Python程式設計：迴圈

**目錄**

[Python程式設計：迴圈 1](#_Toc206344741)

[1. 基本 for 迴圈 1](#_Toc206344742)

[1.1 range() 函式 2](#_Toc206344743)

[1.2 雙重for迴圈 2](#_Toc206344744)

[1.3 範例：九九乘法表 2](#_Toc206344745)

[1.4 強制離開 for 迴圈 - break 指令 3](#_Toc206344746)

[1.5 範例：尋找第一個超過 90 分的成績 3](#_Toc206344747)

[1.6 for 迴圈暫停但不終止 - continue 指令 3](#_Toc206344748)

[1.7 範例：只列印偶數 3](#_Toc206344749)

[2. while 迴圈 3](#_Toc206344750)

[2.1 範例：使用者輸入 q 時終止 4](#_Toc206344751)

[2.2 範例：計算使用者輸入數字的總和，輸入 0 終止 4](#_Toc206344752)

[2.3 設立明確終止值 (Sentinel value) 4](#_Toc206344753)

[2.4 範例：計算使用者輸入數字的總和，輸入 0 終止 4](#_Toc206344754)

[2.5 範例：使用 while 迴圈印出 9x9 乘法表 4](#_Toc206344755)

[2.6 強制離開 while 迴圈 - break 指令 5](#_Toc206344756)

[2.7 暫停但不終止 while 迴圈 - continue 指令 5](#_Toc206344757)

[3. 練習 5](#_Toc206344758)

[3.1 Problem: range() 函式 - 倒數計時 5](#_Toc206344759)

[3.2 Problem: Fibonacci 數列生成 6](#_Toc206344760)

[3.3 Problem: 數字累加與乘積 6](#_Toc206344761)

[3.4 Problem: 巢狀迴圈 - 矩形印製 6](#_Toc206344762)

[3.5 Problem: 迴圈與 break - 尋找第一個偶數 7](#_Toc206344763)

[3.6 Problem: while 迴圈 - 猜數字 7](#_Toc206344764)

[3.7 Problem: while 迴圈與 break - 密碼驗證 8](#_Toc206344765)

迴圈是一種程式控制結構，可以讓程式碼重複執行某個動作，直到滿足特定的條件為止。這對於處理大量資料、遍歷集合或重複執行任務非常有效。

# 基本 for 迴圈

for 迴圈用於遍歷任何可迭代物件（iterable）中的元素。在每次迴圈迭代中，它會將物件中的一個元素指派給一個變數，然後執行迴圈內的程式碼。

**for** 變數 **in** 可迭代物件**:**

# 執行程式碼

players = ['Curry', 'Jordan', 'James', 'Durant', 'Obama']

for player in players:

    print(player)

## range() 函式

range() 函式是一種內建的可迭代物件，主要用於產生一個整數序列，通常用於 for 迴圈中以控制迴圈執行的次數。

**range(**start**,** stop**,** step**)**

# 產生從 start 開始，

# 到 stop-1 結束，

# 每次以 step 為步長的整數序列。step 可以是正數（遞增）或負數（遞減）。

for i in range(5):

    print(i)

for i in range(1, 5):

    print(i)

for i in range(1, 10, 2):

    print(i)

for i in range(10, 0, -2):

    print(i)

## 雙重for迴圈

colors = ["Red", "Green", "Blue"]

shapes = ["Circle", "Square", "Line"]

for color in colors:

    for shape in shapes:

        print(f"顏色: {color}, 形狀: {shape}")

## 範例：九九乘法表

for i in range(1, 10):

    for j in range(1, 10):

        result = i \* j

        print(f"{i}\*{j}={result}", end='\t') # 使用 `end='\t'` 讓輸出在同一行，並以 tab 分隔

    print() # 內層迴圈結束後，換行

## 強制離開 for 迴圈 - break 指令

break 指令用於強制終止並跳出當前迴圈。當在迴圈內執行到 break 時，迴圈會立即停止，並跳到迴圈後面的程式碼繼續執行。這對於在迴圈中找到所需條件後，提前結束迴圈非常有用。

**for** 變數 **in** 可迭代物件**:**

# 程式碼區塊 1

**if** 條件運算式**:**

**break**

# 程式碼區塊 2 (如果滿足條件，此處程式碼將不會被執行)

## 範例：尋找第一個超過 90 分的成績

scores = [85, 92, 78, 95, 88]

for score in scores:

    if score > 90:

        print(f"找到第一個超過 90 分的成績：{score}")

        break # 找到後立即終止迴圈

## for 迴圈暫停但不終止 - continue 指令

continue 指令用於跳過當前迴圈的其餘程式碼，並進入下一次迴圈迭代。它不會終止整個迴圈，只會跳過當前這次的執行。

**for** 變數 **in** 可迭代物件**:**

# 程式碼區塊 1

**if** 條件運算式**:**

**continue** # 如果滿足條件，跳到下一次迴圈

# 程式碼區塊 2 (如果滿足條件，此處程式碼將不會被執行)

## 範例：只列印偶數

for i in range(10):

    if i % 2 != 0:

        continue # 如果是奇數，跳過本次迴圈

    print(i)

# while 迴圈

while 迴圈是一種條件式迴圈，它會一直執行，直到指定的條件運算式變為 False 為止。設計 while 迴圈時，必須確保在迴圈內部有可以改變條件的程式碼，否則會陷入無限迴圈的陷阱。

**while** 條件運算式**:**

# 程式碼區塊

## 範例：使用者輸入 q 時終止

msg = ''

while msg != 'q':

    msg = input("請輸入訊息（輸入 'q' 結束）：")

    print(msg)

## 範例：計算使用者輸入數字的總和，輸入 0 終止

total = 0

num = int(input("請輸入一個數字（輸入 0 結束）："))

while num != 0:

    total += num

    num = int(input("請輸入一個數字（輸入 0 結束）："))

print(f"總和為: {total}")

## 設立明確終止值 (Sentinel value)

在設計 while 迴圈時，可以設定一個特殊的數值或字串作為終止值（sentinel value），當使用者輸入這個值時，迴圈就會結束。

## 範例：計算使用者輸入數字的總和，輸入 0 終止

total = 0

num = int(input("請輸入一個數字（輸入 0 結束）："))

while num != 0:

    total += num

    num = int(input("請輸入一個數字（輸入 0 結束）："))

print(f"總和為: {total}")

## 範例：使用 while 迴圈印出 9x9 乘法表

i = 1

while i <= 9:

    j = 1

    while j <= 9:

        result = i \* j

        print(f"{i}\*{j}={result}", end='\t')

        j += 1

    print()

    i += 1

## 強制離開 while 迴圈 - break 指令

break 指令在 while 迴圈中的作用與在 for 迴圈中相同：強制終止並跳出當前迴圈。這對於在迴圈執行中途滿足某個條件時，需要立即停止非常有用。

while True: # 建立一個無限迴圈

    command = input("請輸入指令（輸入 'exit' 結束）：")

    if command == 'exit':

        break

    print(f"你輸入了: {command}")

## 暫停但不終止 while 迴圈 - continue 指令

break 指令在 while 迴圈中的作用與在 for 迴圈中相同：強制終止並跳出當前迴圈。這對於在迴圈執行中途滿足某個條件時，需要立即停止非常有用。

num = 0

while num < 10:

    num += 1

    if num % 2 != 0:

        continue # 如果是奇數，跳過本次迴圈

    print(num)

# 練習

## Problem: range() 函式 - 倒數計時

|  |  |
| --- | --- |
| Problem Description:  請使用 range() 函式和 for 迴圈，從一個指定的數字開始，以指定的步長倒數計時，直到 0 為止。 | |
| Input:  兩行輸入，第一行為起始數字 start，第二行為步長 step（正數）。 | Output:  倒數計時的每個數字。 |
| Sample Input:  10  2 | Sample Output:  10  8  6  4  2 |
| Sample Input:  15  3 | Sample Output:  15  12  9  6  3 |
| Answer: | |

## Problem: Fibonacci 數列生成

|  |  |
| --- | --- |
| Problem Description:  Fibonacci 數列是一個非常經典的數列，其特點是每個數都是前兩個數的總和。數列從 0 和 1 開始。請你寫一個程式，生成指定長度的 Fibonacci 數列。（Fibonacci 數列可以自己搜尋認識一下~） | |
| Input:  一個整數 n，表示要生成的 Fibonacci 數列的長度。 | Output:  一個串列，包含從第 0 項到第 n-1 項的 Fibonacci 數。 |
| Sample Input:  8 | Sample Output:  [0, 1, 1, 2, 3, 5, 8, 13] |
| Sample Input:  5 | Sample Output:  [0, 1, 1, 2, 3] |
| Answer: | |

## Problem: 數字累加與乘積

|  |  |
| --- | --- |
| Problem Description:  請你寫一個程式，計算從 1 到指定整數 n 的所有數字的總和，以及所有奇數的乘積。 | |
| Input:  一個正整數 n。 | Output:  輸出兩行：  第一行：總和。  第二行：奇數的乘積。 |
| Sample Input:  5 | Sample Output:  15  15 |
| Sample Input:  6 | Sample Output:  21  15 |
| Answer: | |

## Problem: 巢狀迴圈 - 矩形印製

|  |  |
| --- | --- |
| Problem Description:  請使用巢狀迴圈印製一個指定寬度和高度的實心矩形。 | |
| Input:  兩行輸入，第一行為寬度 w，第二行為高度 h。 | Output:  一個 w 寬度、h 高度的星號（\*）矩形 |
| Sample Input:  5  3 | Sample Output:  \*\*\*\*\*  \*\*\*\*\*  \*\*\*\*\* |
| Sample Input:  4  4 | Sample Output:  \*\*\*\*  \*\*\*\*  \*\*\*\*  \*\*\*\* |
| Answer: | |

## Problem: 迴圈與 break - 尋找第一個偶數

|  |  |
| --- | --- |
| Problem Description:  給定一個包含多個整數的串列，請使用 for 迴圈和 break 指令，找出並列印串列中的第一個偶數，如果沒有偶數，則不輸出任何內容。 | |
| Input:  一個包含整數的串列。 | Output:  第一個偶數。 |
| Sample Input:  [1, 3, 5, 8, 9, 11] | Sample Output:  8 |
| Sample Input:  [1, 3, 5, 7, 9, 11] | Sample Output:  (無輸出) |
| Answer: | |

## Problem: while 迴圈 - 猜數字

|  |  |
| --- | --- |
| Problem Description:  寫一個簡單的猜數字遊戲。程式會設定一個秘密數字（例如 50），並要求使用者不斷輸入數字。如果使用者猜對了，程式就結束。如果沒猜對，就給予提示（太高或太低）。 | |
| Input:  使用者不斷輸入的數字。 | Output:  每次輸入後的提示，直到猜對為止。 |
| Sample Input:  75 25 50 | Sample Output:  太高了！  太低了！  恭喜你猜對了！ |
| Sample Input:  10  90  50 | Sample Output:  太低了！  太高了！  恭喜你猜對了！ |
| Answer: | |

## Problem: while 迴圈與 break - 密碼驗證

|  |  |
| --- | --- |
| Problem Description:  請寫一個程式來驗證使用者輸入的密碼。如果使用者輸入的密碼是 password123，就列印「密碼正確」，然後結束程式。如果密碼不正確，就提示「密碼錯誤，請重試」。 | |
| Input:  使用者不斷輸入的密碼。 | Output:  密碼是否正確的提示。 |
| Sample Input:  12345  test  password123 | Sample Output:  密碼錯誤，請重試  密碼錯誤，請重試  密碼正確 |
| Sample Input:  admin  password123 | Sample Output:  密碼錯誤，請重試  密碼正確 |
| Answer: | |