

# Python程式設計

林奇賦 [daky1983@gmail.com](mailto:daky1983@gmail.com)

# 流程控制及迴圈

- If敘述句

- 區塊的表示

- Ex:

```
>>> if 1==1 and 2==2 and 3==3:  
    print(True)
```

True

# 流程控制及迴圈

- 巢狀if敘述句

```
if a == "a":  
    print("Yes!")  
else:  
    if a == b:  
        print("No!")  
    else:  
        print("What?")
```



```
a = 10  
b = 20
```

```
if a > b:  
    print("a is bigger than b")  
elif a == b:  
    print("a is equals to b")  
else:  
    print("b is bigger than a")
```

# 內建函數 input()

- 內建函數 (function) input()，接受使用者的輸入，參數 (parameter) prompt 為提示字串

```
value= input("請輸入:")  
print("value=",value)
```



# import

- 除了內建功能外，Python 的標準模組庫 (standard library) 還有許多已經定義好，並且測試無誤的模組 (module)
- 模組就是已經寫好的 Python 程式檔案，我們在需要的時候使用關鍵字 (keyword) `import` 到我們自己的程式中就可以使用相關定義，同樣的，使用標準模組庫中的模組也要 `import`

# random 模組

- 隨機整數：

```
>>> random.randint(0,100)
```

- 隨機選取0到100間的偶數：

```
>>> random.randrange(0, 100, 2)
```

- 隨機浮點數：

```
>>> random.random()
```

```
>>> random.uniform(1, 10)
```

- 隨機字元：

```
>>> random.choice('abcdefg&#%^*f')
```

- 多個字元中選取特定數量的字元：

```
random.sample('abcdefghij',3)
```

- 隨機選取字串：

```
>>> random.choice ( ['apple', 'pear', 'peach', 'orange', 'lemon'] )
```

- 洗牌：

```
>>> items = [1, 2, 3, 4, 5, 6]
```

```
>>> random.shuffle(items)
```



# 流程控制及迴圈

- if-else敘述句

- Ex:

```
import random
x = input("請輸入一個數字:")
number = random.randint(0,10)
if int(x) >= number:
    print("輸入的文字大於或等於", number)
else:
    print("輸入的文字小於或等於", number)
```

- else敘述的判斷

- if-elif敘述句



# 流程控制及迴圈

- 資料比對運算子

- in與not in運算子

- Ex:

```
>>> strings = "FKRGGgfdaASDGdsf"
```

```
>>> if 's' in strings:  
    print("match")
```

```
match
```

# 流程控制及迴圈

- for迴圈

- 序列變數為tuple型態

- Ex:

- >>> x=[(1,2),(2,3),(3,4)]

- >>> for x1, x2 in x:

- print(x1,x2)

- 

- 1 2

- 2 3

- 3 4

- Ex:

- >>> x=[x\*x for x in range(10)]

- >>> x

- [0, 1, 4, 9, 16, 25, 36, 49, 64, 81]



# 流程控制及迴圈

- for迴圈

- 多個for敘述句進行宣告變數

- Ex:

```
>>> x=[x+y+z for x in '123' if x != '1'  
      for y in 'abc' if y != 'b'  
      for z in 'ABC' if z != 'C']
```

```
>>> x
```

```
['2aA', '2aB', '2cA', '2cB', '3aA', '3aB', '3cA', '3cB']
```

# 流程控制及迴圈

- for迴圈

- 宣告一個序列變數再傳入for敘述句使用

- Ex:

```
>>> LIST=[1,2,3,4,5,6,7,8,9,10]
```

```
>>> sum=0
```

```
>>> for n in LIST:
```

```
    sum+=n
```

```
>>> print("The sum is:", sum)
```

```
The sum is: 55
```



# 課堂練習

- 提供使用者輸入一個變數 $n$ ，並利用for語法算出 $1+2+\dots+n$ 的結果，並將結果印出
- Ex.
  - 請輸入一個數字: 10
  - $sum=55$

# while 迴圈

- 迴圈 (loop) 是用來進行進行重複性的工作，典型的迴圈會進行下列三項基本任務
  - 1. 控制變數初始設定
  - 2. 迴圈結束條件測試
  - 3. 調整控制變數的值
- 關鍵字 (keyword) while 構成 Python 中迴圈的一種，常用於沒有確定重複次數的迴圈，同時 while 後到冒號間的運算式 (expression) 為迴圈結束的條件 (condition) 測試，即為迴圈開始前進行迴圈結束條件的測試，形式如下
  - while condition:



- 由於 **while** 陳述僅需迴圈的結束條件測試，所以有關控制變數 (**variable**) 的初始設定及調整，這都需要放在其它地方。
- 下例計算 1 到 100 所有整數的和，示範 **while** 迴圈的簡單使用

```
sum = 0
i = 1
while i <= 100:
    sum += i
    i += 1

print("1 + 2 + 3 + ... + 98 + 99 + 100 =", sum)
```

# 流程控制及迴圈

- break和continue
  - continue
    - 略過這一次的循環而繼續下一次循環
  - break
    - 中斷迴圈



# continue

```
for i in range(10):  
    if i == 5:  
        continue  
    print(i)
```

# break

```
for i in range(10):  
    if i == 5:  
        break  
    print(i)
```



# 練習迴圈

- 分別用for，while迴圈各寫一個 $n \times n$ 的乘法表
- 程式可以讀取使用者輸入的值  $n$ ,  $n > 1$
- 輸出樣式: ( $n=2$ )
  - $1*1=1$     $1*2=2$
  - $2*1=1$     $2*2=4$
- 若完成上面的練習,試著增加一個功能
  - 亂數選擇一個數字  $r$ , ( $1 \sim n$ )
  - 一樣是印出 $n \times n$ 的乘法表, 但是當 $n=r$  或  $n$ =偶數，將不印出