__ == '__main__':
```

```
    score = 87
```

```
    scoreLevel = "
```

```
    if score >= 90:
```

```
        scoreLevel = '優'
```

```
    elif 80 <= score and score < 90:
```

```
        scoreLevel = '甲'
```

```
    elif 70 <= score and score < 80:
```

```
        scoreLevel = '乙'
```

```
    elif 60 <= score and score < 70:
```

```
        scoreLevel = '丙'
```

```
    else:
```

```
        scoreLevel = '丁'
```

```
    print(str(score) + '分判定為' + scoreLevel)
```

# 條件控制 Cont.

## 範例2: 是否開啟冷氣?

```
if __name__ == '__main__':
```

```
    temperature = 26
```

```
    airConditionerOn = True
```

```
    if temperature > 24 and airConditionerOn == False:
```

```
        airConditionerOn = True
```

```
    elif temperature <= 24 and airConditionerOn == True:
```

```
        airConditionerOn = False
```

```
    else:
```

```
        pass
```

```
    print('是否開啟冷氣?' + str(airConditionerOn))
```

```
temperature = 26
```

```
airConditionerOn = False
```

```
輸出: 是否開啟冷氣? True
```

```
temperature = 23
```

```
airConditionerOn = True
```

```
輸出: 是否開啟冷氣? False
```

```
temperature = 24
```

```
airConditionerOn = False
```

```
輸出: 是否開啟冷氣? False
```

```
temperature = 25
```

```
airConditionerOn = True
```

```
輸出: 是否開啟冷氣? True
```

# 條件控制 Cont.

## 重點整理—

- 單一 if 陳述，條件為真時則執行 if 底下縮排的區塊。
- 縮排長度通常用2到5個空格，在此採用4個空格。
- if-else陳述，條件為真時就會執行 if 底下縮排的區塊，反之，則執行 else 底下縮排的區塊。
- if-elif-else陳述，視哪一個條件為真時就執行該縮排的區塊。如果都不為真，則會執行 else 縮排的區塊。else 就是以上皆非的情況。

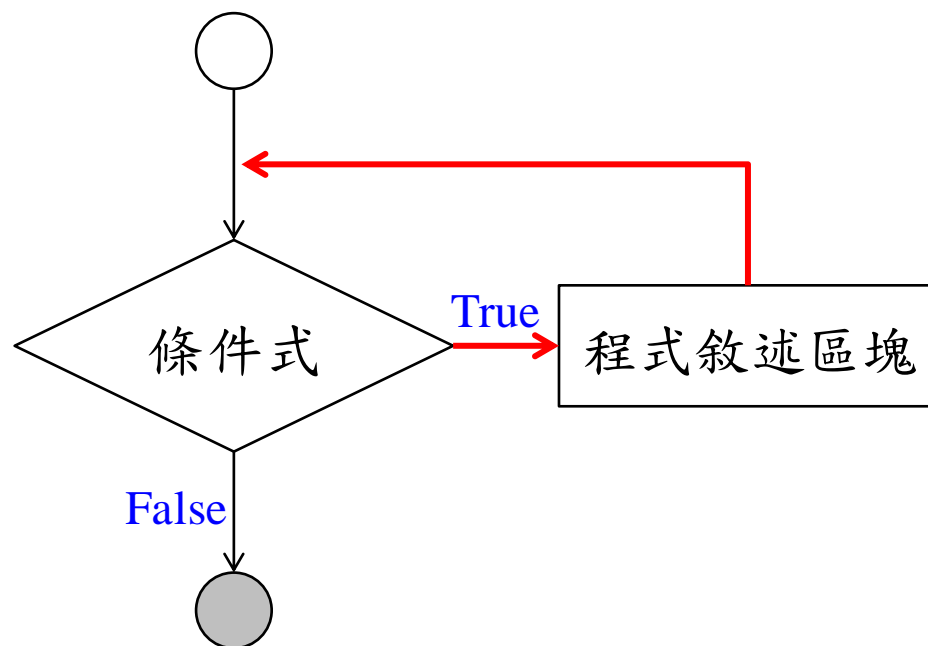
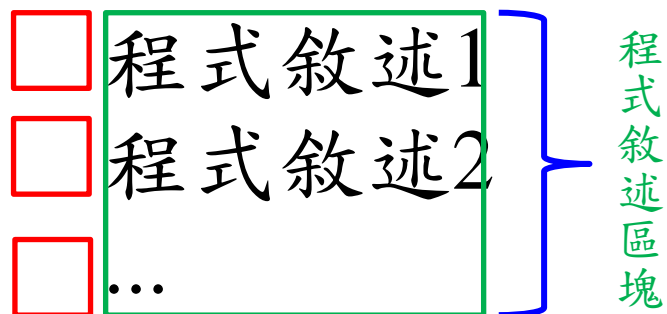
- 
- 條件控制
    - 流程
    - 單一條件控制
    - 實例說明
    - 二選一條件控制
    - 實例說明
    - 多重條件控制
    - 實例說明
  - 迴圈
    - **while**
    - 實例說明
    - **for**
    - 實例說明

# 迴圈

**while –**

當某條件滿足時，則重複執行被定義的動作。

while 條件式:



條件式是一種運算式，其運算結果為**布林值**。



# 迴圈 Cont.

## 範例1: 累加計數

```
if __name__ == '__main__':  
    summation, counter = 0, 1  
    accumulation = input('請輸入累加數值:')  
    while counter <= int(accumulation):  
        summation += counter  
        counter += 1  
    print('從1累加至' + accumulation + ', 總和為' + str(summation))
```

accumulation = 5

輸出: 從1累加至5，總和為15

accumulation = 10

輸出: 從1累加至10，總和為55

# 迴圈 Cont.

## 範例2: 計算2的N次方

```
if __name__ == '__main__':
```

```
    result, counter = 1, 1
```

```
    times = input('請輸入次方數: ')
```

```
    while counter <= int(times):
```

```
        result *= 2
```

```
        counter += 1
```

```
    print('2的' + times + '次方為' + str(result))
```

times = 5

輸出: 2的5次方為32

times = 10

輸出: 2的10次方為1024

- 
- 條件控制
    - 流程
    - 單一條件控制
    - 實例說明
    - 二選一條件控制
    - 實例說明
    - 多重條件控制
    - 實例說明
  - 迴圈
    - while
    - 實例說明
    - **for**
    - 實例說明

# 迴圈 Cont.

**for –**

重複執行N次被定義的動作或迭代某容器的所有值。

迴圈計數索引從0, 1, 2, ..., 次數減1

for 計數索引 in range(次數):

<input type="checkbox"/>	程式敘述1	} 程式敘述區塊
<input type="checkbox"/>	程式敘述2	
<input type="checkbox"/>	...	

for 容器值 in 某容器:

<input type="checkbox"/>	程式敘述1	} 程式敘述區塊
<input type="checkbox"/>	程式敘述2	
<input type="checkbox"/>	...	

計數索引的起始值從0開始。

# 迴圈 Cont.

## 範例1: 累加計數

```
if __name__ == '__main__':  
    summation = 0  
    accumulation = input('請輸入累加數值:')  
    for index in range(1, int(accumulation)+1):  
        summation += index  
    print('從1累加至'+accumulation+'，總和為'+str(summation))
```

accumulation = 5

輸出: 從1累加至5，總和為15

accumulation = 10

輸出: 從1累加至10，總和為55

# 迴圈 Cont.

## 範例2: 計算2的N次方

```
if __name__ == '__main__':
```

```
    result = 1
```

```
    times = input('請輸入次方數: ')
```

```
    for index in range(int(times)):
```

```
        result *= 2
```

```
    print('2的'+times+'次方為'+str(result))
```

times = 5

輸出: 2的5次方為32

times = 10

輸出: 2的10次方為1024

# 迴圈 Cont.

## 範例3: 列印99乘法表

```
if __name__ == '__main__':  
    for x in range(2, 10):  
        for y in range(1, 10):  
            print(str(x) + ' × ' + str(y) + ' = ' + str(x*y))
```

迴圈計數索引從2, 3, 4, ..., 9

輸出如下:

$$2 \times 1 = 2$$

$$2 \times 2 = 4$$

$$2 \times 3 = 6$$

⋮

$$9 \times 7 = 63$$

$$9 \times 8 = 72$$

$$9 \times 9 = 81$$

# 迴圈 Cont.

## 範例4: 列印直角三角形

```
if __name__ == '__main__':  
    drawLine = "  
    height = input('請輸入直角三角形之高度:')
```

```
    for x in range(int(height)):
```

```
        for y in range(x+1):
```

```
            drawLine += '*'
```

```
        print(drawLine)
```

```
        drawLine = "
```

height = 3

輸出: \*

\*\*

\*\*\*

height = 6

輸出: \*

\*\*

\*\*\*

\*\*\*\*

\*\*\*\*\*

\*\*\*\*\*



# 迴圈 Cont.

## 範例5: 某數出現次數

`import random` — 使用亂數產生器，須匯入相關函式庫

```
if __name__ == '__main__':
```

```
    counts = 0
```

```
    number = input('請輸入一個整數:')
```

```
    for x in range(100):
```

```
        digit = random.randint(1,6)
```

```
        if digit == int(number):
```

```
            counts += 1
```

```
    print(number + '共出現了' + str(counts) + '次')
```

number = 5

輸出: 5共出現了18

number = 2

輸出: 2共出現了24

# 迴圈 Cont.

## 範例6: 列印所有串列項目

```
if __name__ == '__main__':  
    counter = 1  
    colorList = ['紅', '橙', '黃', '綠', '藍', '靛', '紫']  
    for color in colorList:  
        print('第' + str(counter) + '個顏色:' + color)  
        counter += 1
```

輸出如下:

第1個顏色: 紅  
第2個顏色: 橙  
第3個顏色: 黃  
第4個顏色: 綠  
第5個顏色: 藍  
第6個顏色: 靛  
第7個顏色: 紫

# 迴圈 Cont.

## 範例7: 尋找串列中的整數

```
if __name__ == '__main__':  
    counts = 0  
    integerString = "  
    itemList = ['字串1', 123, True, 456]  
  
    for item in itemList:  
        if type(item) == int:  
            counts += 1  
            integerString += str(item) + '  
    print(integerString + '=> 共' + \  
          str(counts) + '個整數')
```

輸出如下:  
123 456 => 共2個整數

# 迴圈 Cont.

## 範例8: 尋找第一個3的倍數

```
if __name__ == '__main__':  
    index = 0  
    numberList = [221, 43, 396, 53, 612]
```

```
    for number in numberList:
```

```
        if number%3 == 0:
```

```
            index = numberList.index(number)
```

```
            break
```

```
    print('第一個3的倍數在索引值'+ \  
          str(index) + '的地方')
```

break指令直接跳離迴圈

輸出如下:

第一個3的倍數在  
索引值2的地方

# 迴圈 Cont.

## 範例9: 計算1到10之間的所有偶數和

```
if __name__ == '__main__':
```

```
    totalSum = 0
```

```
    for index in range(1, 11):
```

```
        if index % 2 != 0:
```

```
            [continue]
```

```
        else:
```

```
            [totalSum += index]
```

```
        print('數字為' + str(index))
```

```
    print('1到10之間的所有偶數和為' + str(totalSum))
```

continue指令執行下一輪迴圈

輸出如下:

數字為2

數字為4

數字為6

數字為8

數字為10

1到10之間的所有  
偶數和為30

# 迴圈 Cont.

## 練習1: 尋找第二個4的倍數

```
if __name__ == '__main__':  
    index, counter = 0, 0  
    numberList = ['36', 31.4, 'ABC', 96, True, 64, 'xyz', False, 8]  
  
    for number in numberList:  
        if type(number)==int and number%4==0:  
            if counter==1:  
                index = numberList.index(number)  
                break  
            else:  
                counter += 1  
  
    print('第二個4的倍數在索引值'+str(index)+'的地方')
```

# 迴圈 Cont.

## 重點整理－

- 迴圈用於重複性質的工作，while 迴圈利用控制變數決定何時結束迴圈。for 迴圈則用於具有多個元素的複合資料型態。
- 無窮迴圈是指沒有結束條件，一直執行的迴圈。
- 利用 break 陳述可提前跳出迴圈。
- 利用 continue 陳述可使迴圈直接進行下一輪。

試聽課程終於結束囉^^