

雙重選擇



C/C++ 語言提供三種選擇結構

選擇結構	描述
if選擇敘述式	單一選擇敘述式，只選擇或跳過一項動作
if...else選擇敘述式	雙重選擇敘述式，會取兩種不同動作之一
switch	依運算式的不同，選擇執行許多動作之一

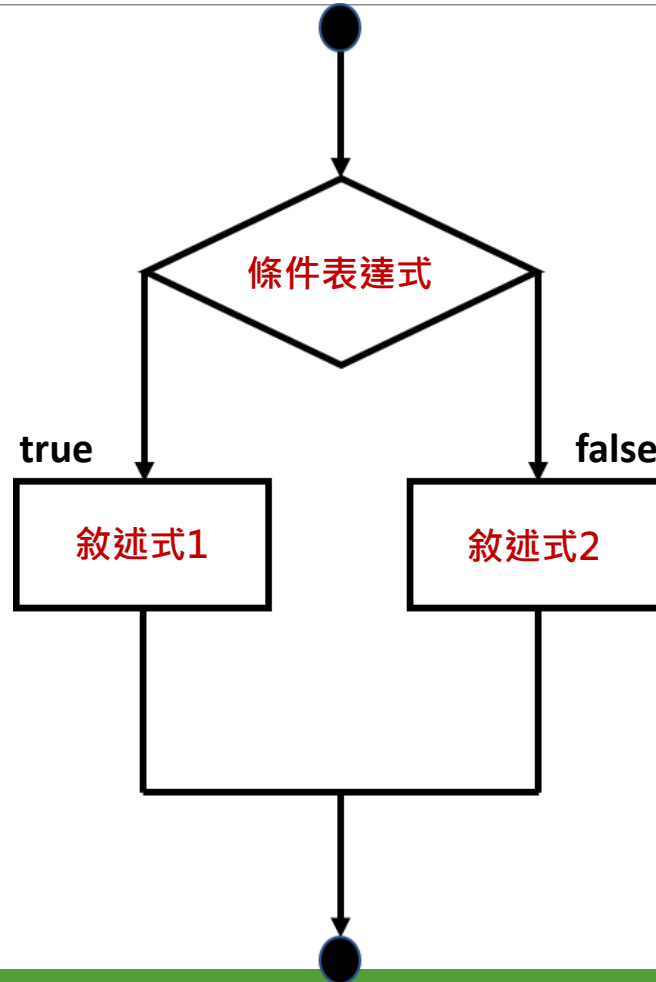
本單元主要學習if...else選擇敘述式



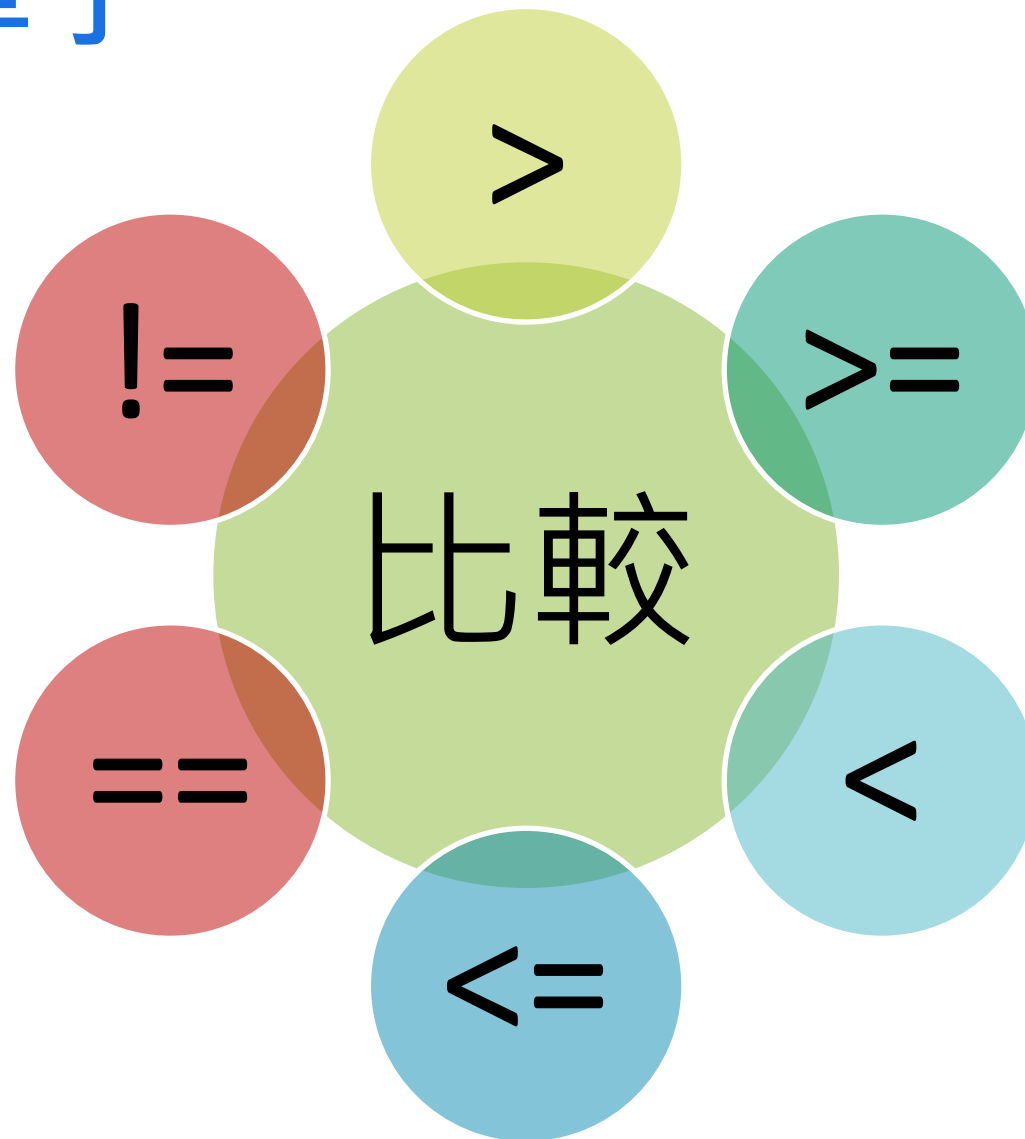
不是這樣
就是那樣

if...else

雙重選擇結構流程圖



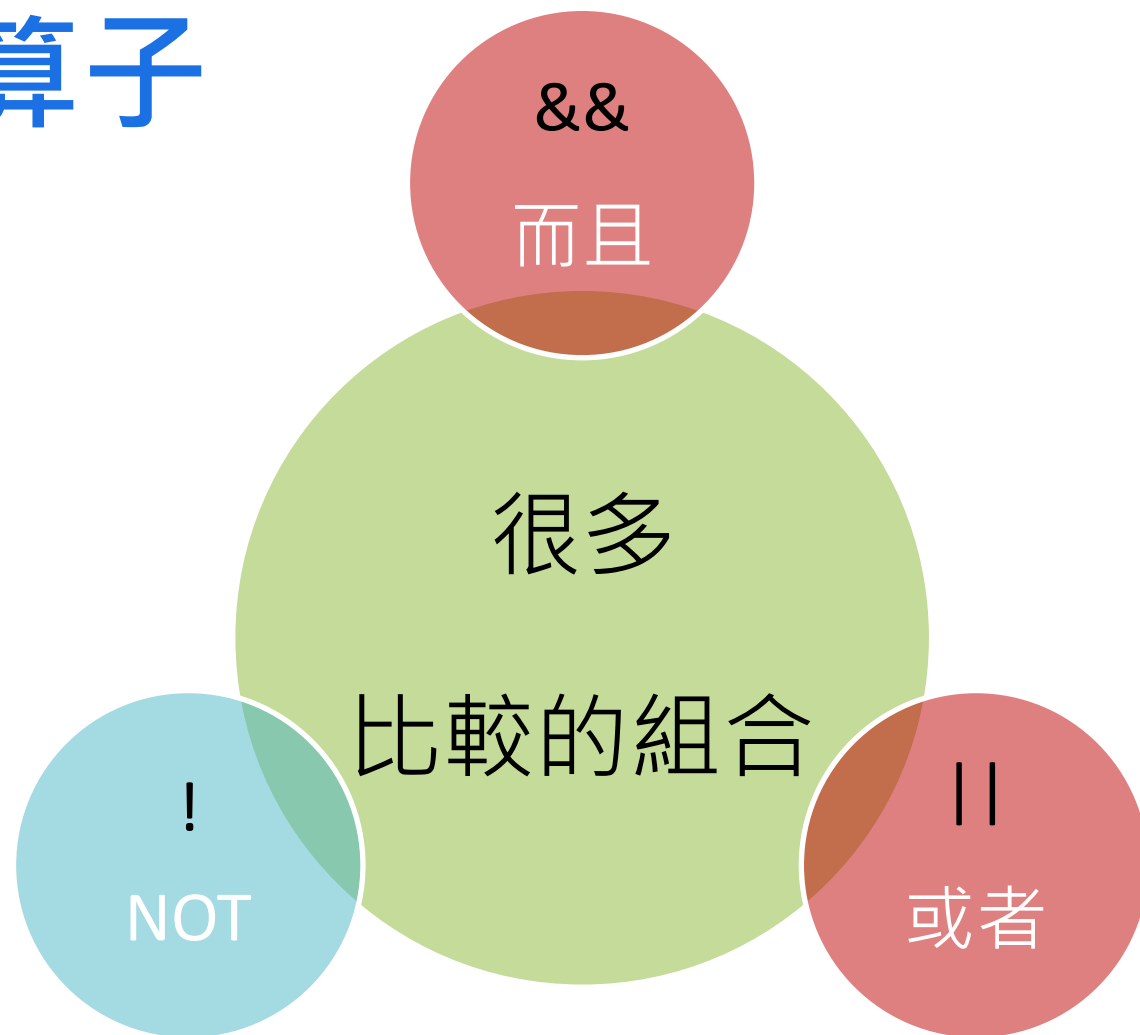
關係運算子



關係運算子與程式碼

代數符號	C語言符號	C的條件式範例	C條件式的意義
\neq	<code>!=</code>	<code>x!=y</code>	x不等於y
$=$	<code>==</code>	<code>x==y</code>	x等於y
$>$	<code>></code>	<code>x>y</code>	x大於y
$<$	<code><</code>	<code>x<y</code>	x小於y
\geq	<code>>=</code>	<code>x>=y</code>	x大於等於y
\leq	<code><=</code>	<code>x<=y</code>	x小於等於y

邏輯運算子



邏輯運算子

而且

運算符號	意涵	範例	說明	一元或二元運算
&&	and 運算	a && b	a與b做 and邏輯運算	二元
	or運算	a b	a與b做 or邏輯運算	二元
^	xor運算	a ^ b	a與b做 xor邏輯運算	二元
!	not運算	! a	a做not邏輯運算	一元

或者

xor

如果a、b兩個值不相同，則xor結果為1。
如果a、b兩個值相同，xor結果為0。

邏輯運算子運算實例

a	b	a && b	a b	a xor b	not a
0	0	0	0	0	1
0	1	0	1	1	1
1	0	0	1	1	0
1	1	1	1	0	0

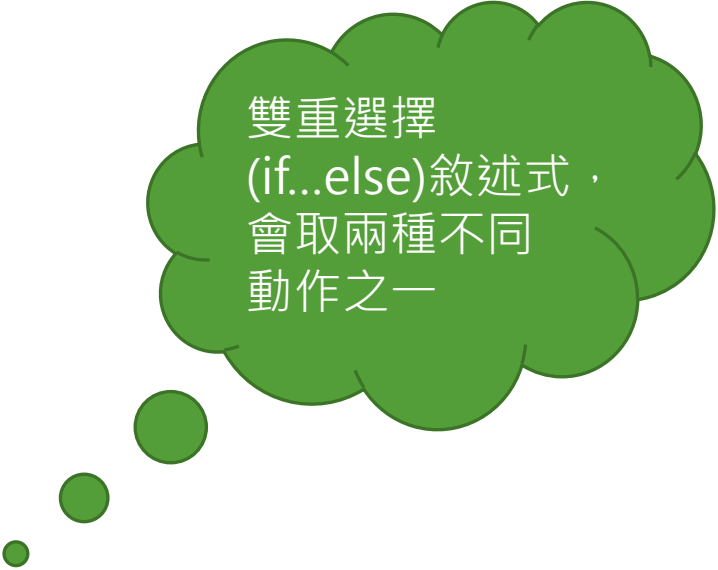
寫一個程式，判斷一個數字為奇數或偶數？

問題



參考程式碼

```
int main()
{
    int num;
    cin >> num;
    if(num%2==0)
        cout << num << "是偶數" << endl;
    else
        cout << num << "是奇數" << endl;
    return 0;
}
```



雙重選擇
(if...else)敘述式，
會取兩種不同
動作之一

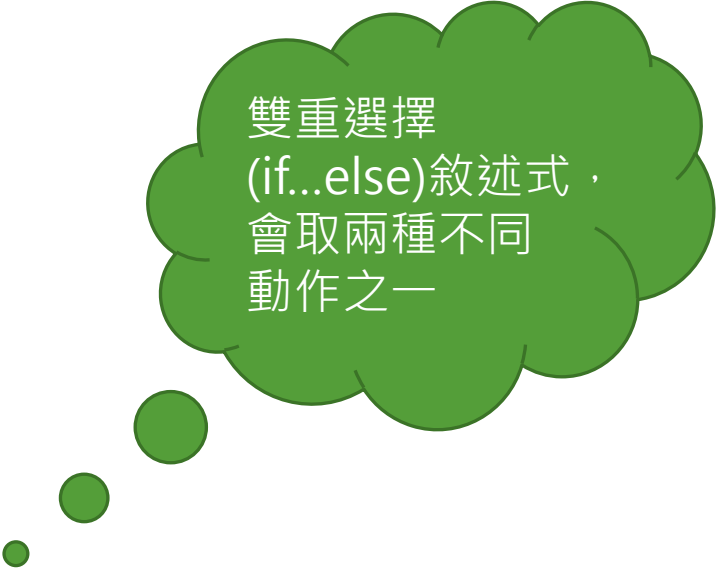
寫一個程式，判斷一個分數是否及格(60分以上)？

問題

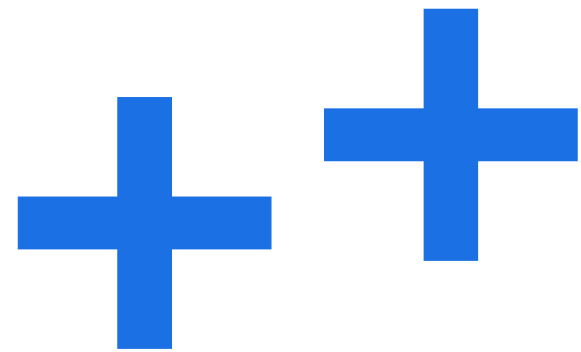


參考程式碼

```
int main()
{
    int score;
    cin >> score;
    if(score >= 60)
        cout << "及格" << endl;
    else
        cout << "不及格" << endl;
    return 0;
}
```



雙重選擇
(if...else)敘述式，
會取兩種不同
動作之一



延伸學習



使用 三元運算子 表達 if...else

```
int main() {  
    int score;  
    cin >> score;  
    cout << ((score >= 60) ? "及格" : "不及格") << endl;  
    return 0;  
}
```

條件運算子是C/C++中唯一的三元運算子-它使用三個運算元

第一個運算元是條件

第二個運算元是當條件為真實，整個條件運算式的值

第三個運算元是當條件為假時，整個條件運算式的值

再談寫作風格

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int num; /*宣告整數變數*/
6     cin >> num; /*輸入整數變數*/
7     if (num % 2 == 0)
8         cout << num << "是偶數" << endl;
9     else
10        cout << num << "是奇數" << endl;
11     return 0;
12 }
```

- if與else的下方分別包含一個區塊，區塊是由大括號{ }所包圍。if與else下方如果只有一個句子，大括號{ }是可以被省略的。
- 第8行與第10行有縮排，會讓程式的可讀性高一點。

區塊如果包含2行以上的語句，要加上{ }

```
1 if(條件1 || 條件2) {  
2     cout<<"符合標準"<<endl;  
3     cout<<n<<endl;  
4 }
```

學會將問題縮小範圍，逐步找出錯誤

```
1 #include <iostream>
2 using namespace std;
3 int main()
4 {
5     int x, a, b, c;
6     cin>>x;
7     a=x/100;
8     //b=
9     //c
10    if(a%3==0)
11        cout<<"百位數"<<a<<"是3的倍數"<<endl;
12    else
13        cout<<"百位數"<<a<<"不是3的倍數"<<endl;
14    //if
15    | //cout
16    //else
17    | //cout
18    //if
19    | //cout
20    //else
21    | //cout
22    return 0;
23 }
```

- 這個題目涵蓋3個項目，如果程式有問題，可以將答案拆成3個部分，一一尋找錯誤。一旦錯誤被找出來，使用紙筆做一次演算，問題很快會被發現。
- 分段檢視程式代碼的方法，可以**試試註解的功能，加上「//」**，後面的程式碼會被編譯器忽略。