

程式設計概論  
Programming 101  
一程式的流程控制與邏輯判斷和初階  
迴圈

---

授課老師：邱淑怡

DATE: 3/23/2021



# Outline

---

- 流程圖
- 基本程式結構
- if 簡單結構
- for 簡單結構

# 複習之前的函式(function)用法

---

- print(x)
- type(x)
- a=input("提示說明文字")
- B1, B2 =int(x1), str(x2)

X, x1, x2 → 稱為參數  
B1, B2 → 稱為接收資料的變數



一旦成為變數(variable)，Python 中自行定義的識別字(identifier)，若將物件的字面常數(literal)或定義形式指派給變數，該變數便具有物件的控制權

# 流程圖的功用

---

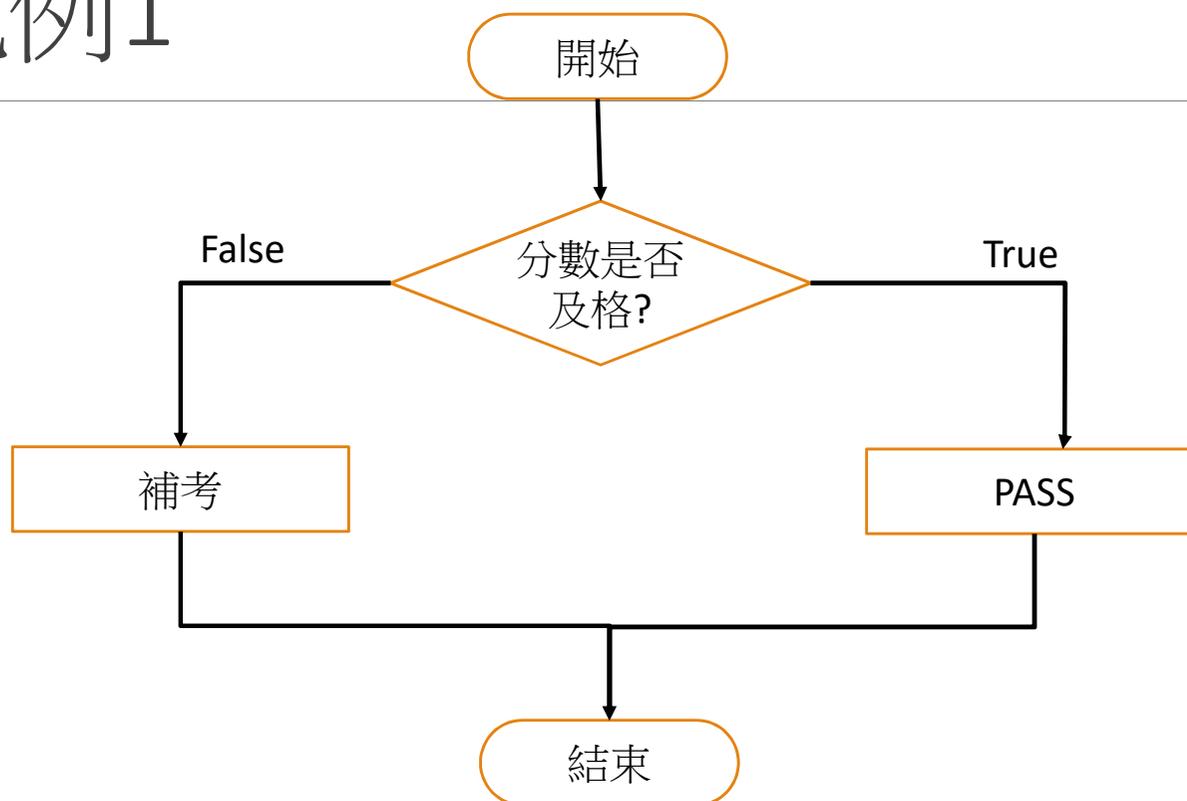
- 以圖形化的呈現方式來摘要程式流程。
- 可在撰寫程式碼前，做為事先規劃的工具。
- 可概覽程式流程的整體結構，做為與他人溝通程式流程的輔助工具。

# 流程圖符號

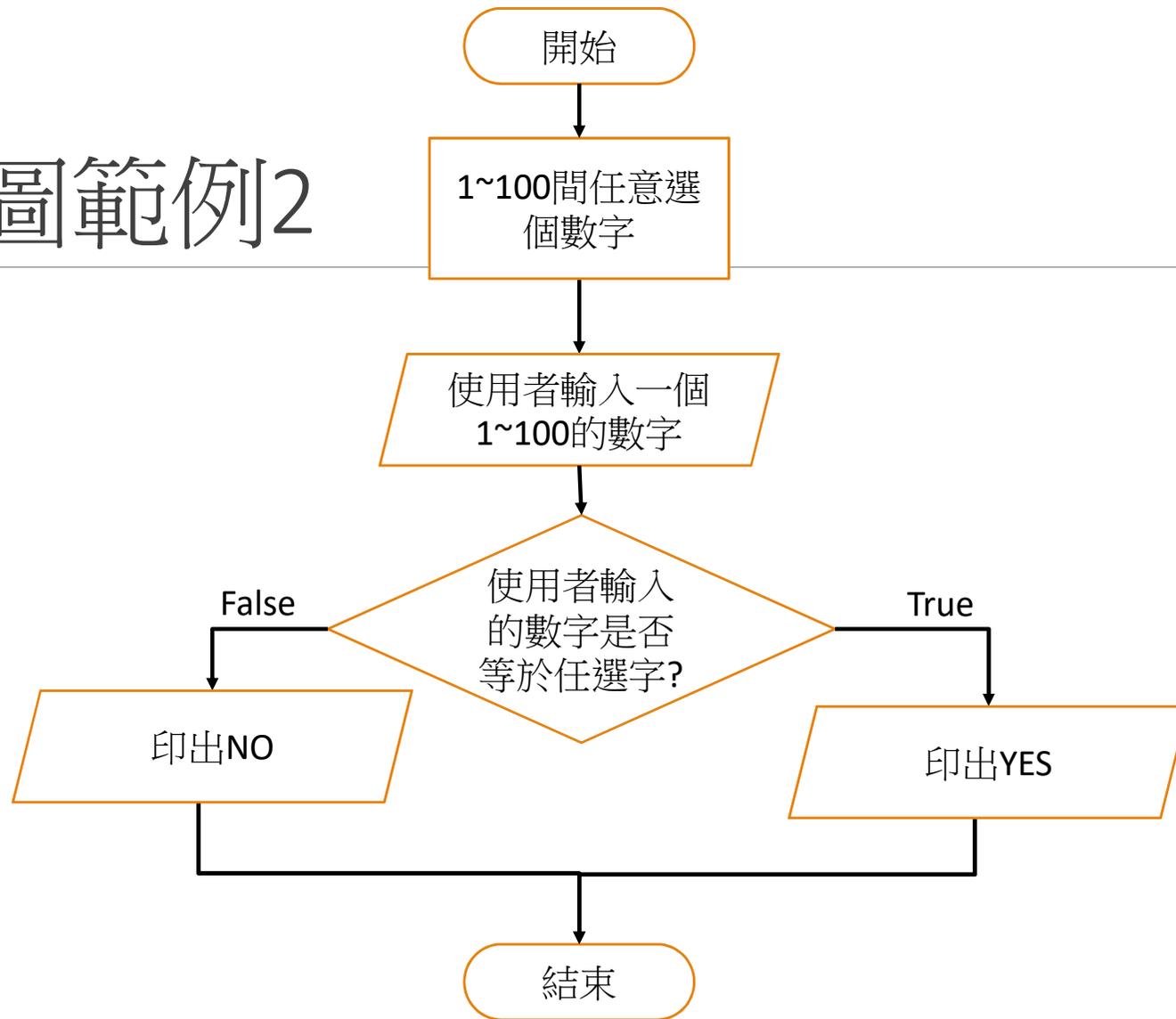
名稱	符號圖形	意義
開始或結束符號		表示程式流程的開始或結束
流程符號		表示程式流程的進行方向
程序處理符號		表示要進行處理的工作
輸入或輸出符號		表示資料輸入或結果輸出
決策判斷符號		表示根據條件式來判斷程式接下來的進行方向
同頁連接點		連接同一頁的流程
跨頁連接點		跨頁連接流程圖



# 流程圖範例1

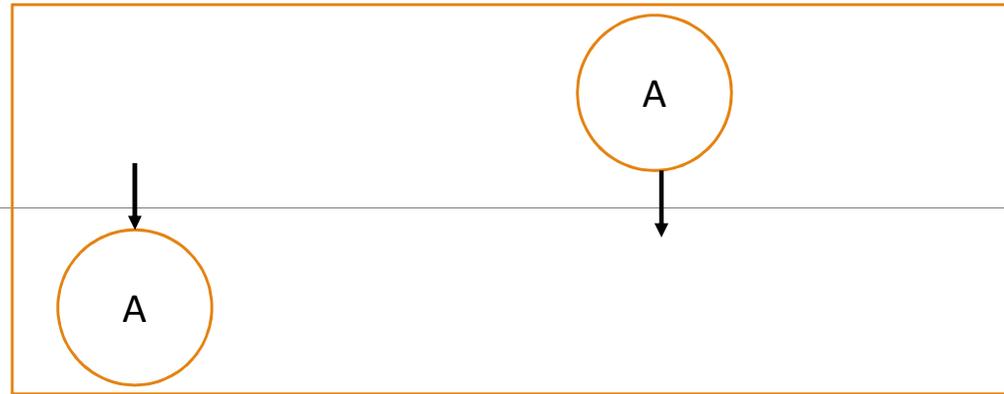


# 流程圖範例2

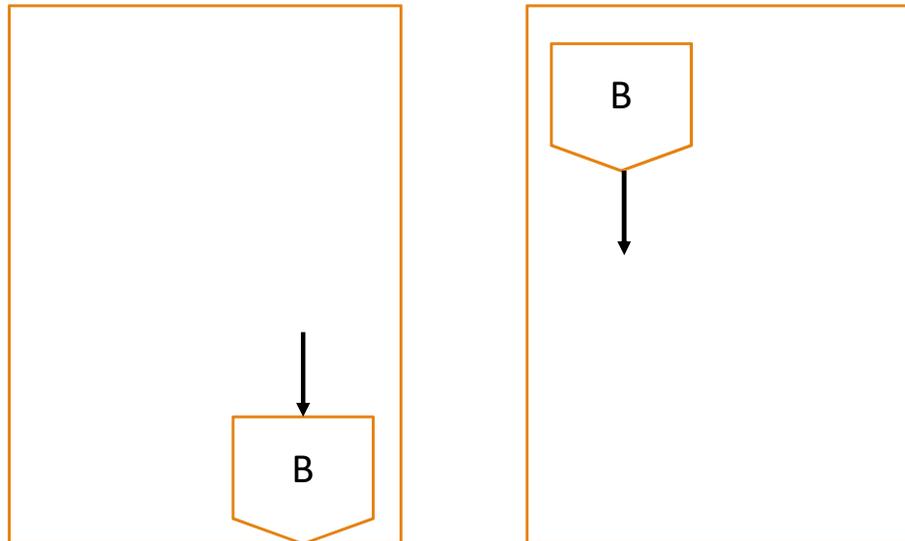


# 連接點

同頁連接點



跨頁連接點



# 三種基本程式結構

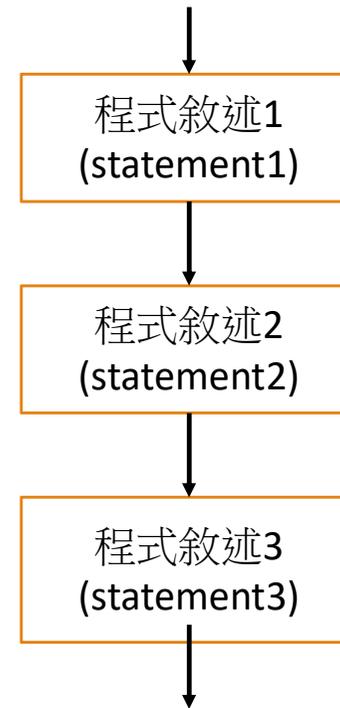
---

1. 循序結構
2. 選擇結構
3. 重複結構

# 1. 循序結構語法

---

- `print('Hello World!')`
- `a= 128//7`
- `print(a+3)`
- `print('Hello'+ 'Everyone')`

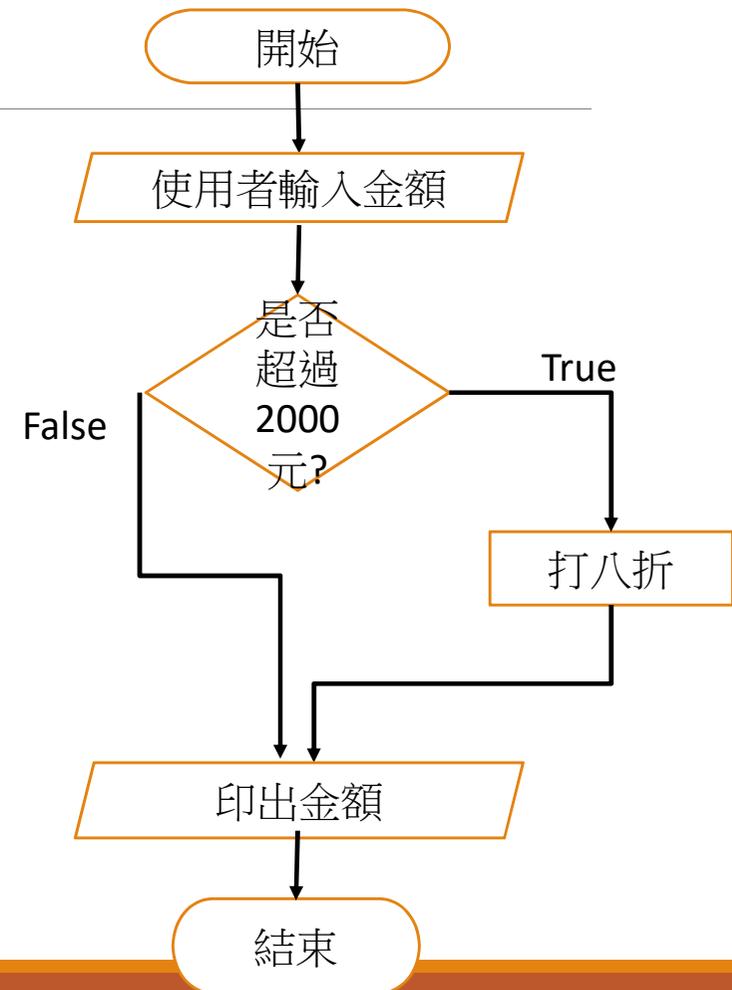
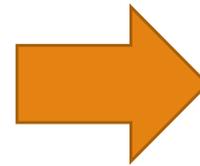


## 2. 選擇結構語法

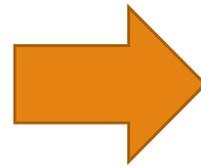
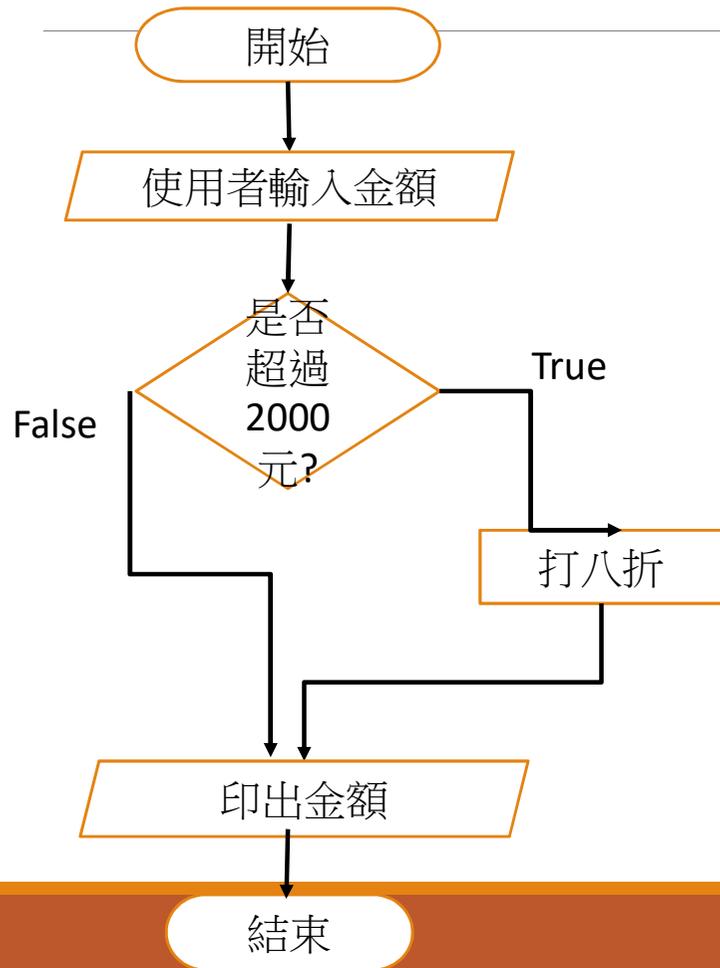
- if 條件運算式:
- 程式敘述一 #可以多筆敘述

題目：

讓使用者輸入金額，是否超過2000元，若超過就金額就打八折，最後印出實付金額。



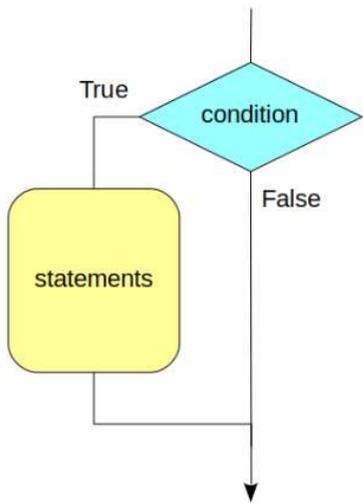
## 2. 選擇結構語法(cont.)



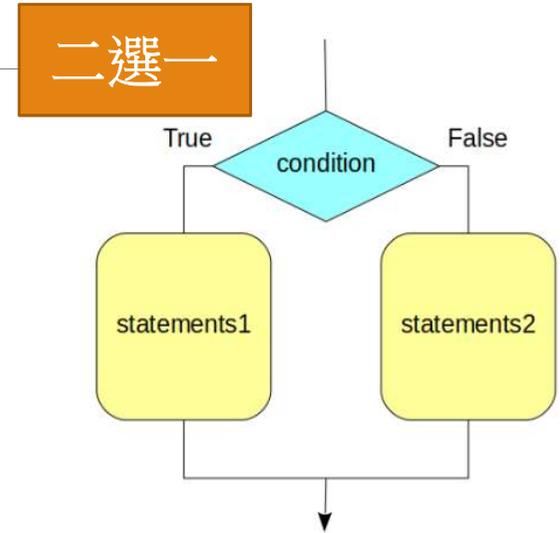
```
Python code:  
money = int(input('輸入金額: '))  
if money > 2000:  
    print('打八折')  
    money *= 0.8 # money=money*0.8  
print("實際付款{:8.2f}".format(money))
```

## 2. 選擇結構語法(cont.)

if <condition(條件運算式)>:  
statements



if <condition>:  
statement1  
else:  
statement2

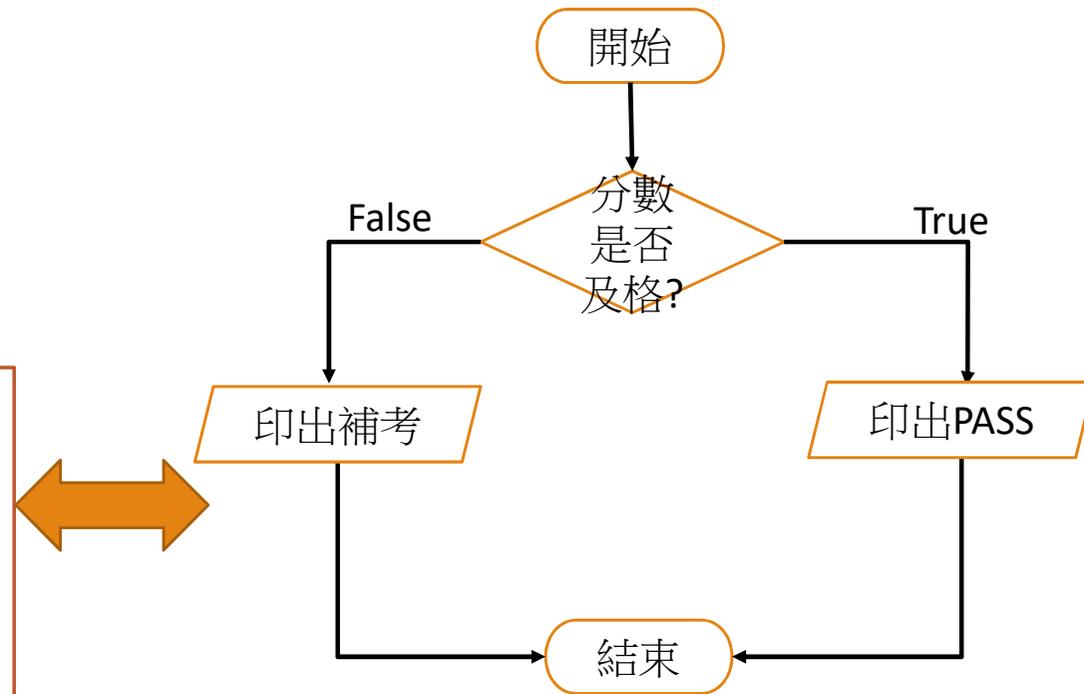


## 2. 選擇結構語法(cont.)

- if 條件運算式:
- 程式敘述一 #可以多筆敘述
- else:
- 程式敘述一 #可以多筆敘述

例子：

```
grade = eval(input('輸入分數: '))  
if grade >= 60:  
    print('PASS')  
else:  
    print("補考")
```



# 條件運算式(condition)說明

□ 「>=」，稱之為條件運算子，可以用來判斷左邊的運算結果，是否大於等於右邊的運算結果

□ 其他的條件運算子

- 大於：「>」
- 等於：「==」
- 小於：「<」

◦ 需注意判斷是否等於的用法是兩個等號(一個等號是變數指派)

- 大於等於：「>=」
- 小於等於：「<=」
- 不等於：「!=」

```
grade = int(input('輸入分數: '))
if grade >= 60:
    print('Pass')
else:
    print("Fail")
```

# if敘述句\_練習題

---

- 請撰寫一個Python程式，它可以要求使用者輸入一個整數，然後檢查該整數若為偶數，就印出“它是偶數”並且顯示使用者所輸入的整數數值，否則就印出“它是奇數”並且顯示使用者所輸入的整數數值。

# 條件運算式說明(cont.)

---

## □ 且：「and」

- 範例：「`a == b and 5 == c`」
- 發現第一個條件不成立時，則不判斷第二個條件

## □ 或：「or」

- 範例：「`x != y or z == 3`」
- 發現第一個條件成立時，則不判斷第二個條件

# 實例練習題

---

- 請撰寫一個Python程式，它可以要求使用者輸入數值，並且判斷是否能整除3而且能整除4，若能整除這兩個數就顯示“正確”；若不是則顯示“不正確”

# for loop 結構

---

# range() function

---

□ range(start, stop[, step]) ，傳入的參數必須是(正、負)整數

- start為起始值(含)，預設值從0開始
- stop為結束值(不含)
- step為遞增值，預設值為1

```
r=range(5)
print(r)
print(list(r))
```

```
print(list(range(10))) # range(0,10)
print(list(range(0,15,5))) #step 5, [0,5,10]
print(list(range(0,10,3))) # [0, 3, 6, 9]
print(list(range(0,-10,-2))) # [0, -2, -4, -6, -8]
```

### 3. 重複結構語法

使用控制變數來控制  
for迴圈的執行次數

- 簡述之，下次上課將深入討論
- 迴圈(loop)中的程式敘述，會依據**控制變數**數值的變化，被重複執行
- for迴圈又稱為“計數迴圈”，而此控制變數則稱為“計數器”
- for **變數** in 串列或任何一串數字或文字:
- 程式敘述    # 可多行程式敘述
- for i in range(1,6):
- print(i)
- print(“完成for迴圈的工作了”)

range(a,b) means (a,a+1,...,b-1)

Use <http://pythontutor.com/>

### 3. 重複結構語法(cont.)

請撰寫一個Python程式，它可以從1加到5，後計算總和？

```
Result=0
for i in range(1,6):
    Result=Result+i # Result+= i
    print(Result, i)
print("完成for迴圈的工作了, Result:", Result)
```

迴圈次數	=左邊的結果	i	=右邊的結果
第1次	0	1	0+1
第2次	1	2	1+2 (3)
第3次	3	3	3+3 (6)
第4次	6	4	6+4 (10)
第5次	10	5	10+5 (15)

# 重複結構 – 重複執行某個動作

---

## □ 練習題

- 若要計算 $1+2+3+\dots+100$ ，印出總和

# 重複結構語法

該變數可視為索引  
值 (index)或計數器



- 迴圈(loop)中的程式敘述，會依據**變數**數值的變化，被重複執行
- for **變數** in 串列或任何一串數字或文字:
  - statement1 # 可多行程式敘述
  - statement2
- for **i** in range(1,10):
  - print(i)
  - print("完成for迴圈的工作了")

# 實例說明

---

□ `for i in range(1,20,3):` #i is the index

□ `print(i)`

□ `print("完成for迴圈的工作1了")`

□ `for i in range(20,1,-2):`

□ `print(i)`

□ `print("完成for迴圈的工作2了")`

# 複習list

---

□ 串列(list)是由一群資料排在一起形成的，定義時必須使用中括號，並在內容之間以逗點隔開；使用時則是依照內容順序，由 0 開始數

□ `a = [12, 24, 46, 18, 90, 145, 576, 791, 133, 673, 700]`

`print(a[4:7])` # a[4] 開始取到 a[7-1]

`print(a[3:8:2])` # a[3] 開始取到 a[8-1]，每 2 個取 1 個

`print(a[-1])`

索引	0	1	2	3	4	5	6	7	8	9	10
內容	12	24	46	18	90	145	576	791	133	673	700
索引	-11	-10	-9	-8	-7	-6	-5	-4	-3	-2	-1

# 用list(串列)資料型態與if, for結合

---

- 串列(list) 中每個元素皆印出其值及對應的索引值
  - `range(len(a)) = range(6) = range(0:6) == range`函式會依序傳回0,1,2,3,4,**5(6-1)**

```
a= [3,6,-7,1,-4,12]
for i in range(len(a)):
    print(i, a[i])
print("完成串列迴圈工作")
```

# 練習題1

---

- 撰寫一個Python程式，令它找出1-100之間可以被13整除的數字，並印出結果

## 練習題2

---

- 搜尋串列  $x = [1, 3, -7, 4, 9, -5, 4]$  當中找出是否有負數，若有請印出它所在的索引值及該數值

# 最後，複習一下

---

- 流程圖
- if ... else
- for 迴圈
- list 與 for, if 結合
- 作業一

