Python 程式設計　講義

迴圈、range() 與流程控制

滄海《Python 程式設計初學指引》二版 · CH04

§ 1　名詞速查表

中文	English	一句話定義	例	易混
迴圈	loop	在條件成立下重複執行一段程式碼	—	for vs while
for...in 迴圈	`for...in`	依序走訪物件中每個元素	`for c in 'Py':`	次數明確時用
可迭代物件	iterable	能逐一取出元素的物件：字串/串列/元組/字典/集合	`'abc'` `[1,2]`	—
while 迴圈	`while`	條件為真就反覆執行，為假才停	`while n<5:`	次數不明確時用
`range()`	—	產生規則數字序列，控制迴圈次數	`range(5)`	含頭不含尾
梯級	step	`range`／切片每次跳幾步，可負（倒序）	`range(5,1,-2)`	預設 1
切片	slice	`物件[起始:結束:梯級]` 取某區間元素	`s[1:4]`	含頭不含尾
巢狀迴圈	nested loop	迴圈裡再放迴圈（外圈 × 內圈）	九九乘法表	內圈跑完外圈才進一格
`break`	—	立即跳出並結束所在的迴圈	—	只跳一層
`continue`	—	跳過本回合剩下的程式碼，回去判斷下一回合	—	不結束迴圈
無窮迴圈	infinite loop	條件永遠為真、停不下來的 while	`while True:`	忘了改變條件

§ 2 核心概念

核心概念

迴圈 (loop) 讓程式在條件成立時**重複執行**一段程式碼，省去複製貼上、降低維護成本。Python 有兩種：`for...in`（走訪可迭代物件的每個元素）與 `while`（條件為真就反覆執行）。判準：**執行次數明確**（跑幾次或走訪固定資料）用 `for...in`；**次數由執行結果決定**（例如猜數字猜到對為止）用 `while`。

兩者都靠**縮排**界定迴圈主體，標題行尾要加**冒號** `:`。`for...in` 常搭 `range()` 產生規則數字序列來控制次數，三種寫法：`range(結束)`、`range(起始, 結束)`、`range(起始, 結束, 梯級)`，一律**含頭不含尾**。序列物件可用**切片** `物件[起始:結束:梯級]` 只走某區間。

兩個流程控制指令：`break` 立即結束所在迴圈、`continue` 跳過本回合剩餘程式碼回去判斷下一回合；兩者在巢狀迴圈中都只作用於**自己所在的那一層**。`while` 必須有能讓條件變為假的程式碼，否則成**無窮迴圈**。

§ 3 主要內容

3.1 一、兩種迴圈的執行流程（招牌）

`for...in` 從物件讀元素、讀到就執行主體、讀完就結束；`while` 先判條件、為真就執行主體再回頭判、為假就結束。兩者主體裡可放 `break`（直接跳出）與 `continue`（回去判斷下一回合）。

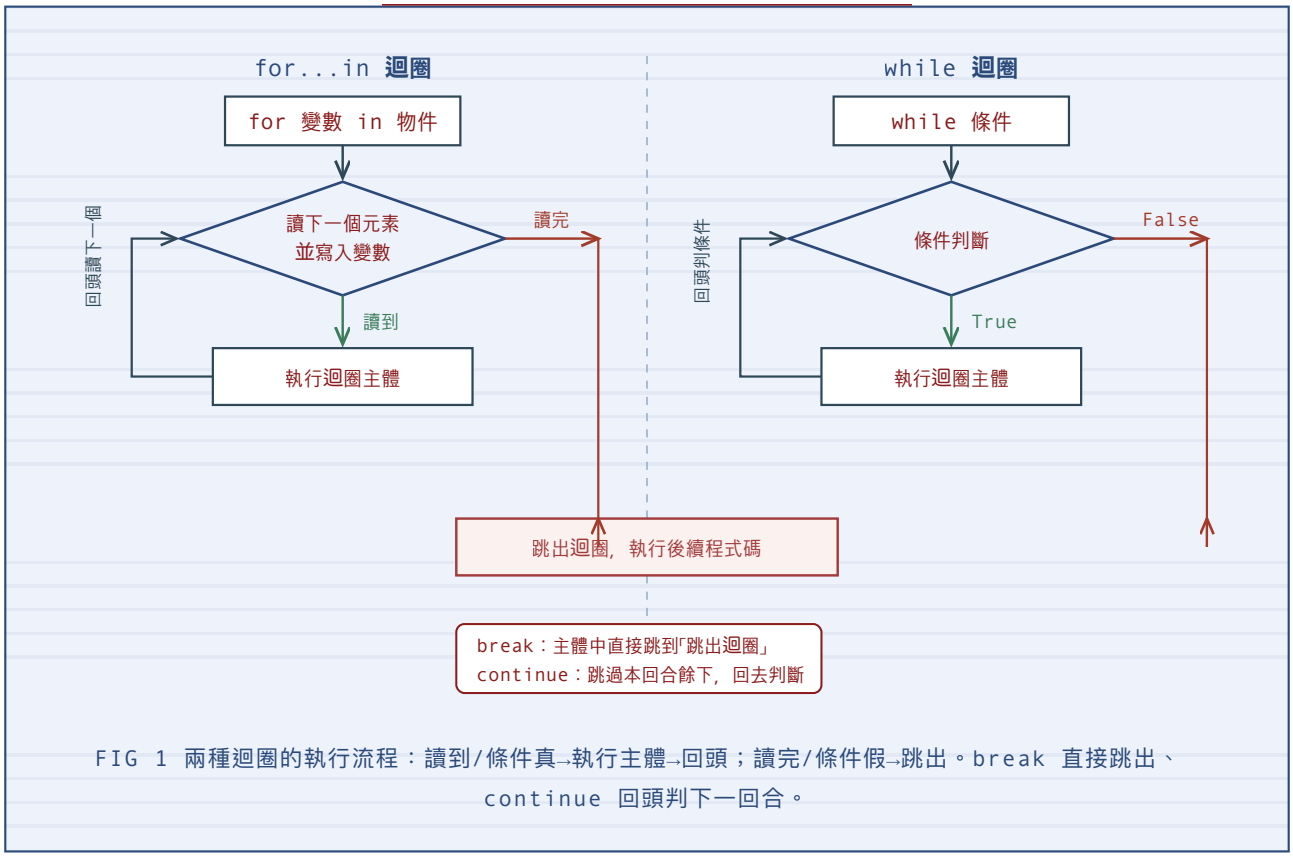

FIG 1 兩種迴圈的執行流程：讀到/條件真→執行主體→回頭；讀完/條件假→跳出。break 直接跳出、continue 回頭判下一回合。

3.2 二、for...in 走訪可迭代物件

```
for c in 'Python':       # 走訪字串：逐字元
    print(c, end='')     # Python (end='' 不換行)

for n in [10, 20, 30]:   # 走訪串列：逐元素
    print(n)             # 10 / 20 / 30 (各一行)
```

`in` 後可接字串 (str)、串列 (list)、元組 (tuple) 等序列型別，以及字典 (dict)、集合 (set) 等可迭代型別。每讀到一個元素就寫入前方變數、執行一次主體；讀完就結束。

3.3 三、range() 控制執行次數

```
range(5)          # 0 1 2 3 4          從 0 到 結束-1
range(2, 6)       # 2 3 4 5            從 起始 到 結束-1
range(1, 10, 2)   # 1 3 5 7 9          加上梯級
range(5, 1, -1)   # 5 4 3 2            梯級為負 → 倒序
range(3, 1)       #  (空)        起始 ≥ 結束又無負梯級 → 空序列
```

三件事要記牢：① **含頭不含尾**（`range(5)` 沒有 5）；② 正梯級時起始須**小於**結束，否則空序列；③ 倒序要把梯級設**負值**（`range(5, 1, -2)` 得 `5 3`）。

```
total = 0
for n in range(1, 101, 2):   # 1, 3, 5, ..., 99 (1~100 的奇數)
    total += n
print('1~100 奇數和 =', total)   # 2500
```

3.4 四、序列切片 物件[起始:結束:梯級]

```
s = 'Python'
s[1:4]      # 'yth'    起始含、結束不含 (序號 1,2,3)
s[:4]       # 'Pyth'   起始省略 → 從頭
s[-4:]      # 'thon'   負序號：-1 是最後一個
s[::2]      # 'Pto'    只給梯級 → 每隔一個
for c in s[-4::2]:   # 在 for 裡用切片，只走某區間
    print(c, end='')  # to
```

序號從 **0** 起算（第一個是 `s[0]`）；最後一個是 `s[-1]`、倒數第二是 `s[-2]`。切片同樣**含頭不含尾**。

3.5 五、while 迴圈與 in / not in 條件

```
# 用索引走訪字串 (等效於 for c in s)
s = 'Python'; i = 0
while i < len(s):
    print(s[i], end='')    # Python
    i += 1                 # 關鍵：改變條件，否則無窮迴圈

# in / not in 當 while 條件：元素還在物件中就繼續
word = 'Python'
while 'thon' in word:
    word = word.upper()    # 變大寫後 'thon' 不在 → 停
print(word)                # PYTHON
```

無窮迴圈

`while` 一定要放能讓**條件變為假**的程式碼（例如 `i += 1`、讀新輸入）。漏了就停不下來，成無窮迴圈。要刻意做無窮迴圈時用 `while True:`，並在裡面用 `break` 設出口。

3.6 六、巢狀迴圈（招牌應用：九九乘法表）

迴圈裡再放迴圈：外圈每跑一回合，內圈就**整個跑完一輪**。互動像時鐘的分針（外圈）與秒針（內圈）。

```
for i in range(1, 10):                  # 外圈：被乘數 1~9
    for j in range(1, 10):              # 內圈：乘數 1~9 (每個 i 都跑完一輪)
        print('%d*%d=%2d' % (i, j, i*j), end=' ')
    print()                             # 內圈跑完換行 (屬於外圈)
```

內圈的 `print(..., end=' ')` 同列不換行；外圈那行 `print()` 在內圈跑完後換行，於是印出 9×9 的表格。

3.7 七、break 與 continue

```
# break：乘數到 5 就跳出內圈（每列只印到 *4）
for i in range(1, 10):
    for j in range(1, 10):
        if j == 5: break            # 只跳出內圈，外圈繼續
        print('%d*%d=%2d' % (i, j, i*j), end=' ')
    print()

# continue：乘數是 5 就跳過不印（每列缺 *5 那項）
for i in range(1, 10):
    for j in range(1, 10):
        if j == 5: continue         # 跳過本回合，回去判下一個 j
        print('%d*%d=%2d' % (i, j, i*j), end=' ')
    print()
```

`break` 與 `continue` 在巢狀中只作用於**自己所在那一層**：放內圈只影響內圈，外圈照常。差別：`break` **結束**迴圈、`continue` 只跳過**這一回合**。

§ 4 語法與方法速查

```
# for...in：走訪可迭代物件
for 變數 in 物件:          # 字串/串列/元組/字典/集合；注意冒號與縮排
    主體

# range（含頭不含尾）
range(結束)                 # 0 .. 結束-1
range(起始, 結束)           # 起始 .. 結束-1
range(起始, 結束, 梯級)     # 梯級可負（倒序）；正梯級須 起始<結束 否則空

# 切片（含頭不含尾；序號 0 起、-1 為末）
物件[起始:結束:梯級]        # 各參數可省；物件[::2] 每隔一個；物件[::-1] 反轉

# while：條件為真就反覆（須有改變條件的程式碼）
while 條件:
    主體
while 元素 in 物件:         # in / not in 當條件
    主體

# 巢狀
for i in ...:
    for j in ...:           # 內圈縮排再一層；外圈每回合內圈跑完一輪
        主體

# 流程控制（只作用於所在那一層迴圈）
break          # 立即結束迴圈
continue       # 跳過本回合餘下，回去判斷下一回合
while True:    # 無窮迴圈，內部以 break 設出口
```

§ 5 常見錯誤

常見錯誤

- `range` **含尾**：`range(5)` 是 `0~4`、**沒有 5**；要含 5 得寫 `range(6)` 或 `range(1, 6)`。
- **倒序忘了負梯級**：`range(5, 1)` 是空序列；倒序要寫 `range(5, 1, -1)`。
- **while 無窮迴圈**：忘了在主體改變條件（如漏 `i += 1`），程式停不下來。
- **忘了冒號**：`for`／`while` 標題行尾要 `:`，漏掉 → `SyntaxError`。
- **縮排錯誤**：主體要縮排；該屬迴圈的程式碼沒縮排會被當成迴圈外（巢狀的換行 `print()` 縮排位置決定屬內圈或外圈）。
- **break／continue 想跳多層**：兩者只作用於**自己所在那一層**；要控制外圈得在外圈寫，或用旗標變數。
- **混淆 break 與 continue**：`break` 結束整個迴圈；`continue` 只跳過這一回合、迴圈還在跑。
- **切片越界不報錯**：`'Py'[0:99]` 不會錯、回 `'Py'`；但**索引** `'Py'[99]` 會 `IndexError`。

§ 6 練習題

練習 1（一般題）：for 走訪與 range

> **引導步驟**
>
> 1. 用 `for...in` 走訪字串 `'Cat'`，逐字元印出（同列、空格分隔）。
> 2. 用 `for n in range(1, 6)` 印 `1 2 3 4 5`。

解答

```
for c in 'Cat':
    print(c, end=' ')      # C a t
print()
for n in range(1, 6):
    print(n, end=' ')      # 1 2 3 4 5
```

練習 2（重要題）：range 三種寫法與倒序

> **引導步驟**
>
> 1. 各印出 `list(range(5))`、`list(range(2, 6))`、`list(range(1, 10, 2))`。
> 2. 印 `list(range(5, 0, -1))`，想想為什麼要負梯級。

解答

```
print(list(range(5)))          # [0, 1, 2, 3, 4]
print(list(range(2, 6)))       # [2, 3, 4, 5]
print(list(range(1, 10, 2)))   # [1, 3, 5, 7, 9]
print(list(range(5, 0, -1)))   # [5, 4, 3, 2, 1]
# 倒序起始>結束，梯級必須為負，否則得空序列
```

練習 3（重要題）：while 累加（1~100 偶數和）

> **引導步驟**
>
> 1. 用 `while` 從 2 開始、每次加 2，累加到 100。
> 2. 記得放改變條件的程式碼，避免無窮迴圈。印出總和。

解答

```
total = 0; n = 2
while n <= 100:
    total += n
    n += 2                # 改變條件，否則無窮迴圈
print('1~100 偶數和 =', total)   # 2550
```

練習 4（一般題）：巢狀迴圈印三角形

> **引導步驟**
>
> 1. 外圈 `i` 從 1 到 5；內圈印 `i` 個 `*`（同列）。
> 2. 內圈跑完換行（`print()` 放外圈）。

解答

```
for i in range(1, 6):
    for j in range(i):
        print('*', end='')
    print()
# *
# **
# ***
# ****
# *****
```

練習 5（重要題）：break 找第一個能被 7 整除的數

> **引導步驟**
>
> 1. 用 `for n in range(20, 50)` 走訪。
> 2. 遇到第一個 `n % 7 == 0` 就印出並 `break`。

解答

```
for n in range(20, 50):
    if n % 7 == 0:
        print('第一個能被 7 整除的是', n)    # 21
        break
```

§ 7　自我檢核

- ☐ 能判斷何時用 `for...in`（次數明確）、何時用 `while`（次數由結果決定）。
- ☐ 會用 `for...in` 走訪字串/串列等可迭代物件，知道要冒號與縮排。
- ☐ 記得 `range` 三種寫法且**含頭不含尾**，倒序要用負梯級。
- ☐ 會用切片 `物件[起始:結束:梯級]`，知道序號從 0 起、`-1` 是末、含頭不含尾。
- ☐ 會寫 `while` 並放能改變條件的程式碼，知道漏了會**無窮迴圈**。
- ☐ 會用 `in`／`not in` 當 while 條件。
- ☐ 理解巢狀迴圈：外圈每回合內圈跑完一輪（會寫九九乘法表）。
- ☐ 分得清 `break`（結束迴圈）與 `continue`（跳過本回合），知道兩者只作用於所在那一層。