

人工智慧

Python基礎#1

吳智鴻 教授

國立臺中教育大學 數位內容科技學系

EMAIL: CHWU@MAIL.NTCU.EDU.TW

WEBSITE: 120.108.221.55/PROFCHWU/DCTAI

2022/10/05

學習目標

1. 輸入輸出
2. 迴圈
3. 計算
4. 邏輯判斷式
5. List 串列
6. Dict 字典

Exercise#1

用 Jupyter notebook 寫出讓使用者輸入 n , 顯示計算 $1+2+\dots+n$ 的程式。

```
sum = 0  
n = int (input("請輸入正整數:"))
```

```
for i in range(起始值, 結束值):  
    欲執行的程式碼
```

```
print(" 1 到 %d 的整數和為 %d" % (n, sum))
```

自己練習一下是否能完成
做不出來再看下一頁的解答

你确定要看吗



Exercise#1 解答

```
In [2]: sum = 0  
n = int (input("請輸入正整數:"))
```

請輸入正整數:100

```
In [5]: for i in range(1, n+1):  
        sum += i
```

```
In [6]: print(" 1 到 %d 的整數和為 %d" % (n, sum))
```

1 到 100 的整數和為 10100

Exercise#2

寫出 99 乘法表。

運用雙重迴圈

利用 `for i in range (1,10)`

Why是10?

```
for i in range (1, 10)
```

```
print("%d * %d = %2d " % (i,j,product), end = "")
```

自己練習一下是否能完成
做不出來再看下一頁的解答

你确定要看吗



Exercise #2 解答

```
for i in range (1, 10):  
    for j in range (1,10):  
        product = i * j  
        print("%d * %d = %2d " % (i,j,product), end = "")  
print()
```

1. 記得要加：
2. 程式縮排用 4個空白

```
1 * 1 = 1 1 * 2 = 2 1 * 3 = 3 1 * 4 = 4 1 * 5 = 5 1 * 6 = 6 1 * 7 = 7 1 * 8 = 8 1 * 9 = 9  
2 * 1 = 2 2 * 2 = 4 2 * 3 = 6 2 * 4 = 8 2 * 5 = 10 2 * 6 = 12 2 * 7 = 14 2 * 8 = 16 2 * 9 = 18  
3 * 1 = 3 3 * 2 = 6 3 * 3 = 9 3 * 4 = 12 3 * 5 = 15 3 * 6 = 18 3 * 7 = 21 3 * 8 = 24 3 * 9 = 27  
4 * 1 = 4 4 * 2 = 8 4 * 3 = 12 4 * 4 = 16 4 * 5 = 20 4 * 6 = 24 4 * 7 = 28 4 * 8 = 32 4 * 9 = 36  
5 * 1 = 5 5 * 2 = 10 5 * 3 = 15 5 * 4 = 20 5 * 5 = 25 5 * 6 = 30 5 * 7 = 35 5 * 8 = 40 5 * 9 = 45  
6 * 1 = 6 6 * 2 = 12 6 * 3 = 18 6 * 4 = 24 6 * 5 = 30 6 * 6 = 36 6 * 7 = 42 6 * 8 = 48 6 * 9 = 54  
7 * 1 = 7 7 * 2 = 14 7 * 3 = 21 7 * 4 = 28 7 * 5 = 35 7 * 6 = 42 7 * 7 = 49 7 * 8 = 56 7 * 9 = 63  
8 * 1 = 8 8 * 2 = 16 8 * 3 = 24 8 * 4 = 32 8 * 5 = 40 8 * 6 = 48 8 * 7 = 56 8 * 8 = 64 8 * 9 = 72  
9 * 1 = 9 9 * 2 = 18 9 * 3 = 27 9 * 4 = 36 9 * 5 = 45 9 * 6 = 54 9 * 7 = 63 9 * 8 = 72 9 * 9 = 81
```


Exercise#3 寫出BMI計算式

輸入身高體重

計算BMI並列印在螢幕

分三個等第。過瘦、標準、過重。

```
BMI = 0
height = int (input("請輸入身高 (Cm):"))
weight = int (input("請輸入體重 (Kg):"))
```

自己練習一下是否能完成
做不出來再看下一頁的解答

你确定要看吗



Exercise#3 解答

```
BMI = 0
height = int (input("請輸入身高 (cm):"))
weight = int (input("請輸入體重 (kg):"))
```

```
請輸入身高 (cm):175
請輸入體重 (kg):80
```

```
BMI = (weight) / ((height/100)*(height/100))
print ("BMI = %f\n" % (BMI))
```

```
BMI = 26.122449
```

```
if BMI < 18.5:
    print("你過瘦了!")
if BMI >=18.5 and BMI <24:
    print("體重標準!")
if BMI >= 24:
    print("體重過重!")
```

```
體重過重!
```

資料型態

Python資料型別有以下幾種

1. 數值型態(Numeric type) - int, float, bool, complex
2. 字串型態(String type) - str
3. 容器型態(Container type) - list, set, dict, tuple

List串列

List 串列，可儲存序列資料(可儲不同型別的資料)，相較於其他程式語言的陣列更為強大。

python: List用中括號[]表示，不同的值用逗號分隔。

```
animals = ["bear", "cat", "dog", "elephant"]
number = [1, 2, 3, 4, 5 ]

print (animals)      # 輸出完整的List
print (animals[0])  # 輸出list索引值0的位置 => 第一個元素
print(animals[-3])  # 輸出list索引值-3的位置 => 倒數第三個元素
print (animals[1:3]) # 輸出第二到第三個元素
print (animals[1:])  # 輸出第二到最後一個元素
print (animals[:3]) # 輸出第一到第三個元素
```

結果

```
['bear', 'cat', 'dog', 'elephant']
bear
cat
['cat', 'dog']
['cat', 'dog', 'elephant']
['bear', 'cat', 'dog']
```

[課堂活動]

試著在Jupyter Notebook中完成剛剛程式

List修改內容#1

Append()=>尾端新增一個元素

```
animals.append("fox")
```

Insert() 在任何位置增加元素

```
animals.insert(1,"bird")
```

Del 將該位置的值刪除

```
del animals[2] # 刪除第三個元素
```

結果

```
原本: ['bear', 'cat', 'dog', 'elephant']  
append: ['bear', 'cat', 'dog', 'elephant', 'fox']  
insert: ['bear', 'bird', 'cat', 'dog', 'elephant', 'fox']
```


List修改內容#2

Del() 將該位置的值刪除

```
del animals[2] # 刪除第三個元素
```

Pop() 將尾端的元素刪除

```
animals.pop() # 刪除最尾巴的"fox"
```

Remove() 將某個數值刪除

```
animals.remove("dog") # 刪除"dog"
```

程式範例

```
animals = ['bear', 'bird', 'cat', 'dog', 'elephant', 'fox']
print (animals)

del animals[2] # 刪除第三個元素"cat"
print(animals)

animals.pop() # 刪除最尾巴的"fox"
print(animals)

animals.remove("dog") # 刪除"dog"
print(animals)
```

結果

```
原本:['bear', 'bird', 'cat', 'dog', 'elephant', 'fox']
del:['bear', 'bird', 'dog', 'elephant', 'fox']
pop:['bear', 'bird', 'dog', 'elephant']
remove:['bear', 'bird', 'elephant']
```

List函數

reverse() 反轉數值

```
animals.reverse()
```

sort() 排序

```
animals.sort()
```

Index() 某數值在List第一次出現的位置

count() 某個數值在List出現的次數

程式範例

```
animals = ['bear', 'cat', 'dog', 'elephant', 'cat']
print(animals.index('cat'))
print(animals.count('cat'))

animals.reverse()
print(animals)

animals.sort()
print(animals)
```

結果

```
index "cat" :1
count "cat" : 2
reverse : ['cat', 'elephant', 'dog', 'cat', 'bear']
sort : ['bear', 'cat', 'cat', 'dog', 'elephant']
```

Dict 字典

key-map的形式

dict 字典，和串列很像但不會使用 0,1,...等的序列，index 來選擇項目，須宣告key鍵值來對應儲存的值。

與LIST不同
LIST用索引位置[index]，DICT用{key, value}

LIST

```
animals = ["bear", "cat", "dog", "elephant"]  
number = [1, 2, 3, 4, 5]
```

DICT

```
fruits = {"apple":10, "banana":7, "orange":8}  
print(fruits["banana"])
```

結果

```
7
```

keys 、 values 、 items

keys() => 取得所有key組合，資料型態為dict_keys

values() => 取得所有value組合，資料型態為dict_values

items() => 取得所有 (key,value) 組合，資料型態為dict_items

建立DICT的程式範例#1

```
fruits = {"apple":10, "banana":7, "orange":8}

print(fruits.keys())    #keys()
print(fruits.values())  #values()
print(fruits.items())  #items()
```

建立DICT的程式範例#2

```
dict_keys(['apple', 'banana', 'orange'])
dict_values([10, 7, 8])
dict_items([('apple', 10), ('banana', 7), ('orange', 8)])
```

結果

```
dict_keys(['apple', 'banana', 'orange'])
dict_values([10, 7, 8])
dict_items([('apple', 10), ('banana', 7), ('orange', 8)])
```

參考資料來源

<https://ithelp.ithome.com.tw/articles/10219098>

Part #1 Finish
