

3.5 三元運算子 ? :

3.5.1 三元運算子 ? :

在 C++ 中，三元運算子 (Ternary Operator) 是唯一一個需要三個運算元的運算子。它的符號是 ? 和 :。

這個運算子主要用來取代簡單的 if-else 判斷式，讓程式碼在一行內就能完成條件判斷與賦值，非常方便。

語法

三元運算子的基本語法結構如下：

```
條件式 ? 運算式1 : 運算式2;
```

條件式 (Condition): 這是一個會回傳 true (真) 或 false (假) 的布林表達式。

- **? :** 這是三元運算子的核心符號。
- **運算式1 (Expression1):** 如果「條件式」的結果為 true，則執行這個運算式，並將其結果作為整個三元運算式的最終結果。
- **運算式2 (Expression2):** 如果「條件式」的結果為 false，則執行這個運算式，並將其結果作為整個三元運算式的最終結果。

3.5.2 實例練習

讓我們來看幾個例子，比較一下使用 if-else 和使用三元運算子的差別。

範例：判斷奇偶數

假設我們要讓使用者輸入一個整數，然後判斷它是奇數還是偶數。

傳統的 if-else 寫法：

```
int number;
cout << "請輸入一個整數: ";
cin >> number;

string result;
if (number % 2 == 0) {
    result = "偶數";
} else {
    result = "奇數";
}

cout << "這個數字是 " << result << endl;
```

使用三元運算子的寫法：

```
int number;
cout << "請輸入一個整數: ";
cin >> number;

// 一行就搞定！
string result = (number % 2 == 0) ? "偶數" : "奇數";

cout << "這個數字是 " << result << endl;
```

解說：

在三元運算子的版本中，(number % 2 == 0) 是我們的條件式。

- 如果 number 除以 2 的餘數為 0 (條件為 true)，則回傳 ? 後面的字串 "偶數"。
- 如果餘數不為 0 (條件為 false)，則回傳 : 後面的字串 "奇數"。

- 最後，回傳的字串會被直接賦值給 `result` 變數。是不是簡潔很多呢？

你還可以進一步像這樣直接使用他的運算結果

```
int number;
cout << "請輸入一個整數: ";
cin >> number;

// 注意：要用 ( ) 包覆整個運算式
cout << "這個數字是 " << ((number % 2 == 0) ? "偶數" : "奇數") << endl;
```

範例：絕對值計算

```
int a;
cout << "請輸入一個整數 a: ";
cin >> a;

cout << "|a|= " << ((a < 0) ? -a : a) << endl;
```

3.5.2 使用時機與注意事項

- **優點：** 程式碼簡潔。
- **缺點：** 不適合處理複雜的邏輯。如果 `if` 或 `else` 區塊中需要執行多行程式碼，就不應該使用三元運算子，否則會讓程式碼變得難以閱讀和維護。
- **原則：** 當 `if-else` 只是為了根據一個簡單條件來賦予變數不同的值時，就是使用三元運算子的最佳時機。

三元運算子 `?:` 是一個非常實用的語法糖 (Syntactic Sugar)，能讓你用更精簡的方式寫出條件判斷。熟練使用它可以提升程式碼的美觀與效率，但切記不要濫用，以免降低複雜邏輯的可讀性。

🕒Revision #1

★Created 27 October 2025 00:02:21 by huihui

✎Updated 27 October 2025 00:38:16 by huihui