

迴圈

Hanly 《C 語言詳論》6e

§ 1 名詞速查表

中文	English	一句話定義	科目	章節
迴圈	loop	用程式碼控制重複執行	程式語言一	迴圈
計數器	counter	控制迴圈次數的變數，常用 <code>i</code>	程式語言一	迴圈
累加器	accumulator	在迴圈外初始化、迴圈內累加結果的變數	程式語言一	迴圈
巢狀迴圈	nested loop	迴圈主體裡再放一個迴圈	程式語言一	迴圈
中斷	break	立刻跳出整個（最近的）迴圈	程式語言一	迴圈
跳過	continue	跳過本圈剩下的、直接進下一圈	程式語言一	迴圈
無窮迴圈	infinite loop	條件永遠為真、停不下來（Ctrl+C 中斷）	程式語言一	迴圈
差一錯誤	off-by-one	邊界多跑或少跑一次（ <code><</code> vs <code><=</code> ）	程式語言一	迴圈

§ 2 核心概念

核心概念

迴圈讓電腦替你重複做事。for 的三段：

```
for (初始化; 條件; 更新) { 主體 }
```

執行順序：**初始化（只一次）** → **[檢查條件 → 真就執行主體 → 更新]** → **[條件…]** → … → **條件為假 → 離開**。while 是「只有條件」的迴圈；do-while 先做一次再檢查（至少跑一次）。有了 if + 迴圈，就能寫出大部分程式邏輯。

§ 3 主要內容

3.1 for 與常見寫法

```
for (int i = 0; i < n; i++) // i = 0..n-1, 共 n 次 (最常用, 配陣列索引)
for (int i = 1; i <= n; i++) // i = 1..n
for (int i = n; i >= 1; i--) // 倒數
for (int i = 0; i < n; i += 2) // 跳著數 (偶數)
```

慣例從 0 開始、用 `i < n` (比 `i <= n-1` 直覺)。 `i < 5` 跑 5 次，離開時 `i=5`。

3.2 while 與 do-while

```
while (cond) { ... } // 先檢查再做, 可能一次都不做
do { ... } while (cond); // 先做一次再檢查, 至少做一次
```

次數已知用 `for`；依條件 (如「讀到 -1 才停」) 用 `while`。

3.3 累加器 pattern (最常用)

```
int sum = 0; // ① 外面初始化 (累加用 0、累乘用 1)
for (int i = 1; i <= 100; i++)
    sum += i; // ② 裡面累加
printf("%d\n", sum); // ③ 外面使用 → 5050
```

3.4 break 與 continue

`break` 立刻跳出整個迴圈；`continue` 跳過本圈剩下的、直接進下一圈。

```
for (int i = 1; i <= 10; i++) {
    if (i == 5) break; // i 到 5 就整個停
    if (i % 2 == 0) continue; // 偶數跳過, 不印
    printf("%d ", i); // 印 1 3
}
```

3.5 巢狀迴圈

迴圈裡再放迴圈：外層跑 1 次、內層跑完整一輪，總次數 = 外 × 內。像時鐘：分針 (內) 轉一圈，時針 (外) 才動一格。

```
for (int i = 1; i <= 9; i++) { // 九九乘法表
    for (int j = 1; j <= 9; j++)
        printf("%d*%d=%-4d", i, j, i*j);
    printf("\n");
}
```

3.6 printf 對齊 (排版)

格式	意義	值=42 → 輸出
<code>%d</code>	預設	42
<code>%5d</code>	靠右佔 5 格	42
<code>%-5d</code>	靠左佔 5 格	42
<code>%05d</code>	補零佔 5 格	00042

3.7 圖形：用內層條件控制每行長度

```
for (int i = 1; i <= n; i++) {           // 直角三角形
    for (int j = 1; j <= i; j++)       // 第 i 行印 i 個
        printf("*");
    printf("\n");
}
```

關鍵：**外層控制行數**、**內層控制每行內容**；內層終止條件 `j <= i` 決定第 `i` 行的星號數。倒三角把外層改倒數；等腰三角先印 `n-i` 個空格再印 `2*i-1` 個星號。

§ 4 語法與函式速查

```
for (init; cond; update) { ... }
while (cond) { ... }
do { ... } while (cond);
break;           // 跳出整個迴圈
continue;       // 跳到下一圈
// printf 對齊：%5d 靠右 / %-5d 靠左 / %05d 補零
```

§ 5 常見錯誤

常見錯誤

- 無窮迴圈：忘了 `i++`，或更新方向錯 (`i--` 卻要往上)。Ctrl+C 中斷。
- 差一錯誤 (off-by-one)：想印 1~10 卻寫 `i < 10` (只到 9)，應 `i <= 10`。
- `for (...)`；後面多了分號 → 空迴圈，主體變成只執行一次。
- 累加器忘了初始化 (垃圾值，結果錯，接 [[變數與記憶體]] 的 UB)。
- 混淆 `break` (跳出整個迴圈) 與 `continue` (只跳過本圈)。
- 巢狀迴圈內外層都用 `i` (變數名撞)，用不同名 `i`、`j`。

§ 6 練習題

練習 1 (一般題) : for 追蹤

寫出輸出。

```
for (int i = 10; i > 0; i -= 3) printf("%d ", i);
```

引導步驟

1. 從 10 開始每次減 3，條件 $i > 0$ 。

解答

```
10 7 4 1
```

10→7→4→1 (1>0 真，印1) → -2 (不>0，停)。

練習 2 (一般題)：巢狀計數

count 的值是多少？

```
int count = 0;
for (int i = 1; i <= 3; i++)
    for (int j = 1; j <= 2; j++)
        count++;
printf("%d\n", count);
```


解答

6。外層 3 次 × 內層 2 次 = 6。

§ 7 自我檢核

- 會寫 for (init/cond/update) ，知道執行順序與離開時計數器的值。
- 分得清 for / while / do-while 的適用情境。
- 會用累加器 pattern (外初始化、內累加、外使用) 。
- 分得清 break (跳出整個迴圈) 與 continue (跳過本圈) 。
- 懂巢狀迴圈總次數 = 外 × 內，會印九九乘法表、三角形。
- 會用 printf 對齊 (%5d / %-5d / %05d) 。
- 會抓無窮迴圈、off-by-one、for 後加分號這些陷阱。