Python程式設計：資料型態(Data type)

**目錄**

[Python程式設計：資料型態(Data type) 1](#_Toc205987072)

[1. Print格式化輸入輸出 1](#_Toc205987073)

[1.1 使用 format() 函數 1](#_Toc205987074)

[1.2 使用 f-string (格式化字串) 1](#_Toc205987075)

[2. input()輸入資料 2](#_Toc205987076)

[3. 處理字串的數學運算：eval() 函數 2](#_Toc205987077)

[4. 輸出資料到檔案 2](#_Toc205987078)

[4.1 輸出資料到檔案：open() 函數 2](#_Toc205987079)

[5. 練習題目 3](#_Toc205987080)

[5.1 Problem: Hello, World! with a Name 3](#_Toc205987081)

[5.2 Problem: Simple Addition 3](#_Toc205987082)

[5.3 Problem: Expression Evaluator 4](#_Toc205987083)

[5.4 Problem: Average of Three Numbers 4](#_Toc205987084)

[5.5 Problem: Type Conversion - Integer 4](#_Toc205987085)

[5.6 Problem: Type Conversion - Float 4](#_Toc205987086)

[5.7 Problem: String Multiplication 5](#_Toc205987087)

[5.8 Problem: Write to File 5](#_Toc205987088)

[5.9 Problem: Write and Overwrite 5](#_Toc205987089)

[5.10 Problem: Write and Append 6](#_Toc205987090)

# Print格式化輸入輸出

## 使用 format() 函數

format() 函數是 Python 2.6 之後引入的一種格式化方法，它使用 {} 作為佔位符，並在 format() 函數的參數中提供要填入的內容。

**print(**"格式化字串 {}"**.format(**變數**))**

s = "Hello"

x = 99

print("字串 = {}".format(s))

print("數字 = {}".format(x))

print("字串 = {}，數字 = {}".format(s, x))

## 使用 f-string (格式化字串)

f-string 是 Python 3.6 之後引入的一種簡潔高效的格式化方法。它在字串前面加上 f，並在 {} 佔位符中直接放入變數或運算式，是目前最推薦的格式化方式。

*變數* **=** "值"

**print(**f"格式化字串 {變數}"**)**

name = "小明"

message = "Hello"

print(f"姓名：{name}，訊息：{message}")

# input()輸入資料

input() 函數用於從鍵盤讀取使用者輸入的資料。它的基本用法如下：

*value* **=** **input(**"prompt:"**)**

# prompt 是提示訊息，會顯示在螢幕上，告訴使用者需要輸入什麼。

# value 是一個變數，用於儲存使用者輸入的內容。

# input() 函數讀取到的所有資料，無論看起來像數字還是文字，都會被當作字串（string）來處理。如果你需要對數字進行數學運算，必須使用 int() 或 float() 等函數將其轉換為整數或浮點數。

print("歡迎使用成績輸入系統")

name = input("請輸入姓名：")

engh = input("請輸入英文成績：")

math = input("請輸入數學成績：")

total = int(engh) + int(math) # 將字串轉換為整數後相加

# 處理字串的數學運算：eval() 函數

eval() 函數可以將一個字串當作 Python 的運算式來執行，並返回運算結果。這在處理使用者輸入的數學公式時非常有用。

# numberStr = input("請輸入數值公式：")

numberStr = "5 \* 9 + 10"

result = eval(numberStr)

print(f"計算結果：{result}")

n1, n2, n3 = eval(input("請輸入3個數字，用逗號隔開："))

average = (n1 + n2 + n3) / 3

print(f"3個數字平均是 {average:.2f}")

# 我們輸入 2, 4, 6，eval() 會將這個字串轉換成一個元組 (2, 4, 6)，然後分別賦值給 n1、n2、n3。

# 輸出資料到檔案

## 輸出資料到檔案：open() 函數

除了在螢幕上顯示結果，我們也可以將資料輸出到檔案中。這需要用到 open() 函數來開啟檔案。

**with** **open(**"out.txt"**,** mode**=**"w"**,** encoding**=**"utf-8"**)** **as** f**:**

**print(**text01**,** file**=**f**)**

開啟檔案後，我們可以用 print() 函數將資料寫入檔案，並指定 file=file\_Obj。完成後，必須用 file\_Obj.close() 關閉檔案。

# 練習題目

## Problem: Hello, World! with a Name

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個字串，代表一個人的名字。然後，在螢幕上印出 "Hello, " 加上這個名字，最後以驚嘆號結尾。 | |
| Input:  輸入只有一行，包含一個字串，代表一個人的名字。 | Output:  輸出只有一行，格式為 "Hello, [名字]!"。 |
| Sample Input:  Alice | Sample Output:  Hello, Alice! |
| Answer: | |

## Problem: Simple Addition

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取兩個整數，計算它們的總和，並印出結果。 | |
| Input:  輸入有兩行，每行包含一個整數。 | Output:  輸出只有一行，為兩個整數的總和。 |
| Sample Input:  10  25 | Sample Output:  35 |
| Answer: | |

## Problem: Expression Evaluator

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個包含數學運算式的字串，並使用 eval() 函數計算該運算式的值，然後印出結果。 | |
| Input:  輸入只有一行，為一個數學運算式字串。 | Output:  輸出只有一行，為運算式計算後的結果。 |
| Sample Input:  (5 + 3) \* 2 | Sample Output:  16 |
| Answer: | |

## Problem: Average of Three Numbers

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一行，該行包含三個以逗號分隔的數字。計算這三個數字的平均值，並將結果輸出到小數點後兩位。(格式部份我沒有交到，你可以用GPT/google找一下如何控制輸出格式) | |
| Input:  輸入只有一行，包含三個以逗號分隔的數字。 | Output:  輸出只有一行，為三個數字的平均值，格式為 X.XX |
| Sample Input:  2, 4, 6 | Sample Output:  4.00 |
| Answer: | |

## Problem: Type Conversion - Integer

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個字串，該字串代表一個整數。將這個字串轉換為整數後，印出這個整數的值以及其資料型態。 | |
| Input:  輸入只有一行，為一個整數字串。 | Output:  輸出有兩行。第一行為轉換後的整數，第二行為其資料型態。 |
| Sample Input:  123 | Sample Output:  123  <class 'int'> |
| Answer: | |

## Problem: Type Conversion - Float

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個字串，該字串代表一個浮點數。將這個字串轉換為浮點數後，印出這個浮點數的值以及其資料型態。 | |
| Input:  輸入只有一行，為一個浮點數字串。 | Output:  輸出有兩行。第一行為轉換後的浮點數，第二行為其資料型態。 |
| Sample Input:  3.14159 | Sample Output:  3.14159  <class 'float'> |
| Answer: | |

## Problem: String Multiplication

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個字串和一個整數。將這個字串重複指定的次數，並將結果印出。 | |
| Input:  輸入有兩行。第一行為一個字串，第二行為一個整數。 | Output:  輸出只有一行，為重複後的字串。 |
| Sample Input:  Python  3 | Sample Output:  PythonPythonPython |
| Answer: | |

## Problem: Write to File

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一行文字，然後將這行文字寫入一個名為 output\_w.txt 的檔案中。如果檔案已存在，則將其內容覆蓋。 | |
| Input:  輸入只有一行，為一個字串。 | Output:  無輸出到標準輸出，但會在程式執行後產生一個 output\_w.txt 的檔案，其內容為輸入的字串。 |
| Sample Input:  This is a test. | Sample Output: |
| Answer: | |

## Problem: Write and Overwrite

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取兩行文字。將第一行文字寫入一個名為  file\_write.txt 的檔案中，使用寫入模式 ("w") 。然後，將第二行文字也寫入同一個檔案，同樣使用寫入模式。程式結束後，檢查 file\_write.txt 的內容。 | |
| Input:  輸入有兩行，每行包含一個字串。 | Output:  無標準輸出。程式執行後，file\_write.txt 檔案的內容應該只有第二行輸入的文字。 |
| Sample Input:  First line to be overwritten.  This is the final line. | Sample Output: |
| Answer: | |

## Problem: Write and Append

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取兩行文字。將第一行文字寫入一個名為 file\_append.txt 的檔案中，使用寫入模式 ("w") 。然後，將第二行文字寫入同一個檔案，但這次使用附加模式 ("a") 。程式結束後，檢查 file\_append.txt 的內容。 | |
| Input:  輸入有兩行，每行包含一個字串。 | Output:  無標準輸出。程式執行後，file\_append.txt 檔案的內容應該為兩行文字。 |
| Sample Input:  This is the first line.  This is the second line. | Sample Output: |
| Answer: | |