Python程式設計：判斷式

**目錄**

[Python程式設計：判斷式 1](#_Toc205990204)

[1. 關聯運算子 (Relational Operators) 1](#_Toc205990205)

[2. 邏輯運算子 (Logical Operators) 1](#_Toc205990206)

[3. if 敘述 2](#_Toc205990207)

[3.1 範例：年齡判斷 2](#_Toc205990208)

[3.2 範例：這個程式根據學生的成績給予對應的等級 (A, B, C, D, F) 3](#_Toc205990209)

[3.3 範例：判斷使用者輸入的字元是英文字母（大寫或小寫）還是數字或特殊字元 3](#_Toc205990210)

[4. 尚未設定的變數值 None 3](#_Toc205990211)

[5. 練習 4](#_Toc205990212)

[5.1 Problem: 判斷奇偶數 4](#_Toc205990213)

[5.2 Problem: 溫度警示系統 4](#_Toc205990214)

[5.3 Problem: 簡易密碼強度檢查 4](#_Toc205990215)

[5.4 Problem: 庫存檢查 4](#_Toc205990216)

[5.5 Problem: BMI 健康判斷 5](#_Toc205990217)

[5.6 Problem: 判斷閏年 5](#_Toc205990218)

[5.7 Problem: 火箭升空判斷 6](#_Toc205990219)

[5.8 Problem: 判斷字母類型 6](#_Toc205990220)

[5.9 Problem: 登入驗證 6](#_Toc205990221)

# 關聯運算子 (Relational Operators)

以下是 Python 中使用的關聯運算子 ：

|  |  |  |
| --- | --- | --- |
| 關聯運算子 | 說明 | 範例 |
| > | 大於 | a > b 表示 a 是否大於 b 。 |
| >= | 大於等於 | a >= b 表示 a 是否大於等於 b 。 |
| < | 小於 | a < b 表示 a 是否小於 b 。 |
| <= | 小於等於 | a <= b 表示 a 是否小於等於 b 。 |
| == | 等於 | a == b 表示 a 是否等於 b 。 |
| != | 不等於 | a != b 表示 a 是否不等於 b 。 |

# 邏輯運算子 (Logical Operators)

|  |  |  |
| --- | --- | --- |
| 運算子 | 說明 | 範例 |
| **and** | 當所有條件都為 True 時，回傳 True。  如果左邊運算式為 False，則直接回傳 False。  如果左邊運算式為 True，則回傳右邊運算式的值。 | True and True → True  False and True → False  5 and 10 → 10  0 and 5 → 0 |
| **or** | 當至少一個條件為 True 時，回傳 True。  如果左邊運算式為 True，則直接回傳 True。  如果左邊運算式為 False，則回傳右邊運算式的值。 | True or False → True  False or False → False  True or 0 → True  0 or 5 → 5 |
| **not** | 回傳相反的布林值。 | not True → False  not False → True  not 5 → False  not 0 → True |

print(10<8) # False

print(not(10<8)) # True

print(True and False) # False

print(True or False) # True

print(not(True and False)) # True

print(not(True or False)) # False

這邊提供一些YT小影片，一樣推薦see see

[Logical operators in Python are easy 🔣 - YouTube](https://www.youtube.com/watch?v=W7luvtXeQTA)

[Python Logical Operators with Examples - APPFICIAL](https://www.youtube.com/watch?v=JGNMAVqybLA)

# if 敘述

if...else 敘述提供了兩種不同的執行路徑，根據條件的真偽來選擇其中之一執行。

當條件判斷為

* True 時，會執行 # 程式碼區塊 。如果條件為
* False，則會跳過該區塊，不執行任何程式碼 。

**if** **(**條件判斷1**):**

# 程式碼區塊1

**elif** **(**條件判斷2**):**

# 程式碼區塊2

**else:**

# 程式碼區塊n

## 範例：年齡判斷

# age = eval(input("請輸入年齡："))

age = 18

if (int(age) >= 20):

    print("您已成年")

else:

    print("您是未成年")

## 範例：這個程式根據學生的成績給予對應的等級 (A, B, C, D, F)

score = input("請輸入分數：")

sc = int(score)

if (sc >= 90):

    print("A")

elif (sc >= 80):

    print("B")

elif (sc >= 70):

    print("C")

elif (sc >= 60):

    print("D")

else:

    print("F")

## 範例：判斷使用者輸入的字元是英文字母（大寫或小寫）還是數字或特殊字元

ch = input("請輸入字元：")

if (ord(ch) >= ord("A") and ord(ch) <= ord("Z")):

    print("這是大寫字元")

elif (ord(ch) >= ord("a") and ord(ch) <= ord("z")):

    print("這是小寫字元")

elif (ord(ch) >= ord("0") and ord(ch) <= ord("9")):

    print("這是數字")

else:

    print("這是特殊字元")

# 尚未設定的變數值 None

在程式設計中，有時會先定義一個變數，但暫時不給它一個具體的數值。這時可以將變數的值設定為 None 。

x = None

print(x) # None

print(type(x)) # <class 'NoneType'>

# 練習

## Problem: 判斷奇偶數

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個整數。判斷這個數字是奇數還是偶數，並印出對應的結果。 | |
| Input:  輸入只有一行，包含一個整數。 | Output:  輸出只有一行。如果數字是偶數，印出 "Even"；如果是奇數，印出 "Odd"。 |
| Sample Input:  15 | Sample Output:  Odd |
| Answer: | |

## Problem: 溫度警示系統

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，模擬一個溫度警示系統。程式會從標準輸入讀取一個溫度值。如果溫度超過 85 度，則發出警示，印出 "溫度過高，請小心！" 。 | |
| Input:  輸入只有一行，包含一個浮點數，代表溫度。 | Output:  如果溫度超過 85，輸出 "溫度過高，請小心！" ；否則不輸出任何內容。 |
| Sample Input:  90.5 | Sample Output:  溫度過高，請小心！ |
| Answer: | |

## Problem: 簡易密碼強度檢查

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個字串作為密碼。程式會檢查密碼長度是否至少為 8 個字元，並且是否只由英文字母和數字組成 。如果符合這些條件，印出 "密碼強度良好" ；否則印出 "密碼太弱" 。 | |
| Input:  輸入只有一行，包含一個字串。 | Output:  輸出只有一行，為 "密碼強度良好" 或 "密碼太弱" 。 |
| Sample Input:  Password123 | Sample Output:  密碼強度良好 |
| Answer: | |

## Problem: 庫存檢查

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個整數，代表某項商品的庫存數量 。如果庫存低於 10，則發出補貨提醒，印出 "庫存不足，請重新訂購。" 。 | |
| Input:  輸入只有一行，包含一個整數，代表庫存。 | Output:  如果庫存低於 10，輸出 "庫存不足，請重新訂購。" ；否則不輸出任何內容。 |
| Sample Input:  5 | Sample Output:  庫存不足，請重新訂購。 |
| Answer: | |

## Problem: BMI 健康判斷

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取使用者的身高（公分）和體重（公斤）。計算 BMI (公式：體重 / (身高/100)^2)，並根據 BMI 判斷健康狀況 。   * 如果 BMI 介於 18.5 至 24 之間 (含)，印出 "BMI健康正常" 。 * 否則，印出 "BMI健康不正常" 。 | |
| Input:  輸入有兩行，第一行為身高（公分），第二行為體重（公斤）。 | Output:  輸出只有一行，為 BMI 判斷結果。 |
| Sample Input:  780  60 | Sample Output:  BMI健康正常 |
| Answer: | |

## Problem: 判斷閏年

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個年份，判斷該年份是否為閏年 。閏年的判斷規則如下：   * 能被 400 整除的年份是閏年。 * 能被 4 整除但不能被 100 整除的年份是閏年。 * 其他情況都不是閏年。 | |
| Input:  輸入只有一行，包含一個整數，代表年份。 | Output:  輸出只有一行。如果是閏年，印出 "閏年"；否則印出 "不是閏年" 。 |
| Sample Input:  2000 | Sample Output:  閏年 |
| Answer: | |

## Problem: 火箭升空判斷

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取火箭升空速度（km/s）。根據速度判斷火箭所處的階段 。   * 第一宇宙速度：7.9 km/s 。如果速度大於等於 7.9 但小於 11.2，印出 "人造衛星可以環繞地球運行" * 第二宇宙速度：11.2 km/s 。如果速度大於等於 11.2 但小於 16.7，印出 "人造衛星可以擺脫地球引力，環繞太陽系" 。 * 第三宇宙速度：16.7 km/s 。如果速度大於等於 16.7，印出 "人造衛星可以脫離太陽引力到太陽系外太空" 。 * 如果速度小於 7.9，印出 "人造衛星無法脫離地球引力" 。 | |
| Input:  輸入只有一行，包含一個浮點數，代表火箭速度（km/s）。 | Output:  輸出只有一行，為火箭所處的階段判斷。 |
| Sample Input:  7.5 | Sample Output:  人造衛星無法脫離地球引力 |
| Answer: | |

## Problem: 判斷字母類型

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，從標準輸入讀取一個字元。判斷這個字元是大寫英文字母、小寫英文字母、數字，還是特殊字元。 | |
| Input:  輸入只有一行，包含一個字元。 | Output:  輸出只有一行，為判斷結果 ("大寫字元", "小寫字元", "數字", "特殊字元")。 |
| Sample Input:  $ | Sample Output:  特殊字元 |
| Answer: | |

## Problem: 登入驗證

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式，模擬一個簡單的登入驗證 。程式會從標準輸入讀取使用者輸入的帳號和密碼。   * 如果帳號是 "user123" 且密碼是 "password"，則驗證成功，印出 "登入成功！" 。 * 否則，驗證失敗，印出 "用戶名或密碼錯誤，請重試。" 。 | |
| Input:  輸入有兩行，第一行為帳號，第二行為密碼。 | Output:  輸出只有一行，為驗證結果。 |
| Sample Input:  user123 pass | Sample Output:  用戶名或密碼錯誤，請重試。 |
| Answer: | |