4.4 for 迴圈

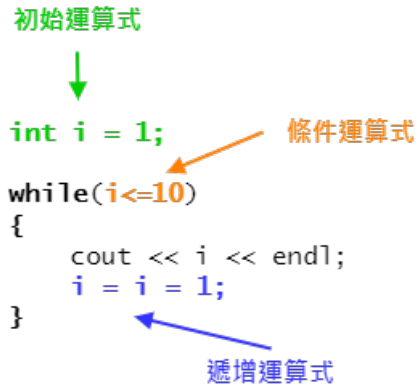
while 和 do...while 迴圈很適合用在「你知道什麼條件下迴圈要繼續或停止」，因為決定是否再繞一圈的就是一個條件判斷式。

但是在你很清楚一共要繞幾圈的情況下，使用接下來介紹的 for 迴圈，會輕鬆很多。

for 迴圈

使用 while 迴圈來繞指定圈數，我們多採用這樣的架構，其中變數 i 擔任計數器，我們會：

1. 指定計數器的初始值
2. 每圈檢查計數器的值是否仍符合條件
3. 每圈遞增計數器的值



for 迴圈可以一次搞定這三者。

for 的基本語法

```
for(初始運算式;條件運算式;遞增運算式)
{
    ......
    要重覆做的事情
    ......
}
```

以輸出 1~10 為例，程式看起來比較簡潔，而且還是很清晰。

練習：輸出 1 ~ 10

```
for(int i=1; i<=10; i=i+1)
{
    cout << i << endl;
}
```

```
1
2
3
4
5
6
7
8
9
10
```

練習：輸出 n 的所有正因數

因為 n 的所有正因數是 1~n 之間的整數，所以我們用一個 for 迴圈來遍歷整個區間做篩選。

```
int n = 16;

cout << n << "的正因數有：";

for(int i=1; i<=n; i++)
{
    if(n%i==0)
    {
        cout << " " << i;
    }
}
cout << endl;
```

```
16的正因數有： 1 2 4 8 16
```

變數的生命週期

輸入以下這段程式後編譯執行，在編譯時期就會發生錯誤。

```
#include <iostream>

using namespace std;

int main()
{
    for(int i=1; i<=5; i++)
    {
        cout << i << endl;
    }

    cout << "now i=" << i << endl;

    return 0;
}
```

```
12:25: error: 'i' was not declared in this scope
    cout << "now i=" << i << endl;
                    ^
```

錯誤訊息表示在 12 行那邊使用到變數 i 但是沒有宣告。但是你往上看會覺得「明明我在第7行，for 迴圈那裡一開始就宣告了啊」。

仔細看一下錯誤訊息第一行末的 - "in this scope"，他是說你沒有在這個 scope 裡宣告 i。這個 scope 是什麼意思呢？

我們來看一下這個程式：

```
#include <iostream>

using namespace std;

int main()
{
    {
        int i=5;
        cout << "1: i=" << i << endl;
        i=i+1;
    }

    cout << "2: i=" << i << endl;

    return 0;
}
```

第 8, 9, 10 行被放在一組大括號裡，整個大括號範圍可以視為一個程式區塊(block)。這個區塊算是一個 **區塊範圍**

(block scope)，宣告在這個區塊裡的變數屬於 **區域變數(local variable)**，該變數的生命週期始於宣告完成，終於離開區塊。

所以在第 8 行開始，到第 11 行結束的範圍內，都可以存取變數 i 的值。但是在第 12 行開始，或第 7 行之前，都看不到也無法存取這個變數 i 的值。

試著編譯並執行這個程式，你會發現第 9, 10 行存取 變數i 都沒有問題，但是第 13 行會發生編譯錯誤，編譯器會抱怨變數 i 沒有在這個 scope 裡宣告。

若是把大括號拿掉，變成這樣。

```
#include <iostream>

using namespace std;

int main()
{
    int i=5;
    cout << "1: i=" << i << endl;
    i=i+1;

    cout << "2: i=" << i << endl;

    return 0;
}
```

現在整個程式只剩下一個 block，即第 6 ~ 14 行。所以變數 i 的生命週期始於第 7 行，終於第 14 行。程式輸出結果如下。

```
1: i=5
2: i=6
```

for 敘述(包含整個大括號範圍)也是一個 block scope，所以如果我們在 for 裡面宣告變數，它的生命週期也只限於該 for 迴圈內。

一般來說若只是單純用於迴圈的計數器，我們會像這樣把它宣告在 for 敘述裡。

```
    for(int i=1; i<=5; i++)  // 宣告在 for 敘述裡面
    {
        ......
    }
```

若是該變數在迴圈結束之後還有用處，我們會把它宣告在 for 迴圈的外面。

```
#include <iostream>

using namespace std;

int main()
{
    int i;  // 宣告在 for 敘述外面

    for(i=1; i<=5; i++)
    {
        cout << i << endl;
    }

    cout << "now i=" << i << endl;

    return 0;
}
```

```
1
2
3
4
```

```
5
now i=6
```

關於 scope 的詳細說明，有興趣的話可以先看一下這份文件 - scope。以後我們會另外開一個主題做更全面的討論。

Online judge 讀取 n 筆測資

在競程的題目中，有一種測資型式是這樣的。

輸入說明：

輸入的第一行有一個整數 t。接下來的 t 行每行有一個正整數 y，代表西元年份。

範例輸入：

```
4
1992
1993
1900
2000
```

這種情形就很適合使用 for 迴圈來讀取測資。

```
int n;
cin >> n;

for(int i=0; i<n; i++)
{
    int year;
    cin >> year;
    // do something
}
```

Revision #26
★Created 12 March 2024 04:04:25 by huihui
Updated 27 October 2025 07:01:58 by huihui