



## PREFACE

## 序

Python 是簡潔而強大的程式語言，具有豐富的第三方程式庫和工具，這讓程式設計變得更加容易和有趣。但是，初學者在學習 Python 時，常常會遇到許多困難和挫折。因此，我們撰寫了這本易於理解的 Python 學習書，以便於 Python 程式設計的學習。

希望能幫助讀者快速入門 Python 程式設計，並且將概念講解清楚，範例具有實用性且有趣味性。期許讀者在閱讀完本書後，具備完整的 Python 程式設計概念和能力，並且能夠自己寫出程式來。

我們建議以下的研讀模式：

1. 全書瀏覽：了解內容大綱，熟悉本書的結構和主題。
2. 各章節詳讀：詳細閱讀各章節內容，照著範例實際操作，熟悉指令、語法、資料結構及邏輯。
3. 程式實作：不看書上的程式碼，自己嘗試實作範例，這是初學者較無挫折感的學習方式。閱讀過的程式，再試著把它寫出來，以訓練自己寫程式的能力。
4. 習題練習：進一步學習，嘗試從習題的練習中，了解問題、分析問題、設計解決問題的步驟，以進一步提升自己的能力。
5. 進階學習：如果您很確實的完成前面四個步驟，您應該已經掌握了 Python 的基礎知識。現在可以透過閱讀官方文件和利用網路資源，進一步提高自己的能力。
6. 學習資源 ChatGPT：當你有特定問題或疑問時，可以使用 ChatGPT 來獲取有關 Python 的解答和建議。

這本書的完成，要感謝很多好朋友，基峰資訊 Jessi 及眾多夥伴們，從這本書的寫作計劃開始，一路上給我很多協助與寶貴意見；寫作期間，家人的體諒與支持，使得這本書得以順利完成，也都是我要感謝的。

吳進北 2023/05

# 4

## CHAPTER



# 為程式加入反覆 執行功能

- for 迴圈
- for i in range
- while 迴圈
- do-while 迴圈
- 巢狀迴路
- 單迴路多變數的應用
- 終止迴圈的執行
- 跳過本次迴圈剩下的程式碼

程式語言中的迴路（loop）是一種重複執行程式區塊的功能，讓我們在寫程式時更有效率地處理重複工作。

迭代（iteration）是重複回饋過程的活動，其目的通常是為了接近並且到達所需的目標或結果。每一次對過程的重複被稱為一次「迭代」，而每一次迭代得到的結果會被用來作為下一次迭代的初始值。

很多書把迴路中，重複執行程式區塊動作，稱為迭代。

## 4-1 for 迴圈

for 迴圈用於重複序列（例如串列、元組或字串）或其他可重複對象中的元素。語法如下：

```
for 變數 in 序列:  
    # 重複操作
```

其中 序列 是可重複對象，變數 是每次重複時將被賦值的變數。

### 範例 透過 for 迴圈列印出串列中的所有元素

```
fruits = ["apple", "banana", "cherry"]  
for fruit in fruits:  
    print(fruit)  
  
for i in ['嘉義市', '中埔鄉', '太保市', '水上鄉']:  
    print('我是', i, '人')  
  
for i in '春夏秋冬':  
    print('我喜歡', i, '天')
```

### 範例 使用 for 迴圈計算串列中所有元素的總和

```
numbers = [1, 2, 3, 4, 5]  
total = 0  
for num in numbers:  
    total += num  
print("總和為:", total)
```

**範例** 透過 for 迴圈將串列中的元素轉換為字串

```
numbers = [1, 2, 3, 4, 5]
number_strs = []
for num in numbers:
    number_strs.append(str(num))
```

## 4-2 for i in range

`for i in range()` 是 Python 中用來追蹤數字序列的一個常見迴圈構造。`range()` 函數可以生成一個整數序列，該序列通常用於循環一定次數的操作。基本語法如下：

```
for i in range(stop):
    # 循環內容
```

其中，`stop` 參數是必需的，指定了整數序列的停止值（但不包括停止值本身）。

**範例** 計數循環

這個範例展示了如何使用 `for i in range()` 指令來執行循環。

```
for i in range(5):
    print(i)
```

**# 執行結果**

```
0
1
2
3
4
```

**範例** 自行定義間隔

這個範例展示了如何使用 `range()` 函數的 `start` 和 `step` 參數來自行定義間隔。

```
for i in range(0, 10, 2):
    print(i)
```

### # 執行結果

```
0
2
4
6
8
```

#### 範例 反向循環

這個範例展示了如何使用 `range()` 函數的負間隔來反向循環。

```
for i in range(5, 0, -1):
    print(i)
```

### # 執行結果

```
5
4
3
2
1
```

#### 範例 嵌套循環

這個範例展示了如何使用嵌套 `for i in range()` 循環。

```
for i in range(3):
    for j in range(2):
        print(i, j)
```

### # 執行結果

```
0 0
0 1
1 0
1 1
2 0
2 1
```

**範例** 串列追蹤

這個範例展示了如何使用 `for i in range()` 循環追蹤串列元素。

```
fruits = ['apple', 'banana', 'cherry']
for i in range(len(fruits)):
    print(fruits[i])
```

**# 執行結果**

```
apple
banana
cherry
```

# 6

## CHAPTER



# 複雜資料結構的處理

- 串列 (List)
- 元組 (Tuples)
- 集合 (Sets)
- 字典 (Dictionaries)
- 堆疊 (Stack)
- 佇列 (Queue)
- 堆 (Heap)
- 字串 (String)
- 樹 (Tree)

當資料量變多，或結構變複雜，我們需要多一點技巧及方法來處理。

## 6-1 串列 (Lists)

在 Python 中，List 是一種資料結構，它是有序的集合，允許儲存不同型態的元素，例如整數、浮點數、字串、元組等。List 是可變的物件，這意味著您可以添加、刪除和修改其中的元素。

- 顏色清單：colors = ['red', 'green', 'blue', 'yellow', 'purple']
- 交通工具清單：vehicles = ['car', 'bus', 'train', 'bike', 'boat']

**範例** 建立 List：可以使用中括號 [] 或 list() 函數來建立 List。

```
a = [1, 2, 3]
```

**範例** 訪問 List 元素：可以使用索引來訪問 List 中的元素。

a[0] 表示列出 List a 中的第一個元素。

**範例** 更新 List 元素：可以使用索引來更新 List 中的元素。

a[0] = 4 表示將 List a 中的第一個元素更新為 4。

**範例** 刪除 List 元素：可以使用 del 關鍵字刪除 List 中的元素。

del a[0] 表示刪除 List a 中的第一個元素。

**範例** 添加 List 元素：可以使用 append() 方法向 List 中添加元素。

a.append(4) 表示向 List a 中添加值為 4 的元素。

**範例** 插入 List 元素：可以使用 insert() 方法在 List 中插入元素。

a.insert(1, 5) 表示在 List a 的第二個位置插入值為 5 的元素。

**範例** 切片 List：可以使用切片來選擇 List 中的一部分。

a[1:3] 表示選擇 List a 中的第一個到第二個元素。



**範例** 追蹤 List：可以使用 for 循環來追蹤 List 中的所有元素。

```
for i in a:
    print(i)
```

## list 物件方法（method）整理

方法	說明
<code>list.append(x)</code>	將新的項加到 list 的尾端
<code>list.extend(iterable)</code>	將另 list 接到 list 的尾端
<code>list.insert(i, x)</code>	將項目插入至 list 中給定的位置。第引數為插入處前元素的索引值
<code>list.remove(x)</code>	刪除 list 中第一個值等於 x 的元素。
<code>list.pop([i])</code>	移除 list 中給定位置的項目，並回傳它。
<code>list.clear()</code>	刪除 list 中所有項目。
<code>list.index(x[, start[, end]])</code>	回傳 list 中第一個等於 x 的項目之索引值（從零開始的索引）。若 list 中無此項目，則丟出 ValueError 錯誤。
<code>list.count(x)</code>	回傳 x 在 list 中所出現的次數。
<code>list.sort(*, key=None, reverse=False)</code>	將 list 中的項目排序。（可使用引數來進行客製化的排序）
<code>list.reverse()</code>	將 list 中的項目前後順序反過來。
<code>list.copy()</code>	回傳淺複製 (shallow copy) 的 list。

## 6-2 元組（Tuples）

Python 的 tuple（元組）是一種不可變的序列資料類型，類似於 list（串列），但元組在定義後就不能被修改。和 list 一樣，tuple 內的元素可以是不同的資料類型，如整數、浮點數、字串、布林值等等，它們可以使用索引來訪問。tuple 可以透過括號來建立，也可以省略括號。例如：

**範例** 包含不同種類水果的元組

```
fruits = ("apple", "banana", "orange")
```

**範例** 包含不同數字的元組

```
numbers = (1, 2, 3, 4, 5)
```

**範例** 包含不同類型資料的元組

```
data = ("Mary Lee", 30, True, ["apple", "banana"])
```

元組允許嵌套，即在一個元組內部可以再包含另一個元組。元組還支持一些常用的內建函數，如 `len()`、`max()`、`min()` 等。元組的不可變性使得它們更安全，更適合用於不需要修改內容的資料。由於元組的訪問速度比 `list` 更快，因此在不需要修改資料的情況下，元組也更加有效率。

## 6-3 集合 (Sets)

Python 中的 `set` (集合) 是一個無序且不重複的集合數據結構，用於存儲可哈希 (hashable) 對象的集合。可以將 `set` 看作是字典 (dictionary) 的鍵 (key) 的集合，但是 `set` 中不存儲值 (value)，只存儲鍵 (key)。

`set` 具有以下特點：

1. 無序：集合中的元素沒有特定的順序，因此無法透過索引訪問集合中的元素。
2. 不重複：集合中不允許有重複的元素。
3. 可變：可以向集合中添加或刪除元素。
4. 可迴路追蹤：可以用迴路列出集合中的所有元素。

`set` 支持以下操作：

方法名稱	說明
<code>add()</code>	添加一個元素。
<code>remove()</code>	刪除一個元素，如果元素不存在則引發 <code>KeyError</code> 異常。
<code>discard()</code>	刪除一個元素，如果元素不存在則不會引發異常。
<code>pop()</code>	隨機刪除一個元素，如果集合為空則引發 <code>KeyError</code> 異常。
<code>clear()</code>	刪除所有元素。

方法名稱	說明
union()	返回兩個集合的聯集。
intersection()	返回兩個集合的交集。
difference()	返回兩個集合的差集。
symmetric_difference()	返回兩個集合的對稱差集。
issubset()	檢查一個集合是否是另一個集合的子集。
issuperset()	檢查一個集合是否是另一個集合的超集。

由於 `set` 是一個數據結構，因此可以使用 `for` 循環列出集合中的所有元素，也可以使用 `in` 關鍵字檢查集合中是否包含某個元素。

#### 範例 建立空集合

```
empty_set = set()
print(empty_set) # set()
```

#### 範例 建立包含多個項目的集合

```
fruits = {"apple", "banana", "orange", "pear"}
print(fruits) # {'apple', 'banana', 'orange', 'pear'}
```

#### 範例 將串列轉換為集合

```
numbers = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
number_set = set(numbers)
print(number_set) # {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

集合是無序的，所以印出來的項目排列不一定與建立時相同。第三個範例則建立了數字串列，然後將其轉換為集合，這可以透過 `set()` 函數實現。

應用例子如下：

#### 範例 次數統計

```
a = [1,2,3,5,6,8,9,1,5,9,1,1,5]
# print(a)
b = set(a) # 使 b 內含元素，不重複
# print(b)
```

```
for i in b:
    print(i,':',a.count(i))
```

### # 執行結果

```
# 1 : 4
# 2 : 1
# 3 : 1
# 5 : 3
# 6 : 1
# 8 : 1
# 9 : 2
```

a.count(i) 算出 a 串列中有幾個 i 參數所代表的元素。

**範例** 賈柏斯在史丹佛的演講講詞約 2200 個字，但用的單字量是多少？

```
a = '''
演講講詞 2200 個字貼這裡
'''
b = [i for i in a.split() if i>='A' and i <='z']
c = set(b)
for i in b:
    print(i,':',b.count(i))
print('賈柏斯總共用了',len(c),'個單字')
```

### # 執行結果

```
# current : 1
# renaissance. : 1
# And : 17
# ...
# you : 25
# all : 16
# very : 8
# much. : 1
# 賈柏斯總共用了 832 個單字
```

說明：用 b = [i for i in a.split() if i>='A' and i <='z']，取出打散後的每一個單字。

# 13

## CHAPTER



# 程式設計比賽試題 參考題實作

- 解題標準結構說明
- 磅數公斤
- 整數商餘
- 四數有權重相加
- 華氏轉攝氏
- 錢
- BMI
- 所有位數值平方和
- 快樂數
- 完美數
- 噁爛數
- 阿姆斯特壯數
- 重複文字只保留第一次出現者
- 質因數
- 輸出星期幾
- 四數有權重相加再算費波那契數
- 漢明距離
- 排序 (Sort) 練習
- 氣泡排序 (Bubble Sort)
- 二維矩陣
- 二維矩陣 II
- OX 棋

比賽對於程式設計學習者，是一個可以展示技能和創造力的機會，並能與其他有才華的人一起競爭。參與比賽也可以幫助他們學習新技術和挑戰自己的限制，進而提高自己的技能和知識水平。

觀摩比賽選手的解題技巧，也是提昇程式設計能力的好方法。我們就由程式設計比賽參考題開始吧！

## 13-1 解題標準結構說明

```
while True:
    try:
        ...
    except:
        break
```

程式碼中的 `while True` 代表無限循環，因為其條件為 `True`，所以會一直循環下去。如果 `try` 塊中的代碼出現異常，則會跳轉到 `except` 塊中，使用了 `break` 關鍵字來終止 `while` 循環，因此程式會退出。

## 13-2 磅數公斤

由使用者輸入磅數，請轉為公斤數再輸出。

註：(1) 1 磅=0.454 公斤（1 磅≈0.45359237 公斤）（大約）。

(2) 輸入值可能有小數，不會是負數。

(3) 輸出值小數留 2 位，四捨五入。

提示：num 取小數 2 位，四捨五入

```
num = round(num, 2)
```

## Input

1	10
2	23
3	10.2
4	23.88
5	23.87
6	100
7	1000
8	10000
9	100000

## Output

1	4.54
2	10.44
3	4.63
4	10.84
5	10.84
6	45.4
7	454.0
8	4540.0
9	45400.0

資料來源:<https://zerojudge.ntub.tw/team>

```
while True:
    try:
        r = 0.454
        p = float(input())
        k = round(r*p,2)
        print(k)
    except:
        break
```

在 `try` 塊中，`r` 被賦值為 `0.454`，`p` 為使用者輸入的數值，這裡使用了 `input()` 函數來獲取使用者輸入的值，該函數返回的是一個字串，因此需要使用 `float()` 函數將其轉換為浮點數類型。接下來，`k` 被賦值為 `r*p` 的乘積，並使用 `round()` 函數將其四捨五入到小數點後兩位。最後，使用 `print()` 函數輸出 `k` 的值。

## 13-4 四數有權重相加

輸入：四個數字  $abcd$  有權重的相加 輸出： $56*a + 24*b + 14*c + 6*d$

### 輸入

輸入有若干列，每列為一組試資料。輸入的每一列有四個整數。輸入以 EOF 作為結束。

### 輸出

對於所輸入的每一列有四個整數，要各別輸出一列，依公式有權重的相加。

### 範例輸入輸出

#### 範例輸入|

```
1 | 1 1 1 1
2 | 1 10 100 1000
```

#### 範例輸出|

```
1 | 100
2 | 7696
```

資料來源：<https://zerojudge.ntub.tw/team>

```
while True:
    try:
        a,b,c,d = map(int,input().split())
        s = a*56+b*24+c*14+d*6
        print(s)
    except:
        break
```

1. 用 `input()` 函數，等待使用者輸入數據，使用者輸入的數據會以空格分隔的形式被讀入。
2. 使用 `map()` 函數將輸入的四個值轉換成整數型別並儲存在變數 `a`、`b`、`c`、`d` 中。
3. 計算 `s` 的值， $s = a * 56 + b * 24 + c * 14 + d * 6$ 。
4. 輸出 `s` 的值。



# 14

## CHAPTER



# APCS 大學程式設計 先修檢測

- 最大和
- 成績指標
- 邏輯運算子
- 小群體
- 特殊編碼
- 完全奇數
- 定時 K 彈
- 秘密差
- 線段覆蓋長度
- 數字龍捲風
- 矩陣轉換
- 棒球遊戲

## 14-7 定時 K 彈

### 問題描述

「定時 K 彈」是一個團康遊戲，N 個人圍成一個圈，由 1 號依序到 N 號，從 1 號開始依序傳遞一枚玩具炸彈，炸彈每次到第 M 個人就會爆炸，此人即淘汰，被淘汰的人要離開圓圈，然後炸彈再從該淘汰者的下一個開始傳遞。遊戲之所以稱 K 彈是因為這枚炸彈只會爆炸 K 次，在第 K 次爆炸後，遊戲即停止，而此時在第 K 個淘汰者的一位遊戲者被稱為幸運者，通常就會被要求表演節目。例如  $N=5$ ， $M=2$ ，如果  $K=2$ ，炸彈會爆炸兩次，被爆炸淘汰的順序依序是 2 與 4（參見下圖），這時 5 號就是幸運者。如果  $K=3$ ，剛才的遊戲會繼續，第三個淘汰的是 1 號，所以幸運者是 3 號。如果  $K=4$ ，下一輪淘汰 5 號，所以 3 號是幸運者。

此題輸入 N、M 與 K，請你計算出誰是幸運者。



### 輸入格式

輸入只有一行包含三個正整數，依序為 N、M 與 K，兩數中間有一個空格分開。其中  $1 \leq K < N$ 。

### 輸出格式

請輸出幸運者的號碼，結尾有換行符號。

範例一：輸入

5 2 4

範例一：正確輸出

3

(說明)

被淘汰的順序是 2、4、1、5，此時 5 的下一位是 3，也是最後剩下的，所以幸運者是 3。

範例二：輸入

8 3 6

範例二：正確輸出

4

(說明)

被淘汰的順序是 3、6、1、5、2、8，此時 8 的下一位是 4，所以幸運者是 4。

評分說明

輸入包含若干筆測試資料，每一筆測試資料的執行時間限制 (time limit) 均為 1 秒，依正確通過測資筆數給分。其中：

第 1 子題組 20 分， $1 \leq N \leq 100$ ，且  $1 \leq M \leq 10$ ， $K = N - 1$ 。

第 2 子題組 30 分， $1 \leq N \leq 10,000$ ，且  $1 \leq M \leq 1,000,000$ ， $K = N - 1$ 。

第 3 子題組 20 分， $1 \leq N \leq 200,000$ ，且  $1 \leq M \leq 1,000,000$ ， $K = N - 1$ 。

第 4 子題組 30 分， $1 < N < 200,000$ ，且  $1 < M < 1,000,000$ ， $1 < K < N$ 。

```
# n 人數
# m 第 m 個人
# k 第 k 次
n,m,k = map(int,input().split())
# n = 5
# m = 2
# k = 4

# 建立一個包含 1 到 10 的串列 a
a = [i+1 for i in range(10)]

# 初始化變數 i 為 1
i = 1
# 當 k 大於等於 1 時進入循環
while (k>=1):
    # 取出 a 中的第一個元素
    t = a.pop(0)
    # 如果 i 是奇數，則把取出的元素加到 a 的末尾
    if i%2!=0:
        a.append(t)
    # 否則，k 減 1 表示已經取出了一個元素
    else:
        k-=1
    # i 加 1，表示已經取出了一個元素
    i+=1

# 最後剩下的一個元素即為要找的數字
print(a[0])
```