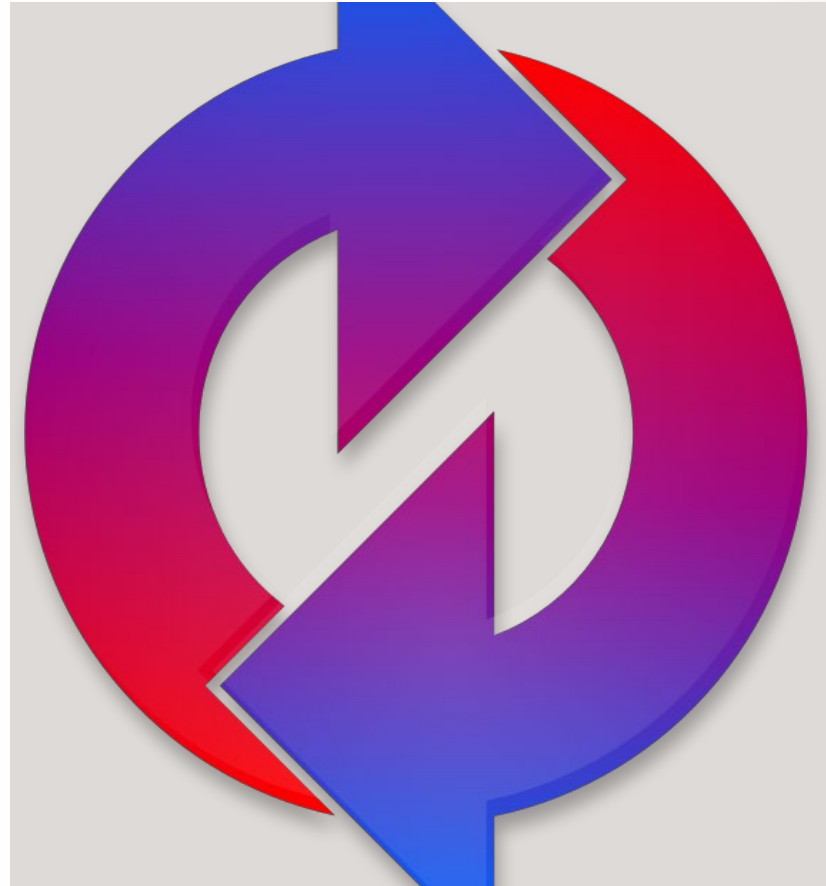




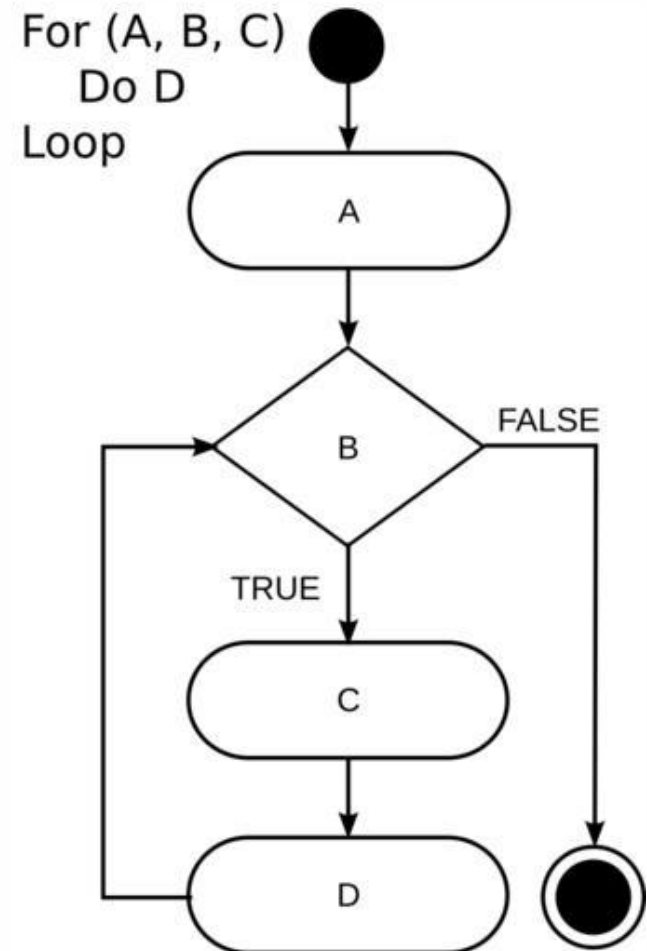
FOR LOOP ITERATIVE STATEMENT

Chih-Chung Hsu (許志仲)
Institute of Data Science
National Cheng Kung University
<https://cchsu.info>



使用時機 & Motivation

- 重複某些繁複的程式碼
- 做數值計算
- 尋找資料
- 排序資料
- 範例：
 - 如果存了一大堆資料，要從裡面找出特定資料
 - 例如分解提到的，翻字典
 - “重複” 做字典拆半 的方式
 - 就是for-loop



Range Function

- 以前的 For loop

- 一個counter
- 一個條件判斷 (like if-else)

```
for i = 1; i < 17; i++  
    print i * i  
end  
  
i = 0  
repeat 16  
    i = i.next()  
    print i * i  
end
```

- Python中，Counter不復存在：range 函式

- Python 迴圈命令有2 個：

- for 迴圈用於執行**固定**次數的迴圈
- while 迴圈用於執行次數**不固定**的迴圈。

- Range 拿來產生 “類似counter” 功能的**整數數列**

- 例如「1,2,3,.....」，每個數列的內容稱為數列的元素，range 函式的功能就是建立整數循序數列。

range 函式的語法

- range 函式單一參數

- range函式使用 1 個參數的語法為：變數 = **range**(正整數值)

- 例如：

```
1 list1 = range(10)    # 產生0, 1, ..., 9 的數字
2 print(list1)
```

range(0, 10)

- range 函式二個參數

- range 函式包含2 個參數的語法為：變數 = **range**(起始, 終止) ##也都整數

- 例如：

```
1 list1 = range(3, 7)    # 產生3, 4, 5, 6 的數字
2 print(list(list1))
```

[3, 4, 5, 6]

- 起始值及終止值皆可為負整數，例如：

```
list1 = range(-3, 0)     # [-3, -2, -1]的數字
print(list(list1))
```

Range 函數的用法 (cont)

- range 函式三個參數

- range 函式包含3 個參數的語法為：變數 = **range**(起始, 終止, 間隔)

- 例如：

```
1 list1 = range(1, 8, 3) #產生1, 4 7 數字序列
2 print(list(list1))
```

[1, 4, 7]

- 負整數間隔值 (由大到小的序列)，起始值必須大於終止值

```
1 list1 = range(7, 0, -3) #產生7, 4, 1 數字序列
2 print(list(list1))
```

[7, 4, 1]

For Loop

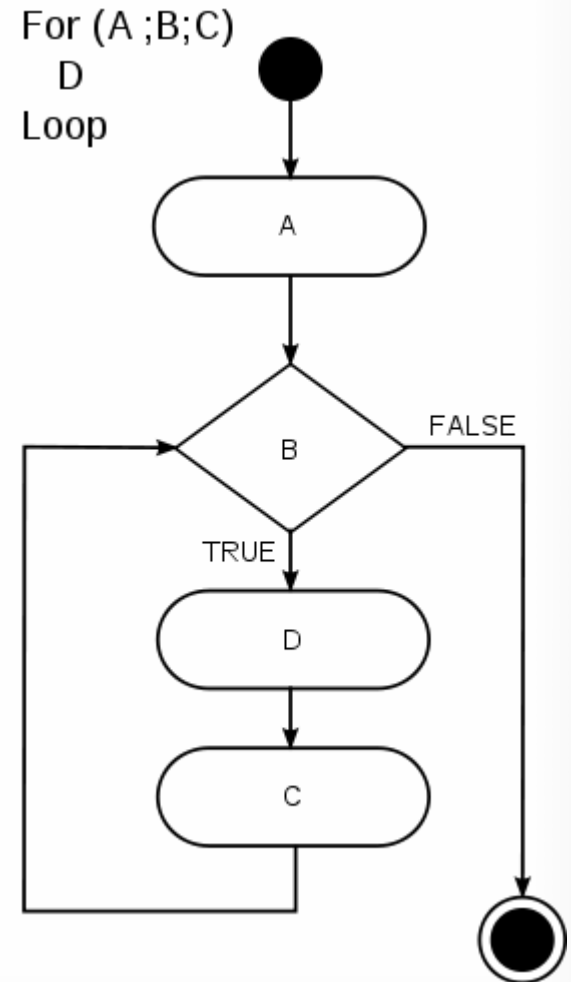
- 以實例解說：

```
for 變數 in range(整數):
```

- 迴圈的流程如圖

- 其實概念在於

- 給定一個要跑幾次的範圍
 - 就像counter的概念
 - 但這個更彈性，是一個數列，所以可以擴增到其他類型的複合資料 (下周講)



實作時間

- 利用For loop 來搭配print，把指定的整數序列給列印出來
 - 輸入：起始值、終止值
 - 輸出：數字序列，間隔為 # 字號

請輸入起始值： 3

請輸入終止值： 9

3#4#5#6#7#8#

請輸入起始值： 22

請輸入終止值： 33

22#23#24#25#26#27#28#29#30#31#32#

巢狀迴圈 Nested For Loop

- 就是 For loop 中還有 For loop

```
1  sum = 0
2  for n in range(10, 20):
3      for z in range(2, 4):
4
5
6
7  print('數列總和為:', sum)
```

20#30#22#33#24#36#26#39#28#42#30#45#32#48#34#51#36#54#38#57#數列總和為: 725

For Loop 進階應用：Break

- break 命令

- 如果有東西在for loop計算過程中已經滿足了，可以提前結束迴圈
- 例如，找資料、計算目標答案等，例如不想看到7的99乘法表

```
4         if j>5:  
5             break  
6         print()
```

1*1= 1	1*2= 2	1*3= 3	1*4= 4	1*5= 5	1*6= 6
2*1= 2	2*2= 4	2*3= 6	2*4= 8	2*5=10	2*6=12
3*1= 3	3*2= 6	3*3= 9	3*4=12	3*5=15	3*6=18
4*1= 4	4*2= 8	4*3=12	4*4=16	4*5=20	4*6=24
5*1= 5	5*2=10	5*3=15	5*4=20	5*5=25	5*6=30
6*1= 6	6*2=12	6*3=18	6*4=24	6*5=30	6*6=36
7*1= 7	7*2=14	7*3=21	7*4=28	7*5=35	7*6=42
8*1= 8	8*2=16	8*3=24	8*4=32	8*5=40	8*6=48
9*1= 9	9*2=18	9*3=27	9*4=36	9*5=45	9*6=54

For Loop 進階應用：Continue

- 當你的 for loop 要 skip 某些東西的時候可以用

```
1  for z in range(66):  
2      if z % 2==0:  
3          continue  
4      print(z, end=',')
```

1, 3, 5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35, 37, 39, 41, 43, 45, 47, 49, 51, 53, 55, 57, 59, 61, 63, 65,

不確定跑幾次：while 迴圈

- 基本語法結構為：

`while` (要跑的條件 [只有True and False]):
跑的程式碼片段

- 應用場域
 - 不確定何時會結束，例如前面的最小公倍數

```
1 target = 6656000000
2 tmp = target
3 digits = 0
4 while (tmp>0):
5     tmp //= 10
6     digits +=1
7
8 print('目標%d共有%d位數' % (target, digits))
9
```

目標6656000000共有10位數

While Loop

- while 迴圈的流程
- 因為最重要的是條件，因此一定要確定，會停止
 - 避免無窮迴圈！
 - 下圖哪裡有無窮迴圈??

```
1 target = 6656000000
2 tmp = target
3 digits = 0
4 while (tmp>0):
5     target //= 10
6     digits +=1
7
8 print('目標%d共有%d位數' % (target, digits))
9
```

