Python程式設計：陣列

**目錄**

[Python程式設計：資料型態(Data type) 1](#_Toc205736778)

[1. 查詢資料型態：type() 2](#_Toc205736779)

[2. 整數與浮點數 2](#_Toc205736780)

[2.1 整數（int） 2](#_Toc205736781)

[2.2 浮點數（float） 2](#_Toc205736782)

[2.3 不同進位制的整數 3](#_Toc205736783)

[2.4 強制資料型態轉換 3](#_Toc205736784)

[2.5 常用數值運算函數 3](#_Toc205736785)

[3. 布林值Boolean 3](#_Toc205736786)

[4. 字串（String）與字元（char） 4](#_Toc205736787)

[4.1 字串（String） 4](#_Toc205736788)

[4.2 字串連接 4](#_Toc205736789)

[4.3 多行字串 4](#_Toc205736790)

[4.4 逸出字元（Escape Characters） 4](#_Toc205736791)

[4.5 強制資料型態轉換 4](#_Toc205736792)

[4.6 字串乘法 5](#_Toc205736793)

[4.7 原始字串 r 5](#_Toc205736794)

[4.8 字元與編碼：ASCII 5](#_Toc205736795)

[5. 二進位資料（byte） 5](#_Toc205736796)

[5.1 範例：字串轉 bytes（encode()） 5](#_Toc205736797)

[5.2 範例：bytes 轉字串（decode()） 6](#_Toc205736798)

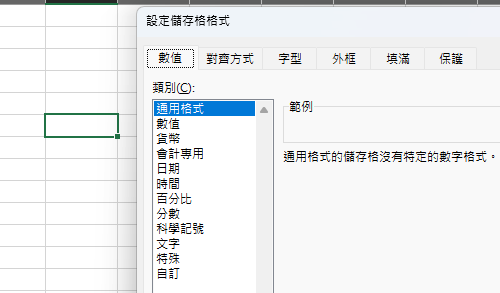
[5.3 練習：看電影要等多久 6](#_Toc205736799)

[5.4 練習：高鐵旅行時間計算 6](#_Toc205736800)

[5.5 練習：公園散步距離 6](#_Toc205736801)

[5.6 練習：外送員送餐時間 6](#_Toc205736802)

[5.7 練習：兩家咖啡廳的距離 7](#_Toc205736803)



串列是 Python 中一種最常見的資料型態，可以儲存多個不同型態的元素，例如數字、字串、布林值等。

# 定義串列

## # 定義一個數字串列

james = [23, 19, 22, 31, 18]

# 定義一個字串串列

fruits = ['apple', 'banana', 'orange']

# 混合型態的串列

leits = [3.5, 6.0, 8.5, 'leits']

# 建立一個空串列，之後可以再加入元素

new\_list = []

# 讀取串列元素

你可以透過索引 (index) 來取得串列中的特定元素。索引從 0 開始。

* x[0]：取得第一個元素
* x[1]：取得第二個元素
* x[-1]：取得最後一個元素

james = [23, 19, 22, 31, 18]

print(james[0]) # 取得第一個元素

print(james[2]) # 取得第三個元素

print(james[-1]) # 取得最後一個元素

# 串列切片 (list slices)

切片可以讓你取得串列中的部分元素，而非單一元素。切片語法為 x[start:end:step]。

* start：開始索引 (包含)
* end：結束索引 (不包含)
* step：步長，每次跳過多少個元素

warriors = ['Curry', 'Durant', 'Iguodala', 'Bell', 'Thompson']

# 取得索引 0 到 3 (不包含) 的元素

print(warriors[0:3]) # 輸出: ['Curry', 'Durant', 'Iguodala']

# 取得索引 2 到最後的元素

print(warriors[2:]) # 輸出: ['Iguodala', 'Bell', 'Thompson']

# 取得所有元素

print(warriors[:]) # 輸出: ['Curry', 'Durant', 'Iguodala', 'Bell', 'Thompson']

# 從頭到尾，每隔一個元素取一個

print(warriors[::2]) # 輸出: ['Curry', 'Iguodala', 'Thompson']

# 反轉串列

print(warriors[::-1]) # 輸出: ['Thompson', 'Bell', 'Iguodala', 'Durant', 'Curry']

# 串列統計函式

Python 提供了一些內建函式，可以對串列進行統計。

* max(x)：回傳串列中的最大值
* min(x)：回傳串列中的最小值
* sum(x)：回傳串列中所有元素的總和 (僅限數字串列)
* len(x)：回傳串列中元素的數量

james = [23, 19, 22, 31, 18]

print(f"最大值: {max(james)}") # 輸出: 31

print(f"最小值: {min(james)}") # 輸出: 18

print(f"總和: {sum(james)}") # 輸出: 113

print(f"元素數量: {len(james)}") # 輸出: 5

# 增加串列元素

## 在串列末端增加元素 append()

cars = ['Honda', 'Toyota', 'Ford']

cars.append('Nissan')

print(cars) # 輸出: ['Honda', 'Toyota', 'Ford', 'Nissan']

## 插入串列元素 insert()

cars = ['Honda', 'Toyota', 'Ford']

# 在索引 1 的位置插入 'BMW'

cars.insert(1, 'BMW') # 插入位置、差入元素

print(cars) # 輸出: ['Honda', 'BMW', 'Toyota', 'Ford']

# 刪除元素

## 刪除**串列**元素 pop()

cars = ['Honda', 'Toyota', 'Ford', 'Nissan']

# 刪除最後一個元素

popped\_car = cars.pop()

print(f"被刪除的元素: {popped\_car}") # 輸出: 被刪除的元素: Nissan

print(f"新的串列: {cars}") # 輸出: 新的串列: ['Honda', 'Toyota', 'Ford']

# 刪除索引為 1 的元素

popped\_car = cars.pop(1)

print(f"被刪除的元素: {popped\_car}") # 輸出: 被刪除的元素: Toyota

print(f"新的串列: {cars}") # 輸出: 新的串列: ['Honda', 'Ford']

## 刪除指定的元素 remove()

cars = ['Honda', 'Toyota', 'Ford', 'BMW', 'Ford']

# 刪除第一個 'Ford'

cars.remove('Ford')

print(cars) # 輸出: ['Honda', 'Toyota', 'BMW', 'Ford']

## 刪除指定索引的元素 del

el 語法可以刪除串列中指定索引的元素，或者刪除整個串列。

* del x[index]：刪除指定索引的元素
* del x[start:end]：刪除指定範圍內的元素
* del x：刪除整個串列

cars = ['Honda', 'Toyota', 'Ford', 'Nissan']

# 刪除索引為 1 的元素

del cars[1]

print(cars) # 輸出: ['Honda', 'Ford', 'Nissan']

# 刪除索引 0 到 2 (不包含) 的元素

del cars[0:2]

print(cars) # 輸出: ['Nissan']

# 串列進階操作

## 尋找元素索引 index()

*x***.***index***(**value**,** start**=**0**,** end**=len(list))**

fruits = ['apple', 'banana', 'cherry', 'date', 'apple']

# 尋找 'cherry' 的索引

index = fruits.index('cherry')

print(f"'cherry' 的索引是: {index}") # 輸出: 'cherry' 的索引是: 2

# 尋找 'apple' 的索引 (會回傳第一次出現的索引)

index = fruits.index('apple')

print(f"'apple' 的索引是: {index}") # 輸出: 'apple' 的索引是: 0

## 計算元素數量 count()

fruits = ['apple', 'banana', 'cherry', 'apple', 'cherry']

# 計算 'apple' 出現的次數

count\_apple = fruits.count('apple')

print(f"'apple' 出現的次數: {count\_apple}") # 輸出: 'apple' 出現的次數: 2

# 計算 'how' 在字串串列中的次數

text = "hello, how are you? How can I help you?"

text\_list = text.split() # 將字串分割成串列

count\_how = text\_list.count('how')

print(f"'how' 出現的次數: {count\_how}") # 輸出: 'how' 出現的次數: 1

## 串列反轉 reverse()

numbers = [1, 2, 3, 4, 5]

numbers.reverse()

print(numbers) # 輸出: [5, 4, 3, 2, 1]

## 串列排序 sort()

numbers = [3, 1, 4, 2, 5]

numbers.sort()

print(f"升序排序: {numbers}") # 輸出: 升序排序: [1, 2, 3, 4, 5]

words = ['apple', 'banana', 'cherry', 'strawberry']

words.sort(reverse=True)

print(f"降序排序: {words}") # 輸出: 降序排序: ['strawberry', 'cherry', 'banana', 'apple']

## sorted() 函式

sorted() 函式會回傳一個新的排序後串列，而不會修改原串列。

numbers = [3, 1, 4, 2, 5]

sorted\_numbers = sorted(numbers)

print(f"原串列: {numbers}") # 輸出: 原串列: [3, 1, 4, 2, 5]

print(f"新排序後串列: {sorted\_numbers}") # 輸出: 新排序後串列: [1, 2, 3, 4, 5]

# 其他串列操作

## 更改串列元素的內容

cars = ['Honda', 'Toyota', 'Ford']

cars[1] = 'Nissan' # 將索引 1 的元素改成 'Nissan'

print(cars) # 輸出: ['Honda', 'Nissan', 'Ford']

## 串列的相加

cars1 = ['Honda', 'Toyota']

cars2 = ['Ford', 'Nissan']

cars\_all = cars1 + cars2

print(cars\_all) # 輸出: ['Honda', 'Toyota', 'Ford', 'Nissan']

## 串列乘以一個數字

num = [1, 2]

num\_repeated = num \* 3

print(num\_repeated) # 輸出: [1, 2, 1, 2, 1, 2]

## 串列為空串列的判斷

my\_list = []

if not my\_list:

    print("這個串列是空的")

else:

    print("這個串列不為空")